PeakPulse Testing

# Unit Testing: subscriptions app

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case | Description | Steps | Expected Results | Actual Results | Pass/Fail | Comments |
| TC001 | **URL Test**:   * Verify that accessing the module's main URL returns a status code of 200 (OK). | Add app to project level urls and settings. Once urls/views and template is created attempt to navigate to url. | Response code should be 200 | Reverse failure | Fail | Code corrected and subsequently passes |
| TC002 | **Template Test**:   * Ensure that the module's main template renders without errors. * Check for the presence of essential elements specific to the module, such as forms, buttons, or data displays. | Once template html is completed, check rendering of elements in browser. Using console to inspect elements | Template should render as expected. | Template renders | Pass | - |
| TC003 | 1. **Functionality Test**:    * Test any specific functionalities introduced by the module, such as form submissions, database interactions, or API calls.    * For example: Submitting a form and verifying that the data is correctly processed and stored. | Once models and forms are created check.  Write code that checks data is being passed to and from the template. | Database should respond to POST and GET commands saving and retrieving data. | Terminal shows Post and GET requests being handled properly | Pass |  |
| TC004 | * + **Integration Test** (if applicable): Test interactions between the new module and existing modules to ensure seamless integration.   + For example: If the new module interacts with the user profile module, ensure that user profile data is correctly accessed and utilized. | Ensure all previous tests have passed.  If applicable check that functions and data can be imported from other apps/modules. | Apps/Modules which share functionality should communicate with each other. | Decorators for subscription only data works | Pass |  |
|  | **Edge Case Test** (if applicable):   * + Test the module's behavior in edge cases, such as empty inputs, invalid data, or extreme values.   + For example: Submitting a form with invalid data and verifying that appropriate error messages are displayed. | Ensure previous tests have passed.  Check that submitting a form with invalid data and verify that appropriate error messages are displayed. | Forms should handle incorrect behaviour from users with responsive error messages. | Empty Inputs such as postcode etc checked all mandatory form inputs return errors when not filled. | Pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |