Table of Contents

1. Introduction

1.1 Overview

This is a **mobile interface** for the existing DCU **Machine Translation (MT)** System, the **MaTrEx** system. The interface is portable as it's developed as a **web application**. Thus it works on the majority of modern mobile devices. The mobile market is ever increasing, thus tapping it is essential for future expansion. This application hopes to provide users across all platforms with a translation app fulfilling the needs others don't. It's hoped the extra functionalities will help it become a leader in the market.

With this interface, like other translation services; the user can input a sentence or a paragraph in a language and have it translated into a desired language. Despite the high quality of the system, some translations may need further **post-editing** by human translators in order to achieve publishable quality. The interface also offers the functionality for the user to post-edit the MT translations, a possibility not often offered to users.

Not only will post-editing improve the output for the user, it also has the potential to improve the MaTrEx system for future translations. As a further addition all translations, edits and specific edit details are stored in a database by this interface. **CNGL** can make future use of this information to pinpoint errors in the translations.

Whether or not the user post-edits, they are left with several end choices for dealing with the output. The user can simply copy the text and use it elsewhere, which is supported by most devices. Or they can choose between emailing on the output, or searching for it with a search engine chosen in the application settings.

Users also have the option of choosing the subject of their inputted text from a supported list. This is a feature which truly makes this system stand out among the others. Machine translation services are infamous for their inaccuracy, and though they may be improving, this is one way of guaranteeing a more correct output.

Simply put this is a more complete translation application. It gives the user more control over their results without becoming overly complex. It's one more step towards MT perfection.

1.2 Glossary

Mobile Interface: This is a graphical user interface built specifically for mobile devices.

Machine Translation (MT): This is the application of computers to the task of translating texts from one natural language to another.

Matrex: A free/open-source system for Example Based Machine Translation developed by CNGL.

CNGL: The Center for Next Generation Localisation - A dynamic Academia-Industry partnership with over 100 researchers funded by Science Foundation Ireland, who are developing novel technologies addressing the key localisation challenges of volume, access and personalisation.

Post-editing: The process of manually editing and modifying text after it has been compiled or translated by a machine.

Web Application: A website that behaves like software.

SQL: A simple, commonly used, standard database programming language that is only used to create queries to retrieve data from a database.

Remote procedure call (RPC): This is an Inter-process communication technology that allows a computer program to cause a subroutine or procedure to execute on a another machine.

Language pairs: This is a term used to denote the source language and the target language involved in text translation.

URI: This is a IP or website address with parameters appended to the end.

eg. A user searches for Will Smith on Youtube -

http://www.youtube.com/results?search_query=will+smith&aq=f

Encoding/Decoding: Encoding is the process of putting a sequence of characters (letters, numbers, punctuation, and certain symbols) into a specialized format for efficient transmission or storage. Decoding is the opposite process.

UTF-8: This is a variable-length character encoding for Unicode. It is able to represent any character in the Unicode standard.

Unicode: International encoding standard that provides a superset of many separate encodings.

Base-64: This is used as a generic term for any encoding scheme that encodes binary data by treating it numerically and translating it into a base 64 representation.

Textarea: The <textarea> tag defines a multi-line text input control used in HTML pages.

Stylesheet: Stylesheets are used to control the style and layout of multiple Web pages all at once.

Viewport: The rectangular area that determines how content is laid out and where text wraps on the webpage.

IP Address: A unique number that identifies the precise location of a particular node on the Internet. The address is a 32-bit number usually written in dotted decimal format.

2. System Architecture

Below is the system architecture of the application. Each component is illustrated and its operation explained in the following sections.

2.1 System Architecture Diagram

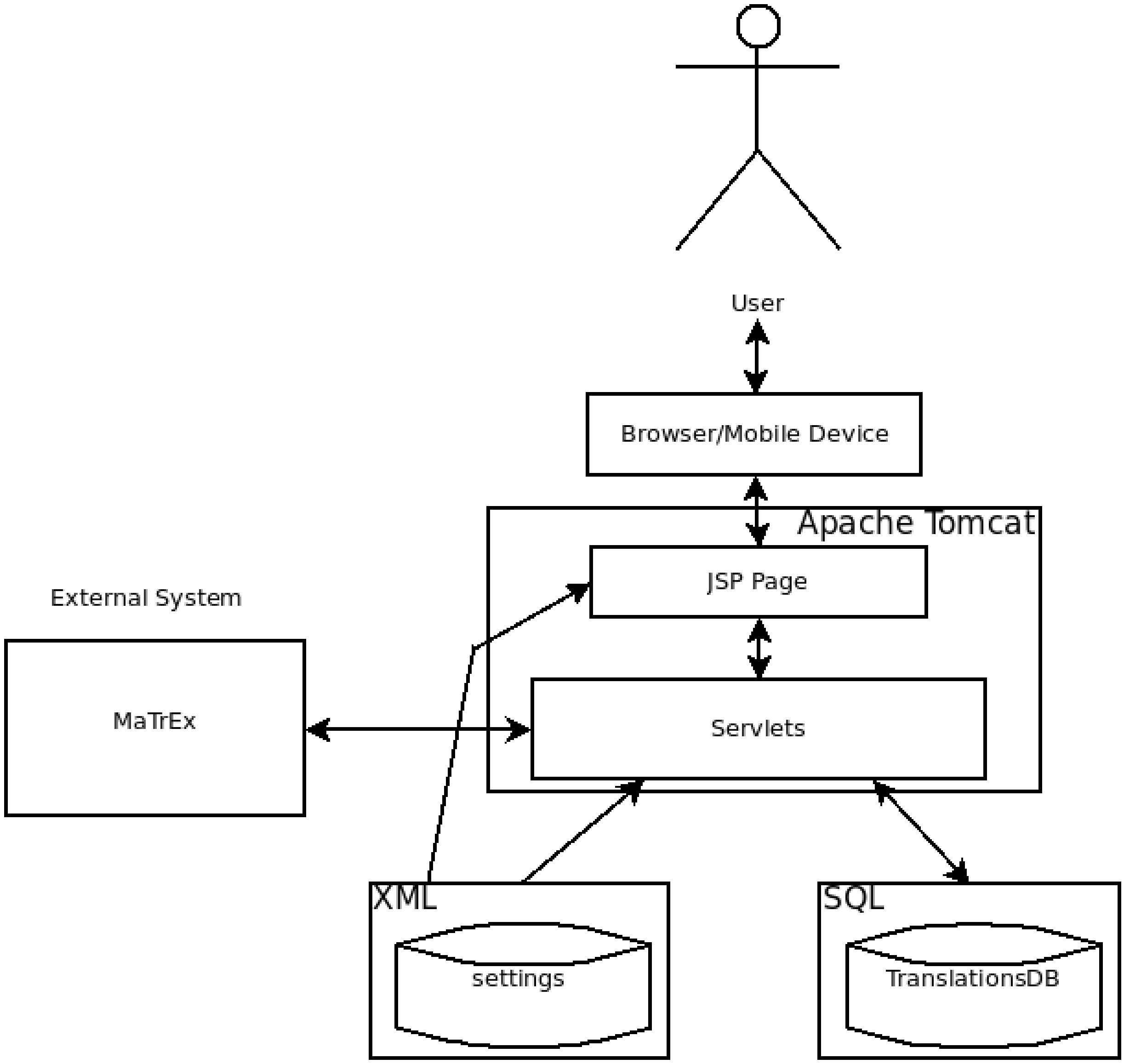

Fig 2.1

Fig 2.1 Depicts the architecture of the system in place. It shows clearly all the categories it can separated into.

2.2 Apache Tomcat

This is the application server. It holds the servlets and dynamic web pages which perform the business logic of the application. It is the center of all communications.

2.3 Servlets

Each servlet fulfills a specific role in the running of the application. The MatrexServlet looks after the translations. The PostEditsServlet takes care of the post-editing functionalities, and the TrackingServlet tracks the precise details of the post-editing.

2.4 JSP page

The JSP page calls on the servlets to keep it dynamically updated. This is what the user sees in their mobile device or browser.

2.5 MaTrEx

This element of the architecture represents the existing DCU MT system. It allows the user to submit text for translation. The text along with the source language, the target language, and the domain are passed as parameters to the MatrexServlet, which in turn makes a **remote procedure call** to MaTrEx at the IP set up for this translation. Upon completion it returns an encoded message to the server, which is decoded and passed to the webpage with the result of the translation. All information from this translation will be stored on the server.

2.6 SQL Database

The application also deals with post-editing. All edits must be stored in a database for future use. All translations (input, source language, target language, domain etc.) are stored along with any post-edits and specific information of the editing (end results of edits, time spent editing, each step of the editing).

2.7 XML Database

The application supports flexible addition of **language pairs** and other miscellaneous settings information. These are stored in an xml file, which both the servlet and the jsp use to ensure consistency.

3. High-Level Design

This is a high-level design of the users interaction with the interface. It displays all the user options along with the flow of the program. These are described in greater detail underneath.

3.1 High-Level Design Diagram

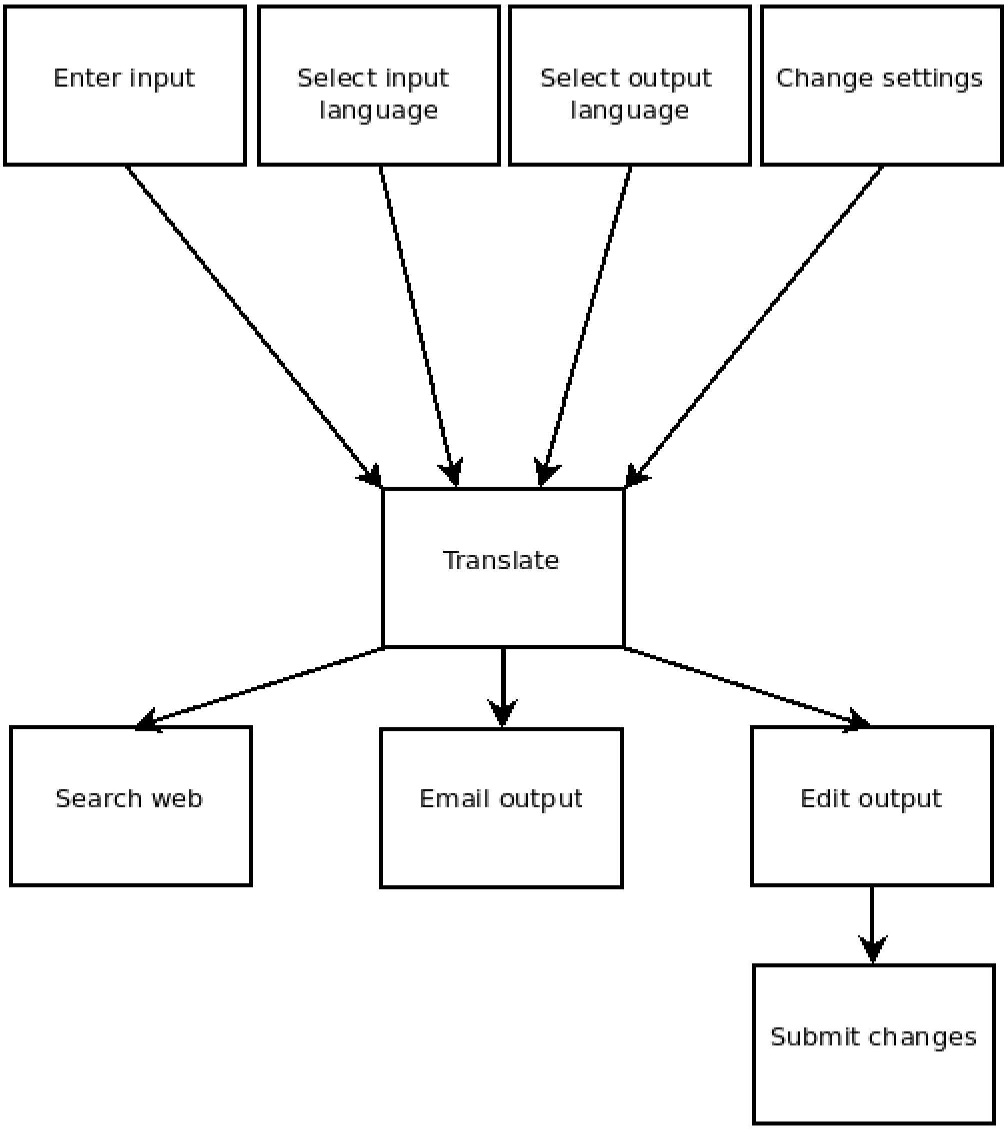

Fig 5.1

3.2 High-Level Design Description

Fig 3.1

- **Enter input :**

  Enter the text for translation in the input box. This can be typed, or pasted.

- **Select input language :**

  Choose the language of the input text from a list of supported languages if the default value is incorrect.

- **Select output language :**

  Choose the language of the output text from a list of supported languages if the default value is incorrect.

- **Change settings :**

  If the user chooses to they can specify the subject of the input text. This will increase the accuracy of the translation. They can also choose a search engine for the “Search Web” option from those offered.

- **Submit information :**

  After the language selections have been made, and the text inputted, the user can submit the information for translation.

- **Search web :**

  The user can search the web using the output in an offered search engine.

- **Email output :**

  The user can email the translated output. This will open their default mail application with the text in the body of the email.

- **Edit output :**

  If the user chooses to they can edit the output.

- **Submit changes :**

  The user can submit the changes to the server after editing the text.

4. Problems and Resolution

4.1 Encoding and decoding

Encoding and **decoding** text is essential at various stages during the translation phase as well as the post-editing phase. CNGL has a standard method for doing this with the translation service, which is utilised. The problem lay in encoding between the servlet and the web page. Specifically with use of special characters, which many languages use.

It is essential that both are encoded in **UTF-8**, a trivial task. The ordering of the encoding/decoding for receiving and responding is the difficult part, and this was the cause of all such issues encountered. This was solved by means of trial and error.

Example:

mobile device ---***unicode***---> servlet ---*utf-8*---> MatrexClient ---***base64***---> MaTrEx

MaTrEx ---*base64*---> MatrexClient ---*utf-8*---> servlet ---*unicode*---> mobile device

4.2 Satisfying different screen sizes

Satisfying different screen sizes has long been a problem for web designers. It is even more prevalent now with the massive increase in mobile browsing. The problem is making your work presentable on all screens without wasting time and resources making unique solutions for all cases. Dynamic approaches are possible using percentage measures in **stylesheet**s, by specifying that an element the page takes a certain percentage of the screen size; however these don't work well with **textarea**s, which were necessary in this application.

My approach is by no means elegant, but after much research and experimentation with other suggestions, I decided this is the most effective solution. All but the textareas are fixed in size. Using the **viewport** settings of the various devices these fit nicely. To calculate the size of each textarea I simply subtract the measure of all these fixed elements from the screen height, and divide the result appropriately among the two textareas. This has worked perfectly on all devices tested.

Example:

windowSize = 300px;

header = 50px;

choiceSection (language selections and flags) = 100px;

bottomButtons = 60px;

input/output = ((windowSize – (header + choiceSection + bottomButtons))-windowSize/10)/2

= ((300 – (50 + 100 + 60)) – 300/10)/2

= (300 – 210 – 30)/2

= 30

4.3 XML Files

XML files are used for configurations and settings in the application server. It was decided to make use of web.xml (used by the servlets) for the language pairs and other application specific settings. However this could only be accessed by the servlets. Due to security settings in the application server the web page could not view it.

To solve the problem a new xml file was created, which both the servlets and web page could access. In order for the servlet to view it it's pointed to by web.xml. This increases the flexibility of the application, and prevents repetition of information.

4.4 Edit tracking

To utilise post-edits effectively each specific edit is needed. This helps pinpoint the most obvious errors. It is difficult to track the cursor position effectively in a web browser. The idea was contemplated and attempted, but there was also the problem of the action taken at that position.

The chosen approach was the return the text in the output box each time a key is pressed. These are stored in a database sequentially, giving a story line of the edits.

Example:

Output = “Hello le monde”

User believes it should be “Hello monde”, so they make the change and submit it.

Input received by server =

1: “Hello l monde” - User started cursor after “e” of “le” and pressed backspace.

2: “Hello monde” - User again presses backspace.

3: “Hello monde” - User presses backspace for the last time and submits.

4.5 Copying to clipboard

Web apps are excellent for their portability, however they fall down in their capabilities as they are limited by those of the browser. Copying to the clipboard within the application by means of a button was not possible, this is a security feature in place of the OS of each mobile device tested.

As an alternative the user can still select and copy the text manually. This is device dependent, but a user familiar with their device should have no difficulties.

5. Installation Guide

5.1 CNGL

Below is the installation instructions for deployment of the system on any machine. It includes the software versions that must be installed, along with further setup instructions.

5.1.1 Apache Tomcat

This is the application server.

Version: Apache Tomcat 6.0

Installation instructions: *http://tomcat.apache.org/tomcat-6.0-doc/setup.html*

5.1.2 MySQL

This is the database management system. It runs as a server providing multi-user access.

Version: 5.5.9 MySQL Community Server (GPL)

Installation instructions: *http://dev.mysql.com/doc/refman/5.5/en/installing.html*

5.1.3 Java EE

This is the platform used for the server programming.

Version: Java EE 5

Installation instructions:

http://www.oracle.com/technetwork/java/javaee/javaee5sdku8-install-jsp-135084.html

5.1.4 Language/Search engine addition

Once all the necessary software is installed the next step is to add some language pairs and search engines for the user to choose from.

Open settings.xml. Between the <settings> tags you do the following -

<u>Language addition:</u>

Enter a new set of <pair></pair> tags. Within the <pair> tags enter the language pairs information. This includes the <source> tags, within these you enter the source/input language of the model; the <target> tags, into which you enter the target/output language of the model; the <domain> tags, wherein you enter the domain/speciality of the model; and finally the <ip> tags, inside which you enter the **IP address** of the machine on which the model is hosted. A remote procedure call will be made to that address.

<u>Search engine addition:</u>

Enter a new set of <engine></engine> tags. Within the <engine> tags input the search engine information. You must include the <name> tags, within these you enter the name of the site. This is what the user will be presented with. You must also include the **uri** (without any parameters) of the site. The output will be appended to this to create the search link.

5.1.5 Database

The IP address of the translations database can be specified in web.xml. This must be specified for all servlets, as each one connects to it.

There is also the possibility of recreating the database on another machine. To do this first ensure the correct version of MySQL is installed, and run translationsdb.sql. Again the databaseURL must be specified in web.xml.

5.1.6 Deploying application

Once the correct version of Apache Tomcat is installed you may continue.

First ensure Tomcat is running, *http://tomcat.apache.org/tomcat-6.0-doc/RUNNING.txt*

Copy MatrexWeb.war into $CATALINA_BASE/webapps. The application will now be available at *http://localhost:8080/MatrexWeb*.

5.2 Mobile User

To use the application on a mobile device visit www.cngl.ie/MatrexWeb within your device's browser. For the best possible experience add this page to your home screen. This allows the page to act like a native application. To do this try the following -

5.2.1 iDevice

http://www.apple.com/iphone/features/home-screen.html (bottom right)

5.2.2 Android (2.0+)

http://philwilson.org/blog/2010/01/adding-a-bookmark-to-an-android-home-screen

5.2.3 Blackberry (6.0+)

http://docs.blackberry.com/en/developers/deliverables/7117/add_a_web_link_as_an_application_619196_11.jsp

5.2.4 Windows Phone 7

http://techblog.ginktage.com/2011/01/how-to-add-apps-and-webpage-to-the-home-screen-in-windows-phone-7

5.2.5 Other

Using your preferred search engine enter “bookmark to home screen <your phone type>”.

6. Testing

Before testing the application thoroughly feedback was received from individuals within CNGL, friends, and family. Criticism of the presentation and functionality of the application was encouraged. This varied as some users were experienced mobile users, while others were not. Their responses have led to a number of revisions in the final interface design, and have inspired potential future improvements.

6.1 User Comments, Responses and Actions Taken

- CNGL: *"Add edit button to make editing clearer. The alert box is annoying."*

  Status: "Edit Output" button added under output text box. Selection starts editing process.

  Previous version: Users were alerted to the possibility of post-editing by use of an alert box. The output box was editable at any point. On editing users could submit their changes.

- Non-smartphone user: *"Translation button is difficult to find."*

  Status: Button is in standard position. Other users have confirmed it was easy to find.

- CNGL: *"Change order of bottom buttons. Draws more attention to post-editing facility."*

  Status: The "Edit output" button has been moved to the top.

  Previous version: The "Edit output" button was at the bottom.

- Blackberry user: *"App doesn't appear correctly."*

  Status: Blackberry user. Upgrading to 6.0 solved problem.

- Non-smartphone user: *"I could use the output box when there was no output. Why?!"*

  Status: The output box is now disabled unless the user chooses the "Edit Output" option.

- iPad user: *"Translating from English to English didn't return my original input?!"*

  Status: Users may inadvertently make this error. Updated to simply return input.

  Previous version: Returned output - "Sorry we don't have a suitable language model for this translation".

- iPad user: *"I was able to translate, email and search for empty output."*

  Status: Fixed to check output box actually has output.

  Previous version: Actions were taken on empty output.

- Android user: *"I couldn't go back to the app after searching online without reopening it."*

  Status: Future updates will have a back button.

- iPhone user: *"I'd like if the page only showed the output once I translate."*

  Status: Future versions will implement this combined with a "Done" button to return to the original screen. Users will be able to enable or disable this feature in the application's settings.

  Previous version: The output box remains the same size, and all other content remains.

6.2 Test cases

Below is a final run through of the interface, testing all functionalities. The system was tested thoroughly throughout. This run was error free as previous testing exploited any issues, for which fixes were produced.

6.2.1 Translation

Step no.	Steps	Data	Expected results	Actual results
1	Without entering data try to translate.	Input -> None Source -> english Target -> french Subject -> general Output -> None	Alerts user to lack of input.	Alerts user to lack of input.
2	Enter data and translate.	Input -> “hello world” Source -> english Target -> french Subject -> general Output -> None	“Bonjour le monde”	“Bonjour le monde”
3	Enter data that will demonstrate special character output support.	Input -> “that was a great goal” Source -> english Target -> french Subject -> general Output -> “Bonjour le monde”	“C'était un grand objectif”	“C'était un grand objectif”
4	Enter data changing subject.	Input -> “that was a great goal” Source -> english Target -> french Subject -> sport Output -> “C'était un grand objectif”	“Qui a été un grand but”	“Qui a été un grand but”
5	Enter data changing target language.	Input -> “that was a great goal” Source -> english Target -> spanish Subject -> sport Output -> “Qui a été un grand but”	“Que ha sido un gran gol”	“Que ha sido un gran gol”
6	Enter data changing source language, demonstrating special character input support.	Input -> “Qui a été un grand but” Source -> french Target -> english Subject -> sport Output -> “Que ha sido un gran gol”	“That was a great goal”	“That was a great goal”

6.2.2 Refreshing

Step No.	Steps	Data	Expected results	Actual results
7	Press “⟳”	Input -> “Qui a été un grand but” Source -> french Target -> english Subject -> sport Output -> “That was a great goal”	Page returns to default settings. (empty text boxes).	Page returns to default settings.

6.2.3 Post-editing

Step No.	Steps	Data	Expected results	Actual results
8	Press “Edit output”	Input -> None Source -> french Target -> english Subject -> general Output -> none	Alerts user to lack of output to edit.	Alerts user to lack of output to edit.
9	Enter data and translate.	Input -> “hello world” Source -> english Target -> french Subject -> general Output -> None	“Bonjour le monde”	“Bonjour le monde”
10	Press “Edit output”	Input -> “hello world” Source -> english Target -> french Subject -> general Output -> “Bonjour le monde”	Output box becomes active, and cursor is in it.	Output box becomes active, and cursor is in it.
11	Change output and press “Submit”.	Input -> “hello world” Source -> english Target -> french Subject -> general Output -> “Bonjour monde”	Receive notification that edits are received.	Receive notification that edits are received.

6.2.4 Email

Step No.	Steps	Data	Expected results	Actual results
12	Press “Email”	Input -> “hello world” Source -> english Target -> french Subject -> general Output -> “Bonjour monde”	Default mail client opens with “Bonjour monde” in body of message.	Default mail client opens with “Bonjour monde” in body of message.

6.2.5 Search Web

Step No.	**Steps**	**Data**	**Expected results**	**Actual results**
13	Close mail application.	None.	Return to home screen of device.	Device dependent.
14	Reopen application.	None.	Loading screen appears for brief period, followed by loaded application.	Device dependent.
15	Enter data and translate.	Input -> “hello world” Source -> english Target -> spanish Subject -> general Output -> None	“Sorry, we do not have the appropriate model for this translation.”	“Sorry, we do not have the appropriate model for this translation.”
16	Press “Search web”	Input -> “hello world” Source -> english Target -> spanish Subject -> general Output -> “Sorry, we do not have the appropriate model for this translation.”	Chosen search engine opens with results for output.	Chosen search engine opens with results for output.

6.2.6 Database

Below is the above test case informations' entries in the database implemented as part of this interface.

translations

trans id	time	date	source	target	domain	input	output	edits	time InMS
187	18:09:01	11/03/23	english	french	general	hello world	Bonjour le monde	null	null
188	18:10:13	11/03/23	english	french	general	that was a great goal	C'était un grand objectif	null	null
189	18:13:45	11/03/23	english	french	sport	that was a great goal	Qui a été un grand but	null	null
190	18:15:45	11/03/23	english	spanish	sport	that was a great goal	Que ha sido un gran gol	null	null
191	18:19:29	11/02/23	french	english	sport	Qui a été un grand but	That was a great goal.	null	null
192	18:21:01	11/03/23	english	french	general	hello world	Bonjour le monde	Bonjour monde	4949
193	18:25:21	11/03/23	english	french	general	hello world	Bonjour le monde	null	null

edits

editid	transid	text
0	192	Bonjour l monde
1	192	Bonjour monde
2	192	Bonjour monde