**Product API**

A screenshot of a graph

Description automatically generated

1. **Controller:**
   1. **Test Documentation for CateAndSubCategoryControllerTest.java**
2. **testGetAllCate\_Success**

Purpose: To validate that the method retrieves all categories correctly.

Expected Outcome: The method should return a list of all categories.

1. **testGetAllSubCate\_Success**

Purpose: To check if the method correctly retrieves all subcategories.

Expected Outcome: The method should return a list of all subcategories.

1. **testGetAllSubCateByCate\_Success**

Purpose: To validate the correct retrieval of subcategories for a specific category.

Expected Outcome: The method should return a list of subcategories associated with the given category.

1. **testGetAllSubCateByCate\_NotFound**

Purpose: To test the method's behavior when there are no subcategories for a provided category.

Expected Outcome: The method might return an empty list, a not found response, or throw an exception.

* 1. **Test Documentation for FilterControllerTest.java**

1. **testFilterForNameApp\_Success**

Purpose: To validate the filtering functionality for a product name, specifically the name "app".

Expected Outcome: The method should correctly retrieve relevant categories, chains, and subcategories associated with products that contain the name "app".

1. **testFilterForSubCateApples\_Success**

Purpose: To test the filtering functionality based on a subcategory, in this case, "apples".

Expected Outcome: The method should correctly retrieve products and associated data (categories, chains, etc.) for the "apples" subcategory.

* 1. **Test Documentation for ChainControllerTest.java**

1. **testGetAllChains\_SuccessWithChains**

Purpose: To validate the functionality of retrieving all chains.

Expected Outcome: The method should return a list of all chains, specifically the mock chains: "Coles", "Aldi", and "Woolworths".

1. **testGetChainByName\_SuccessWithChain**

Purpose: To test if a specific chain can be retrieved by its name, in this case, "Coles".

Expected Outcome: The method should return the mock chain "Coles" with a rating of 4.33.

1. **testGetChainByName\_NullForWrongChainName**

Purpose: To validate the behavior when trying to retrieve a chain by a name that doesn't exist, specifically "Coless".

Expected Outcome: The method should return a "not found" message with an HTTP status of "NOT\_FOUND".

* 1. **Test Documentation for OpenTimeControllerTest.java**

1. **testGetOpenTimesByName**

Purpose: To validate the functionality of retrieving opening times for a store based on its name, specifically "*Woolworths Metro*".

Expected Outcome: The method should return the mock opening times:

"*Monday*" with times from "*900.0*" to "*1800.0*"

"*Tuesday*" with times from "*1000.0*" to "*1600.0*"

* 1. **Test Documentation for StoreControllerTest.java**

1. **testFindAll\_SuccessWithStores**

Purpose: To validate the functionality of retrieving all stores.

Expected Outcome: The method should return a list of all stores, specifically the mock stores: "Aldi" and "Woolworths".

1. **testFindStoresByPostcodesAndChains\_SuccessWithStores**

Purpose: To test the functionality of retrieving stores based on specified postcodes and chains.

Expected Outcome: The method should return stores that match the given postcodes ("1000" and "1020") and chains ("Woolworths" and "Coles").

* 1. **Test Documentation for ProductControllerTest.java**

1. **testGetAllProducts\_SuccessWithProducts**

Purpose: To validate the getAllProducts method in the ProductController class. It checks if the method can retrieve all products correctly.

Expected Outcome: The method should return a list of all products, based on the mock data set up for testing.

1. **testGetProductById\_SuccessWithProduct**

Purpose: To test the functionality of retrieving a product based on its ID.

Expected Outcome: The method should return the specific product associated with the given ID.

1. **testGetProductByName\_SuccessWithProducts**

Purpose: To validate the method that retrieves products based on a specified name.

Expected Outcome: The method should return products that match the given name.

1. **testGetProductByChain\_SuccessWithProducts**

Purpose: To test the functionality of retrieving products associated with a specific chain.

Expected Outcome: The method should return products associated with the specified chain.

1. **testGetProductByCategory\_SuccessWithProducts**

Purpose: To validate the method that retrieves products based on a given category.

Expected Outcome: The method should return products under the specified category.

1. **testGetProductBySubcategory\_SuccessWithProducts**

Purpose: To test the method that retrieves products based on a specified subcategory.

Expected Outcome: The method should return products under the specified subcategory.

1. **testGetSearch\_ByNameAp\_Success**

Purpose: To validate the search functionality for products by name.

Expected Outcome: The method should return products that match the search criteria.

1. **testGetSearch\_ByNameApAndSubcategoryGrapesApples\_Success**

Purpose: To test the product search functionality by name and subcategory.

Expected Outcome: The method should return products that match the given name and belong to the specified subcategories.

1. **testGetSearch\_ByNameApAndSubcategoryGrapesApplesAndChainWoolworths\_Success**

Purpose: To validate the product search functionality by name, subcategory, and chain.

Expected Outcome: The method should return products that match the search criteria, including name, subcategory, and chain.

1. **Repository:**
   1. **Test Documentation for ChainRepositoryImplTest.java**
2. **testFindAll\_SuccessWithChains**

Purpose: To test if the findAll method correctly fetches a list of chains.

Expected Outcome: The method should return the list of chains as set up in the test scenario.

1. **testGetByName\_SuccessWithChain**

Purpose: To validate if the getByName method correctly fetches a chain by its name.

Expected Outcome: The method should return the correct chain based on the provided name.

1. **testGetByName\_NullForWrongChainName**

Purpose: To check if the getByName method returns null when given an incorrect chain name.

Expected Outcome: The method should return null for an incorrect chain name.

* 1. **Test Documentation for OpenTimeRepositoryImplTest.java**

1. **testGetByName\_SuccessWithOpenTimes**

Purpose: To check if the getByName method correctly fetches a list of opening times by a store's name.

Expected Outcome: The method should return a list of opening times corresponding to the given store name.

1. **testGetByName\_NullForWrongName**

Purpose: To validate that the getByName method returns null when provided an incorrect store name.

Expected Outcome: The method should return null for an incorrect store name.

* 1. **Test Documentation for ProductRepositoryImplTest.java**

1. **testFindAll\_SuccessWithProducts**

Purpose: To validate that the findAll method fetches a list of products correctly.

Expected Outcome: The method should return the list of products as set up in the test scenario.

1. **testFindAll\_SuccessWithNoProducts**

Purpose: To check if the findAll method returns an empty list when no products are found.

Expected Outcome: The method should return an empty list.

1. **testFindAll\_FailureDueToInternalError**

Purpose: To test the behavior of the findAll method when an internal error occurs.

Expected Outcome: The method should handle the error gracefully, potentially throwing an exception or returning a specific error response.

1. **testGetById\_SuccessWithProductInfo**

Purpose: To validate that the getById method fetches product information correctly based on a given product ID.

Expected Outcome: The method should return the product details corresponding to the provided ID.

1. **testGetById\_SuccessWithNoProductInfo**

Purpose: To check if the getById method returns no product information when given an invalid or non-existent product ID.

Expected Outcome: The method should return no product details or a specific indicator (e.g., null or empty response).

1. **testGetById\_FailureDueToInternalError**

Purpose: To test the behavior of the getById method when an internal error occurs.

Expected Outcome: The method should handle the error gracefully.

1. **testGetByName\_SuccessWithProducts**

Purpose: To validate that products can be fetched correctly based on their name using the getByName method.

Expected Outcome: The method should return products matching the given name.

1. **testGetByName\_NoProductsFound**

Purpose: To check if the getByName method returns no products when provided with a name that doesn't match any product.

Expected Outcome: The method should return no products or a specific indicator (e.g., empty list).

1. **testGetByChain\_SuccessWithProducts**

Purpose: To validate that products associated with a specific chain can be fetched correctly.

Expected Outcome: The method should return products associated with the given chain.

1. **testGetByCategory\_SuccessWithProducts**

Purpose: To check if products under a specific category can be fetched correctly.

Expected Outcome: The method should return products under the specified category.

1. **testGetByCategory\_NoProductsFound**

Purpose: To validate that the getByCategory method returns no products when there are no products under a specified category.

Expected Outcome: The method should return an empty list or a specific indicator that no products are found.

1. **testGetByChain\_NoProductsFound**

Purpose: To check if the getByChain method returns no products when there are no products associated with a given chain.

Expected Outcome: The method should return an empty list or another specific indicator of no results.

1. **testGetBySubcategory\_SuccessWithProducts**

Purpose: To test if the getBySubcategory method fetches products correctly based on a specified subcategory.

Expected Outcome: The method should return products under the specified subcategory.

1. **testGetBySubcategory\_NoProductsFound**

Purpose: To validate that the getBySubcategory method returns no products when there are none under a specified subcategory.

Expected Outcome: The method should return an empty list or a specific indicator of no results.

1. **testGetSearch\_ByNameAppleAndCategoryFruitAndVeg\_Success**

Purpose: To test the product search functionality by name and category.

Expected Outcome: The method should return products that match the given name and belong to the specified category.

1. **testGetSearch\_ByNameAppleAndCategoryBakery\_EmptyResult**

Purpose: To validate the product search functionality by name and category where no results are expected.

Expected Outcome: The method should return an empty list, indicating that no products match the given criteria.

1. **testGetSearch\_ByNameAppleAndChainWoolworthsAldi\_Success**

Purpose: To test the product search functionality by name and multiple chains.

Expected Outcome: The method should return products that match the given name and are associated with the specified chains.

* 1. **Test Documentation for StoreRepositoryImplTest.java**

1. **testFindAll\_SuccessWithStores**

Purpose: To validate that the findAll method fetches a list of stores correctly.

Expected Outcome: The method should return the list of stores as set up in the test scenario.

1. **testFindStoresByPostcodesAndChains\_SinglePostcode1000**

Purpose: To check if the method fetches stores correctly for a single postcode.

Expected Outcome: The method should return stores associated with the given postcode.

1. testFindStoresByPostcodesAndChains\_NoStoresForPostcode1001

Purpose: To validate that the method returns no stores when provided with a postcode that doesn't have associated stores.

Expected Outcome: The method should return an empty list or a specific indicator of no results.

1. **testFindStoresByPostcodesAndChains\_Postcodes1000And1020**

Purpose: To test the functionality of finding stores by multiple postcodes.

Expected Outcome: The method should return stores associated with the specified postcodes.

1. **testFindStoresByPostcodesAndChains\_Postcodes1000And1020ChainWoolworths**

Purpose: To validate the functionality of finding stores by multiple postcodes and a specific chain.

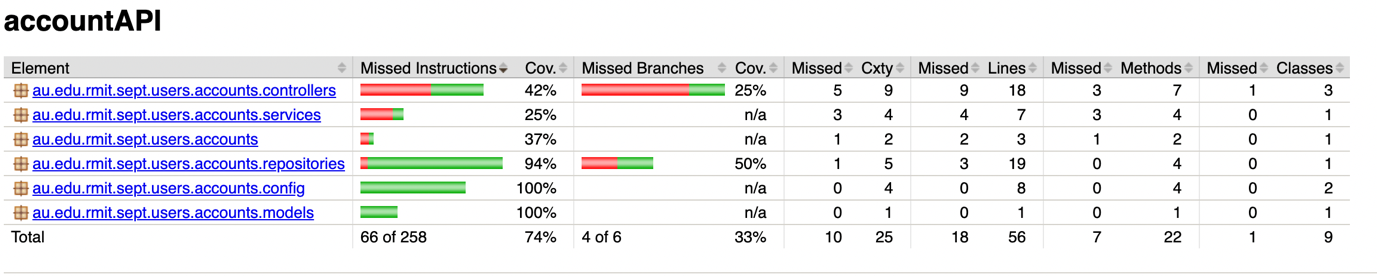
Expected Outcome: The method should return stores that match the given postcodes and chain.

1. **testFindStoresByPostcodesAndChains\_Postcodes1000And1020ChainsWoolworthsAndColes**

Purpose: To test the functionality of finding stores by multiple postcodes and multiple chains.

Expected Outcome: The method should return stores that match the given postcodes and chains.

**Account API**



1. **Controller:**
   1. **Test Documentation for AccountControllerTest.java**
      1. **newAccount\_should\_callCreateService**

Purpose: To ensure that when a new account is made that the service create is called.

Expected Outcome: The method should create a new store and run through the service create once.

* + 1. **get\_should\_returnAccountDetails\_When\_available**

Purpose: To validate when a get call is made, that the same details are returned.

Expected Outcome: The method should return the email that matches the one requested.

* + 1. **get\_should\_throwException\_when\_NotFound**

Purpose: To ensure that a runtime exception is called when an account is not found during a get call.

Expected Outcome: runtime exception.

1. **Test Documentation for AccountRepositoryImplTest.java**
   * 1. **create\_should\_addNewAccountToDB**

Purpose: To ensure that when the create call is made, a new entry is made in the database

Expected Outcome: When a new account is created, it can be called using get.

* + 1. **update\_should\_updateAccountToDB**

Purpose: To ensure that when an update call is made to an account, the updated information is in the database under the correct account.

Expected Outcome: When an account is updated, the get request reflects the changes.