# QuickTix

By: Matthew Kachensky, Louis Marfone, Diego Marrero Zilenziger, Liam McChesney, Daniel Medvedev, and Cheng Zhou

## Description:

QuickTix aims to provide CU students with a better experience when acquiring or selling local tickets in addition to connecting those with similar interests. Hosted on CU's server, users can expect access to the following features: register an account with the website and log in, view available tickets and mark tickets of interest, sell previously purchased tickets, view the profile pages of other users and their contact information, and the ability to add or remove personal reviews about other users. The website also affords each user the ability to filter results through the implementation of a search bar. Various API calls enable QuickTix to accurately pull relevant information necessary for the seamless functionality of its features and pages. Inline javascript was used to implement test cases as well as generate the necessary HTML for functional conditionals. NodeJS served as the primary framework and was used to: assign values to buttons, modify the display conditions for various elements, and include partials, which played a key role when implementing cards.ejs, a partial which constituted a core functionality of the website.

## Tracker:

https://github.com/users/chengzhou-31/projects/1/views/1

## Video:

https://youtu.be/5hlTFxqU2no

## VCS:

https://github.com/chengzhou-31/lab10-group-project-014-01

# Contributions:

**Matthew Kachensky**:
I worked on creating the database using a relational diagram to ensure that the database was in third normal form. Some of my work was done in the index.js file where all of the api calls are made. The api's I worked on were: the one for the home page, tickets being put into and removed from the database both for putting new tickets and adding tickets the user are interested in, the profile page, and reviews for users. Outside of the api calls. I worked on the add reviews partial, the add new listing partial, profile page and various changes to other pages including implementing the api calls, quality of life improvements, some stylization, and the creation of the use-case diagram.

**Louis Marfone**:
I contributed a lot to the styling of the pages and worked mostly with html and embedded js. I created and implemented the cards that we use on almost every page to display information about each ticket. It was made so that we could implement it for any field in which tickets were displayed and simply pass the information form the api call to specialize the card to that field. I also figured out how to convert the datetime type passed from the database for ticket dates into a string that could be edited because when outputting a date directly from the database response to a card  a lot of extraneous details were printed making the cards look less professional. Finally I helped with the initial versions of the search page that cheng eventually finished.

**Diego Marrero Zilenziger**:
Coded initial versions of the login API and page. Tested the usability of both with dummy data to ensure a user could successfully login if an account existed and implemented an option which would redirect the user to the registration page if an account did not exist. Worked on a preliminary version of the profile page. Provided minor assistance with troubleshooting and development of the presentation. Fulfilled the role of scribe for some team meetings.

**Liam McChesney**:
I worked on a few different parts of the project. I put together the partials for the header, footer, and navbar. I also assisted in starting the partial for the cards which was used to display tickets throughout the website. I built the home page as well. One thing I particularly enjoyed doing was styling the website. While this didn't take long it was cool to choose a color scheme, fonts, and design a logo. The last thing that I contributed was adding the front end for removing a ticket from a users list and tweaking the back end to finalize the function.

**Daniel Medvedev**:I coded the early stages of the registration page. I coded a basic api and html to run with it. I also created some styling for that page. You could register a user with a username, email and password. It would then direct you to the login page. I worked as a scribe for most of the TA meetings and wrote all of the version notes as well which include a detailed report of what mostly happened.

**Cheng Zhou**:
I worked on implementing the database for the users and tickets, which includes making the sql tables. I also helped redesign the login and registration page, as well as implementing hashing for the login and register API. I assisted in creating the initial layout of the profile page, along with the deleting tickets feature. I implemented the edit profile feature, as well validation for it, along with validation for registration. I also designed and implemented the search feature of our website, which includes the backend and frontend, as well. I also created the filter feature for search, which allows users to search by category, including price range. I also helped with the initial file structure of our project (including the docker-compose and index.js), and helped people troubleshoot GitHub issues. Aside from that, I helped out with styling and planning of our website.

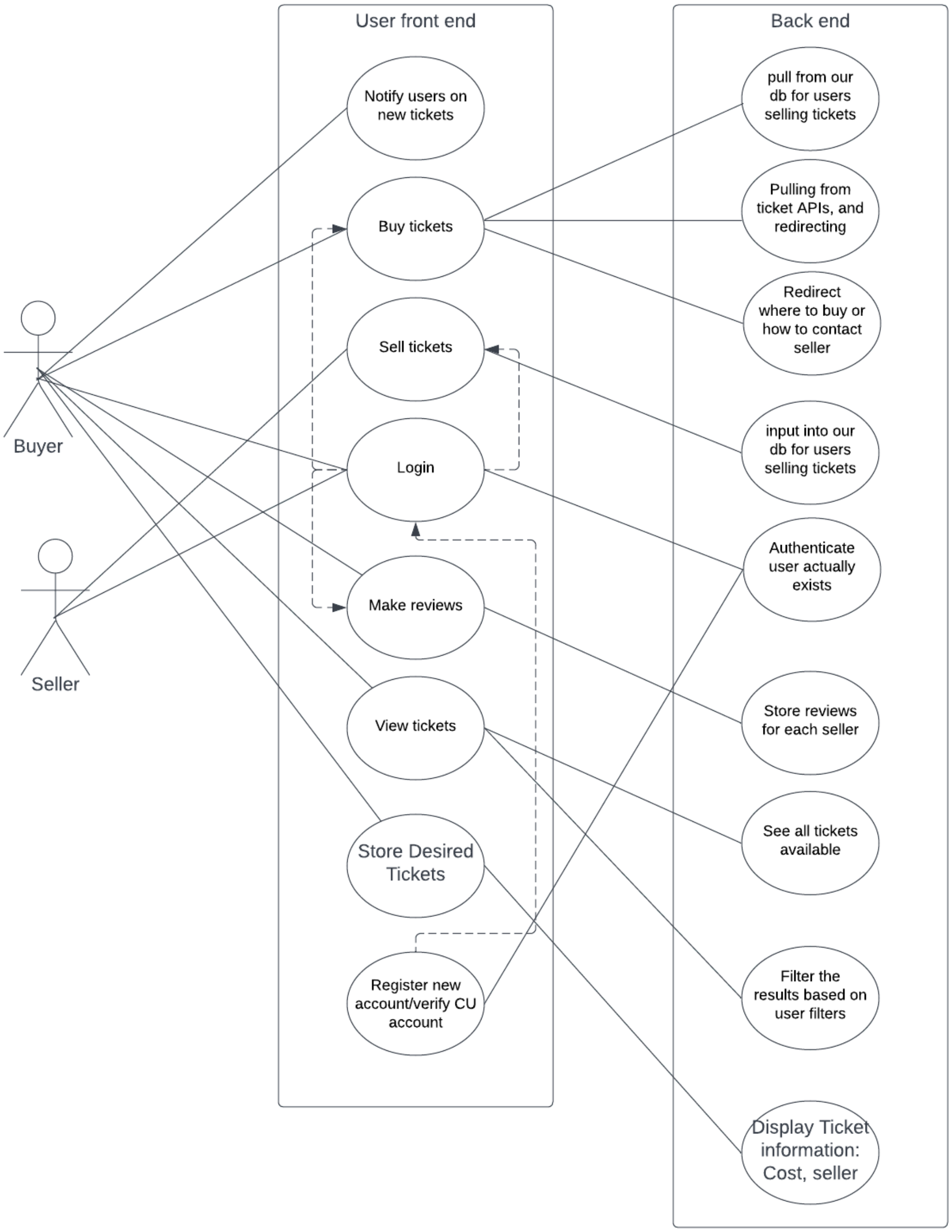
## Test results:

Based on our test cases from lab 11. Using a few people outside of the class and our project we found that when using our website and trying to push its limits that: the email fulfilled their expectations, password requirements were sufficient including confirmation, and that the username was a bit restrictive but works nonetheless.

## Deployment:

We were unable to deploy our website. Upon trying to launch our project it would stay active for a few minutes then subsequently shut down unexpectedly. This occurred multiple times, after which we learned that the CU hosting was having problems out of our control.

# Use case Diagram:


User front end
Back end
Buyer
Seller
Notify users on new tickets
Buy tickets
Sell tickets
Login
Make reviews
View tickets
Store Desired Tickets
Register new account/verify CU account
pull from our db for users selling tickets
Pulling from ticket APIs, and redirecting
Redirect where to buy or how to contact seller
input into our db for users selling tickets
Authenticate user actually exists
Store reviews for each seller
See all tickets available
Filter the results based on user filters
Display Ticket information: Cost, seller