# Artefacts for Michael Leontieff-Smith #N9455396

## Scrum Master

(95% of the development for the system In Release One and Release Two was completed by myself, group members handled the planning, static pages, testing and acceptance criteria)

Artifacts are evidenced through the entire project repository, which is included.

# Release One Artifacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **What did you do that made a meaningful contribution to the teamwork in producing the final solution?** | **Evidence of that work (artefact)** | **How did it fit in the team work/project work?** |
| 1 | Completed the entire skeleton structure of the Package Delivery Application, this includes the routing, views, models, controllers and dependencies for development. Forms the basis for both release one and the following releases | Project Repository  (app/controllers), (app/models),  (app/config/rotues) | This vast amount of work accommodates a majority share in all work completed for release 1. This artifact represents the design and implementation of the Model View Controller (MVC) design pattern. After this artifact was completed, group members entered their work into the system, without having to route/control/model it. |
| 2 | Implemented ALL database systems into the application required for release one. This includes scaffolding, migration and the schema for tables: Customers, Drivers, Orders.  The active record associations were also completed as part of this artifact along with all queries used in the system for this release | Contents of  (app/controllers), (app/models),  (app/db ) <- migrations included within here, scaffolding is contained within the sub folders of (app/views) where the folders are the name of the database table | This was a critical component of our web solution as it allowed the storage of user supplied information, including accounts and orders. After this was completed, the web solution was in a position where the online forms controlling user accounts and orders could be implemented as part of release one. |
| 3 | Created and implemented the base styling (CSS) for: homepage, customer login page, customer creation page, customer management page, driver creation page, driver login page, driver management page. including bootstrap elements like the navigation bar and buttons, text formatting and other refinements. | Majority of the contents of the Style sheets Folder | This work was done due to another team member not submitting their work by the deadline. This benefitted the team as we were the able to present a complete system by the presentation date. After sprint one, where his work was due, he then was able to present something, and his work build on top of this. |
| 4 | Created and implemented the various forms required by the system to collect user input:   * Customer Creation * Customer Login * Driver Creation * Driver Login * Order Creation   For release one | Customer Creation Screen, Driver Creation Screen, Customer Login Screen, Driver Login Screen, New Order Form | This allowed data to be submitted to the database from the website. And as such represented the major functionality of the web solution for release one. Functionality includes   * Customer Login * Customer Creation * Order Creation * Driver Creation * Driver Login |
| 5 | All work required to create and manage Sessions was created, this includes authentication and session storage | Project Repository  (app/controllers), (app/models),  (app/views) | Provided front-end interaction with the database and allowed the various user types to log into the system. |
| 6 | The HTML required for sprint one and two was implemented by myself, with the exception of a few code snippets provided by other team members. This largely involves page layouts, div containers etc. as well static content such as links/body text etc. | Project Repository | This provided the front end implementation of the backend features, and allowed the user to interact with the website. |
| 7 | Implemented the production server branch and configured it to allow deployment as a production version to an online server | onthespotdelivery.  herokuapp.com | Allowed a production version of the project to be configured and demonstrated during team meetings |

# Release Two

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **What did you do that made a meaningful contribution to the teamwork in producing the final solution?** | **Evidence of that work (artefact)** | **How did it fit in the team work/project work?** |
| 1 | Implemented Driver Management Page, Operator Management Page | Project Repository | These were the pages that presented all of the functionality required by the individual users, each page is restrict based on user, which is defined by the current session id. |
| 2 | Implemented Database Schema for Driver Logging and Operators, including the following tables:   * Operator * CheckingIn * Pickup * Deliver * Payment | Project Repository | These tables encompassed the database interaction for release two, they enabled Operator accounts to be created and stored, and orders to be logged throughout their delivery process. |
| 3 | Implemented all forms required to collect information during operator creation and driver logging. | Project Repository | These forms are the front-end interaction with the database, and collect the supplied information, format it and store it in the tables as records. This is a major frontend component of the website. |
| 4 | Implemented the Code Behind for the Order Tracking System. | Project Repository (operators\_controller.rb) | The numerous methods required to generate reports for each order at various stages was completed. Each report type (New Orders, Tacking, Awaiting Payment, Completed) took the form of an array which included every order pertaining to that record. This was passed onto the View. |
| 5 | CSS, HTML, Embedded Ruby implementation of tracking interface View | Project Repository | This involved the implementation of each cell type, then hooking it up to the reporting systems created in the controller, this allowed the generated reports to be displayed to the user. |
| 6 | Added JQuery Order Confirmation | Project Repository | This allowed the order submission to be prevented unless the user double checks and accepts the information a second time. This prevents incorrect orders entering the system. |