

# RDBMS lab project report

On

# Intelligence database

By, Group no. 4

## Members:

Abhishek Mohite.	E-mail : <a href="mailto:b518004@iiit-bh.ac.in">b518004@iiit-bh.ac.in</a>
Sambit Sahoo .	E-mail : <a href="mailto:b518039@iiit-bh.ac.in">b518039@iiit-bh.ac.in</a>
Suraj Nayak.	E-mail : <a href="mailto:b518057@iiit-bh.ac.in">b518057@iiit-bh.ac.in</a>
Aayushman Lath.	E-mail : <a href="mailto:b518002@iiit-bh.ac.in">b518002@iiit-bh.ac.in</a>
Raman Bankra.	E-mail : <a href="mailto:b518038@iiit-bh.ac.in">b518038@iiit-bh.ac.in</a>

[our software is easily usable, steps to use are given on last page]

## Contents:

1. [Abstract](#)
2. [Introduction](#)
3. [Detailed record](#)
4. [Resulting software](#)
5. [Source code](#)
6. [Conclusion and thanks](#)
7. [Let's build](#)

# **Abstract**

So basically to maintain the incoming data in the intelligence agencies such as raw in India we need a secure and offline database and a user interface interface to work with it i.e. view the information using various tools such as searching all people who were arrested in the month on June 2020. So here is presented a prototype for implementation of such a system with mysql database along with the UI written using tkinter library of python and backend with mysql connector python library.

# **Introduction**

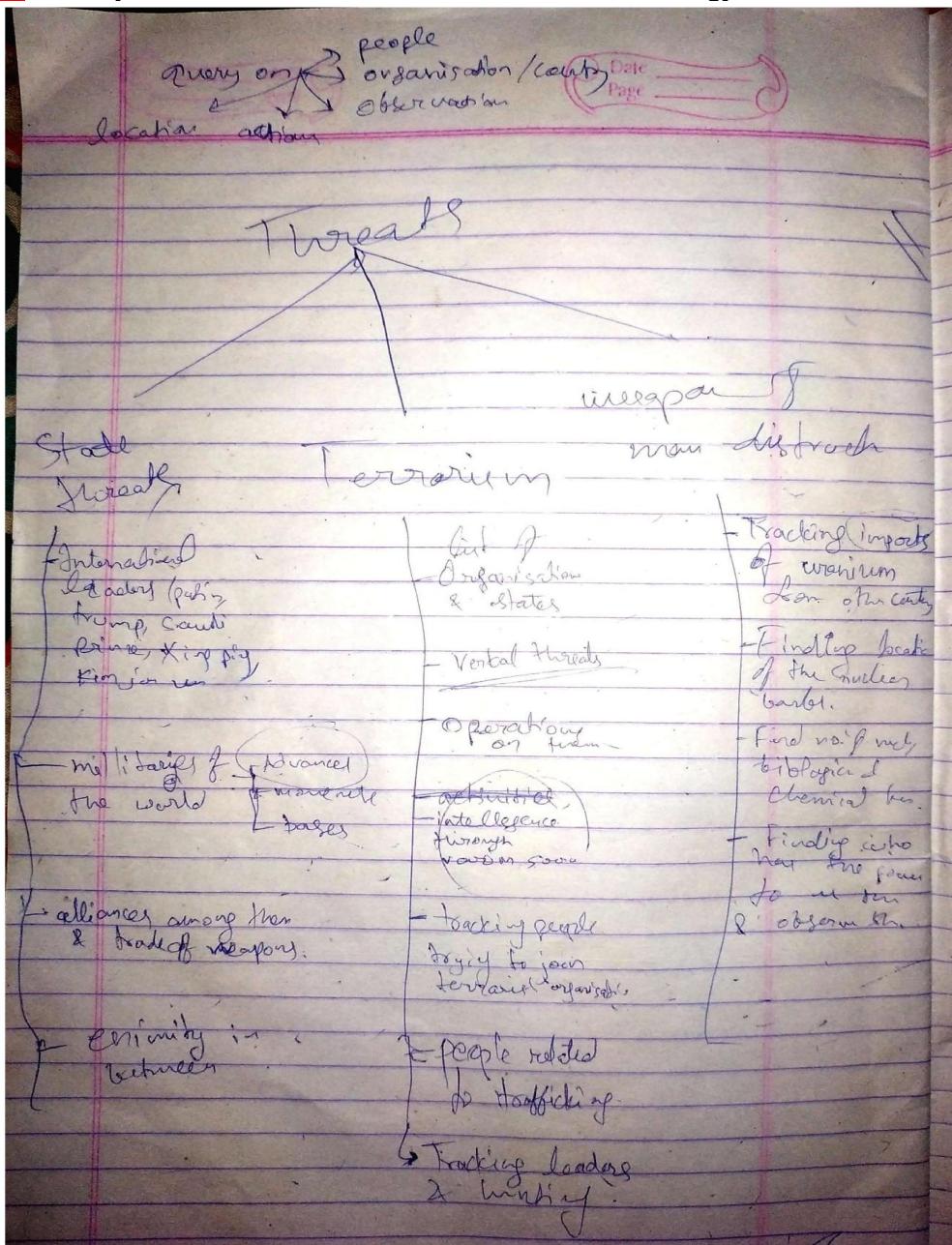
The steps followed in making the Intel database:

1. Identify the data required such as list of people's names, addresses, dob, list of organizations etc. and create table formats for them.
2. Identify relations between those tables if any and make an er diagram using this data.
3. Try to remove any redundancy and normalize the er diagram.
4. Design an user interface prototype on paper/ paint program on computer to finalize the UI.
5. Design the tables in mysql and decide foreign keys according to er diagram.
6. Design the UI using tkinter in python and then connect them using mysql connector.

# Detailed record:

1. Initially scribbled requirements and tables  
(pardon the bad handwriting).

[Note: this is just to illustrate about our workflow, not for reading]



Scanned by TapScanner

here the three branches represent the different concerns about which we need to maintain data.

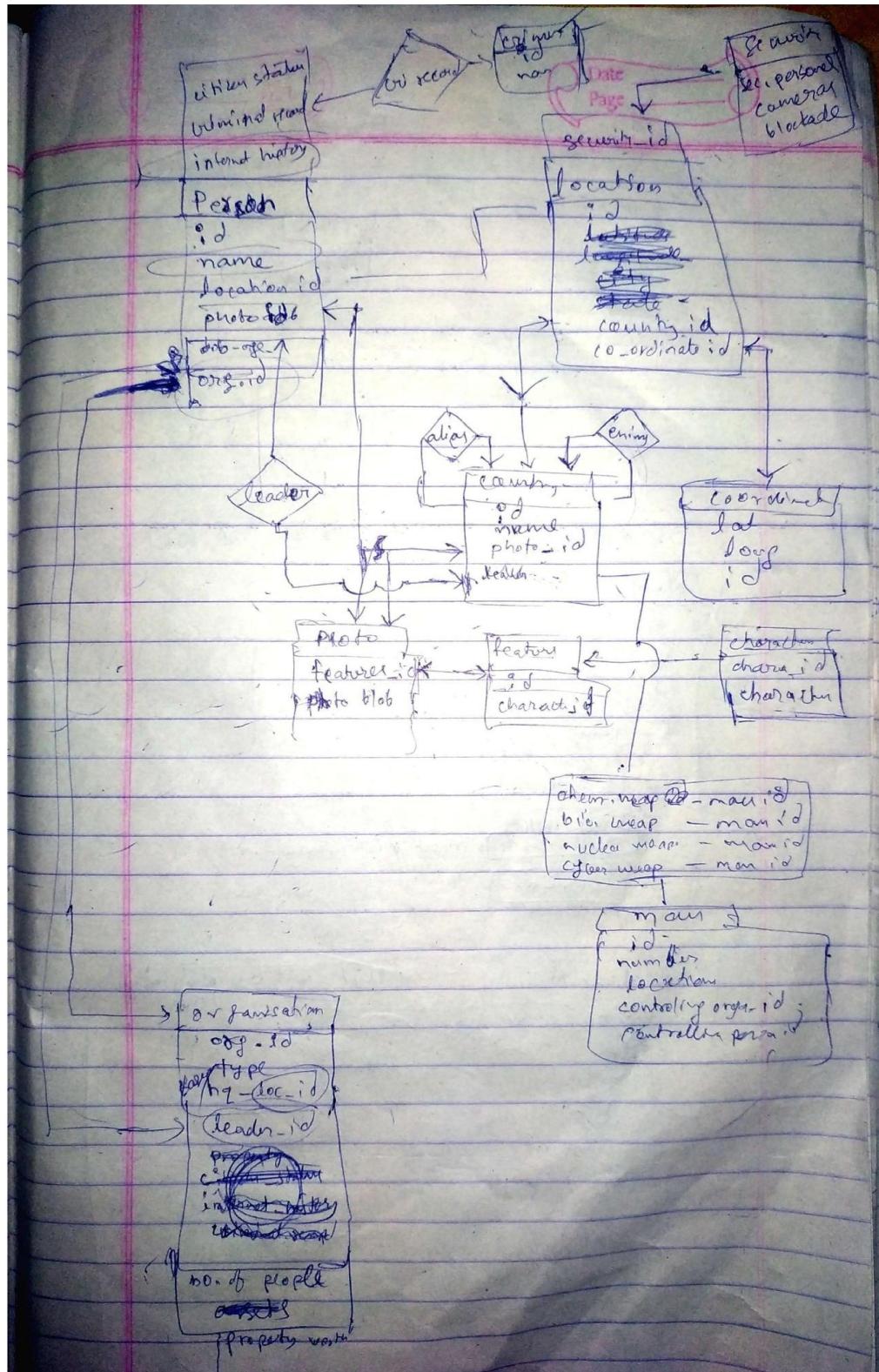
[Note: this is just to illustrate about our workflow, not for reading]

table - country, location, people, organisation
relation → operation, observation, relationship
table ✓ observation (audio, visual, location, on-the-premises)
activity (activity)
✓ action (arrest, kill, kidnap, black market, terrorist, bribe), (bombing, surveillance, drill, traffic jam, decision, imposed meeting)
✓ organisation (type (military, naval, terrorist, corporate, political))
✓ site - observation (strength, media share, movement, density, threat, activity, advantage)
✓ location (type (area, special point (e.g. forward plan)), location, street address, city, country, defence id)
✓ defence (defence id, border (trap, barrier, min.))
✓ place - des - wegr (nuclear id, chemical id, biological id)
✓ people (name, profession, age, permanent, felony, status)
✓ country (id, name, capital, currency, allies, leader)
✓ special intelligence (source (people - id), observation, location, target (target), frequency)
✗ goods (guns, uranium, drugs, human, valuable)

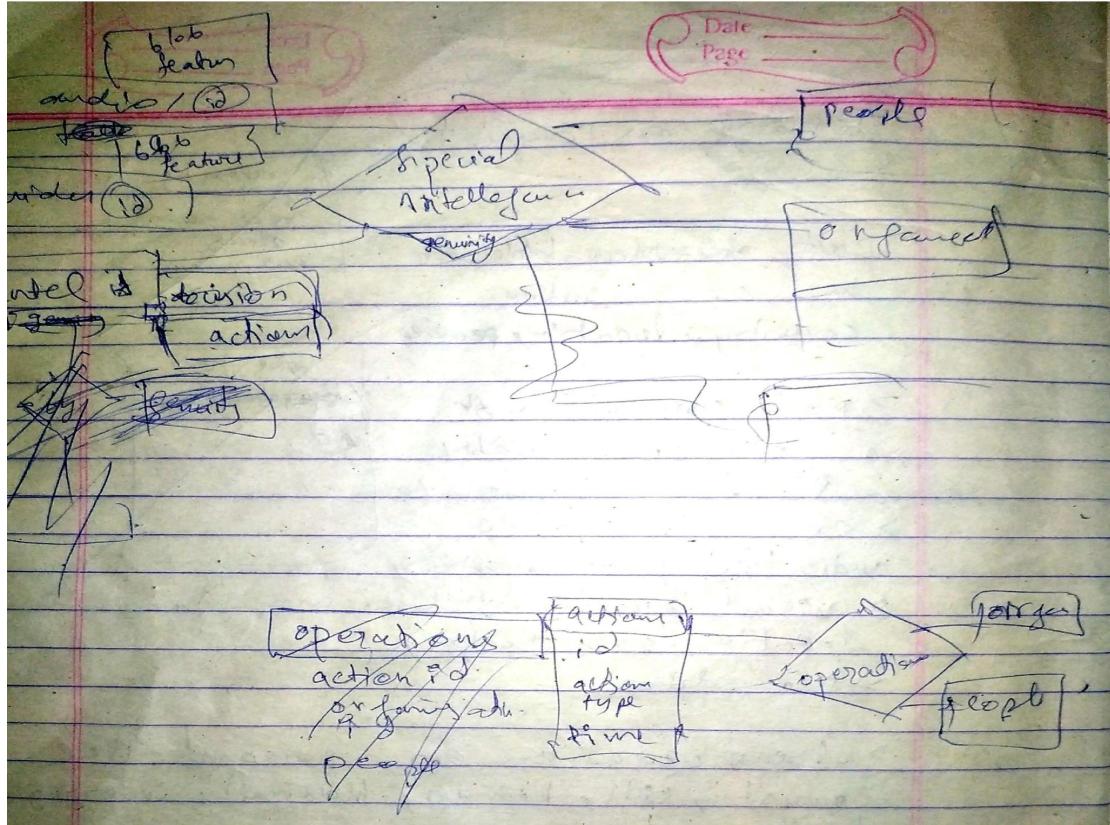
Here the ticks represent table name and brackets contain column names.

## 2. Raw and final er diagram (again pardon the bad handwriting):

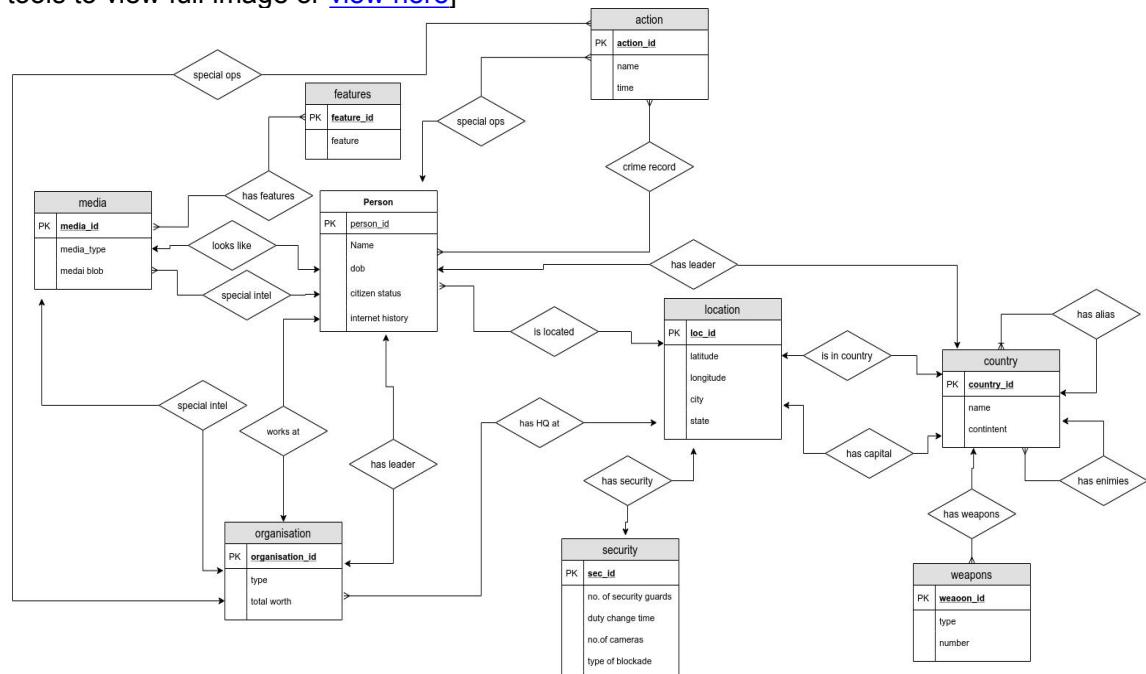
[Note: it is just for illustrating work-flow, better designed er diagram is also provided below]



[Note: don't try to read these as it is just for illustrating our work-flow, an er diagram designed with computer is provided below]



[Note: this is **final er diagram**, please click on use picture preview tool in picture tools to view full image or [view here](#)]

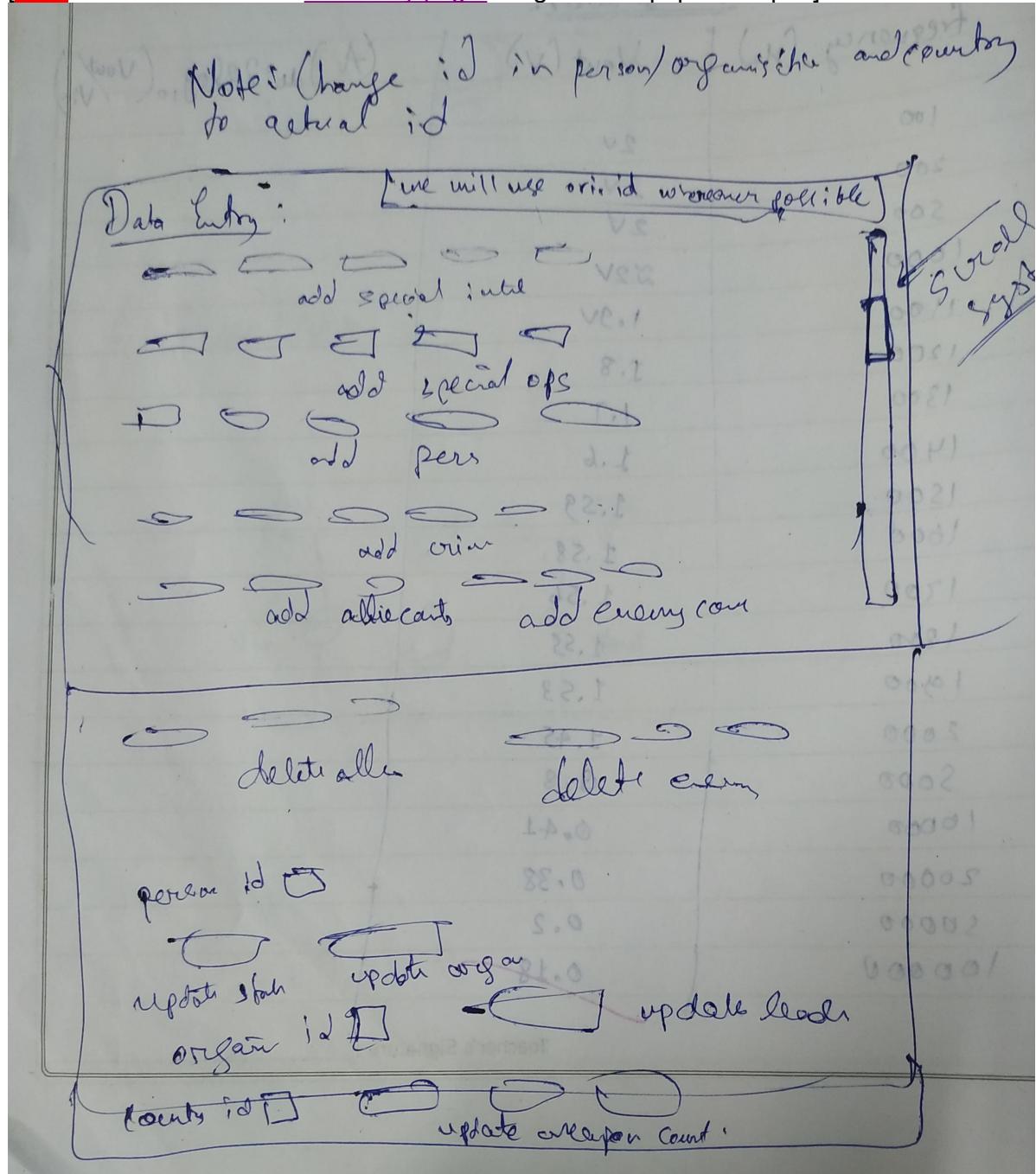


### 3. Removing redundancy

The database was designed according to BCNF moralization and hence we tried to minimize the redundancy.

### 4. Deciding final look of UI:

[Note: this is illustration of data entry page designed with paper and pen]



## 5. Coding mysql database

- I. The database was written in mysql with a little help of HeidiSQL software on a manjaro linux machine.
- II. Views were used at some places instead of relationships to reduce the space complexities while compromising a bit on the time complexity.
- III. This decision was made to reduce the complexity of code and the effort required to write it.
- IV. Much more research was required with the help of stackoverflow for creating a working database.

## 6. Designing the UI

- I. The UI was designed copying the initial paper based design into the code.
- II. An initial login screen takes to different pages for either for analyst or for data-entry according to the credentials.
- III. Treeview was used for providing scrollable tables.
- IV. The whole data entry was designed on the scrollable Frame inside tk, carefully fitting each text field, button and labels into the row-column based architecture.

# Resulting software:

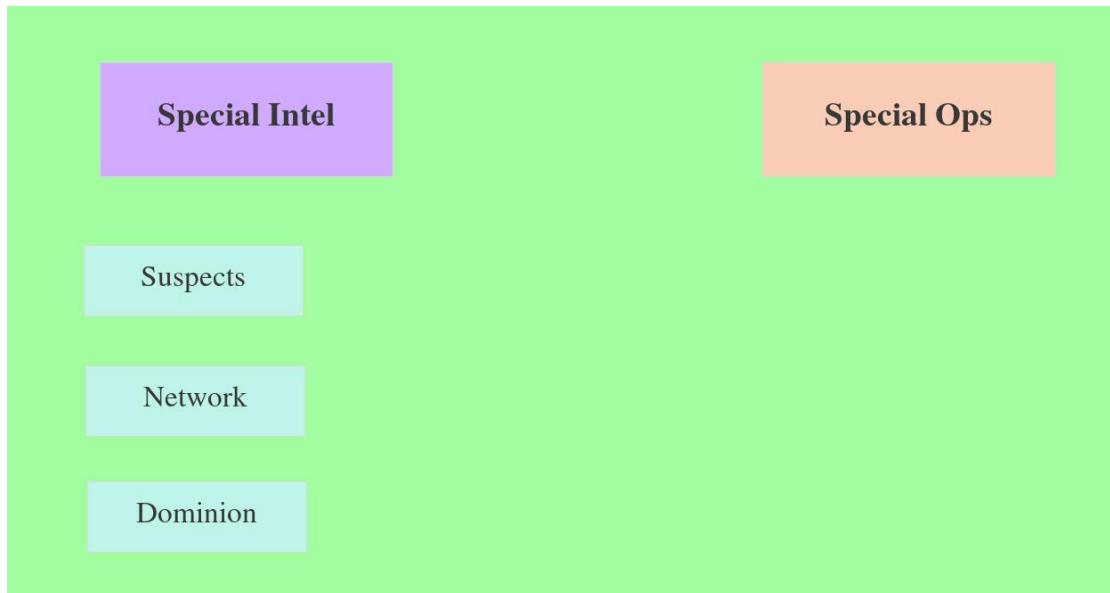
Following pages were formed as a result of dedicated coding in python:



Login screen

A screenshot of a data entry application with four main sections: Intel, Ops, Person, and Crime.   
**Intel:** Contains fields for 'person id/organ id', 'about the intel', 'date in YYYY-MM-DD', and 'address of intel'. It includes two blue buttons: 'insert by person id' and 'insert by organ id'.  
**Ops:** Contains fields for 'person id/organ id', 'about the operation', 'date in YYYY-MM-DD', and 'insert by person id' / 'insert by organ id'.  
**Person:** Contains fields for 'name', '1900-01-01', 'f--', '0', '0', and '/demo.png'. It includes an 'insert' button and a 'pic' button.  
**Crime:** Contains fields for 'person id', 'date in YYYY-MM-DD', and 'crime details'. It includes an 'insert' button.

Data entry as depicted in the paper pen prototype with a few changes.



Analyst screen to go to various tabs suspects=people, network=organization, dominion=country

The screenshot shows the "Suspects" tab. At the top, there is a search bar and a "filter organisation" button. Below the search bar is a "search" button. The main area contains a table with the following data:

no	id	name	organisation	country	soc/eco status
1	56	someone unknown	unknown	india	D -
2	78	shah rule	some company	unknown	D ++
3	4	kartik pandya	unknown	unknown	F -
4	0	unknown	unknown	india	F -
5	81	name	unknown	unknown	F -

A general view inside all of these tabs with some things different here and there.

The screenshot shows a web application interface. At the top left, there is a green header bar with the text "suspect\_detail". Below this, there is a table with several rows, each containing a label and a value. The labels are: Name, Dob, Organisation, Status, City, State, Country, and Crime record. The values are: someone unknown, (redacted), unknown, D --, delhi, delhi, india, and 1. To the right of the table is a portrait photo of a woman with short brown hair. Below the photo is a small caption that reads "V. J. De Faria". To the right of the photo is a table with one row, showing the number "1" and the crime "murder by knife".

no.	crime
1	murder by knife

A more detailed view selected from the list columns above

## Source Code:

The following link shows the mysql code:

<https://drive.google.com/file/d/1nxNaL2KYfsitpIXLW6zo1IYQMhx39VAc/view?usp=sharing>

The following link shows the python(UI + backend) code:

[https://drive.google.com/file/d/13JCz88M7RLeW\\_JuEfxaECNny4MvQmlW2/view?usp=sharing](https://drive.google.com/file/d/13JCz88M7RLeW_JuEfxaECNny4MvQmlW2/view?usp=sharing)

## Conclusion and thanks

It was a team's effort to create the fully working project. Though has somewhat gone to waste because we can not present in front of the class and brag our creation. But still this experience will go with us all the way till our graduation. Thanks to you we got this awesome opportunity.

# Let's build

Run this .py file with internet turned on, **after** reading below steps :

[https://drive.google.com/file/d/1FITJ\\_uL8k-yzM3rOdMdSlkZYtqJG\\_bY4/view?usp=sharing](https://drive.google.com/file/d/1FITJ_uL8k-yzM3rOdMdSlkZYtqJG_bY4/view?usp=sharing) (size 77kb)

## README:

Our software is a standalone software, which means running the python file shared above will launch the software, if you have python installed in your system along with the dependency python library and your internet is connected:

- tk or tkinter
- python-mysql-connector
- pillow
- numpy

You can install these libraries with help of pip on windows/linux/macos. (mysql connector is installed with package manager in tested arch based linux machines)

Our mysql database temporarily hosted online at <https://www.freemysqlhosting.net>, if it is expired then this won't work, so in that case use this .sql file to import the database somewhere

[https://drive.google.com/file/d/1\\_DX59FUJD8wmMUoKzv2Zx\\_X7xTg9Ty3I/view?usp=sharing](https://drive.google.com/file/d/1_DX59FUJD8wmMUoKzv2Zx_X7xTg9Ty3I/view?usp=sharing)

and then change these variables inside .py file to your own credentials.

```
18 globalhost = "sql12.freemysqlhosting.net"  
19 globaluser="sql12352363"  
20 globalpwd="qssD26v9uQ"  
21 globaldb="sql12352363"
```

