I. Developing a FitNesse Wiki site

You are going to develop a set of FitNesse Wiki pages that will be used by the developers attending the Practical Test Driven Development.

A. Creating your top level page (your main page)

Create a new Wiki page (call it whatever you like but note this page will point to other wiki pages, so it acts as a suite). Be sure to specify the link in such a way that the item the user clicks on is not a URL but an alias.

Once the page has been created, remove any text you find in the edit pane and enter a title like

A series of general learning tests

Be sure to make it bold and stand out. {see course notes and look under User Guide/Markup Language Reference/Headers}

Add a comment regarding what the pages are about.

Once you have completed these details, save the page using the save button (can be found in the lower left part of the browser) and add a footer indicating your companies details.

B. Creating Subpages

You are going to create four subpages from your main page. The four pages are A Basic Hello World, String Concatenation, Loan Calculator and, Open New Account. For each numbered bullet below, be sure that you are looking at your main page (the one where the title is **"A series of general learning tests"**. For each page make sure that it is a test page

1. Edit the current page (your main page).
    a. Navigate to below any current text within the page (leaving an empty line before you begin to type) and enter ReverseString.
    b. Save the page.
    c. Notice on the main page there is a piece of text ReverseString[?]. Select the [?] and FitNesse will create a new page.
    d. Edit it as described above (creating your top level page) and enter information specific to this page. When completed select save.
    e. You should notice that the URL is showing this page being directly off the fitnesse server and not a subpage of your main page.
2. Edit the current page (your main page)
    a. Navigate to below any current text within the page (leaving an empty line before you begin to type) and enter ConcatenateStrings.
    b. Save the page.

   c. Notice on the main page there is a piece of text ConcatenateStrings [?]. Select the [?] and FitNesse will create a new page.
   d. Edit it as described above (creating your top level page) and enter information specific to this page. When completed select save.
   e. You should notice that the URL is showing this page being directly off the fitnesse server and not a subpage of your main page.
3. Edit the current page (your main page)
   a. Navigate to below any current text within the page (leaving an empty line before you begin to type) and enter[[Loan Calculator][LoanCalculatorTests]] (pay close attention to the way in which we have applied Camal Case).
   b. Save the page.
   c. Notice on the main page there is a piece of text Loan Calculator [?]. Select the [?] and FitNesse will create a new page.
   d. Edit it as described above (creating your top level page) and enter information specific to this page. When completed select save.
   e. You should notice that the URL is showing this page being directly off the fitnesse server and not a subpage of your main page.
4. Edit the current page (your main page)
   a. Navigate to below any current text within the page (leaving an empty line before your begin to type) and enter >OpenNewAccountPage (pay close attention to the way in which we have applied Camal Case and the '>' character).
   b. Save the page.
   c. Notice on the main page there is a piece of text OpenNewAccountPage, select this text and FitNesse will create a new page (unlike the previous examples, this page is a subpage of your main page, so inherits the header and footers. A subpage may be accessed at anytime by typing the URL <parentpage>.<subpage> without the angle brackets and with the dot between the pages).

C. Working with Fitnesse tables

These labs are designed to help you understand how to work with FITNesse tables. Each lab relates directly to a lab from above, so lab 1 here relates to lab 1 above in section B. For each lab add the following lines to the top of the edit pane (you will find these settings in the file **default_fitnesse_paths.txt**)

!define TEST_SYSTEM {slim}

!define COMMAND_PATTERN {%m -r fitSharp.Slim.Service.Runner, c:\java\fitnesse\dotnet4\fitSharp.dll %p}

!define TEST_RUNNER {c:\java\fitnesse\dotnet4\Runner.exe}

!path C:\Java\fitnesse\fitnesse_learning_tests\FitNesse_tests.dll

1. **Echo:** (from lab 1 above) In this lab, you will create a table that echoes whatever you put in the table in reverse character order. Open the Wiki page ReverseString and select Edit. In the edit frame create the table shown here

   Add the following statements to your page just after the *!path* statement and before the definition of table (leave a blank afterwards)

   |import |
   |FitNesse_tests|

Inverted Echo	
Original Text	Text Reversed?
Hello World	
My name is Selvyn	nyvleS si eman yM

   When completed, save and test the page.

2. **Concatenation:** (from lab 2 above) In this lab, you will create a table that concatenates two strings. Open the Wiki page ConcatenateStrings and create the table shown here

   Add the following statements to your page just after the *!path* statement and before the definition of table (leave a blank afterwards)

   |import |
   |FitNesse_tests|

Concatenate Strings		
String 1	String 2	Concatenate?
Hello	World	HelloWorld
Some other	Value	**$value=**
$value	Night	OneDay

Notice use of wiki symbol

   When completed, save and test the page.

3. **Loan Calculator:** (from lab 3 above) In this lab you are going to define and create two tables, the first allowing you to supply some basic figures for a loan (such as loan amount, period and interest rate) in which the result should be a figure indicating the monthly repayments. The second allowing you to invoke predefined operations on a fixture. This is more complex than the previous examples because here you have to define the tables and the data. Use Excel to define the table,

copy it into FitNesse and use the format button to convert the excel sheet into a fitnesse table. Use the following characteristics for your first table, a decision table

Add the following statements to your page just after the *!path* statement and before the definition of table (leave a blank afterwards)

|import |
|FitNesse_tests|

a. Table name: **Loan Calculator**
b. Field names: (in this order) Loan Amount, Period, Percentage Down, Fixed Rate, Monthly Payments (this is a result column)
c. Create some data for your table (ask the lecturer to give you an expected value for the monthly payments or use excel's functions if you know how to)

For the second part of the question you are going to use a fitnesse script table rather than a decision table

d. Table name: **Loan Calculator Script** with parameters (in this order), principal, length of loan in months, interest rate and initial down payment
e. The following operations are available
    i. **Monthly Payments**; returns the payments per month (ppm),
    ii. **Monthly Payments Match**; takes an amount as input and returns true if it matches the monthly payments,
    iii. **Initial Loan**; returns the initial loan amount less the initial down payment,
    iv. **Initial Loan Minus**; returns an initial loan amount but allows you to specify the down payment as a percentage of the original loan
f. Use the script keywords (such as check, reject, ensure, show etc) to drive the underlying fixture. Try using wiki symbols; $<symbol name>= to create and assign a value to a symbol and, $<symbol name> to use a symbol.

4. **Opening an Account:** In this lab, you will create two tables to test opening an account (see handout for this case study). This lab is more complex than the previous ones because it demonstrates the use of FitNesse and TDD in a real world workflow context.

To get started we need to create two child pages from the current page. Give your pages the following names, NewAccountApplication and NewApplicationVerification. Once you have the two new pages, change the property of the OpenNewAccountPage to suite (be sure that the two subpages are test pages). Add the two child pages to the suite using **!see**.

1. Open the wiki page NewAccountApplication and select edit. Add any information such as titles and comments if you wish. Add the information in **default_fitnesse_paths.txt** but change *FitNesse_tests.dll* field to *BankingSystem.exe*.

2. Begin by defining a **Open New Account**, this decision table will be used to open an account. Focus on the following columns; Name, DOB (dd/mm/yyyy), Employment (Y, N, N/S), Three IDs (Y, N, Y/NS), IDs copied (Y, N), Opening Deposit, Application Date (dd/mm/yyyy) and Application Number
3. Specify two output columns; Application Status (V, Q, C)

If you have constructed this table correctly, when you test it you should get no errors. You now need to populate it with some data. Use the data in the excel sheet provided. Save and test the page, you may get errors, if so your data is incorrect for the business rules.

Another problem that should be highlighted, is that we have assigned account numbers to the Account Number column. This is a redundant test. The real test is in the application status.

4. Add another column to your table; Account State (H, A, I)

We now need to test the last part of the workflow, this will be done in the Wiki page NewApplicationVerification

1. Open the wiki page NewApplicationVerification and select edit. Add any information such as titles and comments if you wish. Add the information in **default_fitnesse_paths.txt** but change *FitNesse_tests.dll* field to *BankingSystem.exe*.
2. Define the decision table **Deposit Taken**. It should have the columns as shown in the excel sheet
3. For each row in the decision table Open New Account, there should be a row in this table.
4. A value in the Amount taken cell should match the value in the Opening Deposit cell of Open New Account. To achieve this, we need to carry a value over from the OpenNewAccount table into the DepositTaken table. This will be the application number. The application form has an application number and, the application form can be used to navigate to the Customer or the Account if the application form has cleared.