HORIZONS 2020 PROGRAMME

Research and Innovation Action – FIRE Initiative

Call Identifier:	H2020–ICT–2014–1
Project Number:	643943
Project Acronym:	FIESTA-IoT
Project Title:	Federated Interoperable Semantic IoT/cloud Testbeds and Applications

Ecosystem Planning and Reporting V1

Document Id:	FIESTA-IoT-D74-170310-Draft
File Name:	FIESTA-IoT-D74-170310-Draft.pdf
Document reference:	Deliverable 7.4
Version:	Draft
Editor:	Tuan Tran Nhat, Martin Serrano
Organisation:	National University of Ireland Galway – NUIG-Insight
Date:	10 / 03 / 2017
Document type:	Report, Other
Dissemination level:	PU

Copyright © 2017 FIESTA-IoT Consortium: National University of Ireland Galway – NUIG-Insight / Coordinator (Ireland), University of Southampton IT Innovation – ITINNOV (United Kingdom), Institut National de Recherche en Informatique & Automatique – INRIA (France), University of Surrey – UNIS (United Kingdom), Unparallel Innovation, Lda – UNPARALLEL (Portugal), Easy Global Market – EGM (France), NEC Europe Ltd. – NEC (United Kingdom), University of Cantabria – UNICAN (Spain), Association Plateforme Telecom – Com4innov (France), Athens Information Technology – AIT (Greece), Sociedad para el desarrollo de Cantabria – SODERCAN (Spain), Ayuntamiento de Santander – SDR (Spain), Fraunhofer Institute for Open Communications Systems – FOKUS (Germany), Korea Electronics Technology Institute KETI (Korea). The European Commission within HORIZON 2020 Program funds the FIESTA-IoT project.

PROPRIETARY RIGHTS STATEMENT

This document contains information, which is proprietary to the FIESTA-IoT Consortium.
Neither this document nor the information contained herein shall be used, duplicated or communicated by any means to any third party, in whole or in parts, except with prior written consent of the consortium.

DOCUMENT HISTORY

Rev.	Author(s)	Organisation(s)	Date	Comments
V01	Tuan Tran Nhat	NUIG	2017/27/02	Initial ToC
V02	Tuan Tran Nhat	NUIG	2017/27/02	Content provided Sectiosn 1, 2, 3
V03	Tuan Tran Nhat	NUIG	2017/27/02	Documentation included
V04	Tuan Tran Nhat	NUIG	2017/28/02	Design Architecture provided
V05	Martin Serrano	NUIG	2017/03/05	Technical Reviews
V06	Aqeel H. Kazmi	NUIG	2017/03/08	Quality Reviews
V07	Martin Serrano	NUIG	2017/03/10	Format Update and Final Checks
V08	Martin Serrano	NUIG	2017/03/10	Circulated for Approval
Draft	Martin Serrano	NUIG	2017/03/10	EC Submitted

Copyright © 2016 FIESTA-IoT Consortium

TABLE OF CONTENTS

LIST OF FIGURES

EXECUTIVE SUMMARY

This document presents the first version of the FIESTA-IoT Ecosystem Planning and Reporting Tool that will provide the contact point for the experimenters and testbed integrators of the FIESTA-IoT platform. The FIESTA-IoT Planning and Reporting is also the support dimension of the FIESTA-IoT experimentation, services and testbeds integration activities. This deliverable provides the design, specification of the system and the deployment instructions and provide the interaction support lifecycle for Fiesta-IoT testbeds, and experimenters.

The Ecosystem Planning and Reporting also named as Ecosystem support t is an integrated system that has different modules. These modules provide complementary features to each other. At the front end, Ecosystem Support has a point of contact for users to send their questions, after that the expert team can answer their questions and a ticketing management tool that tracks and organizes issues for faster resolution. The Ecosystem Planning and Reporting tools also have a feature that aggregates and organizes questions and answers into a knowledge base. Additionally the system can have multiple points of contact, dashboard and analytics section. It also includes features that allow agents to escalate issues to a higher level.

1 INTRODUCTION

Ecosystem Support (Planning and Reporting) is a point of contact for users to send request in the form of questions, by using the ecosystem support the FIESTA-IoT expert team can answer their questions using a backend ticket management tool that tracks and organizes issues for faster resolution. The ecosystem support also have a feature that aggregates and organizes questions and answers into a knowledge base, such as FAQs. The system backend system can have multiple points of contact, a dashboard and analytics tools. It also has a feature that allows agents to escalate issues to higher levels.

In the background, there is analytical processing and automated processes, which provide more advanced features such as correlate multiple contacts, automatic reporting tools, and intelligent ticket management tool. The automated tasks supported by the Ecosystem Support are ticket categorization and prioritization, alerts and notifications, ticket status management. Along this line, the Ecosystem Support can track and understand how visitors are interacting with the website in real time. Therefore, it can use that insight to trigger rule-based proactive chat conversations. As results, it can identify new, returning visitors and start chats based on visitor activities and their questions. It also allows the file transfer from the support team to users.

Such integrated frontend and backend features are packaged in three modules: Mantis module, Live Chat module and Question & Answer module. The Mantis module is mainly responsible for interfacing with Mantis database which managing tickets. The Live Chat module is responsible for managing the live chat features including storing and retrieving chat messages. The Question & Answer module in the Ecosystem Support sets up a page on the WordPress site that lets logged in users submit questions that can be answered by the community, or answered and closed by administrators. Questions can be tagged, and marked as Resolved, Closed, etc. or can be grouped into categories. Questions can also be asked and answered completely on the front-end, though user can make edits to them on the back-end. Moreover, all of these features can be customized, via the plugins, including email notifications, advanced permissions, and more complex organization, but getting everything up and running is fairly simple.

The section 2 in the document describes the design architecture of the ecosystem support system. The architecture description includes the overall structure of the system as well as detailed descriptions of each component composing the whole integrated one. The section 3 includes the deployment guideline will follow. In the deployment guideline will give detailed step-by-step instructions on how to setup up such a system. The section 4 help the user on how to use the system, the guidelines on how to use each of the common features and administrate settings, detailed the instructions with screenshot are provided. The guidelines are given for every module of the integrated system.

2 ECOSYSTEM SUPPORT DESIGN & ARCHITECTURE

2.1 Ecosystem Support Lifecycle

The ecosystem support desk follows the idea design of complementary activities to cover all the range of support services that goes from simply online Questions and Answer section to a full set of on-site assistance, for the case of FIESTA-IoT project the idea of deploying full functionalities is used for the best practice on defining the lifecycle design (Figure 1) however not all the functionalities will be implemented. First because the expected demands of the users in FIESTA-IoT are coming from a developers background and thus it is expected their technical skills are advanced and second because the restriction on budget allocated to this task allows only the remote support. At the figure 1 the lifecycle starts by using online tools support like blogs sections and/or questions and answers section, if the Q&A section is not enough to cover the required support, then a life support online or “live chat” system is enabled to allow interaction with FIESTA-IoT platform experts, the back end of this online support is an open source planning and ticketing system to optimize and control the resolution of registered incidences, i.e. code bugs, questions, support, deployment problems etc. By opening a ticket via the back end technologies an expert is informed and the ticket is assigned to be resolved, if the online expert can solve the inquiry, then the case is mark as resolved and closed. However if the query is more specialized and require a more knowable expert by using the ticketing system the task is escalated and assigned to the appropriate expert, the resolution of the problem can require the participation of a resolution feedback team and /or the technical expert, which he can pass this to a third party if this is not in his technical capacity to resolved it, otherwise the feedback will be provided via the ticketing system that automatically send the response via email. If the response is satisfactory and received correctly by the opener of the case the ticket is closed and the case closed. This FIESTA-IoT Planning and Reporting lifecycle design is totally open source oriented with the main intention to provide insights to explore and receive contributions for extensions and more functionalities in subsequent versions.

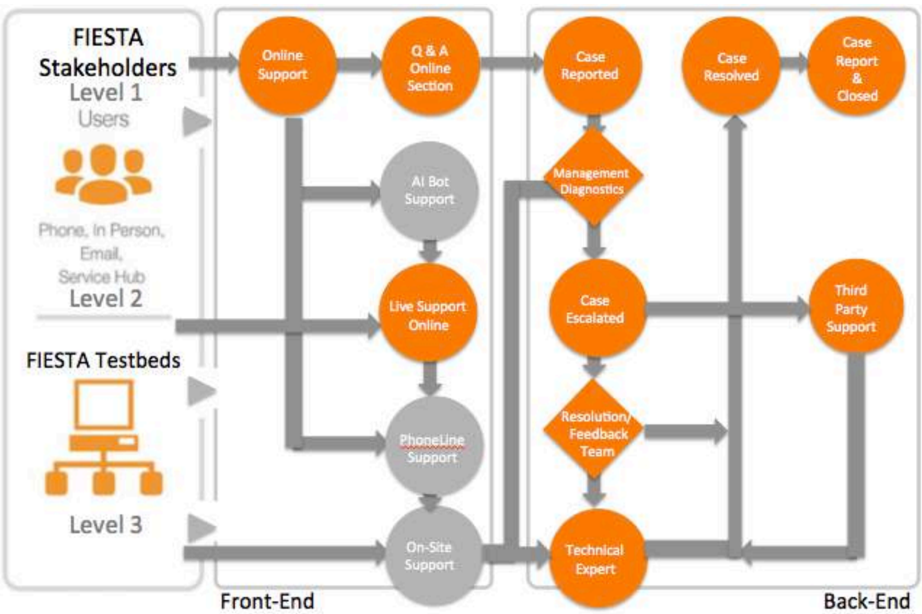

Figure 1 Ecosystem Support Lifecycle Design

2.2 Ecosystem Support Architecture

The architecture of Fiesta ECOSYSTEM SUPPORT has four modules as depicted in Figure 2 below. The main module is ECOSYSTEM SUPPORT, an integrating module that connect the three other modules as follow:

1. Q&A (Question and Answer) module: This module enable users to browse all question and experts answers for helping users to gain a better understanding about FIESTA-IoT platform applications.
2. LIVE CHAT module: This module allows users to chat with the support team.
3. Mantis API module: This module allows create a new ticket to mantis system.

The detailed descriptions of these modules are presented in following sections.

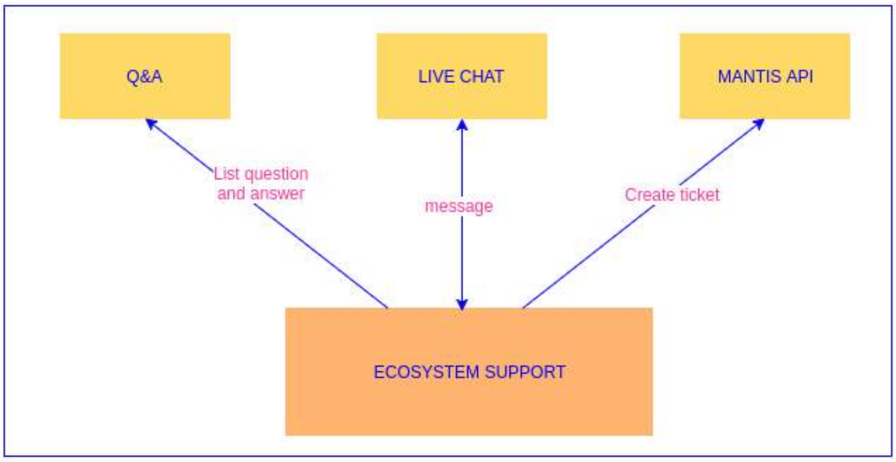

Figure 2 Ecosystem Support Architecture

2.3 Mantis Architecture

The Mantis module provides the interfaces for the Ecosystem Support to create a new ticket in Mantis database. The workflow of interaction is depicted in Figure 3. In this figure, we can see the interaction is carried via node.js to interface with Mantis APIs to access the MANTIS database. Mantis Database stores the organization data of projects, submitted tickets and users' profiles. Via Mantis APIs, the ECOSYSTEM Support can create, delete, and assign, ... a ticket to users that then can be stored in Mantis Database that is then made accessible for different users or application APIs that have access to this database.

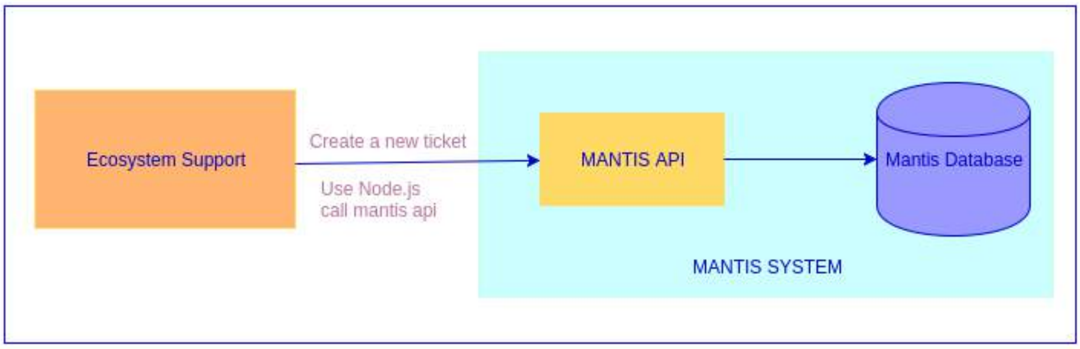

Figure 3 Mantis Architecture

2.4 Live Chat Architecture

Live Helper Chat module is an open source Live Support chat for a website. As illustrated in Figure 4, powered this module, the Ecosystem Support allows website's visitor to chat with expert support team. A chat message will send to PHP core module. After that the expert supports person will answer and PHP core module send the answer message back to the visitor.

Live Chat module stores chat messages in its Live Chat Database. The Live Chat Data also store users' information together with those chat messages. The data in the database is made available to others users and application via APIs.

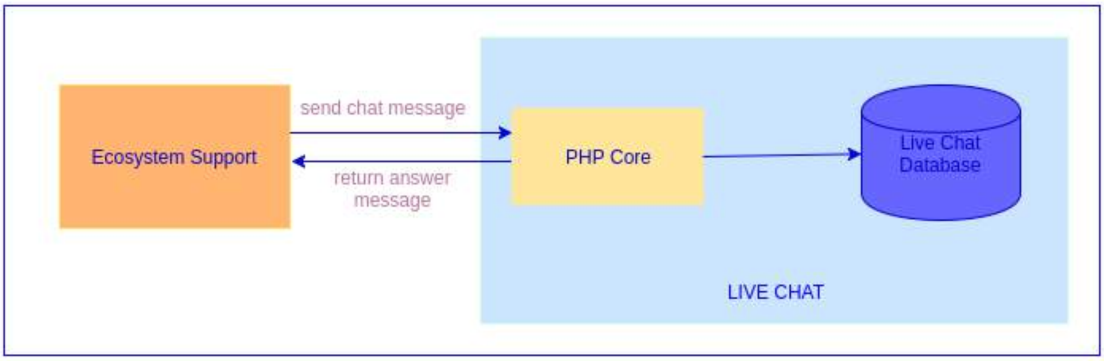

Figure 4 Live Chat Architecture

2.5 Question & Answer Architecture

Question & Answer module is an open source plugin for adding a question and answer section to WordPress website as shown in Figure 5. Question & Answer module allows visitor to view the questions and the answers of the question that they're choose. It also allows an operator to post or delete questions and answers, set permission to users via the Plugin Manager. The Plugin Manager redirects the module to the right place to store data in the WordPress Database such as posts, pages, comments, categories, tags, custom fields, users, and other site options such as site urls etc.

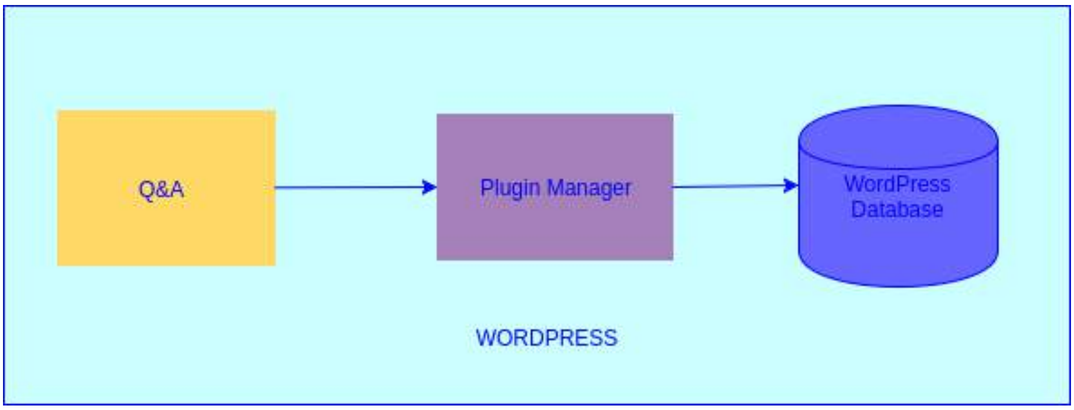

Figure 5 Question & Answer Architecture

3 ECOSYSTEM SUPPORT DEPLOYMENT

3.1 Mantis Deployment Guide

3.1.1 Installation

These section is dedicated to developers and describe in the form of tasks the deployment of MantisBT and should be performed prior to any new installation or upgrade.

1. Unzip mantisbt.zip

```
sudo unzip mantisbt.zip -d /var/www/html/
```

2. From your web browser, access

```
http://server_domain_name_or_IP/mantisbt/admin/install.php
```

Please don't sett Admin username and Admin password

3. Finally, click Install/Upgrade Database button. (Figure 6)

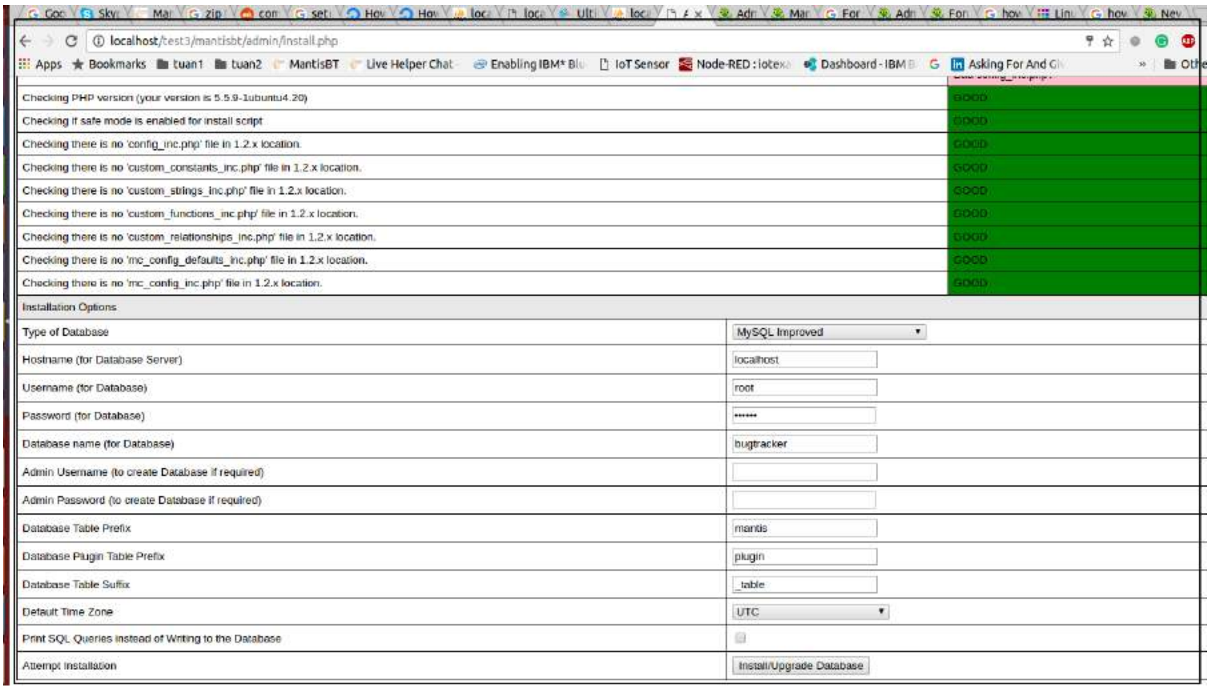

Figure 6 **Mantis Install page**

3. The script attempts to write a basic *config_inc.php* file to define the database connection parameters.

This operation may fail if the web server's user account does not have write permissions to the directory (which is recommended for obvious security reasons). In this case, you will have to manually create the file and copy/paste the contents from the page.

4. Delete the *admin* folder

For security reasons, the scripts within this directory should not be accessible on a live MantisBT site or on any installation that is accessible via the Internet.

Once you have confirmed that the install/upgrade was successful, you should delete this directory

```
rm -r admin
```

5. After a new installation, login to your ticket management use the default Administrator account. The id and password are *administrator / root*.

- Create a new Administrator account: go to Manage > Manage Users and create a new account with 'administrator' access level.
- Change the default Administrator account to 'reporter' access level and change a new password for the default Administrator account.
- Create a new Project: go to Manage > Manage Projects and create a new project with a name is Fiesta.

3.1.2 Upgrading

This section explains how to upgrade an existing MantisBT installation.

1. Put the site down for maintenance

   ```
   cp mantis_offline.php.sample mantis_offline.php
   ```

   This will prevent users from using the system while the upgrade is in progress.
2. Always *Backup your code, data and config files* before upgrading!

   This includes your Mantis directory, your attachments, and your database. Refer to the Backups section for details.
3. Copy the configuration files

   To preserve your system settings, you should copy the files listed below to the new directory, as well as any other custom files such as logo, favicon, css, etc.

   config_inc.php custom_strings_inc.php custom_constants_inc.php custom_functions_inc.php
4. Execute the upgrade script. From your web browser, access

   ```
   http://yoursite/mantisbt/admin/install.php
   ```

5. Provide required information for the upgrade

   High-privileged database account

   Additional privileges required: INDEX, CREATE, ALTER, and DROP

   If this account is not specified, the database user will be used.
6. Click the *Install/Upgrade Database* button.
7. At the end of the upgrade, review and correct any warnings or errors.

3.1.3 Backups

MySQL databases are easy to backup using the *mysqldump* command:

```
mysqldump -u<username> -p<password> <database name>  <output file>
```

To restore a backup you will need to have a clean database. Then run:

```
mysql -u<username> -p<password> <database name> <input file>
```

You can also perform both of these tasks using phpMyAdmin

3.1.4 Uninstall

To uninstall MantisBT:

- Delete the MantisBT directory and all files and subdirectories.
- Drop all MantisBT tables from the database, these can be identified by the configured prefix for the installation. The default prefix is 'mantis'.
- Remove any customizations or additions that you may have made.

If you have the permissions to create/drop databases and you have a specific database for MantisBT that does not contain any other data, you can drop the whole database.

3.2 Live Chat Deployment Guide

3.2.1 Installation

Live Helper Chat is available for deployment on a new server.

In detail, the steps will do the following:

1. Unzip livehelper.zip to /var/www/html :

```
unzip livehelper.zip -d /var/www/html/
```

2. Go to phpMyAdmin and create an empty database with the name is livechat

3. After that you gave permission to write to cache folder (you should see the following the images below)

a. Type *sudo su* and cd to */var/www/html/livehelper*

b. Give write permission to cache folder: *chmod -R 777 cache/*

c. Your browser is pointing to the URL

Copyright © 2016 FIESTA-IoT Consortium

```
http://server_domain_name_or_IP/livehelper/index.php/site_admin/install/install
```

d. You gave recursive write permission to cache folder. (Figure 7)

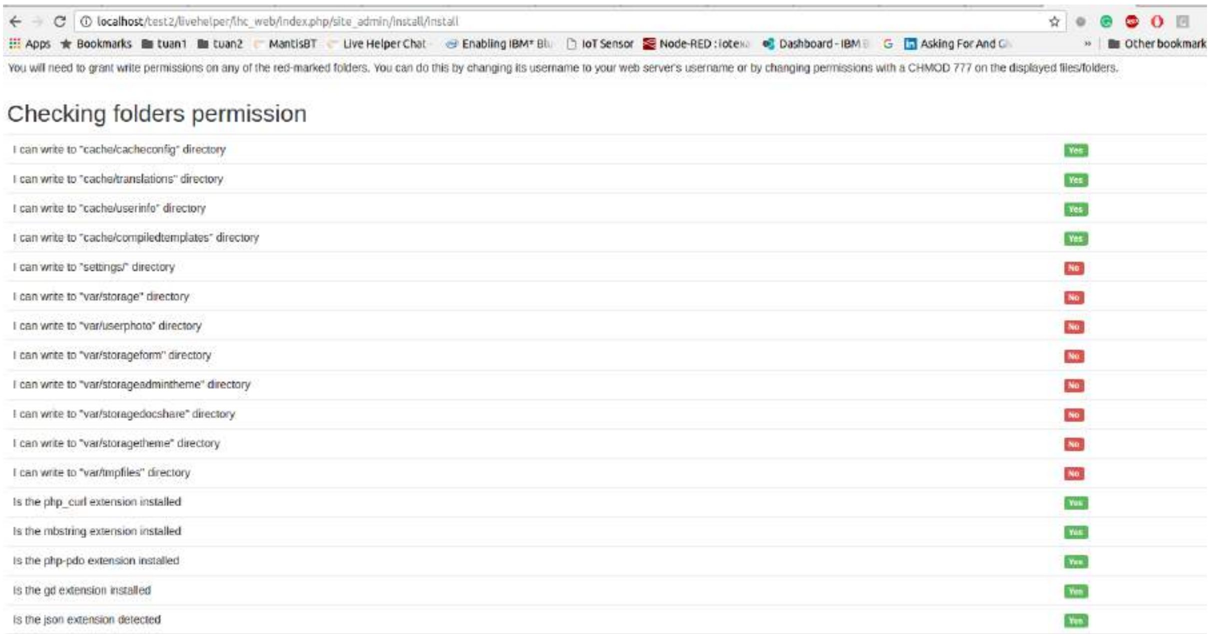

Figure 7 Live Chat Install page

After that you have to give write permission to few other folders also:

```
chmod -R 777 settings/settings.ini.default.php
chmod -R 777 var/storage
chmod -R 777 var/userphoto
chmod -R 777 settings/
chmod -R 777 var/storageform
chmod -R 777 var/storageadmintheme
chmod -R 777 var/storagedocshare
chmod -R 777 var/storagetheme
chmod -R 777 var/tmpfiles
```

After that you should see all items in green colour and click Next button (Figure 8)

Copyright © 2016 FIESTA-IoT Consortium

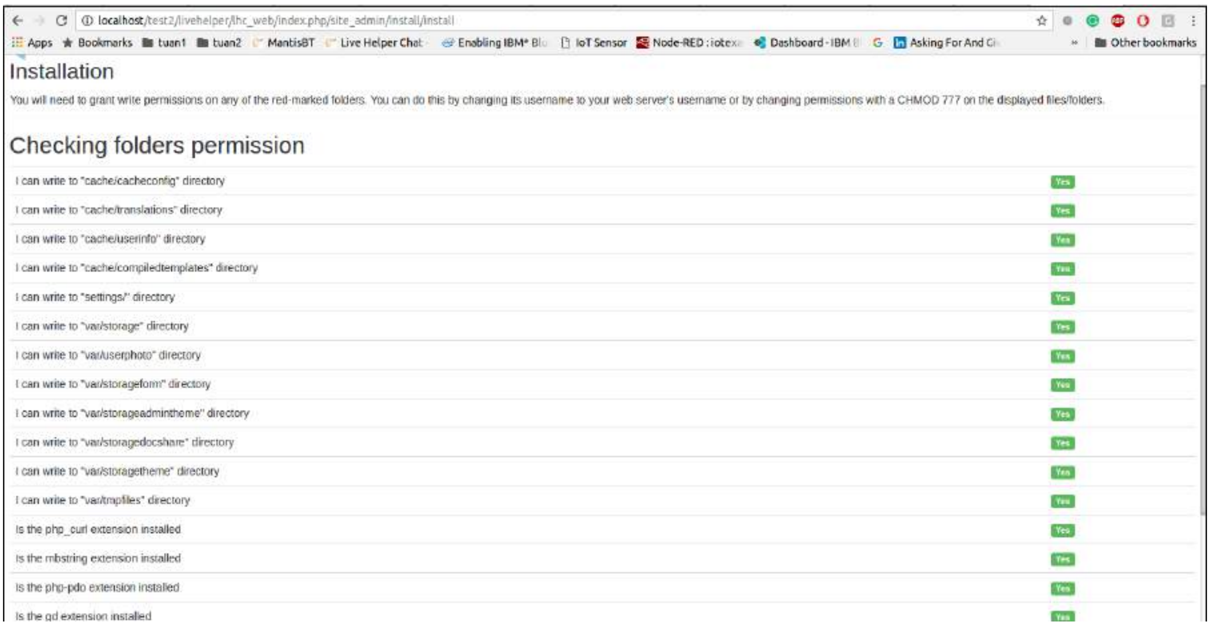

Figure 8 **Live Chat Install page**

Next you have to enter database settings (example: livechat database). Database have to be precreated (Figure 9)

Figure 9 **Database Setting page**

At next step you will see initial application settings. Please setting:

Admin username: administrator

Admin password: (password)

Email: (admin email)

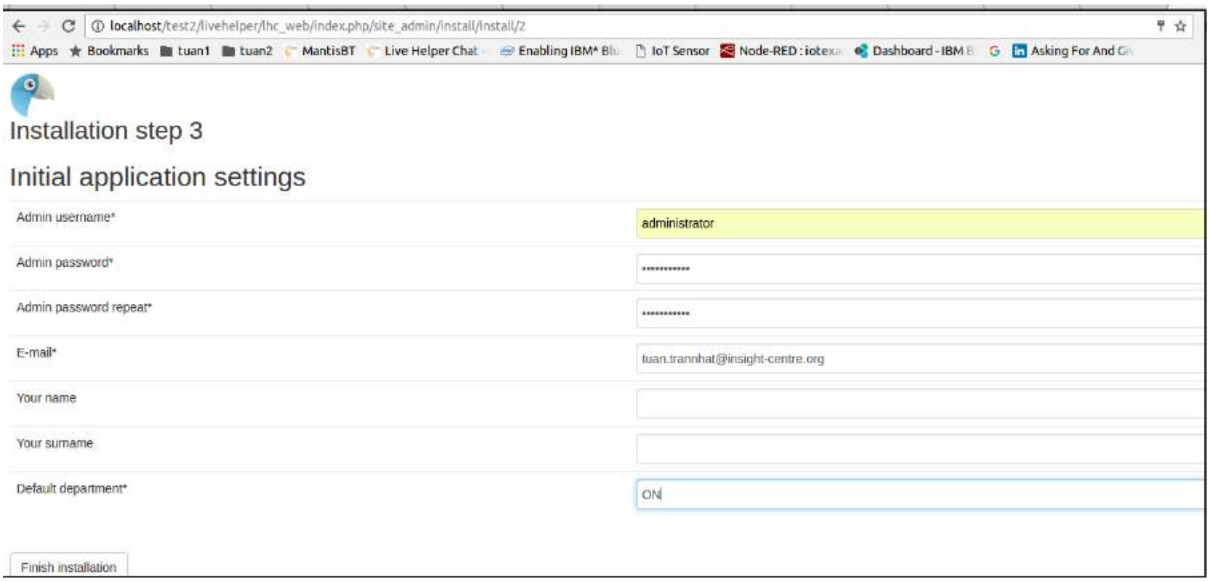

Figure 10 **Application Setting page**

Finally, click Finish installation button.

4. Delete design folder in */var/www/html/livehelper/*

```
sudo rm -r /var/www/html/livehelper/design
```

5. Unzip the new theme in deployment file to lhc_web folder

```
sudo unzip design.zip -d /var/www/html/livehelper/
```

6. Upload the theme file in System Configuration:

a. From your web browser, access

```
http://server_domain_name_or_IP/livehelper/index.php/site_admin/system/configuration#/
chatconfiguration
```

b. Click Import new themes link

c. And select file lhc-theme-1 (1).json in the folder */var/www/html/livehelper/design/defaulttheme/lhc-theme-1*

*d. S*et default theme

7. Clear cache in System Configuration:

a. From your web browser, access

```
http://server_domain_name_or_IP/livehelper/lhc_web/index.php/site_admin/system/config
uration#/system
```

b. Click Clean cache link

Copyright © 2016 FIESTA-IoT Consortium

3.2.2 Backups

MySQL databases are easy to backup using the *mysqldump* command:

```
mysqldump -u<username> -p<password> <database name>  <output file>
```

To restore a backup you will need to have a clean database. Then run:

```
mysql -u<username> -p<password> <database name> <input file>
```

You can also perform both of these tasks using phpMyAdmin

3.2.3 Uninstall

To uninstall Live Chat:

- Delete the Live Chat directory and all files and subdirectories.
- Drop all Live Chat tables from the database livechat.
- Remove any customizations or additions that you may have made.

If you have the permissions to create/drop databases and you have a database livechat that does not contain any other data, you can drop the whole database.

3.3 Add New Ticket Deployment Guide

3.3.1 Installation

In detail, the steps will do the following:

- Unzip CreateTicketMantis.zip
- Edit config.js in the folder CreateTicketMantis (Figure 11)

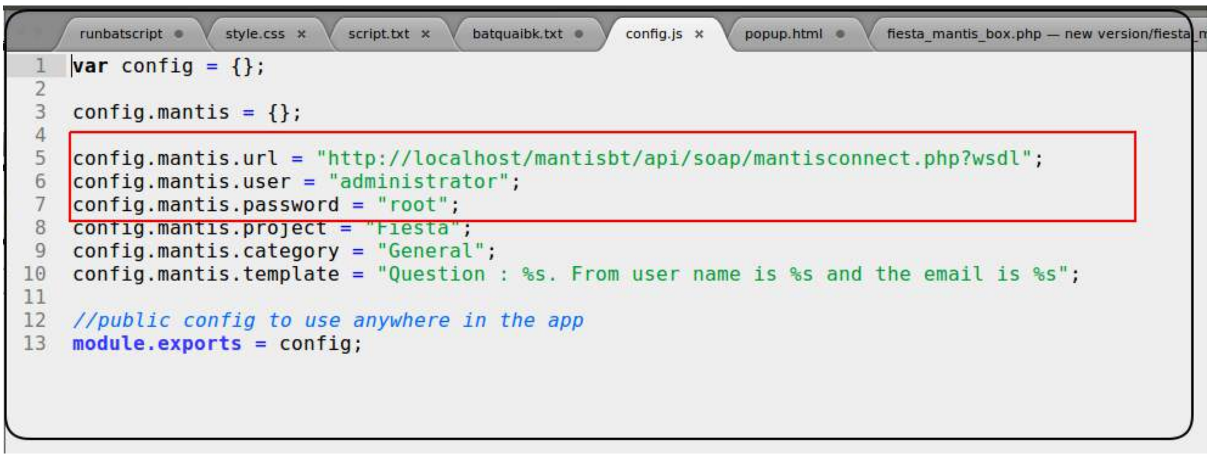

Figure 11 Edit config.js

Copyright © 2016 FIESTA-IoT Consortium

- Edit my-script.js in the folder CreateTicketMantis/html/js/ (at row 5)

```
'http://hostserver:port/ticket/create',
```

Example: '*http://140.203.154.171:8030/ticket/create'*

- Use node.js to run index.js in the folder CreateTicketMantis

```
node index.js
```

Note: Keeping Create a Mantis Ticket module running with Forever

The purpose of Forever is to keep a child process (such as your node.js web server) running continuously and automatically restart it when it exits unexpectedly.

Forever exposes its functionality via a command line interface available after installation via npm:

```
[sudo] npm install forever -g
```

The usage options for Forever expose four simple command line tasks: start, stop, stopall, list.

You can use Forever as a long running process by omitting the 'start' option. With these tasks available, starting a nodejs script with Forever is simple:

```
$ forever start index.js
```

- In the server, run this command to setup PHP SOAP

```
[sudo]  apt-get install php5.6-soap
```

- Go to FIESTA Plugins --> Add New, find and upload fiesta_mantis_box.zip
- Activate the plugin through the 'Plugins' menu in WordPress
- Go to FIESTA dashboard, cilck Appearance and click Menus
- After that , click Fiesta Mantis Box and select Sidebar 2
- Finally click Add Widget button (Figure 12)

Copyright © 2016 FIESTA-IoT Consortium

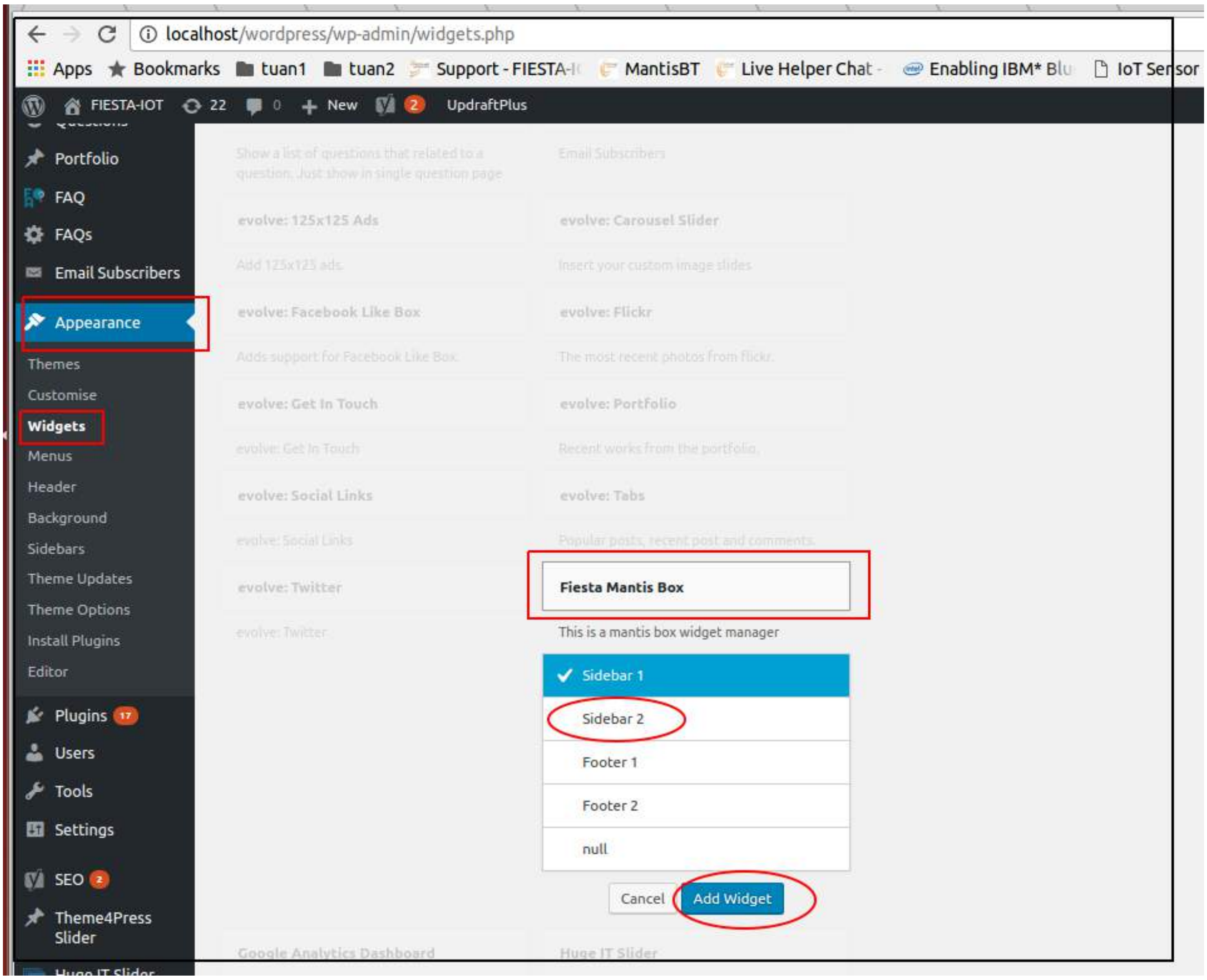

Figure 12 Setting Fiesta Mantis Box in Wordpress

- Click Save.
- In FIESTA dashboard click edit Support page and select Sidebar 2 in Page Option – Sidebar.

3.3.2 Uninstall

To uninstall the Add New Ticket module, please follow the instruction below:

- Delete the CreateTicketMantis directory and all files and subdirectories.
- Go to FIESTA Plugins --> Installed Plugins, find the Fiesta Mantis Box Widget plugin.
- Click on "Deactivate" link.
- After deactivating the plugin, continue to click on "Delete" link to delete the plugin.

3.4 Question and Answer Deployment Guide

3.4.1 Installation

In detail, the steps will do the following:

- Go to FIESTA Plugins --> Add New, find and upload dw-question-answer.zip
- Activate the plugin through the 'Plugins' menu in WordPress.
- Log In to your WordPress Dashboard and go to menu Dashboard -> Questions -> Settings then choose pages where to put submit question form and list questions page.

3.4.2 Uninstall

To uninstall the DW Question & Answer plugin, please follow the instruction below:

- Go to Plugins --> Installed Plugins, find the DW Question & Answer plugin.
- Click on "Deactivate" link. (Figure 13)

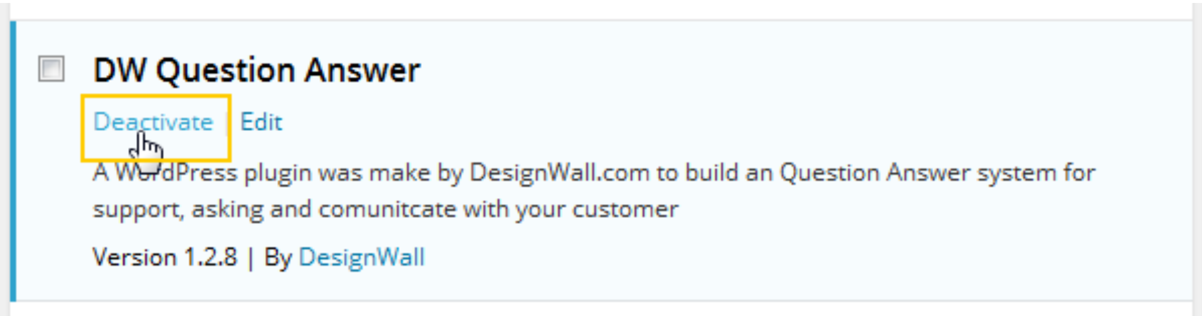

Figure 13 Deactivate DW Question & Answer

- After deactivating the plugin, continue to click on "Delete" link to delete the plugin. (Figure 14)

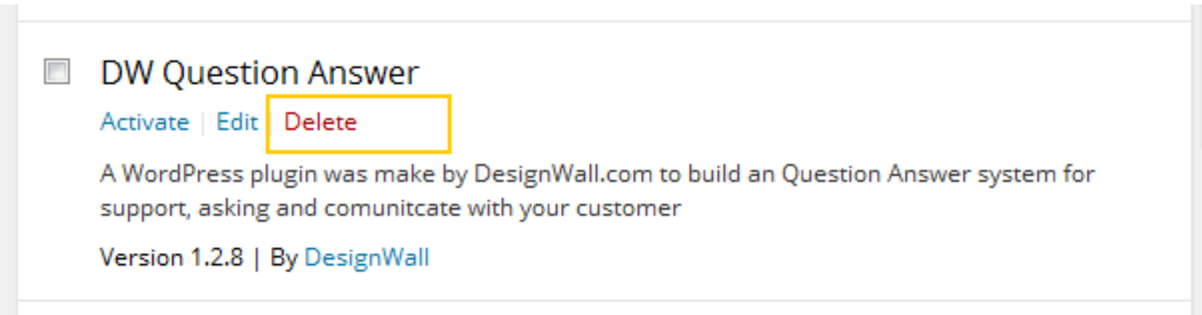

Figure 14 Delete DW Question & Answer

Copyright © 2016 FIESTA-IoT Consortium

4 ECOSYSTEM SUPPORT GUIDE

4.1 Mantis Administration Guide

4.1.1 Introduce

MantisBT is a web-based bug tracking system that was first made available to the public in November 2000. Over time it has matured and gained a lot of popularity, and now it has become one of the most popular open source bug/issue tracking systems. MantisBT is developed in PHP, with support to multiple database backends including MySQL, MS SQL, PostgreSQL and DB2.

MantisBT, as a PHP script, can run on any operating system that is supported by PHP and has support for one of the DBMSes that are supported. MantisBT is known to run fine on Windows, Linux, OS/2, Mac OS X, System i and a variety of Unix operating systems.

4.1.2 Creating User Accounts

In MantisBT, there is no limit on the number of user accounts that can be created. Typically, installations with thousands of users tend to have a limited number of users that have access level above REPORTER.

By default users with ADMINISTRATOR access level have access to create new user accounts. The steps to do that are:

- Click "Manage" on Main Menu.
- Click "Manage Users" (if not selected by default).
- Click "Create New Account" button just below the alphabet key.
- Enter user name, email address, global access level (more details about access levels later). Other fields are optional.
- Click "Create Users". (Figure 15)

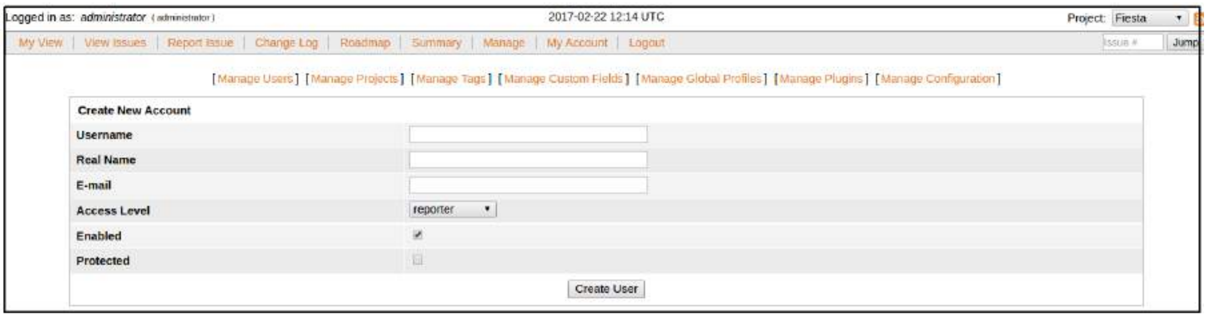

Figure 15 Create New Account page

4.1.3 Deleting User Accounts

By default administrators are the only users who can delete user accounts. They can delete accounts by clicking Manage, Manage Users, locating the user to be deleted and opening it details page, then clicking on the "Delete User" button that deletes the user. (Figure 16)

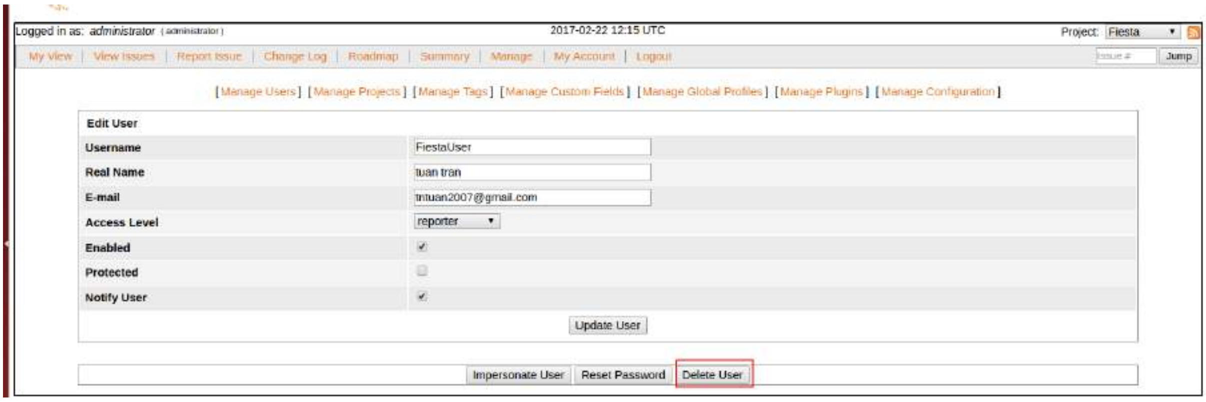

Figure 16 Delete User Accounts

4.1.4 Issue Creation

An issue can be created via one of the following channels:

- MantisBT Web Interface - This is where a user logs into MantisBT and reports a new issue. (Figure 17)
- SOAP API - Where an application automatically reports an issue into MantisBT using the SOAP API web services interfaces. For example, the nightly build script can automatically report an issue if the build fails.
- Email - This is not supported out of the document, but there are existing MantisBT patches that would listen to emails on pre-configured email addresses and adds them to the MantisBT database.
- Others - There can be several other ways to report issues. For example, applications / scripts that directly injects issues into MantisBT database (not recommended, except for one-off migration scripts), or PHP scripts or Node.js scripts that use the core MantisBT API to create new issues.

The steps to create a new issue are:

- Log into Mantis
- Click on the option "Report Issue".
- Select Assign To
- Enter Category.
- Enter Summary.
- Enter Description.
- Enter Steps to reproduce.
- Click Submit Report.

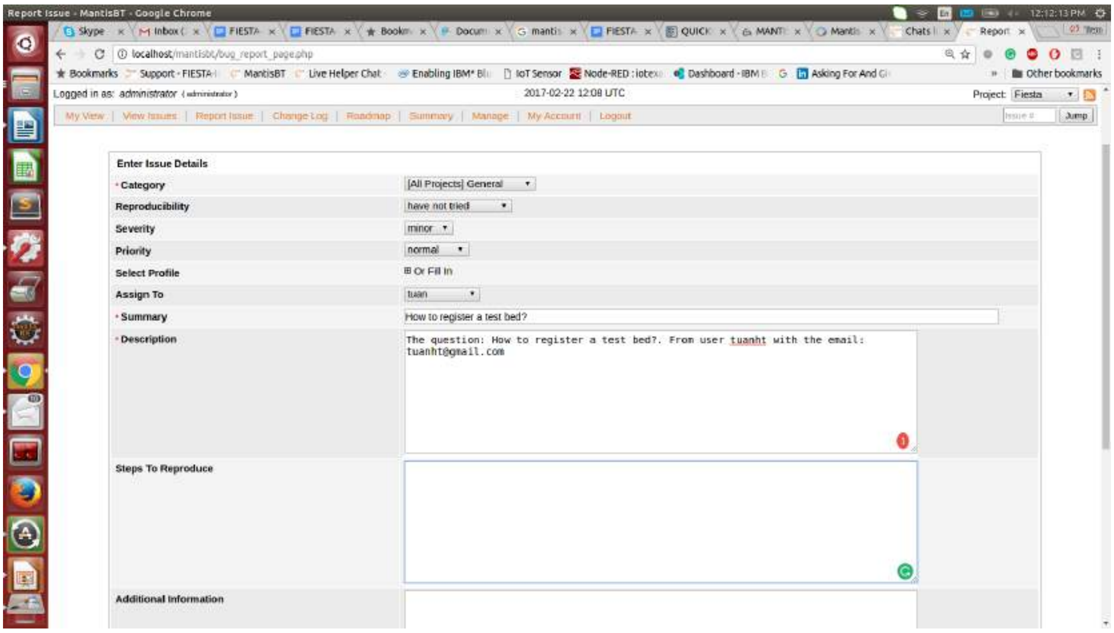

Figure 17 Report Issue page

4.1.5 Issue Statuses

An important part of issue tracking is to classify issues as per their status. Each team may decide to have a different set of categorization for the status of the issues, and hence, MantisBT provides the ability to customize the list of statuses. MantisBT assumes that an issue can be in one of three stages: opened, resolved and closed. Hence, the customized statuses list will be mapped to these three stages. For example, MantisBT comes out of the box with the following statuses: new, feedback, acknowledged, confirmed, assigned, resolved and closed. In this case "new" -> "assigned" map to opened, "resolved" means resolved and "closed" means closed. (Figure 18)

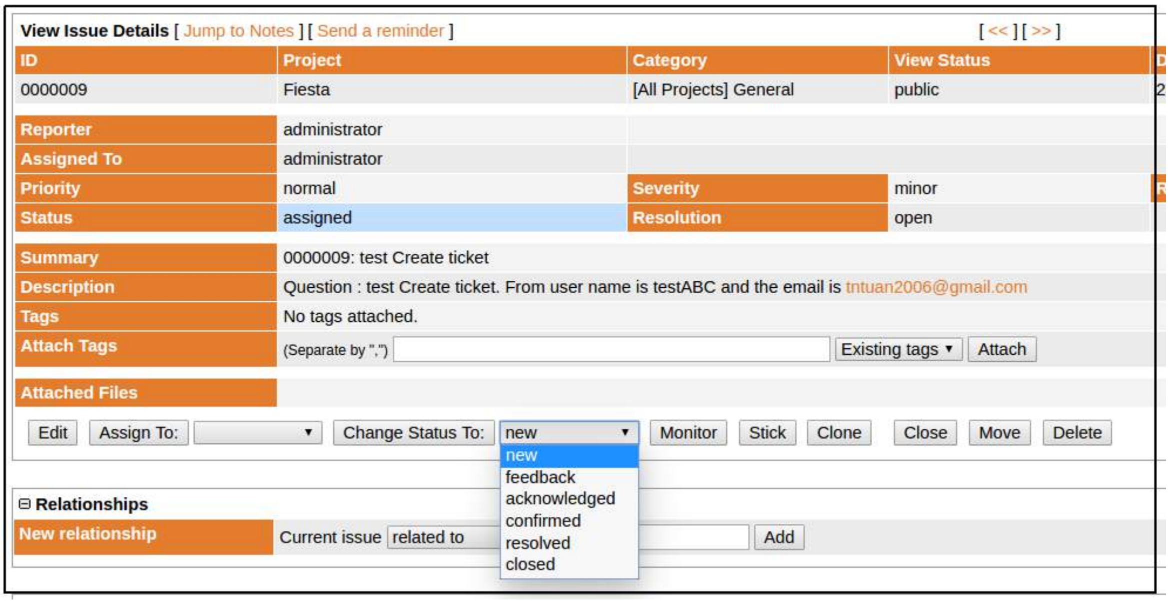

Figure 18 View Issue Details page

Following is the explanation of what the standard statuses that are shipped with MantisBT means.

- **New** - This is the landing status for new issues. Issues stay in this status until they are assigned, acknowledged, confirmed or resolved. The next status can be "acknowledged", "confirmed", "assigned" or "resolved".
- **Acknowledged** - This status is used by the development team to reflect their agreement to the suggested feature request. Or to agree with what the reporter is suggesting in an issue report, although they didn't yet attempt to reproduce what the reporter is referring to. The next status is typically "assigned" or "confirmed".
- **Confirmed** - This status is typically used by the development team to mention that they agree with what the reporter is suggesting in the issue and that they have confirmed and reproduced the issue. The next status is typically "assigned".
- **Assigned** - This status is used to reflect that the issue has been assigned to one of the team members and that such team member is actively working on the issue. The next status is typically "resolved".
- **Resolved** - This status is used to reflect that the issue has been resolved. An issue can be resolved with one of many resolutions (customizable). For example, an issue can be resolved as "fixed", "duplicate", "won't fix", "no change required", etc. The next statuses are typically "closed" or in case of the issue being re-opened, then it would be "feedback".
- **Closed** - This status reflects that the issue is completely closed and no further actions are required on it. It also typically hides the issue from the View Issues page. Some teams use "closed" to reflect sign-off by the reporter and others use it to reflect the fact that the fix has been released to customers.

4.1.6 Workflow

By default, there is no workflow defined, which means that all states are accessible from any other, by anyone.

The "Manage > Manage Configuration > Workflow Transitions" page allows users with ADMINISTRATOR access level to do the following tasks:

- Define the valid next statuses for each status.
- Define the default next status for each status.
- Define the minimum access level required for a user to transition to each status.
- Define the default status for newly created issues.
- Define the status at which the issue is considered resolved. Any issues a status code greater than or equal to the specified status will be considered resolved.
- Define the status that is assigned to issues that are re-opened.
- Define the required access level to change the workflow.

Note that the scope of the applied change is dependent on the selected project. If "All Projects" is selected, then the configuration is to be used as the default for all projects, unless overridden by a specific project. To configure for a specific project, switch to it via the combobox at the top right corner of the screen.

The Global ("All Projects") workflow can also be defined in the *config_inc.php* file, as per the following example.

```
$g_status_enum_workflow[NEW]='30:acknowledged,20:feedback,40:confirmed,50:assigned,80:resolved';
$g_status_enum_workflow[FEEDBACK] = '30:acknowledged,40:confirmed,50:assigned,80:resolved';
$g_status_enum_workflow[ACKNOWLEDGED]='40:confirmed,20:feedback,50:assigned,80:resolved';
$g_status_enum_workflow[CONFIRMED] = '50:assigned,20:feedback,30:acknowledged,80:resolved';
$g_status_enum_workflow[ASSIGNED ]= '80:resolved,20:feedback,30:acknowledged,40:confirmed';
$g_status_enum_workflow[RESOLVED] = '90:closed,20:feedback,50:assigned';
$g_status_enum_workflow[CLOSED] = '20:feedback,50:assigned';
```

4.1.7 Page descriptions

4.1.7.1 Login page

Just enter your username and password and hit the login button (Figure 19). There is also a Save Login checkbox to have the package remember that you are logged in between browser sessions. You will have to have cookies enabled to login. If the account doesn't exist, the account is disabled, or the password is incorrect then you will remain at the login page. An error message will be displayed. You can also sign-up for a new account. You must supply a valid email address and select a unique username. Your randomly generated password will be emailed to your email account.

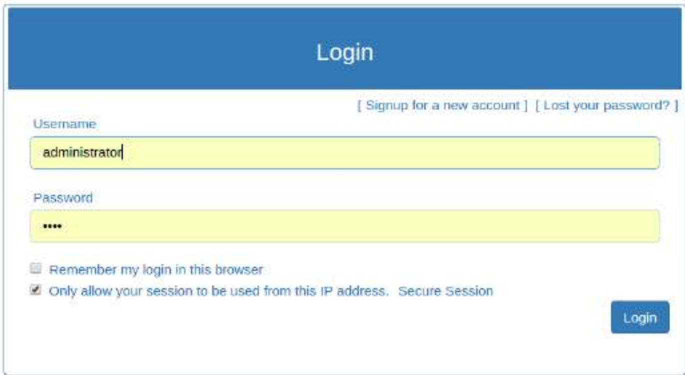

Figure 19 Mantis Login page

4.1.7.2 Main page

This is the first page you see upon logging in. It shows you the latest news updates for the ticket management (Figure 20). This is a simple news module and is to keep users aware of changes in the ticket management or project. Some news postings are specific to projects and others are global across the entire ticket management. This is set at the time of posting in the Edit News section. The number of news posts is controlled by a global variable. When the number of posts is more than the limit, a link to show "older news" is displayed at the bottom. Clicking on the link will bring up the specified article. This listing will also only display items that are either global or specific to the selected project.

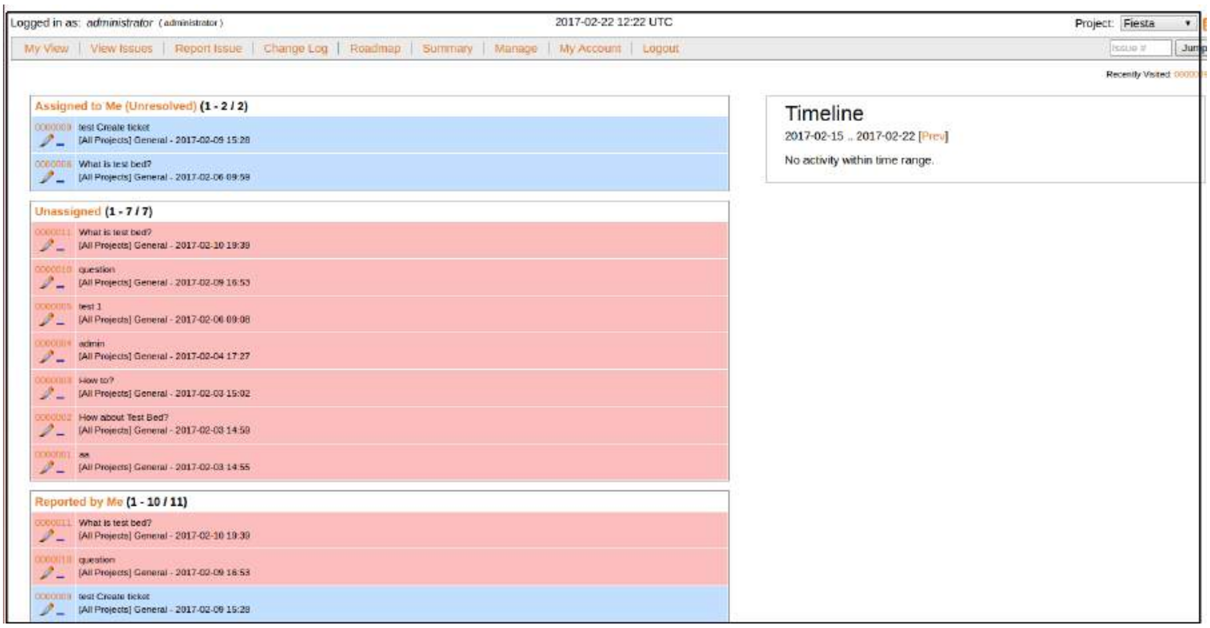

Figure 20 Mantis Main page

4.1.7.3 *View Issues page*

Here we can view the issue listings. The page has a set of viewing filters at the top and the issues are listed below. The filters control the behavior of the issues list. The filters are saved between browsing sessions but do not currently save sort order or direction. If the number of issues exceeds the "Show" count in the filter a set of navigation to go to "First", "Last", "Previous", "Next" and specific page numbers are added. (Figure 21) The Search field will look for simple keyword matches in the summary, description, steps to reproduce, additional information, issue id, or issue text id fields. It does not search through issue notes.

Figure 21 View Issues page

Issue List - The issues are listed in a table and the attributes are listed in the following order: priority, id, number of issue notes, category, severity, status, last updated, and summary. Each (except for number of issue notes) can be clicked on to sort by that column. Clicking again will reverse the direction of the sort. The default is to sort by last modification time, where the last modified issue appears at the top. The issue id is a link that leads to a more detailed report about the issue. You can

also add issue notes here. The number in the issue note count column will be bold if an issue note has been added in the specified time frame.

The addition of an issue note will make the issue note link of the issue appear in the unvisited state. The text in the "Severity" column will be bold if the severity is major, crash, or block and the issue not resolved. The text in the "Updated" column will be bold if the issue has changed in the last "Changed(hrs)" field which is specified in the viewing filters.

Each table row is color coded according to the issue status. The colours can be customised through MantisBT Configuration Severities block - prevents further work/progress from being made crash - crashes the application or blocking, major - major issue, minor - minor issue, tweak - needs tweaking, text - error in the text, trivial - being nit picky, feature - requesting new feature - Status new - new issue, feedback - issue requires more information from reporter, acknowledged - issue has been looked at but not confirmed or assigned, confirmed - confirmed and reproducible (typically set by an Updater or other Developer), assigned - assigned to a Developer, resolved - issue should be fixed, waiting on confirmation of fix, closed - issue is closed, Moving the mouse over the status text will show the resolution as a title. This is rendered by some browsers as a bubble and in others as a status line text.

4.1.7.4 Issue View page

Here is the simple listing of the issue report. Most of the fields are self-explanatory. "Assigned To" will contain the developer assigned to handle the issue. Priority is fully functional but currently does nothing of importance. Duplicate ID is used when an issue is a duplicate of another. It links to the duplicate issue which allows users to read up on the original issue report. (figure 22)

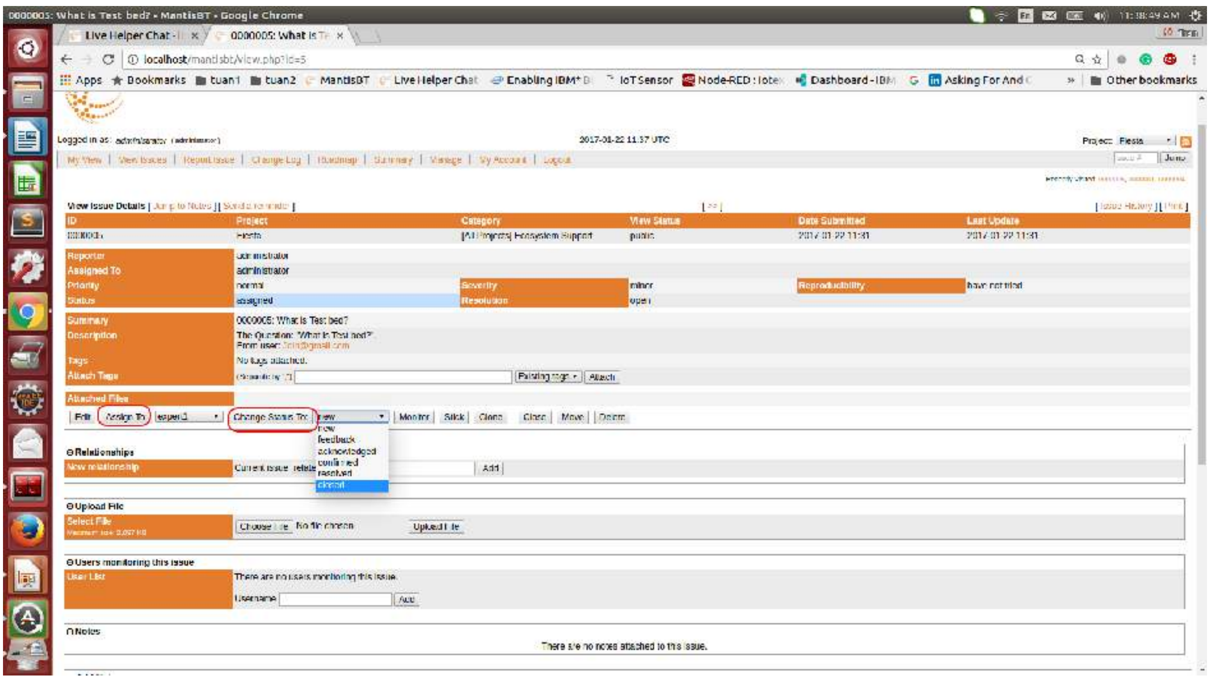

Figure 22 Issue view page

The issue report is a set of buttons that a user can select to work on the issue.

- Update Issue - brings up a page to edit all aspects of the issue

- Assign to - in conjunction with the drop-down list next top the button, this is a shortcut to change the assignment of an issue
- Change Status to - in conjunction with the drop-down list next top the button, this is a shortcut to change the status of an issue. Another page (Change Status) will be presented to allow the user to add notes or change relevant information
- Monitor / Unmonitor Issue - allows the user to monitor any additions to the issue by email
- Create Clone - create a copy of the current issue. This presents the user with a new issue reporting form with all of the information in the current issue filled in. Upon submission, a new issue, related to the current issue, will be created.
- Reopen Issue - Allows the user to reopen a resolved issue
- Move Issue - allows the user to move the issue to another project
- Delete Issue - Allows the user to delete the issue permanently. It is recommended against deleting issues unless the entry is frivolous. Instead issues should be set to resolved and an appropriate resolution category chosen.

A panel is provided to view and update the sponsorship of an issue. Another panel is provided to view, delete and add relationships for an issue. Issues can have a parent/child relationship, where the user is warned about resolving a parent issue before all of the children are resolved. A peer relationship is also possible. Below this, there may be a form for uploading file attachments. The Administrator needs to configure the ticket management to handle file uploads. If uploading to disk is selected, each project needs to set its own upload path. Issue notes are shown at the bottom of the issue report. A panel to add issue notes is also shown.

4.1.7.5 Issue Change Status page

This page is used to change the status of an issue. A user can add an issue note to describe the reason for change. In addition, the following fields may be displayed for update:

- Resolution and Duplicate ID - for issues being resolved or closed
- Issue Handler (Assigned to)
- Any Custom Fields that are to be visible on update or resolution
- Fixed in Version - for issues being resolved
- Close Immediately - to immediately close a resolved issue.

4.1.7.6 Issue Edit page

The layout of this page resembles the Simple Issue View page, but here you can update various issue fields. The Reporter, Category, Severity, and Reproducibility fields are editable. Also modifiable are the Assigned To, Priority, Projection, ETA, Resolution, and Duplicate ID fields.

4.1.7.7 My Account Page

This page changes user alterable parameters for the system. These selections are user specific. My Account page allows the user to change their password, screen name, and email address. It also reports the user's access levels on the current and other projects.

4.1.8 Preferences

This sets the following information:

- Default project
- whether the pages used for reporting, viewing, and updating are the simple or advanced views
- the delay in minutes between refreshes of the view all issues page
- the delay in seconds when redirecting from a confirmation page to the display page
- the time order in which notes will be sorted
- whether to filter email messages based on type of message and severity
- the number of notes to append to notification emails
- the default language for the system. The additional setting of "auto" will use the browser's default language for the system.

4.1.9 System Management Pages

A number of pages exist under the "Manage" link. These will only be visible to those who have an appropriate access level.

4.1.9.1 Manage Users

This page allows an administrator to manage the users in the system. It essentially supplies a list of users defined in the system. The user names are linked to a page where you can change the user's name, access level, and projects to which they are assigned. You can also reset their passwords through this page. At the top, there is also a list of new users (who have created an account in the last week), and accounts where the user has yet to login. New users are created using the "Create User" link above the list of existing users. Note that the username must be unique in the system. Further, note that the user's real name (as displayed on the screen) cannot match another user's user name.

4.1.9.2 Manage Projects Page

This page allows the user to manage the projects listed in the system. Each project is listed along with a link to manage that specific project. The specific project pages allow the user to change:

- the project name
- the project description
- its status

- whether the project is public or private. Private projects are only visible to users who are assigned to it or users who have the access level to automatically have access to private projects (eg: administrators).
- A file directory used to store attachments for issues and documents associated with the project. This folder is located on the web server, it can be absolute path or path relative to the main MantisBT folder. Note that this is only used if the files are stored on disk or via FTP. In case of FTP, the cached version that is saved on the web server, is stored in the specified path.
- common subprojects. These are other projects who can be considered a sub-project of this one. They can be shared amongst multiple projects. For example, a "documentation" project may be shared amongst several development projects.
- project categories. These are used to subdivide the issues stored in the system.
- project versions. These are used to create ChangeLog reports and can be used to filter issues. They are used for both the Found In and Fixed In versions.
- Custom Fields linked to this project
- Users linked to this project. Here is the place where a user's access level may be upgraded or downgraded depending on their particular role in the project.

4.1.10 Monitor Issue

The monitor issues feature allows users to subscribe to certain issues and hence get copied on all notification emails that are sent for these issues. Depending on the configuration, sending a reminder to a user about an issue can add this issue to the user's list of monitored issues. Users who reported the issue or are assigned the issue typically don't need to monitor the issue to get the notifications. This is because by default they get notified on changes related to the issue anyway. However, administrators can change the configuration to disable notifications to reporters or handlers in specific scenarios.

4.1.10.1 Reopen Issue

Re-open issue button is visible in the issue view pages if the user has the appropriate access level and the issue is resolved/closed. Re-opening an issue will allow users to enter issue notes for the reopening reason. The issue will automatically be put into the Feedback status.

4.1.10.2 Delete Issue

The delete issues button appears on the issue view pages for the users who have the appropriate access level. This allows you to delete an existing issue. This should only be used on frivolous or test issues. A confirmation screen will prompt you if you really want to delete the issue. Updater, Developers, Managers, and Administrators can remove issues (you can also configure this).

4.1.10.3 Close Issue

This is a button that appears on the issue view pages for users that are authorized to close issues. Depending on the configuration, users may be able to close issues without having to resolve them first, or may be able to only close resolved issues. After the button is clicked, the user is redirected to a page where an issue note maybe added

4.1.10.4 Assign to me

This button appears in the issue view pages in case of users with access level that is equal to handle_bug_threshold or higher. When this button is clicked the issue is assigned to the user.

4.1.10.5 Resolve Issue

This option on the View Issues page allows you to resolve the issue. It will lead you to a page where you can set the resolution state and a duplicate id (if applicable). After choosing that the user can choose to enter an issue note detailing the reason for the closure. The issue is then set to the Resolved state. The reporter should check off on the issue by using the Close Issue button.

4.2 Live Chat Administration Guide

4.2.1 Introduce

Live Helper Chat is an open source – Live Support chat for the website. It includes: co-browsing, XMPP notifications, GTalk, Jabber, Openfire, Skype, Chrome extension, Node.js support, MySQL, PostgreSQL.

4.2.2 Login Live Chat Admin page

In order to login, point your browser to directory where application is installed (Figure 23). URL address should look like http://<your_domain>/index.php/site_admin/

Example: http://localhost/livehelper/lhc_web/index.php/site_admin/

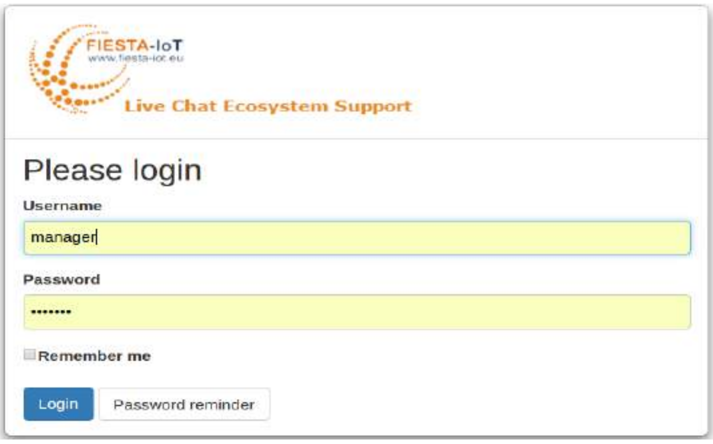

Figure 23 Login Live Chat

4.2.3 Live Chat Main page

The live chat main window consists of three main block: left menu, right menu and middle block (Figure 24) Left menu is used for configuration application. Right menu block displays logged user and resents chats by their status. Middle block browsing in chats messages and information of logged user.

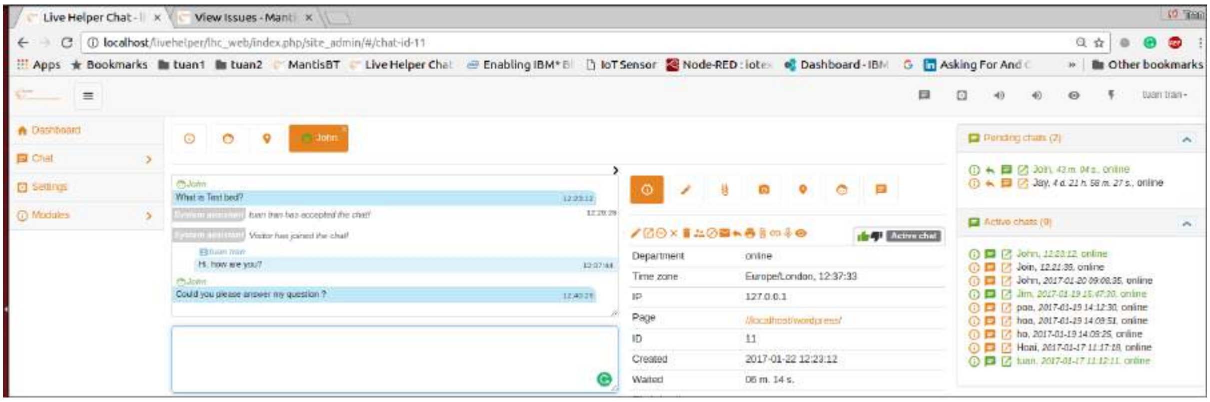

Figure 24 Live Chat Main page

4.2.4 Chat window

In message window you see chat status and user, support messages.

In message writing block you can write messages in multiple lines by pressing Shift+Enter and then you want to send message press enter. Message can be send also by pressing send button (Figure 25).

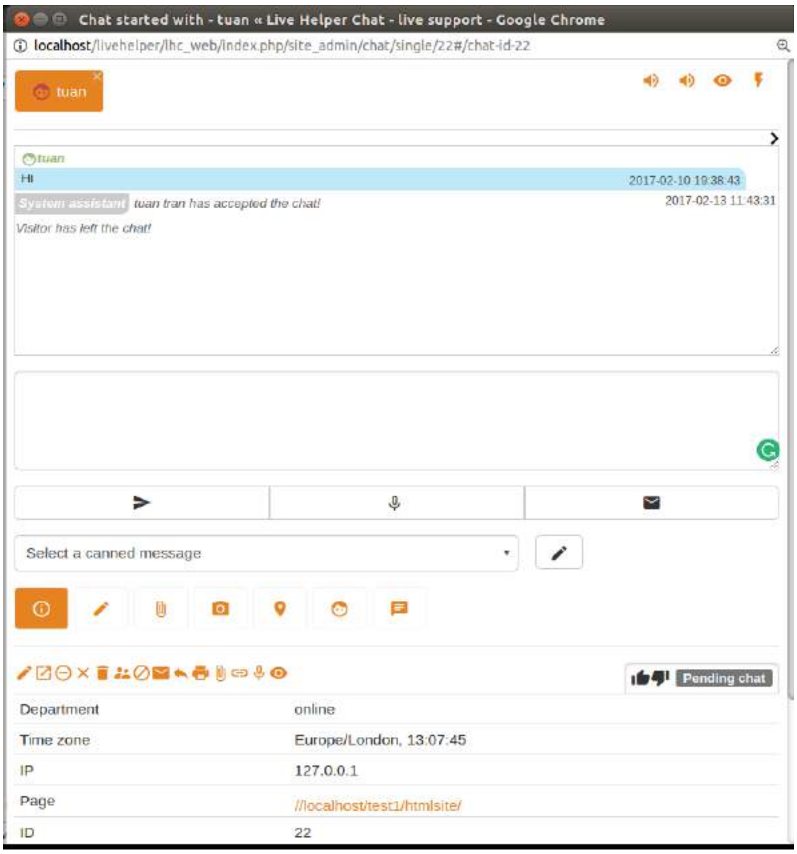

Figure 25 Chat Window

4.2.5 Chat statuses

There are three chat statuses:

- Pending - chat was not opened by operator
- Active - chat is active
- Closed - chat is close

User initiates chat, chat gets Pending status and appears in administration panel as pending one.

Then responsible person start chat it gets active status.

Next responsible person get closes chat it gets closed chat.

4.2.6 Chat listing page

You can access previous chats by clicking left menu “Chat” -> “Chats list” link. You will see in screenshot below (Figure 26).

Chats listing windows in general is the same so only active chats list screenshots provided.

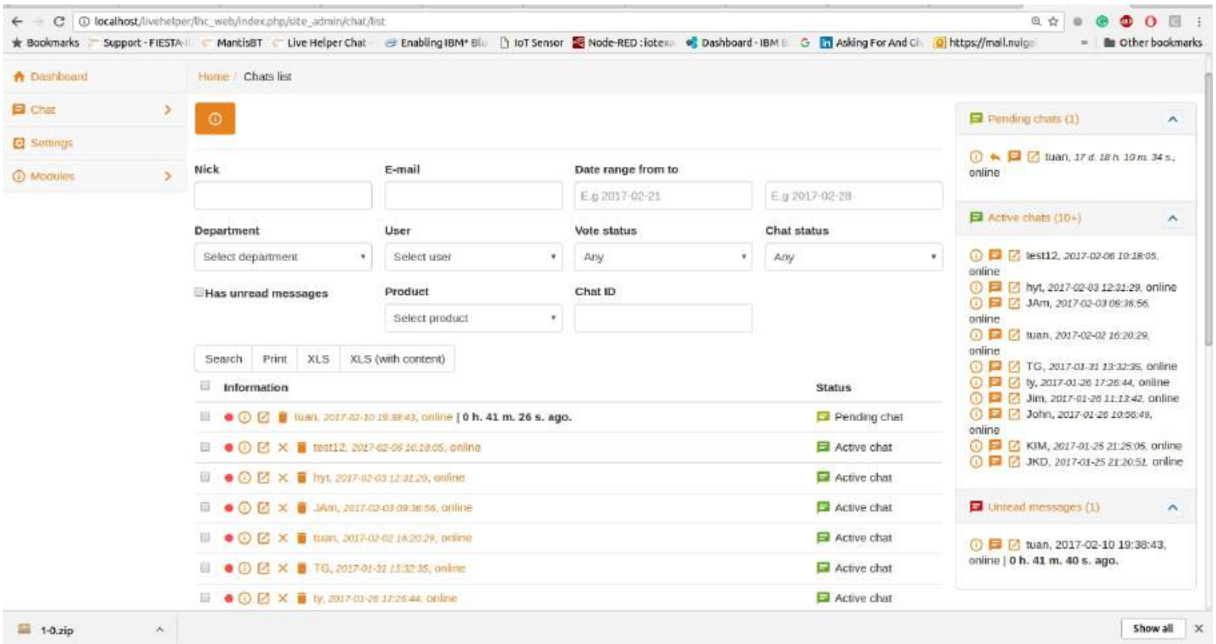

Figure 26 Chat list page

4.2.7 Cleaning cache

Cleaning cache is necessary then user updates roles, users or groups. Expire cache link cleans necessary policy cache and ensures that user rights are updated. It's recommended for non admin users to click "Clean cache" button then administrator updates their permissions. Clean cache link can be seen in system configuration page.

Copyright © 2016 FIESTA-IoT Consortium

4.2.8 Themes

4.2.8.1 Import a theme

Navigation to “System” -> "System configuration" -> "Live help configuration" -> "Import new theme" in this window just upload new theme.

4.2.8.2 Apply imported theme

You can set default theme in Configuration -> Live Chat configuration -> Default theme

Also you can apply themes individually to embed code by selecting combobox in widget generation window. So you can just choose which theme widget should use..

4.2.9 Application default values

Some default values which can be useful to know.

- Chats are synchronized every 3.5 seconds interval.
- Admin interface is synchronized every 10 seconds.
- There is default 10 chats limit in tabs listing. To see all active chats if there is more than ten chats, use listing functionality.

4.3 Create New Ticket Guide

Create New Ticket is a Node.js application that uses the core MantisBT API to create new ticket in Mantis Database (Figure 27).

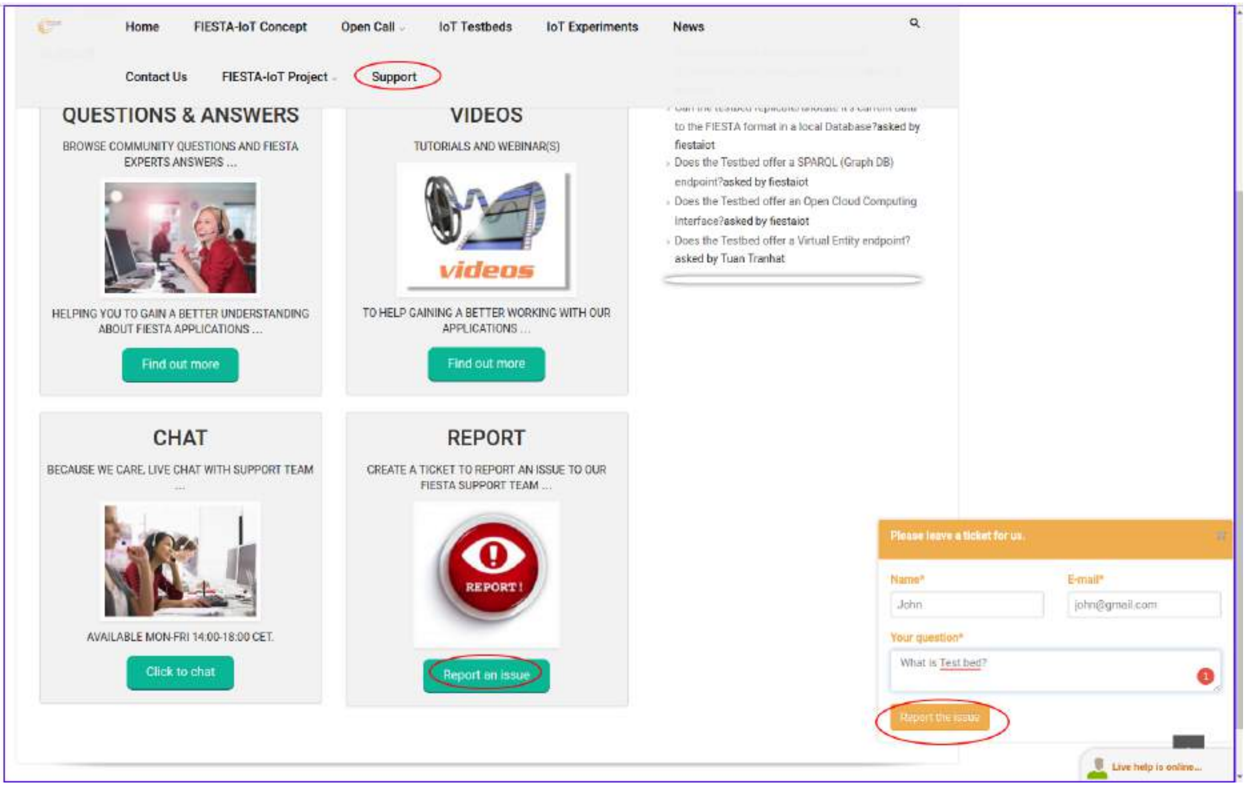

Figure 27 Report an issue widget

The steps to do that are:

- Go to Support page (on FIESTA Portal or FIESTA website)
- Click "Report an issue" button on Support page
- Enter name, email address, your question.
- Click "Report the issue". After that the new ticket will appear in the issue list of MantisBT system.

4.4 Question And Answer Guide

4.4.1 Introduce

DW Question and Answer is a WordPress plugin which builds a complete Question & Answer system for your WordPress site.

The steps to browse questions and fiesta experts answers are:

- Go to Support page (on FIESTA Portal or FIESTA website) (Figure 28)
- Click "Find out more" button in Question & Answer (QA) on Support page (Figure 29)

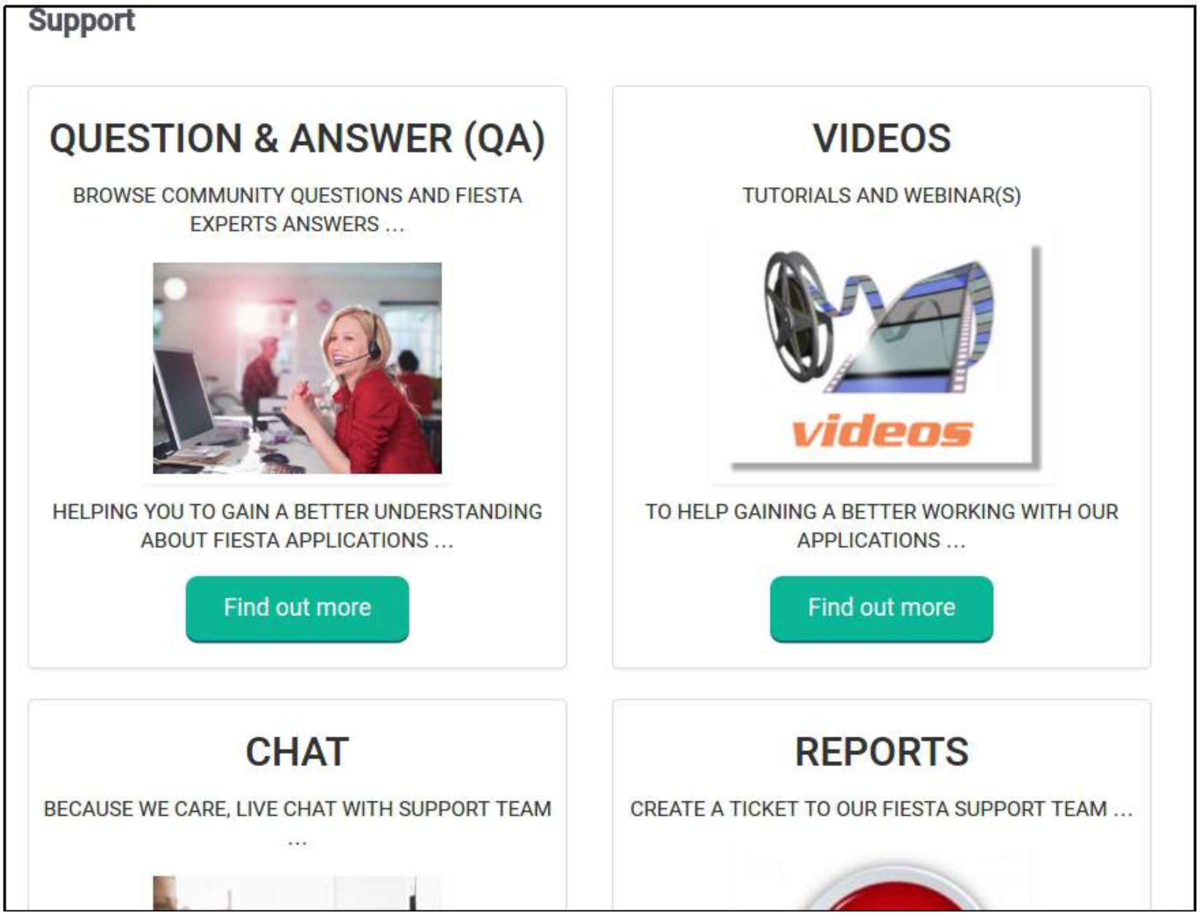

Figure 28 Support page

4.4.2 Front-end appearance

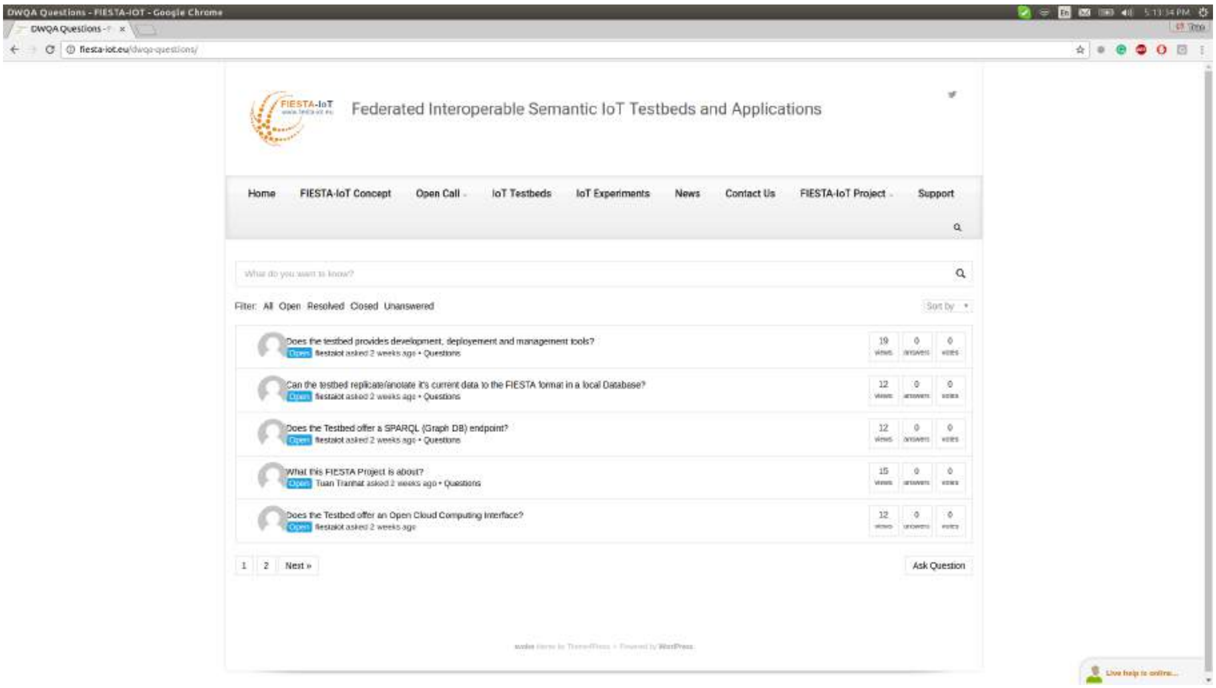

Figure 29 Front-end Questions & Answers

4.4.3 Ask Question page

In Front-end appearance page, click Ask Question button (Figure 30)

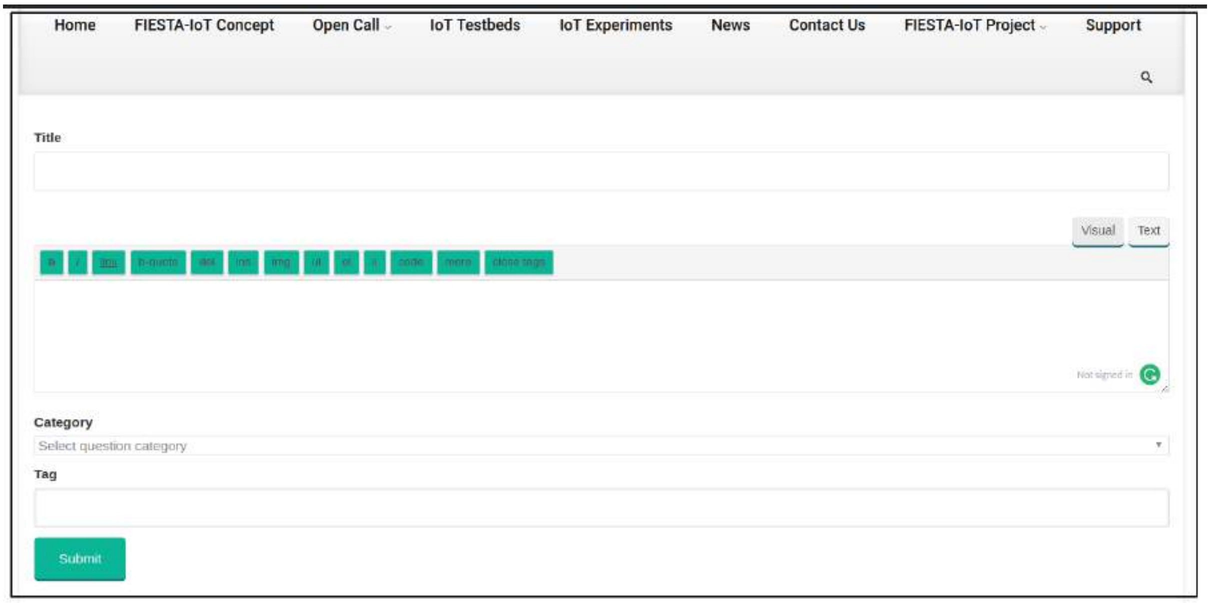

Figure 30 Ask Question page

You can set ask question form on a sidebar or a separate page. With this feature, users can set the title question, provide detailed descriptions with HTML or Markdown, assign it to a category and tag it to help increase site exposure.

Copyright © 2016 FIESTA-IoT Consortium

4.4.4 Search page with Instant search function

Instant search function is Ajax-based solution helps you quickly search information while entering parts of your text. (Figure 31)

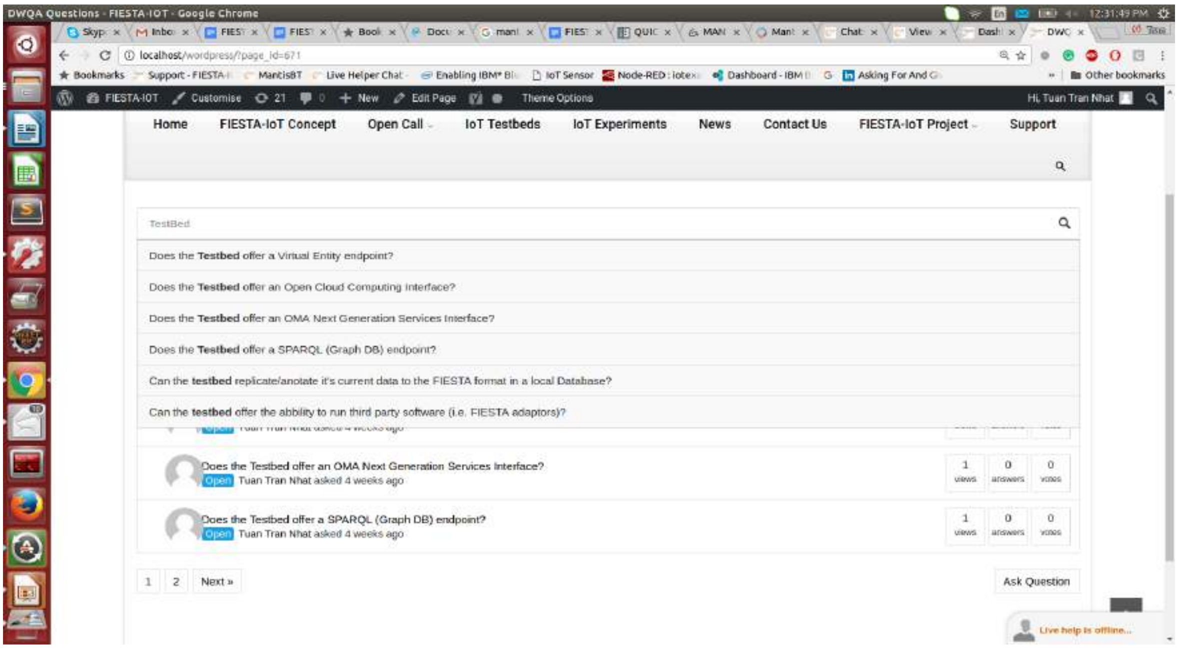

Figure 31 Search page

4.4.5 Back-end settings

This section is in back-end: Dashboard --> Question (Figure 32)

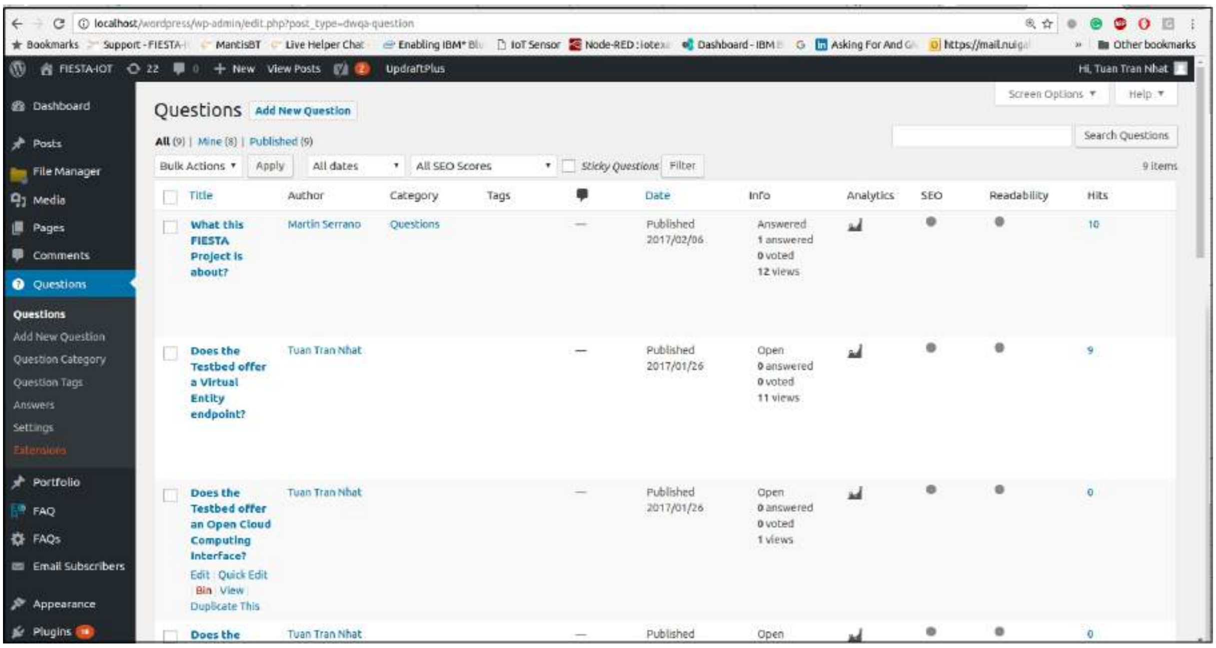

Figure 32 Back-end settings

All features (such as edit, delete, sticky, follow, change question status) also can be handled from back-end. This allowing you to manage the community in an easy and convenient way.

Copyright © 2016 FIESTA-IoT Consortium

4.4.6 Set permissions

Permission Settings section allows you to control who and how they can interact on your Question & Answer page.

This section is in back-end: Dashboard --> Questions --> Settings. There are 3 separate settings for Question, Answer and Comment, applied for default user roles.

By default, one can always edit, delete his own questions, answers and comments. Thus, the Edit and Delete permission here in back-end mean that he can edit and delete ALL other questions, answers and comments. (Figure 33)

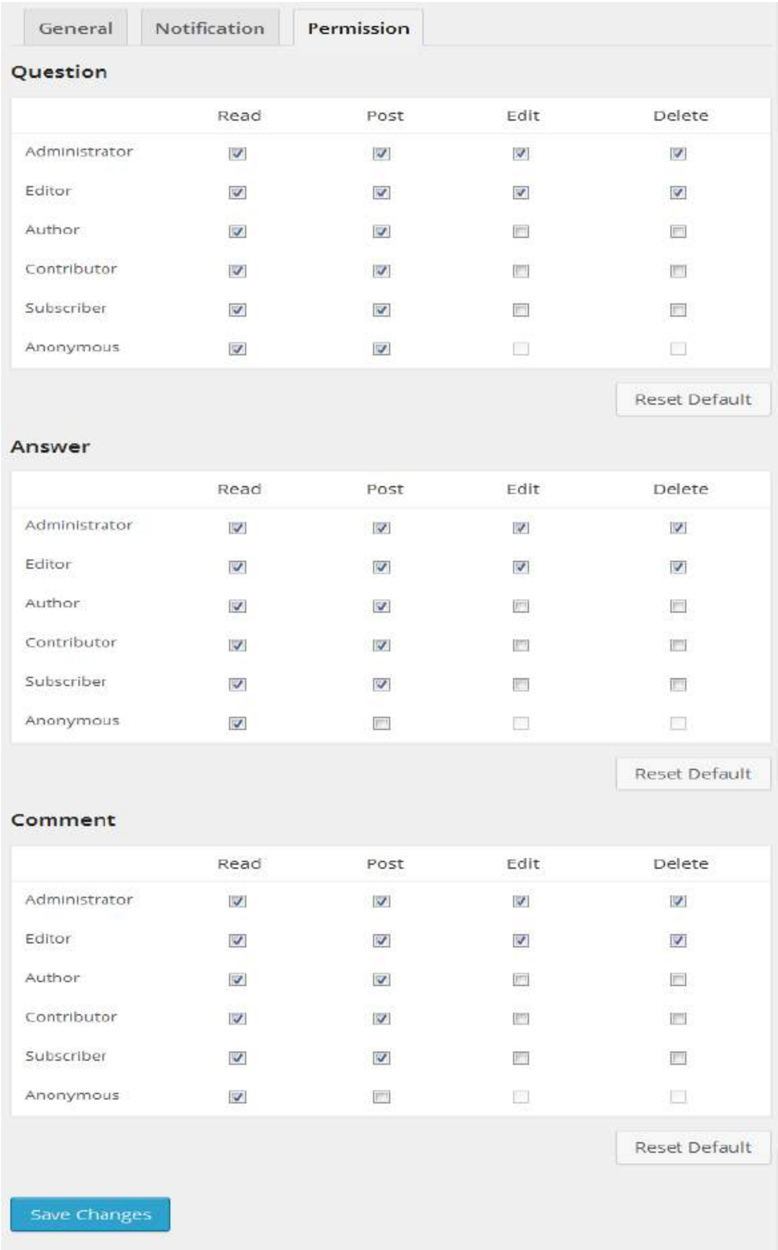

Figure 33 Permission Setting

Copyright © 2016 FIESTA-IoT Consortium

5 CONCLUSIONS

This deliverable reports the work done for creating the supporting system for the ecosystem planning and reporting of Fiesta-IoT. The system build, called, the Ecosystem Support, is the system simplicity connected with powerful features, very good scalability and great tech support. The Ecosystem Support makes online communication easy and efficient for working with FIESTA applications. The system aims to provide speedy feedbacks to the FIESTA-IoT users community (experimenters and testbed providers) and quick solutions from support teams via integrated communication features including live chat, Q&A and Knowledge base.

The whole system is implemented by extend other existing systems such as Mantis, WordPress and Live Chat to build the required features. More importantly, we have integrated them into a single integrated platform that can pipeline data through different modules and coordinate each other. At the current state of implementation, the system have three modules which provide a wide variety of features which are described in the system guidelines which illustrative demonstrations associated with screenshots. The whole system is well packaged and is easy to be installed with step-by-step instructions set that were described in above installation section. Also, the architecture design and other technical implementation details were also described in the first section.

The future functions of the Ecosystem Support will support single sign-on with OpenAM, a report plugin can be used with the Mantis that enables to manually edit the settings or write SQL to fetch reporting information. Moreover, the voice chat function for automation answer is also a desired feature. It is also a plan to integrate the state of the art AI-powered chat bot features to faster the supporting cycle in interacting with Fiesta-IoT platform.

Copyright © 2016 FIESTA-IoT Consortium

6 REFERENCES

[1] FIESTA-IoT, “Deliverable 2.4: FIESTA-IoT Meta Cloud Architecture”, 2015.

[2] FIESTA-IoT, “Deliverable 2.1: Stakeholders Requirements”, 2015.

[3] FIESTA-IoT, “Deliverable 2.2: Analysis of IoT Platforms and Testbeds,” 2015.

[4] FIESTA-IoT, “Deliverable 3.1.1: Semantic models for testbeds, interoperability and mobility support and best practices,” 2016

[5] IoT-A, “Deliverable D1.5: Final Architectural Reference Model for the IoT,” 2013.

[6] IoT-A, “Deliverable D1.2: Initial Architectural Reference Model for IoT,” 2011.

Copyright © 2016 FIESTA-IoT Consortium