



Department of Computer Science
University of Pretoria

Software Engineering
COS301 Main Project

Team CodeX

ReRoute Systems

Bondjobo, Jocelyn 13232852

Malangu, Daniel 13315120

Kirker, Tim 11152402

Hammond, Eunice 13222563

Burgers, Heinrich 15059538

Contents

1	Introduction	1
1.1	Background	1
1.2	Purpose	1
1.3	Scope	1
1.4	Definitions, Acronyms, and Abbreviations	1
1.5	References	2
1.6	Overview	2
2	Overall Description	3
2.1	Product Perspective	3
2.1.1	System Interface	3
2.1.2	User Interface	3
2.1.3	Hardware Interface	3
2.1.4	Software Interface	4
2.1.5	Communications Interface	4
2.1.6	Memory	4
2.1.7	Operations	4
2.1.8	Sit Adaption Requirements	4
2.2	Product Functions	4
2.3	User Characteristics	4
2.4	Constraints	4
2.5	Assumptions and Dependencies	4
3	Specific Requirements	5
3.1	External Interface Requirements	5
3.1.1	Software Interfaces	5
3.2	Requirements	5
3.2.1	Functional Requirements	5
3.2.2	Non-Functional Requirements	5
3.2.3	Use Case Prioritization	6
3.2.4	User Interfaces	6
3.3	Performance Requirements	6
3.4	Design Constraints	7
3.5	Software System Attributes	7
3.6	Other Requirements	7
3.6.1	Quality requirements	7
4	Open Issues	8
4.1	GitHub Repository	8

1 Introduction

1.1 Background

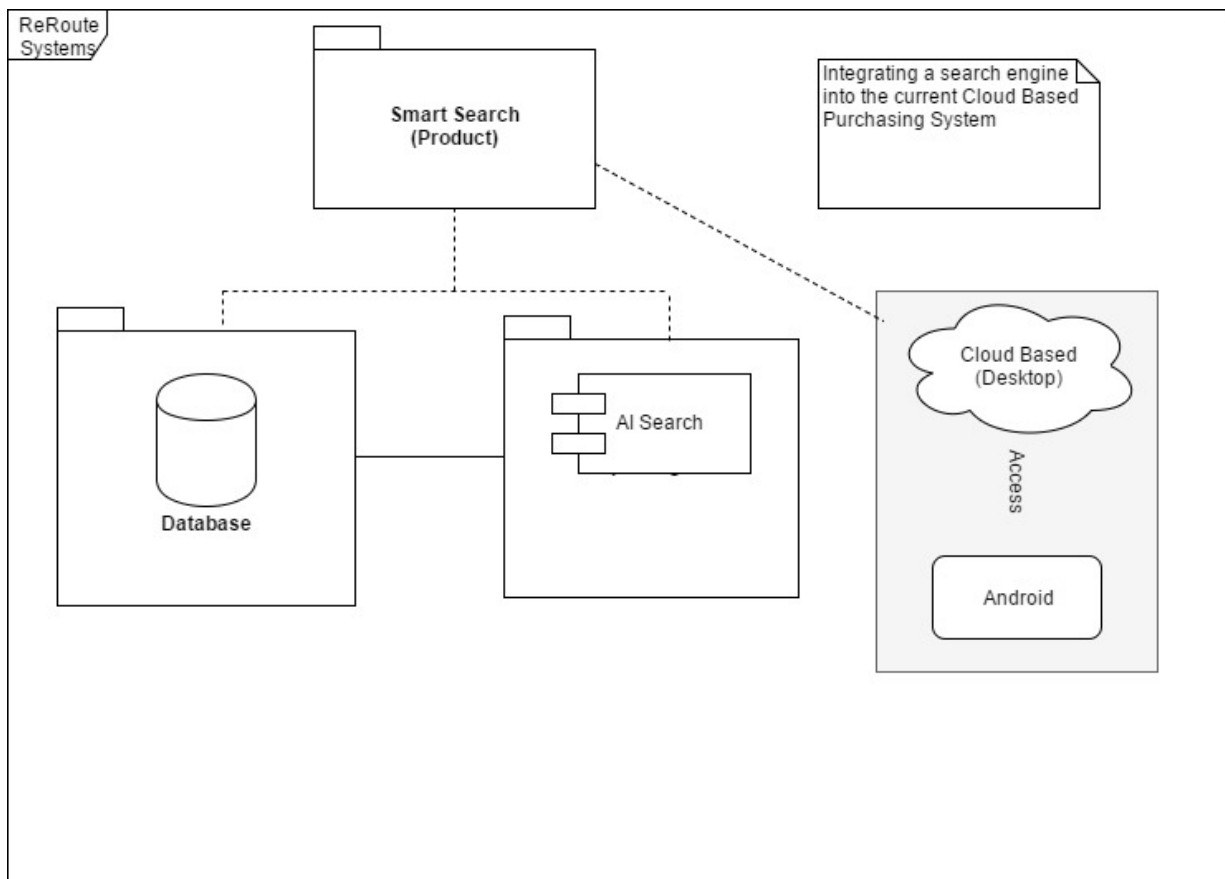
Reroute Systems is a software company with different in-house developed applications. The Purchase Management System application is the main application and mainly active in the pharmaceutical space. The main functionality of the system is routing of request for products from account holders to the various wholesalers / suppliers, receiving the result of the order and route answer back to account holder.

1.2 Purpose

For the account holder to request a product he/she must search for it against the database with all the product information, the challenge is that each wholesalers / suppliers can name/describe the product differently and when the account holder do the search against the master product list the same product must be displayed across all wholesalers / suppliers using the link between master product list and the different wholesalers / suppliers product list (the user compare the prices of same product across the wholesalers / suppliers before making a decision)

1.3 Scope

The core of the system is a search engine which is enriched with functionality of performing some machine learning in order to retrieve the product in a master file found in a Databased running on the server through http calls. The high level modules and their responsibilities are shown in the figure below.



1.4 Definitions, Acronyms, and Abbreviations

- Cloud-based : refers to applications, services or resources made available to users on demand via the Internet from a cloud computing provider's servers.
- Android : Mobile Operating System that allows the mobile phone to operate and install different apps in it.

- AI Search : The use of Artificial Intelligence to search the database by implementing machine learning techniques and other algorithms to find and retrieve data

1.5 References

To be completed

1.6 Overview

The Purchase Management System application is the main application and mainly active in the pharmaceutical space and called **Smart Rep** that can be found on Google Play store. It allows to place orders of product through mobile devices such as Android and also through Desktop as it is a cloud based system. This system monitors the routing of orders from devices to the server to different wholesalers and bring back the result.

2 Overall Description

2.1 Product Perspective

2.1.1 System Interface

1. User Interface:

- **Functionality:**
The functionality of the user interface is to allow the user to interact with the system. Through this aspect of the system the user can perform all the functions necessary to use the smart search of the system. Through this the user is able to specify the search criteria to be used by the system.
- **How functionality achieves requirements:**
The user interface allows the functionality of the application to meet the requirements. The user is able to search for any product based on any value they enter int. The user is able to be displayed information of products, that closely based on keywords and values or any thing resembling the name.

2. Hardware Interface:

- **Functionality:**
The functionality of this system interface is the physical hardware that allows software operation. The hardware in this case is specifically the mobile devices on which the apps run and the computers the websites usually accessed through.
- **How functionality achieves requirements :**
This hardware will allow the achievement of the requirements by facilitating the operation of the application. The user will be able to press the physical device screen in order to use the search capabilities and other features of the application. The networking hardware of the components such as the wireless fidelity infrastructure or LAN connections within the devices which will allow for communication to the system server.

3. Software Interface:

- **Functionality:**
The functionality of the software interface is to facilitate the communication between the hardware infrastructure of the devices running the application front-end and Reroute Server. An example of this would be the operating system of the device in use. The operating system would allow the application to make use of the networking infrastructure and communicate the necessary information to the application via RESTful calls. This would enable the device to communicate with the server and perform the necessary requests.
- **How functionality achieves requirements :**
This would allow the achievement of the requirements as the communication of the application to the system server is vital to the operation of the Smart Search. By allowing communication over WI-Fi the user is able to communicate to the server from any location or just stick to older and more familiar devices such as computers, fulfilling the requirement .

2.1.2 User Interface

All users will be able use to the Smart Search system from their device when the application is launched. For the account user interface, the registered userswill be able to login to their account, using their account information.

2.1.3 Hardware Interface

The actual Wi-Fi interface between the mobile device / computer and Wi-Fi will be abstracted by the phone and will be managed by the actual underlying operating system on the phone. On a computer the LAN interface may also be abstracted by the computer and will be managed by the actual underlying operating system on the computer if a wired connection is to be used.

2.1.4 Software Interface

to be completed

2.1.5 Communications Interface

- Users using the web will use HTTP/HTTPS protocol.
- REST calls from the different platform to the server API.
- Cloud-based server running the database

2.1.6 Memory

The desktop application will use a minimum amount of space of roughly 300MB. the size could dramatically increase due to caching and storage of data locally. The size will depend solely on caching of a lot of data. Primary memory(RAM) will use 50MB on average, based on how long the application is being used.

2.1.7 Operations

The user will require a series of usual and special case operations to be fulfilled by the system. The user will use the system in normal case operations that comprise of searching on any of the different properties of a product. On a usual case the user will search for a product based on the name of the product of interest .

The system will provide a user the option to either save there data, such as variations of product names, locally on the device or remotely on the server.

2.1.8 Sit Adaption Requirements

to be completed

2.2 Product Functions

The main function of the Smart Search will be to retrieve product iformation based on the values given to the user. The basic functions will include:

- A search based on a specific property of a product such as the name

2.3 User Characteristics

The general user will use the system to retrieve product information. They will need minimal skill in order to utilise the application, including the ability to connect to the Wi-Fi and use . The general user will have restricted usage of the in-app game and certain privileges that require a user profile. They will not require any additional expertise or education to function the application.

2.4 Constraints

to be completed

2.5 Assumptions and Dependencies

to be completed

3 Specific Requirements

This section gives a detailed description of the system requirements. It describes all the functional as well as the quality requirements of the system.

3.1 External Interface Requirements

3.1.1 Software Interfaces

The application will run on the Android operating system, specifically version 4.0. and upwards. It will also run on iOS operating system version 7.3 and above.

3.2 Requirements

3.2.1 Functional Requirements

1. Fuzzy Search:

This program must be able to locate and return products that are related to the searched term. It should be able to return the same product if various ways of writing the name is used.

2. Fast Run-time:

Because there are very large amount of data to be searched, it is vital for our program to be very efficient. Even though this could be a non-functional requirement, it is vital for this program to be as fast and efficient as possible.

3. Handle Concurrent Users:

This program will be used by many people, in many different fields simultaneously. It is therefore very important for it to be able to handle multiple users at the same time.

4. Machine Learning:

In order for this program to run efficiently it must be able to adapt to the needs and habits of its users. It should be able to learn search patterns and learn from previous searches.

3.2.2 Non-Functional Requirements

1. Predictive Typing:

Predictive typing is when the program suggests a possible solution as the user is typing. This is often based on previous searches and most common searches. This will help improve the user experience.

2. Return Generic Medicine:

In some cases it will be handy if the program suggested the different generic medicine for the medicine you searched for. This could help users find the product they need even if they are unsure of the name.

3. Robustness:

This product should be able to adapt to different users in various fields as well as to different habits, interfaces, environments and needs of users.

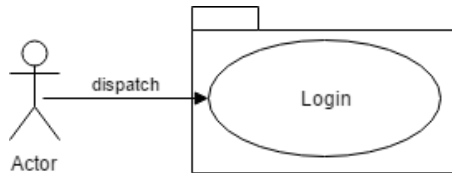
4. Storing and using search history: Being able to store the search history will increase the user experience, help with predictive typing and improve the programs ability to learn and adapt.

3.2.3 Use Case Prioritization

1. Critical

- Login

3.2.4 User Interfaces



- Retrieve Wholesalers Accounts
- Search Product

Pre-condition: The user has to have been logged in successfully.

Post-condition: System returns possible matches, or search not found.

Actor: User	System: Purchase Management System Application Searching Product
-	0. TUCBW System displays search bar
1. User clicks search bar, and enters product name	2. System retrievees profile information and displays it
-	4. System returns and displays all matches found (name variants inclusive) as well as generics for product name, arranged according to prices and suppliers.
5. User selects appropriate match, or changes search conditions	-
7. TUCEW user makes decision by selecting preferred choice of product.	-

- Logout

2. Important

3. **Nice to have** System using a search history, predictive typing, returning generic medicines.

3.3 Performance Requirements

Performance Requirements bullet list:

The performance requirements of the Rerouting Purchasing System can be sub-divided into two components, which are the following: Client Application

1. The Software needs to be responsive and lightweight enough to run on both mobile hardware as well on desktop (i.e. web).
2. During User search the Smart Searching system will need to perform in real time showing results as necessary.
3. The data usage should be kept to a minimum as to not overload the whole system's Wi-Fi infrastructure being used for non-searching purposes.

Server Performance Requirements

1. The system should be able to handle multiple users using the smart seart.
2. The system has to be able run at a high speed, returning the results of a search immediately.
3. The system should be able to handle a capacity of requests made to the database to make efficient searches and matches.
4. The system should be able to return matches of a high probility of available data in the database on the cloud.

3.4 Design Constraints

3.5 Software System Attributes

The software system will have the following attributes:

Availability: The system and its functions will be available to any registered user as long as there is an Internet connection.

Reliability: The system will provide results that are available on the database to match whatever a user performs a search/smart search.

Portability: The system will be able to function across a multitude of interfaces. Therefore, it will be able to port between Android and iOS with ease, as well as on ordinary web.

Maintainability: The application will be able to be extended easily with new functionality. There will also be a good test environment to reduce and make it easy to find system errors.

Robustness: The system will be able to stand against errors made by users in the search, returning no results and error messages as a result.

3.6 Other Requirements

3.6.1 Quality requirements

4 Open Issues

4.1 GitHub Repository



Team CodeX Repository: <https://github.com/josephbondjobo/CodeX>

This repository contains:

- All work done by team members.