**C868 – Software Capstone Project Summary**

**Task 2 – Section C**

|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | http://www.idevnews.com/views/images/uploads/general/wgu_logo.png  Project Themis |
| **Student Name:** | Michael Tillotson |

**Table of Contents**

Contents

[**Table of Contents** 2](#_Toc61852675)

[Application Design and Testing 4](#_Toc61852676)

[Design Documentation 4](#_Toc61852677)

[Class Design 4](#_Toc61852678)

[UI Design 5](#_Toc61852679)

[Unit Test Plan 11](#_Toc61852680)

[Introduction 11](#_Toc61852681)

[Purpose 11](#_Toc61852682)

[Overview 17](#_Toc61852683)

[Test Plan 17](#_Toc61852684)

[Items 17](#_Toc61852685)

[Features/Functions 17](#_Toc61852686)

[Deliverables 18](#_Toc61852687)

[Tasks 18](#_Toc61852688)

[Needs 18](#_Toc61852689)

[Pass/Fail Criteria 18](#_Toc61852690)

[Source Code 18](#_Toc61852691)

[User Guide 19](#_Toc61852692)

[*Introduction* 19](#_Toc61852693)

[*Installation and Using the Application* 19](#_Toc61852694)

[*Logging In* 19](#_Toc61852695)

[*Navigation-Main Menu* 20](#_Toc61852696)

[*Navigation-Reports Menu* 21](#_Toc61852697)

[*Generating a Report – Reports Menu* 22](#_Toc61852698)

[*Navigation-Main Tickets Menu* 23](#_Toc61852700)

[*Searching for a Ticket -Main Tickets Menu* 24](#_Toc61852701)

[*Navigation-Add Tickets Menu* 25](#_Toc61852702)

[*Creating a Ticket -Add Ticket Menu* 26](#_Toc61852703)

[*Navigation-Edit Tickets Menu* 27](#_Toc61852704)

[*Editing a Ticket – Edit Ticket Menu* 27](#_Toc61852705)

[*Deleting a Ticket – Edit Ticket Menu* 28](#_Toc61852706)

[*Searching for a Ticket – Edit Ticket Menu* 29](#_Toc61852707)

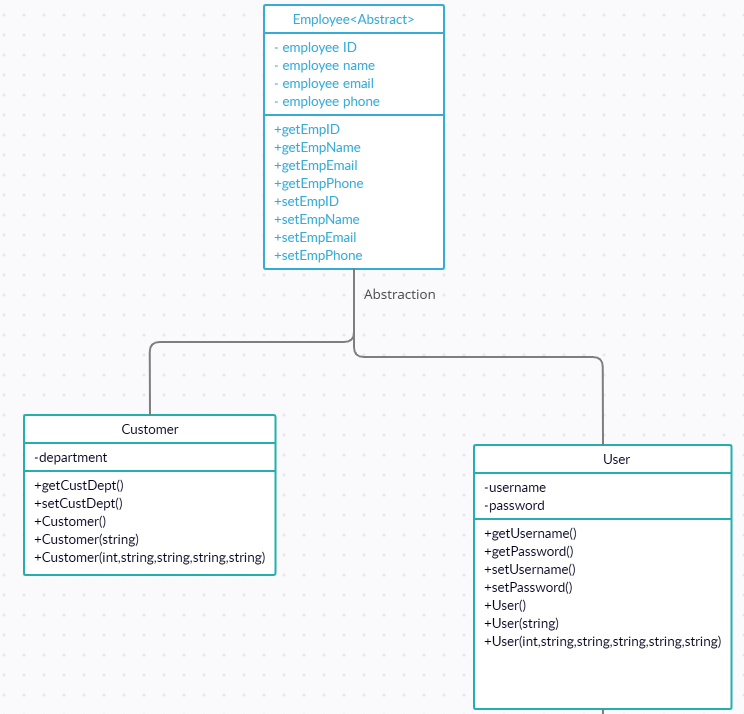
# Application Design and Testing

# Design Documentation

## Class Design

Themis, the ticket management system, will contain six classes, including one abstract. The six classes encompass the main pieces of data being gathered, manipulated, and stored within Themis. See the referenced diagram for each of the classes, and their associated data points. The six classes are, Tickets, Employee (the abstract), Customer, User, database connection (DBConnection), and Query.

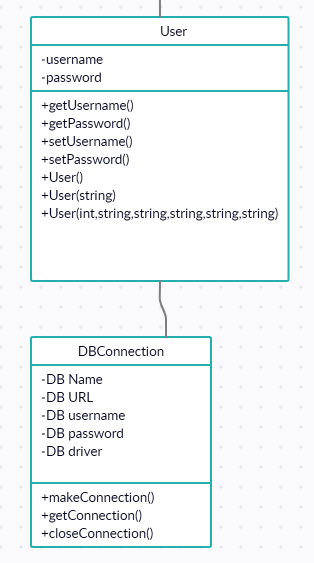
Employee is used as a parent class of Customer and User, as both reference Temple Forestry employees, with the primary difference being User has additional variables for login information, while Customer has additional variables for contact information. The functionality to add or edit Employees is not necessary for the Themis application at this time and will be handled by the database Administrator.



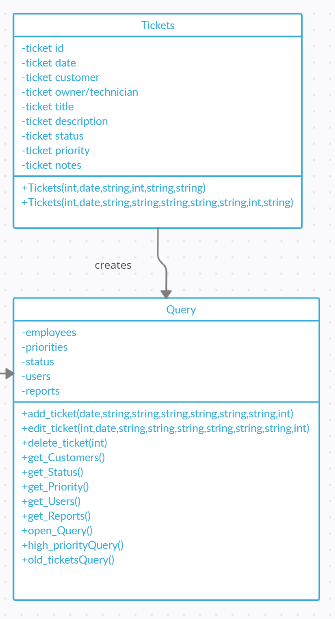
The Customer child class has only one use within Themis, the information fills into a dropdown menu, which is used when a ticket is being added or edited to select the Customer who is having technical issues.

The User child class is used in two ways. First it serves a similar role as the Customer class. A technician can be chosen from the drop-down menu when adding or editing a ticket. Any technician in the User list will also require access to Themis to create, update, and close their tickets. The second purpose of the user class will be covered.

On startup, the application will present with a login screen. This is the second function of the User class, where a technician can log in with their username and password. This also incorporates the database connection class, which has two methods, one to open a connection on successful login, and another to close the connection.



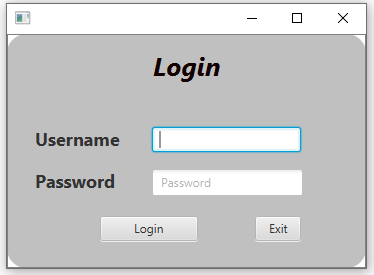
The query class contains all major repeatable queries used within the application, to maintain application simplicity and reuse code efficiently. It is called in multiple places throughout the application, from generating reports to filling in drop-down menus within the ticket management screens. It is also the primary method of adding, updating, and deleting tickets.



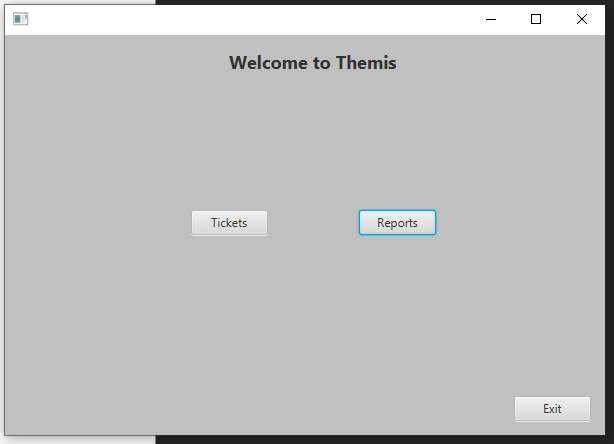
The last class utilized by Themis is the Tickets class. This class pulls relevant ticket data from the database using the Query class and instantiates it as an object which can be edited or deleted within the application. The application also contains methods for generating new ticket objects, which are stored to the database, also using the Query class. It also pulls pieces from the User and Customer classes to fill its menus with the appropriate data for user selection.

## UI Design

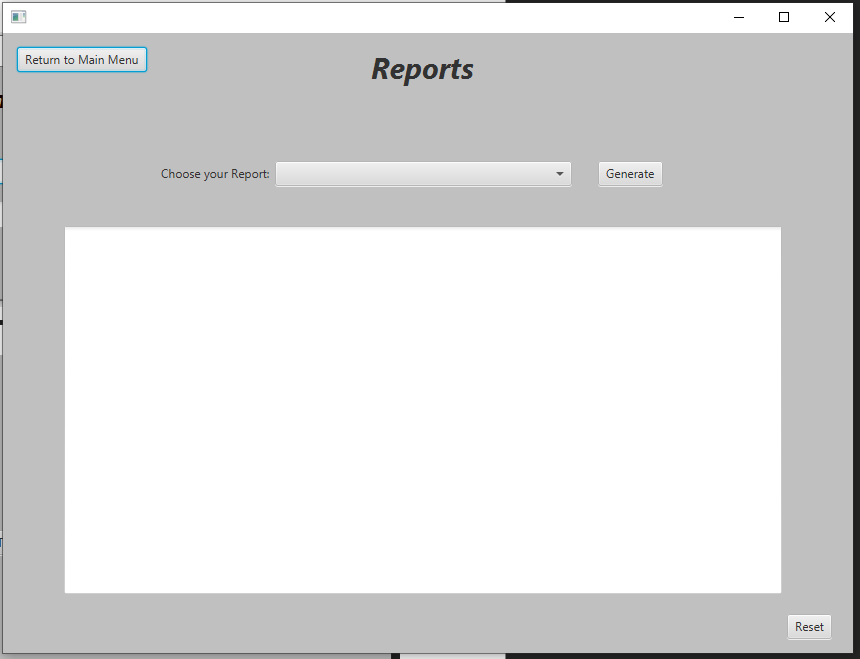
The UI of the application contains six screens. As design simplicity and ease of use are of paramount importance in UI development all screens are designed in a similar uniform fashion, and operate with simple easy to intuit controls, as can be seen in the high-fidelity examples.



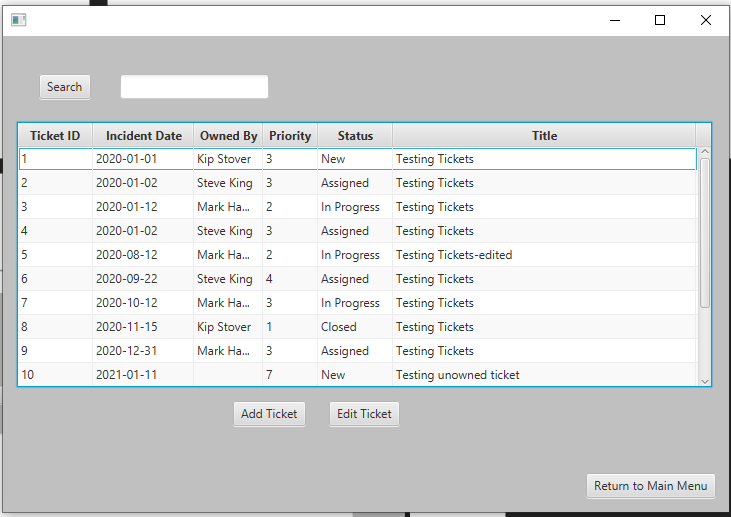
The Login Screen is the first thing that will come up on application startup. Its design function should be self-apparent to users, with labels and suggestion text within the boxes. Feedback will be given if username or password fail to matchup following standard UI guidelines.



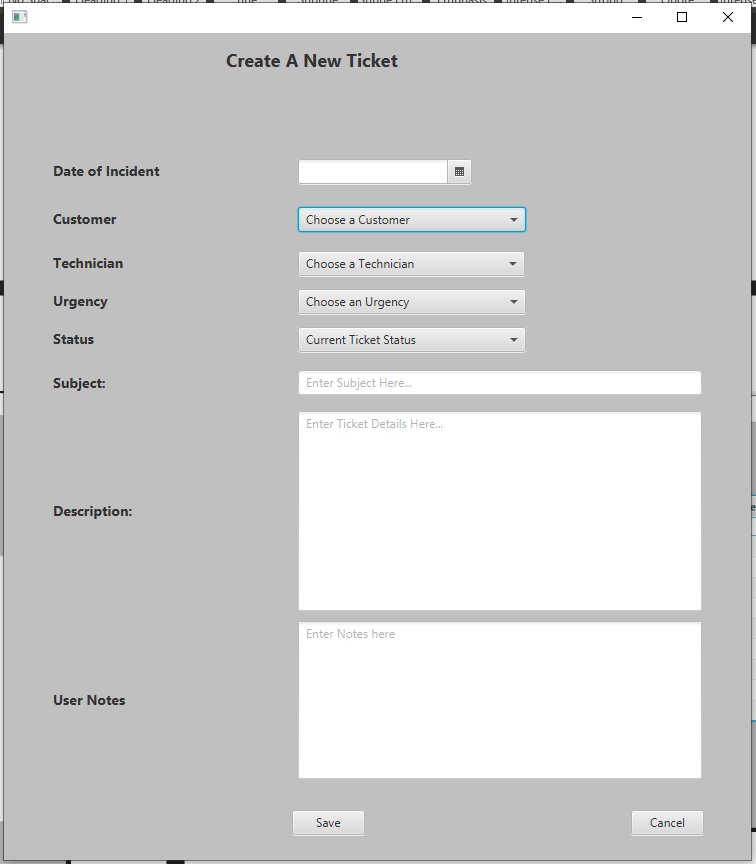
Upon successful login the user will be taken to the Main Screen which has two options available, as well as an option to Exit out of the application. One button will take the users to the Ticket screen, while the other goes to the Report screen for report generation.



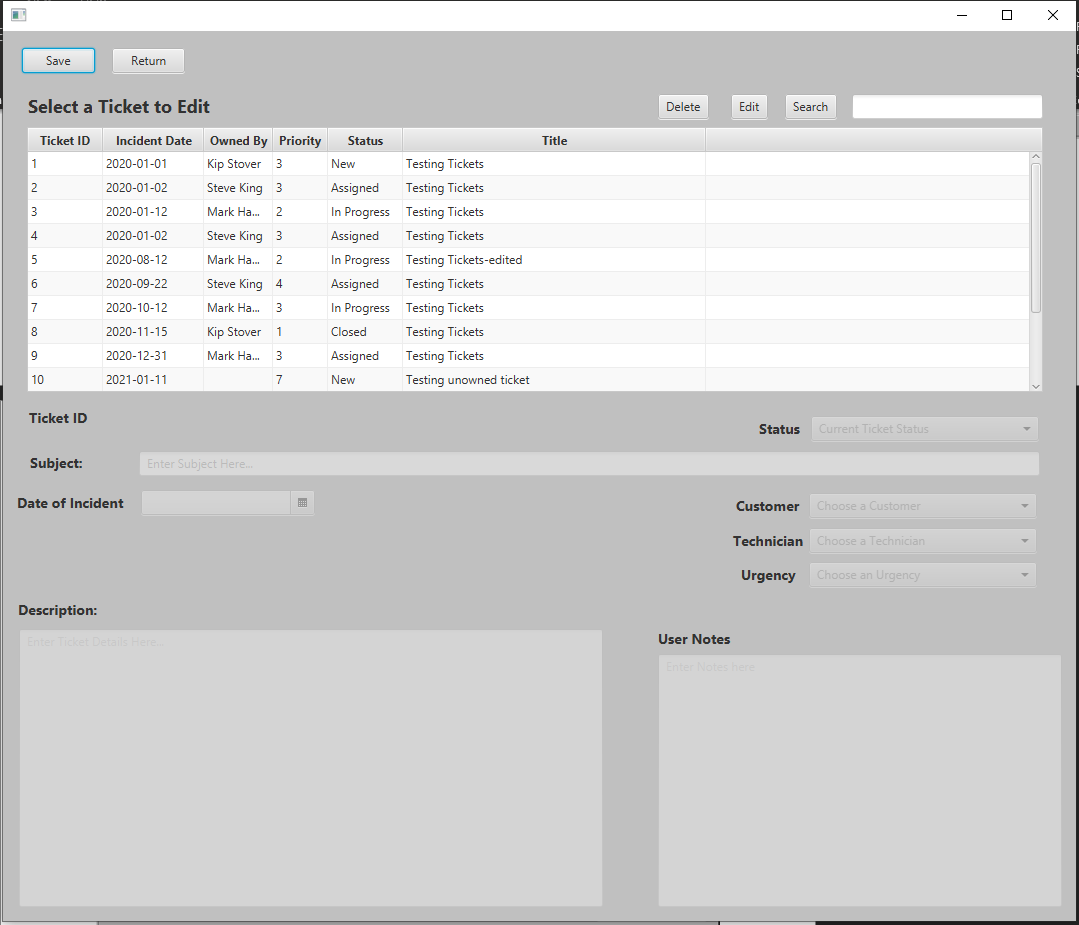
If “Reports” is selected, then users are taken to the Report Screen. It contains a large unalterable text box, a dropdown menu of reports, and an option to generate and clear reports from the textbox. Instructions are labelled on the screen telling the user how to generate a report. There is also an option to return to the Main Screen.



If the “Tickets” option is chosen from the Main Screen, then the users are taken to the main Ticket Screen. This contains a table view, a search bar, and the option to Navigate to the Add Ticket or Edit Ticket screens down at the bottom. Return to Main Menu will send the user back to the Main screen, while Add and Edit Ticket will open the associated ticket screens. Search will run a search of data entered in the text field next to it.



Should Add Ticket be selected the Add Ticket screen will open, leading to a fillable form where the necessary ticket information is gathered and when save is selected the information is stored, provided no errors are found within the given fields. Assuming no errors are found, the Save button will then return the user to the Main Ticket screen. Cancel will also return the user to the Main Ticket screen.



If on the Main Ticket screen, Edit Ticket is selected it will open the Edit Ticket screen. This contains all the fields of data generated for tickets, however on screen loading all fields will be empty and disabled from editing. It is only by selecting a ticket and clicking edit that the data will be loaded into the fields and become mutable. The Search button operates in identical fashion to the one on the Main Ticket Screen. Delete will Delete the selected ticket after the user confirms in a popup alert that is generated on button click. Save will update the edited ticket provided no errors are found in the given fields.

# Unit Test Plan

## Introduction

### Purpose

Manual testing of the application through the user interface was the primary method used to test the program. It was tested for base functionality, data validation, and error handling. Each mutable screen view was tested as an individual unit and added into the application piecemeal. Once all screen view units were tested and incorporated, a final encompassing test was performed to ensure all units operated in sync with each other. The plan below details each test done on a unit. All errors discovered were noted and returned to the development team for correction. Once the updates were made, an entirely new Unit test was performed to ensure functionality was correct and that no new errors were discovered. Zero failures discovered during the full test run.

**Login Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Test #** | **Test Desc** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | Login | User: test  Pw: test | Login Success | Login Success | Pass |
| 2 | Login w/ null | User: null  Pw: null | Login Failure  Error msg, | Login Failure  Error msg, | Pass |
| 3 | Login w/typo | User: test pw: test1 | Login Failure  Error msg, | Login Failure  Error msg, | Pass |

**Main Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Test #** | **Test Desc** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | Exit Button | N/A | Alert to Exit the program | Alert to Exit the program | Pass |
| 2 | Tickets Button | N/A | Main Tickets Screen will open | Main Tickets Screen will open | Pass |
| 3 | Reports Button | N/A | Reports Screen will open | Reports Screen will open | Pass |

**Report Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Test #** | **Test Desc** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | Return Button | N/A | Return to Main Screen | Return to Main Screen | Pass |
| 2 | 30 Days Report | N/A | Generate report with datetime stamp, lock report button until clear button used | Generate report with datetime stamp, lock report button until clear button used | Pass |
| 3 | High Priority Report | N/A | Generate report with datetime stamp, lock report button until clear button used | Generate report with datetime stamp, lock report button until clear button used | Pass |
| 4 | Open Tickets Report | N/A | Generate report with datetime stamp, lock report button until clear button used | Generate report with datetime stamp, lock report button until clear button used | Pass |
| 5 | Clear Button | N/A | Clear report field, unlock Generate Report button | Clear report field, unlock Generate Report button | Pass |

**Main Ticket Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Test #** | **Test Desc** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | Return Button | N/A | Return to Main Screen | Return to Main Screen | Pass |
| 2 | Search Button | Text field: 1 | Table view display ticket with ID 1 | Table view display ticket with ID 1 | Pass |
| 3 | Search Button | Text field: Kip Stover | Table view display all tickets owned by Kip Stover | Table view display all tickets owned by Kip Stover | Pass |
| 4 | Search Button/nothing found | Text field: blahblah | Table view empty, Alert “No results found” | Table view empty, Alert “No results found” | Pass |
| 5 | Add Ticket button | N/A | Add Tickets screen will open | Add Tickets screen will open | Pass |
| 6 | Edit Ticket button | N/A | Edit Tickets screen will open | Edit Tickets screen will open | Pass |
| 7 | Return to Main Menu button | N/A | Main Screen will open | Main Screen will open | Pass |

**Add Ticket Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Test #** | **Test Desc** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | Cancel button | N/A | Return to the Main Ticket Screen | Return to the Main Ticket Screen | Pass |
| 2 | Save button | Date: 1-15-21  Customer: John Doe  Technician: Kip Stover  Urgency: 3  Status: Assigned  Subject: Test Ticket  Description: Test ticket  Notes: note | System print confirmation of ticket creation,  Return to main menu | System print confirmation of ticket creation,  Return to main menu | Pass |
| 3 | Save button/missing date | Date: 1-15-21  Customer: John Doe  Technician: Kip Stover  Urgency: 3  Status: Assigned  Subject: Test Ticket  Description: Test ticket  Notes: note | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 4 | Save button/missing customer | Date: 1-15-21  Customer: John Doe  Technician: Kip Stover  Urgency: 3  Status: Assigned  Subject: Test Ticket  Description: Test ticket  Notes: note | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 5 | Save button/missing technician | Date: 1-15-21  Customer: John Doe  Technician: Kip Stover  Urgency: 3  Status: Assigned  Subject: Test Ticket  Description: Test ticket  Notes: note | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 6 | Save button/missing Urgency | Date: 1-15-21  Customer: John Doe  Technician: Kip Stover  Urgency: 3  Status: Assigned  Subject: Test Ticket  Description: Test ticket  Notes: note | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 7 | Save button/missing Status | Date: 1-15-21  Customer: John Doe  Technician: Kip Stover  Urgency: 3  Status: Assigned  Subject: Test Ticket  Description: Test ticket  Notes: note | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 8 | Save button/missing Subject | Date: 1-15-21  Customer: John Doe  Technician: Kip Stover  Urgency: 3  Status: Assigned  Subject: Test Ticket  Description: Test ticket  Notes: note | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 9 | Save button/missing description | Date: 1-15-21  Customer: John Doe  Technician: Kip Stover  Urgency: 3  Status: Assigned  Subject: Test Ticket  Description: Test ticket  Notes: note | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 10 | Save button/missing notes | Date: 1-15-21  Customer: John Doe  Technician: Kip Stover  Urgency: 3  Status: Assigned  Subject: Test Ticket  Description: Test ticket  Notes: note | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 11 | Enter date past Today | Date: any date after current date | Unable to select | Unable to select | Pass |
| 12 | Confirm Technician ComboBox filled | N/A | Dropdown menu populated with names | Dropdown menu populated with names | Pass |
| 13 | Confirm Customer ComboBox filled | N/A | Dropdown menu populated with names | Dropdown menu populated with names | Pass |
| 14 | Confirm Status ComboBox filled | N/A | Dropdown menu populated with Status’s | Dropdown menu populated with Status’s | Pass |
| 15 | Confirm Priority ComboBox filled | N/A | Dropdown menu populated with priority list | Dropdown menu populated with priority list | Pass |

**Edit Ticket Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test #** | **Test Desc** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | Return button | N/A | Return to the Main Ticket Screen | Return to the Main Ticket Screen | Pass |
| 2 | Search Button | Text field: 1 | Table view display ticket with ID 1 | Table view display ticket with ID 1 | Pass |
| 3 | Search Button | Text field: Kip Stover | Table view display all tickets owned by Kip Stover | Table view display all tickets owned by Kip Stover | Pass |
| 4 | Search Button/nothing found | Text field: blahblah | Table view empty, Alert “No results found” | Table view empty, Alert “No results found” | Pass |
| 5 | Delete Button | Ticket Selected from Table | Alert, Confirm Deletion | Alert, Confirm Deletion | Pass |
| 6 | Delete Button/No Selection | N/A | Alert, No Ticket Selected | Alert, No Ticket Selected | Pass |
| 7 | Edit Button | Ticket Selected from Table | Fields fill with selected ticket data | Fields fill with selected ticket data | Pass |
| 8 | Edit Button/No Selection | N/A | Alert, No Ticket Selected | Alert, No Ticket Selected | Pass |
| 9 | Save button | N/A | Alert, Confirm Ticket “ID” updated | Alert, Confirm Ticket “ID” updated | Pass |
| 10 | Save button/missing date | N/A | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 11 | Save button/missing customer | N/A | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 12 | Save button/missing technician | N/A | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 13 | Save button/missing Urgency | N/A | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 14 | Save button/missing Status | N/A | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 15 | Save button/missing Subject | N/A | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 16 | Save button/missing description | N/A | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 17 | Save button/missing notes | N/A | Error message, “One or more fields is missing information” | Error message, “One or more fields is missing information” | Pass |
| 18 | Enter date past Today | Date: any date after current date | Unable to select | Unable to select | Pass |
| 19 | Confirm Technician ComboBox filled | N/A | Dropdown menu populated with names | Dropdown menu populated with names | Pass |
| 20 | Confirm Customer ComboBox filled | N/A | Dropdown menu populated with names | Dropdown menu populated with names | Pass |
| 21 | Confirm Status ComboBox filled | N/A | Dropdown menu populated with Status’s | Dropdown menu populated with Status’s | Pass |
| 22 | Confirm Priority ComboBox filled | N/A | Dropdown menu populated with priority list | Dropdown menu populated with priority list | Pass |

### Overview

Manual testing was used by the development team for each of the units, with each set of tests being performed as the unit’s addition to the core application was completed. The units were separated by view, as each view has an associated controller class following the View-Model-Control model of development. The individual View/Controller pairings are each considered a single unit. Any errors found were documented in the bug report and assigned to the development team for resolution. All tests were performed within the user interface.

## Test Plan

### Items

* Netbeans 8.2
* Java version 8
* Login access to the Application

### Features/Functions

The functions that are used in each test are all methods from all controllers. The testing itself is completed by using the application UI. All methods are tested through comprehensive use of the interface. Data integrity is also examined during testing, as data validation occurs within the controllers. Error handling and messages are also generated within the application interface as part of the test plan.

### Deliverables

The test tables will show a pass/fail mark for each test performed, as well as the nature of the test, the data used, the expected result, and the actual result of the test. Data has been recorded within this document for reference purposes.

### Tasks

1. Follow the test cases.
2. Log test failures and assign to the development team for remediation.
3. Complete test cases.
4. Repeat steps 1-3 until all test cases are passed.

### Needs

There will need to be a Windows 10 work environment available, of any version. The test machine will require Java 8 and access to the Netbeans IDE, version 8.2. The machine will also require internet access to make a connection to the MySQL database.

### Pass/Fail Criteria

The test plan accounts for Pass/Fail criteria within the Expected and Actual Results column. A test case will Pass if the expected results match the actual. It will fail if the results differ. Discrepancies are noted in the bug report log and assigned to the development team for remediation.

# Source Code

Source code can be found in the uploaded zip file titled C868. It is also available at

<https://github.com/WalkinDude19/Themis>

# User Guide

## *Introduction*

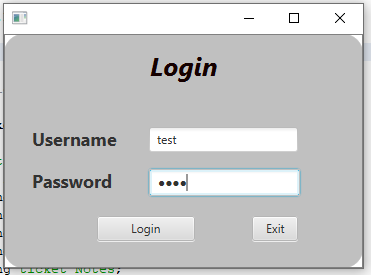
This guide is for users of the Themis Ticket Management Application. Themis is a lightweight Java application designed for use with minimal installation or user or backend setup to get running. Please follow the instructions below for a tutorial on how to setup and use Themis.

## *Installation and Using the Application*

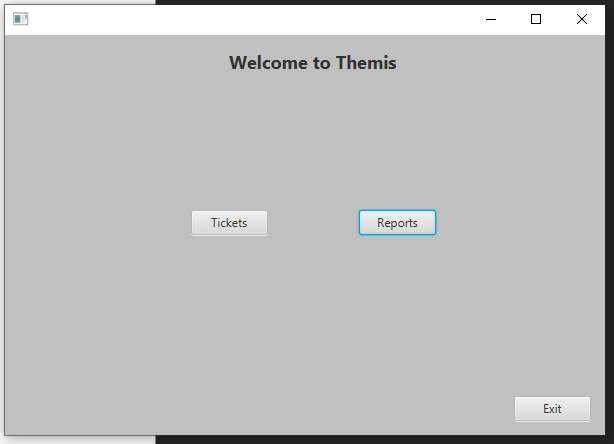
1. Ensure the client machine meets specification. Machine running Windows 10 OS, any version. Java 8SE is required. If user will be maintaining customer/user data they will also need MySQL Workbench, but it is not necessary to run the application.
2. The application will be delivered in .zip format. Your IT professional will extract the file to a launch folder and use the .jar file for launching the application.
3. After the user downloads the .zip file extract to the desired launch folder.
4. Double click the highlighted folder to open its contents.
5. Open the ‘dist’ folder.
6. Double click the .jar file to open the application.
7. The login screen should appear as the application has started.

## *Logging In*

1. Fill out the text fields with username and password.
   1. **For testing purposes, use ‘test’ for username and ‘test’ for password, without the quotations.**

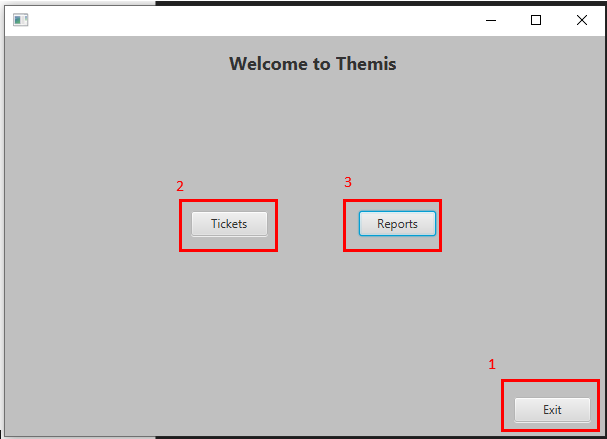


1. Click login, the Main screen will display.



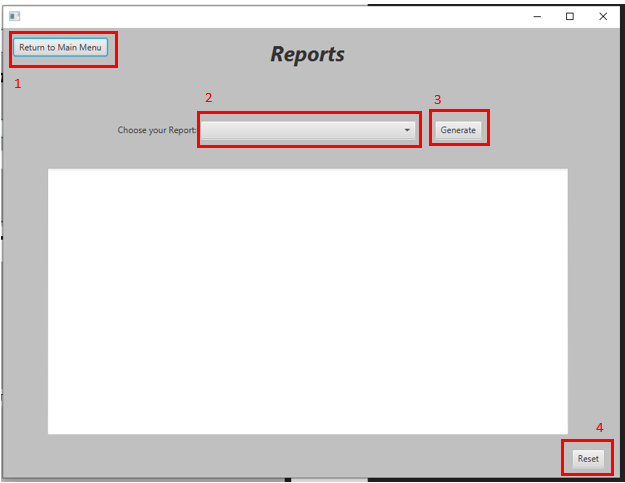
## *Navigation-Main Menu*

1. Select Exit, from the bottom right of the Main screen to exit the application.
2. Select Tickets, to navigate to the Main Tickets Menu
3. Select Reports to navigate to the Main Reports Menu



## *Navigation-Reports Menu*

1. Click “Return to Main Menu” to return to the Main Ticket Screen
2. Select a Report Query
3. Click Generate to generate a Report.
4. Clears the Report text Field.



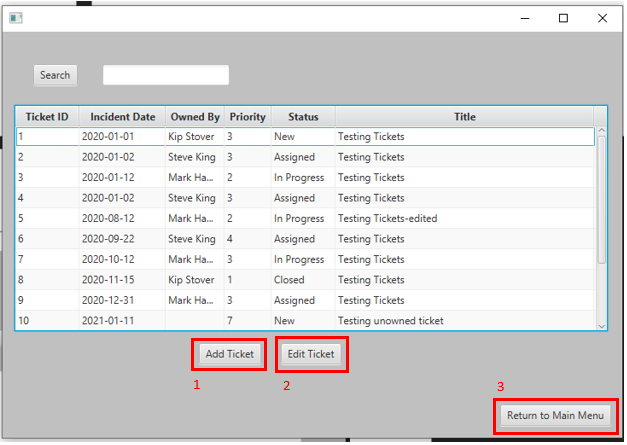
## *Generating a Report – Reports Menu*

1. Select a report from the Dropdown menu.
2. Once the report is chosen, click Generate
3. The output will load into the text field.
4. Select clear to clear the report before any further reports can be generated.

## 

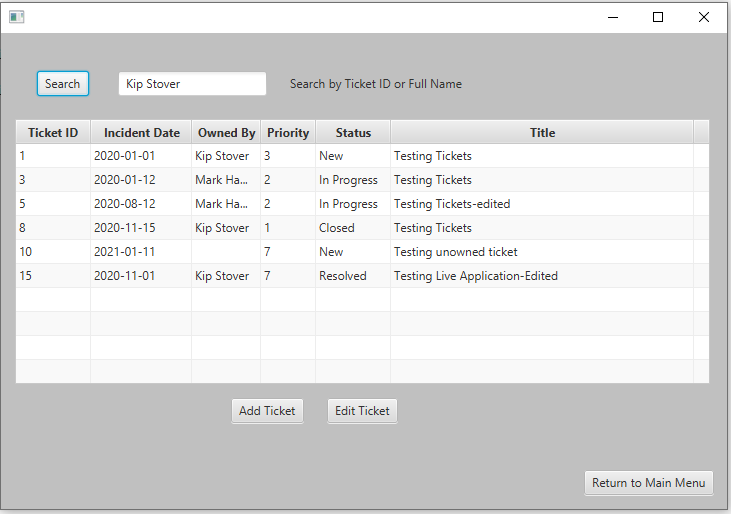
## *Navigation-Main Tickets Menu*

1. The Add Ticket button opens the Add Ticket Screen
2. The Edit Ticket button opens the Edit Ticket Screen
3. The “Return to Main Menu” button will take the user back to the Main Screen.

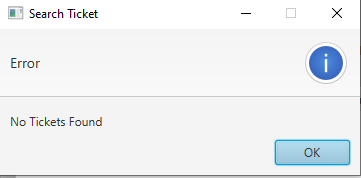


## *Searching for a Ticket -Main Tickets Menu*

1. Enter either the Ticket ID or full Technician Name into the search bar.



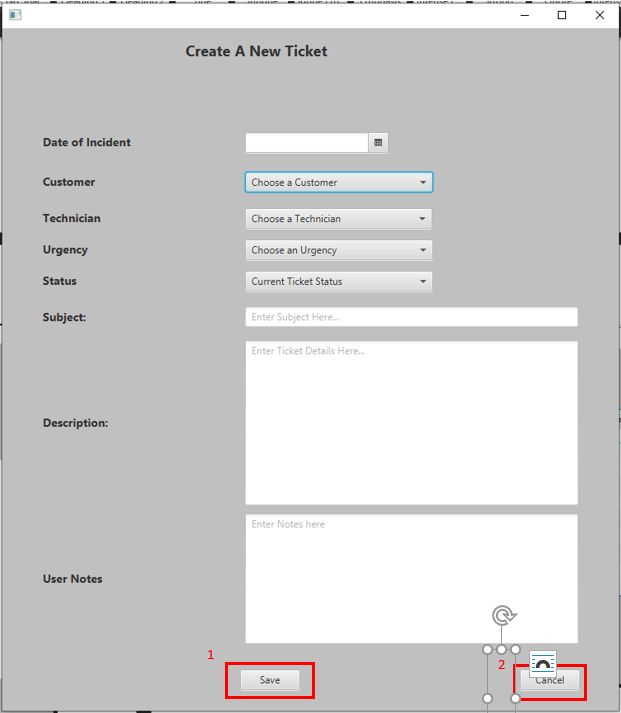
1. Click Search
2. The results will be displayed in the table. If there are no results an alert will be displayed.



## *Navigation-Add Tickets Menu*

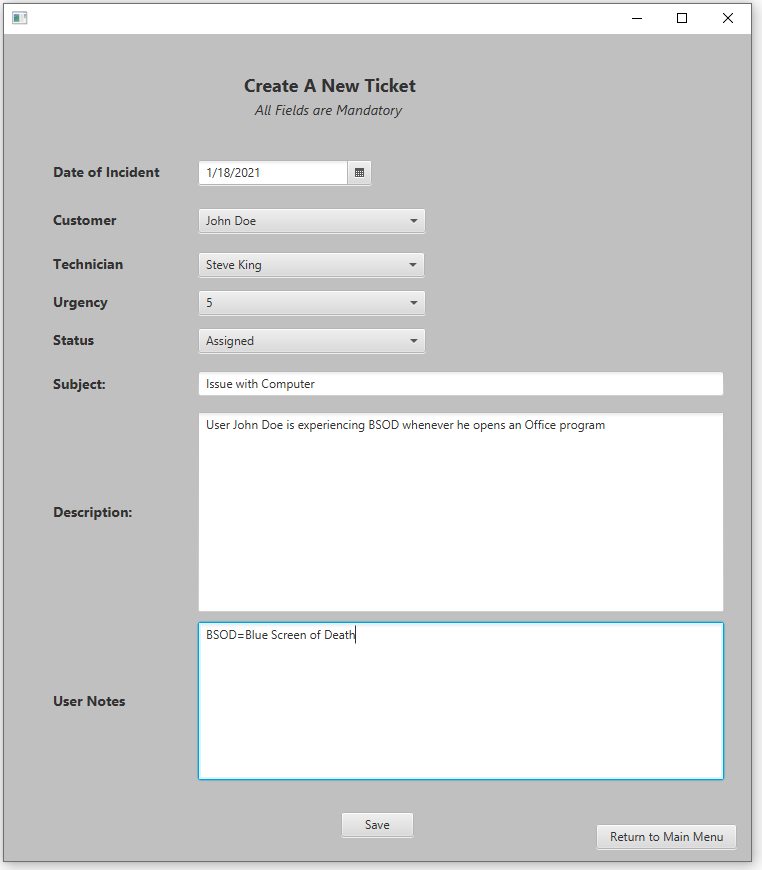
1. The save button saves the ticket to the database and returns the user to the Main Ticket Screen.

2. The Cancel button returns the user to the Main Ticket Screen



## *Creating a Ticket -Add Ticket Menu*

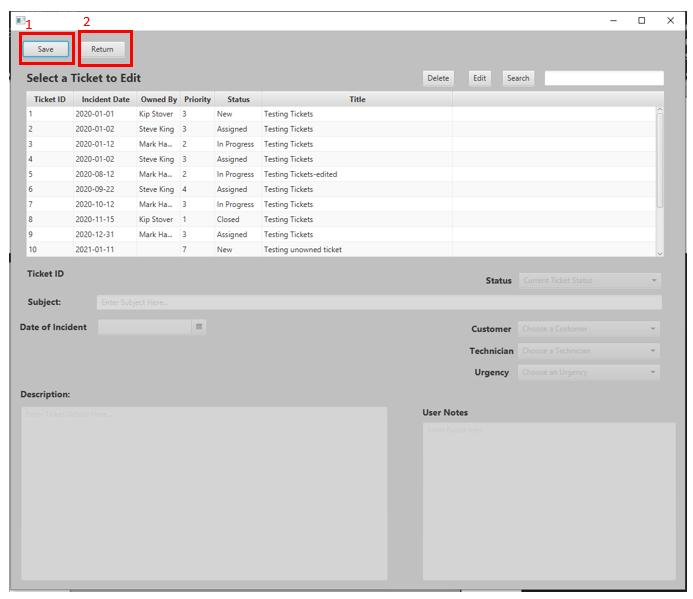
1. Starting from the Date field, fill in each field.
   1. Date Field can not select a date past the current date.
   2. **All fields must be filled in.**



1. Once all fields are filled, select Save.
   1. The ticket will not be saved unless all fields are filled.

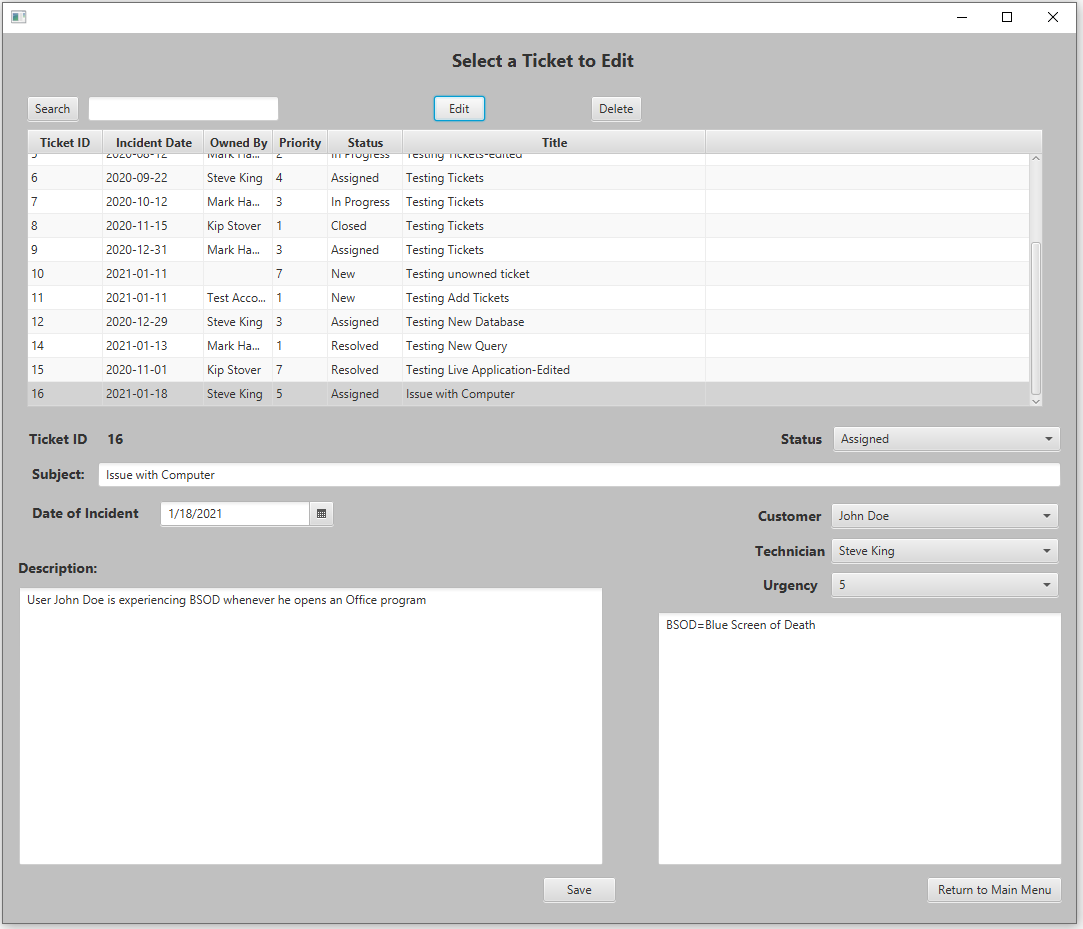
## *Navigation-Edit Tickets Menu*

1. The save button updates the ticket to the database and clears the fields for a new selection.
2. The return button returns the user to the Main Ticket Screen



## *Editing a Ticket – Edit Ticket Menu*

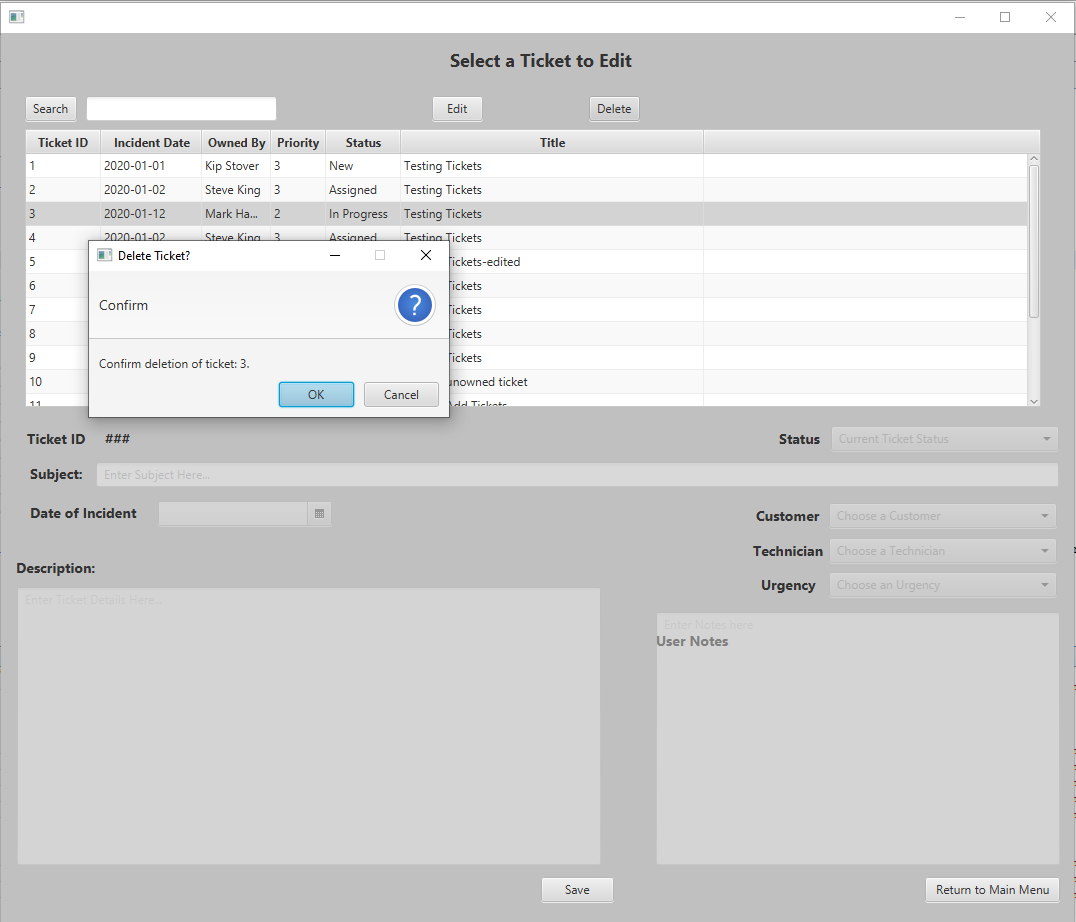
1. Select a ticket from the table.
2. Click Edit.
3. When the fields are no longer locked, they may be edited.



1. Edit which fields need editing.
   1. **All fields must be filled before saving**
2. Click Save when done.

## *Deleting a Ticket – Edit Ticket Menu*

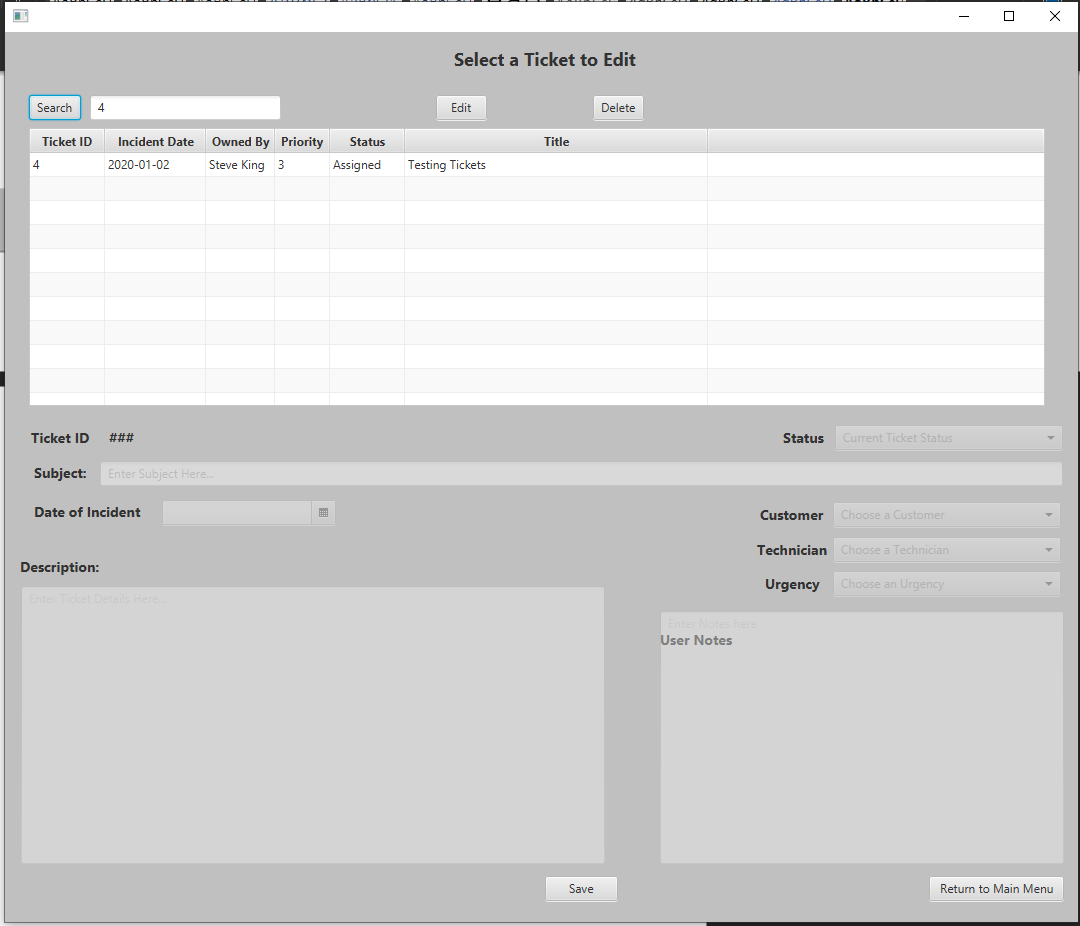
1. Select a ticket from the table.
2. Click Delete
3. An Alert will ask you to confirm deletion.



1. Click OK.

## *Searching for a Ticket – Edit Ticket Menu*

1. Enter either a Ticket ID or full Technician name into the search bar.
2. Click Search
3. Results will be displayed in the Table.



1. If there are no results, an error message will be displayed.

