|  |
| --- |
|  |

|  |
| --- |
| **Project Name:** |

|  |
| --- |
| **Document Name :** |

|  |
| --- |
| **Document Summary:** |
| Details the requirements for the project. |

|  |
| --- |
| **Document Release Note** |
| Reviewed By: Date:  Date:  Approved By: Date: |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document Revision History** | | | | |
| **Version** | **Status** | **Update Comment** | **By** | **Date** |
| 1.0 | Created |  | Stephan Grieger | 15/07/2016 |
| 0.3 |  |  |  |  |
| 0.2 |  |  |  |  |
| 0.1 |  |  |  |  |

|  |  |
| --- | --- |
| **Distribution List** | |
| **Name** | **Department** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Definitions & List of Abbreviations**

| **Abbreviation** | **Description** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Table of Contents**

Contents

[Introduction 4](#_Toc457208751)

[Scope 5](#_Toc457208752)

[Home Page 5](#_Toc457208753)

[Article Page 6](#_Toc457208754)

[Create Article Page 8](#_Toc457208755)

[Moderation 9](#_Toc457208756)

[Moderation List Page 10](#_Toc457208757)

[Moderation Page 11](#_Toc457208758)

[GitHub 12](#_Toc457208759)

[Project Management 14](#_Toc457208760)

**s**

# Introduction

The site is a review site for vehicles. Members are able to create their own reviews or articles that can then be read by other members or members of the public.

Members can up or down vote articles which then adds to the articles popularity and gives the author reputation. More on this later on.

The articles are simple in design. You can add a title image, enter body text including links, and also select from a pre-determined list of tags. More on tags later.

Articles can be searched from the search bar. Articles are searched across the authors name, title and the tags. Do not search across the body text.

When a member submits an article, if they have enough reputation, then the article is automatically accepted by the system. If they do not, then the article is put up for moderation and a member with sufficient reputation can accept or reject the article.

The home page, by default, lists all articles in reverse chronological order, so the latest articles appear at the top.

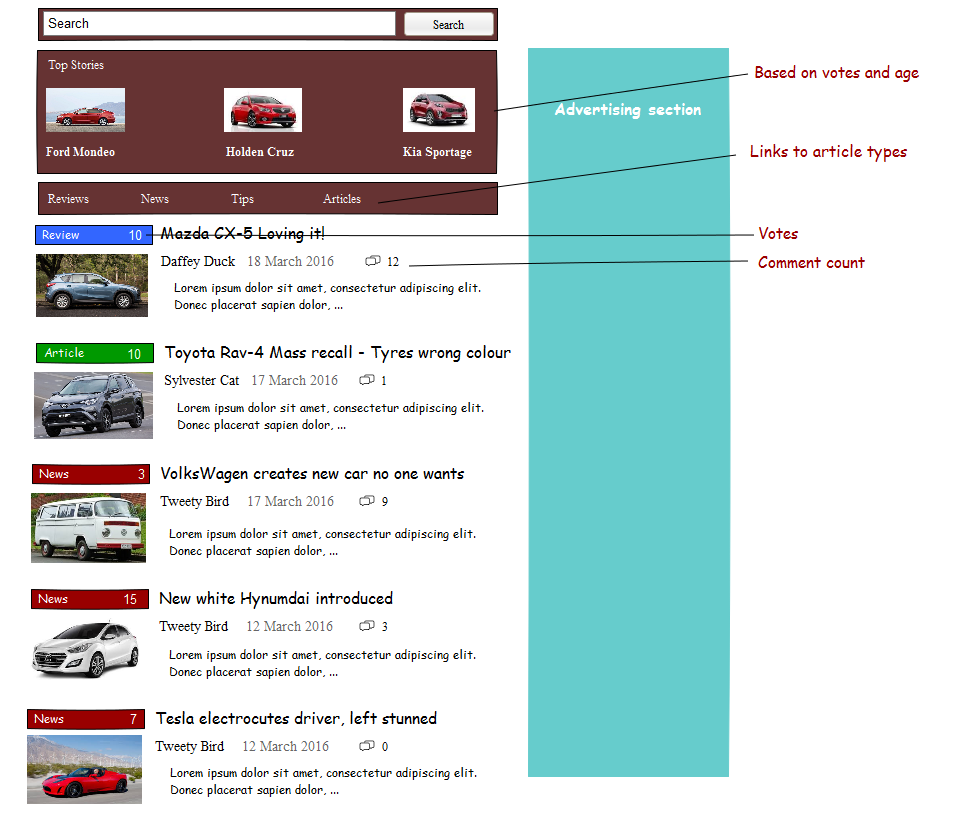
Users can then click a link type from the list at the top and the list is now filtered based on that selection.

Members can also comment on articles. Those articles are also subject to moderation until the member making the comment has sufficient reputation points.

# Scope

### Home Page

Home page



The home page is the landing page when a user navigates to the site. It lists 10 articles and as the user scrolls to the bottom of the page, a further 10 articles are loaded and displayed under the previous 10.

Clicking on an article, will navigate the user to the article page and load the selected article.

Users can also search articles from here. When a user enters search criteria in the search box and then either presses [enter] or clicks on the search button, the system will search all articles and display the best 50 results. The system should search for the words within the Title and Author but not within the Body Text.

Clicking on the article type link at the top will navigate show only articles of that type again in reverse chronological order.

Only the first n words of the article are shown in the list.

Votes are shown to the right of the article type.

Number of comments are shown next to the date of the article.

Space is provided to the right to allow for advertising content. Advertising will most likely be in the form of jpg’s that are shown one under the other. Clicking on them will do two things.

1. Navigate the user, in a new window, to the advertiser’s webpage listed against the advert.
2. Save information to the database;
   1. Which article was clicked and when.
   2. If the user is logged in, then who clicked the article.

When any user clicks on an article, some details are also captured;

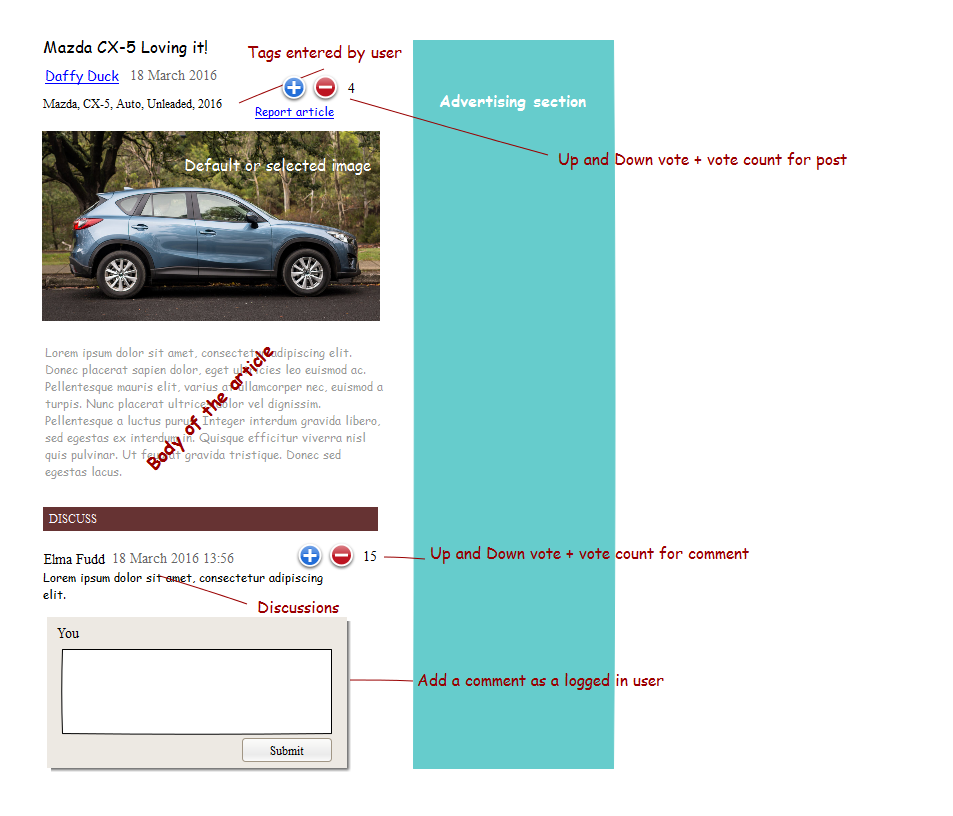
1. Who clicked on the article, if there are a logged in user, guest otherwise.
2. Date and time they clicked on the article.

The top stories section is based on the number of clicks, up-votes and age of the article. So if an article is say an hour old and has had 100 clicks already, then this article should be in the top stories.

You need to come up with an algorithm that makes some sense. You don’t want articles to be in the top three for ever so the age of the article must be a factor.

### Article Page

Article page



This is where a user can read the full article.

The full text of the article is displayed here and is not paged.

 Fig 1

The heading is comprised of several key features. Care will need to be taken to not make this look cluttered and confusing.

Essentially what needs to be up here is the title, the author, date of the article and any tags the author attached to the article. The Author’s name is a link that when clicked will take the user to the search results page listing all the articles that that author has written.

The [report article] link will place a flag against the article in the database. This will then appear in the moderators list of articles. In phase 1, the reason is not required, just the flag. The flag count may get quite high if this is an offensive article so the flag needs to be a count or a new record in a report table.

The Plus and Minus buttons allow the user to vote on the article. Each user can only up-vote one or down-vote once. They can change their vote from a down to an up if they wish. They cannot up-vote twice or down-vote twice.

The number of votes is shown to the right of the down-vote button.

 Fig 2

This is the image that the user selected. If they did not select an image, then the default image from the Glasses web service is used.

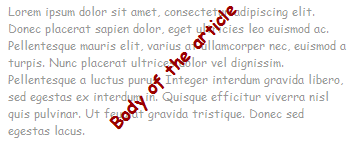
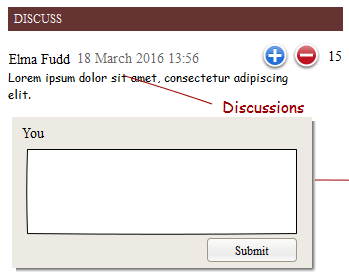
 Fig 3

Fig 3 is full body of the article.

 Fig 4

This is a list of all the comments made against the article. If you are a logged in user, then you can leave a comment and/or up and down vote the comment. There should also be a link to be able to report the comment just like at the top with the article.

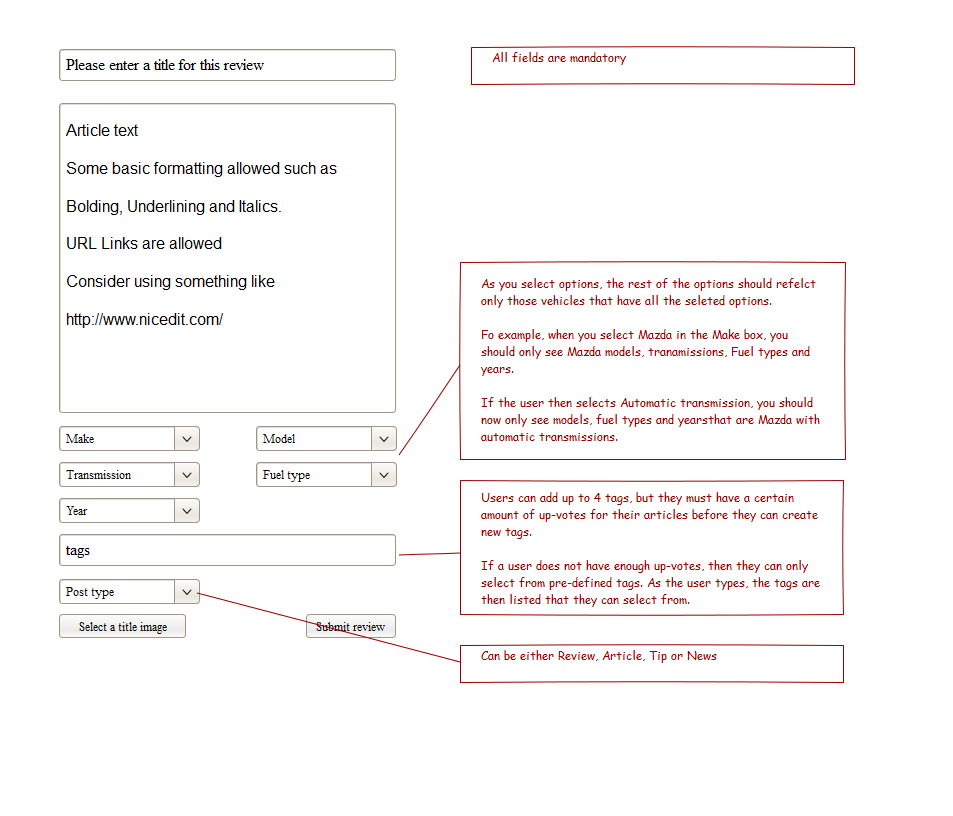
If a user has enough reputation then the comment is immediately posted, otherwise it needs moderation.

The comments are listed in reverse chronological order.

Members can add as many comments as they please.

### Create Article Page

Create article page



This is where any logged in user can create an article. All the fields, other than perhaps the tags field, is mandatory.

If a user has enough reputation, then the article will be posted live immediately otherwise it falls into a moderation queue.

The body text should be rich text and the user should have enough control to be able to bold text, underline and italics text. They should also be able to insert links that a reader can click on. Links should open a new window and not move the user away from our site.

Suggest using a free jQuery editor such as, but not restricted to, [www.nicedit.com](http://www.nicedit.com).

Users should not be allowed to insert pictures into the body text. That is what the title image is for.

The section dealing with the vehicle the user is writing an article about needs to be dynamic. So if a user enters Mazda in the Make box, then only Models, Transmissions, Fuel Types and Years that relate to Mazda are in the lists. If the user then also selects say 2016, then only Models, Transmissions and Fuel Types for 2016 Mazda’s are available in the drop down boxes.

This must happen in place without a full page post back each time. So Ajax needs to be employed heavily here.

The [Select title image] button allows the user to select an image from their computer to use as a title image. This file is then stored on the websites file system with a reference pointing to it within the database.

To select the Make, Model, Transmission etc. of the vehicle the user is writing an article about, they must select them from a series of drop downs.

When the user first enters the site, only the Make dropdown has data in it containing all of the makes available. This list can be obtained by consuming a web service that SafeCode has written and will make available. This URL should not be hardcoded, rather it should be configurable within the web.config file.

Once the user has selected the make, the mode dropdown is filled with all the models that are available for that make of vehicle. This must be done with an Ajax call so that the UX is seamless.

When the user selects the model, the transmission is filled with data, then the fuel type and finally the year.

A good place to start is <http://www.carsales.com.au/all-cars/search.aspx?adv=1&csn_tn=true#{"searchadv":{"Service":"Carsales"}}> where the dropdowns are disabled until you make a selection in the previous dropdown.

Only when the user has made a selection in all the dropdowns, can they save the article.

### Moderation

The whole site is based on the StackOverflow.com method of moderation whereby as a user creates articles, responds with comments and gets up voted, they gain reputation points.

As a user gains reputation, they will be given access to increasing levels of moderation.

The loss of points should be configured, along with the list of reasons, within the database to make it configurable.

The access level of the points should be stored in a database table so that they can be easily configured at a later stage. The access levels, in order are;

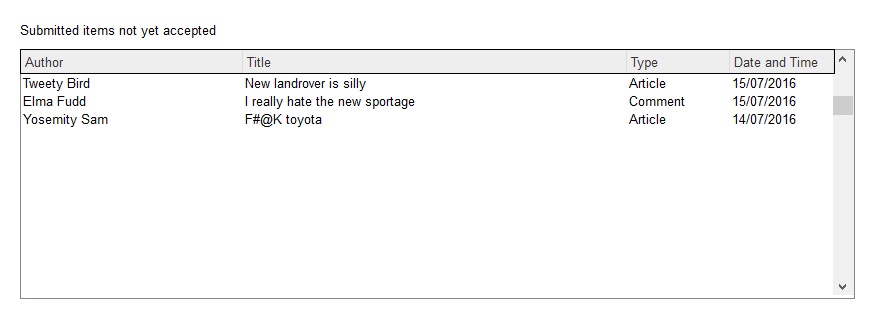
The moderation levels, ignoring the actual points, are as follows;

1. Create an article.
2. Comment on an article.
3. Approve / Reject comments.
4. Delete comments.
5. Approve / Reject articles.
6. Delete articles.
7. Edit articles.

These initial levels need to be built into the system but the point values of each level should be stored within the database for easy configuration.

### Moderation List Page

The moderation page is open to anyone that has enough reputation points. It should be accessed from the title bar.



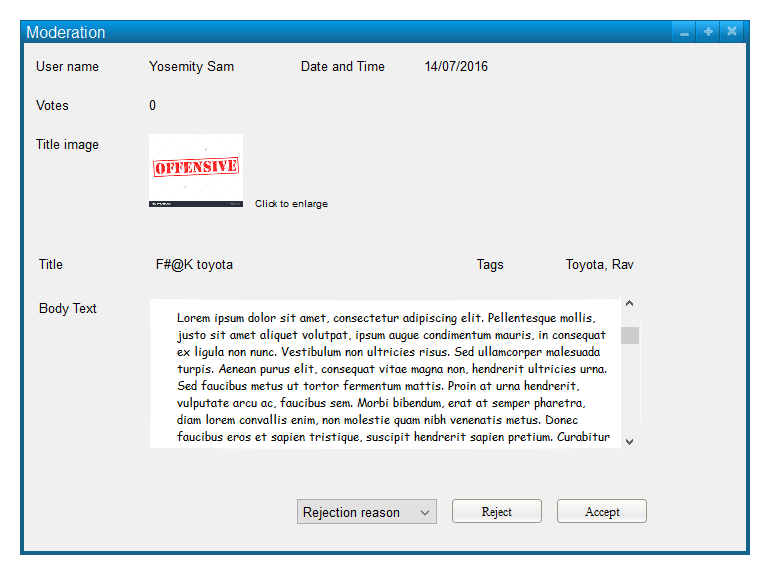
It’s, at its core, simply a page that lists all the articles and comments that need moderation.

A moderator’s job is to ready the article, look at the image, links and tags and ensure that they are not only appropriate for the platform but also not abusive. If an article is not approved, the moderator needs to provide a reason from a drop down list of reasons.

Depending on the reason, the author may also lose a number of reputation points or they may be marked as suspect and need all their future comments and articles reviewed for a certain amount of time or acceptances.

So if the article was inappropriate then maybe nothing happens but if it were highly abusive then they would lose points.

### Moderation Page



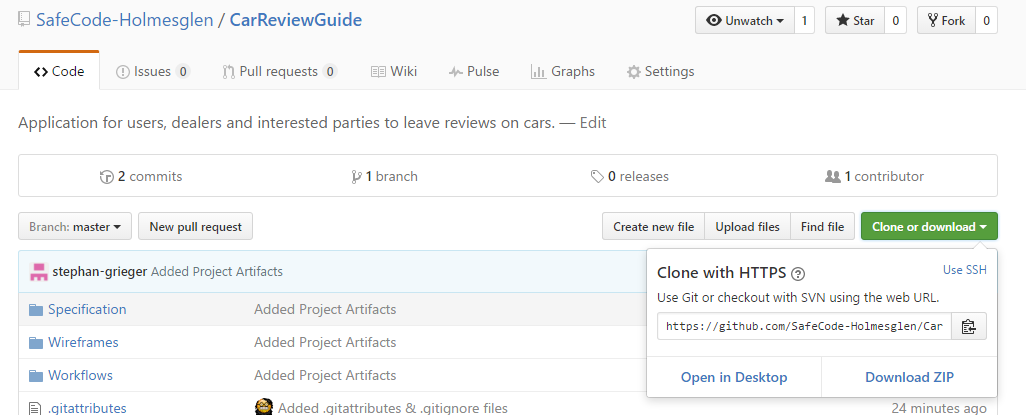
# GitHub

Our source control of choice for this project is GitHub. A repository has been setup for this project here;

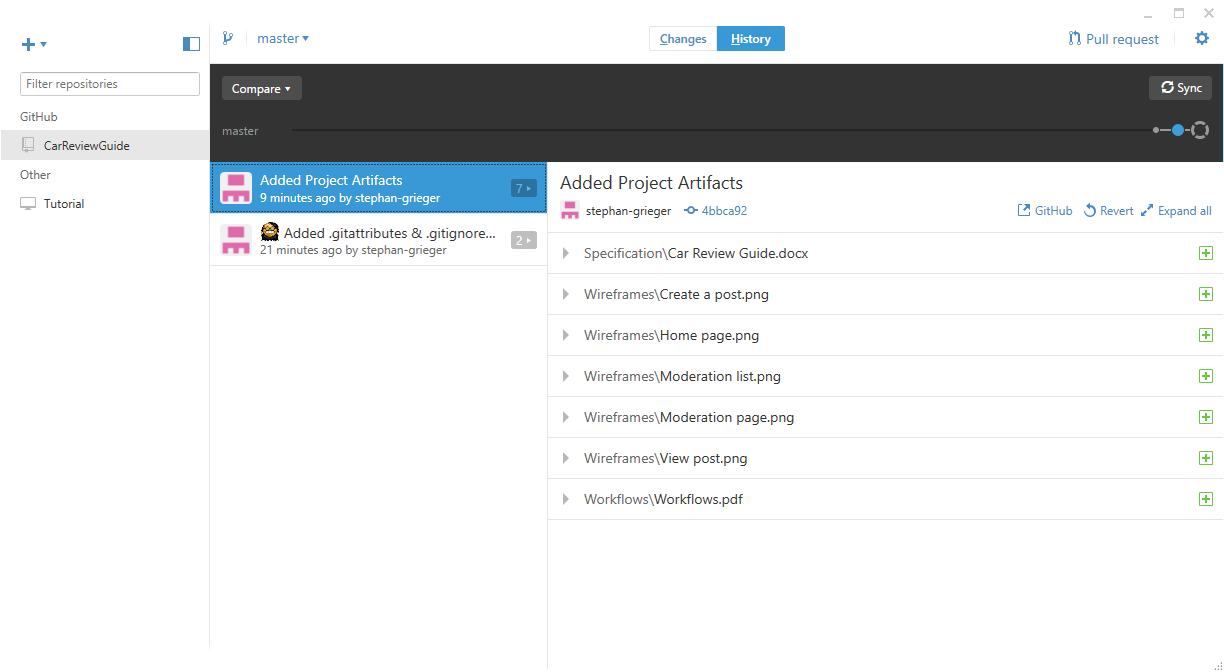
<https://github.com/SafeCode-Holmesglen/CarReviewGuide/>

We recommend you install GitHub desktop and run your source control out of there.

This is a quick start guide for the desktop version.

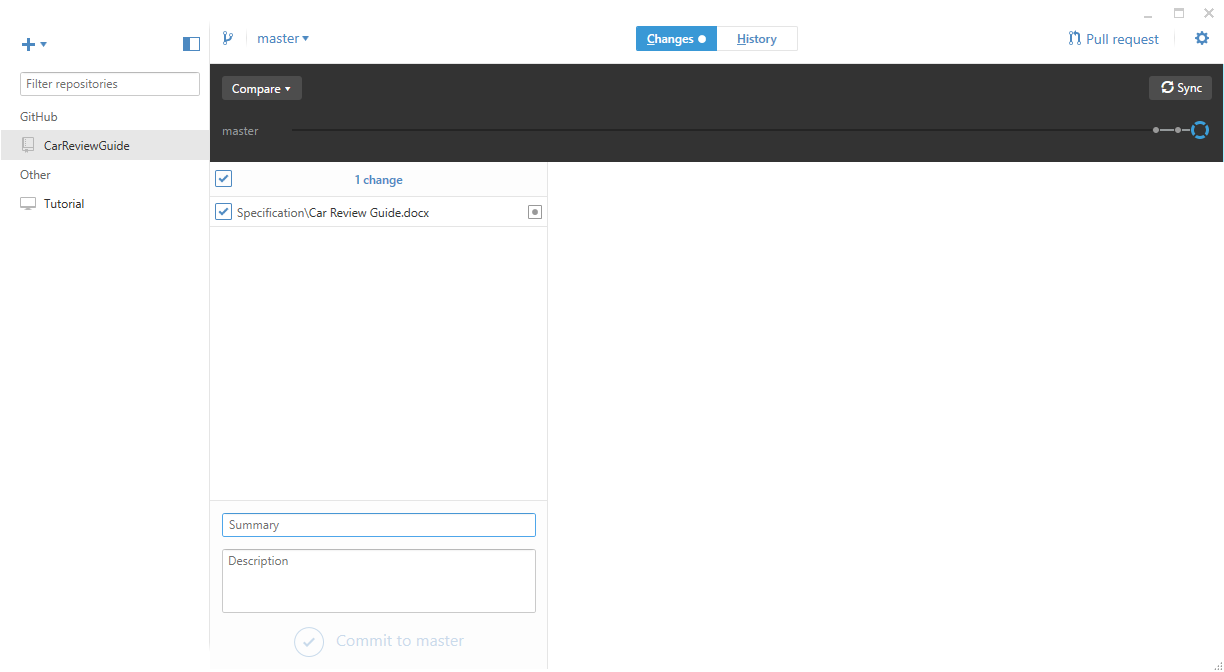


You will need to setup your project first. The easiest way to do this is to log onto the web version of the repository and click the [Clone or download] button followed by [Open in Desktop]. This should create a repository on your hard drive that you can then open using Visual Studio.

Once Git has opened you should see something like this;

This shows all the files that are in the repository.

When you make a change to a file, it will appear in the [Changes] tab located at the top of the window.



Above you can see that the file “Car Review Guide.docx” file has been changed and is awaiting upload to the server.

You must provide a Summary to commit the file. If you do not, then Git will not allow you to upload the change.

We require you to put in a summary of the changes being committed to the repository. Although the Description box is not a required field, you should get into the habit of entering a description so that others can follow what the changes involved.

We don’t require you to enter a thesis on the changes, just dot points. So for example you might upload two files; index.html and HomeController.cs.

Your summary might be;

Added user details to index page

Your description might be something like;

Added code to return the user first name, last name and phone number to the view as well as show it on screen.

# Project Management

We understand that you are most familiar with Trello and we are more than happy to continue this.

We will leave it up to you to organize this but once you have it setup, please email [Stephan.grieger@safecode.com.au](mailto:Stephan.grieger@safecode.com.au) with login details. Also, you will need to make Stephan an administrator of the project.