Task Two: Data analytics platform

Introduction

This assessment is an exercise in applying ideas from artificial intelligence to the problems of planning and decision-making. You must write your application in either Haskell or Clojure and may use any appropriate external libraries to help you, however you must include citations as in-code comments identifying those libraries.

You are to submit your code and an academic poster on which you describe the implementation of your solution. You will be given tuition on writing a poster later in the module.

The task

Acme Clothing are a retailer selling t-shirts. They are expanding and need a new data analytics platform. They would like it to report on their existing data, and provide a real-time dashboard with key business metrics. The essential metrics need to be calculated per day, week and calendar month and compare to last period. These include:

- Turnover
- Profit
- Top 10 biggest selling products (irrespective of variant)
- Current unfulfilled order count
- Minimum, maximum and average time taken to fulfil orders

Your task is to implement the analytics platform. The minimal set of requirements that you must reach is:

- Implement the basic functionality that is outlined above.
- Data is structured appropriately and held in memory. There is no need to connect to an external database. You may use a mocking framework to act as a proxy of a database or (better) design your own in-memory structure.
- Any output is displayed to a terminal.
- You should generate input from unit tests or through similar automation.
- A set of sample data is loaded automatically when the program starts.

More complex functionality that builds on the minimal set includes:

- A GUI dashboard which may be implemented as a Web page or as a desktop application.
- Data stored in an external database.
- Changes made to the data at run-time are reflected accurately on the dashboard.
- Users can generate *ad-hoc* queries.
- Up to three of your own extensions.

Sample data sets are provided via links from Blackboard.

Submission

The deadline for this task is **3pm on Thursday 19th April, 2018**. You will be assessed at walkthroughs in weeks starting 23rd April and 30th April, 2018.

Learning outcomes

This task partially assesses the following module learning outcomes:

- Implement a complex application in a functional language
- Evaluate the merits of functional programming

Marking scheme and criteria

This work will be marked using a grade-based scheme. Details of the scheme can be found by following the link on the module's Blackboard site.

Minimal functionality

Criterion	Marks	We are looking for solutions that include
Implement the minimal set of functionality	15	- Code implements appropriate domain functionality - Understanding of the domain is demonstrated within the code - The business logic is implemented in a sensible way
Define and implement a suitable in-memory data structure	10	- Define an appropriate data structure - The data structure is populated using appropriate code - The chosen data structure may be extensible
Architecture	10	- The application has a clear separation between client and server functions - Code is structured using modules etc. that are meaningful - Libraries, including 3rd part ones, are used to support the development of the application
Automation, tooling and testing	10	- The generation of data is automated - External systems such as database are mocked - You use automation to drive and test the application - You demonstrate the use of testing
Clarity and quality of the code	5	- Code is readable with meaningful names given to functions, data items etc. - Follows the appropriate idioms of the chosen language

Further functionality

Criterion	Marks	We are looking for solutions that include
Dashboard	10	- A simple dashboard displays the state of the data - The dashboard updates as data changes - The dashboard accurately reflects the state of the system
Database	5	- A simple database that may be either relational or use a NoSQL solution - CRUD operations are implemented successfully across data types
Ad-hoc querying	4	- Queries can be created and executed - There is some validation of the queries
Student's own extensions	6	- You implement extensions that may be useful, interesting or demonstrate your understanding of your chosen language - There will be a maximum of two marks per extension

The Poster

Criterion	Marks	We are looking for posters that include
Structure, design and layout of the poster	5	- The layout is clear - Good use of colours and typography - Images, where used are meaningful
Comparison of Clojure and Haskell	5	- Understanding of the two languages is demonstrated - The relevant merits of each are shown - Difficulties using each are given - Criteria are set and provide a reasonable basis for a comparison - Conclusions are based on firm data
Discussion of the data structures and algorithms algorithm	10	- The data structures are described - Algorithms are presented clearly and concisely - Code samples are used throughout - The discussion is technically advanced - The code reasonably reflects the core concepts of the domain

Description of the approach to testing and evaluating the application	5	- The testing regime is explained - Testing is relevant to the problem - The system is shown to work in ways that go beyond those specified

Demonstrating the functionality

The poster

You will demonstrate your implementation on an academic poster.

This type of presentation of complex work is common at academic conferences where it is used to convey information quickly and concisely. The posters will be displayed at a poster fair at which they will be marked by the module tutors and your fellow students. Expect to answer *ad hoc* questions about your work.

- Describe how your implementation works.
  - What are the key data structures and concepts, and how are they reflected in your code.
  - What algorithms have you used/adapted for routing.
- How did you test your code.
- Appropriate code samples and screenshots should be used. Include these sparingly.
- You must include a discussion of an alternative functional (or hybrid) language.
  - How might the system have been implemented in that language?
  - How does it differ from your chosen language for this problem?
- Your poster should be A1 size and contain no more than 1,000 words of text. Your work should be fully referenced using the APA style.
- The layout should be more visually interesting than six continuous pages of text. You can find many good examples by doing an image search for "academic poster example". Use these as inspiration.

The walkthrough

At the walkthrough you must demonstrate or discuss the following aspects of your work:

< SCRIPT HERE >