DocuTrace

F20FC: Industrial Programming
Coursework 2

Sam Fay-Hunt — sf52@hw.ac.uk

December 4, 2020

Contents

The report should have between 10–15 pages and use the following format (if you need space for additionalscreenshots, put them into an appendix, not counting against the page limit, but don't rely on the screenshotsin your discussion)

1 Introduction

State the purpose of the report, your remit and any assumptions you have made duringthe development process. The DocuTrace application is a moderate size, data-intensive application, its purpose is to analyse and display document tracking data from the website issuu.com. The website hosts a substantial number of documents, and provides anonymised usage statistics, the data is provided in the form of a sequence of individual JSON entries seperated by new lines. It is assumed the users of an application like DocuTrace would be someone with enough degree of technical competency to use simple Linux command line applications, the user would likely be a researcher (data science), or a business. The prior assumption leads to the assumption that the hardware running this application would be closer to server class than standard consumer hardware with higher CPU core counts and alot more RAM. Due to the potential scale of the data a signifantly large amount of RAM is not mandatory to run this application, but a pool of aproximately 8GB of RAM should be installed on the system when processing 3 million lines otherwise a significant performance penalty may be incurred. DocuTrace was written in Python 3, it is intended to be run on Ubuntu 20.04, and has not been tested on other operating systems. Comprehensive documentation has been generated, a list of dependencies, installation and run instructions are provided, note the recommended entryjoint of the application is via the "docutrace" shell script, this script will automatically configure an environment variable to specify the output directory of graph files generated.

2 Requirements Checklist

Here you should clearly show which requirements you have delivered andwhich you haven't.

3 Design Considerations

Here you should clearly state what you have done to your application tomake it more usable and accessible.

4 User Guide

Use screen shots of the running application along with text descriptions to help youdescribe how to operate the application.

5 Developer Guide

Describe your application design and main areas of code in order to help another developer understand your work and how they might develop it. You may find it useful to supplementthe text with code fragments.

6 Testing

Show the results for testing all cases and prove that the outputs are what are expected.Preferably, use unit testing to test core functionality of the implementation. If certain conditionscause erroneous results or the application to crash then report these honestly.

7 Personal Development

A short discussion on lessons learnt from the feedback given on CW1and a discussion how you integrated this feedback into CW2. Cover both coding and report writing,possibly more (project management, preparing for interview style questions etc). Lessons learnt from the experience of CW1: Started out using test driven development

Feedback from CW1: — code — Less code duplication too much global state limited input validation no custom exceptions not restrictive enough access modifiers

— report — intro: should cover short spec cover goals cover env

dev section: should discuss class dependencies should discuss method interfaces := params/return, assumptions of args

conclusion: should discuss adv lang features

8 Conclusions

Reflect on what you are most proud of in the application and what you'd have likedto have done differently. You should reflect on the produced software, and compare software devel-opment in scripting vs. systems languages. Most proud of: - Concurrency during file reading - Structure of the code, good decoupling front and back end

Do differently: - Leave more time to work on the gui - Use a different library for the gui - subclass the datacollector class to break it into smaller tasks

A References