*Daniel Christopher Swain*

*PropertyDome (Group 3) | Perth, Western Australia*

*Property Website*

Handover Report

Table of Contents

[1.0 - Introduction 3](#_Toc453590840)

[2.0 - Functionality 3](#_Toc453590841)

[*2.1 - Index/Buy page* 3](#_Toc453590842)

[*2.2 - Sell Pages* 4](#_Toc453590843)

[*2.3 - Search Page* 4](#_Toc453590844)

[*2.4 - Property Page* 5](#_Toc453590845)

[*2.5 - Admin Page* 5](#_Toc453590846)

[*2.6 - Login/Sign-up Pages* 6](#_Toc453590847)

[*2.7 - User Dashboard* 6](#_Toc453590848)

[*2.8 - Seller Listing Page* 7](#_Toc453590849)

[*2.9 - Static Pages* 7](#_Toc453590850)

[3.0 - Testing 7](#_Toc453590851)

[*3.1 - Test 1: User Login* 7](#_Toc453590852)

[*3.2 - Test 2: Suburb/Postcode invalidation on Sell Page* 8](#_Toc453590853)

[4.0 - Validation 9](#_Toc453590854)

[*4.1 Functional Requirements* 9](#_Toc453590855)

[*4.2 Test Results* 10](#_Toc453590856)

[*4.2.1 - Test 1 Results: User Login* 10](#_Toc453590857)

[*4.2.2 - Test 2 Results: Suburb/Postcode invalidation on Sell Page* 11](#_Toc453590858)

[5.0 - Conclusion 13](#_Toc453590859)

# **1.0 - Introduction**

The following document functions as the handover for the completed property website built by Group 3 – PropertyDome (currently hosted at [app.slehmann36.com](http://app.slehmann36.com/)). It details the completed functionality, the testing performed on the site as well as the validation and test performance.

# **2.0 - Functionality**

The final version of the site has the following site structure:

* Index/buy page.
* Sell pages.
* Search page.
* Property page.
* Admin page.
* Login/Sign-up pages.
* User Dashboard.
* Seller Listing page.
* Static Pages (Meet the team, privacy policy, terms and condition, contact page)

The functionality for each page is detailed below. Unless otherwise noted each page(s) was completed with all the requirements as set out in the initial client brief and built to the design as approved by the client on the 6th of April.

All pages have been built to be responsive and handle changes to their viewport size elegantly switching from desktop optimised to tablet and mobile optimised layouts.

## *2.1 - Index/Buy page*

The index/buy page is the first view to the site and the default redirect for certain actions. The page contains the following views and functionality:

* Navigation Bar and Footer are set up here (they appear on all subsequent pages).
* Search panel containing the main state/suburb search bar and the additional features button allowing the users to search by:
  + Suburb/State,
  + Price (either range or fixed price)
  + Bedrooms (either range or single number)
  + Bathrooms (either range or single number)
  + Car Spaces (either range or single number)
  + Land Size (either range or single number)
  + Indoor Features (search not functional at this time, tags can be added)
  + Outdoor Features (search not functional at this time, tags can be added)
* Stats panel/Favourited Properties: this panel shows advertising stats to visitors that aren’t logged in otherwise it shows the logged in user’s favourited properties (or provides a link to search if they don’t have any).
* Featured Properties: a panel were listings that have payed to be featured are shown to all visitors on the main page.
* Company Twitter Feed: a panel showing the latest tweets by the company.

## *2.2 - Sell Pages*

The sell pages include the add listing, edit listing and manage listing pages. When a visitor is not logged in or they don’t have any properties they will see a simple page suggesting they add a property and the reasons they would want to list on this website.

Once a user has a property they will see the following pages and have the following functionality:

* Manage page: this is the main page a user sees when they have properties for sale. This page shows all their listings in an infinitely scrolling and sortable table. Each listing is shown with a short summary, the main image, address, price (or price range) and key listing details. The listing status will also show (i.e. whether it is approved, rejected, pending review or expired). This table allows a user to add more listings or edit/delete an existing listing.
* Add Listing Page: this is a simple page with 4 tabs (Listing Information, Photographs, Extras, Preview). This page is where a user enters their property details and submit it for review and listing on the site. Each tab has the following functionality:
  + Listing Information: this is the tab where the user enters the property listing details that were specified in the client brief (address, suburb, bedrooms…). A user can also add additional feature tags (Indoor features, Outdoor Features, Leisure features…) from a curated list. These features are intended to flesh out a listing and provide more options for seller/agents.
  + Photographs: this tab allows users to upload single or multiple images as well as set the main image for their listing or delete old images. Up to 10 images can be added for free.
  + Extras: this tab is where the user would select premium features such as having their listing featured, or allowing more than 10 photo uploads. This tab is built but not functional.
  + Preview: this tab shows the listing as it would appear in search results. Currently the information is only shown when editing the listing.
* Edit Listing Page: this page is identical to the Add Listing Page except it pulls the information for the listing being edited from the database, allowing the user to edit and update their listing.

## *2.3 - Search Page*

The search page shows the search results from the search criteria entered on the index/buy page as well as allowing visitors to update, add to or remove aspects of their search criteria. The search functionality from the buy/index page is fully replicated on this page.

A visitor can enter the search page with no query, in which case the most recently added listings will be shown until the visitor enters a query on the search page.

The search page has the following additional features:

* Search results are paginated with only 5 showing initially, before loading more when the user reaches the bottom of the list using infinite scrolling.
* Search results only include valid listings that have been approved for posting by an administrator and are not expired listings.
* Search tags can be added/removed on the page, re running the search with the new query.
* Search tags search for exact matches if only one exists or a range match if two or more of the same tag type exist (i.e. 2 bedroom tags will result in a bedroom ranged search between the two chosen tag values).
* If a suburb is used for search then once the user has reached the end of the search results the search function will load search results for surrounding suburbs, broadening the results for the user automatically. This repeats until a limit of 8 surrounding suburbs.
* Search results show the agent’s picture and details and can be favourited by a logged in user.

## *2.4 - Property Page*

The property page is the page a visitor sees when clicking on a search result or when they are shared a listing link. It shows all the inputted information for the listing as set by the agent/seller and approved by the administrators.

The property page has the following functionality:

* Only visible if the property is approved by the admin.
* Shows the listing images in a carousel (each image can be clicked to bring up a full screen slideshow).
* Shows the listing details and additional features.
* Shows the listing agent/seller information and provides a link to their other properties.
* Allows a user to favourite the property for status updates.
* Shows the location of the property on a map which allows the user to search for nearby places (such as schools, shops, police stations) and view the surrounding area.
* Provides a link back to the previous page, with the link clever enough to return a visitor to the search page with their previous existing query active.

## *2.5 - Admin Page*

The admin page is restricted to user accounts that are flagged in the database as having admin privileges. The admin page shows listings that are pending approval, have been rejected or have expired and allows an admin to perform the following actions on any listing:

* View the property page for the listing. This allows the admin to view the page as it would be viewed by a user.
* Approve a listing. This will mark the listing as approved and allow it to show in search results.
* Reject a listing. This will reject a listing, removing it from search results and showing the agent/seller it has been rejected, allowing them to edit it for future review.
* Delete a listing. This deletes the listing, the photos, favourites and any other data for that listing from the database.
* Note, an admin cannot perform these actions on any listings that they have created.

## *2.6 - Login/Sign-up Pages*

The login/sign-up pages are accessed from the navigation menu and appear as a modal on desktop/large screen devices and as dedicated pages on mobiles and small screens.

The login/sign-up pages have the following functionality:

* Register a new user with their email address and password.
* Login an existing user with their email address and password
* Allow users to reset their password/send a password reminder
* Allow a user to set them as remembered so they retain their login on subsequent visits to the site.
* Provide error messages to the user for incorrect details.
* On successful login a user will be redirect to the previous page but logged in (i.e. if they were on the sell page then when logged in they will redirect back to that page as a logged in user).
* On successful registration the new user will be redirected to the user dashboard with a prompt to fill in any additional details.

## *2.7 - User Dashboard*

The user dashboard is where a user views their favourited listings, can see the latest status updates from their favourited listings and can update their details to add additional information like username, phone number and name.

The user dashboard has the following functionality:

* Activity feed: this tab shows the latest status updates from the properties the user has favourited.
* Favourited: this tab shows all the properties a user has favourited and includes links to their listing pages.
* User Settings: this tab allows a user to update their details including the following actions:
  + Upload a profile picture
  + Crop a profile picture
  + Update their name, username, company name and phone number
  + Change their email and password
  + Delete their account.

## *2.8 - Seller Listing Page*

The seller listing page shows all the listings that the selected user has active on the site. It uses the search results view to show the listings for the user in a simple table view. Clicking on a listing will take the visitor to that listings property page.

## *2.9 - Static Pages*

The static pages include the privacy policy, terms and conditions, meet the team and contact pages. They can be accessed from the site wide footer and navigation bar.

Each site has the following functionality:

* Privacy Policy: details the site’s privacy policy for visitors/users.
* Terms and Conditions: details the site’s terms and conditions for visitors/users.
* Meet the Team: shows the development team and their roles/profiles.
* Contact Us: allows a visitor to contact the site using a user form. The user will receive an email confirmation when the message is sent to the site.

# **3.0 - Testing**

The testing for this project was a continuous process and included dedicated test criteria along with user testing where developers tested each module during development, particularly modules from other developers.

GitHub was utilised for testing of development functions and modules, with GitHub issues created for any bugs or problems identified. During the project a total of 48 GitHub issues were created with 46 closed at the time of this report. These issues ranged from JavaScript functions firing twice on page load ([link](https://github.com/slehmann36/Group3-Real-Estate-Site/issues/156)), to browser specific issues for browsers such as Safari or Firefox ([Safari issue](https://github.com/slehmann36/Group3-Real-Estate-Site/issues/120), [Firefox issue](https://github.com/slehmann36/Group3-Real-Estate-Site/issues/138)). To see the project issues please visit this [link](https://github.com/slehmann36/Group3-Real-Estate-Site/issues).

In addition to the continuous testing completed throughout the development of the project we also completed discrete tests for individual functions and functionality. These tests helped validate our assumptions and confirm that the functionality being added was robust.

The following are two such tests performed during the development of the project. See section 5.0 – Validation for the test performance.

## *3.1 - Test 1: User Login*

The user login functionality was completed using the Devise gem. This gem handles the authentication checking (validating the username and password with the database records) as well as producing error messages for the user.

A user logs into the website using a login modal that is accessed from the navigation bar and uses the Devise gem for user verification and login. The following test cases were used to verify the performance of the login modal. They list the test action that was performed, as well as the expected results from that action.

The account used for the following tests was. email: [nic2015@email.com](mailto:nic2015@email.com) and password: L3tsBu1ld

|  |  |  |  |
| --- | --- | --- | --- |
| # | Task | Expected Result | Successful |
| 1.1 | Log in using correct user details (email + password) | * This should log the user in; * Update the nav menu with their email address/username; * Show “My Dashboard” instead of “Contact Us” in the nav menu; and * Redirect the user to the previous page they were on when they clicked log in. | * Y * Y * Y * Y |
| 1.2 | Log in using incorrect email address (correct password) | * This should redirect user to root\_url/login; * Show the following error: “Incorrect email address or password.”; * The password field should be cleared; and * When successfully logged in it should redirect the user to root\_url/buy. | * Y * Y * Y * Y |
| 1.3 | Log in using incorrect password (correct email address) | * This test has the same expected outcomes as test 2. | * Y |
| 1.4 | Log in using incorrect email and password | * This test has the same expected outcomes as test 2. | * Y |

Test methods and passing criteria for the Login functionality (see 4.2.1 for test results)

## *3.2 - Test 2: Suburb/Postcode invalidation on Sell Page*

A useful functionality on our website when a user is adding or editing a property listing is the ability to select the suburb and postcode for the property from a large database of choices. This helps ensure the integrity of the listing data as well as provides consistency for our search functions as we don’t need to cover for Perth, perth, Perth (WA) and perth (WA).

This was completed via a JavaScript function that fires when the user selects a state in the Add/Edit page. When a state is chosen from the state dropdown the suburb and postcode choice lists are invalidated and an Ajax call is sent to the server to get the list of suburbs and postcodes for the new state selection. When this call succeeds the suburb and postcode choices are filled with these new values, ensuring only choices within the chosen state can be made.

This function is a critical feature as it provides a front end check for the data integrity of the listing before it is sent to the server. The following test cases were used to verify the performance of this JavaScript function They list the test action that was performed, as well as the expected results from that action.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Task | Expected Result | Successful |
| 2.1 | On the add view, select a state | * This should clear any selections from the Suburb and postcode field; * Change the suburbs to suburbs from the chosen state; and * Change the postcodes to postcodes from the chosen state. | * Y * Y * Y |
| 2.2 | On the edit view, change a state with data already present in the suburbs/postcode fields | * This test has the same expected outcomes as in test 1. | * Y |
| 2.3 | On the add view, change the suburb selection after a state has been chosen. | * This should clear any selections from the postcode field; and * Change the postcodes field choices to postcodes that match the chosen state and suburb. | * Y * Y |
| 2.4 | On the edit view, change the suburb selection after a state has been chosen | * This test has the same expected outcomes as in test 3. |  |

Test methods and passing criteria for the Suburb/Postcode invalidation functionality (see 4.2.2 for test results)

The results for both sets of testing criteria can be viewed in section 4.0 – Validation. In addition to the testing performed above during the project, the final project repository also includes the full log records from the development and early production phase. These logs detail all site actions performed and provide information for http requests, server crashes, rendering issues and other performant issues. These logs were a vital part in the testing of the web site and are included to detail the continuous testing throughout the project.

# **4.0 - Validation**

## *4.1 Functional Requirements*

The client brief included the following functional requirements for the completed project:

1. Design, construction and deployment of a site with a simple interface for people to search for houses to buy and/or upload their own content and images of their house(s) for sale to a central database using the following criteria:
   1. Location: Suburb/State
   2. Price: Min/Max/Range
   3. Bedrooms: Min/Max/Number
   4. Property type: House/ Apartment/Land/Other
   5. Bathrooms: Min/Max/Number
   6. Car Spaces: set number in list
   7. Min Land: sqm
   8. Indoor features: Air con/ Alarm / etc.
   9. Outdoor features: Garage/ Swimming Pool/ Spa/ etc.
2. Website to be built to the approved design criteria received from the client on the 6th of April.
3. Website to work on latest version of major browsers (Chrome, Firefox, Safari, Internet Explorer) and be responsive on major mobile devices (iPhone 5, 6, 6+; Android phones with similar screen sizes 5 - 6”)
4. Each house posting to last up to 8 weeks after which they are expired and flagged for removal via an admin, unless a payment is made by the user to extend the post (Payments outside of scope).
5. Provision for users to upload 10 images for free. Whilst not part of our scope, extra image uploads and listing features will be able to be unlocked via additional payments by users.
6. A admin interface shall be provided for staff to approve user uploads prior to their appearance on the main site.

The completed site has satisfied all of the functional requirements except for 1(h), 1(i) and the functions marked for exclusion from scope (the payment functions). The tags for Indoor features (1(h)) and Outdoor features (1(i)) can be added to a listing and will be displayed on a listing, however they aren’t used in search.

As of this report all search functionalities are working except the identified Indoor and Outdoor feature search. These tags can be selected for use in the search query, however they won’t result in a search being performed within the database with the selected tags.

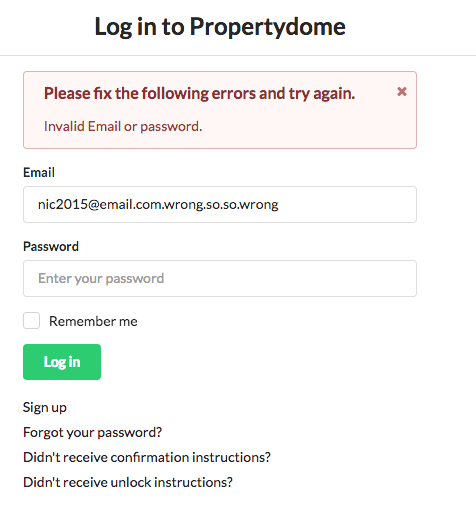
## *4.2 Test Results*

In addition to meeting the functional requirements as identified above in section 4.1 – Functional Requirements, we also performed testing on discrete functions of the website. These test criteria were detailed in section 3.1 and 3.2. The results of these tests are detailed below.

### *4.2.1 - Test 1 Results: User Login*

The site successfully passed all four tests of the Login action as built using Devise and both a modal and login page. The testing criteria also tested the redirect action of the gem as a successful login sent the user to the previous page, whereas an unsuccessful login redirect the user to the login page with a warning, then the buy page with the next successful login.

The following images show selected screenshots from the tests performed.



The resulting error message when a user enters the wrong details when logging in. Entering the correct details redirected back to /buy as expected.



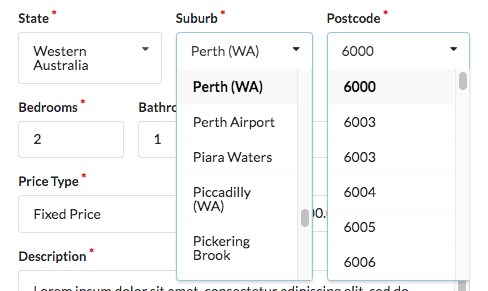


This image shows the change to the navigation menu when a user is successful logged in (logged out view is the first image).

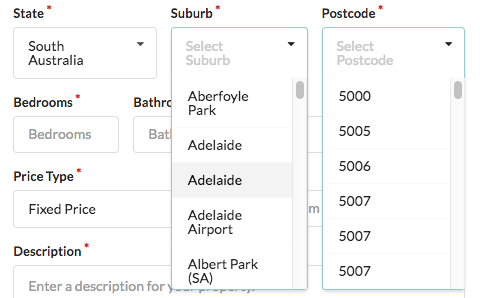
### *4.2.2 - Test 2 Results: Suburb/Postcode invalidation on Sell Page*

The site successfully passed all four tests of the Suburb/Postcode invalidation action as built for both the add and edit views on the sell listing page. This is an important front end check to help ensure that we’re getting data for a listing that is consistent as this reduces the complexity of the search methods for the website.

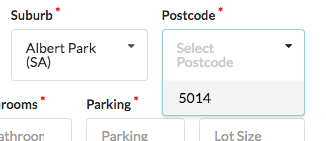
The following images show selected screenshots from the tests performed.



This image shows the field choices available to the user before the change of state.



This image shows that the suburbs and postcode choices have changed to match the chosen state.



This image shows that the postcode choices have changed to match those that are valid for the chosen state and suburb.

# **5.0 - Conclusion**

The site was completed to the functional requirements as specified in the client brief (excluding the Indoor and Outdoor feature search as mentioned in section 4.1). Each page and function has been tested either by developers during development (as shown via the GitHub issues page) or using dedicated discrete testing criteria as detailed in section 3.1 and 3.2.

This report concludes our development on this site and forms the handover to the client’s development team. It details the current functionality, the testing performed and any exclusions from the functional requirements.

The latest version of the site can be seen at [app.slehmann36.com](http://app.slehmann36.com/) and the latest code at the GitHub [repository](https://github.com/slehmann36/Group3-Real-Estate-Site) (private).