

---

## Vishal Gupta

Chennai, India

[py-ranoid.github.io/Resume](https://py-ranoid.github.io/Resume)

[www.linkedin.com/in/vishalg8897](https://www.linkedin.com/in/vishalg8897)

# Click To Dial Popup Window for the Linux Desktop

19<sup>th</sup> March 2018

**Confirmed Mentor** : Keith Larrimore, Thomas Levine

## OVERVIEW

As someone who has clicked on countless tel: links in vain, I would really like to create an application (invoked as a MIME handler) with a GUI that can place a phone call, display details about the country in which the phone number is registered, save the number to an address book and hopefully trigger an intent on my Android phone that I could use to call.

I have also initiated work on the project here : [Hello-from-the-Debian-side](#)

Project Link : [Click To Dial Popup Window For Linux Desktop](#)

## GOALS

1. Configure the application as the handler for tel: URLs with **xdg-mime** and parse phone numbers
2. Get country, timezone and carrier details of phone number
3. Add contact to Address Book or Google Contacts
4. Make a GUI to
  - a. Dial a number and place a call with a “Call” button
  - b. Display country details
5. Set up User Configuration to choose program that will be used to place the call.
6. Document project and create a Debian Package
7. Place the call
  - a. Using [sipDial](#)
  - b. Using [pjsua](#)
  - c. Initiating a intent on a connected Android Phone by modifying [kde-connect](#)  
(If call is placed on an Android Phone, maybe try to stream the call to the Debian desktop so that the user doesn't have to pick up his phone to place the call)

---

## ABOUT ME

Hello world. I'm Vishal, a 3rd yr CSE undergrad from Chennai, India. While most people generally pick up a topic, or a concept (like say Computer Vision, Big Data, or just Algorithms), understand it and aspire to excel at it... I fell in love with a language, Python. As someone who has started out by learning C++ in school, learning Python was as easy as surprising. The speed at which I could translate ideas to code was amazing, and oh boy, all I wanted to do was make things, write simple scripts to automate everyday tasks. And hence I continued to explore Python, the countless modules and possibilities with Python. I went to Hackathons, won some but more importantly made something that others could use. And I guess that's always been my mantra for programming and my way to "help the world" : Make cool stuff.

Since awesomeness is universal and often found in Github repos, I dived into open-source, an ocean of cool stuff that could potentially be made cooler.

### Projects I've done

- Used Django, Celery and other Python modules to develop this : [www.pygeon.tech](http://www.pygeon.tech)
  - Hackathons and Developer events are hard to keep track of
  - Used celery to scrape events from some sites every day
  - Used Messenger as a medium to keep subscribers updated with new events as and when they're found
  - Used Docker to deploy
- Developed a chatbot for a Startup, which was deployed with Django on an EC2 instance
- Made an Android applications to convert algorithms in Natural Language and Flowchart images to Code (Python/C++/Perl)
- An application to generate MUN placards and ID cards using PIL (Python Image Lib.)
- Taught 45 freshers in my college, the basics of Flask, application, context and REST API
- ... complete list here : <https://py-ranoid.github.io/Resume/>

### Open Source contributions

- Fixed bugs and added documentation to <https://github.com/HTTP-APIs/hydrus>
- Added a feature to kdeconnect to let users zip a folder and send it to their mobile devices
- Added common graph algorithms to <https://github.com/TheAlgorithms/Python>

### Other Details about me :

Country of Residence : India

Location : Chennai, India

Timezone : IST (UTC + 05:30)

Graduation-date: 2021

Github username : [py-ranoid](https://github.com/py-ranoid)

Salsa username : [comfortablydumb-guest](#)

---

## TOOLS

**Communication** : Email preferably. [vishalg8897@gmail.com](mailto:vishalg8897@gmail.com)

**Editor** : Atom + Hydrogen or jupyter-notebook

**OS** : Ubuntu 16.04

## MILESTONES

### Week 1 : MIME processing

Create script to be parsed on clicking on tel: URIs with **xdg-mime**.

Parse phone number from URI

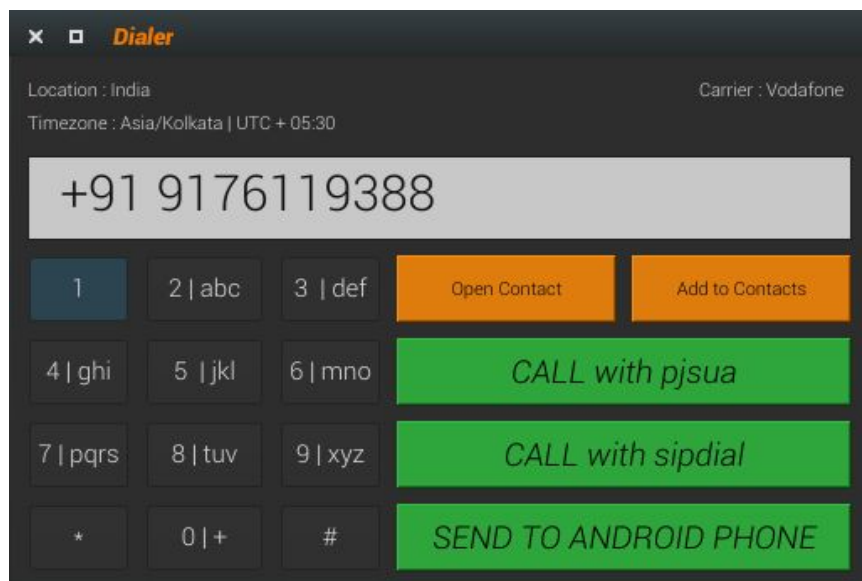
Document work

### Week 2 : GUI work

Improve GUI and add necessary components based on goals and feedback from mentors and community. Document work and instructions on how to modify GUI.

GUI will be able to

1. Display Phone number
2. Display a dialer
3. Dialer country name & flag, timezone and carrier.
4. Button to add to address book.
5. Button to send a text (optional)



---

## Week 3 : Get country details

Use libphonenumber to get the country corresponding to a phone number.

Additionally, classify the number into *Fixed-line*, *Mobile*, *Toll-free*, *Premium Rate*, *Shared Cost*, *VoIP*, *Personal Numbers*, *UAN*, *Pager*, and *Voicemail*.

Display name in the pop-up window along with country code and flag (using [iso3166](#))

Document work to display more details about country in the future

## Week 4 & 5 : Use pjsua and execvp to place call

Write a script to launch pjsua with execvp to place a phone call.

Document the configuration file so that others may repeat my work

## Week 6 : Contact addition

Do one or more of the following

1. Maintain an address book within the dialer application using a JSON file
2. Add option to integrate Google Contacts using [API](#)
3. Add contact to Thunderbird or other application to maintain Address book

Document the process so people can contribute to more methods of contact addition.

## Week 7-8: Tweaking [kde-connect](#) to place a call on Android device

KDE-connect is a Linux + Android application that syncs a desktop with an Android device if they are on the same local network. Currently, it only supports File sending and notification synchronisation (primarily). I would like to tweak it in order to raise an intent on the Android device so that the user can call the number from his Android device instead of his laptop.

Document work.

## Week 9: User configuration and documentation

Modify the pop-up window program so that phone calls are placed with a user-configurable command specified by the environment variable "TELEPHONE".

Begin writing a manual page for the pop-up window program, by converting my existing documentation into mdoc format.

---

## Week 10: User configuration

If the "TELEPHONE" environment variable is specified, display the name of the program that will be used to place the call. If the "TELEPHONE" variable is not specified, put a button in the pop-up window that says "Choose application". When you click that button, you get another pop-up window where you can choose the application.

Revise the manual page.

## Week 11: Documentation and installation script

Finalize the manual page and patch up any bugs in application.

Write a script that installs the pop-up program and the manual page into directories under the PREFIX environment variable. Distribute it to at least three people other than my mentor for testing.

## Week 12: Packaging

Write a Debian package of the program.

Distribute the Debian package to at least three people other than my mentor for testing.

## OTHER COMMITMENTS

Semester Exams : April 27th to May 15th

## DEBIAN <3

This is an opportunity of a lifetime to be able to contribute to Debian as a GSoC intern over the summer. Like most developers, Debian has been, is, and will to continue to be my daily OS and work environment. This is a chance for me to give back to the Debian by making something that will increase productivity and make full use to **tel:** links to place calls, add to contacts and get additional details about the number. Last but not the least, I have heard a lot about DebConf would love to take this chance to attend DebConf18.

Cheers,  
Vishal Gupta