
Functional Requirements

Project: Insurance profiling from social media
Client: RetroRabbit

Team: Valknut Solutions

- 13054903 - Charl Jansen van Vuuren
- 10297902 - Bernhard Schuld
- 13044924 - Kevin Heritage
- - Quinton Weenink

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF PRETORIA

DATE

Contents

1	Introduction	3
2	Vision	3
3	Scope of system	3
4	Architectural requirements	3
4.1	Access channel requirements	3
4.2	Quality requirements	3
4.2.1	Performance	3
4.2.2	Security	3
4.2.3	Scalability	3
4.2.4	Integrability	4
4.2.5	Reliability	4
4.2.6	Flexibility	4
4.2.7	Maintainability	4
4.2.8	Auditability	4
4.2.9	Cost	4
4.2.10	Usability	4
4.3	Integration requirements	4
4.4	Architecture Constraints	4
5	Functional Requirements	4
5.1	Use case name	4
5.1.1	Use case prioritization	4
5.1.2	Use case service contracts	4
5.1.3	Process specification	4
5.2	Domain model	4
5.3	Open Issues	4

List of Figures

1 Introduction

This document contains information on the development of a system to generate insurance profiles based on social media inputs. The project is being developed for Retrorabbit as part of the COS301 module at the University Of Pretoria.

2 Vision

The primary focus of this project is to generate quick, reliable insurance/risk profiles from user's social media information. The profile generated will be ideally used for portable possession insurance (cellphones, laptops, purses). Insurance profiling and risk analysis often require large amounts of data to generate in-depth profiles, our project seek the means to eliminate the need for vast amounts of data gathering by using a user's social media information. A user would log into our system, provide the necessary permissions and our engine will generate a risk profile for that person based on certain criteria. The generated profile can assist insurers to create more accurate risk profiles or clients to get personally tailored quotes, instantly.

3 Scope of system

The system will primarily be web-based. A user visiting the website will request a quote and be prompted to log into Facebook. A service will then acquire data from the users facebook profile. This data will be fed into the profiling engine, which will process the data to create a risk profile. This risk profile will be used to generate a quote, which will then be displayed to the user on the website.

4 Architectural requirements

4.1 Access channel requirements

4.2 Quality requirements

–(CLIENT'S)

4.2.1 Performance

The user should be able to request a quote in less than 5 minutes, this is easily achieved by means of the Facebook login and will only be user-network dependent. Once the REST api call is made it saves the users data in our database, increasing the efficiency of future requests and processing of the data.

(Save only temporary?)

4.2.2 Security

Security is our most important architectural requirement. User's personal information is used to generate these risk profiles and as a result the user will trust that this information is not shared with other parties. Only authorized persons will have access to the generated profile and access to the database will need to be restricted to the highest authority.

4.2.3 Scalability

Since this project is a web-solution the possibility of multiple concurrent users should be considered. The server should account for a vast amount of concurrent users.

4.2.4 Integrability

The project will integrate with Facebook mainly as the primary data provider. A Facebook approved login button on our website will allow access to the information requested. The ability to integrate with other social media should be considered and modularized accordingly as to ensure future integration of other social media platforms. Further integration includes the connection from the website to the profiling engine and back to the website as a report.

–(Template recommends these further)

4.2.5 Reliability

4.2.6 Flexibility

4.2.7 Maintainability

4.2.8 Auditability

4.2.9 Cost

4.2.10 Usability

4.3 Integration requirements

4.4 Architecture Constraints

The client limited our development stack technologies to:

- An ASP.net web solution with a Microsoft SQL Server database system.
- A NodeJS web solution with a PostgreSQL database system.

Other constraints in terms of architecture include:

- Browser independent as to ensure any web-client can make use thereof.
- Operating system independent.
- The system should be as time efficient as possible to ensure a user gets a quote in less than 5 minutes – (can state this in quality requirements -Performance, as per proposal pdf)
- No PHP code as per client's request.

5 Functional Requirements

5.1 Use case name

5.1.1 Use case prioritization

5.1.2 Use case service contracts

5.1.3 Process specification

5.2 Domain model

5.3 Open Issues