**CHAPTER THREE**

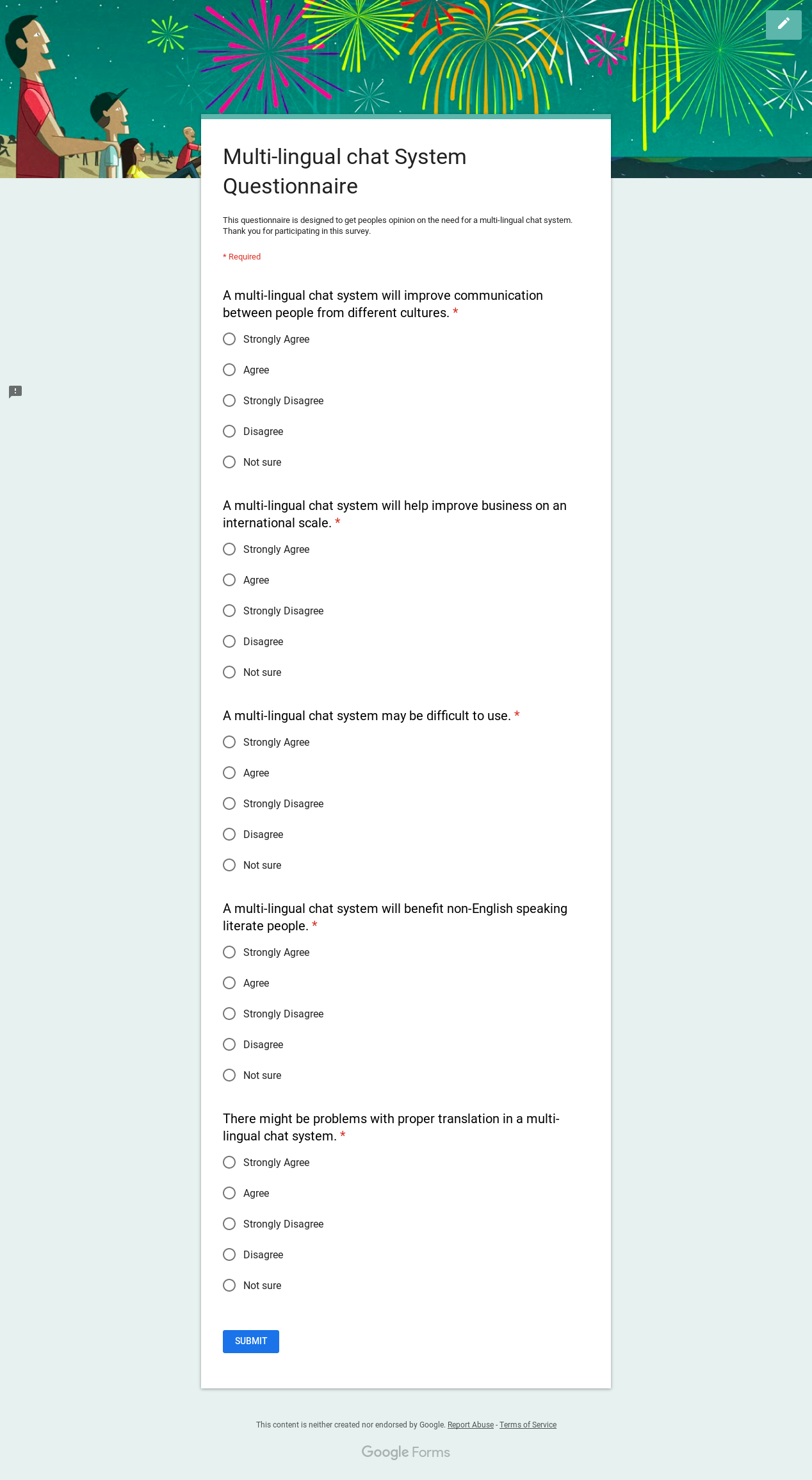
**SYSTEM ANALYSIS AND DESIGN**

**3.0 INTRODUCTION**

What is prevalent today as already seen in our literature review are chat systems with single end to end language. This solution provided communication regardless of distance but did not solve the problem of language diversity between communicating parties. In view of this, this work intends to bridge the literary gap left by the existing systems by designing and implementing a Multilingual Chat System. This chapter discusses the analyzes the proposed system and provides a detailed description of it’s design.

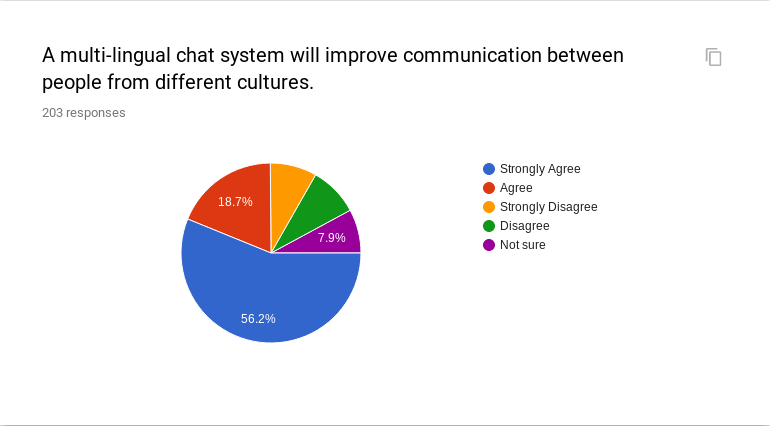
**3.1 DATA COLLECTION AND ANALYSIS**

A questionnaire was created with google forms. This questionnaire was created to get people’s opinion on the need for a multilingual chat system. The questionnaire comprises five statements with five options attached to each. These options are: strongly agree, agree, strongly disagree, disagree and not sure. This questionnaire was filled by students and staff of Yaba College of Technology. The figure below is a sample of the questionnaire.

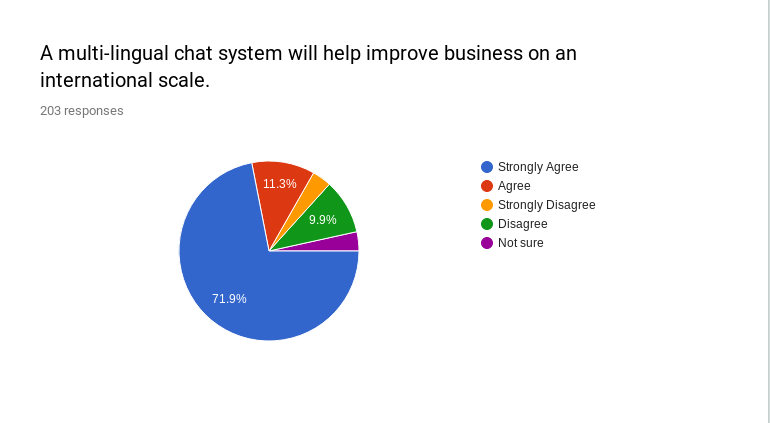


**Fig 3.1 Sample questionnaire**

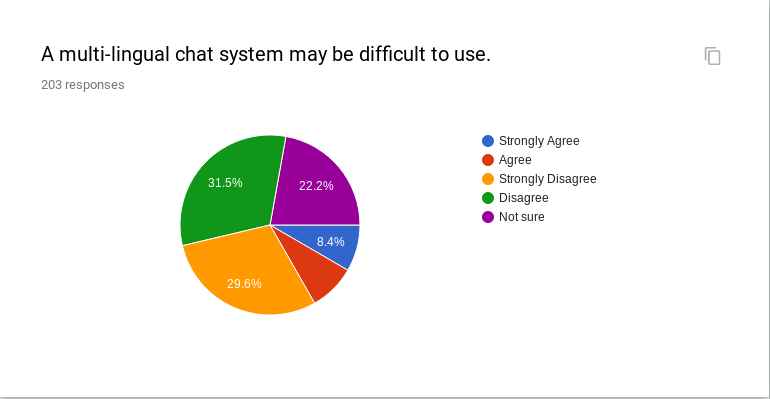
A total of 203 responses was gotten. The figures below shows a pie chart for each question in the questionnaire that shows the relative proportions of the responses received.



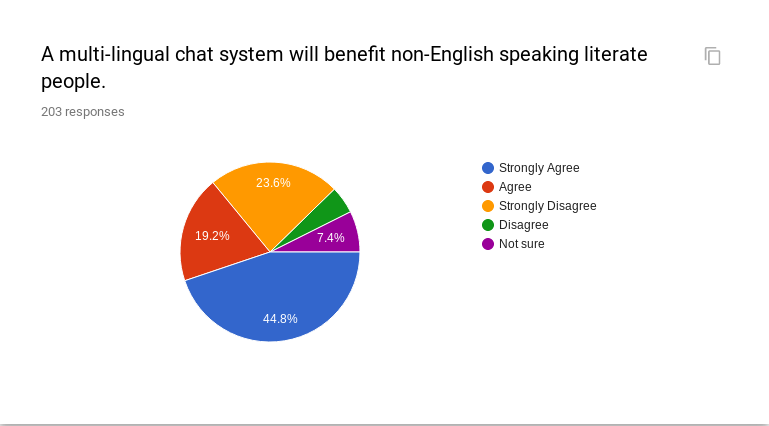
**Fig 3.2 Statistics of Response to the first question**



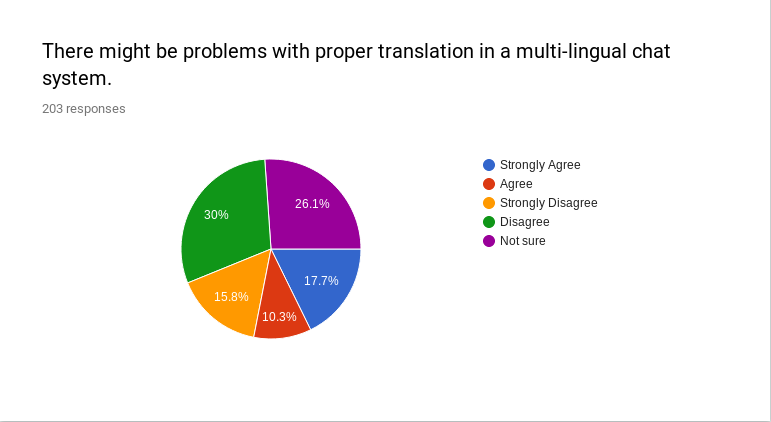
**Fig 3.3 Statistics of Response to the second question**



**Fig 3.4 Statistics of Response to the third question**



**Fig 3.5 Statistics of Response to the fourth question**



**Fig 3.6 Statistics of Response to the fifth question**

From the figures above we can clearly see that users believe that the proposed multilingual chat system will be of benefit by improving communication among people from different culture and as a result also help to improve business on an international scale. However, some people seem to be concerned about the ease of use and the translation efficiency of the system. These factors were, therefore, put into consideration in the design of the system.

**3.2 SYSTEM ARCHITECTURE**

For this system, the input data is the chat message sent by a user at one end. During the processing, the input is passed into a Language Translation System (whose methodology will be explicitly discussed in subsequent sessions). In the Language Translation System, algorithms are implemented to enable the input message reach the recipient in the language of his choice and also retain the context of the discussion as much as possible.

**Fig 3.7 demonstrates the architecture of the Multilingual Chat System**

**3.3 THE MACHINE TRANSLATION SYSTEM**

The basis of statistical machine translation comes from information theory. A document is translated according to the probability distribution  p(e|f) that a string *e* in the target language (for example, English) is the translation of a string *f* in the source language (for example, French). The probability distribution is modeled using Bayes Theorem, that is p(e|f)^(f|e)/p(e), where p(e|f) is the probability distribution that a string *e* in the target language is the translation for a string f in the source language while the translation model p(f|e) is the probability that the source string is the translation of the target string, and the [language model](https://en.wikipedia.org/wiki/Language_model) p(e) is the probability of seeing that target language string. This approach splits up the problem into two subproblems; finding the best translation is done by picking up the one that gives the highest probability.

For an implementation of this one would have to perform an exhaustive search by going through all strings in the native language. Performing the search efficiently is the work of a machine translation decoder that uses the foreign string, heuristics and other methods to limit the search space and at the same time keeping acceptable quality.

Systems that have implemented statistical machine translation include Google Translate, Microsoft Translator and Yandex Translate. For the proposed system Yandex Translate was used.

**3.3.1 HOW YANDEX TRANSLATE WORKS (ALGORITHM)**

As explained in the last section, Yandex Translate uses Statistical Machine Translation as its translations are not based on language rules. In order to learn a language, the system does a comparison of hundreds of thousands of texts containing the same information but in different languages. These texts might be from different language version of a company website for example. The system first identified parallel texts by their web addresses - web addresses differ by country codes or language codes (eg .en for English or .es for Spanish).

For each studied text, the system makes a list of unique characteristic. These characteristics may be rarely used words, numbers or special symbols found in a text in a certain sequence. When the system has gathered a large enough volume of texts with such characteristics, it begins to use them in its search for parallel texts – comparing those characteristics in new texts with those in texts it has already studied.

To meet current quality standards for machine translation a large amount of resources must be harnessed including lots of hard disk space, tons of RAM, and so on dude to the need of the system to study hundreds of millions of phrases in different languages.

In Yandex machine translation, there are three main components: a translation model, a language model, and a decoder.

1. **THE TRANSLATION MODEL**

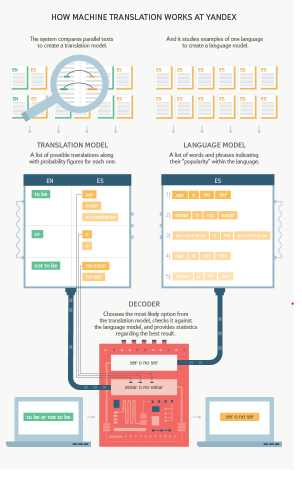
The translation model is a list of all the words and phrases known to the system in a single language, with all their possible translations into another language, including each translation’s probability. Each pair of languages has its own list. To create a translation model, the system has to, first, find matching – parallel – texts, then, find pairs of matching phrases within these texts, and only then find pairs of matching words or word combinations.

1. **THE LANGUAGE MODEL**

To create a language model – another component of Yandex’s machine translation technology – the system scans hundreds of thousands of texts in a target language – a language into which a text is translated – and puts together a list of the most frequently used words and word combinations, including information about frequency of use. This is the system’s knowledge of the target language.

1. **THE DECODER**

The actual translation is performed by the decoder. For every sentence of the source text, it selects all the translation options, combining phrases from the translation model, and sorts them in a descending order of probability. The decoder uses the language model to evaluate all the variations, and finds out the frequency of the occurrence of the different translation of the same word and then chooses the sentence with the highhest probability based on the translation model and the highest frequency of use based on the language model.



**Fig 3.8 Image showing how the Yandex Machine Translate Works Source: Yandex Legal Documents (2019)**

For the purpose of this work, the entire work process of the translation algorithm is packaged as an Application Program Interface (API) hence the working process and implementation of the above algorithms is implemented as a black box to the user. The result of the translation is returned in JavaScript Object Notation (JSON) format and rendered at the recipient's end.

**3.4 PROCESS DESIGN FOR THE TRASLATION SYSTEM**

**Fig 3.9 Process Design Flow Chart for the Translation System**

**3.5 SYSTEM DESIGN**

Considering the methodology discussed above, the system was designed using Ionic framework for building the mobile interface and AngularJS which is a Javascript library for building state of art user interfaces.

**3.5.1 SYSTEM FLOW CHART**

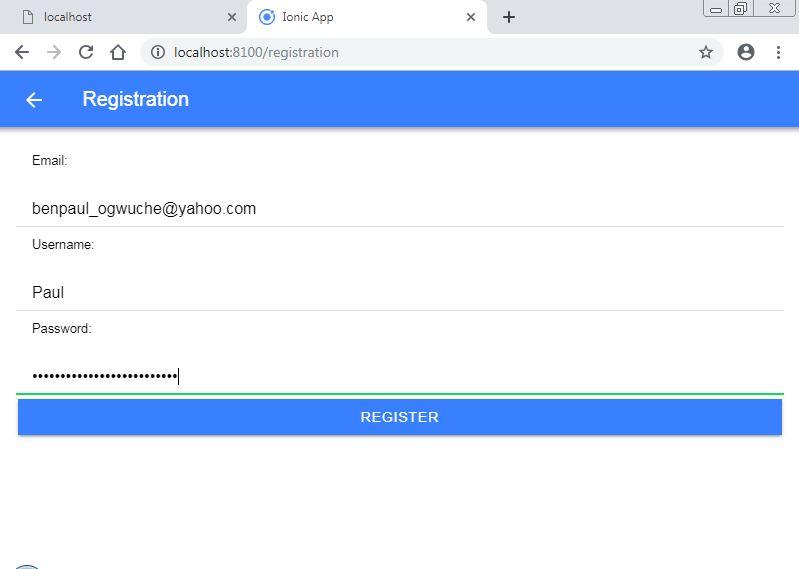
**Fig 3.10 System Flowchart**

**3.5.2 USER INTERFACE**

The user interfaces consists of the following:

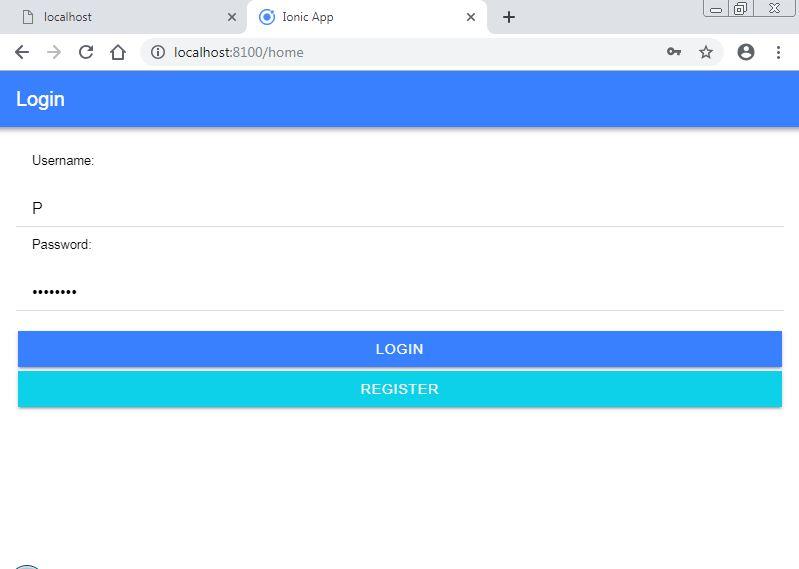
1. **Registration Page**

New users are required to register on the registration page in order to create an account in the database.



**Fig 3.11 Registration Interface**

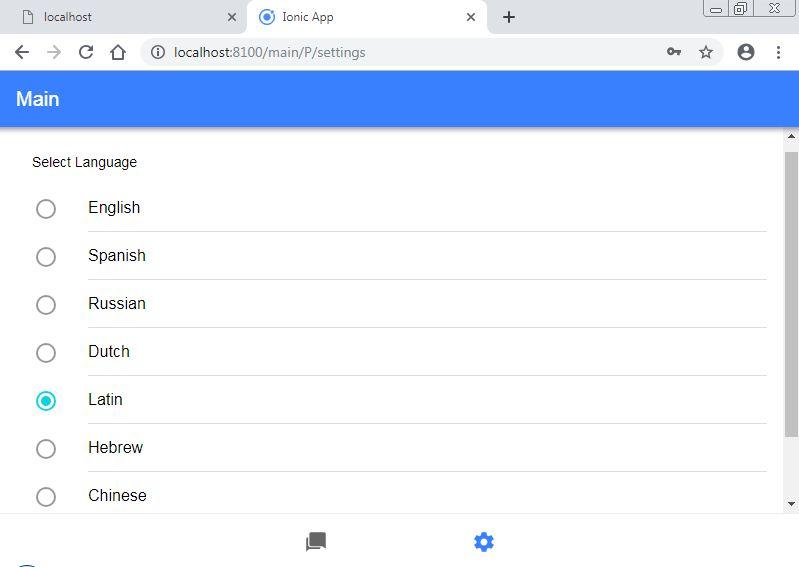
1. **Login Page**

There is a login page for existing users where they can log in to the chat system using their username and password. 

**`Fig 3.12 The user login interface**

1. **Select Language Interface**

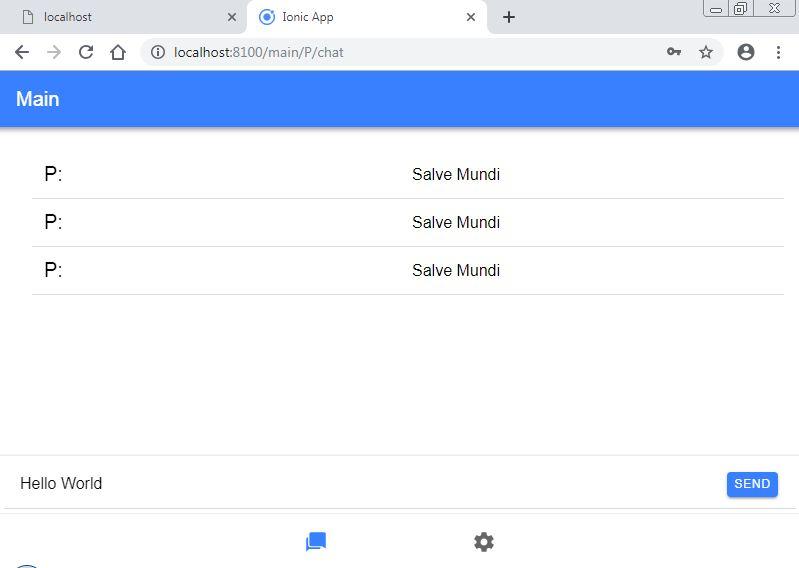
This interface is the most important and most relevant to the implementation of this system as it is the point where a user can select the language with which chats will be received. A Spanish speaking user for example can have his messages sent in spanish to an English speaking user who has selected English language has his chat language, the translation occurs between the sender and the recepient who only sees that the message he is receiving is in his prefered language.



**Fig 3.13 Language Selection Page**

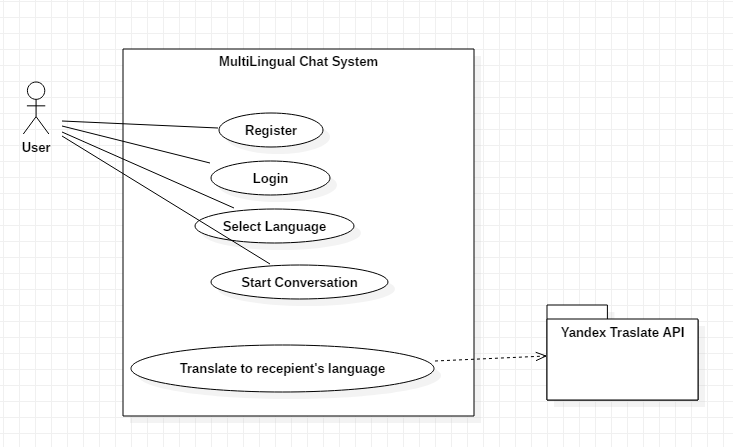
1. **Chat Room**

As natively called, the chat is where conversation takes place between users using the translation API as an intermediary to bridge language gaps.



**Fig 3.14 The Chat room**

**3.5.3 USE CASE DIAGRAM**



**Fig 3.15 Use case diagram for the implementation of the system**

**3.6 SYSTEM PROGRAMMING LANGUAGE AND ASSOCIATED LIBRARIES USED**

**3.6.1 PROGRAMMING LANGUAGE USED**

In implementing the system, AngularJS serves as the backbone language used. AngularJS is a structural framework for building dynamic web apps. HTML is used as the template language and its syntax are extended to express the application’s components clearly and succinctly. Angular’s data binding and dependency injection eliminate much of the code that might have been written. All these optimization happens within the web browser, this makes AngularJS good partner with client-side technologies.

**3.6.2 SOFTWARE LIBRARIES USED**

The system is implemented using Ionic framework which uses AngularJS as it’s backend engine. The Ionic framework is a framework used for building mobile applications. AngularJS is a javascript library that is widely used to develop state of the art user interfaces. Another library used is Cordova. Cordova is a library used by Ionic to provide native features to a mobile application. All the libraries used are downloaded through the Node Package Manager (npm). The Ionic framework is chosen for the following reasons: It is open source, code once, run on all mobile devices, one programming language for all mobile OSes, use of well-known web technologies, which you probably will employ if you need to develop a server-side of the system and good availability of "plugins"

**3.6.2.1 Ionic framework**

Ionic is a framework on top of Cordova which is a framework for building "hybrid apps". In general, using the hybrid app approach allows you to write apps as they were webpages (HTML, CSS and Javascript), which are then "run" inside a webview of a native app (basically an instance of the system's browser). What Cordova does, is extending the standard Javascript API with new APIs (packaged as "plugins") that give access to the operating system's features that are not usually available to the browser (like sensors, contact list, cameras etc.). Given that most mobile operating systems have decent browsers which are, roughly, all compatible with standards, and given that plugins are written (or should be) for all mobile operating systems, hybrid apps can be developed once for all mobile OSes.

**3.6.2.2 YANDEX API**

As discussed in previous sections, the Yandex API is a machine translation API that does the main work of translating between participant languages in a chat. It implements Baye’s Theorem in forming a statistical model for learning, translating and decoding texts written in a language to another language. The Yandex API has been optimized as much as possible to reduce the time latency in translating from source to destination language.

**3.7 HARDWARE SPECIFICATION**

The application requires hardware of a minimum of 4 gb RAM and 500GB HDD for the API. This is however provided by the Yandex API hence the user is saved from the headache of having to get a device with this configuration. The API only sends the output of the processing back to the client system. For deployment (Hosting), a host server with the above Hardware configuration of more can be used.

**3.7.4 SOFTWARE SPECIFICATION**

The implementation was done on a Windows 7 operating system but can be deployed (Hosted) on any server that provides with the software library dependencies listed in the System Programming section of this documentation. The user specifically needs to install on his PC one of Google Chrome, Mozilla Firefox, Safari or IE11 or above for desktop and the mobile version of the afore listed browsers for mobile devices.

**3.8 DATABASE IMPLEMENTATION**

In implementing the Database, MongoDB is used. MongoDB is a noSQL database management system. The database has two collections: a picture collection where pictures are stored. This information includes: the picture id and the binary format of the picture. It also has a user collection where information about the user is stored. This information includes: the user id, the user’s username and password, the user’s chat log and the user’s language preference.

**CHAPTER FOUR**

**RESULT AND DISCUSSION**

**4.0 INTRODUCTION**

The multilingual chat system works effectively for communication between parties with different languages. The problem of language barrier in socialization for Business or other purposes is been dealt with and also accompanying this is an intuitive user interface that enables a user know his way around the application without being thought how to use it.

**RESULTS**

The translation functionality of the application was tested using several test data from already known database of language to language translation equivalences.The test set is gotten from the official records (Hansards) of the 36th Canadian Parliament. It contains 1.3 million pairs of aligned text chunks (sentences or smaller fragment) from English and French.

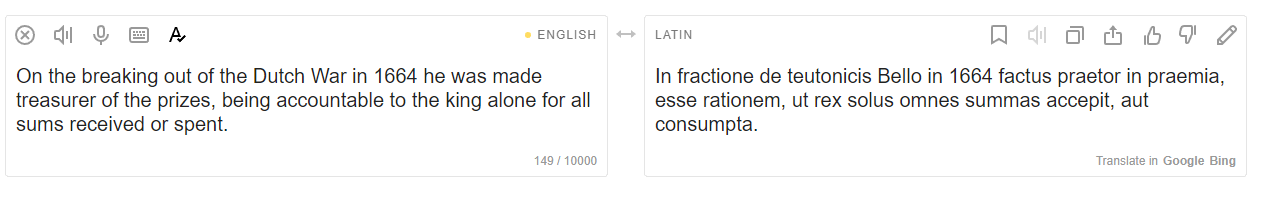
A unit test was performed on the Yandex API as a major component of the entire system. The results from this translation was evaluated by comparing the accuracy to the standard translations from the test set above. 

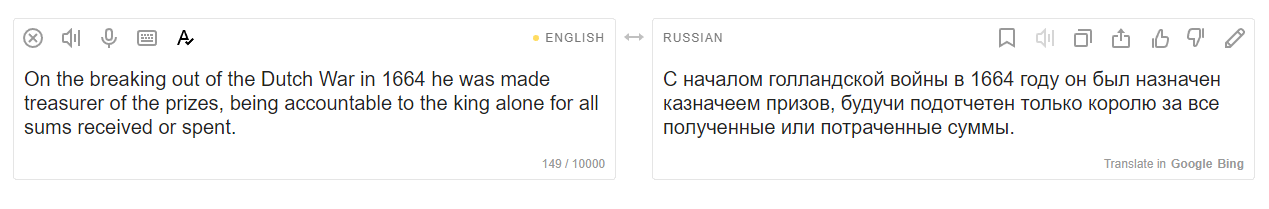
**Fig 4.1 Text from English to French**



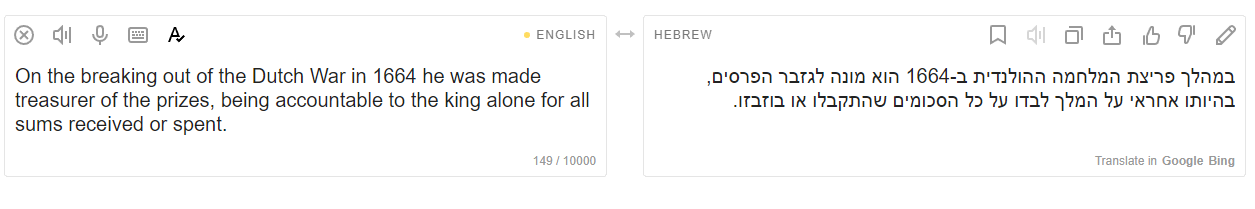
**Fig 4.2 Text from English to Spanish**



**Fig 4.3 English to Dutch translation**

**Fig 4.4 English to Latin Translation**

**Fig 4.5 English to Russian Translation**

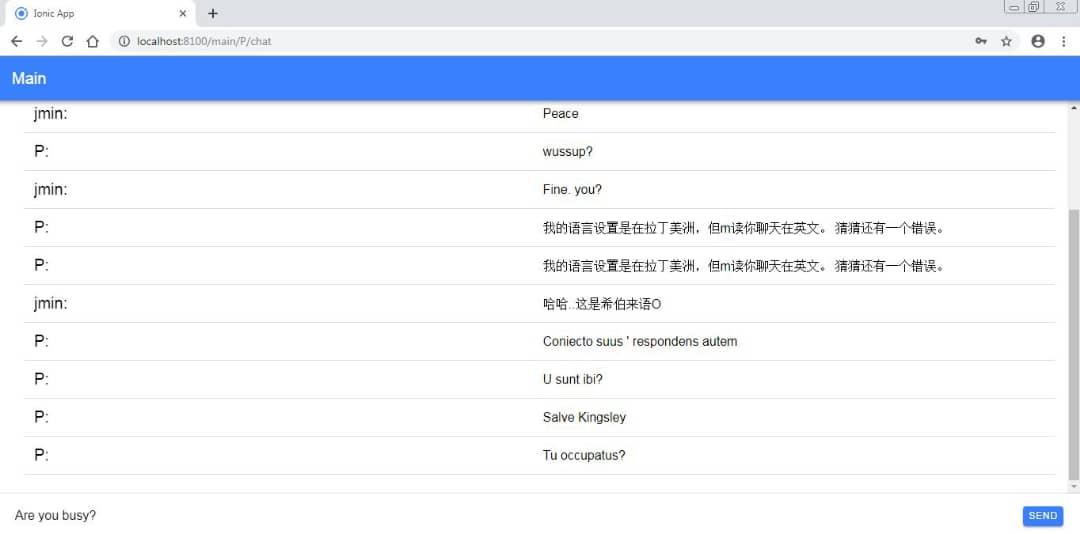


**Fig 4.6 English to Hebrew Translation**



**Fig 4.7 English to Chinese Translation**

The user interfaces were also tested by running the application on a local host which had already installed all the software dependencies mentioned in the chapter 3 of this work.

**Fig 4.8 System before translation**

**Fig 4.9 System after translation**

User Interface showing how text typed in English on clicking send button at the bottom right is translated to Select Language of a user with username P.

**DISCUSSION**

From the tests done, it is seen that the system not only translates but keeps the context of the translation as it moves from one language to the other.

Evaluation shows that the Yandex API translates from source to destination language very accurately.The user is also at the liberty of typing his message in any language of his choice and can receive messages in that same language.

The result of the test shows that the main aim of this work has been achieved, that is to Build a Multilingual Chat System which is up and running.

**CHAPTER FIVE**

**SUMMARY OF FINDING, CONCLUSION AND RECOMMENDATION**

**5.0 INTRODUCTION**

During the course of this project, the proposed model was introduced, analyzed, designed, implemented and evaluated in order to achieve the aforementioned objectives. We used the Yandex API together with other technologies to Build a Multilingual Chat System which translate between languages accurately keeping the context of the discussion intact.

**5.1 SUMMARY OF FINDINGS**

As far as Machine Translation is concerned, the accuracy of translation from one language to the other is strongly determined by the context of the discussion hence a need for a statistical approach as used in this work. The best approach is to implement a statistical approach that does not only translate between languages but also choose the best translation based on the context of the conversation. Translation in a Medical Register for example might not make sense if done directly without applying a statistical model in knowing which of the synonyms of a word fits the context best.

**5.2 CONCLUSION**

The best approach if considering translation between foreign languages is the Yandex Translate Engine which is free to use with a user friendly, easy to implement API. Google Translator could however be used if the problem domain cuts across indigenous languages as it has the largest translation database. Google Translator however does not have a free API.

**5.3 RECOMMENDATION**

The main target of this system is to aid communication without considering language barrier. This project provides a major breakthrough to this. Research has shown that there is no work of life that does not require communication hence this system can be deployed anywhere language barrier needs to be removed. However, this system is recommended for students who communicate more comfortably in languages other than that which is used in their place of study.

**5.4 SUGGESTION FOR FURTHER STUDY**

As at when this work was completed the following are not implemented in the work:

* Translation to indigenous languages
* Implementation as an Android App rather than a web based app

The above subjects may be looked into as a progression of this work.