CO550
Web Applications
Coursework 3

Build a Simple Web Database Application

Alex Paulo da Costa

21611431

James Harris

21606555

19th January 2018

Computing & Web Development
Buckinghamshire New University

Summary

Background & Rationale

A Enterprise Model

A clerk in the head office (or operational manager) sets the routes and times they want for the bus lines.

Drivers are assigned to the routes, reducing the total time of the route to their contracted hours. The closer it is to zero (hours left to allocate) the better.

Customers can look at the timetable for the line, without knowing who the driver is, and choose the bus they want to take.

The customers board the bus, pay the driver, or show them a previously bought and still valid ticket, and take a seat.

B Functional Requirements

- Enable users to login and register
  - Normal (unprivileged users) can register anytime
  - Privileged users must be added by an admin
- Enable users and visitors to see bus timetables
- Enable registered users to save their favourite routes
- Allow admins and authorised staff to change timetables and rotas
  - Create bus lines and routes
  - Allocate staff to lines
- Allow staff to see their rota
  - Staff assigned to particular routes
  - Staff dashboard allows to print weekly rota as a pdf
- The users should be able to search routes by terms, e.g route code, town names or bus stop names

C Outline of the Project

1. Review functional requirements against the business goals
2. Develop a user management system for customers, drivers and managers
    (a) Implement a user login system
    (b) Implement registration for customers
    (c) Implement a timetable for drivers (dashboard)
    (d) Restrict data modification to staff with the required role (e.g. managers)
3. Develop a timetabling service for buses
    (a) Implement bus lines
    (b) Implement bus routes for each line
    (c) Implement bus stops and allow associating routes with these stops
    (d) Build a front-end for route information
    (e) Allow customers to favourite the routes and lines
4. Check the implemented system satisfies the functional requirements outlined by the business goals
    (a) Make any required changes to meet the requirements

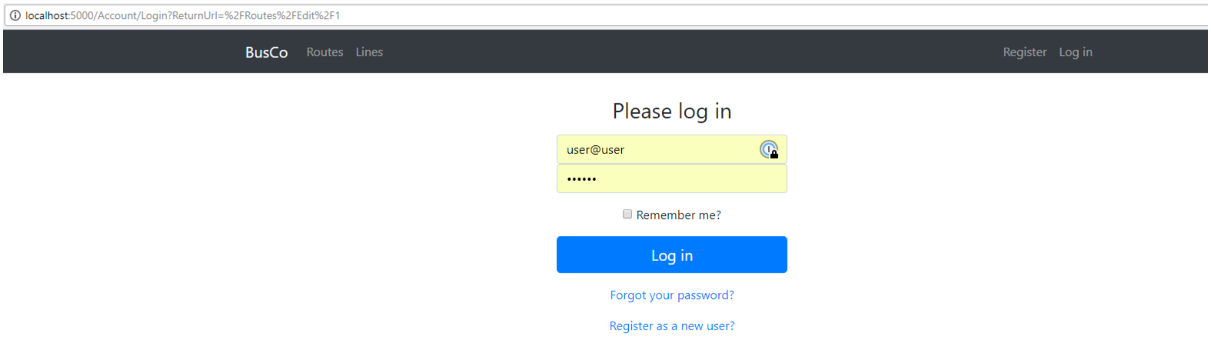
Application Design

Updated ERD and UML needed.

Technical Report

For this project we used ASP.NET Core 2 and an SQLite3 database. This is interacted with the Entity Framework. We also opted for a code-first approach as it was the most intuitive to add user authentication to.

For our user authentication system we used ASP.NET Core Identity. With this system we were able to create a unified user database and add roles to the registered users. These roles were then used to restrict access to specific methods - for example guests can only get the route index and details page, while managers and admins can create, edit and delete routes. If guests try to access these pages they will get redirected to the login page. Normal users will receive an $Unauthorised$ error.



localhost:5000/Account/Login?ReturnUrl=%2FRoutes%2FEdit%2F1

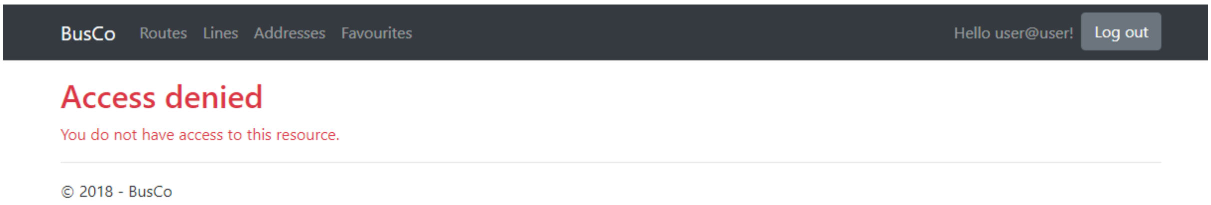
Please log in

user@user

••••••

□ Remember me?

Figure 3.1: Guest accessing $/Routes/Edit/1$, redirected to login page

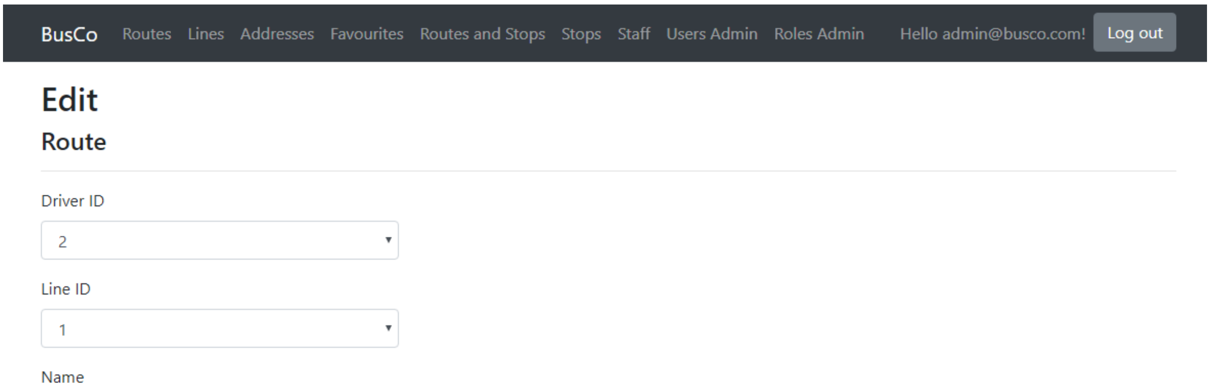


Access denied

You do not have access to this resource.

© 2018 - BusCo

Figure 3.2: User accessing $/Routes/Edit/1$ showing unauthorised error



Edit

Route

Driver ID

2

Line ID

1

Name

Figure 3.3: Manager accessing $/Routes/Edit/1$ and getting the edit page displayed

Evidence of Testing

Appendix