

# Table of Contents

Table of Contents	1
ISSUE-TRACKER USERSTORIES AND SCENARIOS	3
1. TEAM LEADER BIZ PERSPECTIVE	3
1.1. Projects management	3
1.1.1. Create new projects	3
1.1.2. Remove existing projects	3
1.1.3. Update existing projects	3
1.1.4. Search for existing projects	3
1.1.5. Switch between projects	3
1.1.5.1. Web UI for switch between projects	3
1.1.5.2. Current project visibility	3
1.2. Time management	3
1.2.1. Total planned time tracking	3
1.2.2. Total remaining allocated time tracking	4
1.2.3. Total spent time tracking	4
1.2.4. Tracking of issues per period	4
1.3. Security Management	4
1.3.1. Users management	4
1.3.1.1. Add new users to a project	4
1.3.1.2. Update existing users in a project	4
1.3.1.3. Delete existing users from a project	4
1.3.1.4. Search for users in a project	4
1.3.2. Users access management	4
1.3.2.1. Grant access per project	4
1.3.2.2. Personal data handling minimization	4
2. TEAM MEMBER BIZ PERSPECTIVE	5
2.1. Issues management	5
2.1.1. Create new issues	5
2.1.2. Update existing issues	5
2.1.3. Remove existing issues	5
2.1.4. Search for existing issues	5
2.1.5. Track issues progress	5
2.1.6. Track issues history	5
2.2. Items management	5
2.2.1. Create new Items	5
2.2.2. Update existing Items	5
2.2.3. Remove existing Items	5
2.2.4. Search for existing Items	5
2.2.5. Track Items progress	6
2.2.6. Track Items history	6
2.3. Track issues relations	6
2.3.1. Generic search for items from a single entity ( table )	6
2.3.2. Generic list for the searched items from a single entity ( table )	6
2.3.2.1. Generic list labels for the searched items from a single entity ( table )	6
2.3.2.2. Generic cloud list for the searched items from a single entity ( table )	6
2.3.3. Generic create capability from UI for new items from a single entity ( table )	6
2.3.4. Generic edit capability from UI for existing items from a single entity ( table )	6
2.3.5. Generic delete capability from UI for existing items from a single entity ( table )	6
2.4. Measure success	7
2.5. Monitor success	7
2.6. Time management	7
2.6.1. time centric planning	7
2.6.2. time centric reporting	7
2.7. Generic CRUDS for items	7
2.8. Project's persons issue combinations	7
3. PROJECT OBSERVER BIZ PERSPECTIVE	7
3.1. Projects observation	7
3.2. Issues observation	7
4. SYSADMIN PERSPECTIVE	7
4.1. System deployability	7
4.2. System performance	8
4.3. System stability	8
4.4. System reliability	8
4.5. Ease of use	8
4.6. Security	8
5. ETL AND INTEGRATIONS PERSPECTIVE	8
5.1. Database to json files data load	8
5.2. Json files to db data load	8
5.3. Xls-to-mysql-db hierarchical data load	8
5.3.1. error reporting in xls-to-mysql-db hierarchical data load	8
5.4. Xls-to-postgres-db hierarchical data load	8
5.4.1. error reporting in xls-to-postgres-db hierarchical data load	9
6. DEVOPS PERSPECTIVE	9
6.1. System verifiability and testability	9
6.1.1. Clarity and brevity of the end to end tests	9
6.1.2. Abort end-to-end tests on single test fail	9
6.2. Logging	9
6.2.1. Control flow logging	9
6.2.2. Log entries format configuration	9
6.2.3. Single entry point for end to end tests	9
6.2.4. Tool run log to human readable description	9
6.3. Application's source code and documentation integrity	9
6.3.1. Userstories to test case relations	10
6.3.2. UUID trackability for test files and userstories	10
6.3.3. Components start run message print	10
6.3.4. Tool exit with exit code and exit message	10
6.3.5. Execution path tracing by UUID's	10
6.3.6. Issues directories naming conventions	10
6.3.7. Issues files naming conventions	10
6.3.8. Issues file open	10
6.3.9. Issues files history	10
6.3.10. Issues files naming conventions	10
6.3.10.1. Issues files naming conventions for the project	11
6.3.10.2. Issues files naming conventions for current date	11

6.3.10.3. Issues files naming conventions for the time frame	11
6.4. Issues transformations	11
6.4.1. Load by txt-to-db action	11
6.4.1.1. Load issues file from file system to db	11
6.4.2. Load issues by db-to-xls action	11
6.4.3. Load issues by xls-to-db action	11
6.4.3.1. Load issues by xls-to-db action for insert or upset	11
6.4.3.2. Load issues by xls-to-db action by truncating or not the loadable table	11
6.4.4. Load issues by db-to-txt	12
6.4.4.1. xls-to-db action load sort by issues prio attribute	12
6.4.4.2. db-to-txt action load sort by issues start_time attribute	12
6.4.4.3. db-to-txt action load sort by issues start_time attribute	12
6.4.5. Load issues file from db to file system	12
6.5. issues file filtering	12
6.6. Single shell call for projects switching	12
6.7. Issues publishing from shell calls	12
6.7.1. Issues publishing in e-mail format	12
6.7.2. Issues handling in google sheet format	13
6.7.3. Issues publishing in google calendar format	13
6.8. Metadata handling	13
7. UI PERSPECTIVE COMMON FOR ALL ROLES	13
7.1. UI Performance	13
7.1.1. UI Page load times	13
7.1.2. UI Page parts load times	13
7.2. SEARCH	13
7.2.1. SearchBox autocomplete for changing the project ( databases )	13
7.2.2. SearchBox autocomplete for item-name	13
7.2.3. SearchBox autocomplete for with <<attribute>> name <<operator>> <<attribute-value>>	14
7.3. Items listing	14
7.3.1. items listing in table format	14
7.3.1.1. Automatic issue items sequencing	14
7.3.1.2. items re-ordering by desired or default attribute in list view	14
7.3.1.3. items list default row height	14
7.3.1.4. Drag and drop columns in table to reorder columns order	14
7.3.1.5. Columns resizing in table listing	14
7.4. Items editing	14
7.4.1. Single Cell Items editing in list as table view	14
7.4.1.1. Single Cell Items editing in list as table view in-line	14
7.4.1.2. Items editing in list as table view via form	14
7.4.1.3. Refresh list on item edit	15
7.5. Items creation	15
7.5.1. Items creation inline	15
7.5.2. Items creation by form	15
7.6. Items deletion	15
7.6.1. Items deletion inline	15
7.6.2. Items deletion from the edit form	15
7.7. Items export	15
7.7.1. Items xls export	15
7.7.2. Issues export to Google calendar	16
7.8. Items import	16
7.8.1. Items xls import	16
7.9. Items move	16
7.10. Mobile UI	16
7.11. UI for accessing different projects	16
7.12. UI for Time management in Google Calendar	16
7.13. items data transfer between different projects	16
7.13.1. Copy an issue-tracker instance issue to a google calendar event	16
7.14. Issues import from Google calendar	16
7.15. Access issues txt format from email	16
7.16. Access issues data from Google sheet	16
7.16.1. Apply publish filter while posting to Google Sheet	16
8. UI DEVELOPER PERSPECTIVE	17
8.1. Testability	17
8.2. Code traceability	17

# ISSUE-TRACKER USERSTORIES AND SCENARIOS

## 1. TEAM LEADER BIZ PERSPECTIVE

As a team leader  
In order to operate successfully one or many projects of my team(s)  
I want to have a nice user experience while using the issue-tracker tool.

### 1.1. Projects management

As an team leader  
In order to be able to manage multiple projects  
I want to be able to create , update and remove projects.

#### 1.1.1. Create new projects

As an team leader  
In order to be able to manage new projects  
I want to be able to create projects via the issue-tracker

#### 1.1.2. Remove existing projects

As an team leader  
In order to be able to stop the work on existing projects  
I want to be able to remove projects via the issue-tracker

#### 1.1.3. Update existing projects

As an team leader  
In order to be able to change attributes of the projects I am responsible for  
I want to be able to update the projects' data.

#### 1.1.4. Search for existing projects

As an team leader  
In order to be able to quickly access existing projects  
I want to be able to search the projects.

#### 1.1.5. Switch between projects

As a team leader  
In order to manage issues from different projects  
I want to be able to switch between different projects easily and quickly

##### 1.1.5.1. Web UI for switch between projects

As a team leader  
In order to manage issues from different projects  
I want to be able to switch between different projects easily and quickly  
by simply changing the first token of the url of the app

##### 1.1.5.2. Current project visibility

As a team leader  
In order to avoid confusion between different projects  
I want to be able to see the current project name from any interface I am working in quickly and easily

### 1.2. Time management

As an team leader  
In order to be able the maximize the performance of the team for issue-tracker used periods  
I want to be able to manage time efficiently  
by accessing a simple page containing its value and the period it is related to.

#### 1.2.1. Total planned time tracking

As a team leader  
In order to see the planned time left for achieving the goals of a period  
I want to be able to have a +-3% approximation of the planned time left for a period  
by accessing a simple page containing its value and the period it is related to

### **1.2.2. Total remaining allocated time tracking**

As a team leader

In order to see the remaining time left for achieving the goals of a period

I want to be able to have a  $\pm 3\%$  approximation of the allocated time left for a period

### **1.2.3. Total spent time tracking**

As a team leader

In order to see the spent time left for achieving the goals of a period

I want to be able to have a  $\pm 3\%$  approximation of the spent time left for a period

by accessing a simple page containing its value and the period it is related to

### **1.2.4. Tracking of issues per period**

As a team leader

In order to see the relation of the issues to the daily, weekly, monthly, yearly, quinquennially and decadelly periods

I want to be able to manage the issues within those periods per period and mother period

## **1.3. Security Management**

As an team leader

In order to keep my business data secure

I want to be able decide which users to which projects will have access to the data my issue-tracker application has access to.

### **1.3.1. Users management**

As an team leader

In order to keep my business data secure

I want to be able decide which users to which projects will

#### **1.3.1.1. Add new users to a project**

As an team leader

In order to be able to add new users into a project

I want to be able to add them via the UI only by their e-mail address.

#### **1.3.1.2. Update existing users in a project**

As an team leader

In order to be able to update the existing users from a project

I want to be able to update their details via the UI only by their e-mail address.

#### **1.3.1.3. Delete existing users from a project**

As an team leader

In order to be able to delete the existing users from a project

I want to be able to delete their details via the UI only by their e-mail address.

#### **1.3.1.4. Search for users in a project**

As an team leader

In order to be able to search the existing users from a project

I want to be able to search their details via the UI only by their e-mail address.

### **1.3.2. Users access management**

As a team leader

In order provide the persons and programs access to my project

I want to provide read, write access to the data and execute access ( run DDL's) per table

#### **1.3.2.1. Grant access per project**

As a team leader

In order to enroll authenticated users into the project I am responsible to

I want to be able to grant them with access by only writing their e-mail into a text field and clicking invite button.

#### **1.3.2.2. Personal data handling minimization**

As a team leader

In order to avoid legal obligations and complex procedures while handling personal data

I want to be able to handle the interpersonal exchange of data by collecting ONLY the e-mail of the persons or programs participating in the project

## **2. TEAM MEMBER BIZ PERSPECTIVE**

As a team member  
In order to operate successfully in the project  
I want to have a nice user experience while using the issue-tracker application  
by being able to manage all the items in the application ( issues,questions,problems , etc. )

### **2.1. Issues management**

As a team member of the issue-tracker  
In order to achieve the best possible efficiency during the work on one or many projects  
I want to be able to manage the issues in those projects.

#### **2.1.1. Create new issues**

As an team member  
In order to be able to manage multiple issues  
I want to be able to create , update and remove issues.

#### **2.1.2. Update existing issues**

As an team member  
In order to be able to manage new issues  
I want to be able to create issues via the issue-tracker

#### **2.1.3. Remove existing issues**

As an team member  
In order to be able to stop the work on existing issues  
I want to be able to remove issues via the issue-tracker

#### **2.1.4. Search for existing issues**

As an team member  
In order to be able to change attributes of the issues I am responsible for  
I want to be able to update the issues' data.

#### **2.1.5. Track issues progress**

As an team member  
In order to be able to quickly access existing issues  
I want to be able to search the issues.

#### **2.1.6. Track issues history**

As a team member  
In order to keep track on what and when was planned on daily basis  
I want to be able to keep track what was planned on a project term - day,week,month,year,quinquennial or decade

### **2.2. Items management**

As a team member of the issue-tracker  
In order to achieve the best possible efficiency during the work on one or many projects  
I want to be able to manage the Items in those projects, where items could be ( problems , questions etc. )

#### **2.2.1. Create new Items**

As an team member  
In order to be able to manage multiple Items  
I want to be able to create , update and remove Items.

#### **2.2.2. Update existing Items**

As an team member  
In order to be able to manage new Items  
I want to be able to create Items via the issue-tracker

#### **2.2.3. Remove existing Items**

As an team member  
In order to be able to stop the work on existing Items  
I want to be able to remove Items via the issue-tracker

#### **2.2.4. Search for existing Items**

As an team member  
In order to be able to change attributes of the Items I am responsible for  
I wanto to be able to update the Items' data.

#### **2.2.5. Track Items progress**

As an team member  
In order to be able to quickly access existing Items  
I wanto to be able to search the Items.

#### **2.2.6. Track Items history**

As a team member  
In order to keep track on what and when was planned on daily basis  
I wanto to be able to keep track what was planned on a project term - day,week,month,year,quinquennial or decade

### **2.3. Track issues relations**

As a team member of a project  
In order to trace the issues relations to userstories, features and tests or any other objects  
I wanto to be able to access the related objects to an issue by means of a link

#### **2.3.1. Generic search for items from a single entity ( table )**

As a team member  
In order to be able to find all the items from a single entity  
I want to be search for those items from the UI of the application  
by using a single SearchBox UI interface.

#### **2.3.2. Generic list for the searched items from a single entity ( table )**

As a team member  
In order to be able to list and review all the items from a single entity  
I want to be able to review the searched items for those items from the UI of the application  
by using a single label forms like interface

##### **2.3.2.1. Generic list labels for the searched items from a single entity ( table )**

As a team member  
In order to be able to list and review all the items from a single entity  
I want to be able to review the searched items for those items from the UI of the application  
by using a single label forms like interface

##### **2.3.2.2. Generic cloud list for the searched items from a single entity ( table )**

As a team member  
In order to be able to list and review all the items from a single entity  
I want to be able to review the searched items for those items from the UI of the application  
by using a single tag cloud like interface

#### **2.3.3. Generic create capability from UI for new items from a single entity ( table )**

As a team member  
In order to be able to create new items from a single entity  
I want to be able to create new items from the UI of the application  
by using a simple form.

#### **2.3.4. Generic edit capability from UI for existing items from a single entity ( table )**

As a team member  
In order to be able to edit existing items from a single entity  
I want to be able to edit those items from the UI of the application  
by using a simple form.

#### **2.3.5. Generic delete capability from UI for existing items from a single entity ( table )**

As a team member  
In order to be able to delete existing items from a single entity  
I want to be able to edit those items from the UI of the application

by using a simple form.

## **2.4. Measure success**

As a team member

In order to measure the success of the planned issues

I want to be able to measure the deliverables of each issue by comparable metrics.

## **2.5. Monitor success**

As a team member

In order to monitor the success of the planned issues

I want to be able to monitor the metrics of the issues.

## **2.6. Time management**

As an issues-manager

In order to be prepared for issues such as ( events , tasks ) which have start and stop time

I want to be able to save their start time and stop time per issue in every possible interface

### **2.6.1. time centric planning**

As an issues-manager

In order to be able to plan the issues data for a certain term - day,week,month,year,quinquennial or decade

I want to be able to perform all the features of the issue-tracker on that specific period regardless whether it is today , in the past or in the future

### **2.6.2. time centric reporting**

As an issues-manager

In order to be able to report the issues data for a certain term - day,week,month,year,quinquennial or decade

I want to be able to perform all the features of the issue-tracker on that specific day regardless whether it is today , in the past or in the future

## **2.7. Generic CRUDS for items**

As a team member

In order to be able to manage all the items in the application I have access to

I want to be able to create,update,delete and search for those items from the UI of the application.

## **2.8. Project's persons issue combinations**

As the project manager of an issue-tracker project

In order to be able to quickly and reliably combine the reported hours by the project's people

I want to be able to read their issue-tracker formatted google sheets and combine them into a single project's google issue-tracker sheet

## **3. PROJECT OBSERVER BIZ PERSPECTIVE**

As a project observer

In order to observe the advancement of a project

I want to have a nice user experience while using the issue-tracker application.

### **3.1. Projects observation**

As a project observer

In order to observe the advancement of a project

I want to be able to observe the project's data.

### **3.2. Issues observation**

As a project observer

In order to observe the advancement of the project's issues

I want to be able to observe the project's issues.

## **4. SYSADMIN PERSPECTIVE**

As a sysadmin of the issue-tracker application

In order to complete the tasks and activities of my role

I want to have a nice user experience while using the issue-tracker application.

### **4.1. System deployability**

As the SysAdmin

In order to be able to provide access to a new database driven application to my organization

I want to be able to deploy an instance of the issue-tracker application and spawn a new project out of it in less than a hour from a clean Linux host.

#### **4.2. System performance**

As the SysAdmin

In order to ensure the performance of the issue-tracker application

I want to the System containing the issue-tracker application to perform its functions within the defined performance criteria

#### **4.3. System stability**

As the SysAdmin

In order to minimize downtimes and ensure continuous operations

I want to the System containing the issue-tracker application to perform its defined functions on request without interruptions or unknown side effects

#### **4.4. System reliability**

As the SysAdmin

In order to be able to rely on the operations of the tool

I want to the System containing the issue-tracker application to perform its functions as specified consistently

#### **4.5. Ease of use**

As the SysAdmin

In order to be efficient and decrease the amount of errors

I want to generally perform any command the system within the sysadmin scope via clean and memorable oneliners

#### **4.6. Security**

As the SysAdmin

In order to be able to provide the best possible security level of operation for the Issue-tracker instance I am responsible for

I want to have pre-defined and clear set of tasks and activities to perform related to the security.

### **5. ETL AND INTEGRATIONS PERSPECTIVE**

As an ETL and integrations specialist

In order to complete the tasks and activities of my role

I want to have a nice user experience while using the issue-tracker application.

#### **5.1. Database to json files data load**

As the ETL and Integration Specialist

In order to be able to quickly move all the project data into a different storage format

I want to be able to export the project db data into json files - one per table via a single shell call.

#### **5.2. Json files to db data load**

As the ETL and Integration Specialist

In order to be able to quickly move all the project data from json files into the db tables

I want to be able to import the exported json files ( one per table ) into the database.

#### **5.3. Xls-to-mysql-db hierarchical data load**

As the Data Integrator

In order to be efficient while handling the System's hierarchical data

I want to be able to

use a single shell call to load all or chosen table(s) to the mysql db

##### **5.3.1. error reporting in xls-to-mysql-db hierarchical data load**

As the Data Integrator

In order to be efficient while troubleshooting data loading errors

I want to be able to see :

- which table's load failed

- what was the error in failed to

#### **5.4. Xls-to-postgres-db hierarchical data load**

As the Data Integrator

In order to be efficient while handling the System's hierarchical data

I want to be able to

use a single shell call to load all or chosen table(s) to the postgres db



#### **5.4.1. error reporting in xls-to-postgres-db hierarchical data load**

As the Data Integrator

In order to be efficient while troubleshooting data loading errors

I want to be able to see :

- which table's load failed
- what was the error in failed to

### **6. DEVOPS PERSPECTIVE**

As a devops operator of issue-tracker application

In order to complete the tasks and activities of my role

I want to have a nice user experience while using the issue-tracker application.

#### **6.1. System verifiability and testability**

As an ITOPS

In order to be able to rely on the operations of the tool

and manage easily its features and functionalities

I want to easily verify and test parts or the whole System

by issuing a single shell call.

##### **6.1.1. Clarity and brevity of the end to end tests**

As an ITOPS

In order to be able to verify all the features and functionalities of the tool within the System

I want to see the results of each test in 1 flow in the following format:.

##### **6.1.2. Abort end-to-end tests on single test fail**

As an ITOPS

In order to be able to run continuously end-to-end tests and skip for several runs failing tests

I want to be able to configure the single e2e entry point script to skip certain tests, but report me what was skipped.

#### **6.2. Logging**

As a Full-Stack Developer

In order to quickly understand what is happening in the application

I want to have easy-to-use and highly customizable logging to both file and console.

##### **6.2.1. Control flow logging**

As a CLI user

In order to be able to understand what the issue tracker tool is executing

I want to have configurable logging with stderr, stdout and file output

##### **6.2.2. Log entries format configuration**

As a Full-Stack Developer

In order to be able to get the msg of any component of the application

I want each log entry to content:

- the type of the entry - log , error, warn , fatal
- the timestamp of the log entry event
- the name of the component issuing the msg and the line num of the src file
- the msg as it was echoed by the application

##### **6.2.3. Single entry point for end to end tests**

As an DevOps

In order to be able to verify all the features and functionalities of the tool within the System

I want to run a single shell call running all the end-to-end test of the application ensuring the prespecified features and functionalities.

##### **6.2.4. Tool run log to human readable description**

As a CLI user

In order to be able to get a human readable description of the log of the specific run of the tool

I want to be able to translate the recorded uuid's in the execution run log to their respective records

#### **6.3. Application's source code and documentation integrity**

As a Full-Stack Developer

In order to make easy the entry of other developers to the projects

I want to be able to point to written documentation for user-stories, issues, features and functionalities, which will be linked to parts of the source code.

#### **6.3.1. Userstories to test case relations**

As a Developer

In order to ensure the stability and expandability of the application

I want to be able to run for each implemented user-story a single test

#### **6.3.2. UUID trackability for test files and userstories**

As a Developer

In order to identify each user-story to be tested with its according test

I want to be able to track each user-story or test code entry point file by UUID.

#### **6.3.3. Components start run message print**

As a CLI user

In order to know when a component has been started

I want to be able to see the "START <<COMPONENT NAME>>" on either the STDOUT or the log file of the component

#### **6.3.4. Tool exit with exit code and exit message**

As a CLI user or calling automated component

In order to be able to understand whether or not the execution of the call to the tool was successful or not

I want to get the exit code from the tool execution and see the exit message

#### **6.3.5. Execution path tracing by UUID's**

As a DevOps operator

Foreach execution run of the tool

I want to be able to walk through the execution path of the tool programmatically.

#### **6.3.6. Issues directories naming conventions**

As a issues manager

In order to be able to manage lots of issues from different projects stored in plain txt files

I want to be able to store large quantity of issues txt files by their date on daily, weekly, monthly, yearly and decadal basis

#### **6.3.7. Issues files naming conventions**

As a issues manager

In order to be able to manage lots of issues from different projects stored in plain txt files and open them quickly

I want to be able to just type the first letter in a text editor supporting select opened file by typing its first letter and jump to that file

#### **6.3.8. Issues file open**

As a CLI user

In order to be able quickly to access my issues ( daily , weekly, monthly , yearly )

I want to be able to view my daily issues by simply opening a single txt file

#### **6.3.9. Issues files history**

As a CLI user

In order to be able quickly to search through old issues

I want to be able to access old issues files by their date held in their file names from the file system

#### **6.3.10. Issues files naming conventions**

As a DevOps

In order to be able quickly to access and manage programmatically issues  
I want to be able to guess the file paths of the issues file by their date

#### **6.3.10.1. Issues files naming conventions for the project**

As a DevOps

In order to be able quickly to switch between different projects

I want to have the project name of the issues file in its name as the first token as follows:

<<issue\_tracker\_project>>-issues.<<current-iso-date>>.<<daily|weekly|monthly|yearly>>.txt

#### **6.3.10.2. Issues files naming conventions for current date**

As a DevOps

In order to be able quickly and programmatically to go back in the history

I want to have the current registration date in the file name and path

<<issue\_tracker\_project>>-issues.<<current-iso-date>>.<<daily|weekly|monthly|yearly>>.txt

#### **6.3.10.3. Issues files naming conventions for the time frame**

As a DevOps

In order to be able quickly and programmatically to go back in the history

I want to have the current registration date in the file name and path

<<issue\_tracker\_project>>-issues.<<current-iso-date>>.<< ( daily|weekly|monthly|yearly) >>.txt

### **6.4. Issues transformations**

As a cli user of the issue-tracker application

In order to be able to sort the issues according to their attributes

and edit them in both txt file and xls file

I want to be able to perform the following loads:

txt-to-db - to load a txt file with issues to an issues table in db

db-to-xls - to load a xls file from db table to xls

xls-to-db - to load a xls file with issues to an issues table in db

db-to-txt - to load a txt file from db table

#### **6.4.1. Load by txt-to-db**

##### **action**

As a cli user of the issue-tracker application

in order to store my issues in structured form to db for further processing

I want to be able to load any issue file with a single line shell call to a db

##### **6.4.1.1. Load issues file from file system to db**

As a cli user of the issue-tracker application

- in order to be able to handle issues from different projects

- and load them to db for further processing

I want to:

- pre-set the variables of the project

- and then load any issue file with a single line shell call to a db

- and optionally specify the period of the issues file ( daily , weekly , monthly , yearly ) with daily as default

#### **6.4.2. Load issues by db-to-xls**

##### **action**

As a cli user of the issue-tracker application

in order to be able to sort and edit my issues in Excel

I want to be able to unload my issues from one or many tables in the db at once in a single shell call

#### **6.4.3. Load issues by xls-to-db**

##### **action**

As a cli user of the issue-tracker application

in order to store my issues in structured form to db for further processing after being sorted in Excel

I want to be able to load my latest xls file with a single line shell call to a db

by choosing the names of the tables to load

##### **6.4.3.1. Load issues by xls-to-db action for insert or upsert**

As a cli user of the issue-tracker application

in order to insert or upsert my issues in structured form to db for further processing after being sorted in Excel

I want to be able to load my latest xls file with a single line shell call to a db

by choosing the names of the tables to load and specifying upsert by adding the guid column to the xls sheet

##### **6.4.3.2. Load issues by xls-to-db action by truncating or not the loadable table**

As a cli user of the issue-tracker application  
in order to store my issues in structured form to db for further processing after being sorted in Excel  
I want to be able to load my latest xls file with a single line shell call to a db  
by choosing the names of the tables to load

#### **6.4.4. Load issues by db-to-txt**

As a cli user of the issue-tracker application  
in order to store my issues in more structurized data format for further procesing  
I want to :  
- be able to load the issues for a period from the db to a file  
- by choosing the names of the tables to load

##### **6.4.4.1. xls-to-db action load sort by issues prio attribute**

As a cli user of the issue-tracker application  
during the db-to-txt action load  
in order to understand the priority of my issues  
I want to :  
- be able to specify the order in the issues txt files lines to be based on the prio attribute  
by choosing the names of the tables to load

##### **6.4.4.2. db-to-txt action load sort by issues start\_time attribute**

As a cli user of the issue-tracker application  
during the db-to-txt action load  
in order to understand the priority of my issues  
I want to :  
- be able to specify the order in the issues txt files lines to be based on the start\_time attribute ( start\_time could be in the following format YYYY-mm-dd HH:MM in start\_time or HH:MM )  
by choosing the names of the tables to load

##### **6.4.4.3. db-to-txt action load sort by issues start\_time attribute**

As a cli user of the issue-tracker application  
during the db-to-txt action load  
in order to view the issues by categories  
I want to :  
- be able to specify the order in the issues txt files lines to be based on the category attribute

#### **6.4.5. Load issues file from db to file system**

As a cli user of the issue-tracker application  
in order to store my issues in more structurized data format for further procesing  
I want to :  
- be able to load the issues for a period from the db to a file  
- and optionally specify the period of the issues file ( daily , weekly , monthly , yearly ) with daily as default

#### **6.5. issues file filtering**

As a CLI user  
In order to filter quickly my issues  
I wanto to be able to show the issues with their categories of only certain status

#### **6.6. Single shell call for projects switching**

As an issues-manager  
In order to be able to switch between different projects quickly  
I wanto to be able to issue a single shell call for loading a project's configuration  
and run the issue-handler against this pre-loaded configurtion

#### **6.7. Issues publishing from shell calls**

As a DevOps  
In order to be able to quickly share the current issues data in tabular format  
I wanto to be able to issue a single shell call for copying the current items data to a medium by specifying the tables to be published

##### **6.7.1. Issues publishing in e-mail format**

As a DevOps  
In order to be able to quickly share the current issues data in email format

I want to be able to issue a single shell call for copying the current items data to email by specifying the tables to be published

#### **6.7.2. Issues handling in google sheet format**

As a DevOps

In order to be able to quickly share the current issues data in tabular format

I want to be able to issue a single shell call for copying the current items data to google sheet by specifying the tables to be published

#### **6.7.3. Issues publishing in google calendar format**

As a DevOps

In order to be able to quickly share the current issues data in google calendar format

I want to be able to issue a single shell call for copying the current items data to google calendar by specifying the tables to be published for the items having set start\_time and stop\_time attributes.

### **6.8. Metadata handling**

As a DevOps

In order to be able to programatically manage all aspects of my data

I want to have a single entry point to manage the meta data per tables , columns and UI elements

so that even a table, column or whatever object is not populated in the meta still there will be default values for it usable by the application

## **7. UI PERSPECTIVE COMMON FOR ALL ROLES**

As an UI user of the issue-tracker application

In order to manage my issues via the UI successfully

I want to have a nice user experience while using the issue-tracker application.

### **7.1. UI Performance**

As an UI user of the issue-tracker application

In order to enjoy the usage of the tool and interact efficiently

I want to have responsive and quick UI.

#### **7.1.1. UI Page load times**

As an UI user of the issue-tracker application

In order to enjoy the usage of the tool and interact efficiently

I want to have a maximum page load time on efficient network less than 0.5 seconds and preferably even less than 0.3 seconds

#### **7.1.2. UI Page parts load times**

As an UI user of the issue-tracker application

In order to enjoy the usage of the tool and interact efficiently

I want to have a maximum page part load time on efficient network less than 0.5 seconds and preferably even less than 0.3 seconds

### **7.2. SEARCH**

As an issue-tracker ui user

In order to be able to search ANY items ( issues, problems, ideas etc. )

I want to have a pop-up search SearchBox with dimmed background providing with interactive autocomplete, which would assist me in specifying the search criteria for any item I want to list.

#### **7.2.1. SearchBox autocomplete for changing the project ( databases )**

As an issue-tracker ui user

In order to be able to quickly switch between projects of the issue-tracker application instance I have access to

I want to be able to start typing with the ":in" operator and the System to autocomplete a dropdown of databases my instance has access to.

#### **7.2.2. SearchBox autocomplete for item-name**

As an issue-tracker ui user

In order to be able to quickly specify the name of the item I am searching for ( issue,problem,idea etc. )

I want to be able to use a special natural-language like syntax in the autocomplete when the SearchBox appears, by having the System displayed the ":for" string and starting providing me autocomplete for the items names in a dropdown

list I could choose from.

### **7.2.3. SearchBox autocomplete for with <<attribute>> name <<operator>> <<attribute-value>>**

As an issue-tracker ui user

In order to be able to quickly specify the search criteria to re-strict the already chosen item to search for

I want to be able to use a special natural-language like syntax in the autocomplete when the SearchBox appears, by having the System displayed the "with:type-here-the-attribute op:type-here-operator val:type-here-the-value" string and starting providing me autocomplete for the filtering criteria.

## **7.3. Items listing**

As an issue-tracker ui user

In order to be able to quickly see as much items ( issues, problems, ideas etc. )

I want to list the items in a web page according to the filtering criteria I might have specified earlier on ...

### **7.3.1. items listing in table format**

As an UI user of the issue-tracker application

In order to quickly display as much items ( issues,problems, questions, etc. ) as possible

I want to be able to list the items per period.

#### **7.3.1.1. Automatic issue items sequencing**

As an UI user of the issue-tracker application

In order to save time while arranging all the different issue items

I want the System to automatically sequence each item in list view by a default incremental sequence unless I have specified my own sequence.

#### **7.3.1.2. items re-ordering by desired or default attribute in list view**

As a UI user

In order to prioritize and re-arrange to a logical sequence my items

I want to be able to drag and drop items up and down , which would correspondingly increase or decrease their attribute to which they are currently sorted or ordered by.

#### **7.3.1.3. items list default row height**

As a UI user

In order to quickly comprehend the data in the lists

I want each row of the ui to have a certain minimum height and whenever the data cannot fit into this height to be greater than it

#### **7.3.1.4. Drag and drop columns in table to reorder columns order**

As the UI user of an issue-tracker instance

In order to list the attributes of an item in a order significant for the moment

I want to be able to drag and drop columns in the table so that the System would re-render the table with the new order.

#### **7.3.1.5. Columns resizing in table listing**

As the UI user of an issue-tracker instance

In order to see better the contents in the table columns

I want to be able to resize the columns width ( excel like or by other means )

## **7.4. Items editing**

As an UI user of the issue-tracker application

In order to update the application data via the UI

I want to be able to edit the data for ANY of the items in the application I have access to.

### **7.4.1. Single Cell Items editing in list as table view**

As an UI user of the issue-tracker application

In order to be able to quickly edit the data of ANY item I

I want to be able to quickly navigate to the item to edit and update the data of exactly this item

#### **7.4.1.1. Single Cell Items editing in list as table view in-line**

As an UI user of the issue-tracker application

In order to be able to quickly edit the data of ANY item I

I want to be able to quickly navigate to the cell of the table of the data I want to edit and update the data by clicking with the mouse or navigating with the tab and typing the new data, so that when navigating out of the cell the data will be updated unless the Esc case has been clicked.

#### **7.4.1.2. Items editing in list as table view via form**

As an UI user of the issue-tracker application  
In order to be able to quickly edit the data of ANY item  
I want to be able to click on the edit button , fill in the popping-up dialog with the form of the full data of the item and clicking the Save button.

#### **7.4.1.3. Refresh list on item edit**

As an UI user of the issue-tracker application  
In order to see the updated list after editing it  
I want the System to refresh the list according to the new updated value - that is for example if the order was set to be prio and the prio was updated, the page should be refreshed with the new prio order applied

### **7.5. Items creation**

As an UI user of the application  
I order to create new items in the application  
I want to be able to create them via the UI  
by clicking "create new button" and filling as few as possible data entries and clicking a Save button for ANY of the items in the application.

#### **7.5.1. Items creation inline**

As an UI user of the application  
I order to create new items in the application  
I want to be able to create them via the edit-table listing UI page  
by clicking jsut "create new button" ,  
when the System will add a new row which will appear automatically in top of the listing sorted by id ( which will be automatically generated based on the timestamp of the creation moment )

#### **7.5.2. Items creation by form**

As an UI user of the application  
I order to create new items in the application  
I want to be able to create them via the edit-table listing UI page  
by clicking jsut "create new button" ,  
when the System will add a new row which will appear automatically in top of the listing and the edit form with the pre-filled "type the <<attribute-name >> will occur ...

### **7.6. Items deletion**

As an UI user of the application  
I order to delete existing items in the project  
I want to be able to delete them via the UI  
by clicking a "delete " button and confirming the deletion for the item

#### **7.6.1. Items deletion inline**

As an UI user of the application  
I order to delete existing items in the project  
I want to be able to delete them via the UI  
by clicking a "delete " button and confirming the deletion for the item by clicking on the ok dialog presented by the System.

#### **7.6.2. Items deletion from the edit form**

As an UI user of the application  
I order to delete existing items in the project  
I want to be able to delete them via the UI of the edit form  
by clicking a "delete " button in the form  
and confirming the deletion for the item by clicking on the ok dialog presented by the System.

### **7.7. Items export**

As an UI user of the application  
I order to export the data of the items in the application  
I want to be able to perform every possible export in the UI from a single button click

#### **7.7.1. Items xls export**

As an UI user of the application  
I order to export the data of the items in the application into xls format  
I want to be able to perform the exprot by clicking on an xls icon like button and point to the file path to save to the listed in the UI data

### **7.7.2. Issues export to Google calendar**

As the UI user of an issue-tracker instance

In order to be able to visualize and manage my start- and stop\_time having issues better

I want to be able to export my issues to Google calendar

## **7.8. Items import**

As an UI user of the application

I order to import data of the items in the application

I want to be able to perform every possible import in the UI from a single button click and pointing to the file path of a file

### **7.8.1. Items xls import**

As an UI user of the application

I order to import the data of the items in the application into xls format

I want to be able to perform the exprot by clicking on an xls icon like button and point to the file path to save to the listed in the UI data

## **7.9. Items move**

As an UI user of the application

I order to move the items into different tables

I want to be able to move them from a button in the listing page by specifying the target table

## **7.10. Mobile UI**

As an UI user of the issue-tracker application

In order to enjoy to be able to access it quickly on the go

I want to be able to use the same UI on an advanced mobile phones.

## **7.11. UI for accessing different projects**

As an issue-tracker ui user

In order to be able to quickly switch between projects of the issue-tracker application instance I have access to

I want to be able to access a web page providing autocomplete to preloaded configuration entries for the different projects

## **7.12. UI for Time management in Google Calendar**

As an issue-tracker ui user

In order to be prepare for issues such as ( events , tasks ) which have start and stop time

I want to be able to view the issues with the same title, start\_time and stop\_time in google calendar

## **7.13. items data transfer between different projects**

As the UI user of an issue-tracker instance

In order to save be able to track my personal time usage between different projects and the different interdependancies

I want to be able to move items data from one project to another via the UI

### **7.13.1. Copy an issue-tracker instance issue to a google calendar event**

As an issue-tracker ui user

In order to be able to see my issues time-schedule via phone and browser in a calendar view

I want to be able to copy via the ui an issue as a new google calendar event

## **7.14. Issues import from Google calendar**

As the UI user of an issue-tracker instance

In order to be able to visualize and manage my start- and stop\_time having issues better

I want to be able to import my Google calendar issues into my issue-tracker profile on an issue-tracker instance

## **7.15. Access issues txt format from email**

As a user of the issue tracker tool

In order to be able to access and read my issues from a mobile device

I want to be able to send each period txt file from the daily folder via gmail.

## **7.16. Access issues data from Google sheet**

As the biz user of the issue tracker tool

In order to be able to share and edit the data with multiple users authenticated within the Google eco system

I want to be able to access , edit and update the issues data from google sheet

### **7.16.1. Apply publish filter while posting to Google Sheet**



As the biz user of the issue tracker tool

In order to show only relevant data to the future viewers of the published to Google sheets issues data

I want to be able to apply publishing filter for columns to be left unpublished per item table per project

## **8. UI DEVELOPER PERSPECTIVE**

As the UI Developer

In order to be able to deliver working solutions for the UI

I want to have user friendly development experience.

### **8.1. Testability**

As the UI Developer

In order to deliver working ui units

I want to be able to quickly setup the existing project with minimalistic default set of data

### **8.2. Code traceability**

As the UI Developer

In order to be able to grasp the inner working of the application

I want to be able to search by user-story uuid from the source code of the application.