UNIVERSITY OF WESTMINSTER

5COSC024W SERVER-SIDE WEB DEVELOPMENT
2021-2022 homteq Project Brief & Initial Specifications

Case Study: homteq

homteq is a highly specialised retailer that offers a wide range of devices at the most competitive prices to make home and life SMART. homteq retails many smart products to the general public and to corporate business organisations.

The devices sold by homteq include smart security cameras that can be set up to check on your home remotely, smart energy systems that can be configured to control your home appliances from a distance, smart speakers that use voice-controlled assistants to provide a range of information and smart watches that can be used for a wide array of purposes, amongst many other smart products.

homteq is undertaking a project that aims to analyse, design, and develop a database-driven web application to retail and manage their products and process their orders online and has hired you as a server-side developer to drive this project.

The homteq Web app should be accessible by 2 types of users.

homteq customers should be able to use the Web app to browse through the range of products on offer, view the details of single products, temporally reserving products by adding them onto a basket, signing up to create a personal account as new customers, logging in as returning customers, and checking out to finalise their order with homteq.

homteq administrators should be able to use the system to maintain the range of devices on offer on the online platform i.e., to add new products and edit details of existing products such as stock levels and prices. Administrators should also be able to view and process the orders that have been placed. They should be able to specify whether an order is placed, ready to ship or has actually been shipped by modifying the status of the order. The system should record the order date and time and automatically calculate the shipping date accordingly based on the business rule defined by homteq e.g., shipping should occur within 2 days of the order being placed.

System Specifications

1) FUNCTIONAL REQUIREMENTS

Using the homteq project brief, elicit **user requirements** and **functional areas** for the homteq project for two types of users or requirements owner. Prioritise your requirements using the MoSCoW (Must, Should, Have, Won't) or the EDL (Essential, Desirable, Luxury) prioritisation techniques.

- A **user requirement** is a software capability needed by the user and to be performed by the application to complete a particular task and achieve a specific objective e.g., browse through products, select individual product, search for product, etc.

- A **functional area** is a grouping of user requirements brought together to define a specific component of the functional capability offered by the system e.g., product selection.

To elicit your requirements, continue filling in the table below (you will need to add extra lines).

R#	User Requirement	Priority	Complete CHECK	Functional Area	Requirement Owner
R01	Browse through products.	M		**PRODUCT SELECTION**	***Customer***
R02	Search for specific product.	W			
R03	Select individual product.	M			

2) USER INTERFACE DIAGRAM (UID)

Using a drawing tool like draw.io (available on http://diagrams.net), produce a **User Interface Diagram (UID)** for the homteq project.

- A **UID or UI Storyboard** is a series of illustrations interconnected with arrows to schematically represent the user interface components of a system and the navigation between these different UI components (i.e., the pages).

- A UID should clearly illustrate the following:
  - The **structure** of the application in terms of the functional areas.
  - An overview of the **content** of each UI component (each page).
  - The **navigation** between the UI components.

3) Entity-Relationship Diagram (ERD)

a) Conceptual ERD

Using a drawing tool like draw.io (available on http://diagrams.net), produce a **Conceptual ERD** for the homteq project.

- A **Conceptual ERD** shows the high-level structure of the underlying relational database architecture of a system.

- It should include:
  - **Entities**: objects or things with independent existence on which data needs to be stored.
  - **Relationships**: meaningful associations between occurrences of entities.
  - **Multiplicities**: number of possible occurrences of an entity related to a single occurrence of the other entity (participation and cardinality).
  - **Attributes**: properties of the entities that capture data values.
  - **Primary Keys**: unique identifier for each entity that is irreducible and has been selected.

b) Logical ERD

Using a drawing tool like draw.io (available on http://diagrams.net), produce a **logical ERD** for the homteq project by **mapping the conceptual ERD.**

- A **logical ERD** is a relational database schema that is implementable in SQL. It is mapped from the conceptual ERD according to specific mapping rules.

- It should include:
    - **Tables**: clearly defined tables mapped from the conceptual ERDs made of attributes (fields) and rows (records).
    - **Relationships**: meaningful associations between the tables.
    - **Multiplicities**: number of possible occurrences of a table related to a single occurrence of the other table (participation and cardinality).
    - **Attributes or fields**: column of the table that capture data values.
    - **Primary Keys**: unique identifier for each table that is irreducible and has been selected.
    - **Foreign Keys**: attribute added onto a child table to refer to the primary key of a parent table.