DEPARTMENT OF

Chemistry

Repair Ticketing System

Maintenance Guide

Version: 1.0
Date: April 2019

DEPARTMENT OF
Chemistry
Repair Ticketing System

Maintenance Guide

Table of Contents

Introduction

The Chemistry Repair Ticketing System is a web-based application where you can submit repair ticket requests to a Queen's Chemistry Technician. By doing so, you are able to get equipment owned and maintained by the Queen's Chemistry Department fixed, managed all in one place.

Maintenance Suggestions

To debug issues with the website, we have added a small piece of code in config.php. There is a variable called "showErrors" that is normally set to false. If this is set to true, this will display any errors that the website has on the page that is being loaded (in plain text). This should only be enabled when the website is not hosted and under development.

There is also a variable called activateEmail that is true by default. Setting this to false disables email notifications.

If you have everything properly installed and login still isn't working, then there may be a conflict with the information provided. The website assumes that logging in occurs with a username and a password and that the other information available is first name, last name, and email. If any of this information is missing in the active directory, the website may not work. Make sure that the information follows these columns. If you wish to log in with email, you can make the value of the username data the same as the email.

When techs change the status of a ticket and press submit, the page reloads after 3-5 seconds. This delay is due to the emails being sent out. The emails notify users of a change in their ticket status.

The following is a list of file names and their functionality. If a particular feature were to break, this list will serve as a starting point for where to look to begin debugging.

login.php - Handles the login functionality, ensures users credentials are correct and that the login is secure. Blocks unauthorized access.

logout.php - When the user clicks the logout button, their cookie is removed and they are redirected to the login page

index.php - checks if a user is logged in and redirects them to the appropriate page based on if they are a user or an admin

authenticate.php - used to ensure that the user is still authenticated while using the application. This makes sure unauthenticated users cannot perform actions that would affect the system

config.php - configuration file containing server and database connectivity info, active directory configuration, JWT configuration, and email information. When doing the initial setup, this is the only file you should have to change.

connect.php - used to connect to the database whenever a query is needed

createTicketPage.php - backend functionality and frontend style for the Create Ticket page

createTicket.php - Posts a successfully created ticket to the database

showTicketsPage.php - backend functionality and frontend style for the Tickets page

showTickets.php - handles all database access for the Tickets page. For example when using the Search functionality.

showTickets.js - connects showTickets.php, showTicketsPage.php, and showTickets.html

howTo.php - frontend style for the How To page shown to users

invoicePage.php - functionality and style for the invoice page that is generated for a technician

invoiceFunctionality.php - inserts the parts inserted by the Technician on an invoice into the database

invoice.js - handles some of the front end functionality of the invoice page and connects invoicePage.php to invoiceFunctionality.php. Also calls generatePDF to make the pdf

messagePage.php - backend and frontend for the messages page for a given ticket.

messageFunc.php - database functionality for the messages page to retrieve and add messages for a ticket.

messagesPage.js - javascript functionality for the messages page, connects messagePage.php and messageFunc.php

statsPage.php - frontend and backend for the Statistics page

getStats.php - database functionality to retrieve the statistics from the DB

stats.js - handles calculations and creation of graphs on the Statistics page
generatePDF.php - Used to generate the invoice PDF
saveTicketInfo.php - used in showTickets.js to save any changes that are made to a ticket in the Tickets page
sendEmail.php - configuration for the Email sending functionality

Known Limitations

1. If some pages begin to slow considerably, it may be due to the database tables becoming much too large over time. This is unlikely to happen as we have added limits to a few pages, but could possibly occur.
2. Tickets and messages may start displaying the wrong time for closed time and create time, as well as the time a message was sent. This occurs when the system time is not configured properly.

Workarounds

1. A solution for this could be to delete some of the oldest entries from a table. Since most tables and table entries have a date associated with them, so it should be possible to query the database with an SQL DELETE query, deleting based off of old dates. Another solution would be to alter the query to limit the number of entries that get displayed on the webpage.
2. A workaround for incorrect time would be to make sure the system time is corrected as that is where the information gets pulled from.

Future Potential Improvements

Internal System Email Exchange

Currently, email notifications are only enabled to notify the user of a successfully created ticket. If it is desired to include in other places, for example, notify the user when their ticket is marked as “Closed”, please follow the implementation seen in the file “main.js”, which sends an email upon ticket submission. The API used for email sending is PHPMailer.

Tracking Inventory System

Due to there not being a complete list of equipment currently on inventory, an inventory tracking system was not implementable. This could either be accomplished by creating a list of all equipment in inventory, making a table in the database, and filling it, OR by modifying the create ticket functionality to automatically add equipment to a database table, whenever a ticket is made.

Integrate Past Statistics

Currently, the statistics page displays graphs based on information gathered from repair tickets submitted through the system. However, a future potential improvement that can be made to the statistics page is to input data from past repairs that were completed prior to the implementation of the system. As a result, the data collected and displayed would be most accurate to the current state of the Department of Chemistry's machine and equipment repairs.

This would be accomplished by manually inputting the past data into the system's SQL database by either using the SQL INSERT INTO statement in the source code or through the database's GUI interface such as phpMyAdmin.

Closed Ticket Message

Currently, when a ticket is closed, the technician does not see that recorded anywhere. It would be useful if when a ticket is closed, that information is displayed on the message page so that the technician and the user can see it. This would be useful in situations where a ticket is closed and reopened multiple times.

Adding Tax to Invoice

Currently, when an invoice is generated, no tax is taken into account. The API used, Invoice generator (link available on installation guide has information on applying tax), has the ability to apply a tax to the items. The invoice.js file can be altered to include a tax when generating an invoice.

Notes to Techs about a Ticket

A possible feature that would be useful is to add notes about a ticket that only other techs can read. This would help other techs if they need to work on that ticket in the future.