# Integration Tests เขียนตามControllerที่ methodนี่Testได้ไหม

This section details the black box tests needed to ensure that each component of Wat Muen Ngoen Kong is

### Wat Muen Ngoen Kong -ในตารางเปนตัวอย่างนะ ยังไม่ได้ใส่ข้อมูลไรเลย

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the cross-device communication between the mobile device and the mobile device interface is functional. |
| **External Dependencies** | Ensure that the Mobile Device has wireless connectivity. |
| **Test Description** | 1) Start the FrugalPhone client application  2) Select the UPC Entry Type  3) Input a random 12-digit UPC  4) Send the UPC code |
| **Expected Results** | As long as wireless connectivity exists, the mobile device displays information regarding the product or a screen that explains that no products have been found that match the specified UPC code. |

### Wat Muen Ngoen Kong -

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the Mobile Device Interface can handle invalid Product Input Data. |
| **External Dependencies** | None |
| **Test Description** | 1) Start the FrugalPhone client application  2) Select the Product Name Entry Type  3) Input and send one asterisk (\*) character |
| **Expected Results** | The Mobile Device displays a message that says that no products matched the search criteria. |

### Wat Muen Ngoen Kong -

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the Mobile Device Interface can handle the case where no products match the Product Input Data. |
| **External Dependencies** | Find a UPC code that does not appear in the database. |
| **Test Description** | 1) Start the FrugalPhone client application  2) Select the UPC Entry Type  3) Input and send the UPC code that is not in the database |
| **Expected Results** | The Mobile Device displays a message that says that no products matched the search criteria. |

### Wat Muen Ngoen Kong -

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the Mobile Device Interface can handle the case where only one product matches the Product Input Data. |
| **External Dependencies** | Find the UPC code of a product in the database that only has one entry in the database. |
| **Test Description** | 1) Start the FrugalPhone client application  2) Select the UPC Entry Type  3) Input and send a UPC code of a product with only one entry in the database |
| **Expected Results** | The Mobile Device displays the single result. |

### Wat Muen Ngoen Kong -

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the Mobile Device Interface can handle the case where multiple products match the Product Input Data. |
| **External Dependencies** | Find the name of a product in the database that has multiple entries in the database. |
| **Test Description** | 1) Start the FrugalPhone client application  2) Select the Product Name Entry Type  3) Input and send a product name of a product with many entries in the database |
| **Expected Results** | The Mobile Device displays multiple results. |

## Product Pricing Data Retailer Interface – FrugalPhone Server Database

### Unsuccessful Product Pricing Data Upload

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the Product Pricing Data Retailer Interface can handle incorrectly formatted Product Pricing Data. |
| **External Dependencies** | None |
| **Test Description** | 1) Make a Product Pricing Data Location with incorrectly formatted Product Pricing Data  2) Add the Product Pricing Data Location to the list of Product Pricing Data Locations  3) Wait for Product Pricing Data Refresh Rate |
| **Expected Results** | When the Product Pricing Data updates, a message is logged to the Admin Console saying that the Product Pricing Data Location contained incorrectly formatted Product Pricing Data. |

### Product Pricing Data Location Addition

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the Product Pricing Data Retailer Interface can handle new Product Pricing Data Locations. |
| **External Dependencies** | Create one (1) properly formatted Product Pricing Data Location with non-duplicate Product Pricing Data. |
| **Test Description** | 1) Conduct the Add Feed File Location test (1.3.7)  2) Start the FrugalPhone client application  3) Select the UPC Entry Type  4) Input and send the UPC code corresponding to a Product Pricing Data |
| **Expected Results** | 1) The Add Feed File Location test (1.3.7) performs as expected.  2) The Mobile Device displays a result including the one in the Product Pricing Data Location. |

### Product Pricing Data Addition

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the Product Pricing Data Retailer Interface can handle new Product Pricing Data. |
| **External Dependencies** | Find an existing (modifiable) Product Pricing Data Location that has been uploaded. |
| **Test Description** | 1) Add a new, non-duplicate Product Pricing Data to the existing Product Pricing Data Location  2) Wait for Product Pricing Data Refresh Rate  3) Start the FrugalPhone client application  4) Select the UPC Entry Type  5) Input and send the UPC code corresponding to the added Product Pricing Data |
| **Expected Results** | The Mobile Device displays a result including the one in the Product Pricing Data. |

### Duplicate Product Pricing Data

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the Product Pricing Data Retailer Interface can handle duplicate Product Pricing Data. |
| **External Dependencies** | Find an existing (modifiable) Product Pricing Data Location that has been uploaded. |
| **Test Description** | 1) Add a new, duplicate Product Pricing Data to the existing Product Pricing Data Location  2) Wait for Product Pricing Data Refresh Rate  3) Start the FrugalPhone client application  4) Select the UPC Entry Type  5) Input and send the UPC code corresponding to the added Product Pricing Data |
| **Expected Results** | The Mobile Device displays a result including the one in the Product Pricing Data, but the duplicate entry is only shown once. |

### Modified Product Pricing Data Entry

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the Product Pricing Data Retailer Interface can handle modified Product Pricing Data. |
| **External Dependencies** | Find an existing (modifiable) Product Pricing Data Location that has been uploaded |
| **Test Description** | 1) Modify a Product Pricing Data in the existing Product Pricing Data Location  2) Wait for Product Pricing Data Refresh Rate  3) Start the FrugalPhone client application  4) Select the UPC Entry Type  5) Input and send the UPC code corresponding to the modified Product Pricing Data |
| **Expected Results** | The Mobile Device displays a result including the modified Product Pricing Data, with the most up-to-date product details. |

## Admin Console – FrugalPhone Server Database

### Login Success

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the System Administrator can login to the Admin Console. |
| **External Dependencies** | The user has a System Administrator account. |
| **Test Description** | 1) Type the username and password into the Admin Console login screen  2) Click the “Sign In” button |
| **Expected Results** | The user is directed to the AdminOptions Screen (SDS 4.5.1.3) |

### Incorrect Login Failure

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that incorrect username and password pairs do not allow the user to access the main features of the Admin Console. |
| **External Dependencies** | The user knows an incorrect System Administrator account username and password pair. |
| **Test Description** | 1) Type the incorrect username and password pair into the Admin Console login screen  2) Click the “Sign In” button |
| **Expected Results** | The user is directed to the LoginErrorRetry Screen (SDS 4.5.1.2) |

### Add System Administrator

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that System Administrator accounts can be added. |
| **External Dependencies** | None |
| **Test Description** | 1) Login to the Admin Console  2) Click “Manage System Administrators”  3) Click “Add System Administrator”  4) Enter a valid username, password, and confirmation password  5) Click the “Add System Administrator” button  6) Click the “Logout” button  7) Perform the Login Success test (1.3.1) with the username and password in step (4) |
| **Expected Results** | 1) No error message appears after step (5)  2) The same results from the Login Success test (1.3.1) |

### Remove System Administrator

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that System Administrator accounts can be removed. |
| **External Dependencies** | Know the username and password of an existing System Administrator account. |
| **Test Description** | 1) Login to the Admin Console  2) Click “Manage System Administrators”  3) Click “Delete System Administrator”  4) Select a System Administrator account whose username and password you know  5) Click the “Delete Selected” button  6) Click the “Logout” button  7) Perform the Incorrect Login Failure test (1.3.2) with the username and password in step (4) |
| **Expected Results** | 1) No error message appears after step (5)  2) The same results from the Incorrect Login Failure test (1.3.2) |

### Change System Administrator Password

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the password of a System Administrator account can be modified. |
| **External Dependencies** | Know the username and password of a System Administrator account. |
| **Test Description** | 1) Login to the Admin Console  2) Click “Manage System Administrators”  3) Click “Change Password”  4) Enter the existing password, a valid new password, and a confirmation of the new password  5) Click the “Submit Change” button  6) Click the “Logout” button  7) Perform the Incorrect Login Failure test (1.3.2) with the username and **old** password  8) Perform the Login Success test (1.3.1) with the username and **new** password |
| **Expected Results** | 1) No error message appears after step (5)  2) The same results from the Incorrect Login Failure test (1.3.2) after step (7)  3) The same results from the Login Success test (1.3.1) after step (8) |

### Change Connection Timeout

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the connection timeout server management parameter can be modified. |
| **External Dependencies** | None |
| **Test Description** | 1) Login to the Admin Console  2) Click “Manage Server Parameters”  3) Change the number in “Connection Timeout” field  4) Click the “Submit Changes” button  5) Click “View Server Status”  6) Click “Manage Server Parameters” |
| **Expected Results** | 1) No error message appears after step (4)  2) The number in the “Connection Timeout” field matches the number input in step (3) |

### Add Feed File Location

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the connection timeout server management parameter can be modified, specifically by adding a Feed File Location. |
| **External Dependencies** | None |
| **Test Description** | 1) Login to the Admin Console  2) Click “Manage Server Parameters”  3) Add a Feed File Location  4) Click the “Submit Changes” button  5) Click “View Server Status”  6) Click “Manage Server Parameters” |
| **Expected Results** | 1) No error message appears after step (4)  2) The Feed File Location input in step (3) is in the list of Feed File Locations |

### Remove Feed File Location

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the connection timeout server management parameter can be modified, specifically by removing a Feed File Location. |
| **External Dependencies** | At least one existing Feed File Location |
| **Test Description** | 1) Login to the Admin Console  2) Click “Manage Server Parameters”  3) Remove a Feed File Location  4) Click the “Submit Changes” button  5) Click “View Server Status”  6) Click “Manage Server Parameters” |
| **Expected Results** | 1) No error message appears after step (4)  2) The Feed File Location removed in step (3) is no longer in the list of Feed File Locations |

### Change Feed File Refresh Rate

|  |  |
| --- | --- |
| **Purpose** | The purpose of this test is to ensure that the feed file refresh rate server management parameter can be modified. |
| **External Dependencies** | None |
| **Test Description** | 1) Login to the Admin Console  2) Click “Manage Server Parameters”  3) Change the number in “Feed File Refresh Rate” field  4) Click the “Submit Changes” button  5) Click “View Server Status”  6) Click “Manage Server Parameters” |
| **Expected Results** | 1) No error message appears after step (4)  2) The number in the “Feed File Refresh Rate” field matches the number input in step (3) |