Domestic Helper

Co600 Technical Report

Icon

Description automatically generated

Saahil Shah, Samuel Prowse (sp774) & Tope Balogun

University of Kent, Canterbury, UK.

1. **Abstract**

‘Domestic Helper’ is a web-based application aimed at local people within the Canterbury area who want cleaning duties done without having to do said duties themselves. On the other hand, this is also a platform for people within the domestic cleaning industry to advertise the services they offer to people within Canterbury, having a different account type to that of users who want to book services. ‘Regular’ users can make bookings with any services they like (if they’re based within the same County the service is offered in). The user will receive details of any errors they come upon when attempting to make a booking guiding them on appropriate fixes. Otherwise, they will receive a notification of their booking being requested successfully.

1. **Introduction**

Our task was to produce an online based application that is aimed at beginners who are looking for a easier alternative instead of conventional methods of making bookings (i.e. telephone), it can be accessed on any device that has an internet connection, thus allowing many devices to access our website as opposed to just a mobile device. We wanted to make sure the process of using our application was as smooth and simple as possible, therefore, our application requires no specialized software to use (except a web browser, which most, if not all devices already have preinstalled).

The scope of this report will go into detail on all aspects of the project, this will include (but not limited to), initial ideas, technologies used, the design process, what went right/wrong. We started by designing an initial draft of the layout of our website, keeping everything color coordinated, and presenting them via or weekly meetings on Microsoft Teams. Each meeting we would agree on aspects we liked/didn’t like, discussed alternative options till we reached an agreement. After we had our application layout finalized, we then decided what programming languages we could use that would suit our project’s purpose, whilst making sure they could all work together in unison in order to provide more functionality.

After all requirements were finalized and agreed to by all members, we then decided to come to an agreement on what our ‘end product’ would look like, this was supported via our use of agile development, working on a specific section of the website before moving onto the next and repeating this process till we reached our goal. Lastly, this report will contain future improvements that we thought we would have liked to include in the website should we have had extra time to implement such ideas.

1. **Background**

**3.1 Market Research**

We began our research by looking at websites/applications that try to offer the same goals that we intend with our project. However, we felt that some of these websites looked bloated and offered features that we felt hindered the user experience. An example being a ‘one off home cleaning’ fee that was not based on other things that could affect the price (i.e. size of home, how many floors etc.) so a customer could be paying far more for a cleaning service than they otherwise would be been. So, we wanted to negate this by allowing employee to offer services that are more tailored to the user instead of being broad, and the user will be able to see exactly what is include in the service instead of trying to assume, this was one of our prioritized features since this could one of the decided factors if a user decides to pay for a service or not.

We then next decided to look on the websites for features that weren’t present on these competitors’ websites, but we thought would make a great addition to ours. The majority of website assume that you know how to navigate around the website, how you know how to book a service etc. But every user at some point is a \*new\* user to your website, so everyone will not know how to navigate. This could frustrate any visitors, and they may just exit the website altogether due to the navigation process being too difficult. Because we want our website to be used by as many people as possible. We know we had to take this into consideration, and to give them more than 1 option, we discussed this between ourselves and settled on giving both a summary step by step guide and a video walkthrough, to suit different types of users and to give them that choice.

1. **Aims**

Once all members of our team had an idea of what we would like to put into our website, we created a ‘product backlog’ which would include the full feature list that we would like to implement, these would colour coded ‘Green’ ‘Orange’ & ‘Red’ in order to indicate ‘Implemented’ ‘Partially implemented/implemented differently than originally intended’ and ‘Not implemented’ respectively. We took each task, and then checked if any tasks were reliant on other tasks on being completed first before it could be implemented. If there were tasks that weren’t dependant on other, we’d place these in another section, and prioritized them with the team based on each members strengths. We also needed to take into consideration any important updates to any languages we used and weather that would affect our project. Once we was aware of these factors, then we had a good idea of what type of website we wanted to build. We set our sights on creating a web applicated that would have a different experience for each user including many features that a conventional booking process would include over the telephone as well as unique features of our own, including (but not limited to): the ability to create many bookings in one ‘sitting’, review section, safe login, and registration procedures.

1. **System Architecture**

In this section, we will go into details in what technologies we have used to make this project and why we chose them, if there were alternative options and why didn’t we pick them. We will also touch on how does our project take into consideration the user experience (UI), did we feel it was appropriate? If not, why? Lastly, we will also discuss how we handle user data, where and how it stored, and if there were alternative approaches considered for these.

For some members of the group, the technologies that have been used will have not been used by themselves before so it will be a new experience for them. For others, they may be familiar with the technology and how it works but will have not used it as much as they have during this project. Because of this, all members of the group have enhanced their skills with many of the technologies we have used, making us better developers going forward.

Our project backend is based mostly of two technologies of the **LAMP** stack, namely MySQL & PHP. Our MySQL will host our tables which include details of any users/employees, bookings (whether active, completed or any custom booking requests) and reviews any users would like to leave. Because of the variety of these tables, some of them interact with each other which makes the overall website experience better.

The booking process focuses on managing bookings that any users will want to make/have made, they will select a service available on the services page (Service.php) via the book button, directing them to the booking screen section, in this section they’ll be able to chose a date and time they would like, alongside seeing any reviews that have been made for that service from people who have already had a booking of that service. Once a date and time have been chosen, they can confirm this ‘booking’, if there are no errors related to conflicts of interest (multiple bookings or a past date), the booking will go through and appear in the relevant employee and users profile pages (more on that below)

Profile page - which are then visible on their profile page under ‘booking requests’ section, this also involves the employee side as well since it will be visible on the respective employees profile page (who offers the service) under their ‘Bookings requested’ section, these are retrieve via a query to our MySQL database and displayed within an appropriate format (which will be discussed later on, as it involves some front end) in the users browser window. This technique also applies to when bookings are in progress and completed, each having their own dedicated section on both the user/employee profile page. This is also where reviews can be made/viewed, alongside cancelled any bookings that a user has already made.

The login/register system focuses on storing any details of any users/employees in an appropriate format in our MySQL database. This will include names, passwords, Date of births etc. Since some of this information is sensitive and could be of use to hackers in breaching peoples accounts we have to store these appropriately, we do this using a technology called **password\_hash(),** which is a strong one way hashing algorithm function in PHP, we chose this as an alternative over other algorithms such as MD5 because string can easily be used to reverse lookup over the internet, breaching security further. There are two separate register and login systems, one for users, one for employees.

To pass our information safely from the browser to the MySQL database, we are using MySQLi prepared statements, this allows use to create a template SQL statement that don’t explicity have the values of our inputs in the statement(they are replaced by ? values) it will execute the results of this template (without executing it) and then later on bind the values to the template and then execute it. This is useful against SQL injection because our parameter values are not derived from the original statement. So SQL injection cannot happen that way. In order to further enhance the security of our inputs, we must sanitize them to prevent text input from being executed as code and to only appear to our database are as the appropriate format.