Table of Contents

Table of Contents 1
ISSUE-TRACKER USERSTORIES AND SCENARIOIS 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
1.3.2. Add new users to a project 4
1.3.3. Update existing users in a project 4
1.3.4. Delete existing users from a project 4
1.3.5. Search for users in a project 4
2. TEAM MEMBER BIZ PERSPECTIVE 4
2.1. Issues management 4
2.1.1. Create new issues 4
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. Track issues relations 5
2.2.1. Generic search for items from a single entity (table) 5
2.2.2. Generic list for the searched items from a single entity (table) 5
2.2.2.1. Generic list labels for the searched items from a single entity (table) 5
2.2.2.2. Generic cloud list for the searched items from a single entity (table) 5
2.2.3. Generic create capability from UI for new items from a single entity (table) 5
2.2.4. Generic edit capability from UI for existing items from a single entity (table) 6
2.2.5. Generic delete capability from UI for existing items from a single entity (table) 6
2.3. Measure success 6
2.4. Monitor success 6
2.5. Time management 6
2.5.1. time centric planning 6
2.5.2. time centric reporting 6
2.6. Generic CRUDS for items 6
2.7. Project's persons issue combinations 6
3. PROJECT OBSERVER BIZ PERSPECTIVE 6
3.1. Projects observation 7
3.2. Issues observation 7
4. SYSADMIN PERSPECTIVE 7
4.1. System deployability 7
4.2. System performance 7
4.3. System stability 7
4.4. System reliability 7
4.5. Ease of use 7
5. ETL AND INTEGRATIONS PERSPECTIVE 7
5.1. Xls-to-mysql-db hierarchical data load 7
5.1.1. error reporting in xls-to-mysql-db hierarchical data load 7
5.2. Xls-to-postgres-db hierarchical data load 8
5.2.1. error reporting in xls-to-postgres-db hierarchical data load 8
6. DEVOPS PERSPECTIVE 8
6.1. System verifiability and testability 8
6.1.1. Clarity and breavity of the end to end tests 8
6.1.2. Abort end-to-end tests on single test fail 8
6.1.3. Control flow logging 8
6.1.4. Single entry point for end to end tests 8
6.1.5. Tool run log to human readable description 8
6.1.6. Userstories to test case relations 8
6.1.6.1. UUID tracability for test files and userstories 9
6.1.7. Components start run message print 9
6.1.8. Tool exit with exit code and exit message 9
6.1.9. Execution path tracing by UUID's 9
6.2. Manage details of issues with a single txt file 9
6.2.1. Issues directories naming conventions 9
6.2.2. Issues files naming conventions 9
6.2.3. Issues file open 9
6.2.4. Issues files history 9
6.2.5. Issues files naming conventions 9
6.2.5.1. Issues files naming conventions for the project 9
6.2.5.2. Issues files naming conventions for current date 10
6.2.5.3. Issues files naming conventions for the time frame 10
6.3. Issues transformations 10
6.3.1. Load by txt-to-db action 10
6.3.1.1. Load issues file from file system to db 10
6.3.2. Load issues by db-to-xls action 10
6.3.3. Load issues by xls-to-db action 10
6.3.3.1. Load issues by xls-to-db action for insert or upset 10
6.3.3.2. Load issues by xls-to-db action by truncating or not the loadable table 10
6.3.4. Load issues by db-to-txt 10
6.3.4.1. xls-to-db action load sort by issues prio attribute 11
6.3.4.2. db-to-txt action load sort by issues start_time attribute 11
6.3.4.3. db-to-txt action load sort by issues start_time attribute 11
6.3.5. Load issues file from db to file system 11
6.4. issues file filtering 11
6.5. Single shell call for projects switching 11

6.6. Issues publishing from shell calls 11
6.6.1. Issues publishing in e-mail format 11
6.6.2. Issues handling in google sheet format 11
6.6.3. Issues publishing in google calendar format 12
6.7. Metadata handling 12
7. UI PERSPECTIVE COMMON FOR ALL ROLES 12
7.1. Performance 12
7.1.1. Page load times 12
7.2. Mobile 12
7.3. Projects switching 12
7.4. Time management 12
7.4.1. Copy an issue-tracker instance issue to a google calendar event 12
7.5. Issues listing in table format 12
7.5.1. Automatic issue items sequencing 12
7.5.2. Issues re-ordering by desired or defaut attribute in list view 13
7.5.3. issues list default row height 13
7.6. Issues data transfer between different projects 13
7.7. Issues export to Google calendar 13
7.8. Issues import from Google calendar 13
7.9. Access issues txt format from email 13
7.10. Access issues data from Google sheet 13
7.10.1. Apply publish filter while posting to Google Sheet 13
8. UI DEVELOPER PERSPECTIVE 13
8.1. Testability 13
8.2. Multiple control paths for single web action entry point 13

ISSUE-TRACKER USERSTORIES AND SCENARIOIS

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 wanto 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 wanto 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 wanto 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 wanto 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 wanto 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 wanto 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 wanto 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 betweeen different projects
I wanto to be able to see the current project name from any interface I am working in quickly and esily

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 wanto 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 +-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 +-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 wanto 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 wanto to be able decide which users to which projects whill have access to

1.3.1. Users management

As an team leader
In order to keep my business data secure
I wanto to be able decide which users to which projects will have access to

1.3.2. Add new users to a project

As an team leader
In order to be able to add new users into a project
I wanto to be able to add them via the UI only by their e-mail address.

1.3.3. Update existing users in a project

As an team leader
In order to be able to update the existing users from a project
I wanto to be able to update their details via the UI only by their e-mail address.

1.3.4. Delete existing users from a project

As an team leader
In order to be able to delete the existing users from a project
I wanto to be able to delete their details via the UI only by their e-mail address.

1.3.5. Search for users in a project

As an team leader
In order to be able to search the existing users from a project
I wanto to be able to search their details via the UI only by their e-mail address.

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 posible efficiency during the work on one or many projects
I wanto 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 wanto 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 wanto 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 wanto 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 wanto 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 wanto 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 wanto to be able to keep track what was planned on a project term - day,week,month,year,quinquennial or decade

2.2. 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

2.2.1. 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 omnibox UI interface.

2.2.2. 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

2.2.2.1. 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

2.2.2.2. 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

2.2.3. 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.

2.2.4. 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.

2.2.5. 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.

2.3. Measure success

As a team member
In order to measure the success of the planned issues
I wanto to be able to measure the deliverables of each issue by comparable metrics.

2.4. Monitor success

As a team member
In order to monitor the success of the planned issues
I wanto to be able to monitor the metrics of the issues.

2.5. Time management

As an issues-manager
In order to be prepared for issues such as (events , tasks) which have start and stop time
I wanto to be able to save their start_time and stop_time per issue in every possible interface

2.5.1. 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 wanto 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

```
bash src/bash/issue-tracker/issue-tracker.sh -a increase-date -d today
bash src/bash/issue-tracker/issue-tracker.sh -a increase-date -d tomorrow
```

2.5.2. time centric reporting

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

```
bash src/bash/issue-tracker/issue-tracker.sh -a increase-date -d yesterday
bash src/bash/issue-tracker/issue-tracker.sh -a increase-date -d yesterday
```

2.6. 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.

2.7. 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 wanto 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 quickly take into use a new product instance of the issue-tracker application running on a separate host accessible via ssh
I wanto to run a single deploy-host action - which will installl the required OS libraries , Postgres and Perl modules for the operation of the tool silently.

```
# https://serverfault.com/questions/364452/silent-and-scripted-install-of-cpan-and-perl-modules
# https://serverfault.com/a/815650/33346
```

4.2. System performance

As the SysAdmin
In order to ensure the performance of the issue-tracker application
I wanto 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 wanto 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 wanto 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 wanto to generally perform any command the system within the sysadmin scope via clean and memorable oneliners

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. 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

5.1.1. 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

5.2. 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

5.2.1. 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 wanto the easily verify and test parts or the whole System
by issuing a single shell call.

6.1.1. Clarity and breavity 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 fow 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 teists
I want to be able to configure the single e2e entry point script to skip certain tests, but report me what was skipped.

6.1.3. Control flow logging

As a CLI user
In order to be able to understand what the issue tracker tool is executing
I wanto to have configurable logging with stderr, stdout and file output

6.1.4. 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.

6.1.5. 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 wanto to be able to translate the recorded uuid's in the execution run log to their respective records

6.1.6. 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 userstory a single test

6.1.6.1. UUID tracability for test files and userstories

As a Developer
In order to identify each userstory to be tested with its according test
I want to be able to track each userstory or test code entry point file by UUID.

6.1.7. Components start run message print

As a CLI user
In order to know when a component has been started
I wanto to be able to see the "START <<COMPONENT_NAME>> on either the STDOUT or the log file of the component

6.1.8. Tool exit with exit code and exit message

As a CLI user or calling 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 wanto to get the exit code from the tool execution and see the exit message

6.1.9. Execution path tracing by UUID's

As a DevOps operator
Foreach execution run of the tool
I want to be able to walk trough the execution path of the tool programatically.

6.2. Manage details of issues with a single txt file

As a CLI user
In order to be able quickly to view my issues
I wanto to be able to update the names , statuses and categories of my daily,weekly,monthly, yearly and decadally issues by simply editing the issues term file

6.2.1. 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 wanto to be able to store large quantity of issues txt files by their date on daily, weekly, monthly, yearly and decadally basis

6.2.2. 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 wanto 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

6.2.3. Issues file open

As a CLI user
In order to be able quickly to access my issues (daily , weekly, monthly , yearly)
I wanto to be able to view my daily issues by simply opening a single txt file

6.2.4. Issues files history

As a CLI user
In order to be able quickly to search trough old issues
I wanto to be able to access old issues files by their date held in their file names from the file system

6.2.5. Issues files naming conventions

As a DevOps
In order to be able quickly to access and manage programatically issues
I wanto to be able to quess the file paths of the issues file by their date

6.2.5.1. Issues files naming conventions for the project

As a DevOps
In order to be able quickly to switch between different projects

I wanto 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

6.2.5.2. 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 wanto to have the current registration date in the file name and path
<<issue_tracker_project>>-issues.<<current-iso-date>>.<<daily|weekly|monthly|yearly>>.txt

6.2.5.3. 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 wanto 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. 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 filee
I wanto 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 taable in db
db-to-txt - to load a txt file from db table

6.3.1. 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

6.3.1.1. 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 than 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

6.3.2. 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

6.3.3. 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

6.3.3.1. Load issues by xls-to-db action for insert or upset

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

6.3.3.2. 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

6.3.4. Load issues by db-to-

txt

As a cli user of the issue-tracker application
in order to store my isssues 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

6.3.4.1. 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

6.3.4.2. 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

6.3.4.3. 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

6.3.5. Load issues file from db to file system

As a cli user of the issue-tracker application
in order to store my isssues 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

6.4. 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

6.5. 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

6.6. 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

6.6.1. 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 wanto to be able to issue a single shell call for copying the current items data to email by specifying the tables to be published

6.6.2. 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 wanto 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

6.6.3. 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 wanto to be able to issue a single shell call for copying the current items data to google calenda by specifying the tables to be published for the items having set start_time and stop_time attributes.

6.7. Metadata handling

As a DevOps
In order to be able to programatically manage all aspects of my data
I wanto to have a single entry point to manage the meta data per tables , colums and UI elements
so that even a table, column or whatever object is not popullated 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. Performance

As an UI user of the issue-tracker application
In order to enjoy the usage of the tool and interact efficiently
I wanto to to have responsive and quick UI.

7.1.1. 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 wanto to to have a maximum page load time on efficient network less than 0.5 seconds and preferably even 0.3 seconds

7.2. Mobile

As an UI user of the issue-tracker application
In order to enjoy to be able to access it quickly on the go
I wanto to be able to use the same UI on an advanced mobile phones.

7.3. Projects switching

As an issue-tracker ui user
In order to be able to quickly swith between projects
I wanto to be able to access a web page providing autocomplete to preloaded configuration entries for the different projects

7.4. Time management

As an issue-tracker ui user
In order to be prepare for issues such as (events , tasks) which have start and stop time
I wanto to be able to view the issues with the same title, start_time and stop_time in google calendar

7.4.1. 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 wanto to be able to copy via the ui an issue as a new google calendar event

7.5. Issues listing in table format

As an UI user of the issue-tracker application
In order to quickly display as much issues as possible
I want to be able to list the issues per period.

7.5.1. 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 defaut incremental sequence unless I have specified my own sequence.

7.5.2. Issues re-ordering by desired or defaut attribute in list view

As a UI user
In order to prioritize and re-arrange to a logical sequence my issues
I wanto to be able to drag and drop issues up and down ,
which would correspondingly increase or decrease their attribute to which they are currently sorted or ordered by.

7.5.3. issues list default row height

As a UI user
In order to quickly comprehend the data in the lists
I wanto 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

7.6. Issues 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 issues data from one project to another via the UI

7.7. 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 wanto to be able to export my issues to Google calendar

7.8. Issues import from Google calendar

As the UI user of an issue-tracker instance
In order to be able to visualize and manage my my start- and stop_time having issues better
I wanto to be able to import my Google calendar issues into my issue-tracker profile on an issue-tracker instance

7.9. 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 wanto to be able to send each period txt file from the daily folder via gmail.

7.10. 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 wanto to be able to access , edit and update the issues data from google sheeet

7.10.1. 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 wanto 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 wanto to have user friendly development experience.

8.1. Testability

As the UI Developer
In order to deliver working ui units
I wanto to be able to quickly setup the existing project with minimalistic default set of data.

8.2. Multiple control paths for single web action entry point

–