PGS - public grievance system

Problem Statement

Lack of system that lets the public to put up their grievances to the concerned department in an easy and hassle less way has been stopping the citizens to play their part in bringing the government to notice the issues on the ground.

Solution

One stop web application for all the grievances, requests and complaints for citizens, whatever is the concerned department. The system allows the citizens to track the progress of the issue. All the issues are open to the public (unless marked otherwise). The dashboard helps the citizens to analyze the performance of departments, officers and the governments.

Overview of application

The goal:

The primary goal is to introduce the enterprise practices to bring in transparency, accountability in the government and increase the involvement of the citizens in bringing the change.

The proposal:

Software companies use an issue tracking system to keep account of the bugs/lack of functionality in the existing products and to prioritize the new functionality to add to the product in the later releases. The system also helps them to assign the issue to be resolved by an employee and hold them responsible for the same. Moreover, the system is open to the public (customers) so that they can lodge any issues they come across and track its progress and that of others as well.

A similar system to track and resolve public grievances can significantly improve the transparency in the system and make the public as active citizens in the betterment of the society.

The existing system:

Public grievance systems are already in place for most of the governments. But most of the citizens are not aware of them because they are neither advertised nor are easy to use. And some of such systems do not even have a domain name and any user has to remember the IP address to use it. Also, the citizen has to be aware of the concerning department that is responsible to resolve the issue.

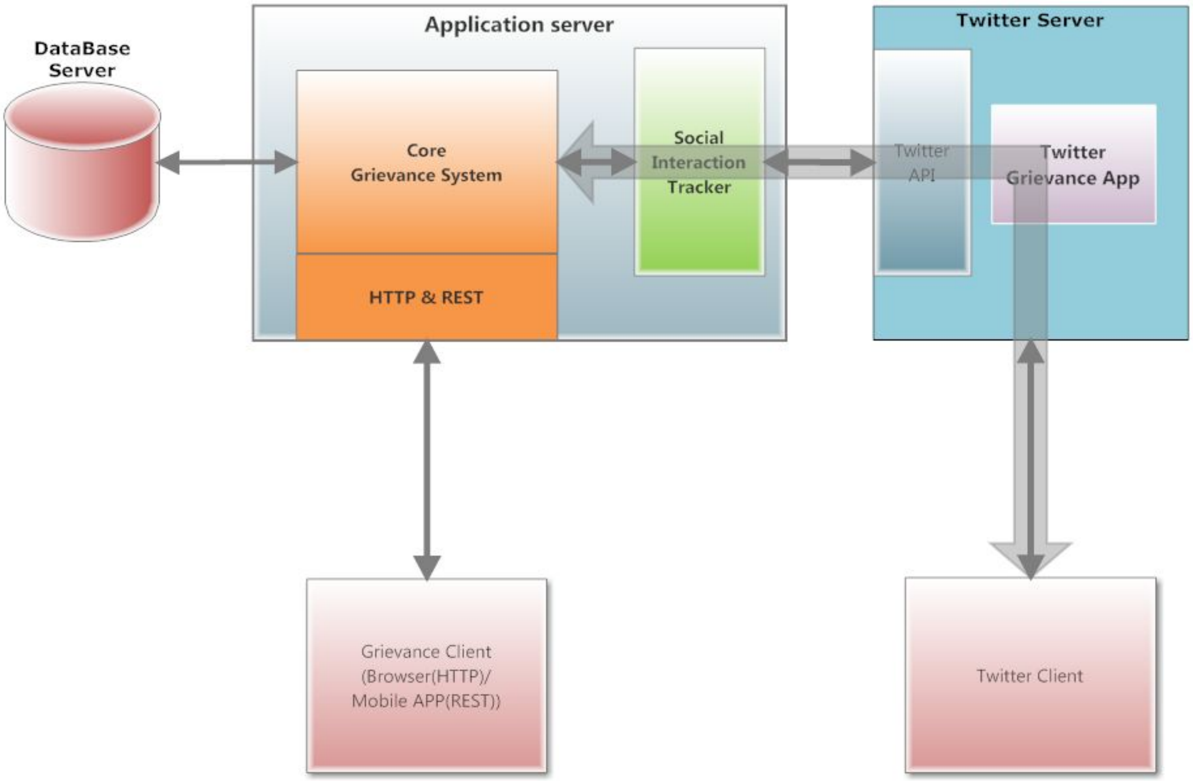
The proposed system:

One stop for all the grievances, requests and complaints for citizens, whatever is the concerned department. The system and the management are responsible to sort-out the issue to the concerning department.

The proposed system will make lodging a complaint as easy as updating your WhatsApp status or tweeting. In fact, a tweet to the right twitter handle will formally lodge a complaint in the system.

The system allows the citizens to track the progress of the issue. All the issues are open to the public (unless marked otherwise). The dashboard helps the citizens to analyze the performance of departments, officers and the governments.

Architecture



Technologies

- Python 3 on Django 1.8
- Nginx and Gunicorn HTTP servers
- PostgreSQL database
- Tweepy - Twitter OAuth client library
- NLTK - Natural Language Tool Kit for machine learning in python
- GoogleMaps - google’s Geolocation API client library
- Celery with RabbitMQ for asynchronous tasks

Internals:

Intro:

The web application runs with Django, a popular web-framework for python. The core module (modules are called apps in Django) handles the registration, updation and tracking of issues. The ‘accounts’ app (module) handles the authentication of the users.

There are two kinds of users in the system.

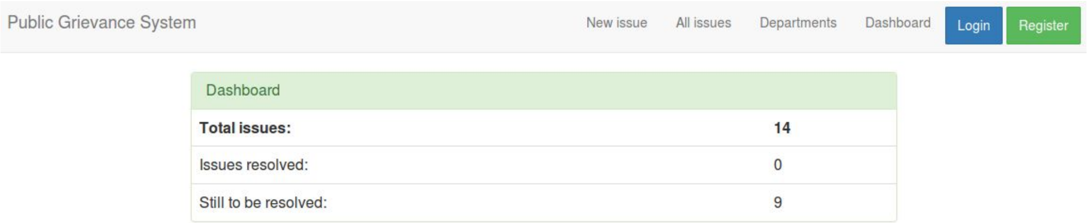
1. Normal user
2. Administrator

The normal users add their grievances to the system. Any registered user can add a grievance and any visitor (not registered) can view statuses of the issues. The administrator is responsible for categorizing the issues to appropriate departments, updating the issue statuses and marking them as resolved.

Features:.

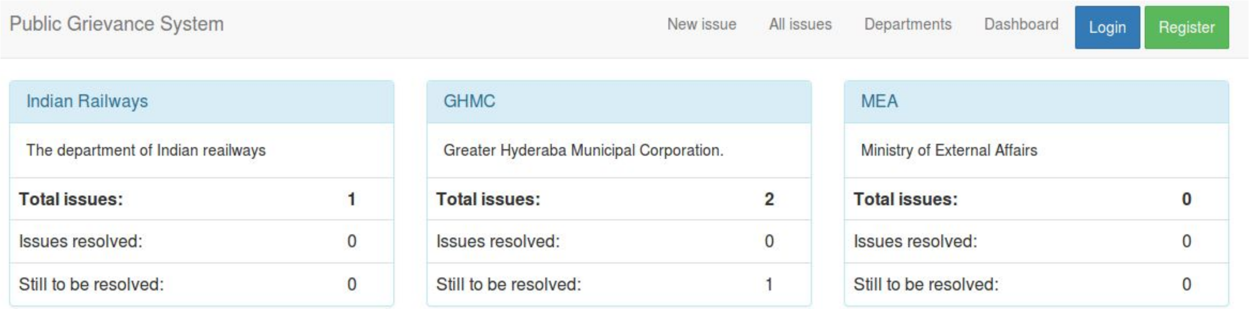
Dashboard:

The home page (landing page) shows the status of all the issues registered in the system in real-time.



Dashboard	
Total issues:	**14**
Issues resolved:	0
Still to be resolved:	9

The department wise dashboards can be seen under departments menu in the menubar.



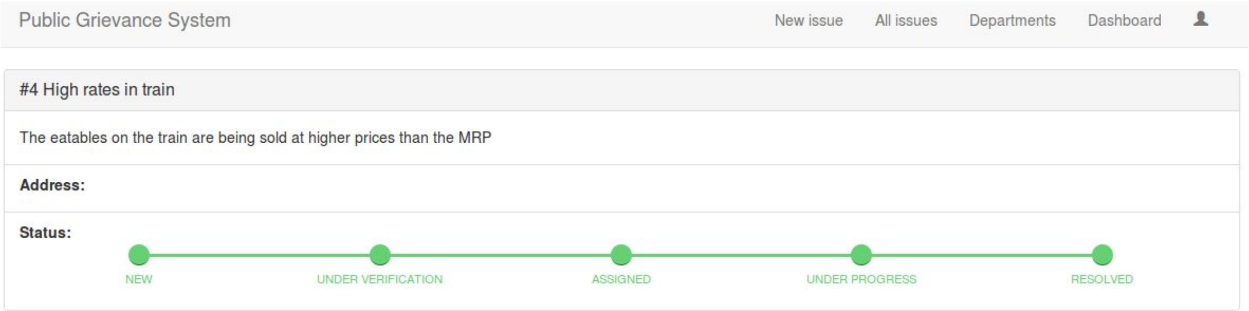
Indian Railways	
The department of Indian reailways	
Total issues:	**1**
Issues resolved:	0
Still to be resolved:	0

GHMC	
Greater Hyderaba Municipal Corporation.	
Total issues:	**2**
Issues resolved:	0
Still to be resolved:	1

MEA	
Ministry of External Affairs	
Total issues:	**0**
Issues resolved:	0
Still to be resolved:	0

Tracking issues:

The status of the issues can be seen by clicking on 'track' against the issues in the 'all issues' page. The status is shown in a graphical manner.

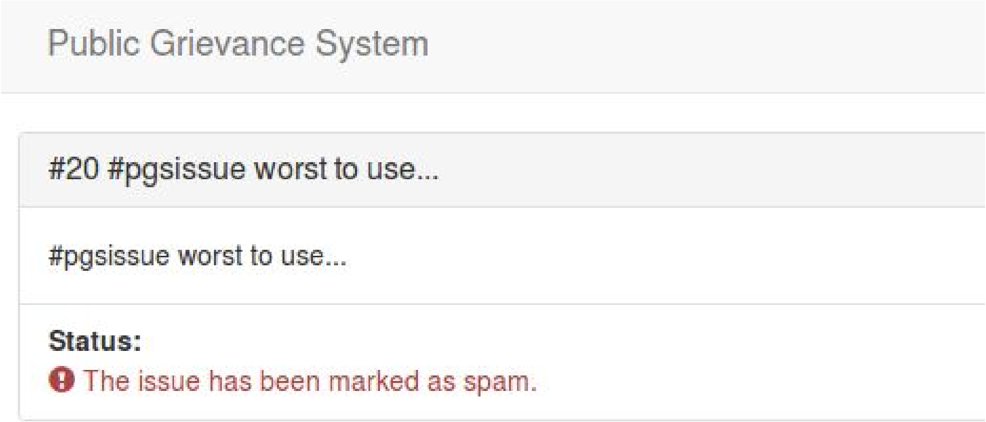


#4 High rates in train

The eatables on the train are being sold at higher prices than the MRP

Address:

Status:



#20 #pgsissue worst to use...

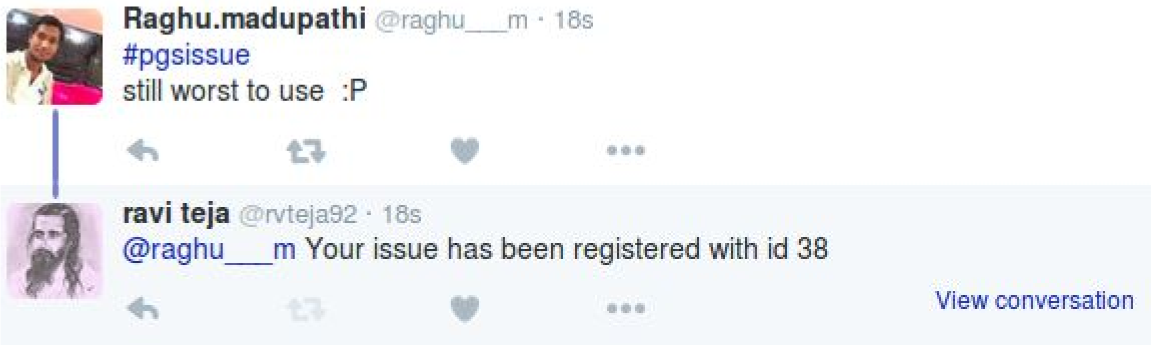
#pgsissue worst to use...

Status:

The issue has been marked as spam.

Social interaction tracker:

The issues can also be registered via twitter. Just tweet with the hashtag "#pgsissue" and you will get a reply if the tweet has been registered. (Here the app is registered with handle @rvteja92)



Raghu.madupathi @raghu___m · 18s
#pgsissue
still worst to use :P

ravi teja @rvteja92 · 18s
@raghu___m Your issue has been registered with id 38

The 'social interaction tracker' uses sentiment analysis to determine the sentiment of the status. It currently uses the "text-processing" project at text-processing.com, which in turn uses python's natural language toolkit - NTLK. So, only tweets with negative sentiment will be registered.

We plan to auto categorize i.e., assign automatically to the concerned departments, filter spam and analyze the severity of issue using text-processing and machine learning techniques.

Integrating geolocation API:

The app integrates with google-maps API to obtain geographical address of the location of issue if needed to resolve the issues. The API needs longitude and latitude values for which the browser prompts to allow to obtain location data while registering an issue. Embedding maps into registration will make it even more useful.

Asynchronous tasks with Celery:

Tasks that take time to complete can block the system to respond to the user. Such tasks or subtasks are better performed asynchronously so that the response to the user is quick. This will also let the task to be retried at a later point of time if it fails. Getting the geographic addresses using google's API is implemented as asynchronous task.

Asynchronous tasks have been handled with Celery and RabbitMQ which is a popular combination with django. Celery is an asynchronous task queues, that queues and executes tasks. It uses RabbitMQ for storing the queues persistently.

Serving HTTP requests:

The python code is interpreted by Gunicorn SGI HTTP server. While Gunicorn takes care of handling requests to the Django applications, Nginx server acts as proxy between the user requests and gunicorn server. This is a recommended approach because nginx is good at handling requests for static files and gunicorn is designed to take advantage of this fact.