Industrial Programming Coursework B

Duncan Cameron
Software Engineering
Heriot-Watt University
dac31@hw.ac.uk
H00153427

November 30, 2016

Chapter 1

Introduction

For this coursework task I created a simple data analysis tool for a document tracker built in python.

Chapter 2

Requirements' Checklist

I completed all of the tasks and requirements for the projects

Chapter 3

Design Considerations

3.1 Data Representation Classes

I created 3 classes to represent the data that was read in from the input file: A Document class, a Reader class and a View class. I decided on this structure as it allowed for a simple, natural way to explore the dataset and the Documents and Readers could be linked by views.

3.1.1 Document

The Document Class stores information about the document, including the document UUID and a list of the views of the document. It contains functions to get the data required for creating relevant histograms for the document.

3.1.2 Reader

The Reader class contains the UUID and views of a reader (visitor). There is also a function to get the users total view time.

3.1.3 DocumentView

The DocumentView class links the Document and Reader class as well as holding additional information about each view - the time the page was viewed for, the user-agent for detailing the browser that was used for the view and the country from which it was viewed. I decided to store the user-agent and country here as opposed to the Reader class as it is possible for a reader to view from different countries or using different browsers.

3.1.4 DataLoader

To load the data I created a class called DataLoader used to load the data in from a JSON file. When created it takes in a filename and will access try to load it and create the relevant classes based on the data.

It contains 2 fields for accessing the readers and documents. They are both dictionaries in which the key is the UUID and the value is the object. I chose this approach as it means that if a key is given the related document can be accessed in O(1) time.

The data loader is also able to either load in certain types of data based off of the event type of each instance in the file by taking in an optional boolean parameter in its constructor. If the value is true it will use all of the event types (as is required by task 3). Otherwise it will only take in 'pageread' and 'pagereadtime' instances as they represent actual reads of the document.

I used the data loader as a class instead of a series of global variables and functions so that multiple files could be loaded and independently managed. While this is not done in the program, it would be possible by simply creating another instance of the DataLoader class. This is also a likely extension of the program as it would be useful for situations such as comparing multiple datasets.

3.2 GUI Classes

To implement the GUI I used the tkinter library. My implementation involved creating a few classes. The Controller class was used to control what page was viewable. The NavigationPage class was used to represent the main page from which many things could be accessed, the GraphPage was used to represent the page displaying the histograms and the HeaderFrame was used for the top half of each graph page.

3.2.1 Controller

3.2.2 NavigationPage

3.2.3 GraphPage

3.2.4 HeaderFrame

3.3 GUI Design

3.4 Coding Style

3.5 Command Line Management

Chapter 4

User Guide

Chapter 5

Developer Guide

Chapter 6

Testing

Chapter 7

Conclusions