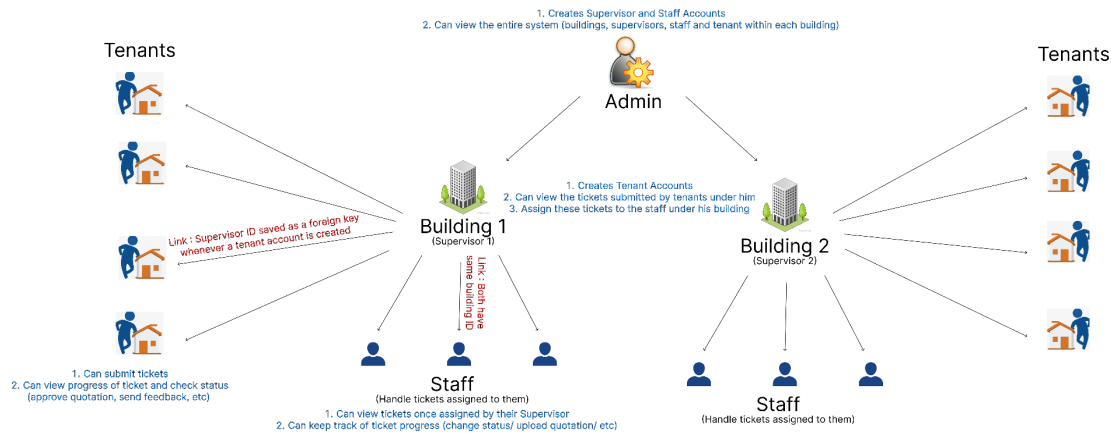


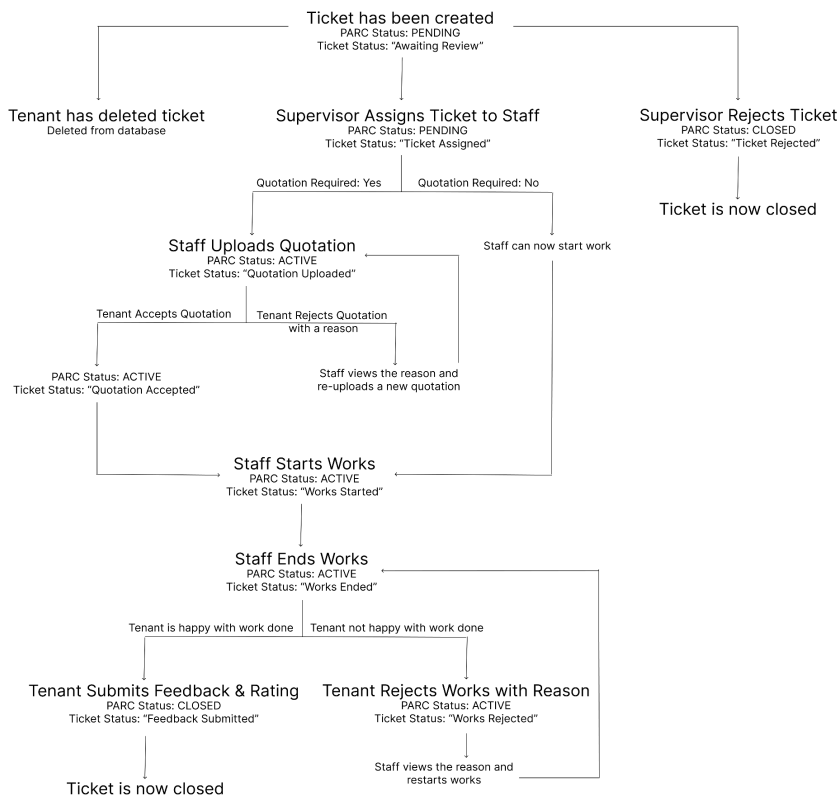
Changes in requirements, design and implementation (compared to 3rd meeting)

Changes in design:

Final user hierarchy of our system:



Final ticket status flow of our system:



Our use case diagram and sequence diagram have been updated to consider this flow.
Here is the link to the diagrams:

[Use Case Diagram](#)

[Sequence Diagram](#)

The rest of the diagrams are the same as they were in PM2 .

Here's a link for those too, for your reference:

- [Class Diagram](#)
- [Database ER Diagram](#)
- [Subsystem Diagram](#)

No change in requirement:

- Client meeting on 01/08/2023
- Client feedback: Happy with our current flow of the app, exactly what he wanted
 - Implement hashing for cookies to mask user login details
 - Can implement 2FA for better security

Changes in implementation:

- Authentication - added in cookies
(when a user logs in, cookies are created, further at every stage checks for the cookies present, if the value on the cookie corresponds to the userID and the userType, only then is the page rendered, else the user is unauthorized.)
This solves the security issue we had earlier of the user typing in the route and accessing the portal without logging in. They cannot access the portal now until they actually log in.
- Added in Reject Ticket (Supervisor) and Delete Ticket (Tenant) in our ticket flow
- Improved our frontend and UI layout

Changes required before the final presentation:

- Hash the userID stored in the cookie (do not want the userID to be visible to client)
- Implement 2FA (OTP each time a user tries to login)

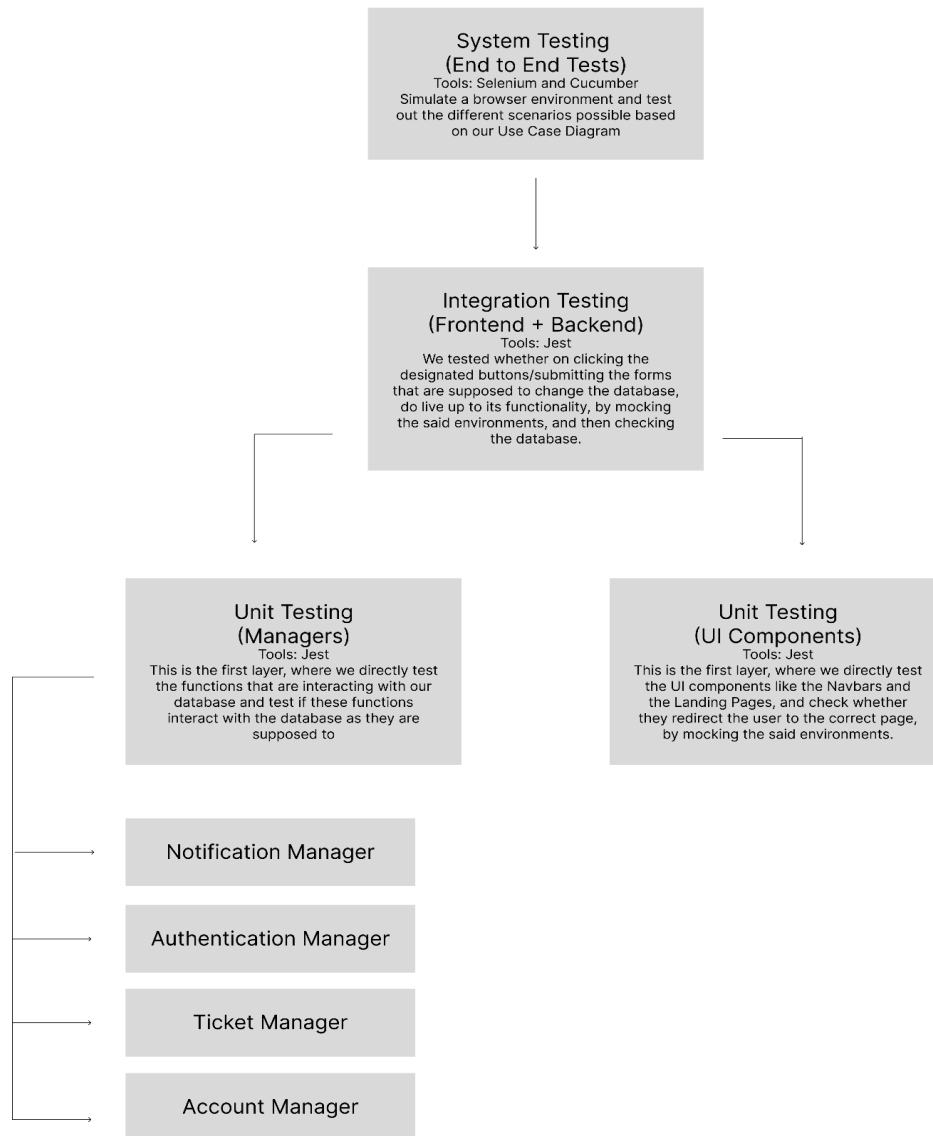
Implementation of features (live demo)

Github repo link:

<https://github.com/RB0810/Housing-Ticket-Managing-System>

Testing of the Software

We followed the bottom-up approach based on the diagram below to implement our tests:



You can find the test case documents and the output of the terminal (the screenshot of us running the test cases, and them passing) below for each of these sections starting from the bottom-up.

Apart from this we also have some fuzzing tests to check for the valid urls/paths of our application. You can find more details on the same later in the doc after the documentation for all the above mentioned test cases in the diagram.

Unit Testing - Managers (Jest)

Notification Manager

Test case ID	TC_NM_1
Test case name	Testing if mails are sent when triggered
Objective	Notifications are sent from the database to the said Email ID whenever it is triggered
Use Case	<ul style="list-style-type: none">- Notify Supervisor that a new ticket has been created- Notify Staff that work is rejected- Notify Staff that quotation is rejected- Notify Staff that quotation is accepted- Notify Staff that ticket is assigned- Notify Supervisor and Staff that Tenant has submitted a feedback survey- Notify Tenant that status of ticket has been updated- Notify Tenant that Quotation has been uploaded
Pre-conditions	The recipient and sender should have valid email accounts
Event sequence	<ol style="list-style-type: none">1. The notification manager is triggered with the email body and the recipient email address.2. Mail successfully sent to the recipient.
Test case descriptor	Should send email successfully

```
PASS src/managers/notificationmanager.test.js
  Sending Notifications
    ✓ should send email successfully (73 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.644 s
Ran all test suites matching /src\managers\notificationmanager.test.js/i.
```

Ticket Manager

Test case ID	TC_TM_1
Test case name	Getting ticket details successfully
Objective	Checks that the ticket details are read from the database successfully
Use Case	- View particular ticket
Pre-conditions	Ticket is already created
Event sequence	1. User logs in to the portal and clicks on the view ticket button 2. The details for the said ticket is returned to the user from the database
Test case descriptor	Getting tickets does not return false

Test case ID	TC_TM_2
Test case name	Checking if ticket status changes successfully
Objective	Checks that the Status field for a ticket can be changed whenever triggered
Use Case	<ul style="list-style-type: none">- Change ticket status to "Works Started"- Change ticket status to "Quotation Uploaded"- Change ticket status to "Works Ended"- Change ticket status to "Feedback Submitted"- Change ticket status to "Quotation Accepted"- Change ticket status to "Quotation Rejected"- Change ticket status to "Works Rejected"
Pre-conditions	Ticket is already created
Event sequence	3. Ticket Status change triggered 4. Ticket Status successfully changed in the supabase
Test case descriptor	Change ticket status returns true if changed

Test case ID	TC_TM_3
Test case name	Ticket Assigned correctly
Objective	Update ticket after Supervisor assigns a particular ticket to a particular staff

Use case	Detects that ticket is assigned to staff
Pre-conditions	Ticket has been created by the Tenant
Event sequence	<ol style="list-style-type: none"> 1. Supervisor logs in and goes to said ticket 2. Supervisor assigns the ticket to the staff. 3. The StaffID column in the database for the said ticket is populated correctly and linked to the staff the ticket has been assigned to.
Test case descriptor	Assign Ticket to Staff

Test case ID	TC_TM_4
Test case name	Checking if PARC Status changes correctly
Objective	Check if the PARC Status for tickets is updated successfully when the function is triggered.
Use Case	Change PARC Status to "PENDING" Change PARC Status to "ACTIVE" Change PARC Status to "CLOSED"
Pre-conditions	Ticket is already created
Event sequence	<ol style="list-style-type: none"> 1. Tenant PARC Status change triggered 2. Ticket PARC Status changes and is saved in the database
Test case descriptor	Change ticket PARC status returns true if changed

Test case ID	TC_TM_5/6/7
Test case name	Checking if any input by Staff/Tenant is correctly saved in the database
Objective	Check if the ticket has been closed successfully and PARC Status changes whenever an event triggers it.
Use Case	<ul style="list-style-type: none"> - Save Feedback survey results from tenant and updates database - Saves "No Quotation Required" and updates database
Pre-conditions	Tenant has submitted feedback survey
Event sequence	<ol style="list-style-type: none"> 1. Tenant submits feedback for a particular ticket

	2. The feedback comments and rating are stored in the database.
Test case descriptors	<ul style="list-style-type: none"> - Submits feedback rating for ticket - Submits feedback comments for ticket - Sends no quotation required to database

Test case ID	TC_TM_8/9/10
Test case name	Checking if get tickets from database work
Objective	Check if the tickets are being successfully read from the database for each Tenant/Supervisor/Staff based on the PARC Status tab they are in
Use Case	<ul style="list-style-type: none"> - View all pending tickets - View all active tickets - View all closed tickets
Pre-conditions	Tenant has logged in his portal and there exists at least one ticket under said tenant with PARC Status pending/active/closed in the database.
Event sequence	<ol style="list-style-type: none"> 1. Tenant logs in to the portal and navigates to the Active Tickets page 2. Tenant can see the tickets under him as pulled from the database
Test case descriptor	<ul style="list-style-type: none"> - View all pending tickets for a particular tenant - View all active tickets for a particular tenant - View all closed tickets for a particular tenant

Test case ID	TC_TM_11
Test case name	Retrieve non existent ticket
Objective	Makes sure that user cannot retrieve non-existent ticket
Test case descriptors	Getting a non-existent ticket should return false

Test case ID	TC_TM_12
Test case name	Update non existent ticket
Objective	Makes sure that user cannot update non-existent ticket
Test case	Updating a non-existent ticket should return false

descriptors	
-------------	--

Test case ID	TC_TM_13
Test case name	Deleting a non-existent ticket
Objective	Makes sure that user cannot delete non-existent ticket
Test case descriptors	Deleting a non-existent ticket should return false

Test case ID	TC_TM_14
Test case name	Creating a ticket with invalid tenant ID
Objective	Makes sure that only tenants who have valid user ids in the database can create ticket
Test case descriptors	Creating a new ticket with invalid Tenant ID should return false

Test case ID	TC_TM_15
Test case name	Assigning ticket to non-existent staff
Objective	Makes sure that user cannot assign ticket to non-existent staff
Test case descriptors	Assigning ticket to non-existent staff should return false

```

PASS src/managers/ticketmanager.test.js (5.052 s)
TicketManager all test cases
  ✓ Getting Ticket does not return False (172 ms)
  ✓ Change ticket status returns true if changed (177 ms)
  ✓ Change ticket PARC status returns true if changed (153 ms)
  ✓ Assign Ticket to Staff (158 ms)
  ✓ Submits feedback rating for ticket (173 ms)
  ✓ Submits feedback comments for ticket (141 ms)
  ✓ Sends no quotation required to database (252 ms)
  ✓ View all active tickets for a particular tenant (168 ms)
  ✓ View all pending tickets for a particular tenant (203 ms)
  ✓ View all closed tickets for a particular tenant (218 ms)
  ✓ Negative Test Case : Creating a new ticket with invalid TenantID should return false (170 ms)
  ✓ Negative Test Case : Getting a non-existent ticket should return false (127 ms)
  ✓ Negative Test Case : Updating a non-existent ticket should return false (165 ms)
  ✓ Negative Test Case : Deleting a non-existent ticket should return false (169 ms)
  ✓ Negative Test Case : Assigning ticket to non-existent staff should return false (159 ms)

Test Suites: 1 passed, 1 total

```


Account Manager

Test case ID	TC_AM_1/2/3/4/5/6
Test case name	Fetch details from database
Objective	Check that the information required for user profiled/buildings can be correctly fetched from the database
Use case	<ul style="list-style-type: none">- Landlord Profile Information → Fetch details- Tenant Profile Information → Fetch details- Manage all accounts and buildings → Fetch details
Pre-conditions	User account must be created
Event sequence	<ol style="list-style-type: none">1. User is logged in to their account2. User clicks on the profile in the navbar3. User can view their profile details
Test case descriptors	<ul style="list-style-type: none">- Get supervisor details- Get staff details- Get tenant details- Get units- Get all building details- Get all staff for a supervisor

Test case ID	TC_AM_7
Test case name	Set new password.
Objective	Check that users can change their passwords.
Use case	Set New Password → Save Supervisor/Staff/Tenant Account Information
Pre-conditions	User Account must be created
Event sequence	<ol style="list-style-type: none">1. User logs in to their account.2. User goes to profile.3. User sets a new password.
Test case descriptor	setPassword should set the password successfully

Test case ID	TC_AM_8
Test case name	Supervisor Account Created
Objective	Check that the Supervisor Account along with the Building has been created correctly and linked in the supabase
Use case	Create supervisor account → Save Supervisor Account Information
Pre-conditions	Admin should be logged in to their account
Event sequence	<ol style="list-style-type: none"> 1. Admin goes to the create supervisor account page and enters all credentials required. 2. Admin clicks on the Submit button. 3. Supervisor Account is created.
Test case descriptor	Create supervisor account

Test case ID	TC_AM_9
Test case name	Staff Account Created
Objective	Check that the Staff Account has been created correctly and linked to the corresponding building in the supabase
Use case	Create staff account → Save Staff Account Information
Pre-conditions	Admin should be logged in to their account
Event sequence	<ol style="list-style-type: none"> 1. Admin goes to the create staff account page and enters all credentials required. 2. Admin clicks on the Submit button. 3. Staff Account is created.
Test case descriptor	Create staff account

Test case ID	TC_AM_10
Test case name	Tenant Account Created
Objective	Check that the Tenant Account has been created correctly and linked to the supervisor in the supabase

Pre-conditions	Supervisor should be logged in to their account
Event sequence	<ol style="list-style-type: none"> 1. Supervisor goes to the create tenant account page and enters all credentials required. 2. Supervisor clicks on the Submit button. 3. Tenant Account is created.
Test case descriptor	Create tenant account with lease and units

Test case ID	TC_AM_11
Test case name	Creating an Account with missing fields
Objective	Makes sure Supervisor fills in all the details before creating Staff/Tenant account
Test case descriptors	Negative Test Case: Creating an Account with missing fields - Error

Test case ID	TC_AM_12
Test case name	Creating an Account with an undefined foreign key link
Objective	Makes sure that user cannot create a staff/tenant account for a non-existent building
Test case descriptors	Negative Test Case: Creating an Account with an undefined foreign key link - Account Not Created

Test case ID	TC_AM_13
Test case name	Creating an account with invalid data type
Objective	Makes sure that Supervisor cannot enter data of invalid type when creating account for Staff/Tenant
Test case descriptors	Negative Test Case: Creating an account with invalid data type - Error

Test case ID	TC_AM_14
Test case name	Getting details for an account that does not exist
Objective	Makes sure that supervisor/staff/admin does not get details for account that does not exist
Test case descriptors	Negative Test Case: Getting details for an account that does not exist - Error

Test case ID	TC_AM_15
Test case name	Setting password for an account that does not exist
Objective	Makes sure that user cannot set password for non-existent account
Test case descriptors	Negative Test Case: Setting password for an account that does not exist - Error

```

PASS src/managers/accountmanager.test.js (8.948 s)
  AccountManager all test cases
    ✓ Create supervisor account (487 ms)
    ✓ Create tenant account with lease and units (845 ms)
    ✓ Create staff account (204 ms)
    ✓ Get supervisor details (164 ms)
    ✓ Get staff details (240 ms)
    ✓ Get tenant details (425 ms)
    ✓ Get units (224 ms)
    ✓ Get all building details (185 ms)
    ✓ Get all staff for a supervisor (182 ms)
    ✓ setPassword should set the password successfully (104 ms)
    ✓ Negative Test Case: Creating an Account with missing fields - Error (117 ms)
    ✓ Negative Test Case: Creating an Account with an undefined foreign key link - Account Not Created (238 ms)
    ✓ Negative Test Case: Creating an account with invalid data type - Error (70 ms)
    ✓ Negative Test Case: Getting details for an account that does not exist - Error (87 ms)
    ✓ Negative Test Case: Setting password for an account that does not exist - Error (93 ms)

Test Suites: 1 passed, 1 total
Tests: 15 passed, 15 total
Snapshots: 0 total
Time: 9.154 s, estimated 11 s
Ran all test suites matching /src\managers\accountmanager.test.js/i.

```

Authentication Manager

Test case ID	TC_Auth_1/2
Test case name	Verify login credentials
Objective	Check that users can login to their accounts only if the credentials they enter match the one on the database.

Use case	<ul style="list-style-type: none"> - Enter tenant login details → Verify login details - Enter supervisor login details → Verify login details - Enter staff login details → Verify login details - Enter admin login details → Verify login details
Pre-conditions	User Account must be created
Event sequence	<ol style="list-style-type: none"> 1. User enters their login credentials 2. User clicks on login 3. User can view their account if credentials are correct, else invalid credentials are prompted.
Test case descriptors	<ul style="list-style-type: none"> - loginAuth should redirect to correct URL on successful login - loginAuth should throw an error on invalid credentials

```

PASS src/managers/authentication.test.js
Authentication all test cases
  ✓ loginAuth should redirect to correct URL on successful login (604 ms)
  ✓ loginAuth should throw an error on invalid credentials (618 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        4.122 s
Ran all test suites matching /src\managers\authentication.test.js/i.

```

UI Unit Testing - Routing + Button Clicks

Admin Landing Page

Test case ID	TC_ALP_1
Test case name	Go to Manage account information page
Objective	To redirect Admin to the landing page with the account information
Use case	Manage all buildings and account information → Fetch details
Pre-conditions	Admin should be logged in
Event sequence	<ol style="list-style-type: none"> 1. Admin logs in 2. Click on 'Manage Accounts' 3. Redirected to Manage Accounts page
Test case descriptors	Go to manage account

Test case ID	TC_ALP_2
Test case name	Go to create supervisor account
Objective	To redirect admin to the page to create supervisor's account
Use case	Create Supervisor Account → Save Supervisor Account Information
Pre-conditions	Admin should be logged in
Event sequence	<ol style="list-style-type: none"> 1. Admin Logs into portal 2. Admin clicks on 'Building Supervisor' 3. Admin clicks on 'Create Supervisor Account' 4. Admin is redirected to the Create Supervisor Account page
Test case descriptors	Go to create supervisor

Test case ID	TC_ALP_3
Test case name	Go to create staff account page
Objective	To redirect the Admin to the page where he can create staff accounts
Use case	Create Staff Account → Save Staff Account Information
Pre-conditions	Admin to be logged in
Event sequence	<ol style="list-style-type: none"> 1. Admin logs in to portal 2. Admin clicks on Staff 3. Admin clicks on the heading 'Create Staff Account' 4. Admin is redirected to the heading
Test case descriptors	Go to Create Staff

```

PASS src/pages/landingpages/__test__/adminlandingpage.test.js
Admin landingpage should send Admin to the right page
  ✓ Go to manageaccount (173 ms)
  ✓ Go to Create Supervisor (26 ms)
  ✓ Go to Create Staff (23 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        2.123 s
Ran all test suites matching /adminlandingpage\.test\.js/i.

```

Staff Landing Page

Test case ID	TC_SLP_1
Test case name	Go to View all pending tickets
Objective	To view all pending tickets in the Staff Portal
Use case	Enter Login Details → Login to your account → View All Pending Tickets
Pre-conditions	Should have logged in to the Staff account
Event sequence	<ol style="list-style-type: none"> 1. Staff logs into portal 2. Staff clicks on Pending Tickets 3. Staff is redirected to the list of pending tickets
Test case descriptors	Click on pending tickets

Test case ID	TC_SLP_2
Test case name	Go to View all active tickets
Objective	To view all active tickets in the Staff Portal
Use case	Enter Login Details → Login to your account → View All Active Tickets
Pre-conditions	Staff is logged in
Event sequence	<ol style="list-style-type: none"> 1. Staff logs into portal 2. Staff clicks on Active Tickets 3. Staff is redirected to the list of active tickets

Test case descriptors	Click on activetickets
-----------------------	------------------------

Test case ID	TC_SLP_3
Test case name	Go to View Closed tickets
Objective	To view all closed tickets in the Staff Portal
Use case	Enter Login Details → Login to your account → View All Closed Tickets
Pre-conditions	Staff is logged in
Event sequence	<ol style="list-style-type: none"> 1. Staff logs into portal 2. Staff clicks on Closed Tickets 3. Staff is redirected to the list of closed tickets
Test case descriptors	Click on closed tickets

```

PASS src/pages/landingpages/__test__/stafflandingpage.test.js
Testing routing functions
  ✓ Click on pending tickets (220 ms)
  ✓ Click on active tickets (40 ms)
  ✓ Click on closed tickets (34 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        2.186 s
Ran all test suites matching /staff/i.

```

Supervisor Landing Page

Test case ID	TC_LL_P_1
Test case name	Go to View pending tickets
Objective	Go to the page to View all the pending tickets under supervisor in the Supervisor portal
Use case	Enter Login Details → Login to your account → View all pending tickets

Pre-conditions	Supervisor is logged in
Event sequence	<ol style="list-style-type: none"> 1. Supervisor logs into portal 2. Supervisor clicks on Pending Tickets 3. Supervisor is redirected to the list of pending tickets
Test case descriptors	Test if can route to pending tickets

Test case ID	TC_LL2P_2
Test case name	Go to View Active tickets
Objective	Go to the page to View all the active tickets under supervisor in the Supervisor portal
Use case	Enter Login Details → Login to your account → View all active tickets
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. Supervisor logs into portal 2. Supervisor clicks on Active Tickets 3. Supervisor is redirected to the list of active tickets
Test case descriptors	Test if can route to active tickets

Test case ID	TC_LL2P_3
Test case name	Go to View Closed tickets
Objective	Go to the page to View all the closed tickets under supervisor in the Supervisor portal
Use case	Enter Login Details → Login to your account → View all closed tickets
Pre-conditions	NA

Event sequence	<ol style="list-style-type: none"> 1. Supervisor logs into portal 2. Supervisor clicks on Closed Tickets 3. Supervisor is redirected to the list of Closed tickets
Test case descriptors	Test if can route to closed tickets

Test case ID	TC_LL_P_4
Test case name	Go to Create tenant account
Objective	Go to the page to Create tenant account using the supervisor portal
Use case	Create Tenant Account
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. Supervisor logs into portal 2. Supervisor clicks on Create Tenant Account 3. Supervisor is redirected to the page to create tenant account
Test case descriptors	Test if can route to create tenants

```
PASS src/pages/landingpages/_test_/supervisorlandingpage.test.js
Testing if routing works for supervisor portal
✓ Test if can route to pending tickets (267 ms)
✓ Test if can route to active tickets (52 ms)
✓ Test if can route to closed tickets (54 ms)
✓ Test if can route to create tenants (51 ms)
```

Tenant Landing Page

Test case ID	TC_TLP_1
Test case name	Go to View pending tickets
Objective	Go to the page For tenant to view a list of pending tickets in the tenant portal

Use case	Enter tenant login credentials → Login to tenant account → View all pending tickets
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. Tenant logs into portal 2. Tenant clicks on Pending Tickets 3. Tenant is redirected to the list of pending tickets
Test case descriptors	Test if can route to pending tickets

Test case ID	TC_TLP_2
Test case name	Go to View active tickets
Objective	Go to the page For tenant to view a list of active tickets in the tenant portal
Use case	Enter tenant login credentials → Login to tenant account → View all active tickets
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. Tenant logs into portal 2. Tenant clicks on Active Tickets 3. Tenant is redirected to the list of active tickets
Test case descriptors	Test if can route to active tickets

Test case ID	TC_TLP_3
Test case name	Got to view closed tickets
Objective	For tenant to view a list of closed tickets in the tenant portal

Use case	Enter tenant login credentials → Login to tenant account → View all closed tickets
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. Tenant logs into portal 2. Tenant clicks on Closed Tickets 3. Tenant is redirected to the list of closed tickets
Test case descriptors	Test if can route to closed tickets

Test case ID	TC_TLP_4
Test case name	Go to Create ticket
Objective	For tenant to raise a service ticket through the Tenant Portal
Use case	Enter tenant login credentials → Login to tenant account → Create new service ticket
Pre-conditions	Should have logged into the tenant portal account
Event sequence	<ol style="list-style-type: none"> 1. Tenant enters credentials in the tenant login page 2. Tenant logs into the account 3. Tenant should be inside the tenant portal 4. Tenant clicks on create ticket 5. Tenant should be at the create ticket page
Test case descriptors	Test if can route back to the create ticket page

```
PASS src/pages/landingpages/__test__/tenantlandingpage.test.js
Testing if routing works for supervisor portal
✓ Test if can route to pending tickets (291 ms)
✓ Test if can route to active tickets (74 ms)
✓ Test if can route to closed tickets (58 ms)
✓ Test if can route to create ticket (71 ms)
```

Main Landing Page

Test case ID	TC_LP_1
Test case name	Go to the tenant login page
Objective	For user to go to the tenant login page from the portal
Use case	User accesses site's landing page → Clicks on login under Tenant Account
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. User goes to site's landing page 2. User clicks on login under Tenant Account
Test case descriptors	Test if can route to Tenant login page

Test case ID	TC_LP_2
Test case name	Go to the admin login page
Objective	For user to go to the admin login page from the portal
Use case	User accesses site's landing page → Clicks on login under Admin Account
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. User goes to site's landing page 2. User clicks on login under Admin Account
Test case descriptors	Test if can route to Admin login page

Test case ID	TC_LP_3
Test case name	Go to the Landlord Login page
Objective	For user to go to the LandLord Login page from the portal

Use case	User accesses site's landing page → Clicks on login under Landlord Account
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. User goes to site's landing page 2. User clicks on login under Landlord Account
Test case descriptors	Test if can route to Landlord login page

```

PASS src/pages/landingpages/__test__/landingpage.test.js
Testing landing page functionality
  ✓ Click on Admin (114 ms)
  ✓ Click on Landlord (13 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        2.03 s

```

Admin Navbar

Test case ID	TC_AN_1
Test case name	Go to the create supervisor page
Objective	For Admin to go to Create Supervisor Page
Use case	Admin clicks on create supervisor account in navbar → Admin goes to create supervisor account
Pre-conditions	Admin is logged into Admin Portal
Event sequence	<ol style="list-style-type: none"> 1. Admin Clicks on 'Create Supervisor Account' in Navbar 2. Admin is redirected to Create supervisor Account
Test case descriptors	Test if can route to Create Supervisor Page

Test case ID	TC_AN_2
--------------	---------

Test case name	Go to the Create Staff page
Objective	For Admin to go to Create Staff Page
Use case	Admin clicks on create supervisor account in navbar → Admin goes to create staff account
Pre-conditions	Admin should be logged into Admin Portal
Event sequence	<ol style="list-style-type: none"> 1. Admin Clicks on 'Create Supervisor Account' 2. Admin is redirected to the page, and scrolls down to the staff section
Test case descriptors	Test if can route to Create Staff Page

Test case ID	TC_AN_3
Test case name	Go to the Manage Accounts Page
Objective	For Admin to go to the Manage Accounts Page
Use case	Admin clicks on Manage Accounts in navbar → Admin goes to manage accounts page
Pre-conditions	Admin should be logged into Admin Portal
Event sequence	<ol style="list-style-type: none"> 1. Admin Clicks on 'Manage Accounts' in navigation bar 2. Admin is redirected to the page
Test case descriptors	Test if can route to Manage accounts

Test case ID	TC_AN_4
---------------------	----------------

Test case name	Logout of Admin Portal
Objective	For Admin to log out of Admin Portal
Use case	Admin clicks on Logout button in navbar → Admin goes to Admin landing page
Pre-conditions	Admin should be logged into portal
Event sequence	<ol style="list-style-type: none"> 1. Admin clicks on 'logout' in navigation bar 2. Admin is redirected to landing page
Test case descriptors	Test if can logout

Test case ID	TC_AN_5
Test case name	Go to Admin Portal Landing Page
Objective	For Admin to go to landing page to login
Use case	Admin clicks on Admin Portal button in navbar → Admin goes to Admin Portal landing page
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. Admin clicks on 'Admin Portal' in navigation bar 2. Admin is redirected to landing page
Test case descriptors	Test if can route to Admin Portal


```

PASS src/components/__tests__/adminnavbar.test.js
Testing if routing works for the admin navbar
  ✓ Test if can route to Admin Portal (380 ms)
  ✓ Test if can route to Create Supervisor Page (360 ms)
  ✓ Test if can route to Create Staff Page (322 ms)
  ✓ Test if can route to Manage accounts (67 ms)
  ✓ Test if can logout (155 ms)

```

Supervisor Navbar

Test case ID	TC_LN_1
Test case name	`Go to profile information page
Objective	For Supervisor to go to Profile page
Use case	Supervisor clicks on Profile in navbar → Supervisor loads the profile
Pre-conditions	Supervisor is logged into Admin Portal
Event sequence	<ol style="list-style-type: none"> Supervisor clicks on Profile in navbar Supervisor loads the profile
Test case descriptors	Test if can route to profile

Test case ID	TC_LN_2
Test case name	Go to view active tickets page
Objective	For supervisor to be redirected to the page filled with all active tickets
Use case	Supervisor clicks on View Tickets in navbar → Supervisor clicks on 'Active' link
Pre-conditions	Supervisor should be logged into Supervisor Portal

Event sequence	<ol style="list-style-type: none"> 1. Supervisor clicks on View Tickets in navigation bar 2. Supervisor clicks on Active link 3. Supervisor sees all active tickets
Test case descriptors	Test if can route to View Tickets active

Test case ID	TC_LN_3
Test case name	Go to view closed tickets page
Objective	For supervisor to be redirected to the page filled with all closed tickets
Use case	Supervisor clicks on View Tickets in navbar → Supervisor clicks on 'Closed' link
Pre-conditions	Supervisor should be logged into Supervisor Portal
Event sequence	<ol style="list-style-type: none"> 1. Supervisor clicks on View Tickets in navigation bar 2. Supervisor clicks on Closed link 3. Supervisor sees all closed tickets
Test case descriptors	Test if can route to View Tickets closed

Test case ID	TC_LN_4
Test case name	Go to view pending tickets page
Objective	For supervisor to be redirected to the page filled with all pending tickets
Use case	Supervisor clicks on View Tickets in navbar → Supervisor clicks on 'Pending' link

Pre-conditions	Supervisor should be logged into portal
Event sequence	<ol style="list-style-type: none"> 1. Supervisor clicks on View Tickets in navigation bar 2. Supervisor clicks on Pending link 3. Supervisor sees all pending tickets
Test case descriptors	Test if can route to View Tickets pending

Test case ID	TC_LN_5
Test case name	Go to Supervisor Portal Landing Page
Objective	For Supervisor to go to landing page to login
Use case	Supervisor clicks on Supervisor Portal button in navbar → Supervisor goes to Supervisor Portal landing page
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. Supervisor clicks on 'Supervisor Portal' in navigation bar 2. Supervisor is redirected to landing page
Test case descriptors	Test if can route to Supervisor Portal

```
PASS src/components/__tests__/supervisornavbar.test.js
```

```
Testing if routing works for the staff navbar
```

- ✓ Test if can route to Supervisor Portal (631 ms)
- ✓ Test if can route to View Tickets pending (272 ms)
- ✓ Test if can route to View Tickets closed (232 ms)
- ✓ Test if can route to View Tickets active (243 ms)
- ✓ Test if can route to profile (49 ms)

Staff Navbar

Test case ID	TC_SN_1
Test case name	Go to view active tickets page
Objective	For Staff to be redirected to the page filled with all active tickets
Use case	Staff clicks on View Tickets in navbar → Staff clicks on 'Active' link
Pre-conditions	Staff should be logged into Staff Portal
Event sequence	<ol style="list-style-type: none">1. Staff clicks on View Tickets in navigation bar2. Staff clicks on Active link3. Staff sees all active tickets
Test case descriptors	Test if can route to View Tickets active

Test case ID	TC_SN_2
Test case name	Go to view closed tickets page
Objective	For Staff to be redirected to the page filled with all closed tickets
Use case	Staff clicks on View Tickets in navbar → Staff clicks on 'Closed' link
Pre-conditions	Staff should be logged into Staff Portal
Event sequence	<ol style="list-style-type: none">1. Staff clicks on View Tickets in navigation bar2. Staff clicks on Closed link3. Staff sees all closed tickets
Test case descriptors	Test if can route to View Tickets closed

Test case ID	TC_SN_3
Test case name	Go to view pending tickets page
Objective	For Staff to be redirected to the page filled with all pending tickets
Use case	Staff clicks on View Tickets in navbar → Staff clicks on 'Pending' link
Pre-conditions	Staff should be logged into portal
Event sequence	<ol style="list-style-type: none"> 1. Staff clicks on View Tickets in navigation bar 2. Staff clicks on Pending link 3. Staff sees all pending tickets
Test case descriptors	Test if can route to View Tickets pending

Test case ID	TC_SN_4
Test case name	Go to Staff Portal Landing Page
Objective	For Staff to go to landing page to login
Use case	Staff clicks on Staff Portal button in navbar → Staff goes to Staff Portal landing page
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. Staff clicks on 'Staff Portal' in navigation bar 2. Staff is redirected to landing page
Test case descriptors	Test if can route to Staff Portal

Testing if routing works for the staff navbar

- ✓ Test if can route to Staff Portal (507 ms)
- ✓ Test if can route to View Tickets pending (150 ms)
- ✓ Test if can route to View Tickets pending (124 ms)
- ✓ Test if can route to View Tickets pending (138 ms)

Tenant Navbar

Test case ID	TC_TN_1
Test case name	`Go to profile information page
Objective	For Tenant to go to Profile page
Use case	Tenant clicks on Profile in navbar → Tenant loads the profile
Pre-conditions	Tenant is logged into Tenant Portal
Event sequence	<ol style="list-style-type: none">1. Tenant clicks on Profile in navbar2. Tenant loads the profile
Test case descriptors	Test if can route to profile

Test case ID	TC_TN_2
Test case name	Go to view active tickets page
Objective	For Tenant to be redirected to the page filled with all active tickets
Use case	Tenant clicks on View Tickets in navbar → Tenant clicks on 'Active' link
Pre-conditions	Tenant should be logged into Tenant Portal
Event sequence	<ol style="list-style-type: none">1. Tenant clicks on View Tickets in navigation bar2. Tenant clicks on Active link3. Tenant sees all active tickets

Test case descriptors	Test if can route to View Tickets active
-----------------------	--

Test case ID	TC_TN_3
Test case name	Go to view closed tickets page
Objective	For Tenant to be redirected to the page filled with all closed tickets
Use case	Tenant clicks on View Tickets in navbar → Tenant clicks on 'Closed' link
Pre-conditions	Tenant should be logged into Tenant Portal
Event sequence	<ol style="list-style-type: none"> 1. Tenant clicks on View Tickets in navigation bar 2. Tenant clicks on Closed link 3. Tenant sees all closed tickets
Test case descriptors	Test if can route to View Tickets closed

Test case ID	TC_TN_4
Test case name	Go to view pending tickets page
Objective	For Tenant to be redirected to the page filled with all pending tickets
Use case	Tenant clicks on View Tickets in navbar → Tenant clicks on 'Pending' link
Pre-conditions	Tenant should be logged into portal
Event sequence	<ol style="list-style-type: none"> 1. Tenant clicks on View Tickets in navigation bar 2. Tenant clicks on Pending link 3. Tenant sees all pending tickets

Test case descriptors	Test if can route to View Tickets pending
-----------------------	---

Test case ID	TC_TN_5
Test case name	Go to Tenant Portal Landing Page
Objective	For Tenant to go to landing page to login
Use case	Tenant clicks on Tenant Portal button in navbar → Tenant goes to Tenant Portal landing page
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. Tenant clicks on 'Tenant Portal' in navigation bar 2. Tenant is redirected to landing page
Test case descriptors	Test if can route to Tenant Portal

PASS src/components/__tests__/tenantnavbar.test.js

Testing if routing works for the staff navbar

- ✓ Test if can route to Tenant Portal (596 ms)
- ✓ Test if can route to View Tickets pending (240 ms)
- ✓ Test if can route to View Tickets closed (232 ms)
- ✓ Test if can route to View Tickets active (227 ms)
- ✓ Test if can route to profile (55 ms)

Nav Bar

Test case ID	TC_NB_1
Test case name	Go to landing page
Objective	For user to go to landing page to login
Use case	User clicks on Housing Portal button in navbar → User goes to main landing page
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none">1. User clicks on 'Tenant Portal' in navigation bar2. User is redirected to landing page
Test case descriptors	Test if can route to Landing Page

Test case ID	TC_NB_2
Test case name	Go to Landlord Login page
Objective	For user to go to Landlord Login to login
Use case	User clicks on Landlord Login button in navbar → User goes to Landlord Login page
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none">1. User clicks on 'Tenant Login' navigation bar2. User is redirected to Tenant Login page

Test case descriptors	Test if can route to Landlord Login page
-----------------------	--

Test case ID	TC_NB_3
Test case name	Go to Admin Login page
Objective	For user to go to Admin Login to login
Use case	User clicks on Admin Login button in navbar → User goes to Admin Login page
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. User clicks on 'Admin Login' navigation bar 2. User is redirected to Admin Login page
Test case descriptors	Test if can route to Admin Login page

Test case ID	TC_NB_4
Test case name	Go to Tenant Login page
Objective	For user to go to Tenant Login to login
Use case	User clicks on Tenant Login button in navbar → User goes to Tenant Login page
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none"> 1. User clicks on 'Tenant Login' navigation bar 2. User is redirected to Tenant Login page

Test case descriptors	Test if can route to Tenant Login page
-----------------------	--

```
PASS src/components/__tests__/navbar.test.js
We test if the navigation bar directs users to the selected pages
  ✓ Click on housing portal brings you to the landing page (264 ms)
  ✓ Click Landlord Login brings you to Landlord Login Page (162 ms)
  ✓ Click Admin Login brings you to Admin Login Page (47 ms)
  ✓ Click Tenant Login brings you to Tenant Login Page (48 ms)
```

Integration Tests

Test case ID	TC_Integration_1
Test case name	Staff Account Created
Objective	Check that the Staff Account has been created in the database when the form is submitted
Use case	Create staff account → Save Staff Account Information
Pre-conditions	Admin should be logged in to their account
Event sequence	<ol style="list-style-type: none"> 1. Admin goes to the create staff account page and enters all credentials required. 2. Admin clicks on the Submit button. 3. Staff Account is created.
Test case descriptor	Should create a new staff account

```
PASS src/pages/admin/createstaffacc.test.js (7.891 s)
CreateStaffAcc Component
  ✓ should create a new staff account (1082 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 8.094 s, estimated 10 s
Ran all test suites matching /src\\pages\\admin\\createstaffacc.test.js/i.
```

Test case ID	TC_Integration_2
Test case name	Supervisor Account Created
Objective	Check that the Supervisor Account along with the Building has been created in the database when the form is submitted
Use case	Create supervisor account → Save Supervisor Account Information
Pre-conditions	Admin should be logged in to their account
Event sequence	<ol style="list-style-type: none"> 1. Admin goes to the create supervisor account page and enters all credentials required. 2. Admin clicks on the Submit button. 3. Supervisor Account is created.
Test case descriptor	Should create a new supervisor account

```

PASS src/pages/admin/createsupervisoracc.test.js (8.521 s)
  CreateSupervisorAcc Component
    ✓ should create a new supervisor account (2157 ms)

Test Suites: 1 passed, 1 total
Tests:      1 passed, 1 total
Snapshots:  0 total
Time:       8.632 s
Ran all test suites matching /src\pages\admin\createsupervisoracc.test.js/i.

```

Test case ID	TC_Integration_3
Test case name	Tenant Account Created
Objective	Check that the Tenant Account has been created correctly in the database when the form is submitted.
Pre-conditions	Supervisor should be logged in to their account
Event sequence	<ol style="list-style-type: none"> 1. Supervisor goes to the create tenant account page and enters all credentials required. 2. Supervisor clicks on the Submit button. 3. Tenant Account is created.
Test case descriptor	should create tenant account

```

PASS src/pages/supervisor/createtenantacc.test.js (7.796 s)
  Create Tenant Component
    ✓ should create tenant account (1662 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        7.971 s, estimated 10 s
Ran all test suites matching /src\\pages\\supervisor\\createtenantacc.test.js/i.

```

Test case ID	TC_Integration_4
Test case name	New Service Ticket Created
Objective	Check that the Service Ticket has been created correctly in the database when the form is submitted.
Pre-conditions	Tenant should be logged in to their account
Event sequence	<ol style="list-style-type: none"> 1. Tenant goes to the create ticket page and enters all details required. 2. Tenant clicks on the Submit button. 3. Service Ticket is created.
Test case descriptor	Should create a new ticket

```

PASS src/pages/tenant/createticket.test.js (7.655 s)
  Create Ticket Component
    ✓ should create a new ticket (1445 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        7.853 s, estimated 10 s
Ran all test suites matching /src\\pages\\tenant\\createticket.test.js/i.

```

System Testing (Cucumber & Selenium)

Tenant Portal

Test case ID	TC 1
Test case name	Login to Tenant Account
Objective	Login function for Tenant
Pre-conditions	Tenant has loaded the login page for Tenant Portal
Event sequence	<ol style="list-style-type: none"> 1. Tenant enters account credentials 2. Tenant Username is shown but Password isn't shown

	3. Tenant clicks “login” button 4. Tenant is routed into the Tenant Portal
Input	Tenant Credentials
Output	Redirected to Tenant homepage or error notification for invalid credentials
Post-conditions	-

Test case ID	TC 2
Test case name	View all pending tickets
Objective	Tenant is able to view all pending tickets
Pre-conditions	The Tenant must have logged into their tenant account successfully.
Event sequence	1. Tenant navigates to the landing page 2. Tenant clicks on “View Pending Tickets” 3. Tenant is able to see all pending tickets in a table format
Input	Click on “View Pending Portal”
Output	Pending tickets are displayed to Tenant
Post-conditions	-

Test case ID	TC 3
Test case name	View all active tickets
Objective	Tenant is able to view all active tickets
Pre-conditions	The Tenant must have logged into their tenant account successfully.
Event sequence	1. Tenant navigates to the landing page 2. Tenant clicks on “View Active Tickets” 3. Tenant is able to see all active tickets in a table format
Input	Click on “View Active Tickets”
Output	Active tickets are displayed to Tenant
Post-conditions	-

Test case ID	TC 4
---------------------	-------------

Test case name	View all closed tickets
Objective	Tenant is able to view all closed tickets
Pre-conditions	The Tenant must have logged into their tenant account successfully.
Event sequence	<ol style="list-style-type: none"> 1. Tenant navigates to the landing page 2. Tenant clicks on “View Closed Tickets” 3. Tenant is able to see all closed tickets in a table format
Input	Click on “View Closed Tickets”
Output	Closed tickets are displayed to Tenant
Post-conditions	-

Test case ID	TC 5
Test case name	Create new service ticket
Objective	Tenant is able to create a new service ticket
Pre-conditions	Tenant has to be logged in
Event sequence	<ol style="list-style-type: none"> 1. Tenant navigates to the landing page 2. Tenant clicks on “Create Ticket” 3. Tenant is redirected to the “Create Ticket” page 4. Tenant fills up information for creating a ticket 5. Tenant clicks on “Create Service Ticket” 6. If creation of ticket is successful, Tenant receives a success notification and the page refreshes. Else, the tenant receives error notification.
Input	Tenant clicks on the “Create Ticket” page, Tenant enters “TESTTICKET”, Toilet, “TESTDESC”, and selects property
Output	Success Notification
Post-conditions	-

Test case ID	TC 6
Test case name	View Tenant Account Information
Objective	Tenant is able to view his account information
Pre-conditions	Tenant has to be logged in

Event sequence	<ol style="list-style-type: none"> 1. Tenant navigates to “Profile” section through the Navbar 2. Tenant’s details are rendered and viewable
Input	Tenant clicks on the “Profile” on Navbar
Output	Tenant profile information is rendered
Post-conditions	-

Test case ID	TC 7
Test case name	Set new password
Objective	Change password
Pre-conditions	Tenant is logged in and in the profile page
Event sequence	<ol style="list-style-type: none"> 1. Tenant goes to the Profile page via NavBar 2. Tenant enters new password 3. Tenant re-enters new password 4. New password is updated in supabase table entry
Input	Tenant clicks on set new password
Output	Success notification
Post-conditions	

Test case ID	TC 8
Test case name	View Quotation
Objective	Quotation displayed via reactpdfviewer
Pre-conditions	Ticket status is Quotation uploaded
Event sequence	<ol style="list-style-type: none"> 1. Tenant receives notification of Quotation uploaded 2. Tenant clicks on view ticket 3. Tenant is redirected to View Ticket page 4. Reactpdfviewer renders quotation
Input	Tenant clicks on View Ticket
Output	ReactPDFViewer renders Quotation
Post-conditions	

Test case ID	TC 9
Test case name	Download Quotation
Objective	Download Quotation into local computer
Pre-conditions	Ticket status is Quotation uploaded
Event sequence	<ol style="list-style-type: none"> 1. Tenant receives notification of Quotation uploaded 2. Tenant clicks on view ticket 3. Tenant is redirected to View Ticket page 4. Tenant clicks on the download button on react pdf viewer
Input	Tenant clicks on Download button on View Ticket page
Output	Quotation downloaded into local computer
Post-conditions	

Test case ID	TC 10_1
Test case name	Accept Quotation
Objective	Check if accept quotation button works
Pre-conditions	Ticket status is Quotation uploaded
Event sequence	<ol style="list-style-type: none"> 1. Tenant receives notification of Quotation uploaded 2. Tenant clicks on view ticket 3. Tenant is redirected to View Ticket page 4. Tenant clicks on accept quotation 5. Ticket status updated to Quotation Accepted 6. Tenant receives a email notification
Input	Tenant clicks Accept Quotation
Output	Ticket status changes and renders accordingly
Post-conditions	

Test case ID	TC 10_2
Test case name	Reject Quotation
Objective	Check if the reject quotation button works
Pre-conditions	Ticket status is Quotation uploaded

Event sequence	<ol style="list-style-type: none"> 1. Tenant receives notification of Quotation uploaded 2. Tenant clicks on view ticket 3. Tenant is redirected to View Ticket page 4. Tenant clicks on accept quotation 5. Ticket status updated to Quotation Accepted 6. Tenant receives a email notification
Input	Tenant clicks Reject Quotation
Output	Ticket status changes and renders accordingly
Post-conditions	

Test case ID	TC 11_1
Test case name	Submit feedback survey
Objective	Submit feedback for works done
Pre-conditions	Ticket status is Works Ended
Event sequence	<ol style="list-style-type: none"> 1. Tenant receives notification of works completion 2. Tenant fills in feedback 3. Tenant submits feedback 4. Feedback updated in supabase
Input	Tenant clicks on submit feedback button
Output	Ticket status changes and renders accordingly
Post-conditions	

Test case ID	TC 11_2
Test case name	Reject work
Objective	Reject work by Staff
Pre-conditions	Ticket status is Works Ended
Event sequence	<ol style="list-style-type: none"> 1. Tenant receives notification of works ended 2. Tenant goes to View Ticket 3. Tenant clicks on reject work 4. Staff receives notification of reject work
Input	Staff clicks on reject work button

Output	Ticket status updated
Post-conditions	

```

13 scenarios (13 passed)
37 steps (37 passed)
0m53.705s (executing steps: 0m52.693s)

Share your Cucumber Report with your team at https://reports.cucumber.io

Command line option:  --publish
Environment variable:  CUCUMBER_PUBLISH_ENABLED=true

More information at https://cucumber.io/docs/cucumber/environment-variables/

To disable this message, add this to your ./cucumber.js:
module.exports = { default: '--publish-quiet' }

```

Supervisor Portal

Test case ID	Supervisor_TC_01
Test case name	Log in to supervisor portal
Objective	To enter supervisor portal
Pre-conditions	Supervisor is on landlord landing page
Event sequence	<ol style="list-style-type: none"> 1. Supervisor enters credentials 2. Supervisor clicks log in 3. Supervisor portal page is rendered
Input	Supervisor enters login details
Output	Supervisor portal is rendered
Post-conditions	

Test case ID	Supervisor_TC_02
Test case name	View pending tickets
Objective	To view pending tickets

Pre-conditions	Supervisor is on supervisor portal page
Event sequence	1. Supervisor clicks on pending tickets button
Input	Supervisor clicks on pending tickets button
Output	Pending tickets page is rendered
Post-conditions	

Test case ID	Supervisor_TC_03
Test case name	View active tickets
Objective	To view active tickets
Pre-conditions	Supervisor is on supervisor portal page
Event sequence	1. Supervisor clicks on active tickets button
Input	Supervisor clicks on active tickets button
Output	Active tickets page is rendered
Post-conditions	

Test case ID	Supervisor_TC_04
Test case name	View closed tickets
Objective	To view active tickets
Pre-conditions	Supervisor is on supervisor portal page
Event sequence	1. Supervisor clicks on closed tickets button
Input	Supervisor clicks on closed tickets button
Output	Closed tickets page is rendered
Post-conditions	

Test case ID	Supervisor_TC_05
Test case name	View particular ticket
Objective	To view particular ticket

Pre-conditions	Supervisor is on view pending tickets tab
Event sequence	1. Supervisor clicks on view ticket for that particular ticket
Input	Supervisor clicks on view ticket for that particular ticket
Output	View ticket page is rendered
Post-conditions	

Test case ID	Supervisor_TC_06
Test case name	Assign ticket to staff
Objective	To assign pending ticket to staff
Pre-conditions	Supervisor is viewing pending ticket
Event sequence	1. Supervisor selects staff from assign staff select input 2. Supervisor clicks on assign ticket
Input	Supervisor clicks on assign ticket
Output	Ticket is assigned to selected staff
Post-conditions	

Test case ID	Supervisor_TC_07
Test case name	Reject ticket
Objective	To reject pending ticket
Pre-conditions	Supervisor is viewing a particular pending ticket
Event sequence	1. Supervisor clicks on reject ticket 2. Supervisor enters reason for rejecting ticket 3. Supervisor clicks reject ticket
Input	Supervisor clicks on reject ticket
Output	Ticket is rejected
Post-conditions	

Test case ID	Supervisor_TC_08
---------------------	-------------------------

Test case name	View supervisor profile
Objective	To view supervisor profile information
Pre-conditions	Supervisor is on any page while logged in
Event sequence	<ol style="list-style-type: none"> Supervisor clicks on profile button in Nav bar Supervisor profile page is rendered
Input	Supervisor clicks on profile button
Output	Supervisor profile page is rendered
Post-conditions	

Test case ID	Supervisor_TC_09
Test case name	Set new password
Objective	To set new supervisor password
Pre-conditions	Supervisor is on profile page
Event sequence	<ol style="list-style-type: none"> Supervisor enters new password and confirm new password Supervisor clicks on reset password
Input	Supervisor clicks on reset password
Output	Supervisor password is changed
Post-conditions	

Test case ID	Supervisor_TC_10
Test case name	Create tenant account page
Objective	To enter create tenant account page
Pre-conditions	Supervisor is logged in and on the supervisor landing page
Event sequence	<ol style="list-style-type: none"> Supervisor clicks on the create tenant account button in the landing page or the Navbar.
Input	Supervisor clicks on create account on the landing page or Navbar
Output	Create tenant account page is rendered
Post-conditions	

Test case ID	Supervisor_TC_11
Test case name	Create tenant account
Objective	To create tenant account
Pre-conditions	Supervisor is on create tenant account page
Event sequence	<ol style="list-style-type: none"> Supervisor enters new tenant account details Supervisor clicks on create account button Alert for new tenant account
Input	Supervisor clicks on create account
Output	Success notification: New tenant account alert is created
Post-conditions	

```

7 scenarios (7 passed)
23 steps (23 passed)
0m07.535s (executing steps: 0m07.028s)

```

```

Share your Cucumber Report with your team at https://reports.cucumber.io

Command line option:  --publish
Environment variable:  CUCUMBER_PUBLISH_ENABLED=true

More information at https://cucumber.io/docs/cucumber/environment-variables/

To disable this message, add this to your ./cucumber.js:
module.exports = { default: '--publish-quiet' }

```

Staff Portal

Test case ID	Staff_TC_01
Test case name	Staff Logins in to Staff account via Staff Login Page
Objective	For Staffs to type their email and password into the Staff Login Page and to log into Staff Account
Pre-conditions	<ol style="list-style-type: none"> Staff is on Landlord Login Page

Event sequence	<ol style="list-style-type: none"> 1. Landlord Portal Login Page loads 2. Staff clicks on the empty email and password fields 3. Staff types in credentials 4. Staff presses the “Login” button 5. Staff Landing Page renders
Input	Staff enters login details
Output	If login details are correct, staff is redirected to Staff portal is rendered
Post-conditions	

Test case ID	Staff_TC_02
Test case name	View Staff Profile Information
Objective	For staff to view profile information
Pre-conditions	<ol style="list-style-type: none"> 1. Staff must already be logged into the Landlord Portal and he must click the “Profile” button
Event sequence	<ol style="list-style-type: none"> 1. Staff clicks on “Profile” button on Staff Landing Page 2. Profile Page is loaded
Input	Staff clicks on “Profile” button
Output	Staff Profile Page is rendered
Post-conditions	

Test case ID	Staff_TC_03
Test case name	Set New Password
Objective	For Staff to reset password
Pre-conditions	Staff has to be in Profile Page
Event sequence	<ol style="list-style-type: none"> 1. Staff types in new password and re enters the same password into the text box 2. Staff clicks “reset password” button 3. Password is reset alert pop up
Input	Staff new password and re entered, same password
Output	Pop up for password reset

Post-conditions	
-----------------	--

Test case ID	Staff_TC_04
Test case name	View All Active Tickets
Objective	Allow Staff to view all active tickets
Pre-conditions	Staff clicks on “Active” tickets button
Event sequence	<ol style="list-style-type: none"> 1. Staff clicks on “Active” tickets button 2. Active tickets Page is rendered
Input	Staff clicks on “Active” tickets button
Output	Active tickets Page is rendered
Post-conditions	

Test case ID	Staff_TC_05
Test case name	View Active Ticket
Objective	Allow Staff to view active tickets
Pre-conditions	Staff clicks on “View Ticket” button
Event sequence	<ol style="list-style-type: none"> 1. Staff clicks on “View Ticket” button 2. Ticket is rendered
Input	Staff clicks on “View Ticket” button
Output	Ticket is rendered
Post-conditions	

Test case ID	Staff_TC_06
Test case name	View All Pending Tickets
Objective	Allow Staff to view all pending tickets
Pre-conditions	Staff clicks on “Pending” tickets button
Event sequence	<ol style="list-style-type: none"> 3. Staff clicks on “Pending” tickets button 4. Pending tickets Page is rendered

Input	Staff clicks on “Pending” tickets button
Output	Pending tickets Page is rendered
Post-conditions	

Test case ID	Staff_TC_07
Test case name	View Pending Tickets
Objective	Allow Staff to view pending tickets
Pre-conditions	Staff clicks on “Pending” tickets button
Event sequence	<ol style="list-style-type: none"> 1. Staff clicks on “Pending” tickets button 2. Ticket is rendered
Input	Staff clicks on “View Tickets” button
Output	Ticket is rendered
Post-conditions	

Test case ID	Staff_TC_08
Test case name	View All Closed Tickets
Objective	Allow Staff to view all closed tickets
Pre-conditions	Staff clicks on “Closed” tickets button
Event sequence	<ol style="list-style-type: none"> 5. Staff clicks on “Closed” tickets button 6. Closed tickets Page is rendered
Input	Staff clicks on “Closed” tickets button
Output	Closed tickets Page is rendered
Post-conditions	

Test case ID	Staff_TC_09
Test case name	View Closed Tickets
Objective	Allow Staff to view closed tickets

Pre-conditions	Staff clicks on “Closed” tickets button
Event sequence	3. Staff clicks on “Closed” tickets button 4. Ticket is rendered
Input	Staff clicks on “View Tickets” button
Output	Ticket is rendered
Post-conditions	

Test case ID	Staff_TC_10
Test case name	Update Status of ticket to “Works Started”
Objective	Allow Staff to update ticket status to “Works Started”
Pre-conditions	Staff has to be in view ticket page Ticket status has to be at “Quotation Accepted”
Event sequence	1. Staff clicks on “Works Started” button 2. Ticket Status Changes to “Works Started” 3. Ticket with “Works Started” status is rendered
Input	Staff clicks on “Works Started” button
Output	Ticket with “Works Started” status is rendered
Post-conditions	

Test case ID	Staff_TC_11
Test case name	Update Status of ticket to “Works Ended”
Objective	Allow Staff to update ticket status to “Works Ended”
Pre-conditions	Staff has to be in view ticket page Ticket status has to be at “Works Started”
Event sequence	1. Staff clicks on “Works Ended” button 2. Ticket status changes to “Works Ended” 3. Ticket with “Works Ended” status is rendered
Input	Staff clicks on “Works Ended” button
Output	Ticket with “Work Started” status is rendered

Post-conditions	
-----------------	--

Test case ID	Staff_TC_12
Test case name	Upload Quotation
Objective	Allow Staff to upload quotation to the Database
Pre-conditions	Staff has to be in view ticket page Ticket status has to be at “Ticket Assigned”
Event sequence	<ol style="list-style-type: none"> 1. Staff set “quotation required” to “yes” 2. Staff clicks on “upload” button and chooses a file to upload 3. Staff clicks “submit” button 4. Ticket status is changed to “Quotation Uploaded” 5. Quotation is rendered in View Ticket Page 6. Ticket with “Quotation Uploaded” status is rendered
Input	<ol style="list-style-type: none"> 1. Staff set “quotation required” to “yes” 2. Staff clicks on “upload” button and chooses a file to upload 3. Staff clicks “submit” button
Output	Ticket with “Quotation Uploaded” status is rendered
Post-conditions	

Test case ID	Staff_TC_13
Test case name	Restart Works
Objective	Allow Staff to restart works
Pre-conditions	Staff has to be in view ticket page Ticket status has to be at “Works Rejected”
Event sequence	<ol style="list-style-type: none"> 1. Staff clicks on “Restart Works” button 2. Ticket status is changed to “Restart Works” 3. Ticket with “Restart Works” status is rendered
Input	Staff clicks on “Restart Works” button
Output	Ticket with “Restart Works” status is rendered
Post-conditions	

```
13 scenarios (13 passed)
33 steps (33 passed)
0m18.415s (executing steps: 0m16.888s)
```

Share your Cucumber Report with your team at <https://reports.cucumber.io>

Command line option: `--publish`

Environment variable: `CUCUMBER_PUBLISH_ENABLED=true`

More information at <https://cucumber.io/docs/cucumber/environment-variables/>

To disable this message, add this to your `./cucumber.js`:

```
module.exports = { default: '--publish-quiet' }
```

Admin Portal

Test case ID	Admin_TC_01
Test case name	Login to admin account
Objective	To log in to admin account
Pre-conditions	Admin is on admin landing page
Event sequence	1. Admin enters login credentials 2. Admin clicks on login
Input	1. Admin clicks on login
Output	1. Admin portal page is rendered
Post-conditions	

Test case ID	Admin_TC_02- part 1
Test case name	Manage all buildings and account information
Objective	To view all buildings and account information
Pre-conditions	Admin is on admin landing page
Event sequence	1. Admin clicks on manage accounts in landing page or manage accounts in navbar

Input	1. Admin clicks on manage accounts button
Output	1. Manage buildings page is rendered
Post-conditions	

Test case ID	Admin_TC_02- part 2
Test case name	View particular building
Objective	To view one building
Pre-conditions	Admin is on manage buildings page
Event sequence	1. Admin clicks on one building in manage buildings page
Input	1. Admin clicks on one building in manage buildings page
Output	1. Particular building page is rendered
Post-conditions	

Test case ID	Admin_TC_02 - part 3
Test case name	View particular building
Objective	To view one building
Pre-conditions	Admin is on manage buildings page
Event sequence	2. Admin clicks on one building in manage buildings page
Input	2. Admin clicks on one building in manage buildings page
Output	2. Particular building page is rendered
Post-conditions	

Test case ID	Admin_TC_03 - part 1
---------------------	-----------------------------

Test case name	Enter create supervisor account
Objective	To enter create supervisor account
Pre-conditions	Admin is logged in
Event sequence	1. Admin clicks on create supervisor account in navbar OR clicks on create supervisor account in landing page
Input	1. Admin clicks on create supervisor account in navbar OR clicks on create supervisor account in landing page
Output	1. Create supervisor account page is rendered
Post-conditions	

Test case ID	Admin_TC_03 - part 2
Test case name	Create supervisor account
Objective	To create supervisor account
Pre-conditions	Admin is in create supervisor account page
Event sequence	1. Admin enters supervisor details 2. Admin clicks on create account 3. Supervisor account is created
Input	1. Admin clicks on create account
Output	1. Supervisor account is created
Post-conditions	

Test case ID	Admin_TC_04 - part 1
Test case name	Enter create staff/engineer account
Objective	To enter create supervisor account
Pre-conditions	Admin is logged in
Event sequence	4. Admin clicks on create staff account in navbar OR clicks on create

	staff account in landing page
Input	2. Admin clicks on create staff account in navbar OR clicks on create staff account in landing page
Output	2. Create staff account page is rendered
Post-conditions	

Test case ID	Admin_TC_04 - part 2
Test case name	Create staff account
Objective	To create staff account
Pre-conditions	Admin is in create staff account page
Event sequence	5. Admin enters staff details 6. Admin clicks on create account 7. Staff account is created
Input	3. Admin clicks on create account
Output	3. Staff account is created
Post-conditions	

```
4 scenarios (4 passed)
12 steps (12 passed)
0m08.859s (executing steps: 0m08.593s)
```

Share your Cucumber Report with your team at <https://reports.cucumber.io>

Command line option: `--publish`

Environment variable: `CUCUMBER_PUBLISH_ENABLED=true`

More information at <https://cucumber.io/docs/cucumber/environment-variables/>

To disable this message, add this to your `./cucumber.js`:

```
module.exports = { default: '--publish-quiet' }
```


Fuzzing

Test case ID	TC_Fuzz_1
Test case name	Fuzzing urls
Objective	To test if the correct url redirects to an authorised/unauthorised page
Pre-conditions	NA
Event sequence	<ol style="list-style-type: none">1. User types in random url<ol style="list-style-type: none">a. If user enters an unauthorised url, they must be redirected to the unauthorised pageb. If user enters an authorised url, they must be not be redirected to the unauthorised page
Test case descriptors	Test if our unauthorised/authorised routing works

```
✓ user should be directed to the appropriate page if path is /createtennantacc/filter (6 ms)
✓ user should be directed to the appropriate page if path is /staffportal (6 ms)
✓ user should be directed to the appropriate page if path is /createtennantacc (9 ms)
✓ user should be directed to the appropriate page if path is / (13 ms)
✓ user should be directed to the appropriate page if path is /supervisorportal/tenantlogin/manageacc (7 ms)
✓ user should be directed to the appropriate page if path is /adminportal/ticket/adminloginadminloginsupervisorportal (9 ms)
✓ user should be directed to the appropriate page if path is /adminportal/ticket/createstaffaccadminportalstaffportal (7 ms)
✓ user should be directed to the appropriate page if path is /supervisorportal/ticket/ (7 ms)
✓ user should be directed to the appropriate page if path is /tenantportal/landingpage/tenantportal/tenantportalticket (5 ms)
✓ user should be directed to the appropriate page if path is /building (8 ms)
✓ user should be directed to the appropriate page if path is /adminportal/landingpage/createstaffaccbuilding/adminportal (12 ms)
✓ user should be directed to the appropriate page if path is / (29 ms)
```

We fuzz random urls (a random mix of unauthorised paths/gibberish) and use regex to check if the url is valid. Then, we check if the user is on the correct page.

```
✓ When the user enters a truly random url, the right page must be shown (8372 ms)
```

We also use fastcheck to generate random url strings to see if unauthorised/authorised routing works

Commit records to show workload distribution

July 2, 2023 – August 2, 2023

Period: 1 month ▾

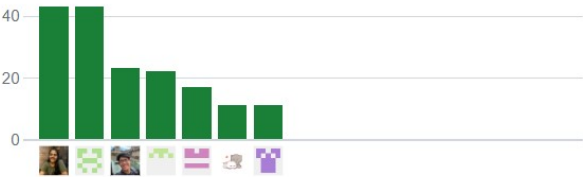
Overview

46 Active pull requests

1 Active issue

<div><div></div><div>45</div><div>Merged pull requests</div></div>	<div><div></div><div>1</div><div>Open pull request</div></div>	<div><div></div><div>1</div><div>Closed issue</div></div>	<div><div></div><div>0</div><div>New issues</div></div>
--	--	---	---

Excluding merges, **7 authors** have pushed **149 commits** to master and **170 commits** to all branches. On master, **117 files** have changed and there have been **25,862 additions** and **7,158 deletions**.



45 Pull requests merged by 6 people