Ubuntu Weather Application

Rishi Dua, 2010EE50557

27 January, 2014

1 WEATHER APPLICATION

1.1 PROBLEM STATEMENT

Develop an ubuntu app which can accomplish following two tasks:

1. Forecasting next few days weather conditions

2. Keep record of past few days' weather conditions (max 30 days) in to your database and show it as a strip chart which has a rewind / fast fwd button.

Use tcl/tk for the GUI to interact with the user and python for other operations. The user will be asked to enter name of a place for which weather conditions have to be forecasted, the number of days (n) for which to show information, and to press a button to decide whether to forecast or tell about the past n days weather conditions. Use MySQL as database. Max 5 days' information need to be recorded. Make use of yahoo api for weather forecasting info. Develop html documentation using Doxygen. Give user an option to provide details of the place either by specifying name of that place or manually entering geographical location (lat/ long).

1.2 ABSTRACT

Tcl (Tool Command Language) is a very powerful but easy to learn dynamic programming language, suitable for a very wide range of uses, including web and desktop applications, networking, administration, testing and many more. Open source and business-friendly, Tcl is a mature yet evolving language that is truly cross platform, easily deployed and highly extensible.

Tk is a graphical user interface toolkit that takes developing desktop applications to a higher level than conventional approaches. Tk is the standard GUI not only for Tcl, but for many other dynamic languages, and can produce rich, native applications that run unchanged across Windows, Mac OS X, Linux and more

1.3 SPECIFICATION AND ASSUMPTIONS

Tool Specifications:
Language used: Tcl
Platform: Ubuntu 12.04
Additional tools used: python
Bash Version: GNU bash, version 4.2.25(1)-release (x86_64-pc-linux-gnu)
TK Version: 8.5.12
API used: http://query.yahooapis.com/v1/public/
MySQL: Used from the package lamp. The SQL dump is attached with the code
DoxyGen: This was used to generate the code documentation

Assumptions
Yahoo API provides weather report for all woeid (cities)

Problem specifications Here is a list of all namespaces:
cleanup
getweather
woeidfromcordi
woeidfromquery
writefutureweather
writepastweather

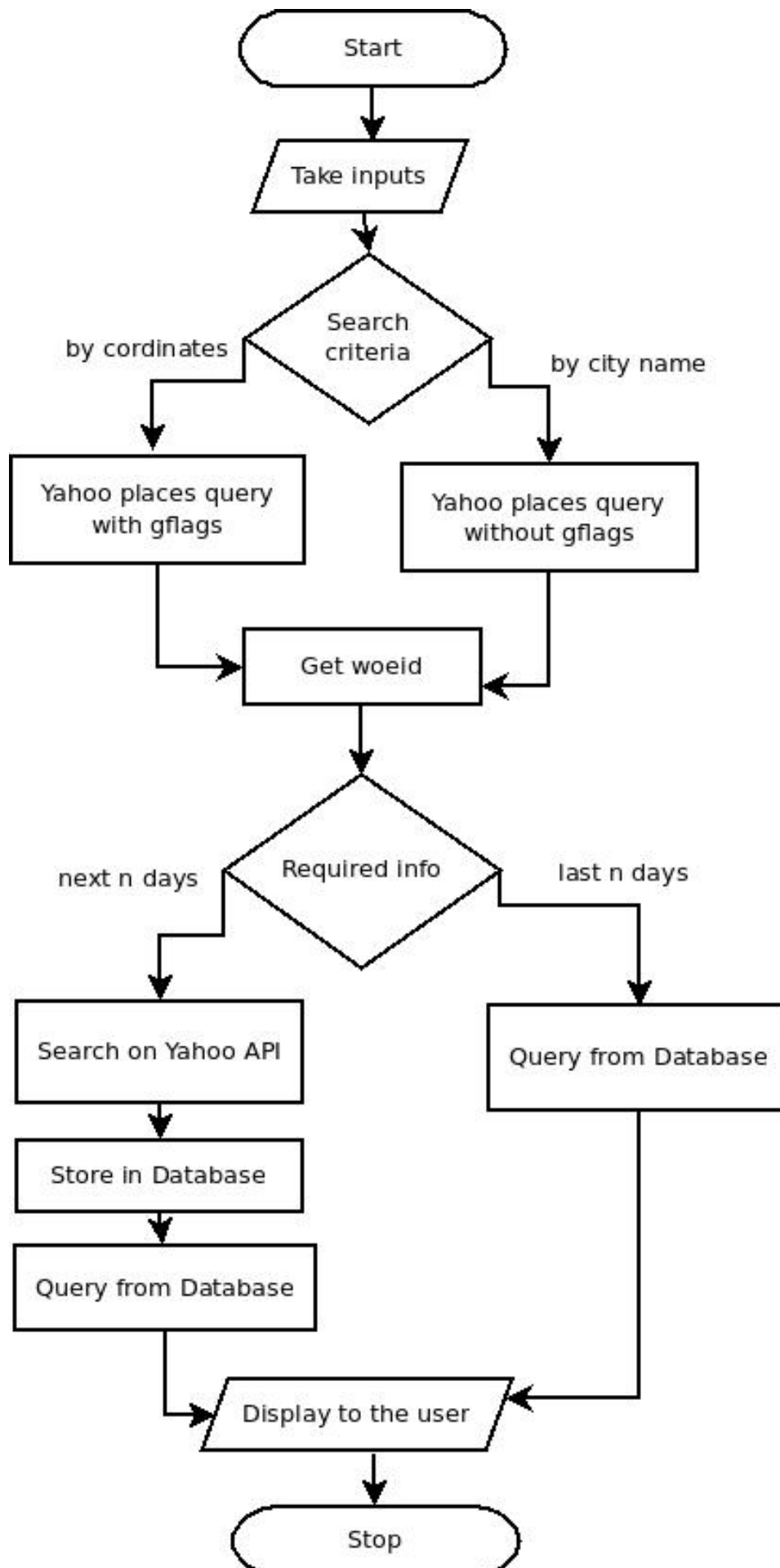Here is a list of all namespace members: cleandb() : cleanup
getwoeid() : woeidfromcordi , woeidfromquery
parseweather() : getweather
showweather() : writefutureweather , writepastweather
yahootosql() : getweather , writepastweather , writefutureweather , woeidfromcordi

```
                        ┌──────────────┐
                        │    Start     │
                        └──────┬───────┘
                               │
                        ┌──────▼───────┐
                        │  Take inputs │
                        └──────┬───────┘
                               │
                          ◇ Search ◇
              by cordinates ◇ criteria ◇ by city name
                     ┌─────────┘   └─────────┐
                     │                       │
         ┌───────────▼─────────┐   ┌─────────▼──────────┐
         │  Yahoo places query │   │ Yahoo places query │
         │     with gflags     │   │   without gflags   │
         └───────────┬─────────┘   └─────────┬──────────┘
                     │      ┌──────────┐     │
                     └─────▶│ Get woeid│◀────┘
                            └────┬─────┘
                                 │
                          ◇ Required info ◇
          next n days  ◇               ◇  last n days
             ┌─────────┘                 └─────────┐
             │                                     │
   ┌─────────▼────────┐                 ┌──────────▼─────────┐
   │ Search on Yahoo  │                 │ Query from Database│
   │       API        │                 └──────────┬─────────┘
   └─────────┬────────┘                            │
   ┌─────────▼────────┐                            │
   │ Store in Database│                            │
   └─────────┬────────┘                            │
   ┌─────────▼────────┐                            │
   │ Query from       │                            │
   │    Database      │                            │
   └─────────┬────────┘                            │
             └──────┐      ┌──────────────┐        │
                    └─────▶│ Display to   │◀───────┘
                           │  the user    │
                           └──────┬───────┘
                                  │
                           ┌──────▼───────┐
                           │     Stop     │
                           └──────────────┘
```

3

The flowchart

for cron job is
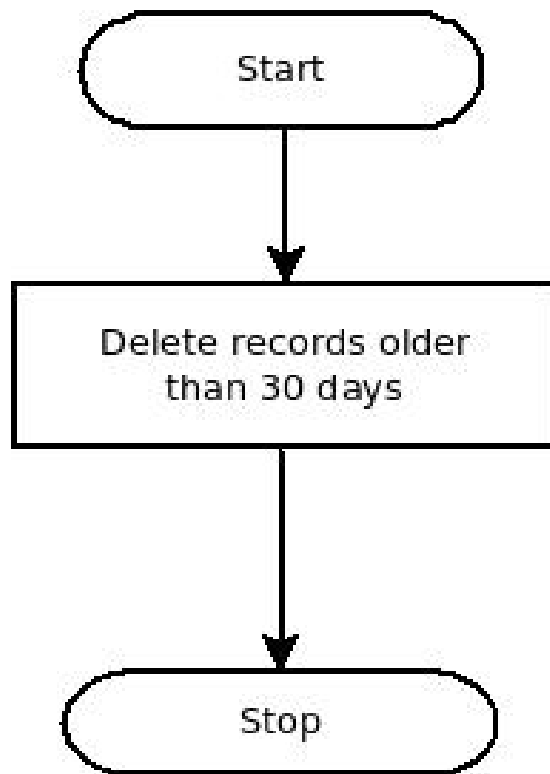
1.5 Logic Implementation

The problem is broken into 3 parts
1. Getting woeid
APIs used to get woeid:

Searching by city name:
http://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20geo.places%20where%20text%3D%22Place%20name%22&format=xml

Searching by cordinates:
http://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20geo.placefinder%20where%20text="37.416275,-122.025092"%20and%20gflags="R"&format=xml

Example YQL Queries for the same:
select city, woeid from geo.placefinder where text="delhi"
select city, woeid from geo.placefinder where text="28.7,77.2" and gflags="R"

2. Getting weather
The user entered values are parsed and passed to the python as shell arguments
Tcl file calls the python code which uses the Yahoo weather API and stores the results in the database
This result is fetched by tcl and shown on the user interface
Exceptions are caught by python file whenever there is no internet connection

3. Cleaning database for records older than 30 days The software utility cron is a time-based job scheduler in Unix-like computer operating systems. People who set up and maintain software environments use cron to schedule jobs (commands or shell scripts) to run periodically at fixed times, dates, or intervals. It typically automates system maintenance or administration. Using a small python script, all records are removed by the cron job.
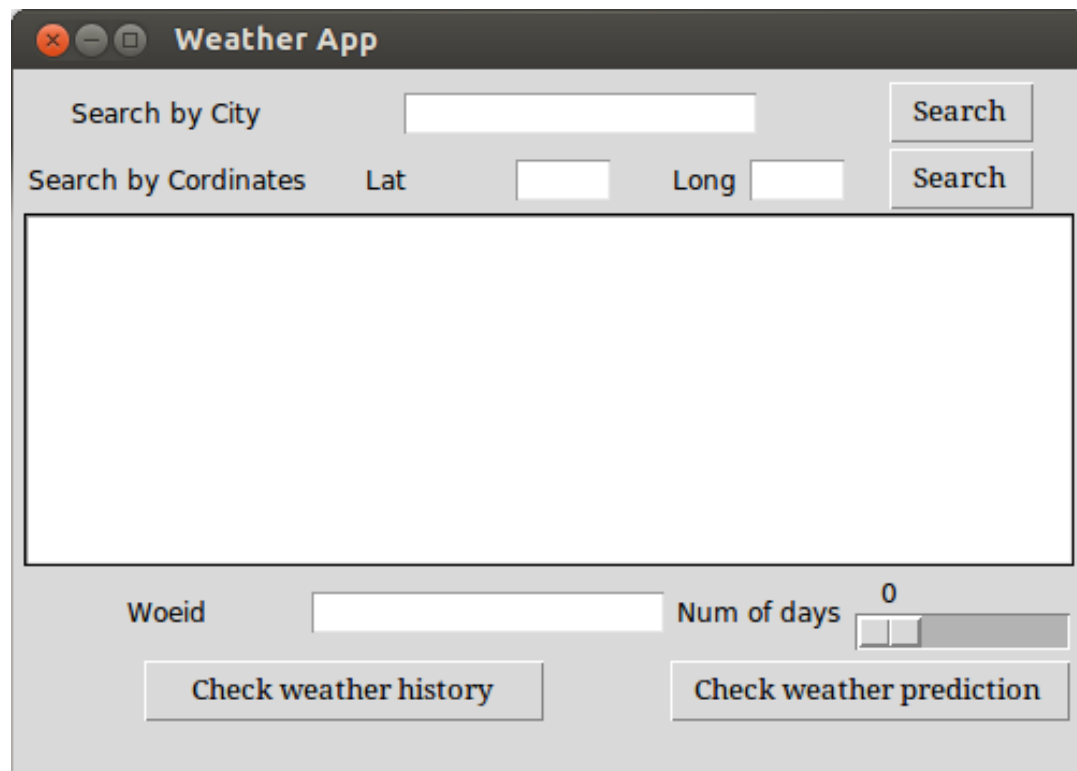
1.6 EXECUTION DIRECTIVE

No compilation required
Directly run by typing
./main.tcl

1.7 OUTPUT OF THE PROGRAM

Weather App

Search by City `[                    ]` **Search**

Search by Cordinates Lat `[28.61]` Long `[77.2]` **Search**

New Delhi, Delhi, India

Woeid `[29229065]` Num of days 0 `[slider]`

Check weather history **Check weather prediction**

Weather App

Search by City `[delhi]` **Search**

Search by Cordinates Lat `[    ]` Long `[    ]` **Search**

Delhi, California, United States
Delhi, Louisiana, United States
Delhi, New York, United States
Delhi, Iowa, United States
Delhi, Minnesota, United States

Woeid `[                ]` Num of days 0 `[slider]`

Check weather history **Check weather prediction**

1.8 RESULT

A weather application is developed using the following:
MySQL: Used from the package lamp. The SQL dump is attached with the code
DoxyGen: This was used to generate the code documentation

Problems encountered:
Testing internet connectivity
Solution: The returned file is parsed and exception is caught if the return size is 0

1.9 CONCLUSION

Successfully developed a code that gets weather from Yahoo API, stores it into a database and
displays to the user.