23 MAY, 2020

Pierce Kinghorn

12091381

Traffic project software documentation

Table of Contents

[Traffic project test cases 2](#_Toc41125741)

[Adding an accident entry 2](#_Toc41125742)

[Cycling through accident entries 3](#_Toc41125743)

[Updating an accident entry 3](#_Toc41125744)

[Filtering accident entries 4](#_Toc41125745)

[Adding a vehicle entry 4](#_Toc41125746)

[Cycling through vehicle entries 5](#_Toc41125747)

[Updating a vehicle entry 5](#_Toc41125748)

[Filtering vehicle entries 6](#_Toc41125749)

[Creating a vehicle accident entry 6](#_Toc41125750)

[Displaying vehicle accident entries 7](#_Toc41125751)

[Filtering vehicle accident entries 7](#_Toc41125752)

[Version Control Discussion 8](#_Toc41125753)

[Introduction 8](#_Toc41125754)

[Current Technologies 8](#_Toc41125755)

[Version Control Models 9](#_Toc41125756)

[Local data model 9](#_Toc41125757)

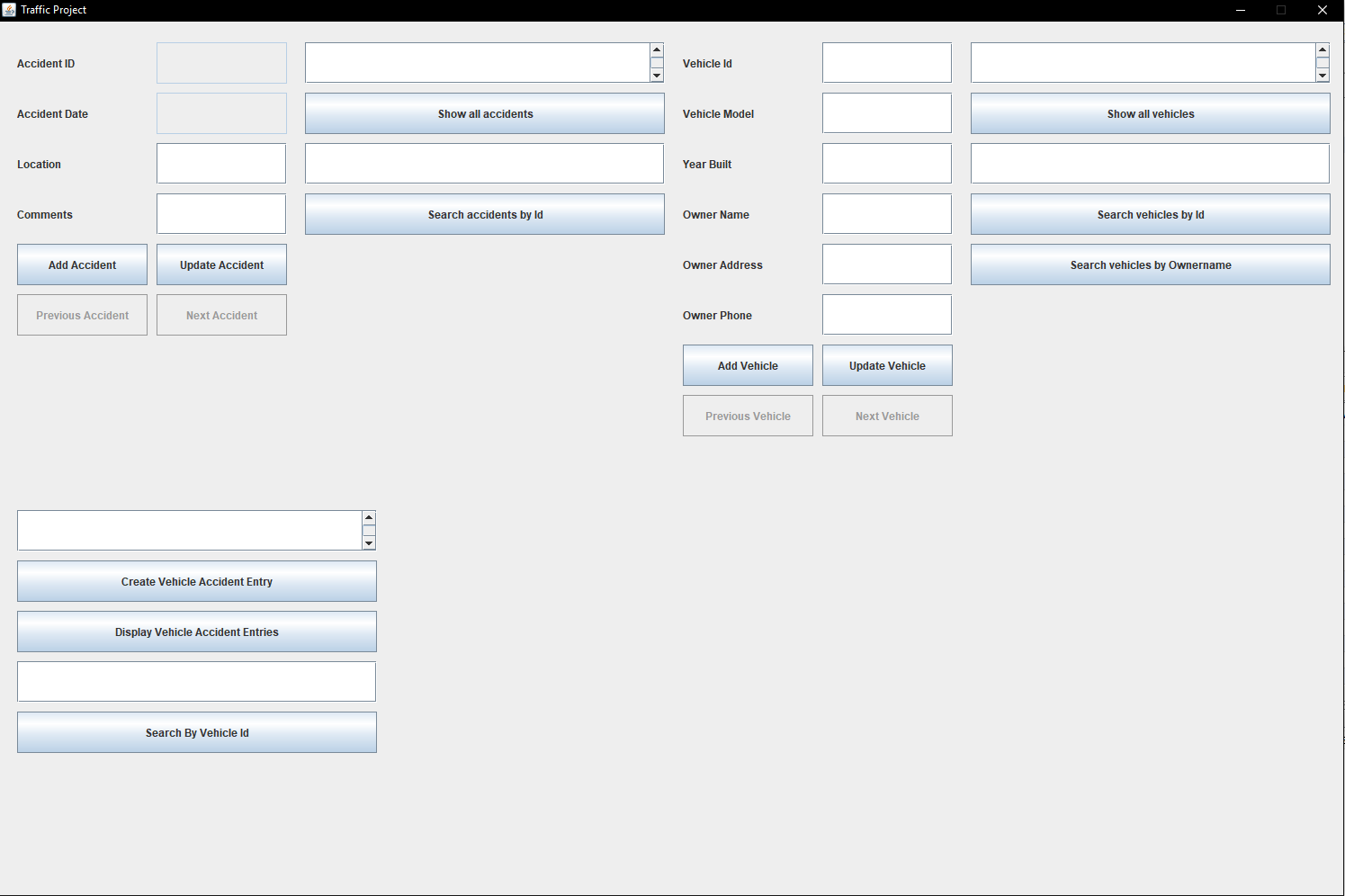
[Client-server model 9](#_Toc41125758)

[Distributed model 9](#_Toc41125759)

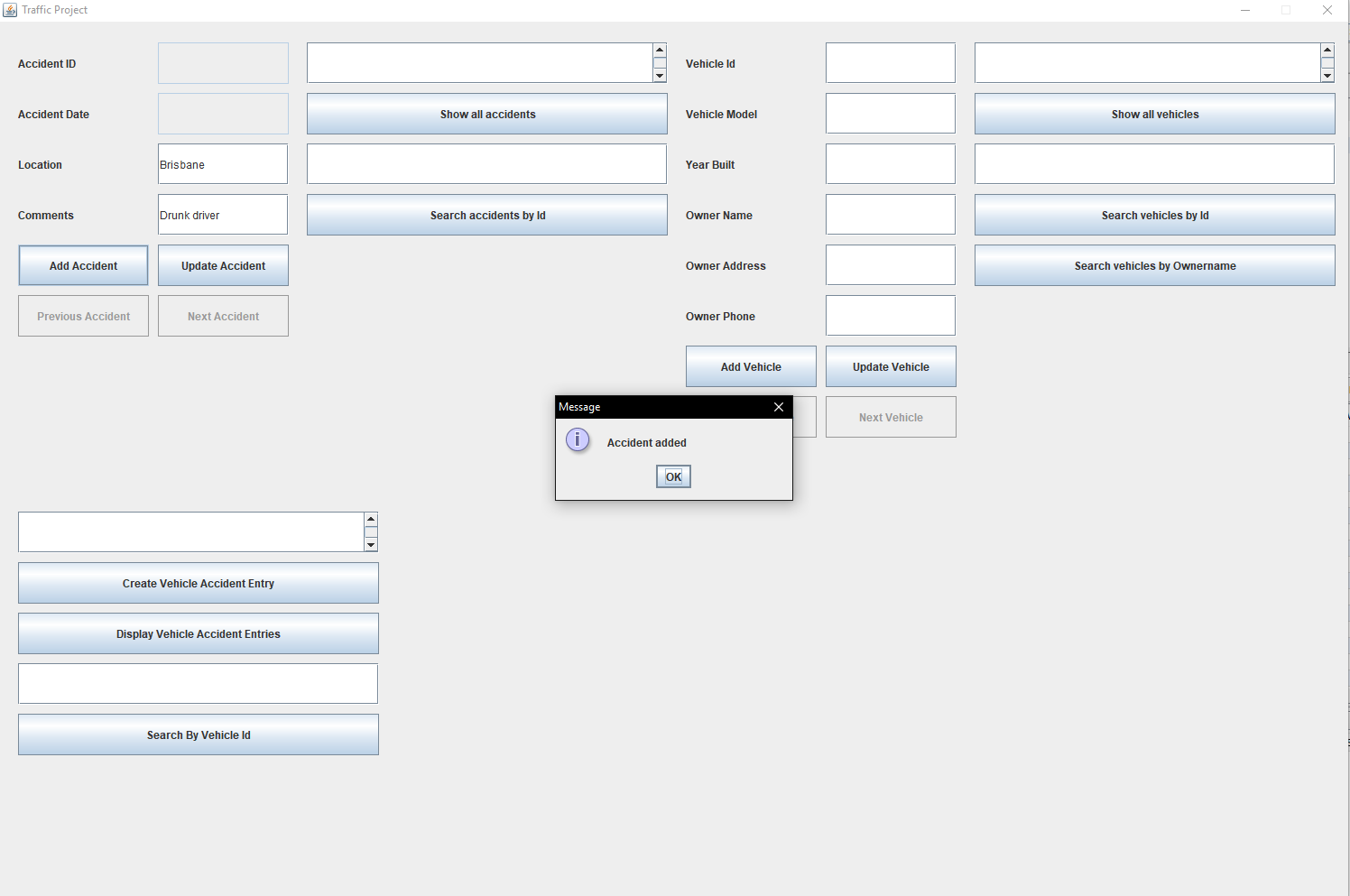
[Bibliography 10](#_Toc41125760)

# Traffic project test cases

## Adding an accident entry

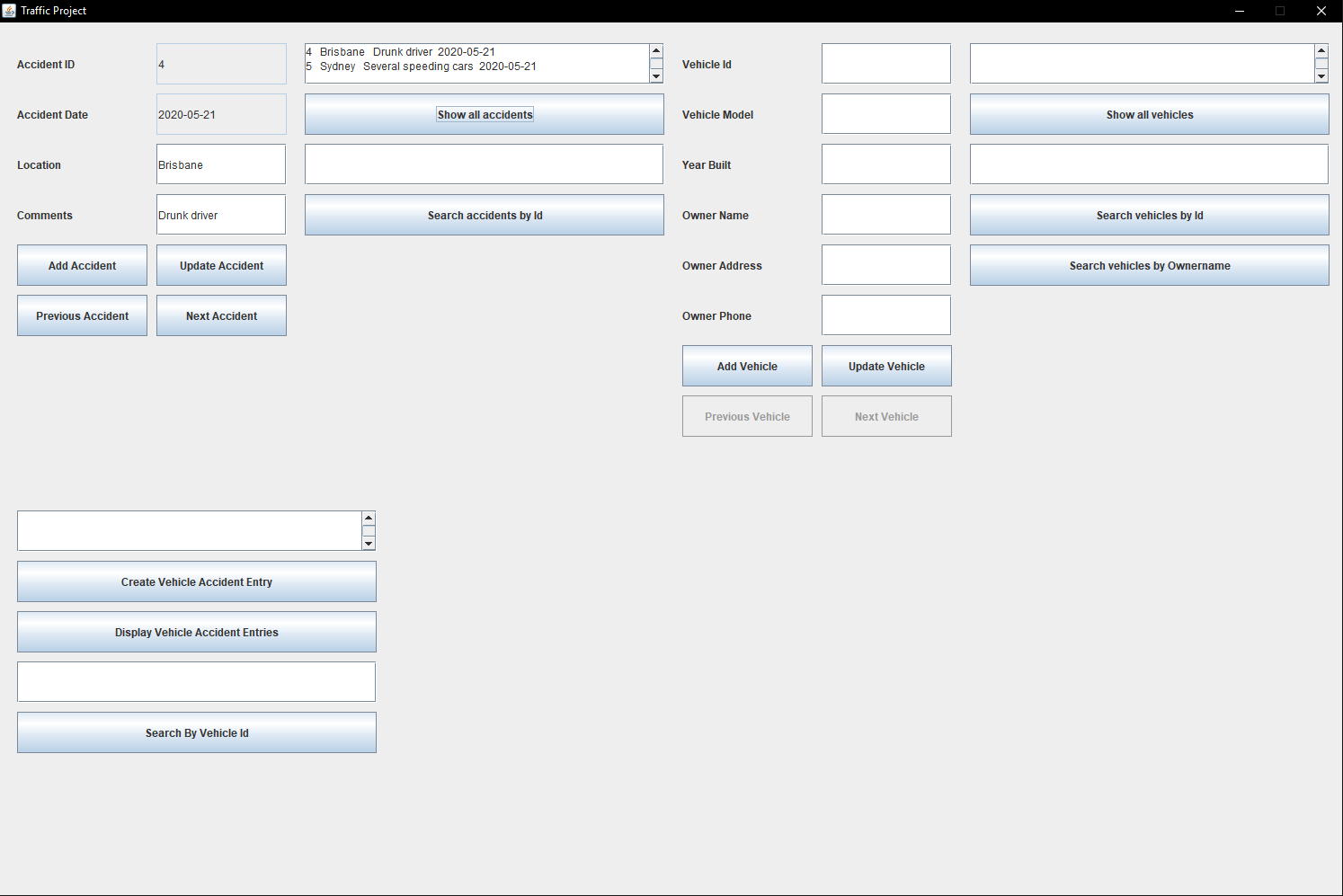


When the use first launches the software, this is the first screen shown to the users



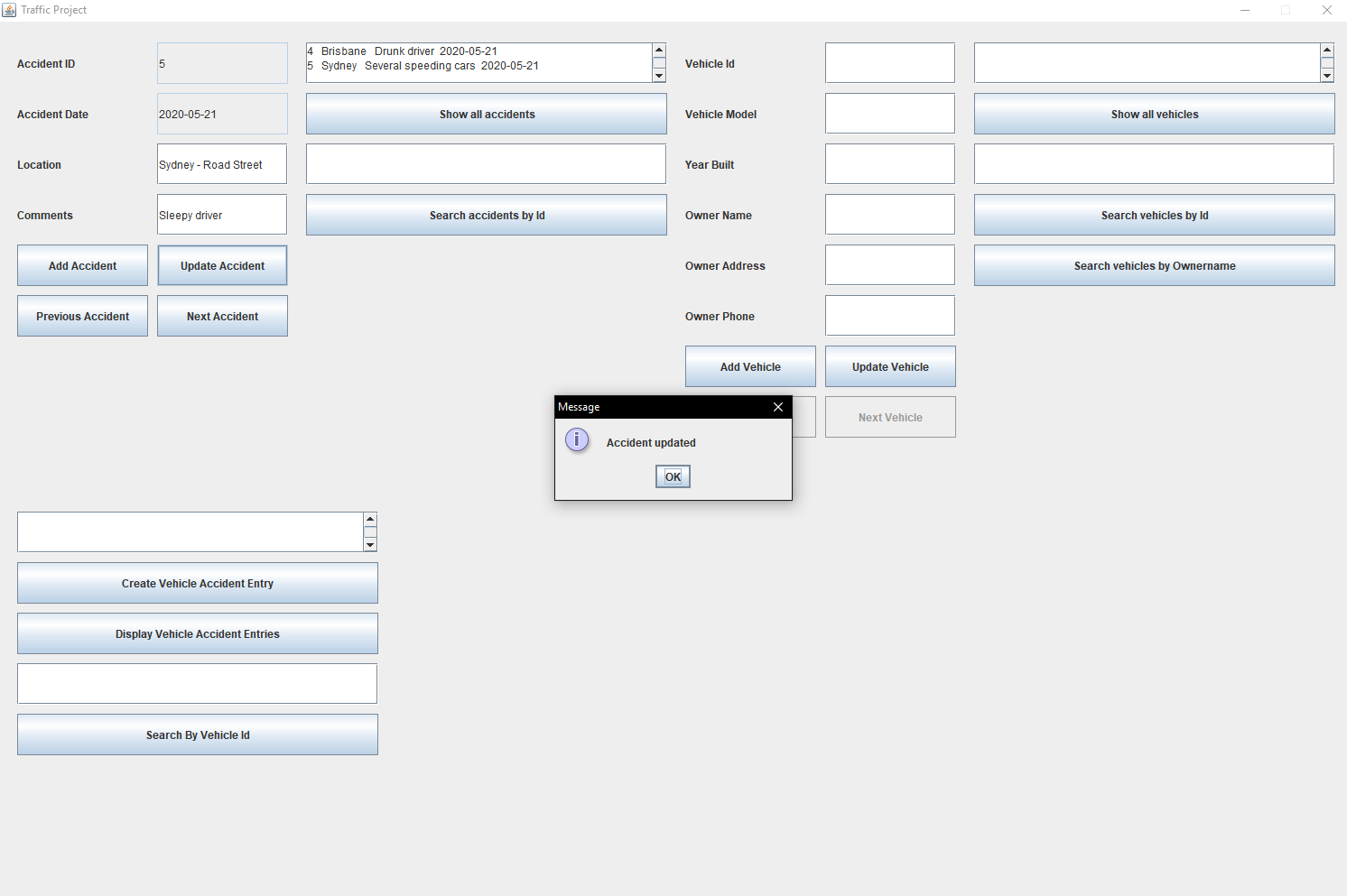
To create an accident entry, fill in the location and comments text fields and click the Add Accident button to add create an accident record. If done successfully a message will appear stating Accident added.

## Cycling through accident entries



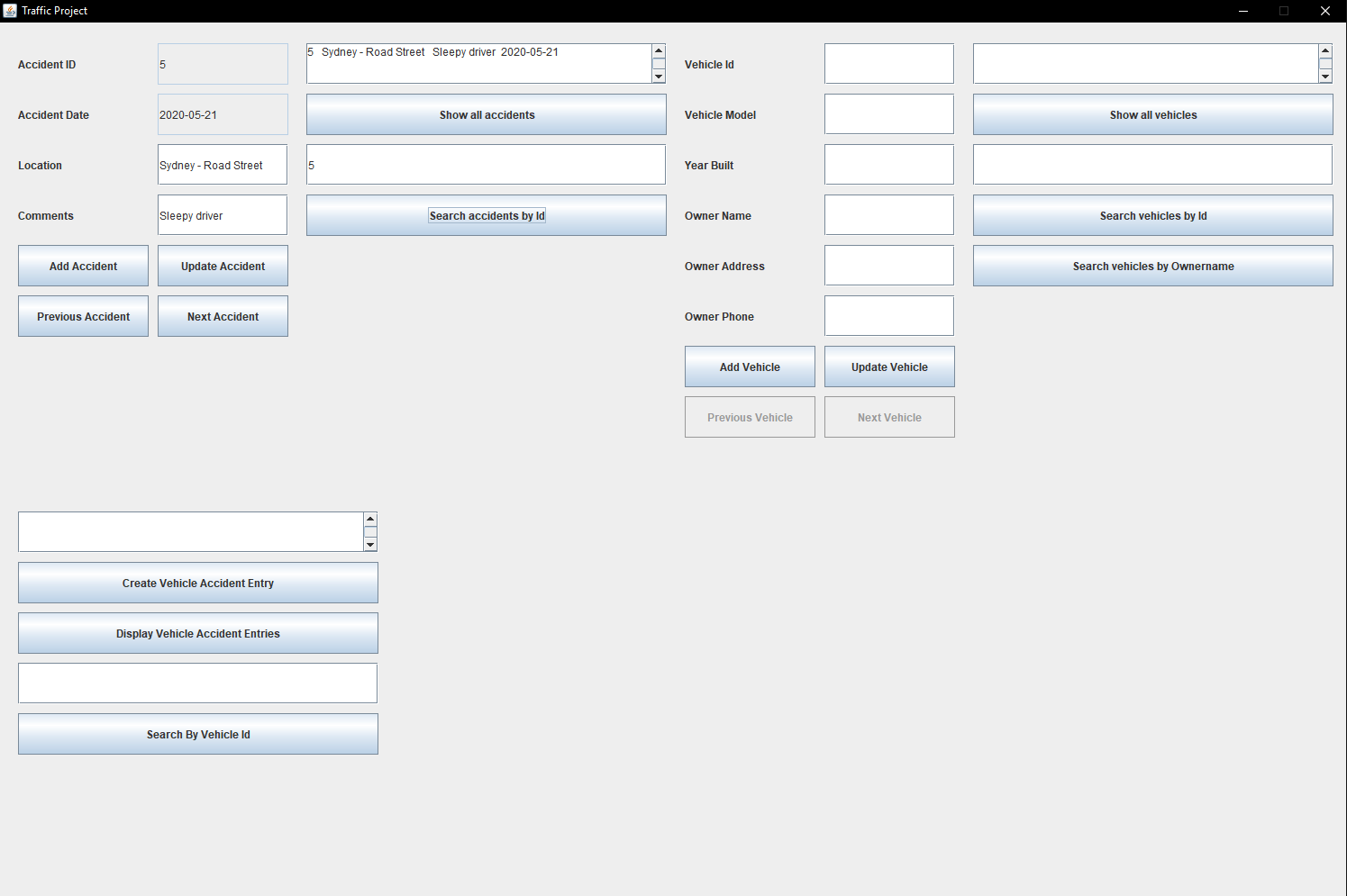
To cycle through accident entries there must be at least two accident records. The user must first select the Show all accidents button. This will display all the accident records that can be cycled through. The user can then navigate by selecting either next Previous Accident button, or the Next Accident button to cycle through the listed entries.

## Updating an accident entry



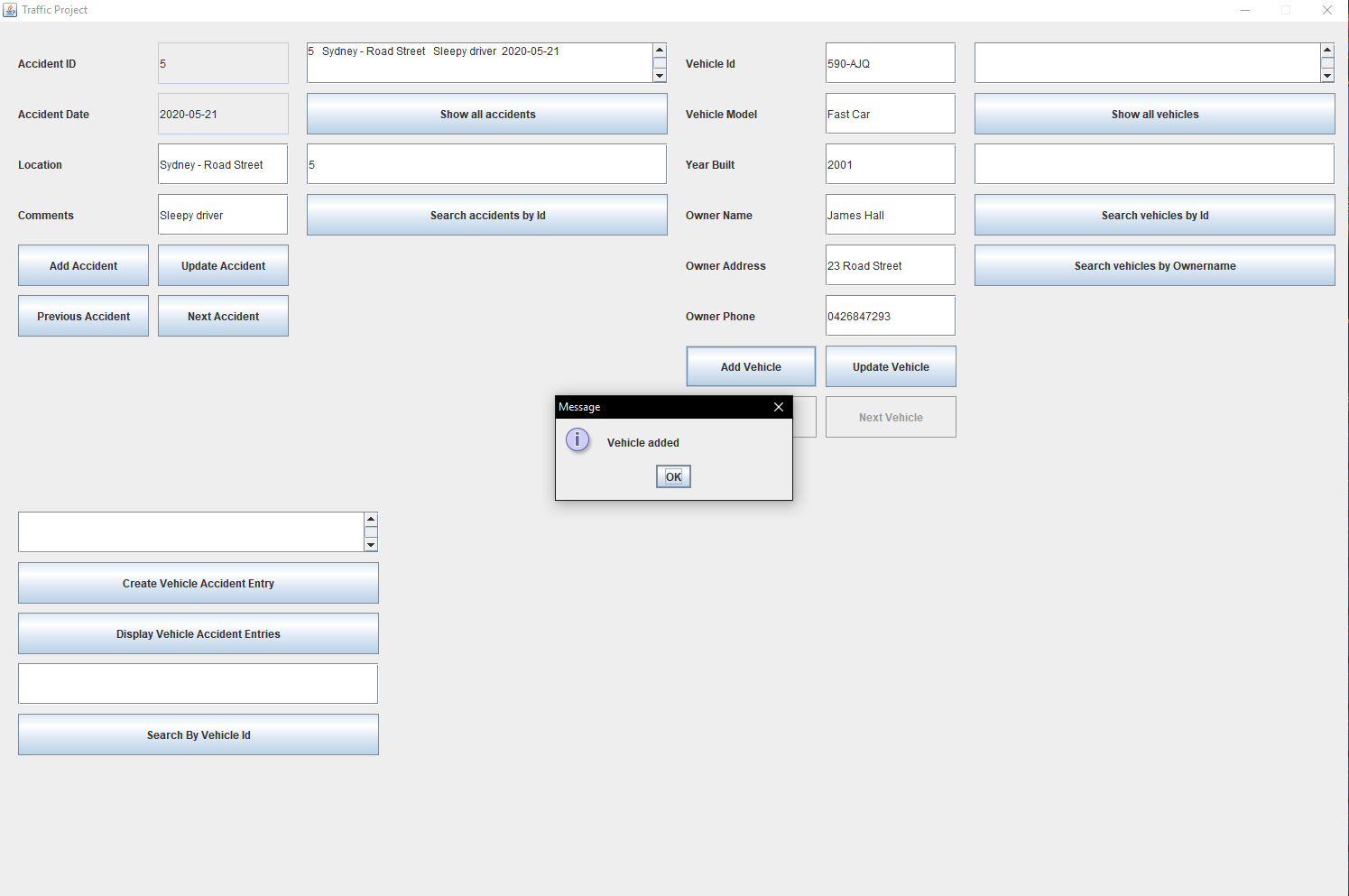
To update an accident entry the user must first cycle to the desired accident the user can then change the desired fields and click update.

## Filtering accident entries



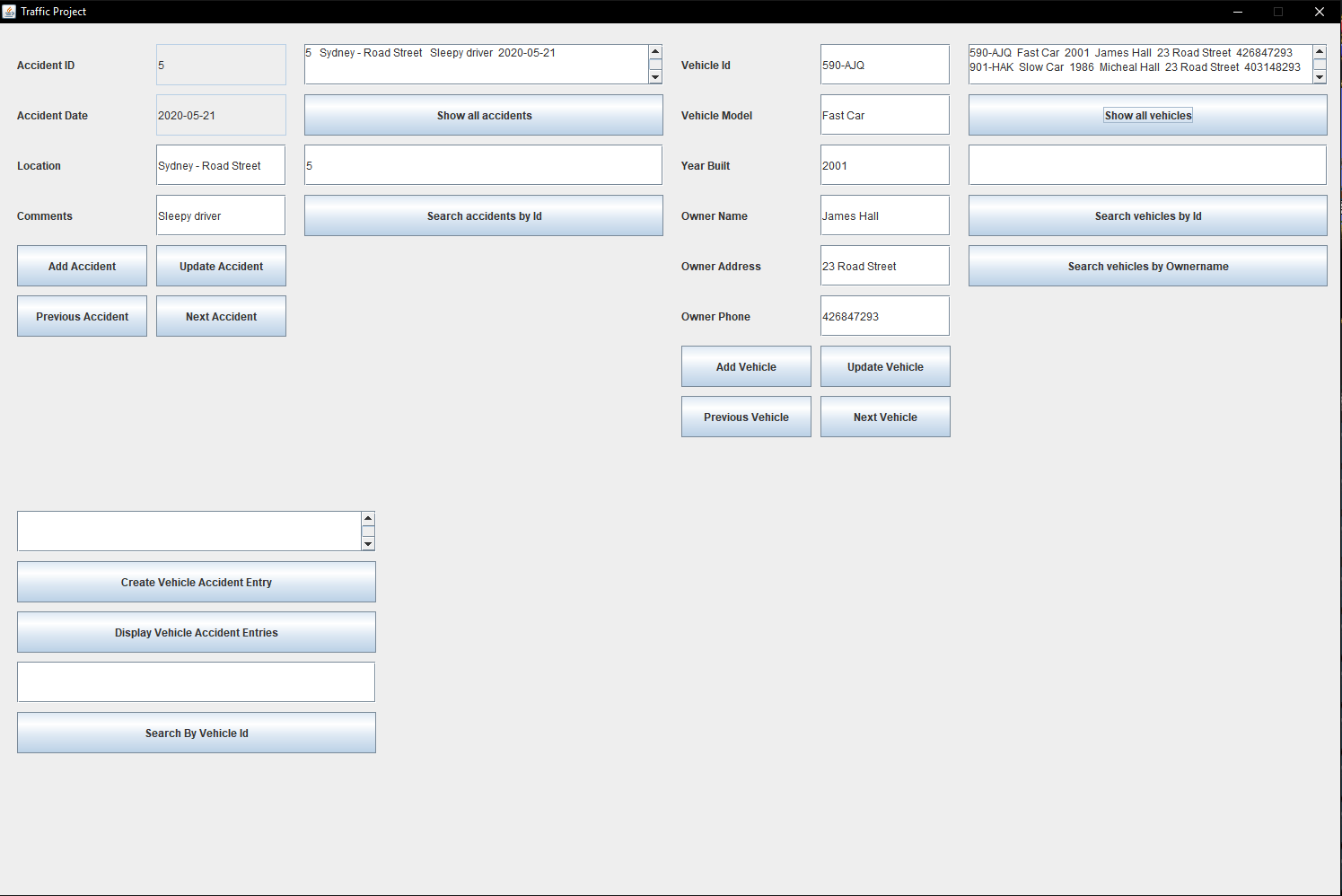
Users can filter the accident entries by typing the id into the search field above the Search by accident id button. Clicking this button will then display the filtered accidents.

## Adding a vehicle entry



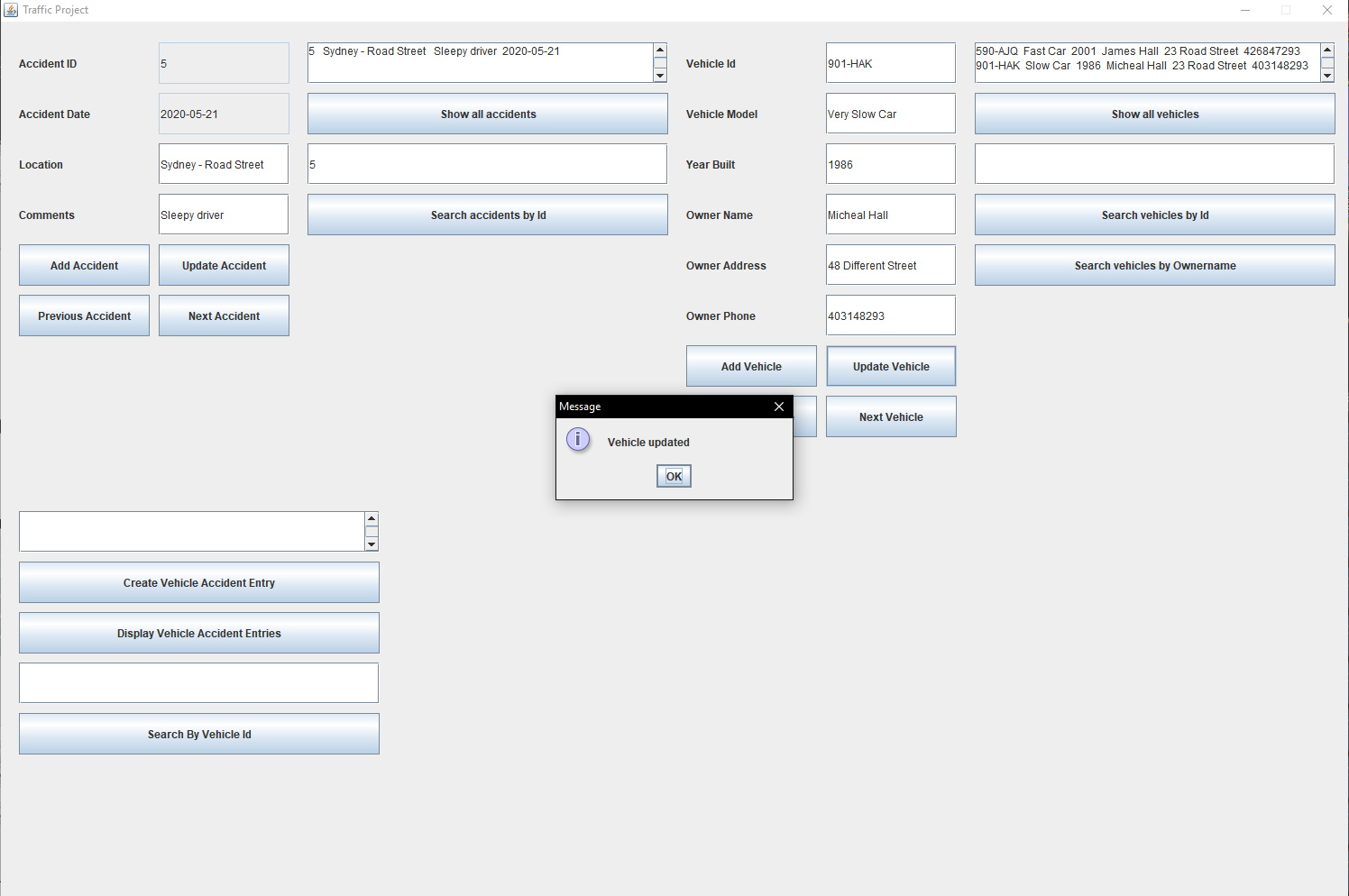
To create a vehicle entry, fill in the text fields and click the Add vehicle button to add create a vehicle record. If done successfully a message will appear stating vehicle added.

## Cycling through vehicle entries



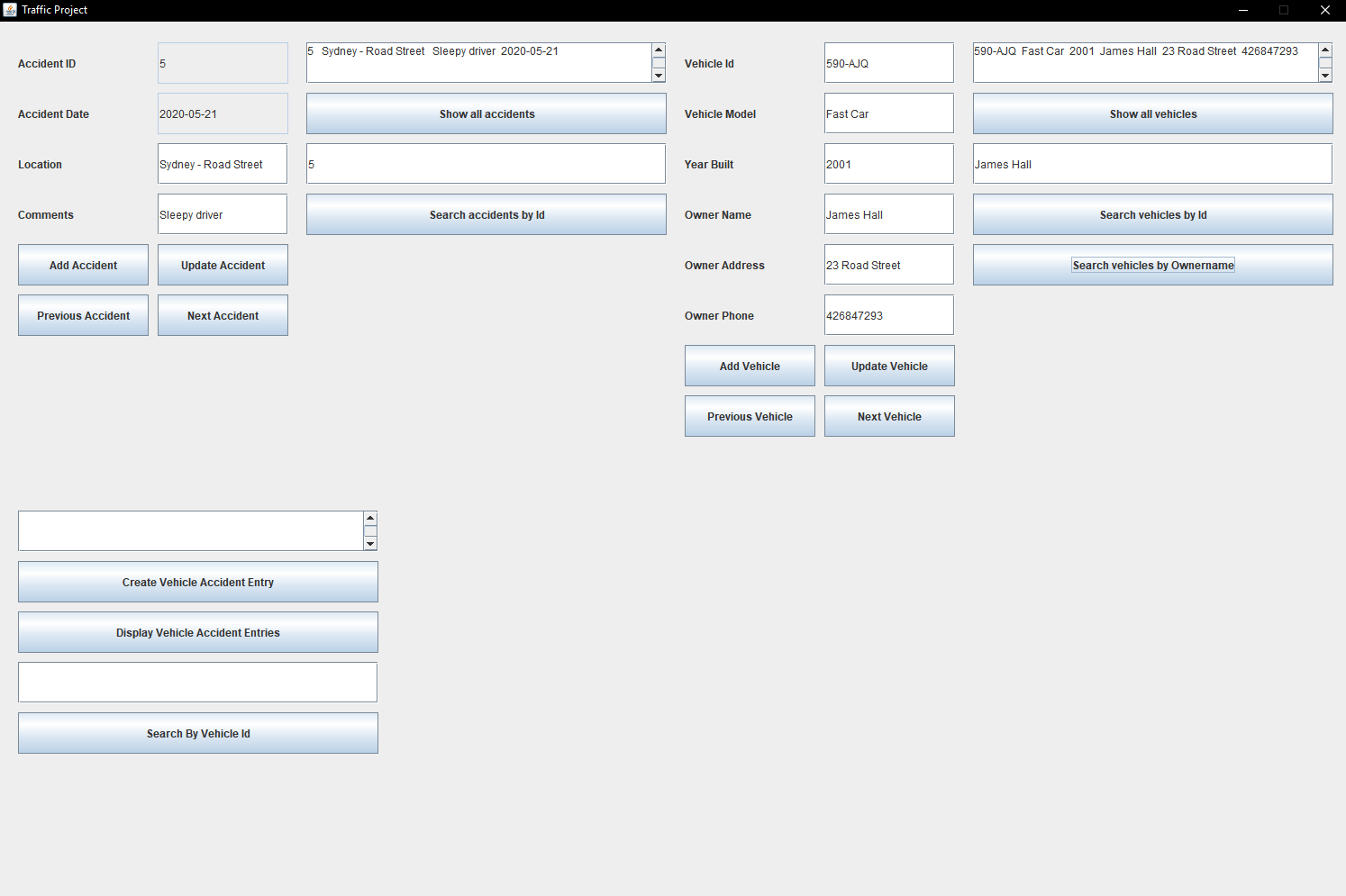
To cycle through vehicle entries there must be at least two vehicle records. The user must first select the ‘Show all vehicle’ button. This will display all the vehicle records that can be cycled through. The user can then navigate by selecting either next Previous vehicle button, or the Next vehicle button to cycle through the listed entries.

## Updating a vehicle entry



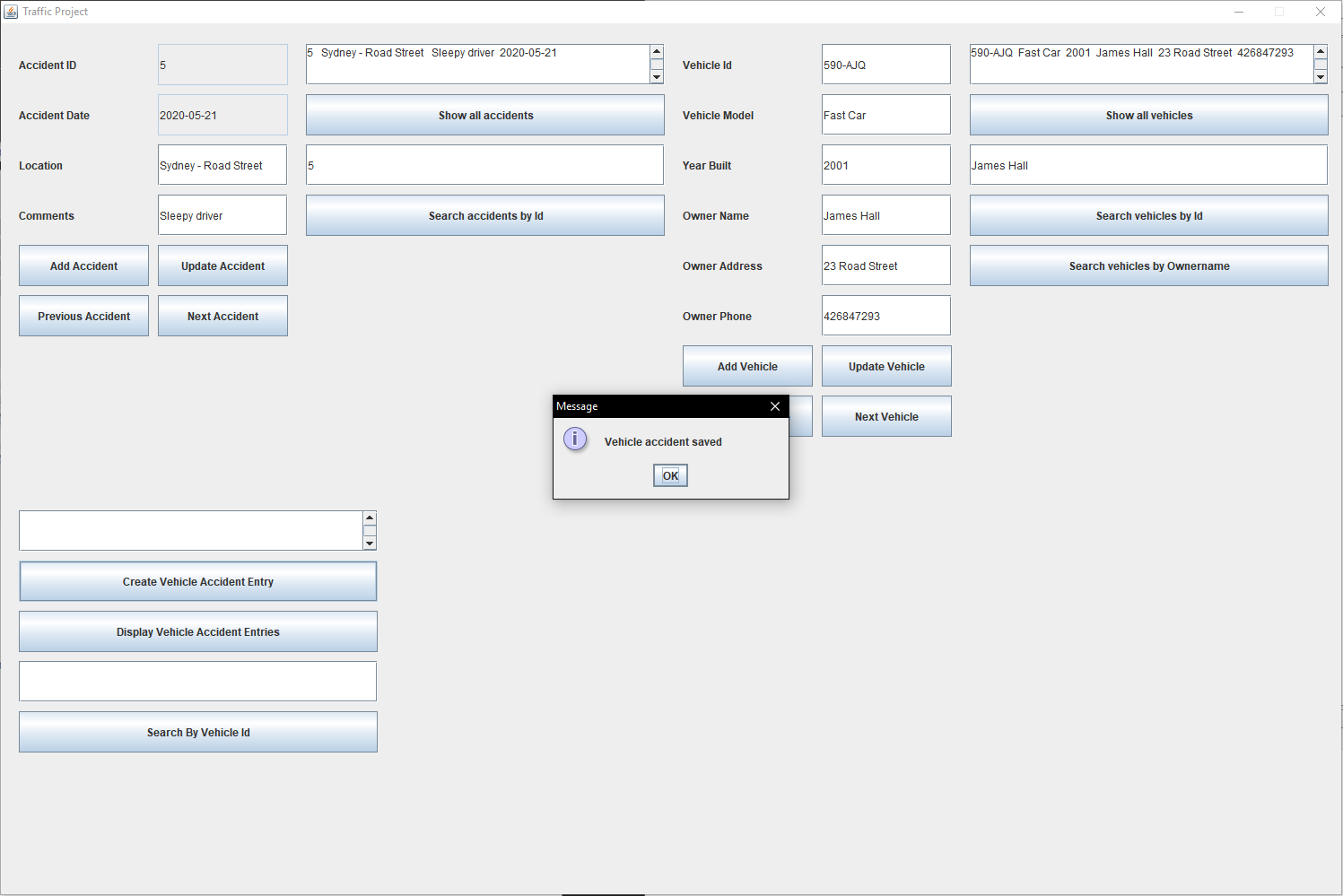
To update a vehicle entry the user must first cycle to the desired vehicle. The user can then change the desired fields and click update.

## Filtering vehicle entries



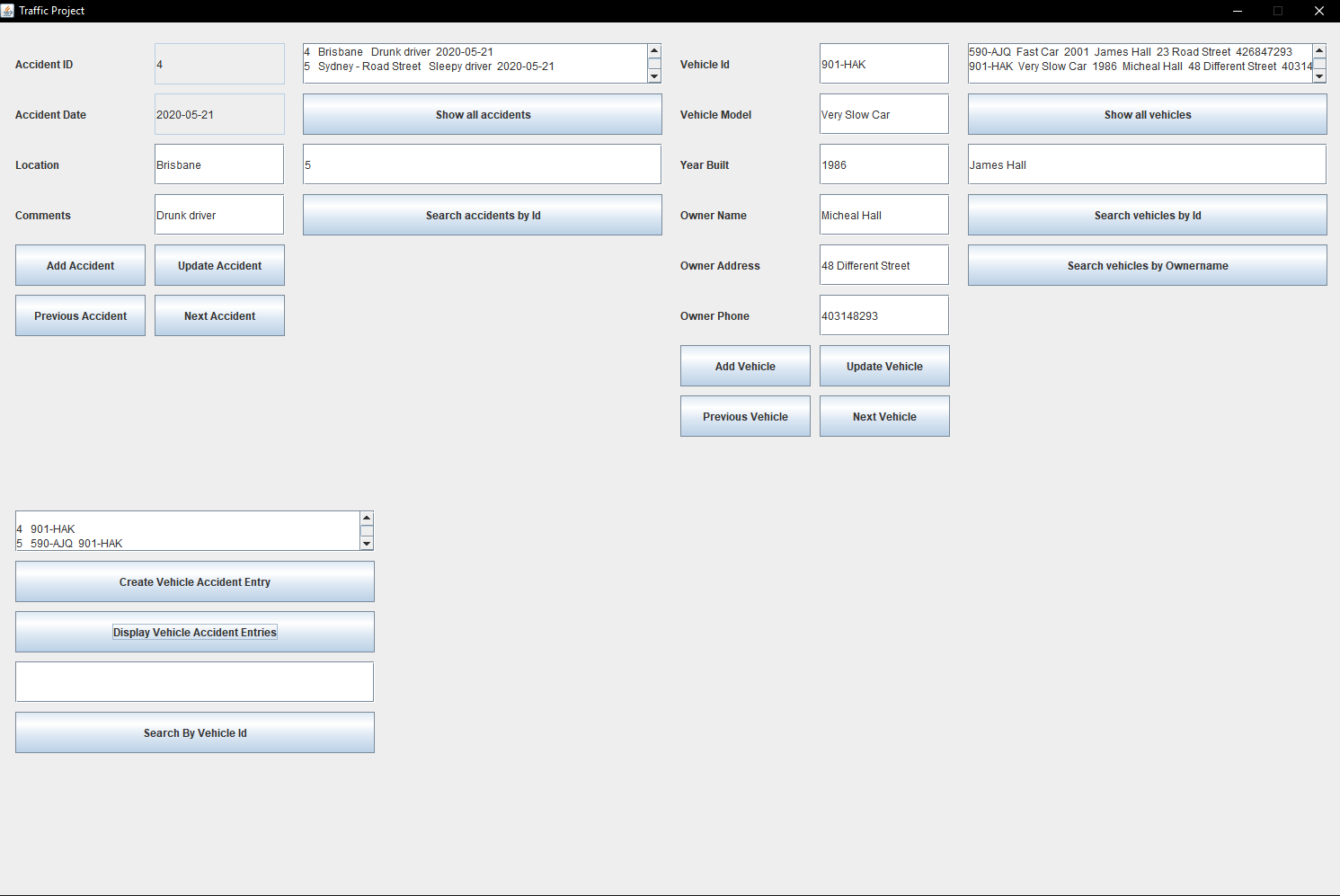
Vehicles can be filtered by either their Id, or their owner name. This is done by filling the text field below the show all vehicles button. The user can then choose which filter is used by clicking either the search by vehicle id button or the search by owner name button.

## Creating a vehicle accident entry



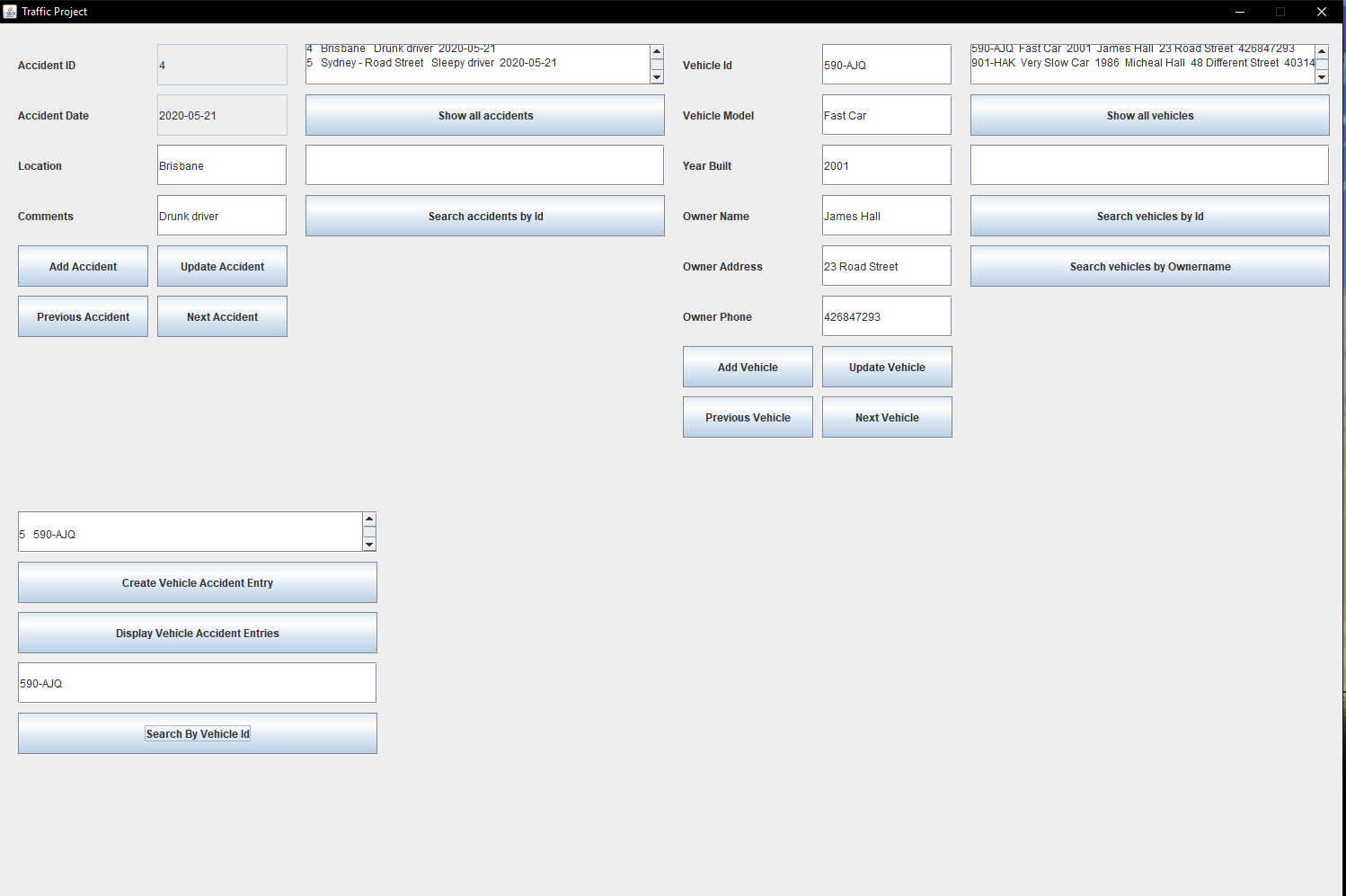
To create a vehicle accident entry the use must first select both a vehicle an accident. The user can then click the create vehicle accident entry button to create the record. If successful, the user will be shown the message vehicle accident saved.

## Displaying vehicle accident entries



To display vehicle accident entries, users can click on the display vehicle accident entries button. This will display the accident vehicle entries grouped by accident id.

## Filtering vehicle accident entries



To filter the accident entries, users can type in the text field above the search by vehicle id button. Clicking the button will then display the filtered entries.

# Version Control Discussion

## Introduction

Version control is an important term to consider when discussing the management and maintenance to software code. Version control systems are a category of software development tools that facilitates the development of code over extended periods of time. This is vital when considering large-scale software projects where the merging of code becomes exceedingly difficult and impossible through standard means of file sharing such as emailing. This report will discuss both the recommended technologies and models to use based on both team size and project scale (Bitbucket 2020).

## Current Technologies

There are a large variety of different version control systems/technologies that has been developed to help manage version control. To be considered as a successful Version control system, the tool must be able to perform a variety of functions. These being.

* Allow developers to see change history
* Allows multiple developers to simultaneously work on the same code
* Allows developers to merge their code
* Allows developers to isolate their code
* Allows developers to track changes made to the code
* Displays code conflicts
* The ability to revert changes made to the code.

There are many different version control systems, as such only the top three rated tools will be analysed and compared (G2 2020).

#### Microsoft Team Foundation Server

Microsoft Team Foundation Server or MTFS is a Microsoft based system of version control. While this tool boasts a large array of functions it relies on use of Microsoft products and the tools cost is expensive. The expense of using MTFS is greater the smaller scale business is using it compared to other tools. Due to this the recommended business scale for MTFS is an Enterprise.

#### AWS CodeCommit

AWS CodeCommit is an Amazon built tool, however unlike MTFS, AWS is not limited to use of Microsoft products and can be used for a variety of software. The main benefit of using AWS is the tools security and scalability. Another popular feature of AWS is its compatibility with GIT, another popular tool. Also, unlike MTFS while the price is still more expensive then average, it is significantly less so. The recommended project scale for AWS is small to medium businesses (Amazon 2020).

#### Git

Git is among the most popular and used version control tools it is used by both small business and enterprises such as Google, Android, Facebook, Netflix e.g. The main selling point of Git is its existing popularity as the more users of the tool the easier it is to implement git into an array of business. Another reason is that Git is a free and open source tool allowing both businesses of all scale easy access. Due to its accessibility, and reliance it is recommended that projects and teams of any scale should use Git (Sridhar, A 2018).

## Version Control Models

Version control models pertain to how data is distributed/shared within projects (DVC 2020).

### Local data model

The local data model is a local-only method where all developers use the same file system for version control. This model is recommended for only small-scale projects and teams. Popular local data model software is, Revision Control System and Source Code Control System.

### Client-server model

The client sever model method is done when developers used a single shared repository to store all files. Repository to a single shared server where all the files are saved for use. This version control model can be used by both low and medium scale businesses but is not recommended to be used by larger enterprise scale businesses.

### Distributed model

The distributed model is a peer to peer approach that has developers working within their own local repositories with code changes being made and merged as a subsequent step. This is the most complex version control model and it is only recommended that enterprise scale business use this model. For smaller scale businesses the previous two models are recommended.

# Bibliography

Bitbucket, 2020 ,‘*What is version control’*, viewed 23 May 2020 , <https://www.atlassian.com/git/tutorials/what-is-version-control>

G2, 2020, ‘*Best Version Control Systems’*, viewed 23 May 2020, <https://www.g2.com/categories/version-control-systems>

Amazon, 2020, *‘AWS CodeCommit’*, viewed 23 May 2020, <https://aws.amazon.com/codecommit/>

Sridhar, A 2018*, ‘An introduction to Git: what it is, and how to use it‘*, viewed 23 May 2020, <https://www.freecodecamp.org/news/what-is-git-and-how-to-use-it-c341b049ae61/>

DVC, 2020,*’Versioning Data and Model Files’,* viewed 23 May 2020, <https://dvc.org/doc/use-cases/versioning-data-and-model-files>