



# Auto-Reply WebApp

09.05.2016 - 08.07.2016

---

Tejesh Raut

B Tech Second year  
Computer Science and Engineering  
IIT Bombay, Powai -400076

Email: [tejesh22.raut@gmail.com](mailto:tejesh22.raut@gmail.com)

## Overview

This is a webapp by which user can get replies to the queries asked. It also has an offline admin interface. Admin can make any kind of changes in the database. There is a graphical interface for admin through which (s)he can Ask a query, Display the entries stored, Add new entries, Load the data from files to program, Save the data to files, Delete an entry, Refresh the keywords, Add new non keywords, Add new synonyms, Terminate the program.

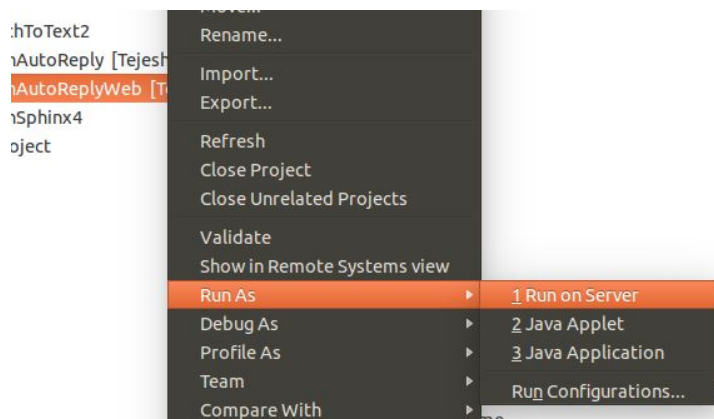
## Goals

1. Create a virtual chat box for any website (for example for an airlines company) where user can ask any questions as if he is chatting with some person from company.
2. Implement some web crawling functions for some special queries so that when user asks for some special query, computer will display information after scraping from the websites.

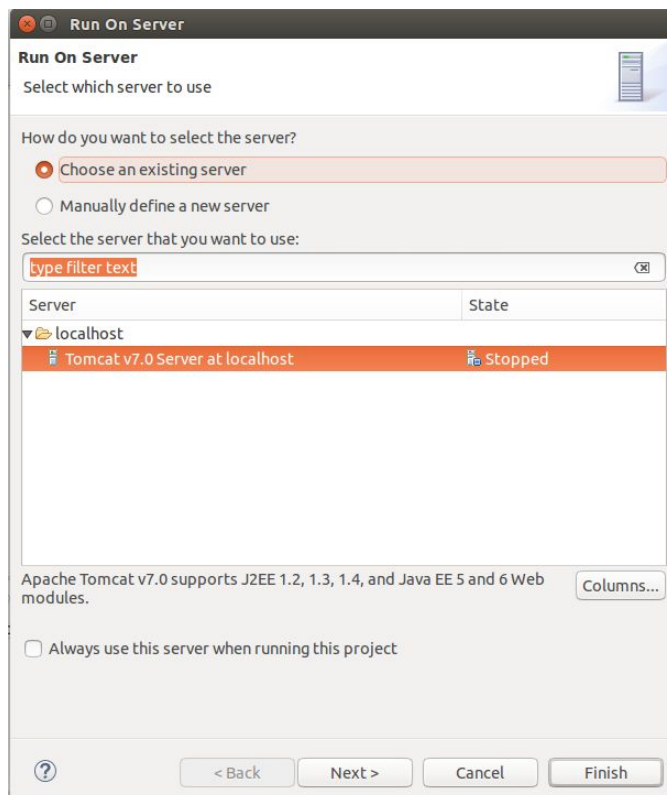
## User's Manual

### Client mode

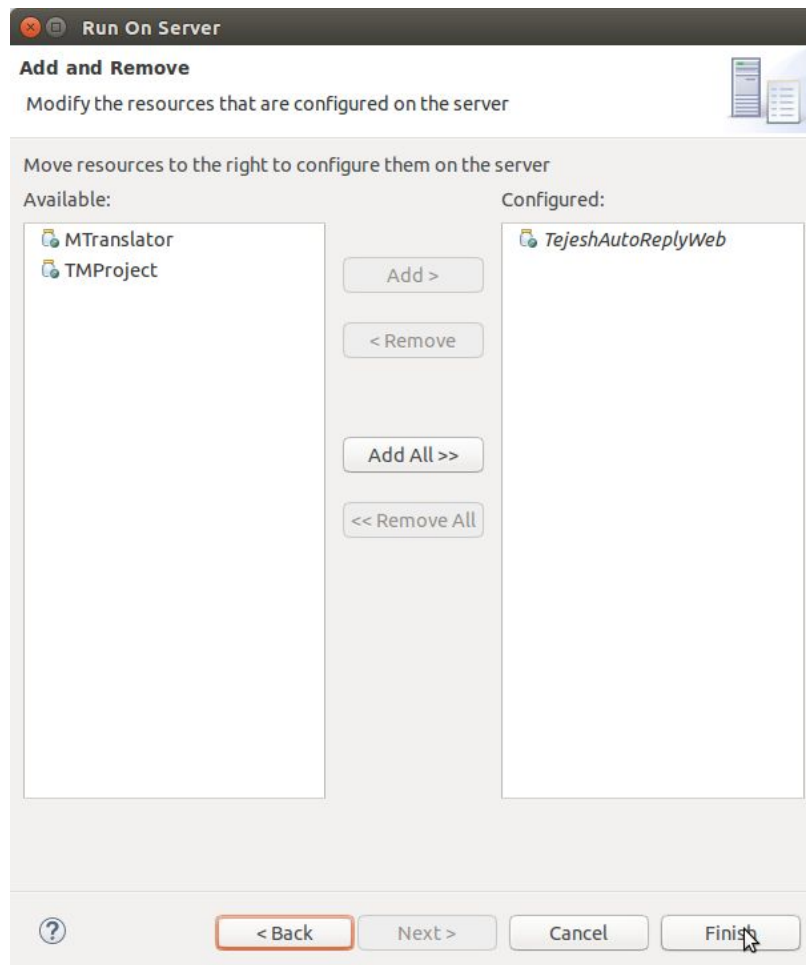
To turn on the client mode you will have to run the program on server. To run the program on server open Eclipse. On the left side you will see the project navigator. Right click on the project from there and then go to "run as", then select "run on server".



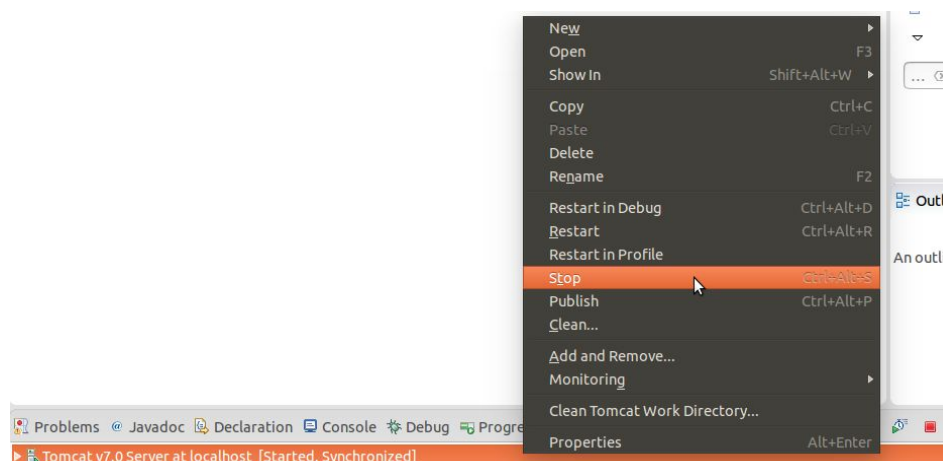
After this run on server window will pop up:



Choose your server and then click next. In the next window make sure that your project is added to the “configured” list and then click on finish.

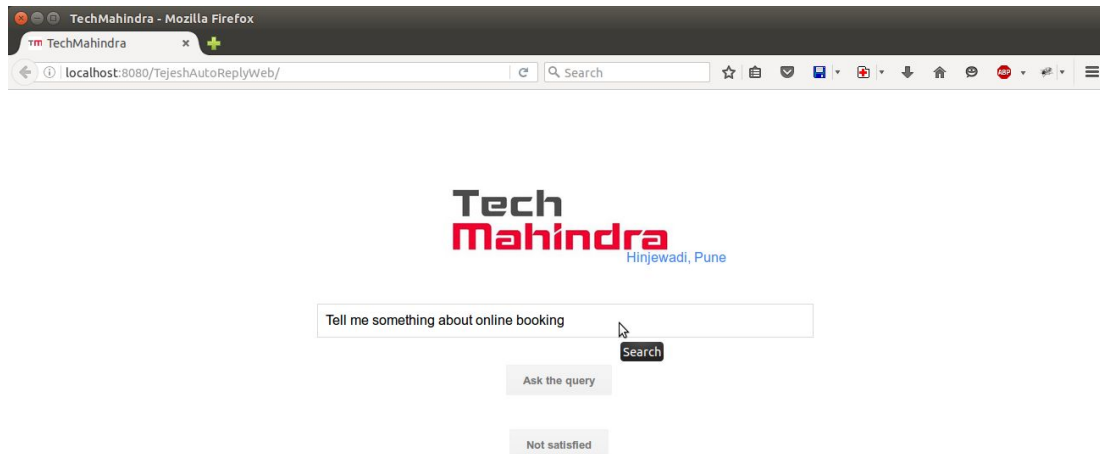


From the next time onwards you can start or stop the server by just selecting the server from the server's tab in bottom and selecting start, stop, restart from the right click menu.

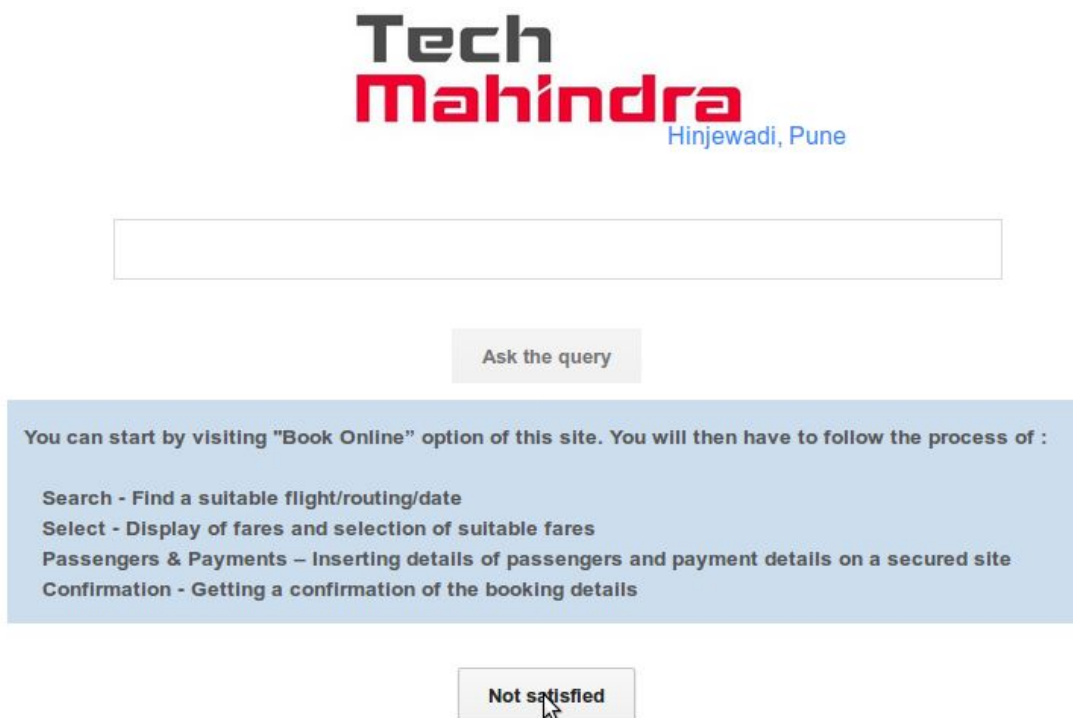


Now that the program is running on server you can test it by opening the link <http://localhost:8080/TejeshAutoReplyWeb/> on your browser

Ask your query in the search box:



Then you click on “ask the query” button.



If you are not satisfied you can get the next answer



Ask the query

Currently we do not offer this service of booking a ticket online and paying later online. The full payment has to be made at the time of booking.

Not satisfied

Go on clicking “Not satisfied” until you get the answer to your query:



Ask the query

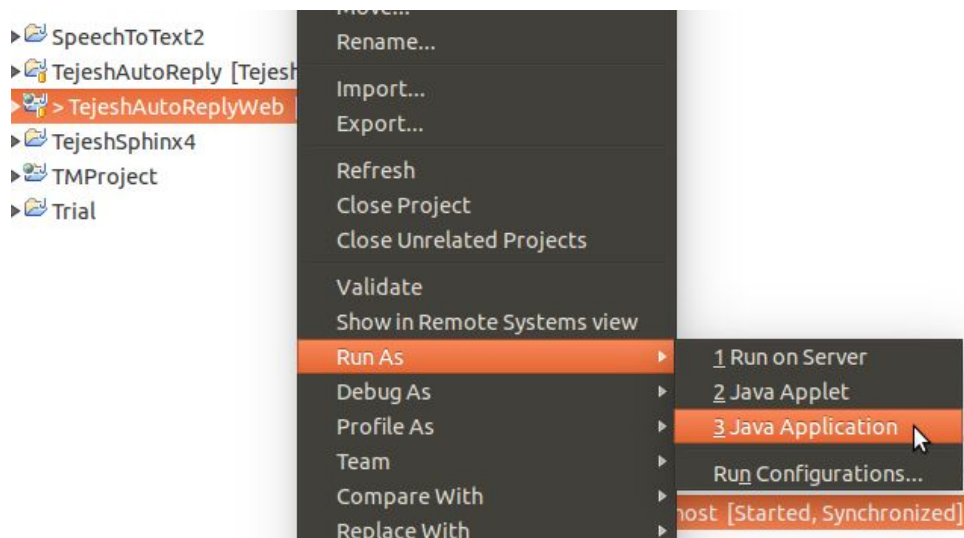
Online booking is a convenient way of booking your travel over the Internet. Using the Air India online services you can:

- Book a flight and pay online for your travel
- Request for a particular seat, special meal or service
- Redeem your frequent flyer miles online
- Check Flight status
- View the Schedule/timetable
- Web Check in
- Make an online booking using the NGPAY mobile app

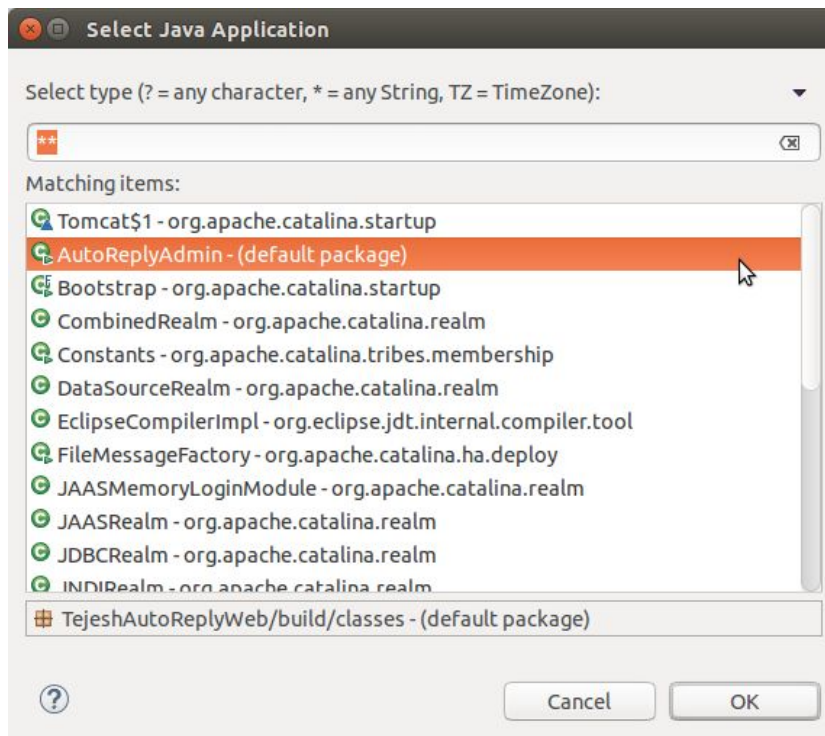
Not satisfied

## Admin mode

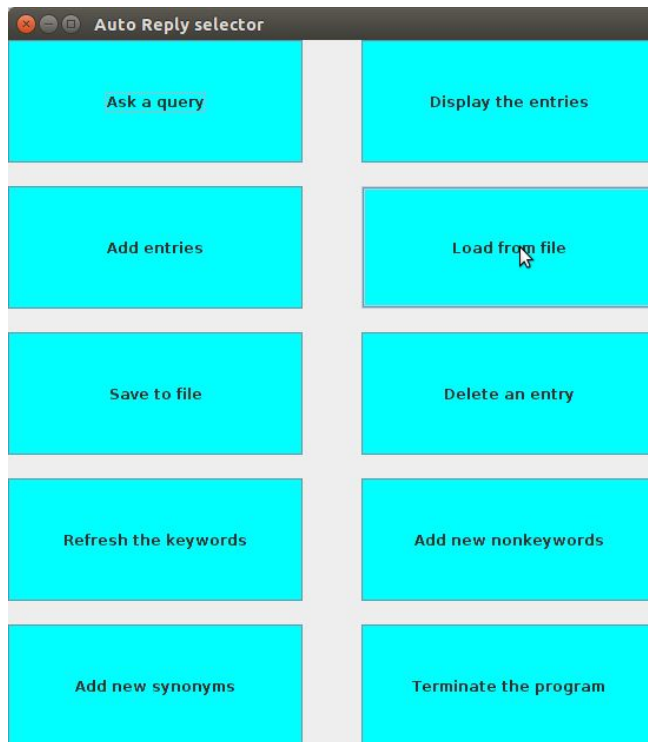
To run the program with admin mode, in Eclipse on the left side select your project from navigator. Right click on the project from there and then go to “run as”, then select “Java application”.



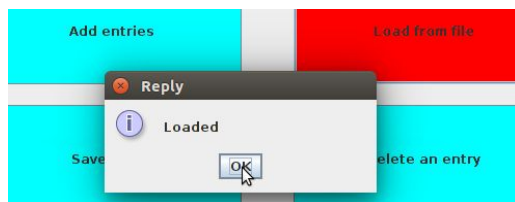
Then it will ask you to select java application. Then select “AutoReplyAdmin - (default package)” and click OK.



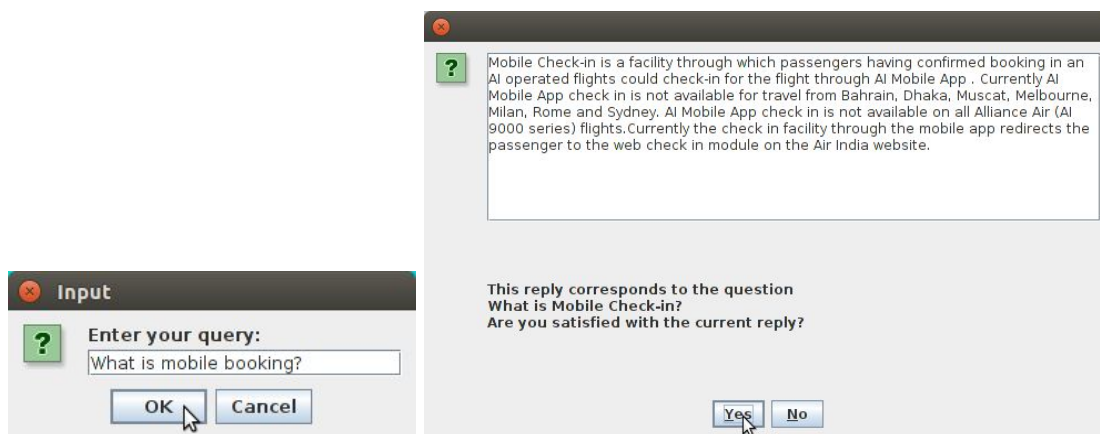
Then you should see the Auto Reply Selector window:



To begin click the button “Load from file”. This will load complete data from files.

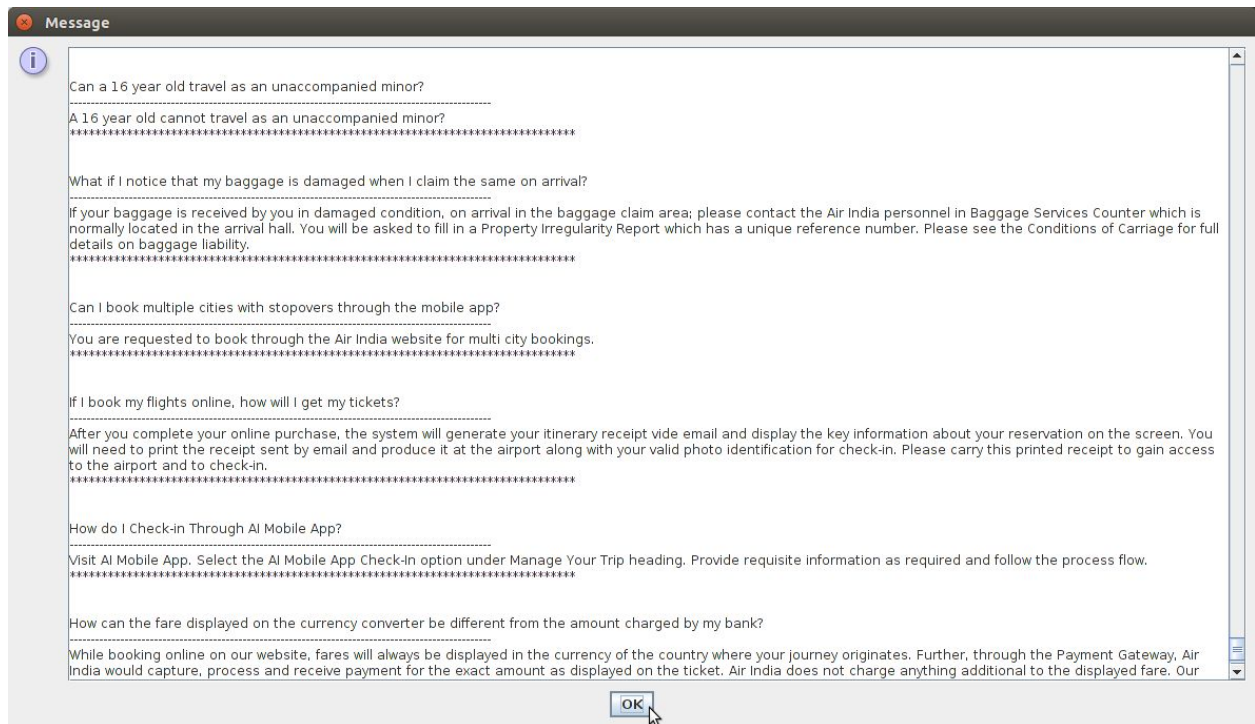


To ask a query click the first button “Ask a query” and then enter your query in the dialog box:

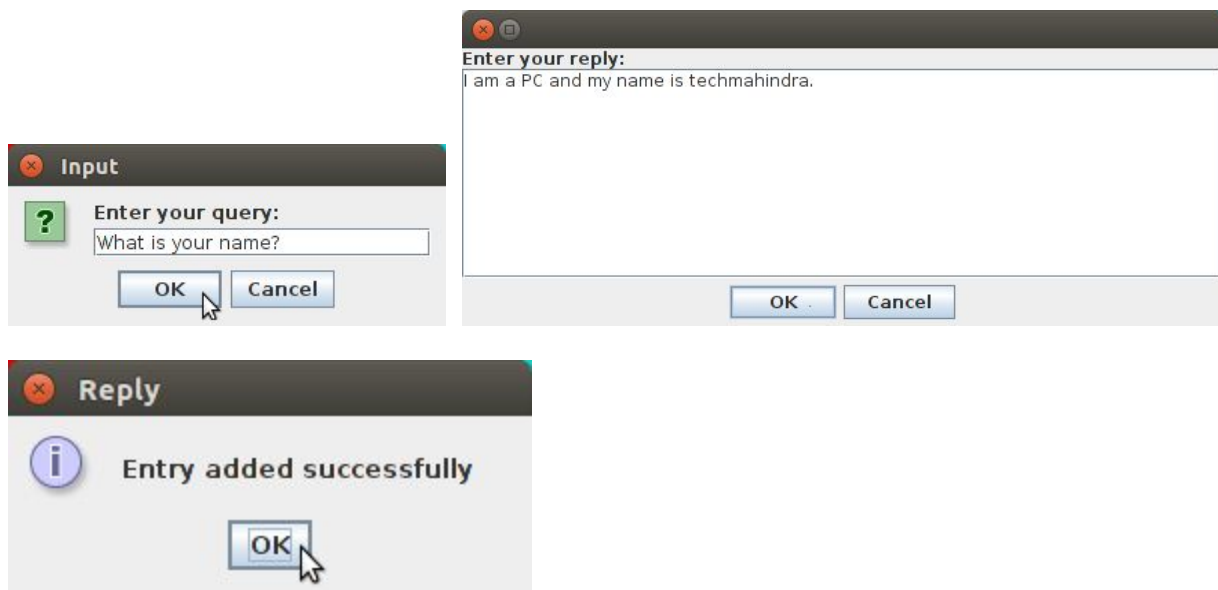




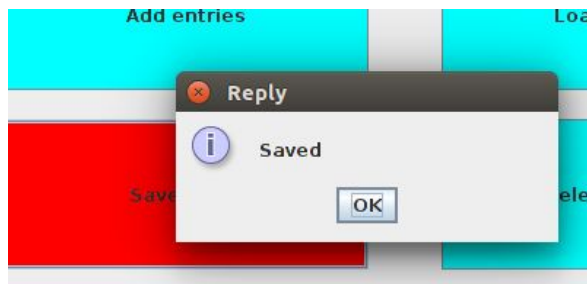
To display the data already stored click “Display the entries”:



To train new data select “Add new entry”:



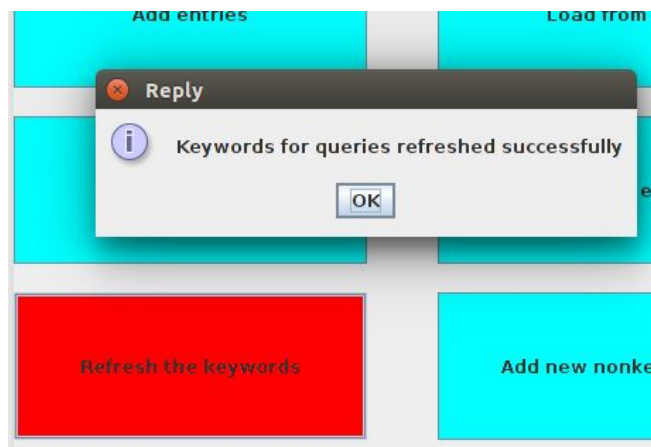
After training data for the present session, you must save it back to file so that you can use it in the next session. Click on the button “Save to file”



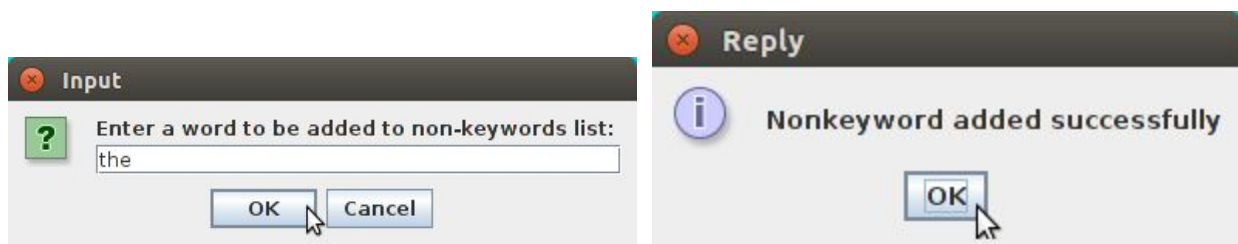
To delete an entry Click on “Delete an entry ” and then enter the query of the entry which you wish to delete. It will ask for confirmation and then delete that entry.



To refresh the keywords of all the queries present select “Refresh Keywords” button. To know more about how keywords are selected please go through the Specifications present in the next sections.



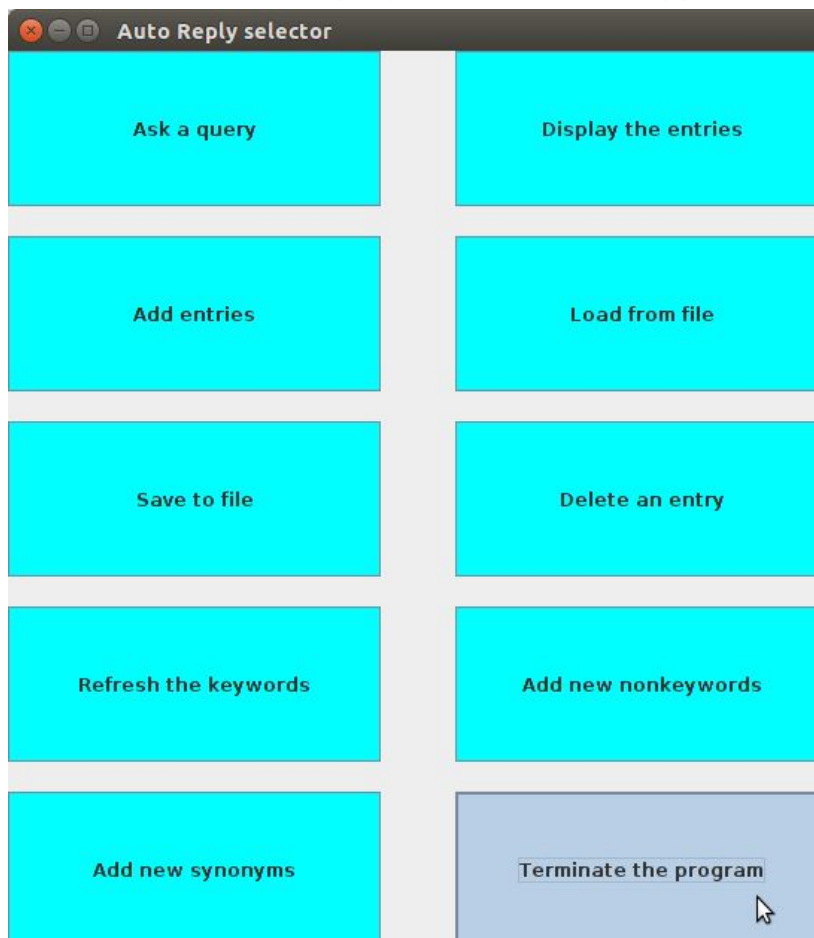
To add a new non-keyword:



To add new synonyms:



Click on "Terminate the program" to close the application:



## Specifications

### Overall working

Admin stores new queries and replies to train itself to answer further queries. Admin adds new synonyms (words with similar meaning ). Admin adds non keywords (words which don't need to be mapped. I.e. words with 0 weight in mapping). User asks a question and then gets a reply from the stored replies. If (s)he is not satisfied, then (s)he can give not satisfied feedback to get some better answer.

### Hashmaps and Properties

Two Hashmaps are used. One to store the mapping of keywords with the replies and another to store the mapping of keywords with the original queries. Hashmaps are stored in the file using Properties class so that the files are readable and in case of any modifications, they can be modified easily. There are two files where hashmaps are stored, query.properties stores hashmap mapping keywords with queries and reply.properties stores hashmap mapping keywords with replies.

### Synonyms and Non Keywords

Synonyms and non keywords are stored in the txt files. Non keywords are all stored in a single line separated using spaces. All synonyms of a word are stored in a single line separated using spaces and different words in a new line.

### Jframe to display the selector menu

A main Jframe consisting of two JPanels, each consisting of 5 buttons for different options is used to display the selector menu. Each button open up a different GUI for performing specific tasks.

### Loading data into Hashmaps

All data stored in the two properties files - reply.properties and query.properties is loaded into their respective hashmaps. If any of these files is not found a dialog box pops up to display "File not found". If the data is successfully loaded a dialog box pops up telling "Data Loaded".

## Displaying the database

Each stored query is displayed along with its answer in a JFrame which is in return embedded in a JTextArea. It contains vertical scrollbar. There is no need of horizontal scrollbar as line wrap and wrap style word are set true which means words get displayed on a new line if they exceed the horizontal length of the text area. Also the text is not editable because it is used just for displaying.

## Training new queries and answers

New query is taken input in a simple input dialog box. For inputting replies I took a readymade template from [this thread](#) . It allows input to be taken in multiple lines. After that query is processed to extract only keywords from the query and then store both the original query and the answer into the two Hashmaps. While extracting the keywords, all special symbols from the query are removed and hyphens and apostrophes are replaced by spaces and then remaining words are checked for non keywords using the data of non keywords from the file. Note that even after training the trained data is stored only in the temporary hashmaps and not stored in the permanent files.

## Saving the Hashmaps into the two properties files

Two Hashmaps are used. One to store the mapping of keywords with the replies and another to store the mapping of keywords with the original queries. Hashmaps are stored in the file using Properties class so that the files are readable and in case of any modifications, they can be modified easily. After saving a message dialog box pops up showing that data is saved.

## Adding new non keywords

Non keywords are stored in the file NonKeywords.txt in a single line separated by spaces. Using this option a input dialog box is displayed to give new non keyword which is then stored in the file NonKeywords.txt. After that the string storing non keywords is also refreshed according to the file.

## Adding new synonyms

Synonyms are stored in the file Synonyms.txt with different words on a new line and synonyms in the same line separated by spaces. Using this option an input dialog box is displayed which prompts to enter synonyms of a word separated by spaces. The input is then stored as it is in a new line in the file Synonyms.txt.

## Refreshing non keywords

Keywords get stored only during training. They don't refresh after launching the application or while loading the data. It may be the case that the admin wishes to store the keywords according to the new non keywords. In this case he can refresh the non keywords. The whole list of original queries is read and keywords of each query are refreshed in this process. Both the hashmaps of queries as well as answers are refreshed in this process.

## Replying to a query in admin version

Query is taken as input in a simple input dialog box. Then five sorted best matching queries are taken from data stored in the hashmaps. The reply of the best matching query is displayed in a non editable text area with a vertical scrollbar. Feedback is asked after displaying the reply. If the person who asks is not satisfied he can click "no" button and then reply of next best matching query will be displayed. He/she can view 5 such answers. After that a message box will be displayed to ask query in a different words.

## Deleting an entry from database

Query is taken as input in a simple input dialog box. Then five sorted best matching queries are taken from data stored in the hashmaps. The best matching query and reply of the best matching query is displayed in a non editable text area with a vertical scrollbar. A confirmation is asked before deleting. If confirmed the entry is deleted from both the hashmaps. Note that it is deleted from hashmaps and still present in the files before being saved.

## Replying to a query in user(browser) version

A web page is created for user. It is mostly copied from [Google's](#) page. The unnecessary elements from the page are removed. Only the search box and submit buttons are kept. Logos of Google are replaced by logos of Tech Mahindra. "Google search" button is renamed as "Ask the query" button. Query is taken as input in the search box and reply is displayed just below "Ask the query" button. If the reply is of many lines, a vertical scrollbar appears automatically. Just below it is a similar button "Not Satisfied". It displays the next best reply just as the "no" button of admin interface. It displays maximum 5 replies after which it asks to ask to enter the query in a different way. The reply string is formatted according to the HTML using readymade template from a [Stackoverflow answer](#) and the web URLs referred in the replies are automatically added a link in HTML using a readymade template from [other answer](#).

## The algorithm used for mapping query

Query inputted from user is formatted as it is done while training new data. All special symbols are removed, hyphens and apostrophes are replaced by spaces. Then the query is splitted into words using spaces as delimiters and stored in an arraylist. Now using synonyms of each word of the query, an arraylist of strings which is created. This arraylist contains all the possible ways of representing this query string.

Eg. If a query of 4 words is asked and number of synonyms present of each words are 2, 5, 3 and 1 respectively, then the number of ways this query can be represented is  $2*5*3*1 = 30$ .

Each of the possible way of representing the inputted query string is checked with each query string present in the database.

For comparison with two strings the Levenshtein distance between them and the number of words matching between two strings are checked. Then Levenshtein distance is divided by number of words to get the final score. The combination having least score is regarded as the best match.

## Milestones

### I. Implementing hashmaps

This was the first step of the project. Hashmap was a good data structure for string to string mapping. I used many codes for saving, loading hashmaps from files. Later on I realized that, it is good if we able to implement these functions during running program for debugging and testing by admin. Initially I had no idea that this is to be implemented on web. I kept the switch case block in an infinite while loop. There was only one hashmap which directly mapped queries with answers.

### II. Initial matching algorithm

The initial matching algorithm was to find the number of common words between the query asked and the queries stored. The query which matched most number of words was selected as the best match. Answer of that query was displayed as the reply.

### III. Need to implement non keywords

During matching the number of words, all words were counted in matching. Because of this some common words like a, an, the, is, etc were also getting included in the number of common words. This often resulted in mapping of different queries. So to solve this problem I kept a file NonKeywords.txt to store the non keywords. Whenever a new query is stored while training, it used to store only the keywords instead of the complete query. So the hashmap stored mapping of keywords with queries.

### IV. Implementing synonyms

To make the application a bit more smart, it should be capable of replying the queries even if they were asked in some other words. So I added a file to store the synonyms. Now number of words common to two strings were calculated even after checking whether their synonyms matched. This improved the search results to a great extent. I found a guy using this technique in a [post in Stackoverflow](#). He was also making a program which can respond to what user said, something like chatter bot.

### V. Storing original queries for refreshing keywords



Some words were initially thought to be non keywords and then later they were found to be important for mapping. Because of this I had to build up whole database again. I deleted the old database and then before making the new database, I included the hashmap for storing the original queries and a function for refreshing the queries according to the new non keywords.

## VI. Creating graphical interface

I had some previous knowledge of swing library of java. Although I had not done any good work using swing, I knew that I can make a complete graphical application using the swing library. So I converted the infinite switch case commands into Jbuttons. Initially I kept the graphical interface just for selecting one of the ten options. Later on slowly I integrated every function into graphical interface, so that nothing needs to be done via console and it gets looks of a finished application. I browsed through many posts on internet to implement particular graphical elements for every function.

## VII. Creating web interface

While starting this project there was no thought of hosting it over web. Before this project I had worked in making of webapps using Django and spring tool set. I had no knowledge of java servlets prior to this. After searching from internet, I came to know that java servlets were good option to turn an existing program in java to a webapp. For making a servlet I had to begin from scratch, so I kept the older part of project as admin interface for the person handling the server and a new part of webapp which would only do the job of answering the queries to user on a webpage. I had to download the eclipse version for web developers and install the tomcat server. Then I created a new dynamic web project in Eclipse and copied the old files to it. Then I used the Google homepage and edited it to the present jsp page. I learnt the use of java servlets and implemented the search function over the web page.

## VIII. Improving the matching algorithm

I filled up a large data of queries and answers from Frequently Asked Questions at Air India [website](#). Then it was working good for many cases but still failed for some cases. For example “online booking” these two words were present in many queries. A good algorithm should map it with query “What is online booking?” and not other long long queries. I searched for many string comparison algorithms on internet and found the Levenshtein distance technique. Levenshtein distance between two strings is the number of characters to be added or deleted from one of the string to make it similar to the second string. Again this way of matching

strings caused errors, as it would get easily mapped with the smallest queries even if the queries did not have any word in common. So I decided a new way of matching the strings. I used a new score to decide the matching. Score was obtained by dividing the Levenshtein distance by the number of words common between the two strings. This solved both the above problems and is still working fine.

## References

[http://www.tutorialspoint.com/java/java\\_hashmap\\_class.htm](http://www.tutorialspoint.com/java/java_hashmap_class.htm)

<http://beginnersbook.com/2013/12/hashmap-in-java-with-example/>

<http://stackoverflow.com/questions/12747946/how-to-write-and-read-a-file-with-a-hashmap/12748101#12748101>

<http://stackoverflow.com/questions/8798403/string-is-immutable-what-exactly-is-the-meaning/17942294#17942294>

<http://stackoverflow.com/questions/13659130/how-to-store-and-check-synonym-of-string-in-java>

<http://stackoverflow.com/questions/21989887/how-to-print-lines-from-a-file-that-contain-a-specific-word-using-java/21993601#21993601>

<http://stackoverflow.com/questions/11536089/making-all-button-size-same/11536419#11536419>

<http://stackoverflow.com/questions/20595418/how-to-simulate-or-bring-console-with-scanner-input-into-iframe-application/20595500#20595500>

<http://stackoverflow.com/questions/21485590/the-final-local-variable-cannot-be-assigned-since-it-is-defined-in-an-enclosing/21485716#21485716>

<http://stackoverflow.com/questions/14737810/jlabel-show-longer-text-as-multiple-lines>

<http://stackoverflow.com/questions/8606008/running-a-java-application-on-a-webpage>

<http://stackoverflow.com/questions/5064733/several-ports-8005-8080-8009-required-by-tomcat-server-at-localhost-are-alre/5065874#5065874>

<http://stackoverflow.com/questions/19247099/how-to-serve-existing-web-page-using-a-java-servlet>

<http://stackoverflow.com/questions/2370960/generate-an-html-response-in-a-java-servlet/2371656#2371656>

<http://stackoverflow.com/questions/5134959/convert-plain-text-to-html-text-in-java/5135149#5135149>

<http://stackoverflow.com/questions/18183296/how-can-i-add-a-vertical-scrollbar-to-my-div-automatically>

<http://stackoverflow.com/questions/5134959/convert-plain-text-to-html-text-in-java/5139200#5139200>

<http://stackoverflow.com/questions/14001799/does-arraylist-clear-resize-the-array-to-the-smallest-possible-size/14003379#14003379>

<http://stackoverflow.com/questions/8611196/how-do-i-import-an-existing-directory-into-eclipse/8611662#8611662>

<http://stackoverflow.com/questions/11219678/apache-tomcat-installation-directory-in-ubuntu-configure-tomcat-in-eclipse-u>

<http://stackoverflow.com/questions/1748887/how-to-create-a-smart-chat-bot>

<http://stackoverflow.com/questions/1748887/how-to-create-a-smart-chat-bot/1749527#1749527>

<http://stackoverflow.com/questions/12053241/sentence-comparision-with-nlp>

<http://stackoverflow.com/questions/62328/is-there-an-algorithm-that-tells-the-semantic-similarity-of-two-phrases/179071#179071>

<http://www.airindia.in/online-booking-faqs.htm>

[https://en.wikipedia.org/wiki/Natural\\_language\\_processing](https://en.wikipedia.org/wiki/Natural_language_processing)

<http://www.javaprogrammingforums.com/awt-java-swing/1210-need-help-jbutton-switch-statements.html>

<http://chatbotfriends.altervista.org/opensource.html>

<https://commons.apache.org/sandbox/commons-text/apidocs/org/apache/commons/text/similarity/package-summary.html>

[https://en.wikipedia.org/wiki/Outline\\_of\\_natural\\_language\\_processing#Chatterbots](https://en.wikipedia.org/wiki/Outline_of_natural_language_processing#Chatterbots)  
[www.pandorabots.com](http://www.pandorabots.com)  
<http://www.coderanch.com/t/379461/java/java/passing-html-text-java-String>  
<https://sourceforge.net/projects/qaiml/files/>  
<http://www.coderanch.com/t/329781/GUI/java/read-text-area>  
<http://rebecca-aiml.sourceforge.net/>  
<http://www.java-examples.com/remove-all-elements-java-arraylist-example>  
<http://www.dailyfreecode.com/forum/running-java-web-page-23432.aspx>  
<https://jsoup.org/cookbook/modifying-data/set-html>  
<http://stanfordnlp.github.io/CoreNLP/>  
<http://www.square-bear.co.uk/aiml/>  
<http://www.mitsuku.com/>  
<https://tips4java.wordpress.com/2008/11/08/message-console/>  
[http://www.tutorialspoint.com/jsp/jsp\\_standard\\_tag\\_library.htm](http://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm)  
<https://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html>  
[http://www.tutorialspoint.com/java/util/properties\\_propertynames.htm](http://www.tutorialspoint.com/java/util/properties_propertynames.htm)  
[http://www.tutorialspoint.com/java/util/hashmap\\_get.htm](http://www.tutorialspoint.com/java/util/hashmap_get.htm)  
[http://www.tutorialspoint.com/java/util/hashmap\\_remove.htm](http://www.tutorialspoint.com/java/util/hashmap_remove.htm)  
<http://www.dreamincode.net/forums/topic/315450-html-format-inside-a-servlet-how-to-print-string-variable/>  
<http://www.textfixer.com/tutorials/html-search-box.php>  
<http://www.coderanch.com/t/595474/GUI/java/put-spaces-buttons-Grid-Layout>  
<http://www.javacreed.com/how-to-modify-a-string-without-creating-a-new-one/>  
<http://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html>  
[https://www.chatbots.org/ai\\_zone/viewthread/492/](https://www.chatbots.org/ai_zone/viewthread/492/)  
<http://beginnersbook.com/2014/08/how-to-copy-one-hashmap-content-to-another-hashmap/>  
<http://stanfordnlp.github.io/CoreNLP/faq.html>