



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Software Requirements Specification and
Technology Neutral Process Design
Mindmap PIM
Client: IMINISYS

Team: A-Cube-N

Dunkley, Nathan	Grobler, Arno	Lochner, Amy
14145759	14011396	14038600
	Maree, Armand	
	12017800	

Department of Computer Science, University of Pretoria

Contents

1	Introduction	1
2	Vision	1
3	Background	1
3.1	The client's problem	1
3.2	Future business/research opportunities	1
4	Scope	1
5	Software Architecture Overview	1
6	Architecture Requirements	1
6.1	Access Channel Requirements	1
6.2	Quality Requirements	1
6.3	Integration Requirements	1
6.4	Architecture Constraints	2
7	Architecture Design	2
7.1	Architectural components addressing architectural responsibilities	2
7.2	Infrastructure	2
7.3	Tactics	2
8	Database and Persistence	2
9	Process Specification	2
9.1	Server	2
10	Functional Requirements	3
10.1	Use Case/Services Contracts	3
10.2	Required Functionality	3
11	Open Issues	3

1 Introduction

People of this day and age often make use of many technologies and platforms for the purpose of staying in contact with people, sharing moments with friends, communicating with people and organising their day-to-day lives. Generally all the above tasks of a person in the 21st century are done on different platforms for example Facebook, Email and Google Calendar. This project serves to provide the people described above, with a single PIM platform with which they can interact to make use of the functionality from other platforms. A PIM as defined by *TechTerms* is "a software application that serves as a planner, notebook, and address book all in one. It can also include things like a calculator, clock , and photo album."

2 Vision

The vision of this project is to create an PIM which extracts data from various existing platforms such as Facebook, Gmail, Google Calendar etc. The application will make use of various means to extract data, determine the general topic of the data then either construct a new branch in the mind map or add the data as a sub branch in the diagram. The hope is that this will simplify the users life by only needing one application to 'monitor' all other platforms on which they might have an account and manager the information of those platforms from our application.

3 Background

3.1 The client's problem

3.2 Future business/research opportunities

4 Scope

5 Software Architecture Overview

6 Architecture Requirements

6.1 Access Channel Requirements

6.2 Quality Requirements

6.3 Integration Requirements

We will be integrating our application with

- Facebook

- LinkedIn
- Gmail
- Google Calendar
- Google Notes
- Hangouts

6.4 Architecture Constraints

Technical constraints include:

- **Programming language** We will be developing the system in Java.
- **Operating system** The server will be run off a Linux machine.

Further there are no constraints and the architects have free roam to implement an architecture that suits their needs.

7 Architecture Design

7.1 Architectural components addressing architectural responsibilities

7.2 Infrastructure

7.3 Tactics

8 Database and Persistence

9 Process Specification

9.1 Server

In the back end, the system would have a polling service that polls the social media sites for new information about the users. If a polling thread receives new data from the platform they add this new data to a queue to be processed. Several worker threads dequeue data from the queue and uses the natural language processing to determine the topic(s) of the data. This data is then added to a database and the corresponding topic(s) as well. Note that since hashtags are used so frequently, any hashtag in the text should be added as a topic of its own. When a user requests their mind map, the system responds with all relevant topics of their life based on frequency and recency. These topics are then returned to the user in the form of JSON objects. Once a user expands one of the nodes a request is sent to the server to retrieve all the relevant articles corresponding to the topic to the user. These new data is then added to the

mind map in the form of nodes. Relevant articles are found on the basis of your own events and the events of your friends.

9.2 Front End

The user starts out with a root node that has relevant topics connected to it including a friends node. All these nodes can be expanded and reveal information related to that topic. The friends node shows contacts you recently communicated with or might want to communicate with now. Expanding one of these contacts will show the different platforms you can interact with this contact, including viewing a profile and sending a message/email. Other topics will expand into relevant articles and those can be expanded again and so forth. Upon expanding an article a user can then interact with that article (comment/reply) in a side panel that will be presented on the right of the screen.

10 Functional Requirements

10.1 Use Case/Services Contracts

10.2 Required Functionality

Functionality that should be included in the system includes:

- A login/sign up system.
- Adding various PIMs to your account.
- Interaction with PIMs directly from the mind map itself (commenting, emailing, etc).
- Expanding bubbles in order to find articles related to that bubble.
- Rapidly integrate new information into the system.

11 Open Issues