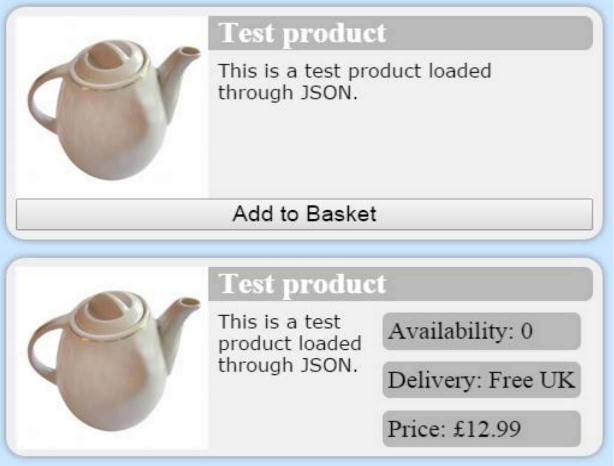
Technical report

This document outlines the current state of the product as of 18th January, approximately two and a half months into development and assumes some prior knowledge of the project's aim, product specification and features.

High fidelity designs for the content management system (CMS) and customer user interfaces are complete, in addition to a number of sketches of potential implementations of the management UI. Designing the user-end of the CMS lead to the realisation that many components intended for the customer UI could be reused for this, meaning many common elements (for example AJAX calls and the item browser) could be separated out into a custom library, with only the most specialised scripts (such as for the basket on the customer UI) needing to be integrated separately. This design technique should significantly cut down the time required to implement and test each user interface, while also increasing maintainability. Unless otherwise specified, the components discussed below are designed to be integrated in both the CMS and customer UIs.

Categories are now loaded dynamically through JavaScript and can be selected by the user, doing so sends an AJAX request to the server requesting details of products in that category, in JSON format; the server currently only responds with test data as the PHP scripts for querying the database are not yet complete. Once the data is received, the products are listed in a minimised, compact form.



Test product

This is a test product loaded through JSON.

Test product

This is a test product loaded through JSON.

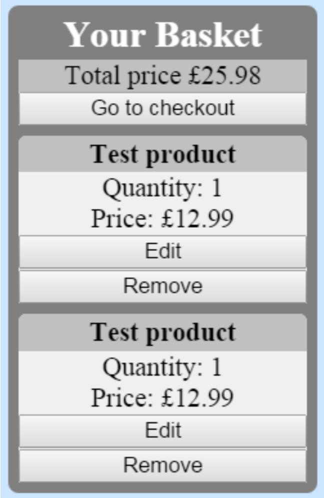
Availability: 0

Delivery: Free UK

Price: £12.99

Clicking on a listed product toggles it between its compact and full information forms, doing so sends another request to the server to get the necessary product information. This approach reduces network traffic and server load due to the minimised form (the default when products are first listed) requiring less data from the server, saving the more expensive, media rich, full information calls for products the user has shown interest in. The expanded form is currently incomplete, acting as a placeholder until further progress has been made on the server-side scripts for processing product requests. A side-by-side comparison of the current state of expanded (top) and minimised (bottom) product listings is included to the right as a reference.

There is no limit to the allowed number of expanded product listings and changes to the page are made dynamically through JavaScript with no reload/redirect. On the customer UI, the expanded form of a product listing additionally features an "Add to basket" button which, when clicked, adds the product to the users basket, displayed on the right of the screen (see right) and updates the total cost of their order accordingly; on the CMS UI this will be replaced by an "Edit listing" button to enable manipulation of product details. There is no validation currently in place for this system but in future a request will be sent to the server to verify that the quantity requested can be supplied. When proceeding to the checkout and periodically while items remain in the basket, it will be also be necessary to perform this check. As of writing, the 'Remove' and 'Edit' buttons on items in the basket are non-functional, however these should soon be complete.



Your Basket

Total price £25.98

Test product

Quantity: 1

Price: £12.99

Test product

Quantity: 1

Price: £12.99

Design of and code for the setup and creation of the database, as well as sample test data is complete, for this purpose, and as a future reference to benefit maintainability, an entity relationship diagram was created (not included here to preserve brevity but will be present in the appendix of the final report). The aforementioned code is currently in the form of SQL files which will soon be integrated into PHP scripts for use on the server. Meanwhile, a simple PHP responder is currently configured to answer AJAX requests with test data in JSON format; other than this, development of the server-side scripts is on-going.

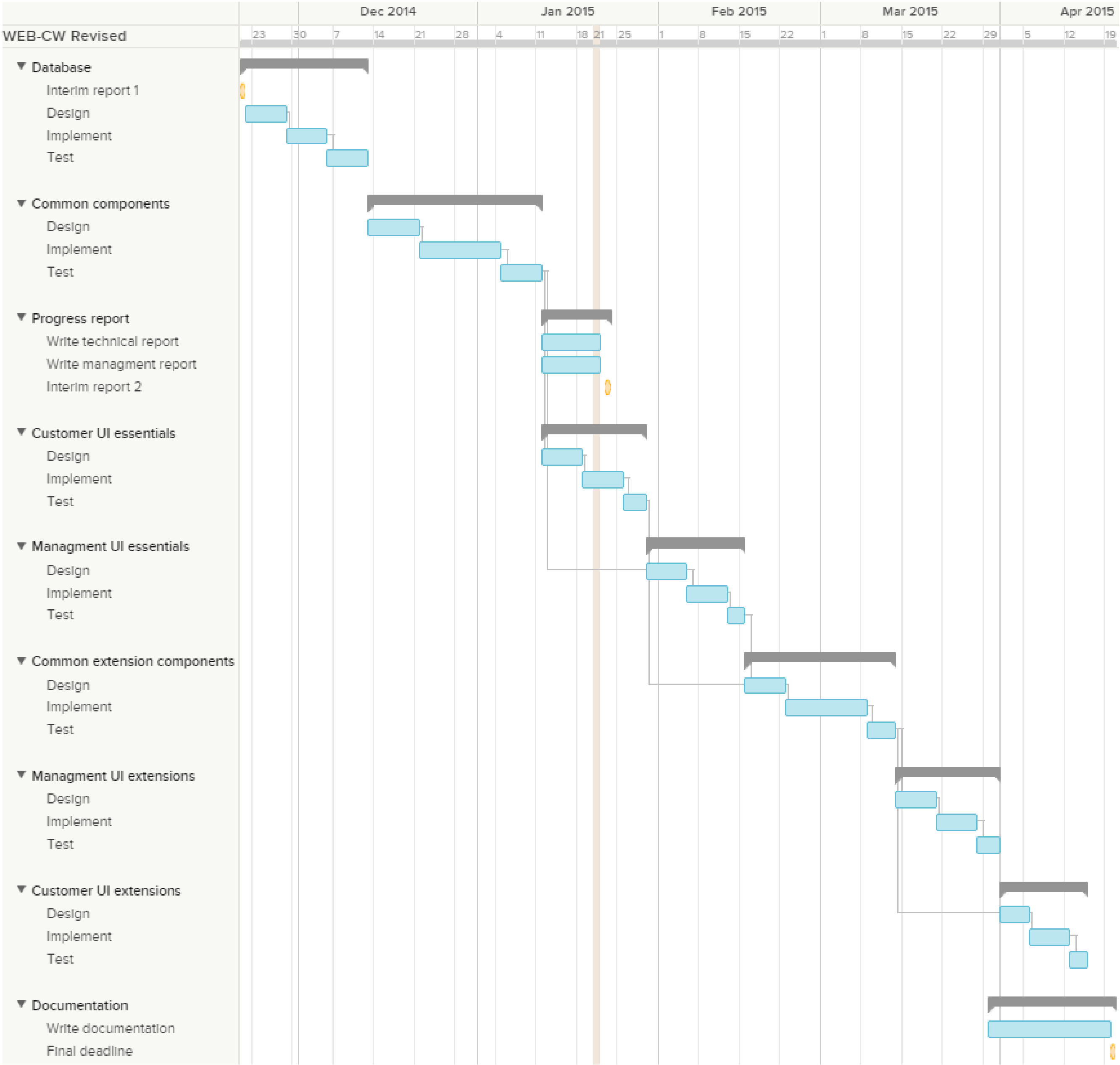
Throughout the project a number of targeted experiments have been conducted to produce working components that can later either be integrated into the product or used as a reference when developing elements of the site. Among others, these include a simple text file reader, file existence tester and various AJAX responders. The knowledge and understanding built up by developing these should facilitate faster production and a higher quality product overall.

Scheduling report

Product development was initially planned to follow a spiral lifecycle model, with component implementation beginning with the content management system (CMS), progressing to development of the management and then customer user interfaces (UIs), before finally extension features, polish and documentation were to be completed.

The design stage of the content management system went much as expected. Design and implementation of the database took somewhat longer than expected due to less time than expected being available around the holiday period, resulting in this being completed in mid-December. The next phase would have been to implement the SQL queries required to access or modify products in the database. However due to the low fidelity of the designs for both the customer and management UIs (partially a result of the spiral based plan employed only looking at one component at a time), accurately determining the data requirements of these systems proved difficult. Combined with the lack of knowledge of PHP required to fully implement these components, this lead to the decision to focus on development of the user-end of the CMS before revisiting these back-end components, allowing time for more targeted experiments and PHP related lectures in the new year. This experience has demonstrated the need to more carefully evaluate the order of design stages for potential prerequisites and proved the value of allowing moderate slack around unpredictable work-times such as holidays.

The initial plan showed development of the CMS, customer and management UIs as independent, however the abstraction and separate development of common components, and the resulting time savings meant making significant changes to the plan to reflect this new, more accurate projection. An updated Gantt chart showing actual progress made to date and the adapted plan for the remainder of the project is included below. Tasks completed before the submission of the first interim report have been omitted to preserve readability as these occurred much as specified in the initial plan.


WEB-CW Revised
Dec 2014
Jan 2015
Feb 2015
Mar 2015
Apr 2015
Database
Interim report 1
Design
Implement
Test
Common components
Design
Implement
Test
Progress report
Write technical report
Write managment report
Interim report 2
Customer UI essentials
Design
Implement
Test
Managment UI essentials
Design
Implement
Test
Common extension components
Design
Implement
Test
Managment UI extensions
Design
Implement
Test
Customer UI extensions
Design
Implement
Test
Documentation
Write documentation
Final deadline