The Design and Build of a Simple Personal Finance System, Focused on Budgeting and Expenditure Analysis

Claudius de Moura Brasil
BSc Computing Project Report
Birkbeck College, University of London

May 2018

1 Abstract

Personal finance systems exist in abundance nowadays, from open source to proprietary ones. They all tend to revolve around a basic common theme: providing accurate information about an individual's income and expenditure. Beyond this, they tend to vary in which features are implemented. The system designed and built for this project focuses on the use of the bookkeeping principle of double entry and the concept of pattern matching to find effective ways to cagetorise a user's expenditure, and provide them with relevant financial information to assist in decision making.

Contents

2 Introduction

Vaasen et al. (2009 cited Boczko, 2012, p. 8) suggests that an accounting information system's main purpose is to provide information to internal and external stakeholders. Although this refers to accounting systems for business, it could be argued that the same concept could be applied for personal finance systems – except that, in this case, the main stakeholder would be the individual using the system for their personal finances (that is, the user). In fact, one of the most widely known accounting systems available in the market, Quicken™, was conceived around the idea that there should be a more efficient and less tedious way to organise one's personal financial information than doing it manually (Quicken Inc., 2017). This project has been developed based on these ideas.

Fowler (1997, Chapter 1, Section 1.3) defines a pattern as "an idea that has been useful in one context and will probably be useful in others". This project will therefore attempt to utilise patterns where appropriate in order to prove this concept, and as an attempt to make use of the experience already acquired in the domain (or domains) in question.

3 Development Method

For this project, an approach similar to that adopted by Bennett et al. (2010, p. 77) regarding methodology will be employed, where no specific named methodology is espoused, but concepts of object-oriented analysis and design shall be applied, in an iterative and incremental fashion, using UML.

3.1 Universal Modelling Language (UML)

UML is a modelling language created with the intention of providing system architects, software engineers and developers with a common set of modelling tools with a defined syntax which would help them better analyse, design and implement software-based systems, and to model business and similar processes (OMG, 2015, p. 43). It defines several constructs which will be employed throughout this report in order to model the specifications of the system, such as:

- Use Case diagrams;
- Activity diagrams;
- Class diagrams;
- Sequencec diagrams;

4 Requirements

Benett et al. (2010,p. 140-142) categorises requirements as of three types:

Functional Requirements The system's functionality – what it is expected to do.

Non-functional Requirements How well the system delivers its functionality. These requirements are related to the performance, scalability, availability, recovery time, security, and others.

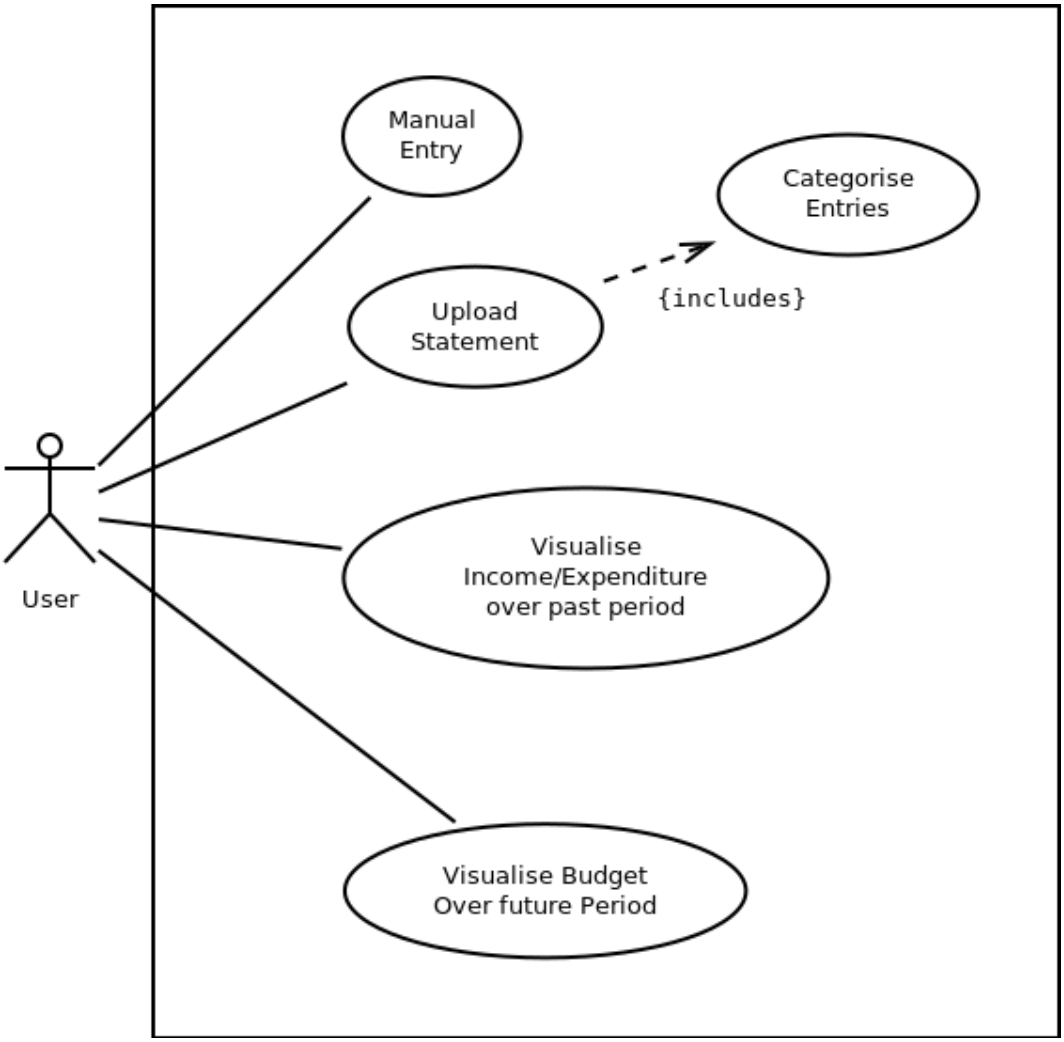
Usability Requirements These relate to how effectively, efficiently and satisfactorily users can achieve their goals in the existing system. User interfaces can play a big part in meeting these requirements.

The initial implementation iterations will be focused more on the functional and usability requirements, paying some attention as well to specific non-functional requirements such as performance and security.

Use case diagrams are an UML construct which were developed by Jacobson et al. (1992, cited Bennett et al., 2010, p. 154). They will be used, along with other tools, to model requirements.

4.1 Functional Requirements

The initial requirements identified for this system are listed below.



References

Bennett, Simon, McRobb, Steve, and Farmer, Ray (2010). *Object-oriented systems analysis and design using UML*. Fourth Edition. McGraw Hill Higher Education.

Boczko, Tony (2012). *Introduction to Accounting Information Systems*. Pearson Custom Publishing.

Fowler, Martin (1997). *Analysis patterns: reusable object models (OBT)*. 1st Edition, Kindle Edition. Addison-Wesley Professional.

OMG (2015). *Unified Modeling Language Specification*. Version 2.5. URL: http://www.omg.org/spec/UML/2.5/PDF (visited on: 16 Dec. 2017).

Quicken Inc. (2017). *About Us*. URL: https://www.quicken.com/about-us (visited on: 17 Dec. 2017).