Maria Thomas

Project Documentation

Executive Summary

Background

The client for this project is the System Administrator (SA) Department at a certain company that needs a Ticketing system. The SA department plays a crucial part in supporting general on-premise administration of Microsoft and Linux operating system, server and storage hardware, and enterprise applications including installation, configuration, maintenance, troubleshooting, and support of all departments. The main goal is to have an organized ticketing system between the user, SA team members, and SA lead.

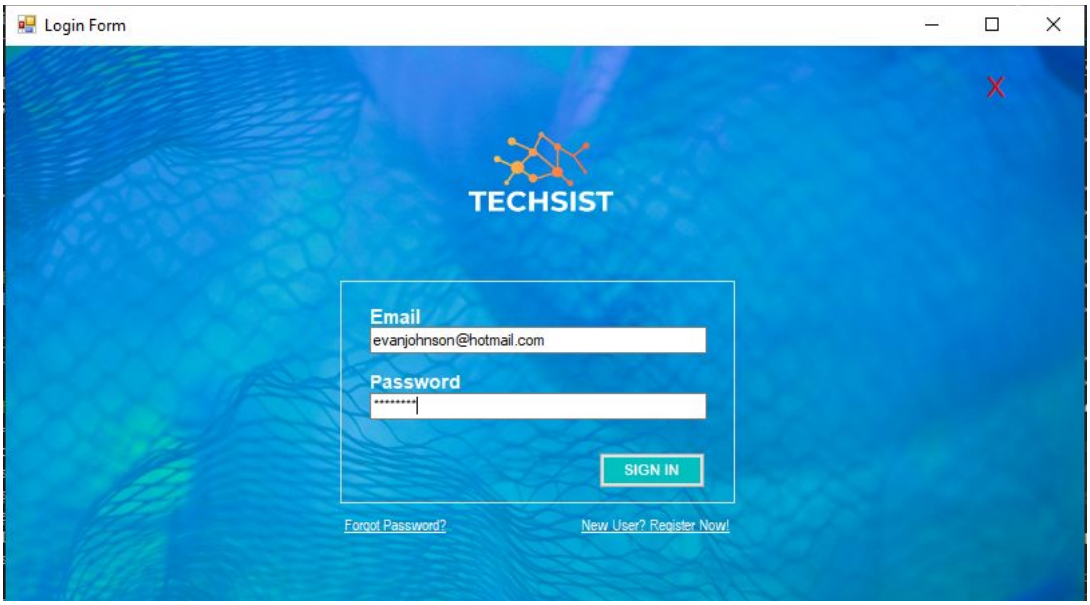
Project Description

The SA department currently is unable to track every trouble ticket, check who was assigned to troubleshoot and failed to monitor the status of every issue. The SA department has tried to hire more system administrator to support the backlog of the computer issues but still failed to complete the tasks since the SA department never had a history or documentation of the system. Solving this issue through an application system is important because the system would make the submission of the trouble ticket process and assigning the SA more efficient. It would also contain a history and documentation of the issue involving the actions have done which is very useful as future reference. It will also contain criticality information to indicate the prioritization status of every issue.

This project is created to organized the ticketing system that allows every employee from different departments to submit a trouble ticket faster, check who SA was assigned and status of the issue. It also provides the SA department team the option to assign who will troubleshoot the issue, displays the status and provides a summary data for the SA team lead.

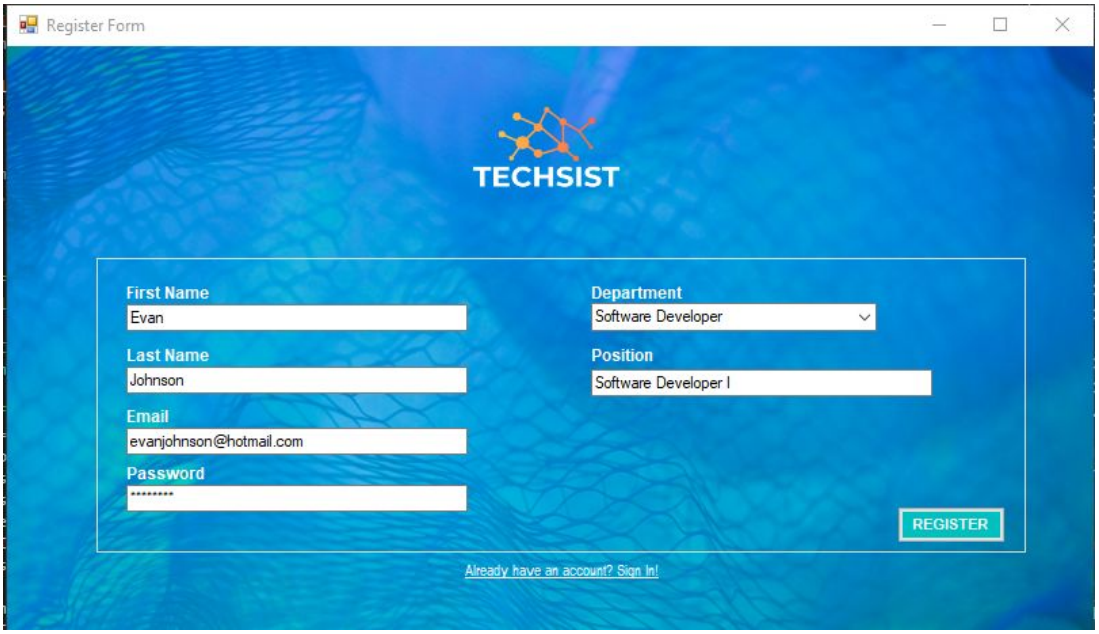
Scenario 1| Login Form

Login Form will first appear once the program is executed.



Note: If you are a new user, you can click the NewUser? Register Now link below the login form.

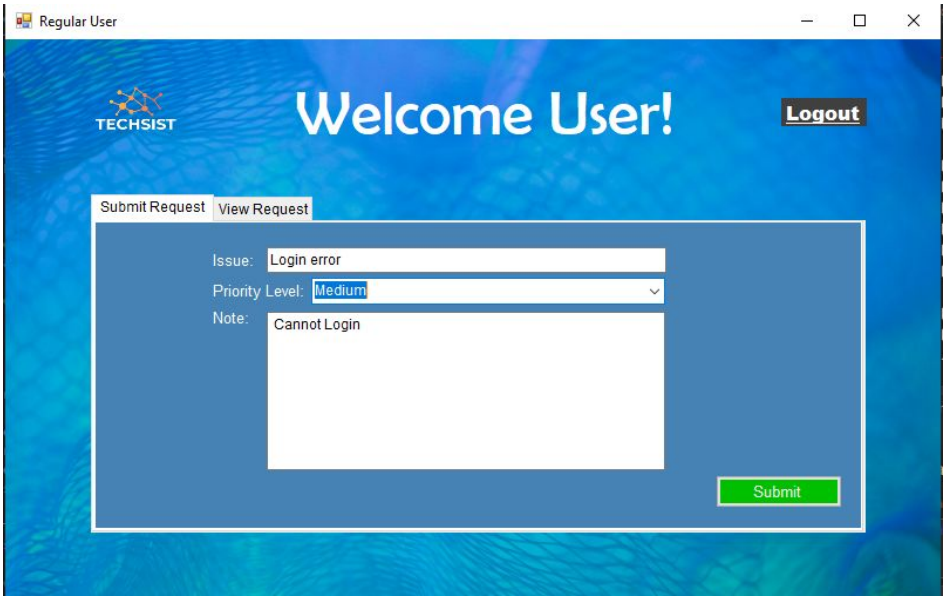
Scenario 2| Register Form



Register Form is intended for new users and once registered, it will direct back to login Form.

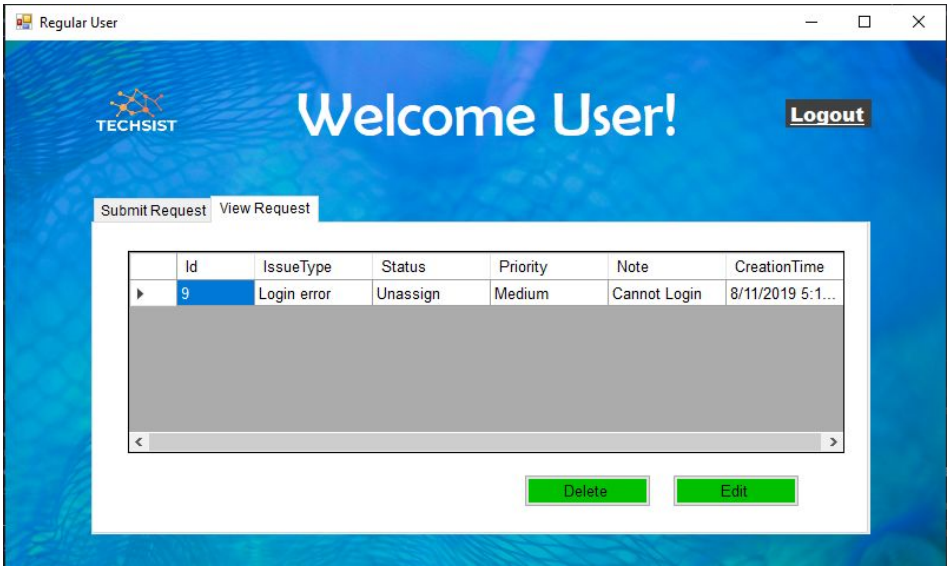
Scenario 3| Submit Trouble Ticket Form

This is the form used by regular user submitting a trouble ticket.



Scenario 4| View Requests by Regular User

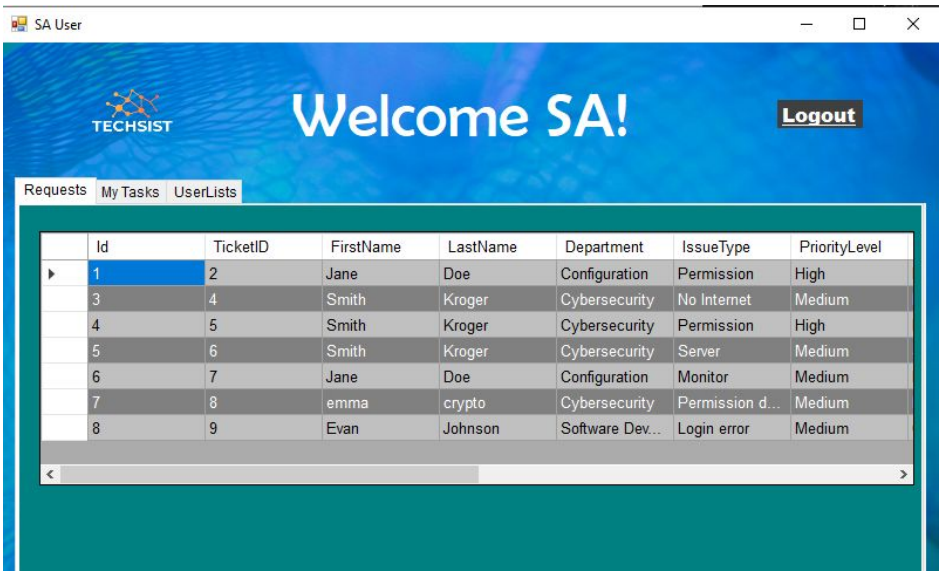
This is the view requests view which contains all the lists of submitted trouble ticket of the active user including the status.



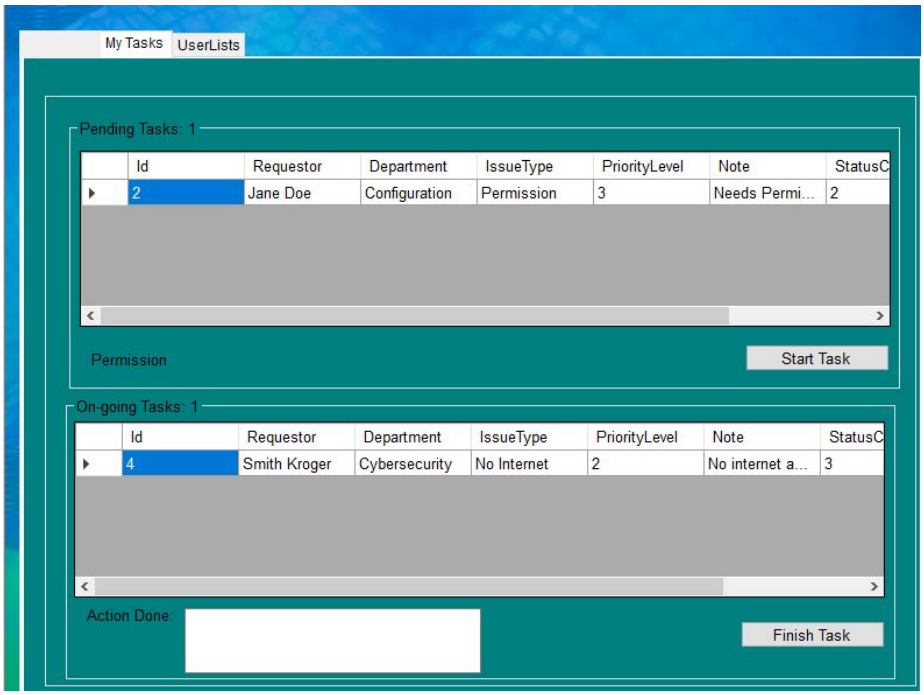
Scenario 5 | View and Functions of SA Member

This is the view for SA Member that had a view for all the requests and MyTasks tab that includes all the pending and in-process tasks.

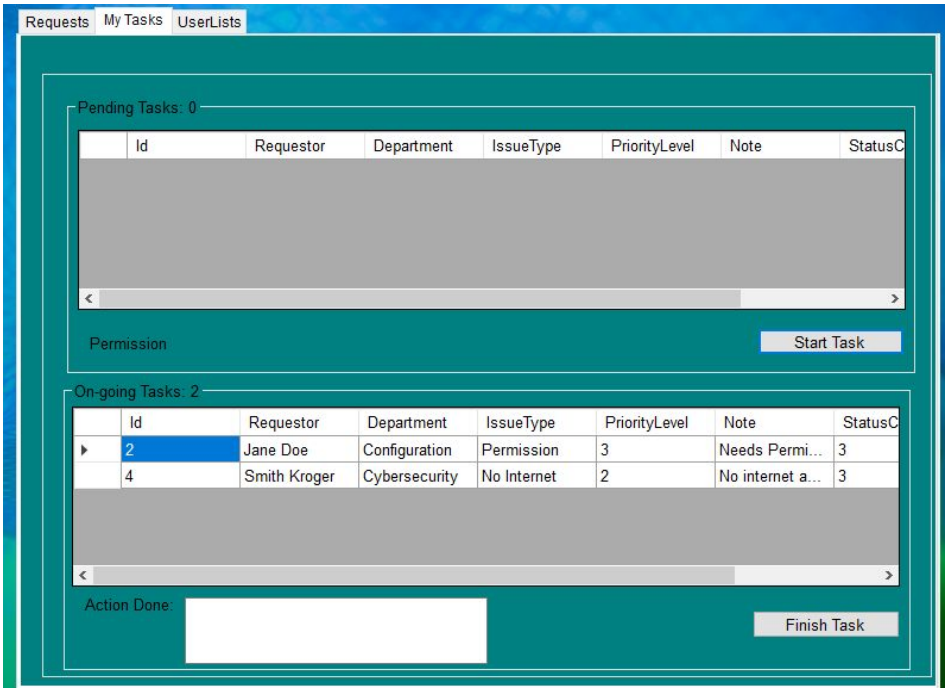
All Requests View



MyTasks View

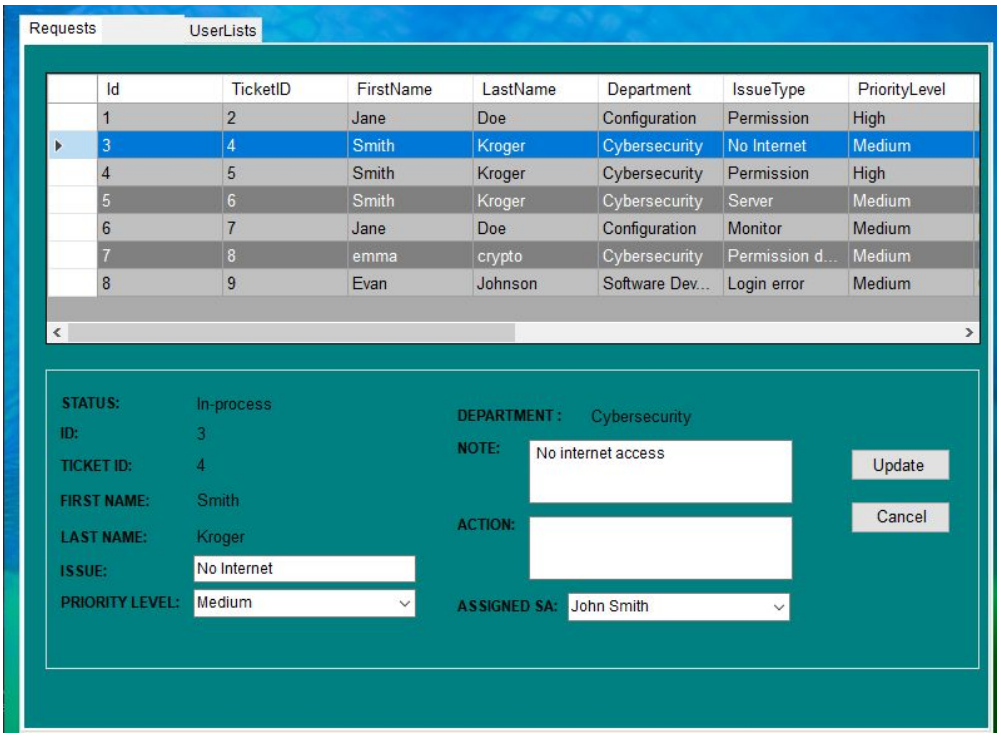


When Start Task is clicked, the pending task will moved to In-process/On-going tasks.



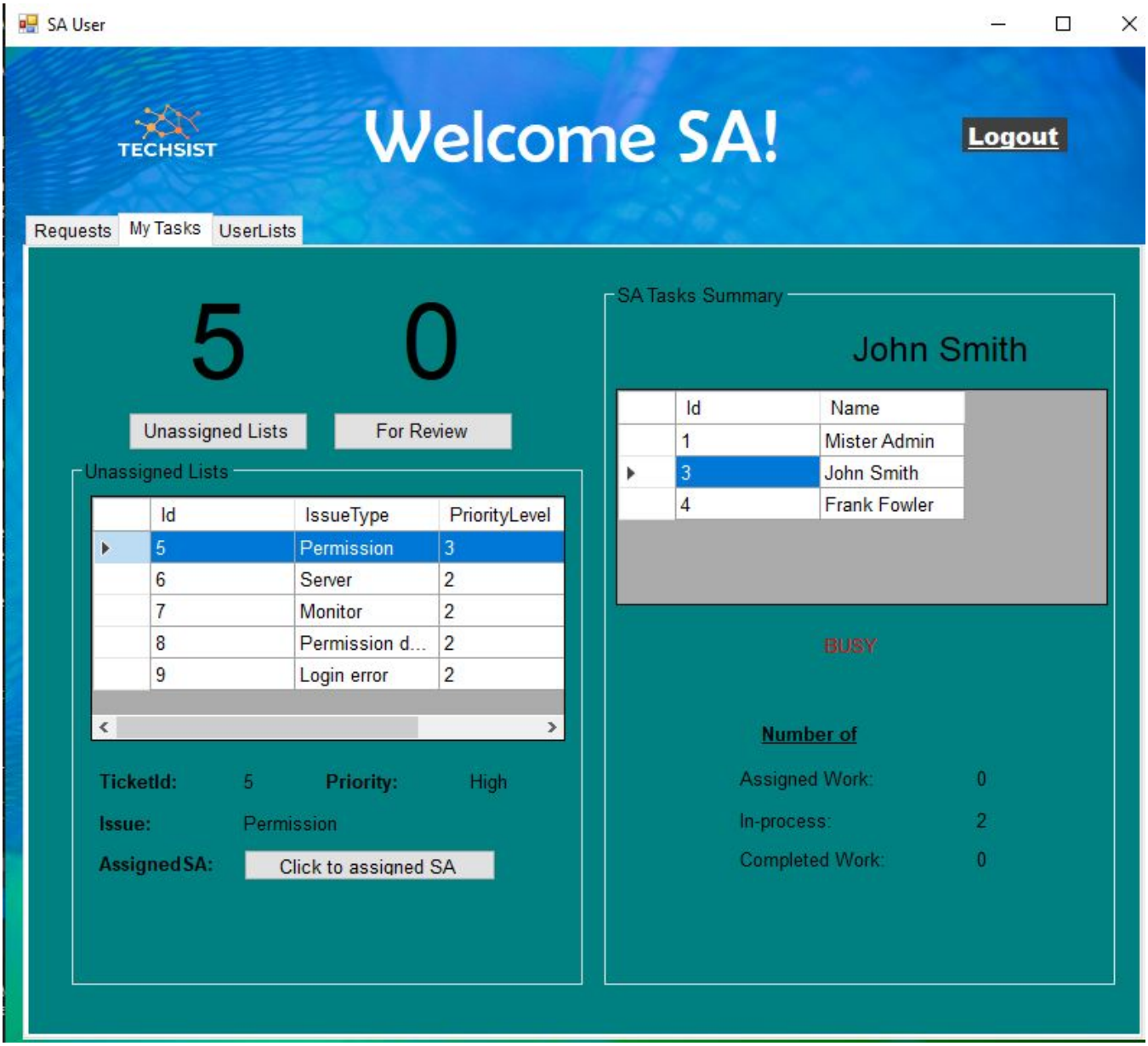
Scenario 6| SA Lead User Form

SA lead has more functions than a regular SA, a SA lead can edit the Requests and assign SAs.



Reports

Reports were mainly seen at the View Requests Form of SA Users where there is a table data that contains all the trouble ticket information. In addition, a report can also be manipulated by the SA lead where there is a number of unassigned tickets, for review and list of SA members that reports if SA is currently busy or available to assign the task.

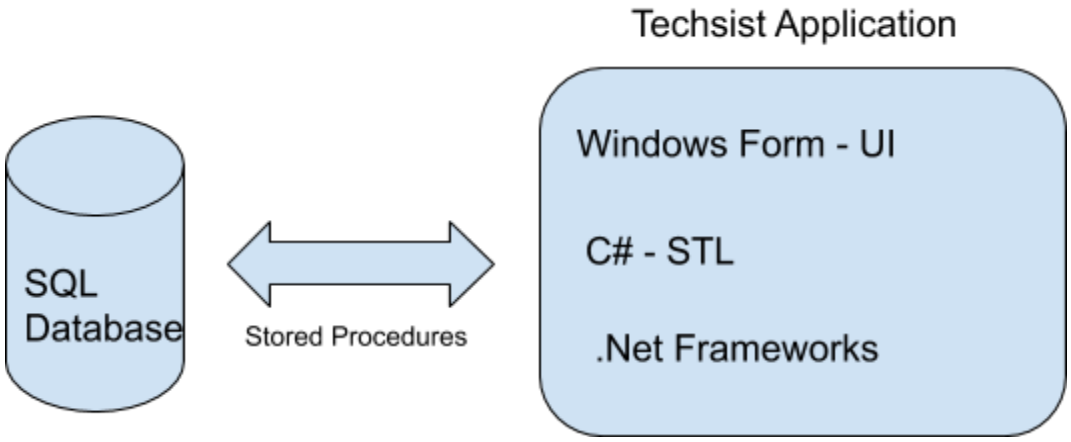


Notice that the selected SA member is flagged as Busy since In-process value is 2.

Note: Summary tab report is removed and downloadable pdf file is still in process.

System Architecture

Basically, the system of my application starts by creating the database and adding stored procedures which will be called using C# and Linq from the help of .Net frameworks and will be display using Windows Form tools.



Source Code Structure

The main directories of the source code are root, the main techsist file, properties, and resources.

Directory	Usage
/	Project root and contains the solution file of the visual studio and initial git control files.
/Techsist	Contains all the forms and classes of the file and other assembly folders
/Properties	Contains the properties of the visual studio like assembly info and settings.
/Resources	Contains the images needed in the application.

Highlighted rows indicate directories containing source code.

Executables

Installer (setup.exe)

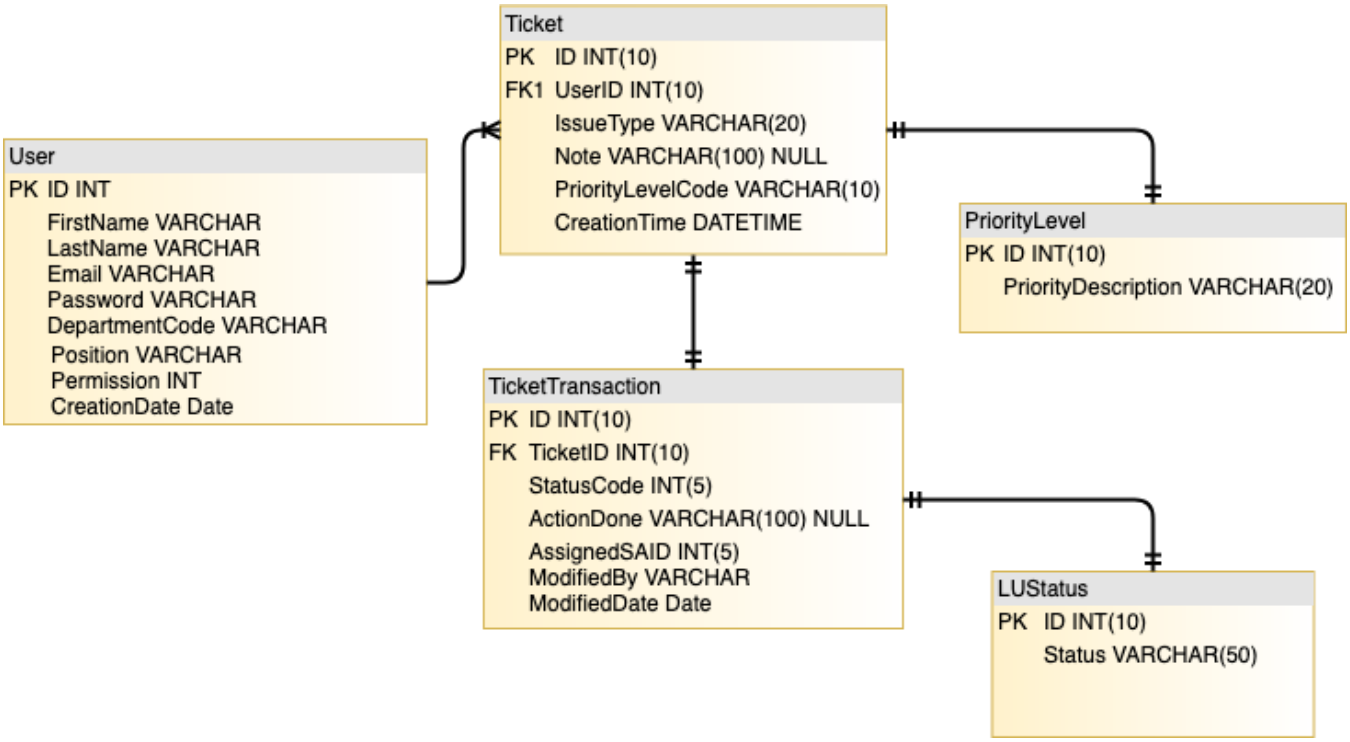
Install the Application and automatically opens the application.

Code Architecture

The heart of the program mainly relies on the database. It contains 5 tables which can be queried by calling all the stored procedures. Using LINQ, it was easy to access and manipulate the data and with the help of C# and .Net frameworks. The main view was implement using the Windows forms and it's corresponding tools. It has 3 different users that can manipulate the view through its permission data. There are two main forms the SA and Regular user plus login and registration form. It contains few sub-tabs that accepts and provides data.

Database or Data Store

The SQL database has 3 main tables and two lookup tables that are all queried by stored procedures for reports and updates.



Views, Stored Procedures and User Defined Functions

There are lot of stored procedures created in this database for the purpose of CRUD functionality and they are as follows:

Stored Procedures	Purpose
AssignedSA	Assign an SA to a ticket
CancelTicket	Delete a Ticket
GetDoneTicket	Set status to done
GetForReviewTicket	Report about all done status

GetSAMembers	Report all existing SA Members
GetSATasks	Report all tasks by SA
GetSATasksByID	Report SA Tasks by a given SA UserID
GetTicketTransactionList2	Report Summary of the ticket transaction
GetUnassignedTicket	Report the counts and data of the unassigned tickets
GetViewRequests	Report all the requests tickets
InsertTicket	Insert Ticket Information
UpdateTicketInformation	Edit or modifies the ticket information
InsertTicketTransaction	Insert Ticket Transaction Information
UpdateTicketTransactionInformation	Edit or modifies the ticket transaction information

Programming Language | C#.NET

Techsist Application is written in C# with .Net 4.6 Frameworks. It uses a Windows form Application for User Interface and different tools have been an instrument in displaying the data from SQL. It uses the data grid view and some labels to provide some reports of the data. Button functions have been used that contains some stored procedures call using LINQ queries.

Below are some of tools used in the application:

- Security.Cryptography for hashing password
- Drawing for color selections
- Linq for SQL queries
- DataGridView for data view
- SQLCL for database connection

Project Classes

Encrypt Password | Encrypt.cs
Encrypts the password by using the MD5Hash Function

Login Functions | Login.cs
Verifies if the user login information matches.

Priority description| Priority.cs
Gets the value of priority description by code and vice versa.

Status description| Status.cs
Gets the value of status description by code and vice versa.

SQL Queries | Query.cs

Contains all the queries from SQL using LINQ and stored procedures.

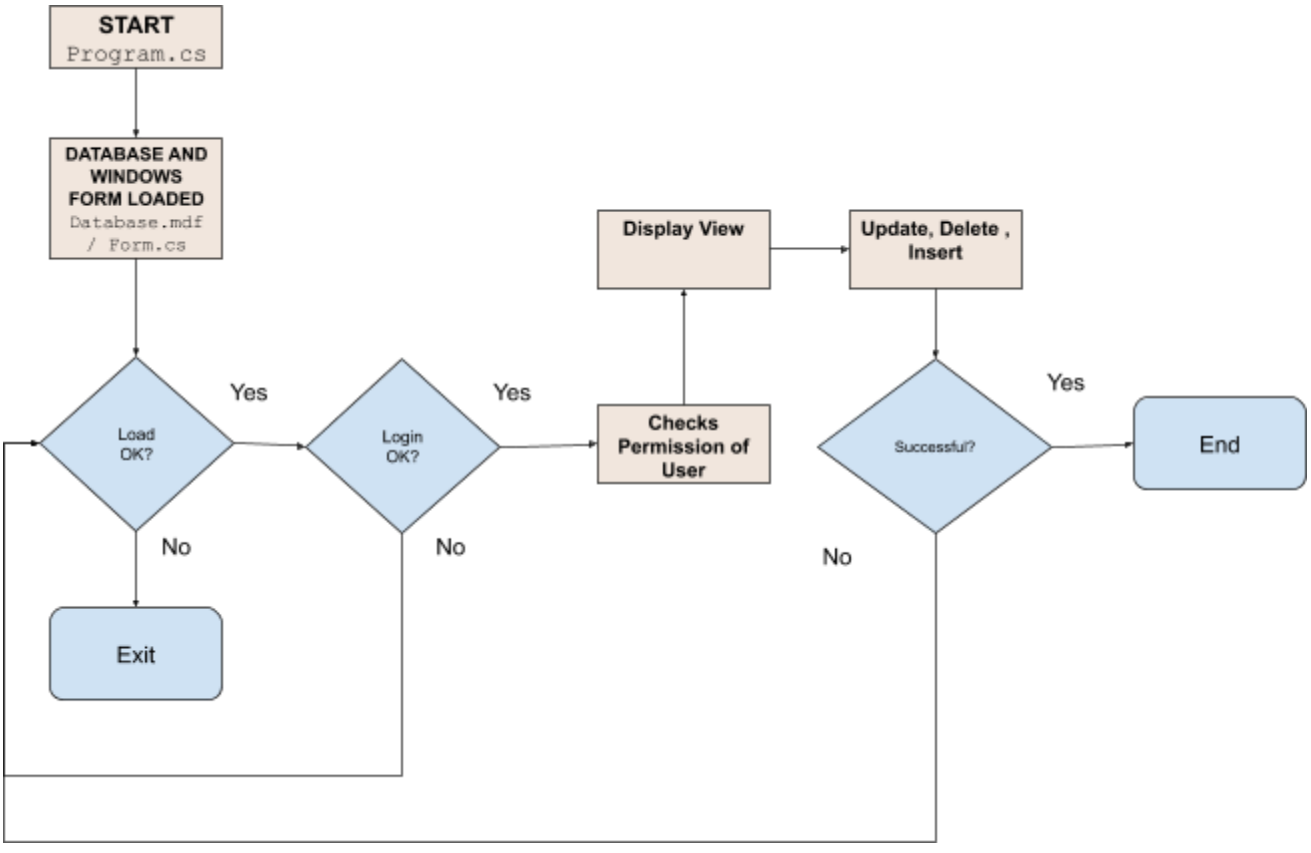
Project Modules

Entity Framework | Entity Framework.dll EntityFramework.SqlServer.dll

Needed for .Net framework for data purposes.

Program Start and End Flow

Describe and then diagram the program flow. Here is an example of program flow from a fat-client based PC application.



Summary

Techsist is an application written in C# using .Net Frameworks. Windows forms have been served as its User Interface design and SQL Database stored all the created data. It has 4 different forms containing the login, register, regular user form, and SA user form. It contains stored procedures for data calls and also Linq queries under the Query class. The database has 5 tables including the 3 main ones and 2 lookup tables. It uses some CTE and complex queries for retrieving complex data. Data grid view has been used in displaying all report stored procedures. Any encountered bugs will be fixed in the future and Improving UI and UX design is continuously working progress.

APPENDIX A (TESTING PLAN)

Scenario testing has been mainly implemented in this application. Different users have to login to assess if all views and functions are appropriate to the users. Typical scenarios for each user are:

1. Regular User
    a. Create login
    b. Submit Request
    c. Check Request
    d. Edit Request
    e. Delete Request
2. SA Member User
    a. View Request
    b. Check Tasks by the active SA Member
    c. Check User Lists
3. SA Lead
    a. View Request
    b. Edit Request
    c. Assign SA
    d. Approve Action
    e. View Summary Report
    f. Check User Lists

APPENDIX B (BUILD AND RELEASE PROCESS)

To build and release a new build, first determine the correct branch to change. To add new features and functionality, check out the commits and the master branch. To issue a bug fix to the released version, checkout the readme.MD file for further information. Any bug fixes and updates will be listed in the readme file.

APPENDIX C (CLIENT INSTALLATION INSTRUCTIONS)

The setup folder and executable file can be downloaded or copied to any device as long as it has a local connection to the database. This folder contains:

- setup.exe
- Application folder
- Techsist (Click once Application)

APPENDIX D (DEVELOPER SETUP INSTRUCTIONS)

The application requires to install the following tools:

- Microsoft Visual Studio
- C# lib
- SQL Database
- .Net Framework 4.6

Setup Instructions:

1. Clone this repository
2. Open it in Visual Studio
3. Right Click the Techsist from the Solution Explorer and click Publish
4. Click the Setup.exe