

Table of Contents

| | |
|---|---|
| Table of Contents | 1 |
| ISSUE-TRACKER | 2 |
| 1. WHY | 2 |
| 2. WHAT | 2 |
| 3. SUITABILITY FOR REAL-LIFE USAGE SCENARIO | 2 |
| 3.1. Assumption and prerequisites | 2 |
| 3.2. Proposed capabilities | 2 |
| 3.3. Installation assistance | 3 |
| 4. DEMO | 3 |
| 5. INSTALLATION AND CONFIGURATION | 3 |
| 6. ADDITIONAL DOCS | 3 |
| 7. ACKNOWLEDGEMENTS | 3 |
| 8. LICENSE | 3 |

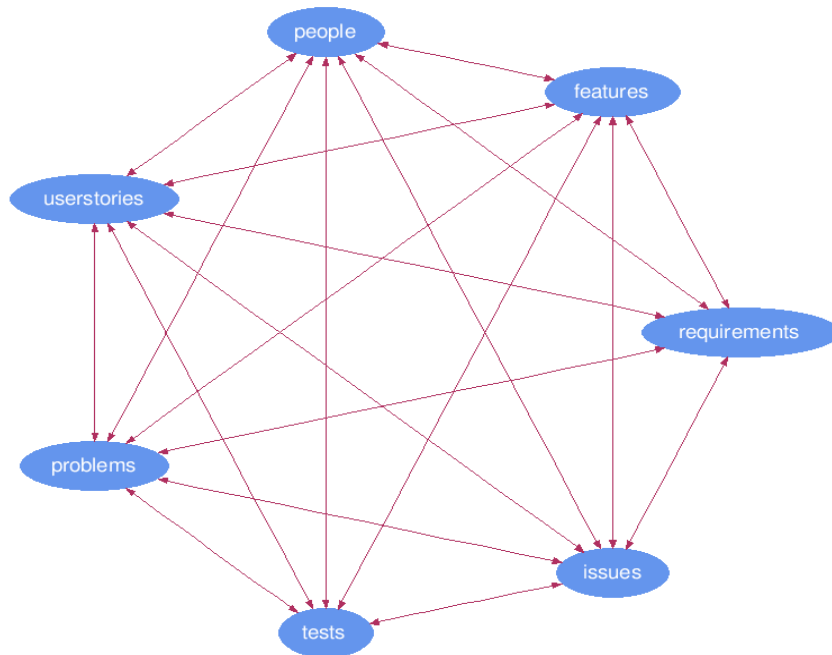
ISSUE-TRACKER

1. WHY

Why ?! Yet. Another. App ?!

Because work should be inspiring and not overwhelming people. Because even good intentions without proper commitment, allocation and resourcing and most importantly, a mean for tracking advancement of an endeavor in open way reflecting the reality, might end-up making people less happy, when in fact a really simple solution could be applied for any bigger challenge requiring progress tracking, communication and coordination ... And tons of other reasons we all having been in project disasters know about ... Still here ?! Let's move on !

Figure: 1 issue tracker objects polygon



2. WHAT

A generic postgres CRUDs (s stands for search) web based app for managing multiple databases from the same web application layer by means of simplest possible UI and/or shell tools for xls export , Google sheets writes etc. An included example application is the "issue-tracker application", which is used to manage multiple projects' issues, including itself ;o)

The full and extensive features and functionalities document could be read from: <https://github.com/YordanGeorgiev/issue-tracker/blob/master/doc/md/issue-tracker-features-and-functionalities.md>

This application is the reflection of the best practices and principles for tens of years in IT resulting into a product of the Multi-environment instance architecture and the Input-Output Controller Model architecture (more about this in the DevOps guide: <https://github.com/YordanGeorgiev/issue-tracker/blob/master/doc/md/issue-tracker-features-and-functionalities.md>

3. SUITABILITY FOR REAL-LIFE USAGE SCENARIO

We have stumbled into real life problem in the ETL integrations of a big Scandinavian telco, which could have been solved even with the current version of the tool, although the versioning starts with the mighty zero ...

Of course next versions will provide much more scenarios in real-life, but this description applies to this current version, that is you can start using it as soon as you deploy your own instance ...

3.1. Assumption and prerequisites

Your organisation:

- has the need to constantly update comparably small (less than 10k rows) hierarchy tables
- has secured intranet access to a Linux box
- has full trust to the persons in the org having http access for CRUD operations, as only basic auth over http exists
- has the a need to load MANY tables into a postgres db, which might be changing constantly DDL wise
- the API of having bigint id and uid as PK as well as default vals for nullable cols is acceptable

3.2. Proposed capabilities

You could:

- deploy an instance of the issue-tracker, bare metal/docker install should take no more than 2h
- demo the simple search feature (working only with name and description cols , but you could ddl hack-it)
- provide access to the non-technical person via http for CRUD operations
- provide them with initial links to grasp the "semi-sql" syntax
- quickly define LOTS of tables DDL by using the existing examples and just changing the columns
- load initial data via xls (less than 10k rows per sheet should be ok)

3.3. Installation assistance

We could provide you with free assistance for the deployment of the first instance in your organization, even we consider the existing documentation good enough to deploy the application only by following the instructions in the Installation Guide:

<https://github.com/YordanGeorgiev/issue-tracker/blob/master/doc/md/issue-tracker-installation-and-configuration-guide.md>

In the real IT life however the mealage always varies, hence this last "bonus offer".

4. DEMO

You can check the following running instance of the tool:

http://ec2-34-243-97-157.eu-west-1.compute.amazonaws.com:8080/prd_issue_tracker/list/monthly_issues?as=grid&pick=id,status,prio,name,weight,start_time,stop_time&page-size=7&page-num=1

5. INSTALLATION AND CONFIGURATION

You could either try quickly to execute the instructions bellow this section or

follow the installation instructions from the Installation and Configuration Guide in the docs/md dir:

<https://github.com/YordanGeorgiev/issue-tracker/blob/master/doc/md/issue-tracker-installation-and-configuration-guide.md>

6. ADDITIONAL DOCS

Additional docs could be found in the doc/md dir.

<https://github.com/YordanGeorgiev/issue-tracker/tree/master/doc/md>

and in pdf format in the doc/pdf dir:

<https://github.com/YordanGeorgiev/issue-tracker/tree/master/doc/pdf>

A good starting point could be the "features doc." :

<https://github.com/YordanGeorgiev/issue-tracker/blob/master/doc/md/issue-tracker-features-and-functionalities.md>

The user-stories doc :

<https://github.com/YordanGeorgiev/issue-tracker/blob/master/doc/md/issue-tracker-user-stories.md>

The Requirements document contains the requirements for the application:

<https://github.com/YordanGeorgiev/issue-tracker/blob/master/doc/md/issue-tracker-requirements.md>

The SystemGuide contains the description of the existing issue-tracker System and it's architecture:

<https://github.com/YordanGeorgiev/issue-tracker/blob/master/doc/md/issue-tracker-system-guide.md>

The DevOps Guide contains the instructions for the devops operators:

<https://github.com/YordanGeorgiev/issue-tracker/blob/master/doc/md/issue-tracker-devops-guide.md>

7. ACKNOWLEDGEMENTS

This project would not have been possible without the work of the people working on the following frameworks/languages/OS communities listed in no particular order.

- Perl
- Mojolicious
- GNU Linux
- Vue

Deep gratitudes and thanks to all those people ! This application aims to contain the best practices of our former colleagues and collaborators and co-travelers in life, which also deserve huge thanks for their support and contributions !

8. LICENSE

All the trademarks mentioned in the documentation and in the source code belong to their owners.

This application uses the Perl Artistic license, check the license.txt file or the following link :

<https://dev.perl.org/licenses/artistic.html>

Should any trademark attribution be missing, mistaken or erroneous, please contact us as soon as possible for rectification.