Non-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

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.

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.

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.

Architectural requirements

Access channel requirements

Quality requirements

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.

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.

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.
- As per integrability, The ability to change the risk analysis algorithm as needed should be considered to a future scalable solution.

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.
- The ability to change the risk analysis algorithm as needed should be considered.
- Integration includes the connection from the website to the profiling engine and back to the website as a report.

Reliability

Since our platform is mainly web-based the platform as a service (PaaS) offered by Heroku hosting will ensure the website is always up to date and reliable.

Maintainability

The system will make use of a database with massive amounts of data, to ensure optimal performance this data will need to be maintained and normalised on a regular basis.

Auditability

- All actions performed in the system should be traceable to the user that performed them.
- The user's IP will be logged as to have a form of accountability in the persistence of the data.

Cost

The majority of our platform is open-source except for:

- Heroku hosting (as mentioned in constraints)
- Travis CI (Commercial Version)

Usability

- The platform is being developed with efficiency in mind, as a result the input and response of the website should be visually pleasing and simple to use.
- The Facebook login API aims to improve efficiency for the user as this eliminates the need to fill various fields in a form for example.

Integration requirements

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 independence 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.

Our client specifically constrained the use of:

- PHP code in any way.
- MySQL, NoSQL database systems.
- The use of a platform as a services (PaaS) hosting solution Heroku.

Functional Requirements

Use case name

Use case prioritization

Use case service contracts

Process specification

Domain model

Open Issues