

# Fixit Ticket

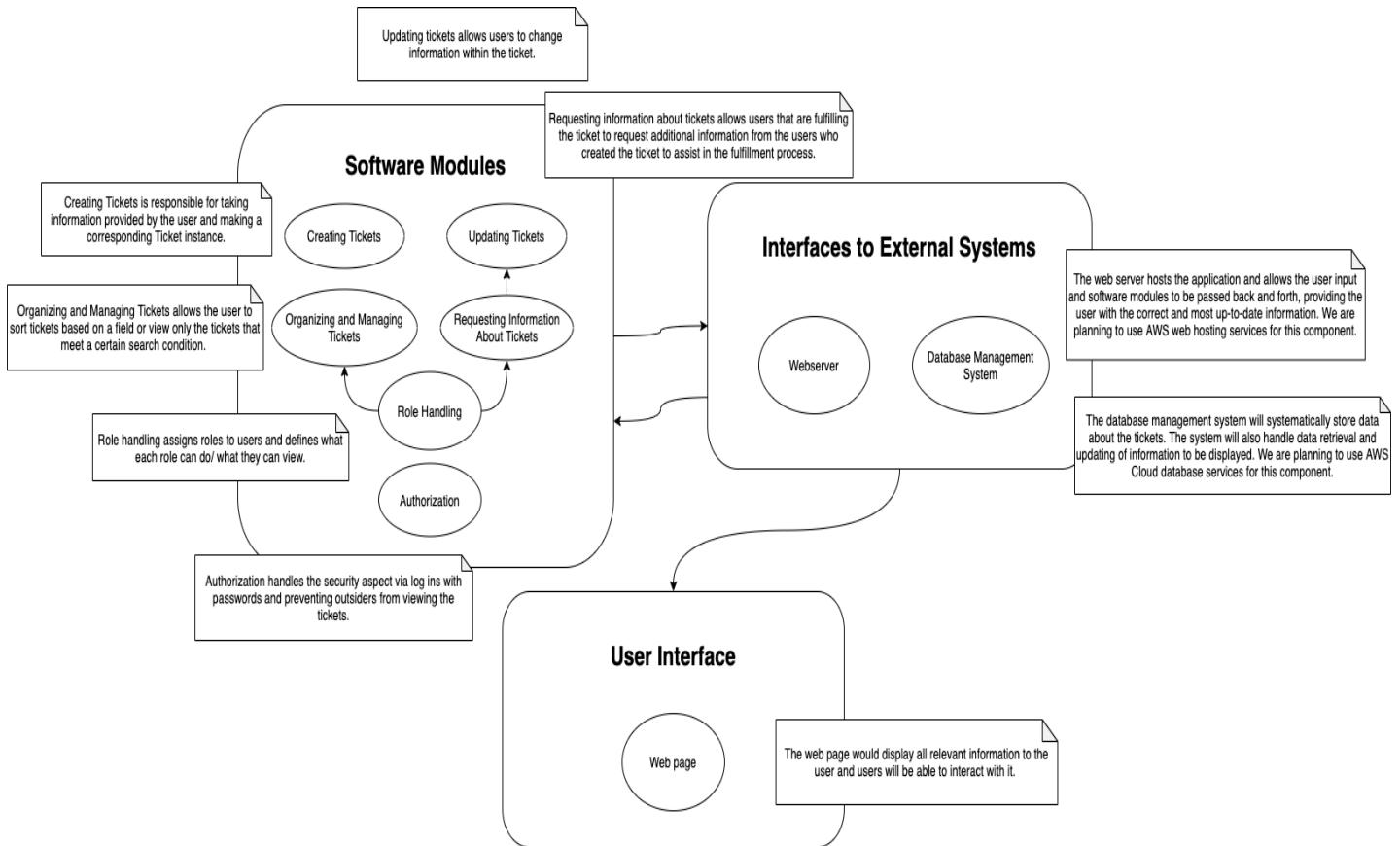
<https://github.com/comp195/senior-project-implementation-fix-it-ticket>

Arshita Sandhiparthi  
[a\\_sandhiparthi@u.pacific.edu](mailto:a_sandhiparthi@u.pacific.edu)

Gabriel Bick  
[g\\_bick@u.pacific.edu](mailto:g_bick@u.pacific.edu)

Last Revision: 09/12/2021

# System Architecture



## Hardware, Software, and System Requirements

As our software will be hosted in the cloud, the only requirement on the end user is a compatible browser. We will support the latest versions of Chrome, Firefox, Safari, and Microsoft Edge.

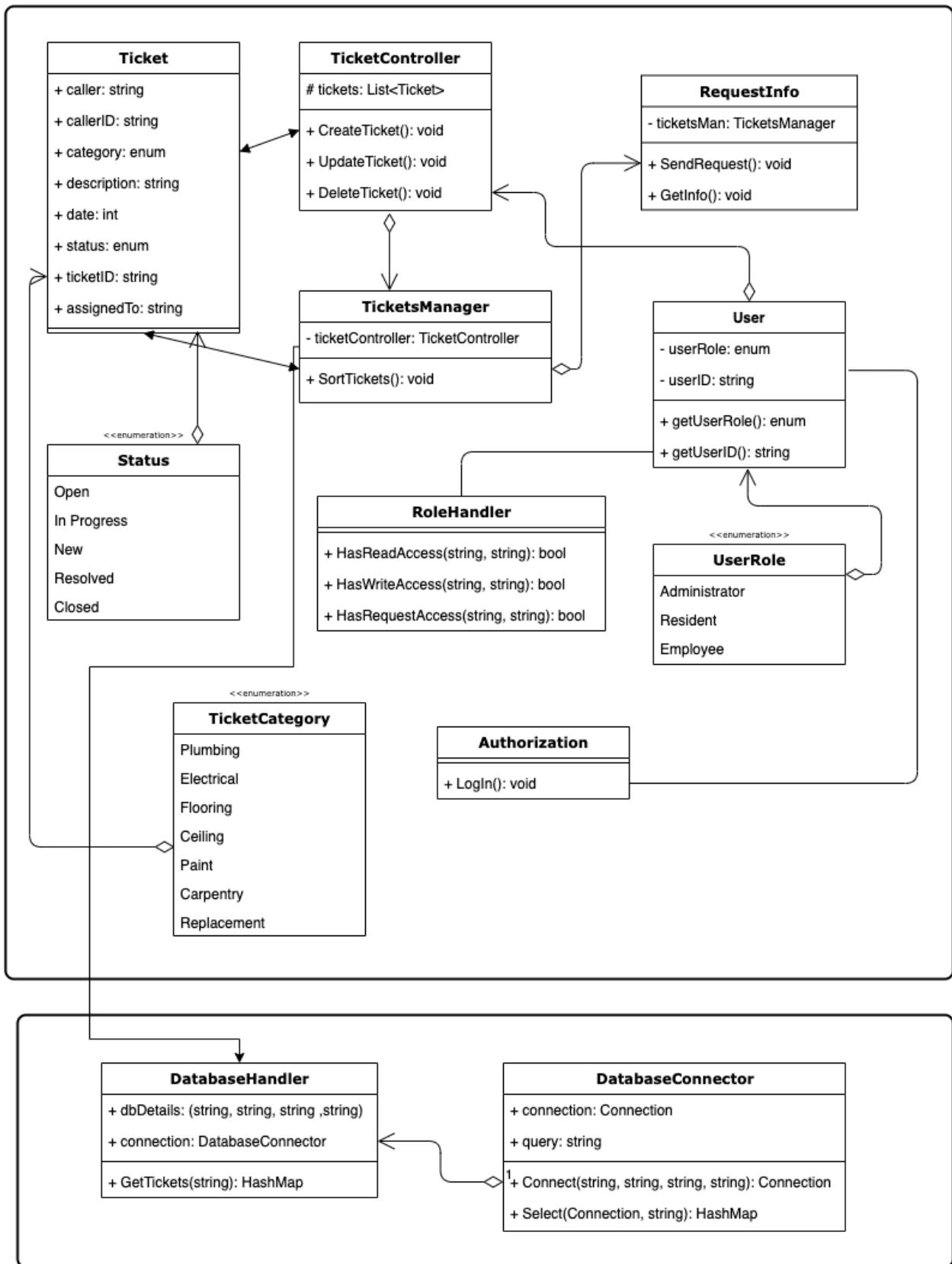
## External Interfaces

For our web server, we will be using Amazon EC2. The documentation is located at this website: <https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/concepts.html>.

For the database management system, we decided to use Amazon Relational Database Service. The documentation can be found at this website:

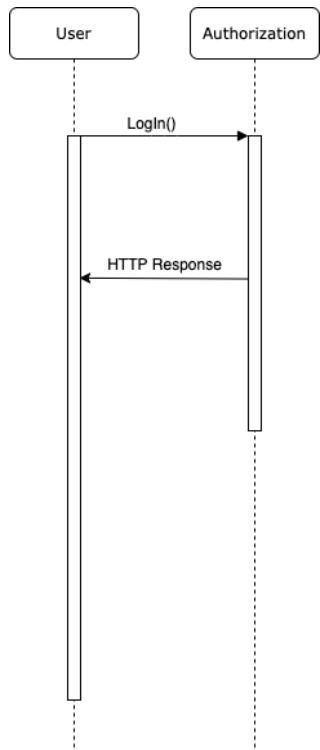
<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html>.

# Software Design

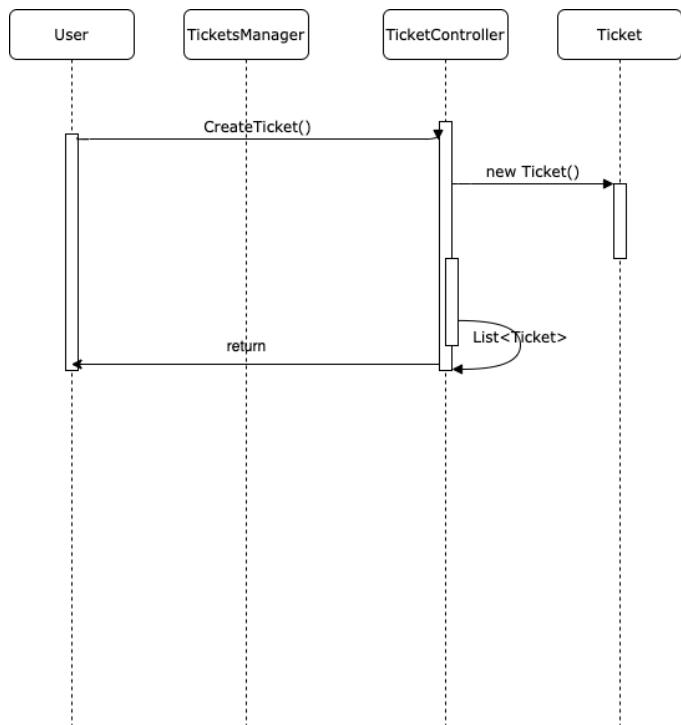


# Interaction Diagrams

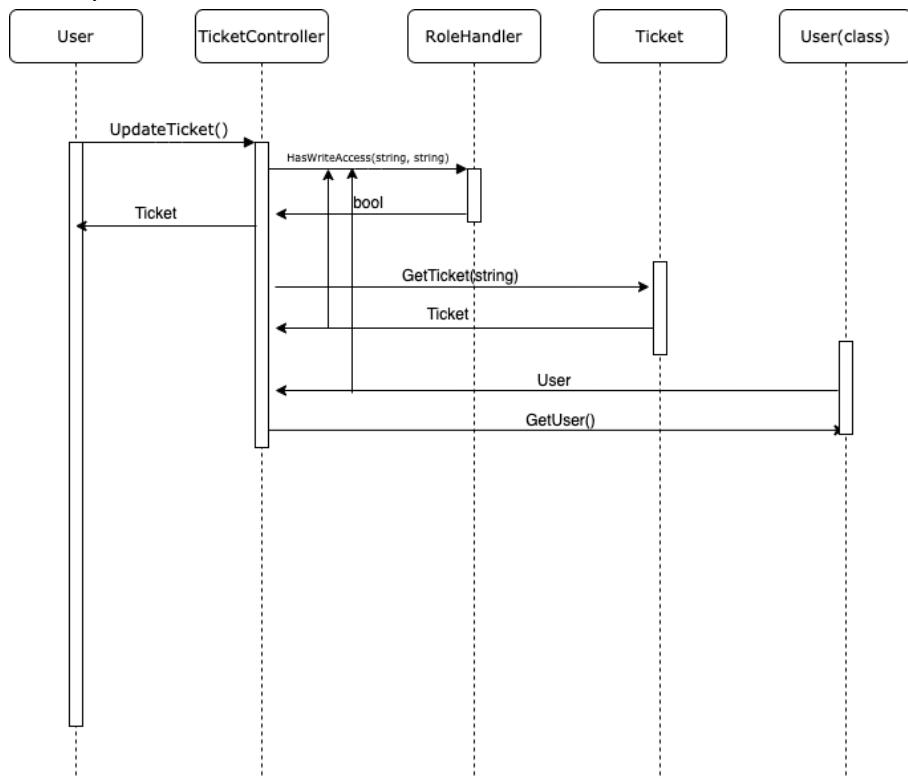
User logs in



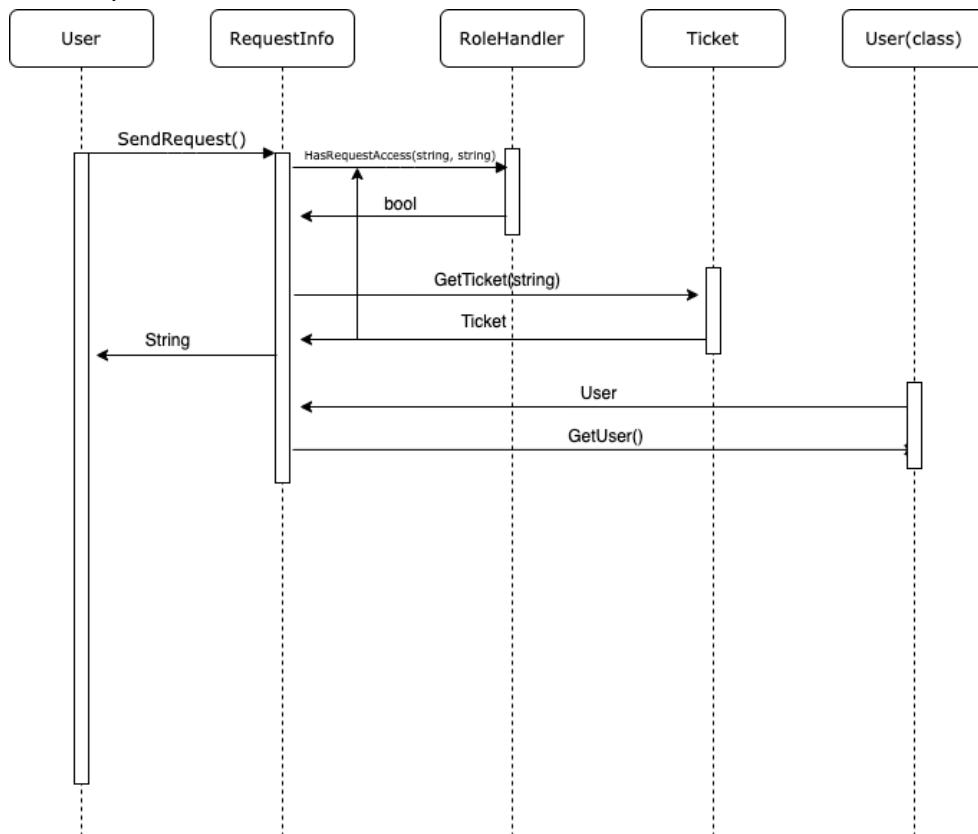
User creates ticket



### User updates ticket



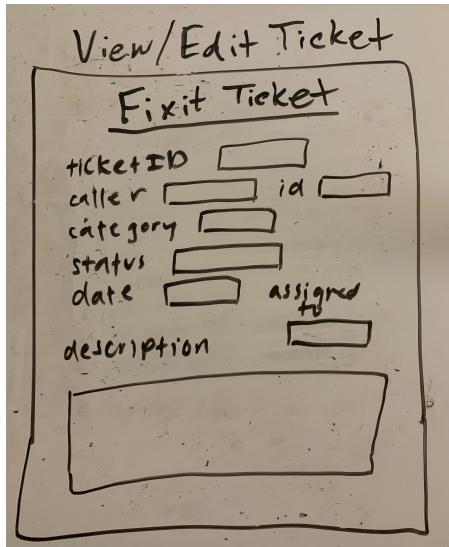
### User requests more information about ticket



## Design Considerations

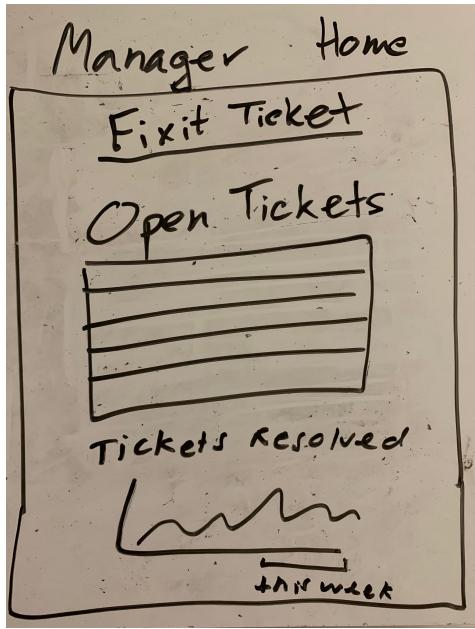
We utilized a singleton class, RoleHandler, because we knew that although we didn't need multiple objects checking user permissions during the user's usage of the program, we did need a class that was solely responsible for doing so.

## User Interface Design



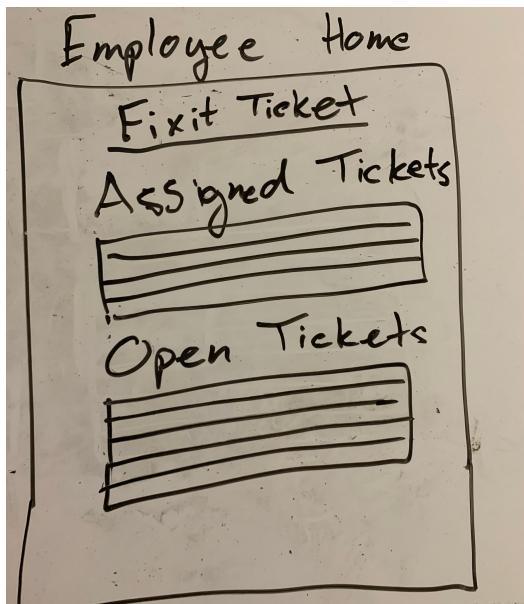
### Create/View/Edit Ticket

This screen will allow the user to view a ticket after having clicked on it from a list of tickets. Alternatively, the user can reach this screen with a blank ticket by clicking on the link to create their ticket in the resident home page listed below. Depending on their permissions, a user will be able to edit or delete an existing ticket.



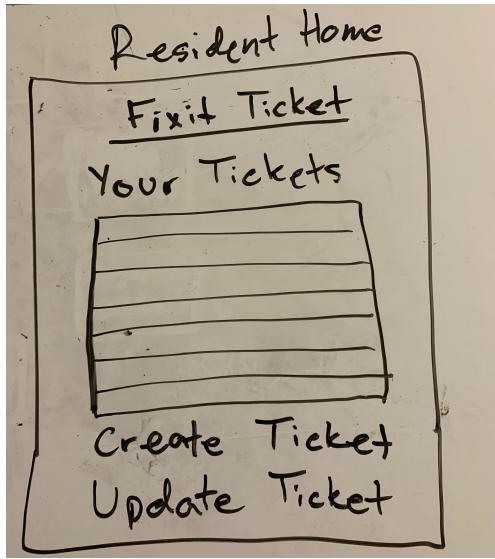
### Manager Home Screen

This is the home screen visible to a manager. This screen provides a view of existing open tickets, as well as statistics pertaining to tickets (a stretch goal). From here a manager can click on a ticket to reach the previous screen to edit or delete the ticket.



### Employee Home Screen

This is the home screen visible to employees, all users who are assigned tickets by managers. Employees can click on tickets leading to the view/edit ticket screen.



### Resident Home Screen

This is the home screen visible to residents, all users who create tickets for repairs. Residents can view their created tickets, and click on tickets to view them or add new details. Residents can also create new tickets by clicking on the 'Create Ticket' link. Residents cannot view or interact with other tickets.

## Glossary of terms

There are no terms we expect to be unfamiliar or ambiguous to the reader.

## References

Wikimedia Foundation. (2021, August 26). *Class diagram*. Wikipedia. Retrieved September 13, 2021, from [https://en.wikipedia.org/wiki/Class\\_diagram](https://en.wikipedia.org/wiki/Class_diagram).

Piter \_OS, & Noviff. (2017, January 12). *How to write a proper class to connect to database in Java*. Software Engineering Stack Exchange. Retrieved September 13, 2021, from <https://softwareengineering.stackexchange.com/questions/339598/how-to-write-a-proper-class-to-connect-to-database-in-java>.