Domestic Helper

Co600 Technical Report

Icon

Description automatically generated

Saahil Shah(ss2357), Samuel Prowse (sp774) & Tope Balogun(tb520)

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 an 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 are 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 weekly meetings on Microsoft Teams. In 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 to provide more functionality.

After all requirements were finalized and agreed to by all members, we then decided to agree 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 on to 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 on 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 is a ‘one-off home cleaning’ fee that was not based on other things that could affect the price (i.e., size of the 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 the 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 included in the service instead of trying to assume, this was one of our prioritized features since this could one of the deciding 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 the website assumes 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 consider this and 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 on our website, we created a ‘product backlog’ which would include the full feature list that we would like to implement, these would be color-coded ‘Green’ ‘Orange’ & ‘Red’ 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 being completed first before they could be implemented. If there were tasks that weren’t dependent on others, we’d place these in another section and prioritized them with the team based on each member's strengths. We also needed to take into consideration any important updates to any languages we used and whether that would affect our project. Once we were 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 detail about 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 is 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 on 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 choose 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 bookings are visible on their profile page under the ‘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 retrieved 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 when bookings are in progress and completed, each having its own dedicated section on both the user/employee profile page. This is also where reviews can be made/viewed, alongside canceled any bookings that a user may have 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, dates of birth, etc. Since some of this information is sensitive and could be of use to hackers in breaching people's 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 the 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 us to create a template SQL statement that doesn’t explicitly 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. 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 in our database are as the appropriate format.

User interface design – We also considered the user experience as the better the interface is the better experience it will be for the user. Considering our audience, we made sure the interface design had similar features to the existing interfaces, as our purpose is to provide service, not to teach users how to use the website so keeping it like others will allow normal users things that they will expect with websites. With the assistance of Norman’s Seven Principles, enabling us to align the purpose of our website for the Users, such as *‘Make things visible’* Which was done via Font decisions as well as the color of our font, making sure it stays consistent for aesthetics, so the users can see it. Furthermore, most of the main information is centered in the main window of the website, our logo is consistently on the top left of the page this is justified via research, an eye-tracking study by Eyequent, says that the upper-left corner gets the most attention when a user visits a website. This was further justified by our own university using that structure to make sure any user is aware of where they are. We also made sure to enable a link on our logo as this is expected on most websites. Colour choices; We kept a consistency of 2 colours Green & Blue, as we know colour schemes have an important impact on the relations between the user and the computer, so we made sure we picked two primary colour just to keep it simple. Another technique we implemented was to make sure the colours we assigned stayed consistent on some sort of information. In the image below, is our Graphical user interface, application

Description automatically generatedservice page, having the colour green assigned as the table header is consistently shown through the website. Furthermore, our buttons are a blue colour consistently throughout the website, this will help the users expect or have a first base assumption of the purpose of a function or feature via the Graphical user interface, table, website

Description automatically generatedcolour that it is consistently on.

For our buttons, we added was the CSS style of *hover*. This feature will bring clarification to the users when they hover across a button that can be interacted with.

Graphical user interface, text, application, chat or text message

Description automatically generatedShneiderman’s Eight Golden Rule was another guideline used within this website, taking into consideration when designing. *Offer Informative feedback* to make sure there is some sort of feedback given to the user when they perform an action. A simple example of this is an error message. The image below is just a simple example of how we integrated it into our website. The scenario follows: The user mistyped their password and our website gives immediate feedback to let the user know the type of error that occurred. We also took into account that users wouldn’t appreciate poor delivery in how the dialogue is addressed to them. So *Incorrect login details, please try again* was the best of both giving clear dialogue on what went wrong as well as mannerisms.

An additional feature we added, was in our viewService page which is the page that you get taken to after you’ve made a booking. This can be seen as a pre-receipt of everything you have confirmed before making a booking by confirming the date and time you’d want this service to happen. The image below displays the additional feature of the recent reviews. Although we already have a place where we display the reviews. We decided to have it side by side with the view service as before you want to make sure this is the service that you want, you’d also get an additional review from someone else of the same service to reassure your decision. We got this inspiration from Amazon, as before you purchase an item at the bottom, they have a variety of different reviews from different customers so you can make sure this is the product you want.

**6. Problems**

A few minor problems we encountered early in our projects were waiting for a raptor set up, as that caused a minor delay that stopped some of our implementations.

During our implementation maintenance was moderate to high. As every time implementation was made it will interfere with some of the sessions made through SQL. This was expected and discussed at the beginning of our project, we assigned throughout the group that after a sprint, they would do a website run-through, by thoroughly running the website where if a problem is found we’d add that to our next sprint. This cycle of a process allowed us to be aware of when a problem has occurred, mitigating the scale of inevitable bugs that would occur as our project progressed.

However, they were problems that we couldn’t overpass: As in our view service page, within our bookings, we had a calendar feature that make it look both aesthetically and beneficial for the users. The feature that we wanted to include was to block the dates that were already booked, this way the user would know that this date has been selected (*Informative feedback*). Although this was a minor feature, we wanted to include this as this would benefit the user however it wasn’t possible.

**7. Testing**

As we were in the development stage, after every sprint, we frequently carried out tests on our website to ensure that the website runs smoothly without encountering any bugs. We carried out this test regularly just so that if we encounter a problem, we know exactly what part of the code needs fixing instead of scheming through a large block of code and trying to check where we went wrong. During our implementation, one of our tests included logging in to the website using the same details but on 2 different computers, as this was an issue of security. The website also allowed registrations of employees under the age of 18. we then had to create a process that will not allow the user to do log in from 2 different computers and prevent the registration of employers under the age of 18. Another test that we carried out was booking a service. We added all information apart from the date and made a booking. The booking was successfully made even though the date field was empty which would not be ideal. We had to fix this by making the date a required field to input and we added an error message/ alert message if the date field was missing.

**User Testing**

After most of our implementation was done, we then decided to carry out tests on the user instead of carrying out the tests on our own just to see how the website is used from different perspectives. We handed the website to 5 different individuals and had different tasks laid out for them to do i.e. Make a booking, log out, make a registration, etc. Whilst they did each task, we stood beside them and observed the way they navigated through the website while performing those specific tasks. We then wrote down notes on what we felt the user was struggling with. We then also asked the specific individuals to rate our website based on how easy it is to use and how easy they were able to carry out those specific tasks on a scale of 1-5, 5 being the easiest. After all those notes that we took down, we noticed a few things that we needed to change. One of the main issues the users encountered was when a user tries to create a custom service, it takes them to the appropriate page but with the employees’ dashboard instead of the users’ dashboard. This seemed to have created confusion for the users. We then fixed this by switching the navbar to the user one. Another frequent issue encountered at the users’ end was after the service was carried out and a review was made by the user, they could not go back and view the review. We fixed this by adding the review to the profile page of the user. Finally, when the users were booking the service, they commented on the review. They said that the rating of the service which is displayed is not enough to see how good the service provided is. As a group, we then decided to add the most recent reviews to the view service page so the user can see the rating of the service, as well as other people’s feedback on the services, are. Another thing that we noticed was that when the services are listed, to view the services offered in a particular area in Kent the user would have to type in the name of the area, for instance, “Canterbury” which would then display the services offered in canterbury only. This was quite long, and some users would cause some spelling mistakes as well so to make it more user friendly we decided that instead of typing the name of the are in Kent we would have the areas already listen below in a drop down so the user can just click on the are they want and filter the search. We found that this made it much easier for the users to use the system. We concluded that the tests we carried out were sufficient and we felt that we thoroughly tested every key aspect of the website. At the end of our implementation, we carried out 1 final test after we met with the supervisor for final refinements and final touches.

**Conclusion**

As a group, we think that we managed to successfully reach our final goal which was to create a domestic helper system that provides different services. In comparison to other similar systems, we think we met all the criteria and requirements for a successful domestic helper system. We did of course face many difficulties along the way, but we never lost the motivation and dedication to reach our final goal. Unfortunately, we lost one of our group members in our early stages of development and one in the middle of our development. This, therefore, increased the workload on the 3 group members that were left. Truthfully, it affected us greatly since apart from the group project we all had other modules to focus on with the assignments, but we made it our primary goal to focus and prioritize our project so that it can turn out as expected. We used GitHub as our workspace and store our files. This was very simplistic and very convenient as many people could work on the project at the same time without interference. Managing GitHub was very easy as once any of our group members have done working on the project, they could simply just use the push command to update the files on GitHub and commit a message to show what files were recently updated and what progress is made. In terms of testing, we feel like we have successfully and exhaustively tested our system and every aspect of the website. The testing, especially the user testing helped us very much in making our website easy to use for a novice and user-friendly. We had a very systematic method in terms of working on the project. After every sprint we would list down the tasks that need to be done and then we split the tasks equally amongst us. If one of our group members was unable to perform a certain task, he would let the others know and we would all work together to find the solution. We all worked as a team and made sure that we got the tasks working before the sprint comes to an end.

This project taught us that teamwork, punctuality, and communication are the main objectives for carrying out a successful project. Our mode of communication apart from the sprint meetings on teams was WhatsApp. If a certain task is completed or is not clear, one of the group members would simply send a message on the WhatsApp group and we would all look into that matter. This was a very easy and effective way of communication as every group member had access to WhatsApp. As though we feel that we made a reasonably appealing website, given more time we could have made it even better for future development and would have also shown the website to different companies around London and gotten feedback from them which would be ideal. There were also certain features we were not

able to implement such as the “remember me ” feature due to the time scope but given more time we would have been able to fully implement it.

All the group members were punctual apart from when there was a slight inconvenience where our group meetings would clash with our university timetables, the group member would simply just send a text on our WhatsApp group, and we would reschedule the meeting for a later time

We would also like to thank our supervisor for this project, Professor Yang, for helping us to successfully complete this project. She constantly gave us advice and key points to make our implementation desirable. She would also push and motivate us thoroughly throughout the project.

In conclusion, we really enjoyed working as a team to complete this project, we learned many new things and advanced our knowledge in the field of computer science.

References:

* Eyequentstudies- https://www.eyequant.com/