UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and Information Technology

COMPUTER SCIENCE

Testing Report
Team CodeX

Burgers, Heinrich	15059538
Bondjobo, Jocelyn	13232852
Kirker, Tim	11152402
Hammond, Eunice	13222563
Malangu, Daniel	13315120

Contents

1 Functional Requirements

1.1 Core Functionality

Our tests for non-functional requirements can be found in Table 1.

1.1.1 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.
The two use cases tested was sending a single email and sending a batch of emails.

Test Case:

- Null Value SOAP Search: 10
- SOAP Serach: 10

1.2 Innovations

1.2.1 Fast Run-time:

Because there are very large amounts of data to be searched, it is vital for the 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.

Test Case:

- TBD

1.2.2 Machine Learning:

The ability to learn from a user's previous interaction and adapt to it will improve the systems capability and ability greatly. This can be done by using the dictionary provided by the ZIZO database, for creating new links between products that appear seemingly unrelated.

Test Case:

- TBD

1.2.3 Handle Concurrent Users:

This program will be used by many people, in many different

elds simultaneously. It is therefore very important for it to be able to handle multiple users at the same time. This will be achieved by the Cloud based architecture. Test Case:

- TBD

1.2.4 Scalability:

Since it is essential for this system to be efficient with large datasets as well as small datasets, scalability will be an essential functional requirement. The systems ability to stay efficient with very large sets of data is an essential part to the system.

Test Case:

- TBD

2 Non-Functional Requirements

2.1 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.

Score: 0

2.2 Spell checker:

The program will check the spelling as the user is typing a product name to enhance the search.

Score: 0

2.3 Roubstness:

The system will not be easily breakable as it will be error proof to a feasible extent, to avoid unnecessary fault and not wholly be affected by the hardware failures. The system should be able to recover quickly from such failures, or at least be able to hold up or return to a valid stage/state.

Score: 0

2.4 Reliability:

This product should be able to adapt to different users in various fi

elds as well as to different habits, interfaces, environments and needs of users.

Score: 0

2.5 Storing and using search history:

Being able to store the search history will increase the user experience, help with predictive typing and improve the program's ability to learn and adapt.

Score: 0

Table 1: Test Details
Test Fuzzy Search:

Test	Input	Expected Result	Actual Result	Comment	Score	Weight
SOAP Search	Product Name: Panado	27	27		10	0.5
Null SOAP Search	Product Name: NULL	0	0		8	8

Table 2: Non-Functonal Requirements Tests

Test Performance