



IT Management System

System Design Document

Latest Revision: 05/06/2022

PROJECT URL

<https://github.com/comp195/senior-project-spring-2022-it-management-system.git>

DEVELOPERS

Daniel Adler

(d_adler@u.pacific.edu)

Andrew Vu

(a_vu16@u.pacific.edu)

Vincent Tran

(v_tran37@u.pacific.edu)

System Architecture

This system is separated into two components: The software application used by high permission level internal parties in order to access and modify relevant information for the business, and the web application provides other employees with the ability to submit tickets regarding maintenance and view relevant information.

Desktop Application

Software Modules

This software is composed of a client-side application where users can access and modify information about their businesses and their assets from both a management perspective and an employee perspective with different permissions assigned to each role.

Hardware Modules

This application is designed to run on a computer running Windows 10 and above.

The User Interface Modules

- This application consists of a series of panels that provide users with access to various different components within the IT management system.
- The **Login** panel provides employees and management with the ability to log in with their own credentials, preventing people with insufficient permissions from accessing private data.
- The **Equipment** panel provides employees with the ability to check equipment in and out while also allowing management to append or remove equipment from this stack.

- The **Tickets** panel allows the user to view all issue tickets submitted by staff and allows them to be sorted by priority, category, and user.
- The **Employees** panel provides information about each employee, including the equipment they've checked out, their IDs, timecard information, etc;

Interfaces to External Systems

The desktop client requires access to an external database in order to both retrieve and update information regarding tickets, employees, and other business information. The intended database for this use case is handled through Amazon Web Services (AWS).

Web Application

Software Modules

This part of the project contains the web client that is used by general employees of the company. It was developed using HTML, CSS, Javascript, and PHP. Its main purpose is to provide employees with an interactive interface to communicate with management and/or the IT support team and view tickets.

Hardware Modules

This web application is designed under the assumption that the user is accessing it through a desktop browser.

The User Interface Modules

- The **Submission Form** consists of:
 - The **Employee ID** field serves as the main means of maintaining a unique identifier for the customer and is a required field. This, along with the **First Name** and **Last Name** fields, are automatically populated based on the credentials provided during login.
 - The **Short Description** text box allows the customer to enter a short description of the issue to be addressed.

- The **Long Description** text box serves as the main point in which the customer can go further into detail and describe the issue that is to be addressed.
- There are **Dropdown Lists** to help categorize and specify the customer's issue: **Ticket Category** (such as **New Equipment Request**, **Software Issue**, and **Damaged Equipment**), **Issue Scope** (individual, team, departmental, or larger), **Ticket Classification**, etc.
- The **Ticket View** consists of a grid view where the user is able to see information regarding all tickets associated with that user

Interfaces to External Systems

The web client requires access to an external MySQL database (which was set up using AWS RDS) to both retrieve and store ticket information, as well as to verify login credentials. The web client is hosted on an AWS EC2 instance in order to be accessed remotely through the web on any device with access to the Internet. It is used to update information regarding customer tickets, which will then be retrieved and potentially updated through the desktop client's access to the data.

Hardware, Software, and System Requirements

Desktop Application

Hardware Requirements:

RAM: 2 GB RAM

Storage: 1GB

Network: Broadband Internet connect

Software Requirements:

Python 3

System Requirements:

Windows 10/11

Web Application

Hardware Requirements:

Network: Broadband Internet connect

Software Requirements:

Web Browsers: Google Chrome, Firefox, Safari, Microsoft Edge, etc.

Python 3, HTML 5, CSS

System Requirements:

Windows 10/11, Linux, MacOS

External Interfaces

AWS:

AWS is the host for the database to store the information for our application.

<https://docs.aws.amazon.com/rds/index.html>

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html>

FuzzyWuzzy:

The library searches for matches and near-matches of a given text and allows us to create a search functionality within the application.

<https://pypi.org/project/fuzzywuzzy/>

bCrypt:

Encrypts data sent between Desktop Client and RDS database

<https://pypi.org/project/bcrypt/>

Tkinter:

Provides essential tools for constructing the user interface

<https://docs.python.org/3/library/tkinter.html>

Paramiko:

Utilized to establish a connection between the desktop client and the EC2 instance

<https://www.paramiko.org/>

Pillow:

Used to display image attachments associated with tickets.

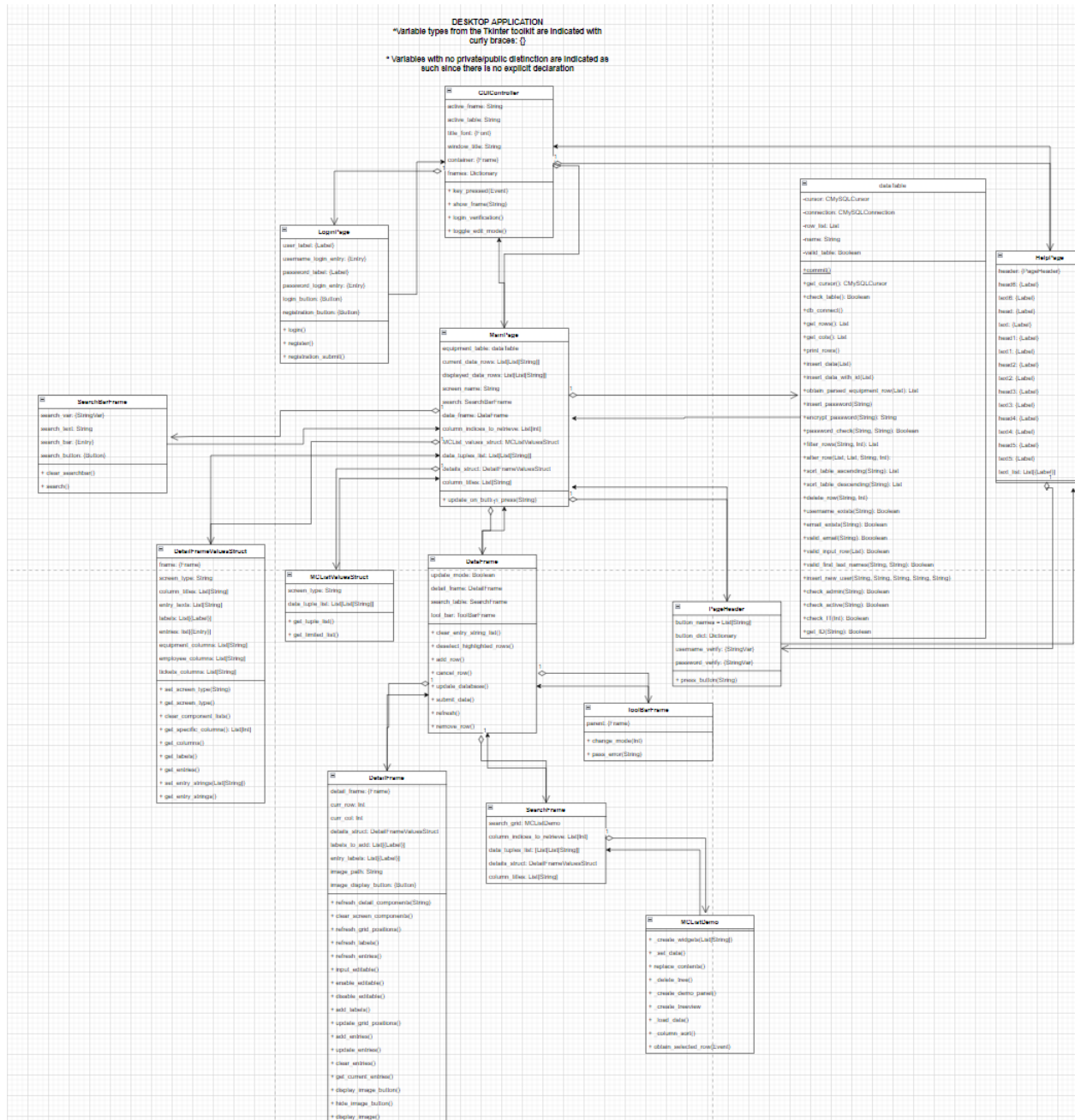
<https://python-pillow.org/>

PyMySQL:

PyMySQL allows us to connect to a SQL database.

<https://pypi.org/project/PyMySQL/>

Class Diagrams

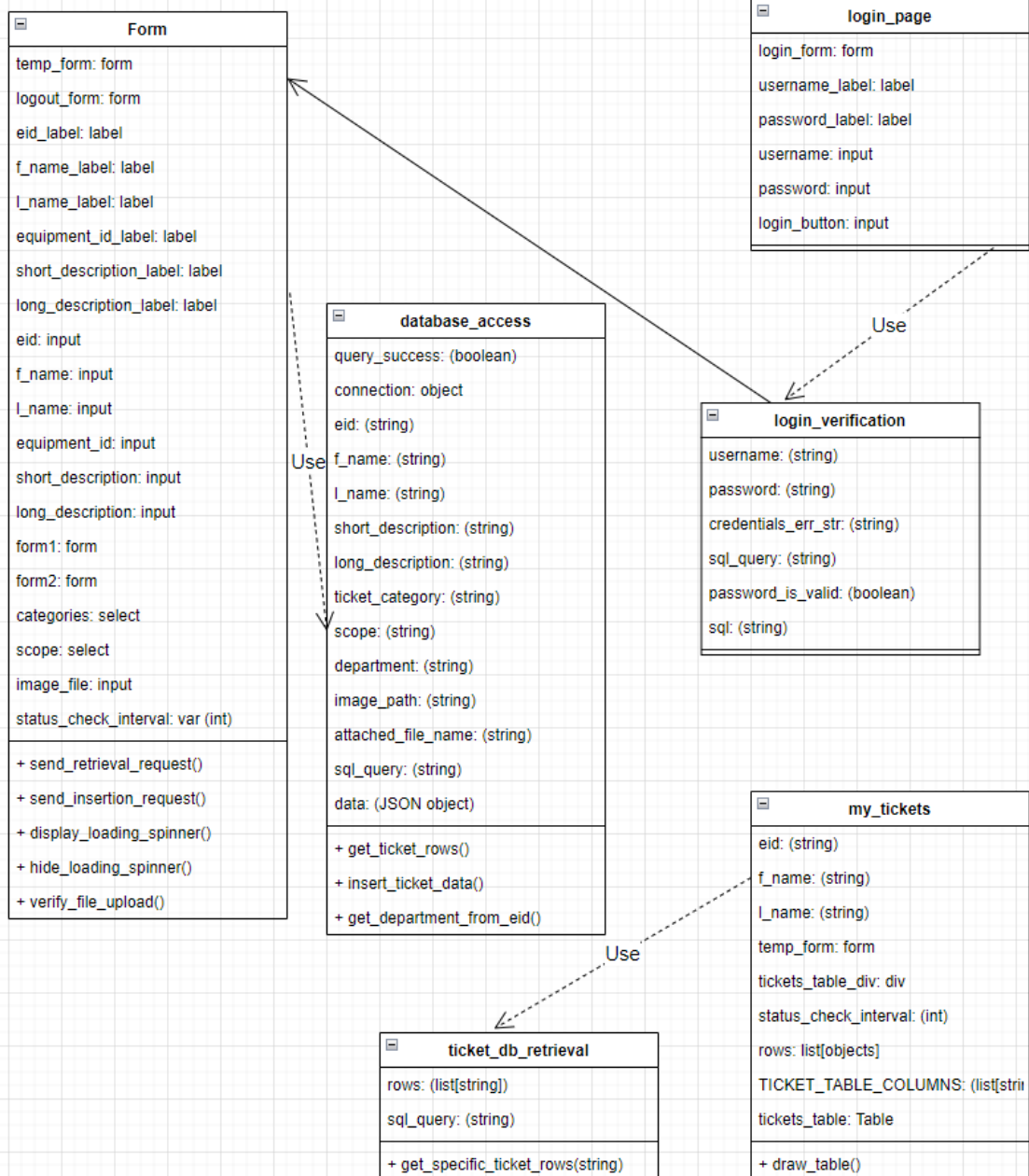


WEB CLIENT

*No classes in particular, so depicting based on client pages and script files

*Generally no return values for these functions

* for variable types placed in parentheses, language used does not have explicit type declarations

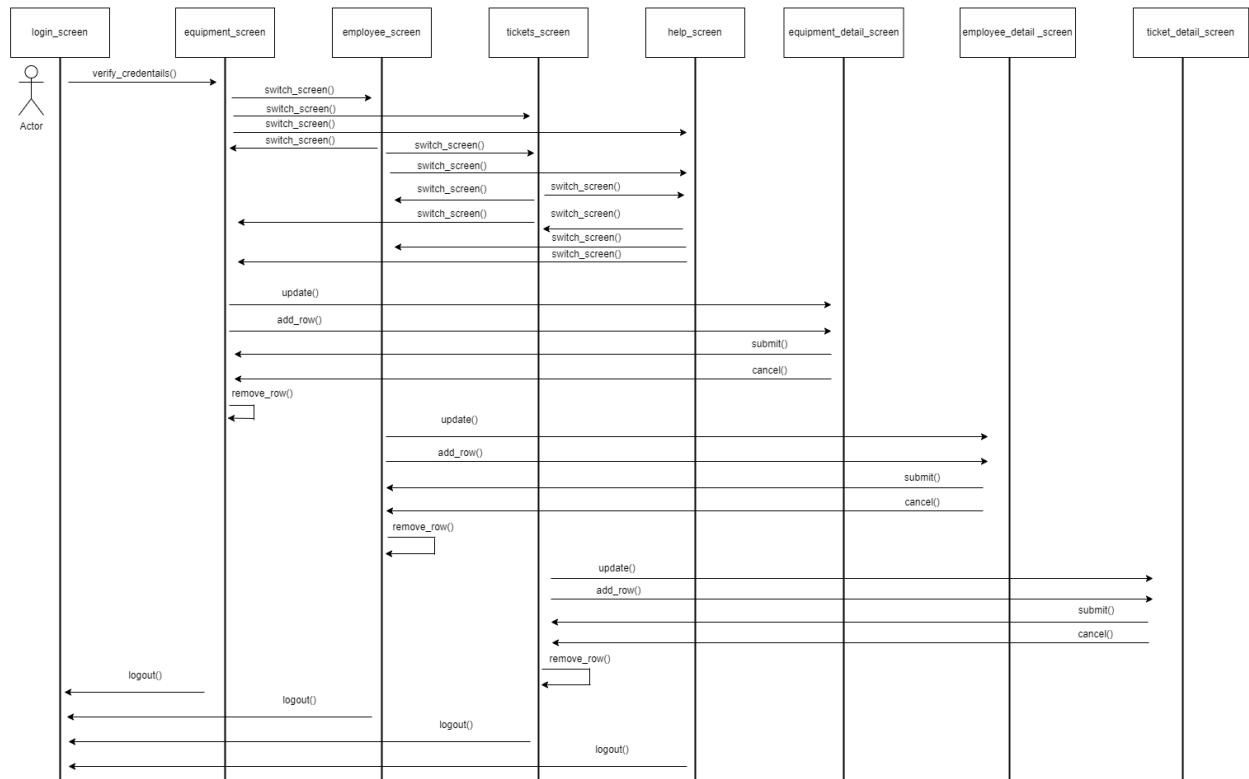


*Link to view the UML diagrams in larger size

<https://tinyurl.com/2kdwr8tb>

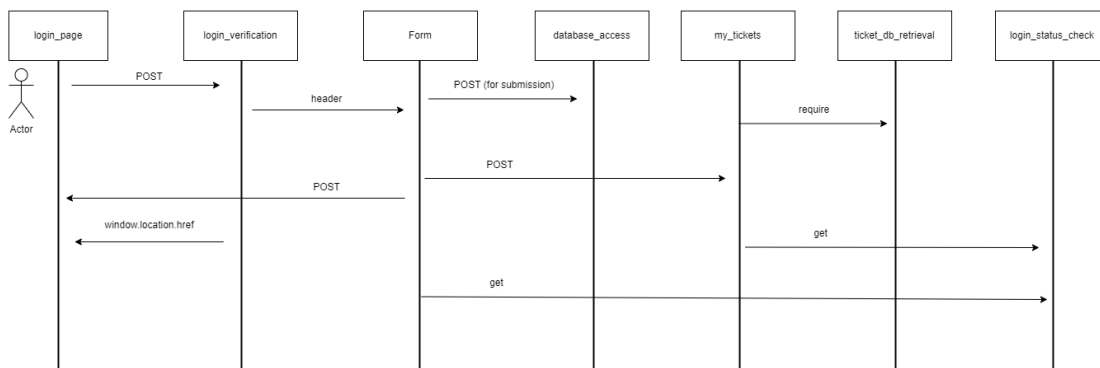
Interaction Diagrams

Desktop Client



Web Application

*Since no classes in Web Client specifically, using pages for interactions



*Link to view the interaction diagrams in larger size

<https://tinyurl.com/6z9zvykr>

Design Considerations

Design Principle: Accommodate Change

In designing the overall structure of the system, being able to accommodate for the change in the application's need of various business information details and components was taken into consideration. For example, while the IT ticketing system is a significant feature, a business's management may want to incorporate other management options to improve workflow.

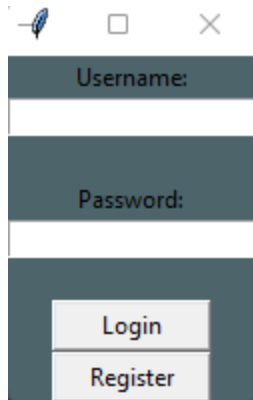
Design Principle: Design for Testability

The idea behind the splitting of the system into three main components (database, desktop client, and web client) is that they can easily assist in the testing of each other's functionality. For instance, the web application's main purpose is to create a ticket, which can then be checked through the database for actual updates in the data. Furthermore, this can be confirmed through the desktop client's pulling of data from the database, and the desktop application's functionality of making changes to the central data can be checked back through the database as well.

User Interface Design

Desktop Client

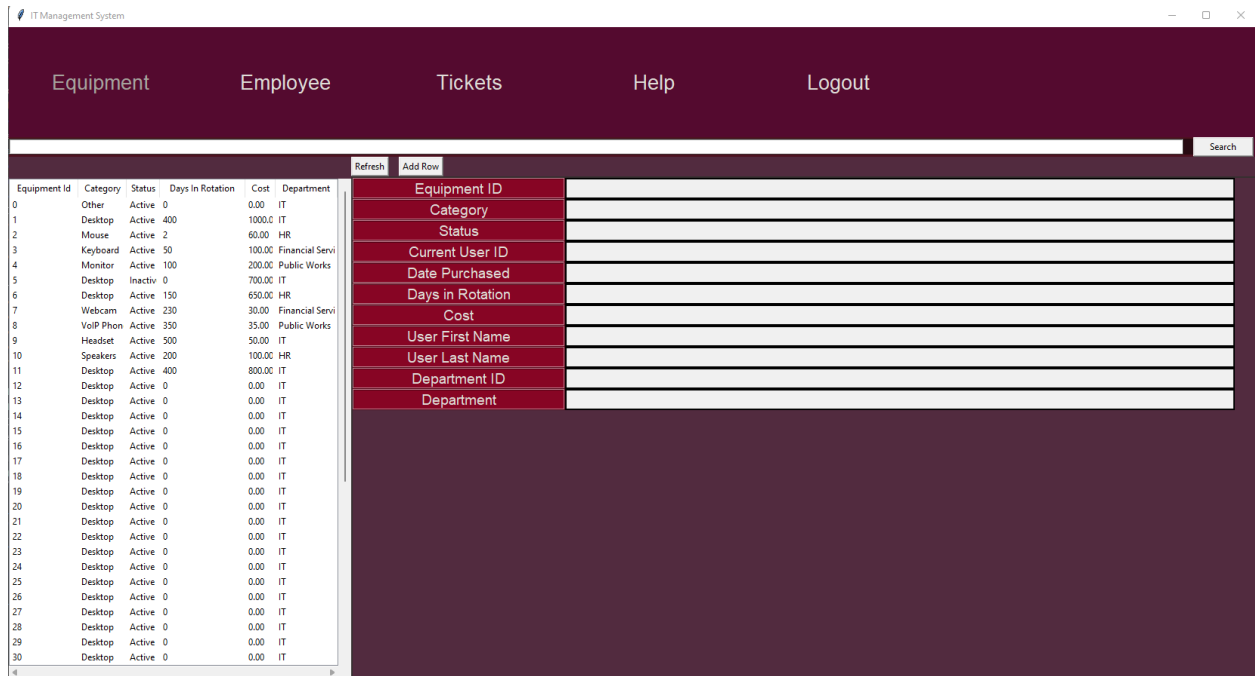
Login Screen



This screen serves the purpose of allowing the user to login.

- **The Username Box** allows the user to enter their email address.
- **The Password Box** allows the user to enter their password.
- **The Register Button** validates the information entered by the user and grants them access into the main application if the login information is correct and they are an authorized user
- **The Register Button** opens up a window that allows the user to create a new inactive account

Database-connected Screens (Equipment, Employees, Ticketing from IT Perspective)

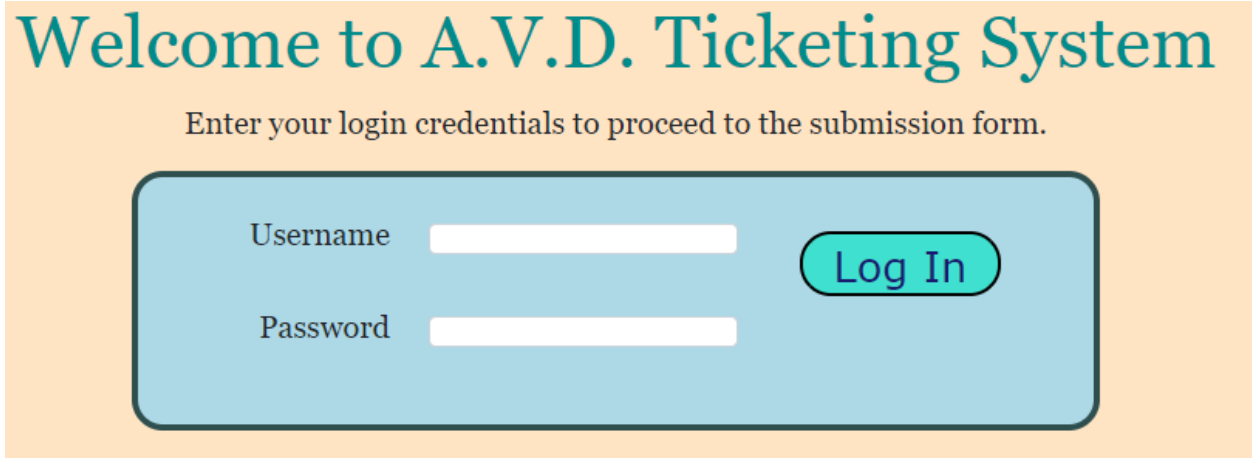


The Database Screens allow the user to see a variety of information about the assets and employees that a business has and make modifications to that data.

- **The Tab Buttons** allows users to select which information they want to view or modify. These are labeled as Equipment, Employees, Ticket, etc.
- **The Search Bar** allows users to try to find an item within the list view based on a fuzzy searching algorithm by typing in either a keyword or item name. The user can then click on the **Search Button** and receive a reduced list based on the criteria provided in the search.
- **The Table Grid** shows users a list of item boxes based on the contents of the search bar.
- **The Toolbar** consists of multiple buttons that allow users to perform edit operations on the currently selected row such as add, update, refresh, remove, etc;
- **The Details View** provides users with more information about the current row than is available immediately within **The Table Grid**

Web Client

Login Page



The screenshot shows a login page with a light orange background. At the top, the text 'Welcome to A.V.D. Ticketing System' is displayed in a large, teal, serif font. Below this, a smaller teal serif font says 'Enter your login credentials to proceed to the submission form.' In the center, there is a light blue rounded rectangle containing two white input fields. The first field is labeled 'Username' and the second is labeled 'Password'. To the right of these fields is a teal rounded button with the text 'Log In' in white.

Welcome to A.V.D. Ticketing System

Enter your login credentials to proceed to the submission form.

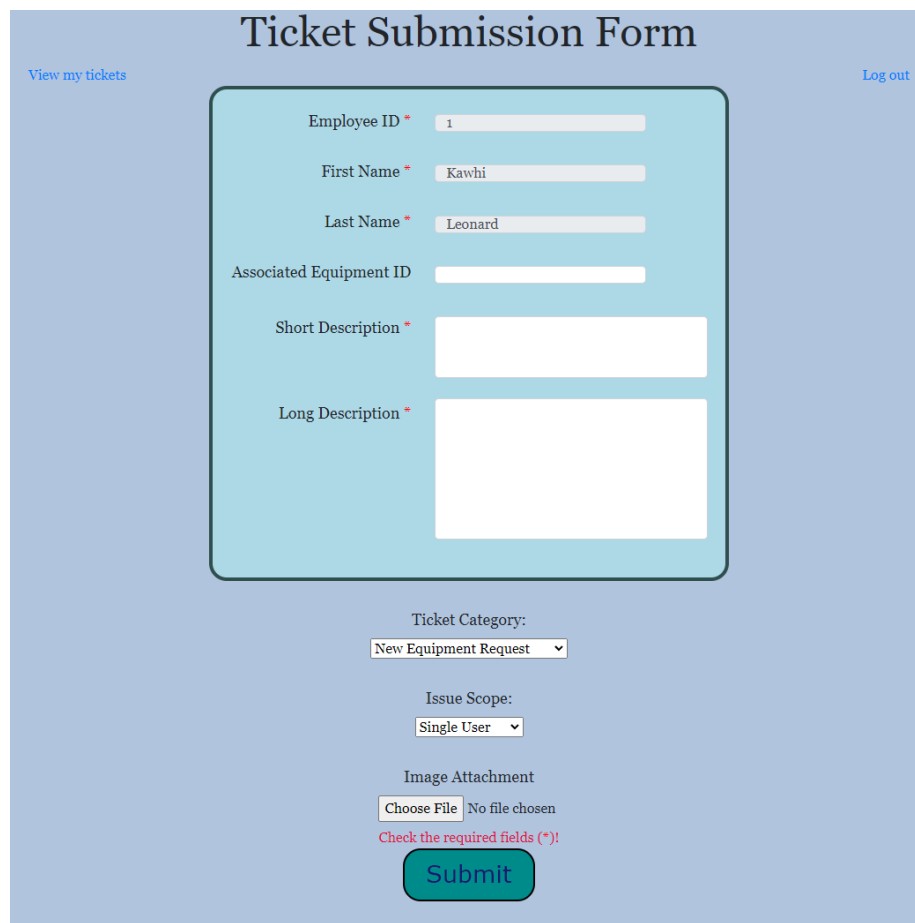
Username

Password

Log In

The Login Page contains two user input fields for the username and password. This page verifies that both a valid username and password are provided in order to proceed to the next screen when the login button is clicked. If invalid credentials are provided, the user will be redirected to this screen.

Ticket Screen



The screenshot shows a 'Ticket Submission Form' on a light blue background. At the top, the title 'Ticket Submission Form' is in a large, black, serif font. In the top left corner, there is a link 'View my tickets' and in the top right corner, a link 'Log out'. The form itself is a light blue rounded rectangle containing several input fields. The first three fields are 'Employee ID *', 'First Name *', and 'Last Name *', each with a value entered. Below these is 'Associated Equipment ID'. Then are 'Short Description *' and 'Long Description *', both with text areas. Below the form, there are two dropdown menus: 'Ticket Category:' with 'New Equipment Request' selected, and 'Issue Scope:' with 'Single User' selected. Below these is an 'Image Attachment' section with a 'Choose File' button and the text 'No file chosen'. At the bottom, there is a red text prompt 'Check the required fields (*)!' and a teal rounded button labeled 'Submit'.

Ticket Submission Form

[View my tickets](#) [Log out](#)

Employee ID *

First Name *

Last Name *

Associated Equipment ID

Short Description *

Long Description *

Ticket Category:

Issue Scope:

Image Attachment
 No file chosen

Check the required fields (*)!

Submit

The Ticket Screen provides users with an interface to submit tickets regarding any issues they may be experiencing.

- **The Employee ID, First Name, and Last Name fields** are automatically populated based on the user credentials provided at the login screen. These are uneditable.
- **The Associated Equipment ID field** allows for the user to optionally provide an equipment ID that may be involved in the issue at hand.
- **The Short Description and Long Description fields** are text boxes that can be used to provide an explanation of the issue at hand. Short Description is meant for a brief idea of the problem, while Long Description is meant to be used as an in-depth explanation point.
- **The Ticket Category and Issue Scope fields** are dropdown lists that provide the employee with predetermined options pertaining to information regarding the issue. Ticket Category is used to group the ticket based on certain qualities of the problem, while Issue Scope is meant to represent the amount of people affected.
- **The Image Attachment field** allows the user to select an image file to upload as part of their ticket. There are checks in place to confirm that the size of the file stays below 3MB and that the file is of one of these types: JPG, JPEG, PNG.
- **The user feedback message** (as indicated in red above) displays any message that needs to be conveyed, such as a ticket submission attempt failing or succeeding.
- **The Submit Button** is used to actually proceed with the submission attempt. This button triggers checks to determine if the ticket submission was successful.

My Tickets View

My Tickets												
Return to Ticket Submission Form												
Ticket Number	Ticket Status	Client ID	Client First Name	Client Last Name	Equipment ID	Ticket Category	Short Description	Full Description	Issue Scope	Priority	Department	
1 2	Under Review	2	Jude	Brabham	2	Equipment replacement	Need new mouse	Mouse is defective. Scroll wheel doesn't work.	Single User	High	HR	
2 56	Submitted	2	Jude	Brabham	1	New equipment request	should work	should work 4/25/22	Single User	Low	HR	
3 57	Submitted	2	Jude	Brabham	1	Damaged equipment	sample	sample2	Team	Medium	HR	
4 58	Submitted	2	Jude	Brabham	2	Equipment replacement	123214	122512512	Single User	Low	HR	
5 59	Submitted	2	Jude	Brabham	2	New equipment request	t1	t2	Single User	Low	HR	
6 60	Submitted	2	Jude	Brabham	2	New equipment request	14124214	142124124124	Single User	Low	HR	

The My Tickets View is provided to show the tickets submitted for the user currently logged in. It displays each ticket as a row of fields as shown above and can be sorted by field values.

Glossary of Terms

Application (n.)

A program that runs on a computer. Software applications run on the desktop while web applications run on a browser.

AWS (n.)

[Amazon Web Services] Services provided by Amazon to handle operations including database management and remote server maintenance

Dropdown List (n.)

A user interface component that allows the user to select an option from a list that is revealed to the user upon clicking on a labeled button.

Incident (n.)

A ticket type in which the user is experiencing a problem (as opposed to a request). For example, an internet outage issue is classified as an incident.

IT Team (n.)

The individual or group of individuals responsible for the maintenance of data stored within this application.

Log in (v.)

Entering username and password credentials for the purpose of only allowing users with a sufficient permission level to access the data available to those at that level.

Management (n.)

High-level parties who are responsible for the assets and processes that occur within an organization.

Panel (n.)

A user interface component that represents everything visible to the user at a specific point in time.

Permission Level (n.)

A value that represents the amount of information that a specific user has access to. A higher permission level indicates that the user has access to more data and vice versa.

Perspective (n.)

Describes what each user sees within the software application. Users with a higher permission level have a wider perspective than users with a low permission level.

Request (n.)

A ticket type in which the customer needs an additional service; for instance, requesting upgraded equipment classifies as a request.

Ticket (n.)

A document submitted by a employee within the organization that references a specific issue said-user may be having with an asset or process within that organization.

Ticket Scope (n.)

The range of impact in which the issue has on a customer/user-base. For example, a scope value of “departmental” indicates that the issue affects the department as a whole and is therefore more serious than a ticket labeled with a scope of “individual”.

User (n.)

Any individual that utilizes the functionality offered by this application.

References

Okyol, O. (2020, August 4). *Using AWS RDS and python together*. Towards Data Science. Retrieved February 7, 2022, from <https://towardsdatascience.com/using-aws-rds-and-python-together-5718a6878e4c>.

Principles of Software Design. GeeksforGeeks. (2020, June 17). Retrieved February 7, 2022, from <https://www.geeksforgeeks.org/principles-of-software-design/>

Python tkinter tutorial. GeeksforGeeks. (2021, March 28). Retrieved February 7, 2022, from <https://www.geeksforgeeks.org/python-tkinter-tutorial/>.