Niall Ryan
12114812

Technical Report – Musical Instrument Shop Web Application

This project was completed on my own due to my project partner dropping out of the course.

The Github repository which we used up until the 13th of December when my partner dropped out can be found at http://github.com/niallryan/Instrument_Shop_NEW

This is a working copy of our original repository which became corrupt due to some merge conflicts that weren't resolved correctly.

The original repository can be found at http://github.com/niallryan/Instrument_Shop

Implementation:

The Musical Instrument shop web app incorporates most of the required elements for this project. It features a number of pages including a searchable product inventory, a registration page which allows new customers to sign up to place orders, checkout pages which allow users to see the items in their carts before entering purchase information and a thank you page that confirms that their order was processed successfully. Work on a user profile page that would allow users to edit their details and view order history has been begun but not implemented successfully. The additional functionality outlined in the project proposal has also not been implemented.

Issues Encountered:

We encountered a number of issues while working on this project. The first major issue encountered was a problem using Github correctly to share our project. We wanted to get as much work done in the first month of the project as possible to allow us to better distribute the heavier workload that would come when other course project deadlines crept nearer but did not spend enough time on learning how to effectively use Git to share project work. We were confronted with merge conflicts and other errors that slowed us down and led to us having to meet up more frequently to work on

the project which led to time management issues as other project deadlines drew nearer.
We worked through the Git problems and completed all the in-class Laptop Shop tutorials without major issues and then separated to finish work on other projects. I kept tinkering with the project and started to work out how we would implement the two major pieces of additional functionality that we had set out in our project proposal.
However, an issue arose when other projects were completed as my project partner informed me he would not be able to contribute to the project any more and intended to drop out of the course.

Decisions Made During Implementation:

The biggest decision I had to make then was how to proceed with the completion of the project on my own. As the deadline drew nearer, I decided the best course of action would be to stop work on implementing the additional functionality we had outlined in the project proposal and to focus on fully implementing the required functionality that was set out in the project brief. This allowed me to not be overwhelmed by the amount of work that was left to complete the project as we had originally envisioned it.

Other decisions that were taken while working with my partner were to stop relying solely on Github for sharing project work, to make regular backups of all our work and to try to effectively manage the time we had to complete the project in advance of the deadline.

Explanation Of Proposed Features Not Implemented:

Some features of the project specification have not been implemented, ranging from small features to larger ones.

I had trouble getting the cart to display properly in a sidebar in the application. I understand the concept behind this feature and tried to use partials to display the show action of the cart controller in the sidebar. I had tried this for the login area too but also encountered issues. While I succeeded in creating a work around for the login area (which, while not exactly DRY at least works), but implementing this work around for the cart did not succeed. I assume the issue I was having relates to an @cart instance variable not being available for each page that needed it but was not successful in fixing the problem as other issues required my attention.

The user profile page, which would allow users to edit there details and view order history has not been implemented successfully. The code for the basic functionality is there in the app but I had issues with the pages displaying correctly: the files are in the view folder but when I tried to load them in my browser I was either greeted with errors I did not understand, with a page that I did not expect to be loaded, or being redirected to the login area. The redirection leads me to believe that the error lies with permissions that I have set somewhere else in the app but due to time constraints I was not able to resolve these. The fact that the tutorial I followed to help me write this functionality was based on Rails 2 may have caused an issue but I was unable to locate any specific problem that I could have fixed by making the code conform to Rails 3 standards. Due to this problem, I was also not able to implement the order history functionality as I would have liked.

The other functionality that is not implemented includes additional functionality that we had described in our proposal, a wishlist and a second-hand ads section where users could create their own advertisements to post on the site, which could be viewed by other users. I had begun to sketch out how this functionality could be implemented but had to drop these features when I was informed I would have to finish the project on my own.

The wishlist feature, in my head at least, would have been implemented similarly to a cart, but with enhanced persistence. The user would be given an option “Add To Wishlist”, like the “Add To Cart” button, which would add a line item to the wishlist, which would then be saved when the user logged out, and available when they logged back in. Obviously, there would be a lot more detail involved, including permissions so that only the relevant user or an admin could view their cart, as well as security issues that would need to be solved.

The ads feature would have been implemented similarly to the implementation of the products functionality in the app. Users would be able to create an ad, with a name, description, price, pictures and other details which would be viewable on an ads page, similar to the products inventory page. Ads could be viewed by anyone but only edited only by an admin or the original creator of the ad, which would mean incorporating permissions based on a user id field or by screen name. The user profile feature which is not fully implemented in the app would have been useful here as it would display all ads created by each user as well as allowing for interaction between users.

Bibliography:

As well as the Laptop Shop tutorials and other material provided in class, the following materials were also consulted in the implementation of this project.

#178 7 Security Tips - RailsCasts
Railscasts.com (2009) *#178 7 Security Tips - RailsCasts.* [online] Available at: http://railscasts.com/episodes/178-seven-security-tips [Accessed: 22 Dec 2012].

#356 Dangers of Session Hijacking - RailsCasts
Railscasts.com (2012) *#356 Dangers of Session Hijacking - RailsCasts.* [online] Available at: http://railscasts.com/episodes/356-dangers-of-session-hijacking [Accessed: 22 Dec 2012].

Agile Web Development with Rails
Ruby, S. et al. (2011) *Agile Web Development with Rails.* 4th ed. Dallas: Pragmatic Bookshelf.

Chapter 7: User Profile
Rails.francik.name (2009) *Chapter 7: User Profile.* [online]
Available at: http://rails.francik.name/tutorial/chapter/8 [Accessed: 22 Dec 2012].

Ruby on Rails Guides
Guides.rubyonrails.org (n.d.) *Ruby on Rails Guides.* [online]
Available at: http://guides.rubyonrails.org/ [Accessed: 22 Dec 2012].