Artefacts

# Release 1

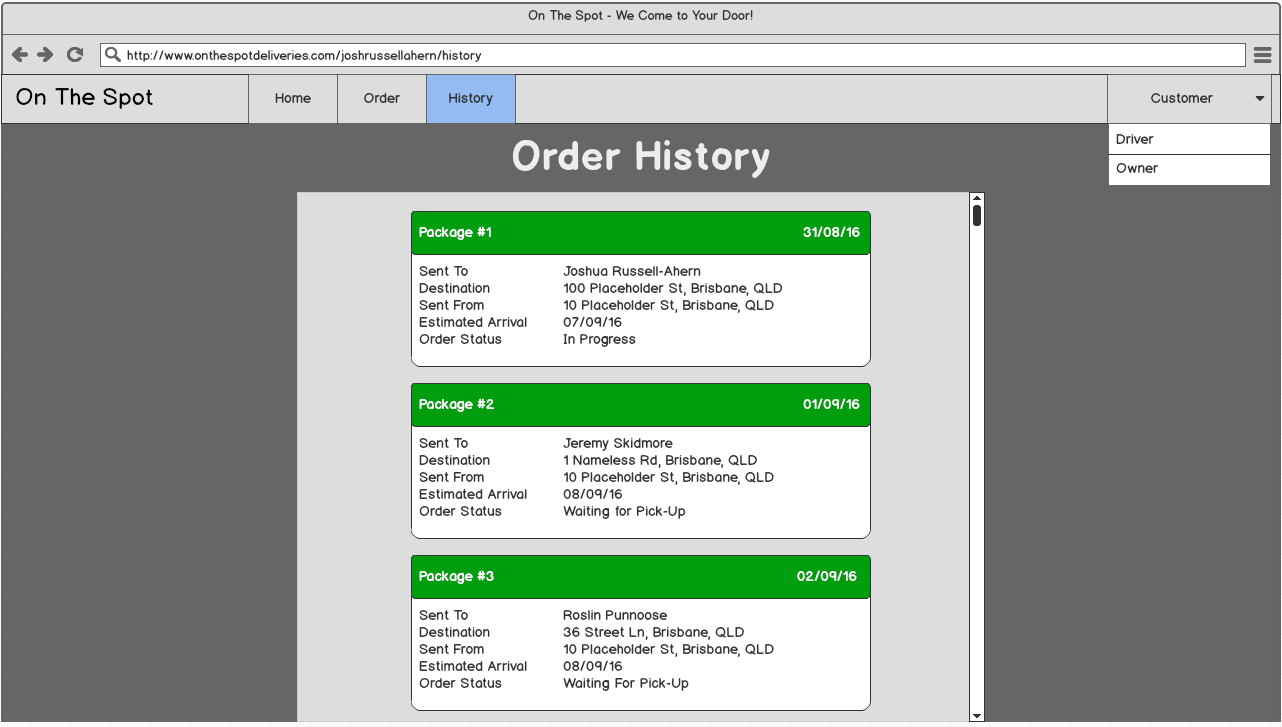
|  |  |  |
| --- | --- | --- |
| **What I did** | **Evidence** | **How it fit the project** |
| My group did not have much programming knowledge so I learnt PHP and HTML (Using Bootstrap) so that we were able to create the website.  PHP was chosen as the other CS student in the group had previous PHP experience. | Followed tutorials on PHP and Bootstrap on <http://www.w3schools.com/>  As well as documentation on <http://getbootstrap.com/css/> and <http://php.net/docs.php> | It allowed for us to convert the user requirements into a product. |
| Deployed the website and database to an online server. Configured the server to auto deploy our GitHub code when a commit was pushed to the main branch. | The website hosted at <https://onthespot.herokuapp.com/> was set up by myself. Used this add-on to deploy the database <https://elements.heroku.com/addons/jawsdb> | This allowed for our project to be easily accessed by the client so they could review progress and other group members who were not coding but doing manual testing. |
| Created mock-ups of the cards system we would use to display orders for different users, as well as how they would be displayed on the web page. | Mock-ups were created using Balsamiq; see Appendix 1. The mock-ups were discussed during the group meeting and the design was accepted. | Gave the group a clear understanding of how we were going to display our information. This design made it to the final product. |
| Created the final release plan for the sprint, as well as updating the sprint plans and inputting the data on time taken. Initial draft feedback from client on the stories we would include to create this document. Also managed burndown charts for both sprints. | See attached files:  Release 1 – Release & Sprint Plan Release 1 – Burndown Chart | Provided a clear goal for our team to work towards that also met the client’s needs. Burndown charts allowed us to see if more work was required or if we were on track. |
| Developed the HTML with Bootstrap integration for every page in Release 1. | Link to repository at time of Release 1 completion <https://github.com/RTLM/IFB299-Group-12/tree/a3872e433054f124c434bcaad2204df84741b08b>  Can also be viewed on the deployed website. Changes made in Release 2 also completed by me, bar minor styling with images and colours. | Core product functionality |

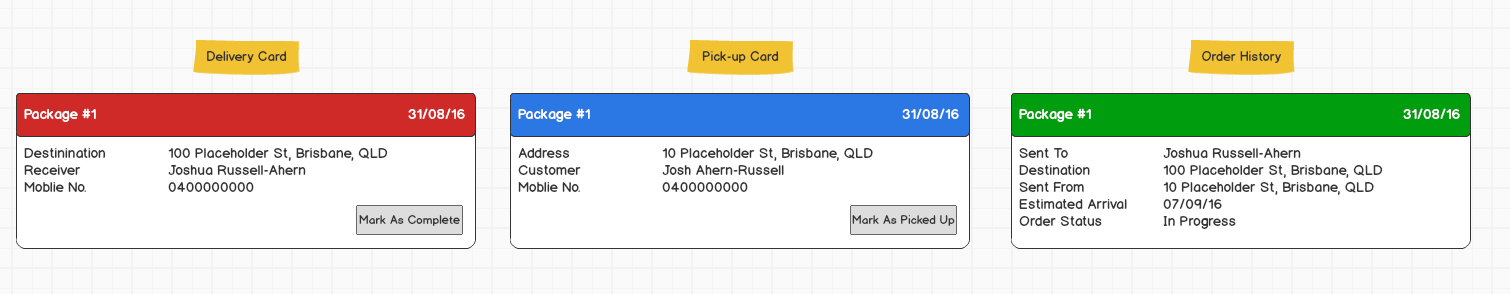
# Release 2

|  |  |  |
| --- | --- | --- |
| **What I did** | **Evidence** | **How it fit the project** |
| Created a PHP script to dynamically generate our website’s header and footer based on the user type that was signed in. Also allowed for changes to be easily made if pages needed to be added or removed. | Located in the file PHPfunctions.php the script is called “updateNav()”. Other functions created by me can also be seen as per the function comments. | This function was created due to client feedback on the footer not being updated. Allows for further changes to our website structure to be made very easily. |
| Further separated the functionality of our “Driver” and “Owner” account types. Attempting to keep data and functionality available completely relevant to the user type. | Signing into the website as an Owner will provide a different set of pages, as well as functionality such as “Assign Driver”. Delivery drivers will only see orders designated to them.  Sign in at: <https://onthespot.herokuapp.com/>  Owner – User: [owner@gmail.com](mailto:owner@gmail.com) Password: owner  Driver – User: [driver1@gmail.com](mailto:driver1@gmail.com) Password: driver | Further meets the clients’ needs. Client does not want the drivers to have access to all functionality and information. |
| Made changes to the Release and Sprint plans based on a review of client requirements. Features that we realised were more important to the client, that we had included in future releases, pushed forward to sprint 4. Continued to manage the burndown chart. | See attached files:  Release 2 – Release & Sprint Plan  Release 2 – Burndown Chart | Ensured that our project was meeting our clients’ requirements instead of what we thought they may want. Continues to provide direction for other team members. |
| Modified an existing JavaScript function for status updates to call on a PHP file that executed an SQL query, updating the timestamp of the package. | Located in the javascriptFunction.js, the script is called “updateStatus()” and the PHP file with the SQL query is named “updateTimestamp.php” | Website functionality. Shows use of a range of technologies |
| Designed the final database structure, as well as contributed to the development of the database throughout the project | In Appendix 2 there are 2 images include. The first is the initial draft of our database design. As our project functionality grew, more fields were required, the second image is a diagram of our current database structure | Core website functionality which allows us to store all data. |

Appendix

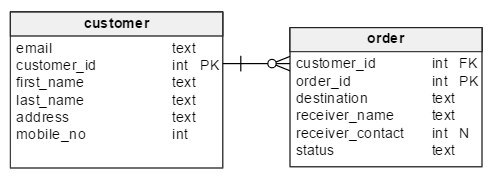
# Website Mock-up





# Database Diagram

## Original Mock-up



## Current Design

