SOFTWARE SOLITIONS

SOFIA

Team Two Implementation Report

Daniel Berhe

Document Control

Editor	Date	Update
Daniel Berhe	**03/06/2015**	**Document created**

Table of Contents

Team Two Implementation Report

1.0 Development Team Two

Development team two is comprised of three members. Daniel Berhe (Specialist Software Developer), Sam Raeburn (Lead Software Tester) and Jake Ransom (Marketing Manager). The team was tasked with implementing functionalities to LearnEasy such as: progress tracking, marks collating, listing of available lessons and the listing of recently opened Lessons.

2.0 Development Process

Initially was carried out with all members of the team being present. Driver/Navigator system was employed. During the initial development, the team modified classes RunTimeData, AnswerBox, AnswerBoxHandler, MultipleChoice and MultipleChoiceHandler. This was a necessary step in finding out questions were attempted. This was done by simply creating methods which check the disabled (greyed out) mark buttons in the appropriate classes.

The Lead Software Tester (LST) proceeded on his own to create a very useful feature which displays a dialog box. The dialog box appears if the user tries to go to the next page without completing all the questions. This dialog box gives the user the option of hiding it in the future. This is done through selecting the “Don’t show me again” check box. This choice is remembered whilst the lesson remains open.

The Specialist Software Developer (SSD) continued on implementing the marks and progress tracking functionality. A new package named learneasy.trackprogress which contains the class ProgressTracker was created. This class keeps track of the visited pages in a given lesson by storing Boolean variable true in an array index corresponding to the lessons page number. This is necessary to ensure that marks on a given page is not counted more than once. Another array with the size corresponding to the number of pages of a lesson is also present in this class. The purpose of this array is to record the total page marks. Similar to the visited pages tracker, marks are stored in the array’s index corresponding to the page numbers. Storing marks for pages separately provides flexibility in the future if our company desires to add more features.

The SSD also created another package named learneasy.homepage which contains the class HomePage. This class is responsible for features such as storing user’s preferences and storing recently opened lessons. This class makes use of java preference API and directory chooser. Java preference API allows storing information such as string, integer and Boolean. This information is retained on the PC even after the program is closed and can be retrieved. When LearnEasy is run, a dialog box is displayed to the user prompting them to choose a default folder for the program. The user can press the “Browse” button to open standard windows file explorer. This allows the user to use a default folder. Having a default folder enables the program to list all the lessons available in that folder to be displayed on the home page of the program. The available lessons are listed in the form of hyperlinks, this enables the user to simply click on the hyperlink to open a lesson. This greatly improves user experience of the program. The dialog box gives the user the option to hide it in the future. This choice as well as the chosen default folder is stored in the java preference API in Boolean format with the key “showPrompt” and “Default folder” respectively. This choice is remembered even if the program is closed down.

The listing of the available lessons in the default folder is carried out by a method written by the SSD, `listAvailableLessons()`, located in the class RunTimeData. This method simply accesses the java preference API for the file path of the user chosen default folder. It then displays all the files with the file extension .xml in that folder in the form of hyperlinks. Listing the recently opened lesson is carried out by a method written by the SSD, `displayRecentlyOpenedLessons()`, residing in class RunTimeData. This method accesses the java preference API to extract the file path of the lessons. It then displays them as hyperlinks which can then be used to open the relevant lesson.

The listing of recently opened lesson is slightly more complex. The approach taken is easily understood by consulting figure 1 below. The index starts at 2 because index 0 and index 1 of the java preference API are storing “Default folder” and “showPrompt” respectively.

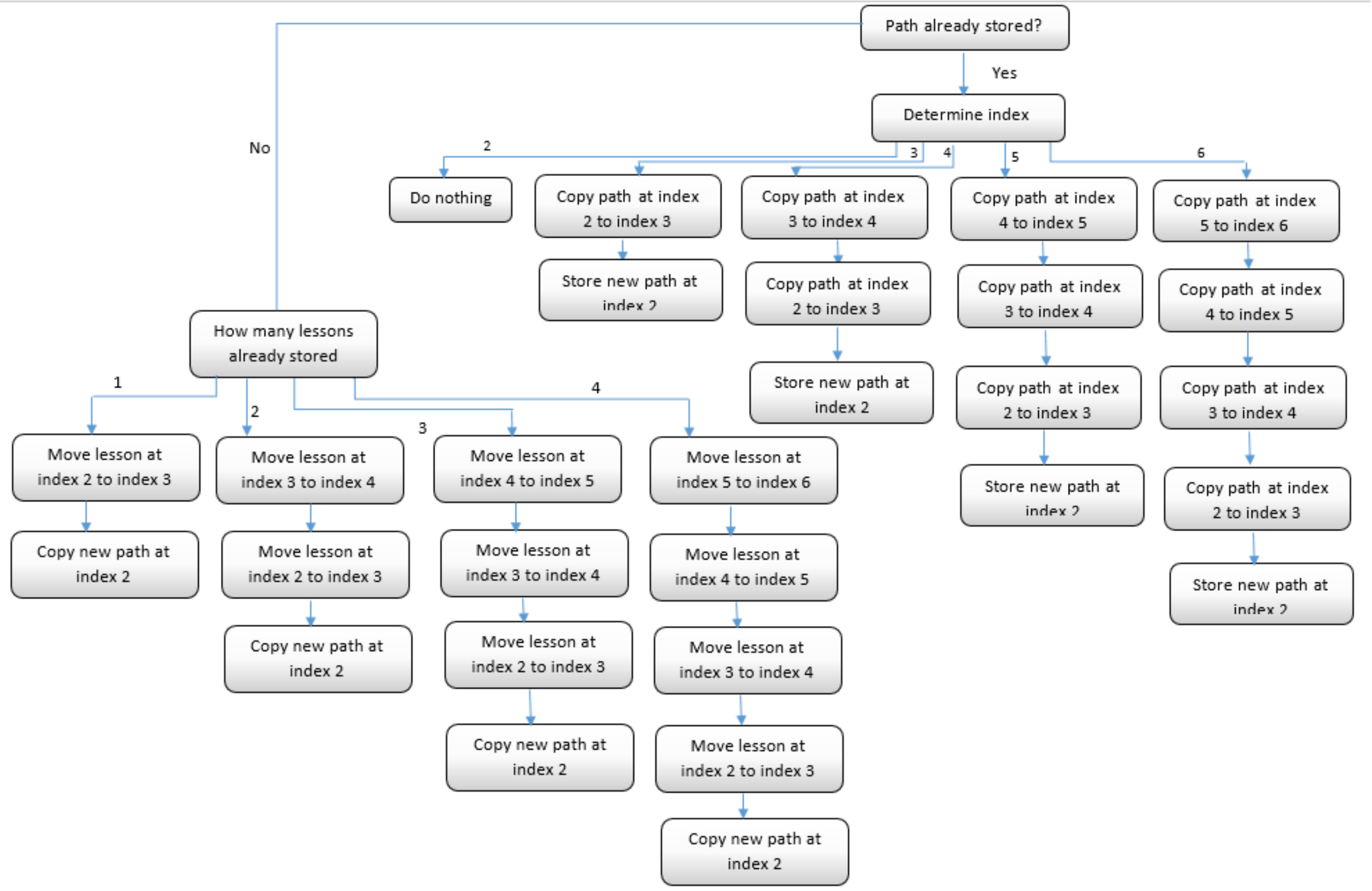
Path already stored?
Yes
Determine index
No
2
3
4
5
6
Do nothing
Copy path at index 2 to index 3
Store new path at index 2
Copy path at index 3 to index 4
Copy path at index 2 to index 3
Store new path at index 2
Copy path at index 4 to index 5
Copy path at index 3 to index 4
Copy path at index 2 to index 3
Store new path at index 2
Copy path at index 5 to index 6
Copy path at index 4 to index 5
Copy path at index 3 to index 4
Copy path at index 2 to index 3
Store new path at index 2
How many lessons already stored
1
2
3
4
Move lesson at index 2 to index 3
Copy new path at index 2
Move lesson at index 3 to index 4
Move lesson at index 2 to index 3
Copy new path at index 2
Move lesson at index 4 to index 5
Move lesson at index 3 to index 4
Move lesson at index 2 to index 3
Copy new path at index 2
Move lesson at index 5 to index 6
Move lesson at index 4 to index 5
Move lesson at index 3 to index 4
Move lesson at index 2 to index 3
Copy new path at index 2

Figure 1. Flow diagram of recently opened lessons organisation