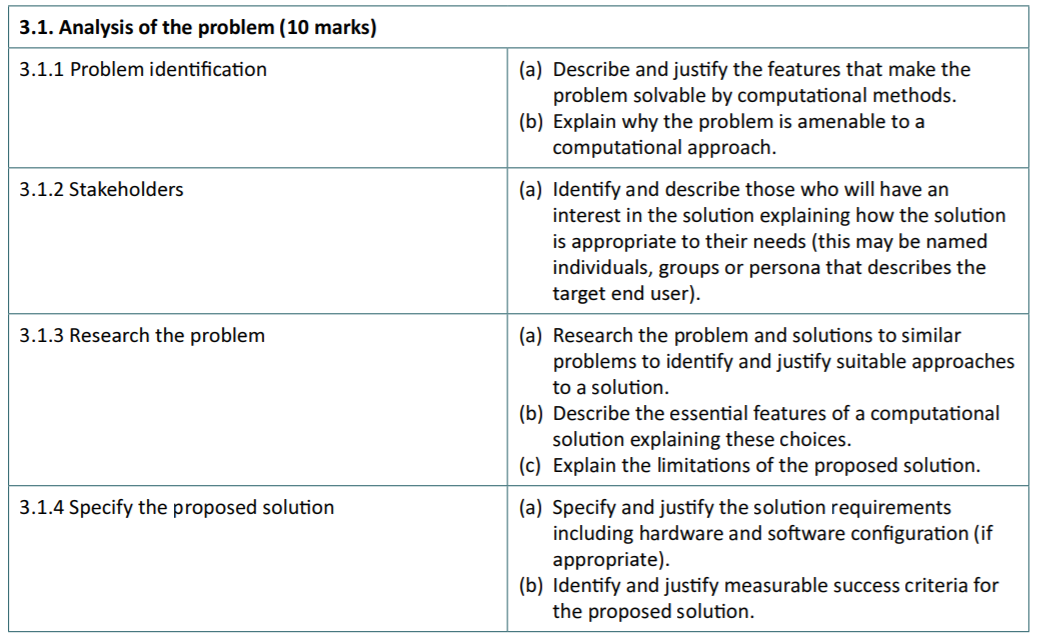
­­­Analysis of the Problem



## Problem Identification

Staffing Database and Timetable: currently for those at work their roster or timetable is scheduled by hand and any gaps in the timetable are filled by people doing extra work. This is not an efficient way to staff your shop as often you have more staff than you need or too little staff, this leads to wasted money.

The shop also requires someone to spend time creating the timetable for the week further costing them more money. A computer can more easily and quickly complete this, with additional features. This problem is ideal to be solved using a computer program as it is a repetitive task and needs to be solved very fast as it is created each week or day. The computer can also automate tasks such as distributing the staff’s timetable to each of them or letting them know of timetable changes.

The problem can be solved easily using a computational approach, as when I break it down many of the components are ideal for such an approach. The storing of data is ideal for a database, and as a large amount of data is stored a relational database may be required. The auto creation of hours would be solved using an algorithm. Such algorithm would most likely have to iterate many times and improve on itself each time creating the best timetable.

## Stakeholders

### Client Company

The company will most likely be the most interested in the solution as it will most likely save them money, something many high street shops are in desperate need of. It also allows them to future proof their staffing operations. Today more and more operations need to turn digital to keep up with the ever-changing world.

### Staff

The staff will have mixed feelings towards this program, as it does allow them easier access to their hours as well as making it easier to change your shift or contract, but they will not like the fact that the program accounts for their 50% flexibility. Meaning if you work 20 hours a week the computer can change by up to 10 hours each week, many staff may also oppose the change for the sake of disliking change.

## https://www.sdworx.co.uk/images/Software/008_-_Self_Service.pngProblem Research

### Existing Solution: SDWORX

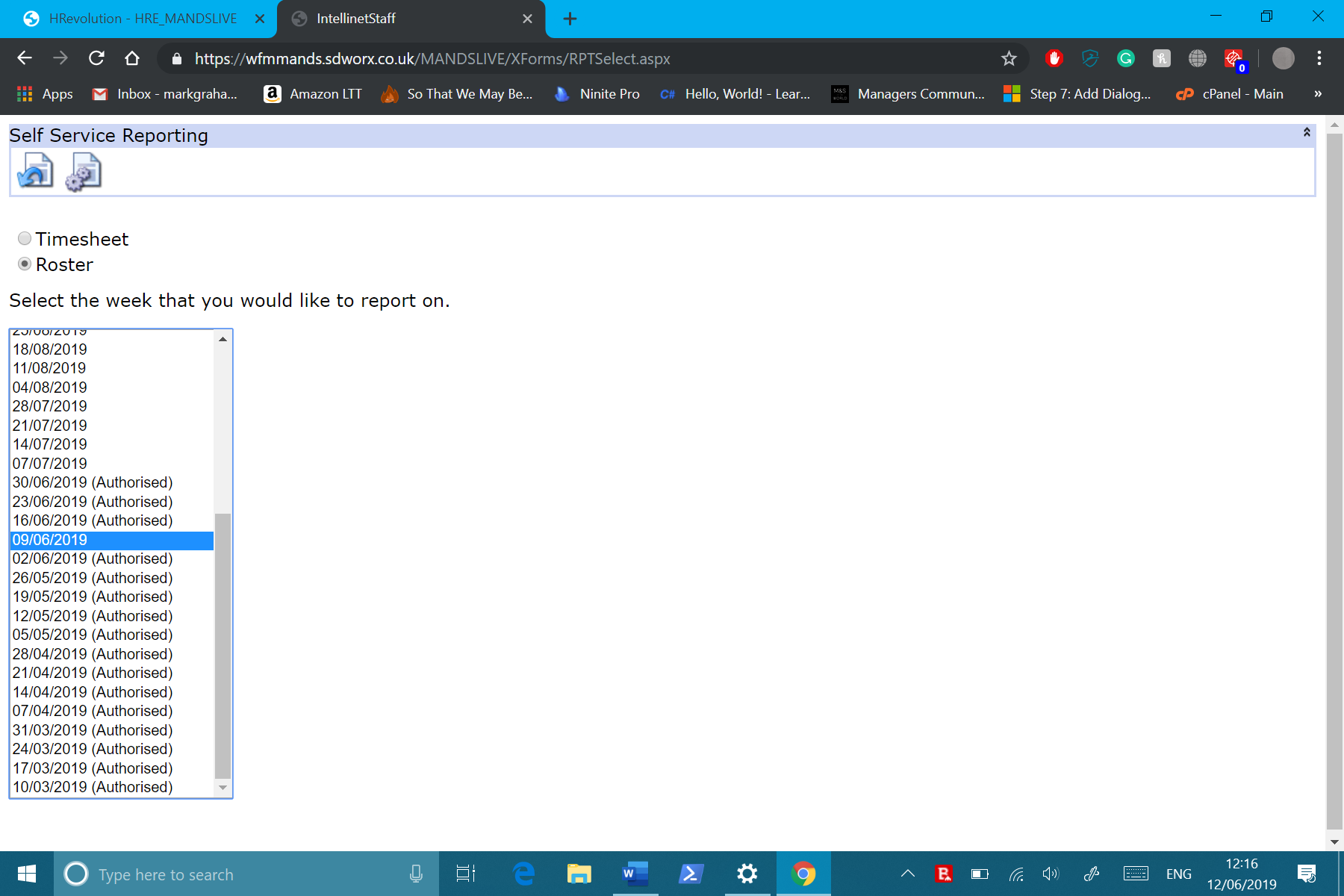


Figure 1 - Allows staff to view their personal timetable for all weeks

Figure 2 - The admin interface, designed for non-IT staff

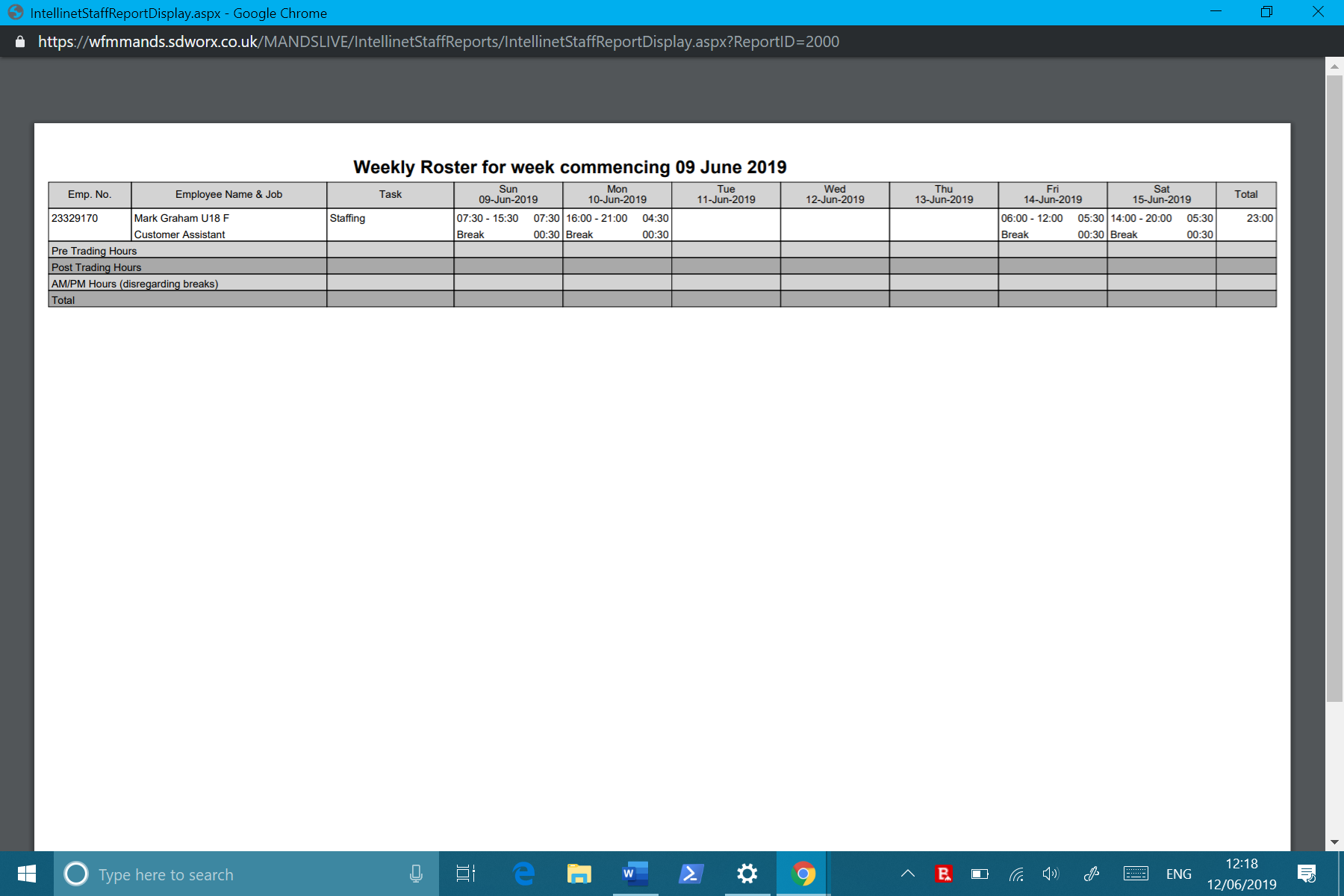


Figure 3 - the timetable for the staff in PDF form

#### Positives:

* You can view previous weeks timetable, which is good for people who want to check their pay versus their hours worked
* It provides a personalised weekly timetable, with information on breaks and the task they are performing
* The admin can easily change the staff’s hours for each day, and before completing it checks their hours against a set of rules.

#### Negatives:

* You cannot change hours from anywhere online
* Nor can you view an entire month or fortnight at once

Overall SDWORX is an industry standard for features and design, it allows the staff to easily on the computer to view their hours and save or print them as a PDF. It also allows managers to view the entire staffs timetable, simply make changes by clicking and dragging. It also allows the manager to print these as they’re in PDF form, but its modern features are lacking such as interaction with the timetable is not possible and the panel can only be accessed on computers. It features no notifications or alerts and relies on a manger to input all the staff’s hours and manually make changes.

### Existing Solution: Excel



Figure 4 - Using excel to record your employee's schedule

#### Advantages:

* Cheaper than an online based solution
* You can more easily print off the schedule

#### Disadvantages:

* No online integration for staff to view their timetable at home.
* No flexibility to change the design
* Does not scale well for companies with many employees i.e. more than 5

In conclusion excel is a great way to staff for small companies as it very cost effective and allows for more modern designs such as above, but it features no interactive elements and cannot be upgraded.

### Secondary Research Analysis

From my research into existing solutions, I have learnt that more interactive applications work better, and can more easily scale for larger companies. They have also shown that there is a lack of mobile apps, and flexibility in the current solutions out there.

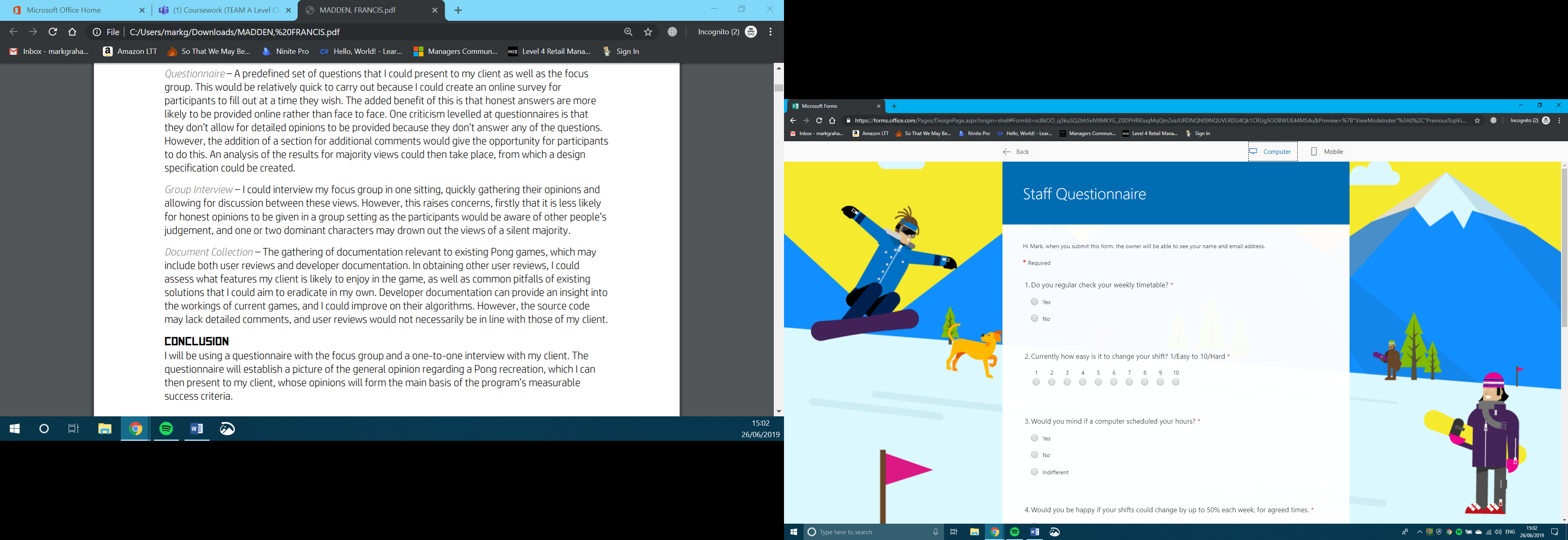
## Data Collection

### Questionnaire

I will use a questionnaire to obtain the views of the staff, they are key to the development as they are most affected by this new system. It would include a predefined set of questions that have previously decided answers, this stops the responders from deviating from the question. The drawback of a questionnaire is that it prevents a two-way dialogue with the staff. The questionnaire will be anonymous to allow the staff to not hold back their opinions.

### Interview

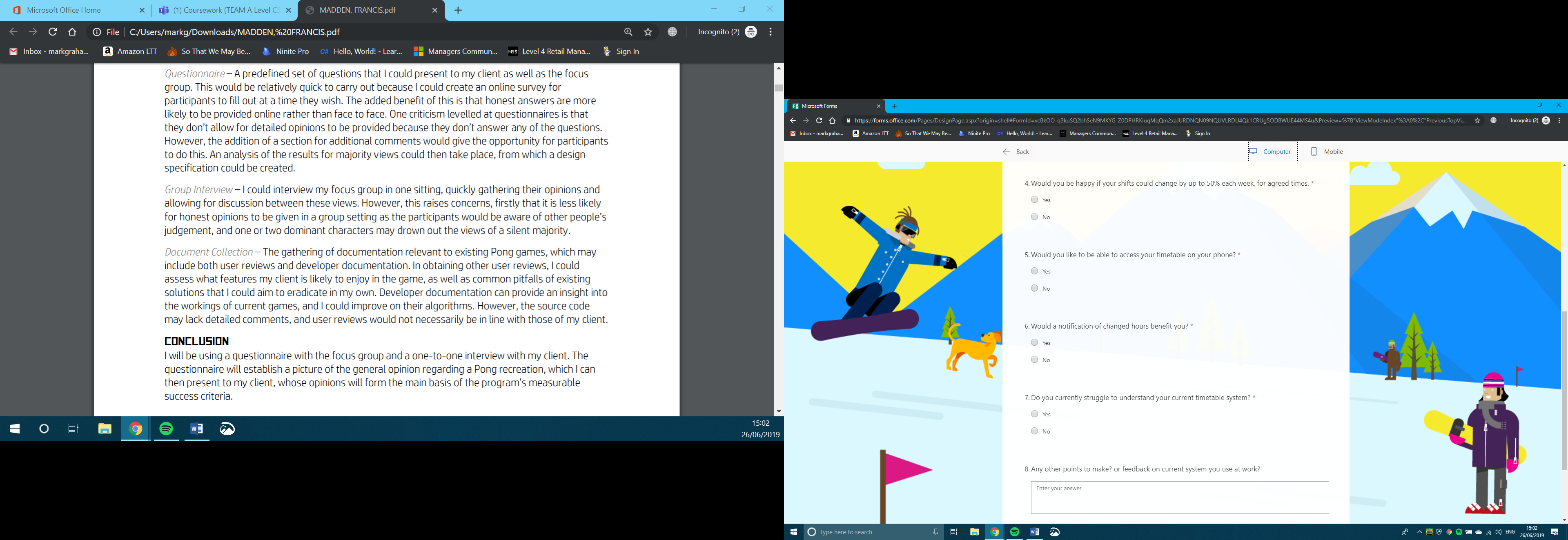
I will conduct an interview to gain some more insightful responses from both the targeted client, and staff that work for the client. This is to allow a two-way dialogue to gauge a better idea of features, and potentially improve on ideas through this conversation. It also allows be two ask some more specific questions that may warrant lengthier answers or answers that lead to further questions. I shall also ask questions targeted to that user, this is to understand as whether customisation for users is important to them.



This is a simple 50/50 questions to check the staff’s opinion

This is used to evaluate the effectiveness of the current systems, and by how much I need to simplify the interface

This question helps me gauge an insight into the usefulness of such system, and whether most of the workforce would adopt it.



This is a simple 50/50 questions to check the staff’s opinion and allows me to evaluate current staff’s opinions

This is a simple 50/50 questions to check the staff’s opinion

This is a simple 50/50 questions to check the staff’s opinion

This is a simple 50/50 questions to check the staff’s opinion

## Interview Questions w/ Richard

### What feedback would you give about your current staffing system?

I’m asking the question so that I can further gain insight into the effectiveness of current staffing systems, this also allows me too to view the understanding by the company of their staffing software.

### Do you think the adoption of technology at your current workplace is slow?

I am asking this question as my system relies on the workforce being able to use computers to view the timetable both at work and at home.

### How would you like to access your staff dashboard?

I am asking this question to gain some first-hand feedback from the company as to what devices they use to manage their staff. This allows me to decide

### Would having the hours only online be a benefit?

## Transcript

Mark: What feedback would you give about your current staffing system?

Richard: It mostly does its job; we think of it to digitalise our timetable. There are a couple of things it’s missing though, such as an option for staff to swap hours between themselves, and the option to view it on anything smaller than a PC.

So, do you think the adoption of technology at your current workplace is slow?

I would say we are catching up, it’s not the greatest in the industry, but we are above others and with the employment of youngers workers we are progressing rapidly.

What do you think could be done to improve this?  
I believe that by working more closely with our IT department, and more feedback being given we can work hard on improving IT for all, not to mention IT will advance itself automatically.

How would you like to access your staff dashboard?

I think most importantly it needs to work on the PC’s, but we have small smartphones and lately these can do more and more, so it would be great if I could change someone’s shift or view their timetable on this device. The feedback from staff is also that they don’t have a PC at home to check the page, so it would be better if they could access it on their phone or tablet.

Would having the hours only online be a benefit?

I think we’re a long way from being paperless, it nice to have the timetable for the day printed out so we can write on it and make changes, but with more and more smart phones being in the hands of our colleagues we are close to being able to do this. Ideally though we can choose between both as the more options there are the more flexible it is for us.

## Results

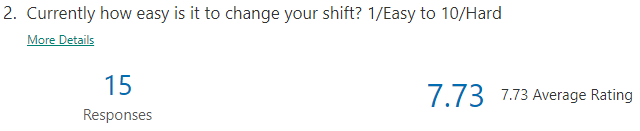
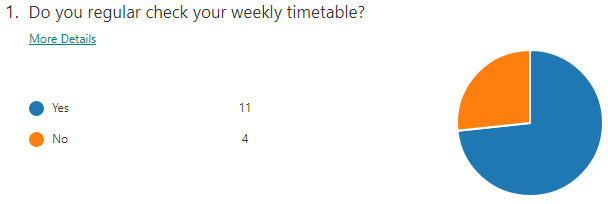
### Primary Research Analysis

After asking these questions, I have analysed the results for further research into what people require with a staffing system.

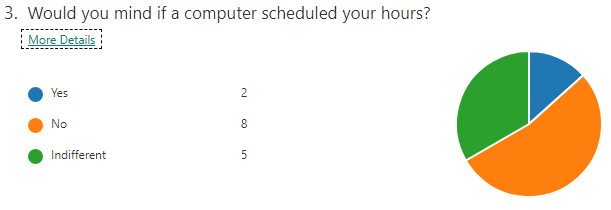
### Interview

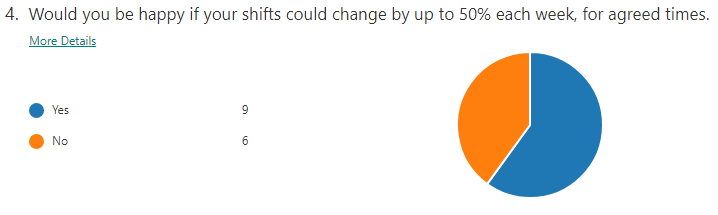
The key points identified in the interview are that more flexibility is needed, this being on access to the dashboard on a wider range of devices, and more choices on how to deliver reports or timetables. We can also see the demand for such a system is required as Richard says that the industry needs to be pushed forward with IT. He also says that many don’t have access to desktop PC’s at home, therefore the UI should also be easy to navigate on such small devices. This simple design also allows those who aren’t tech savvy to easily use the software.

#### Survey

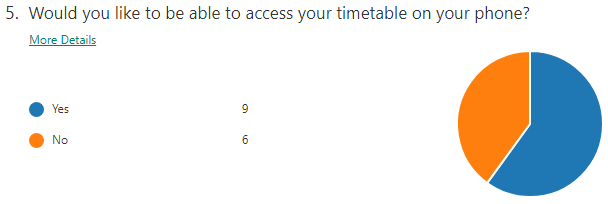
This shows that many people are checking their digital timetable, this also means that accessibility of said timetable needs to be a priority.

This shows that the current technology is hard to use and therefore we must take this into account when designing the application.

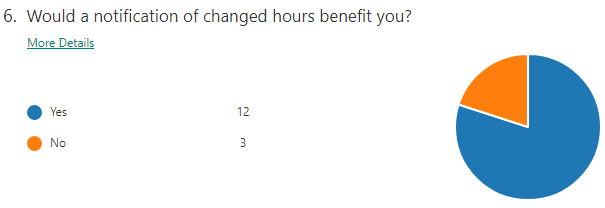
The results from this are understandable, as a computer scheduling the hours is very similar to a human doing so. Therefore, I can draw from this that such change will most likely pass without much conflict.



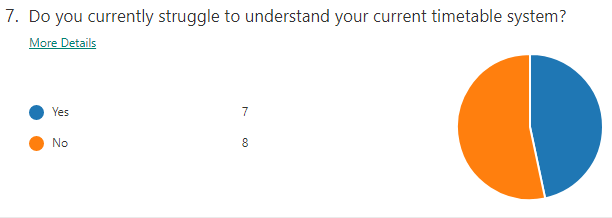
This shows that the feature of changing staff’s hours can be implemented.



This also shows that I must implement some mobile features, as many people desire such.



I can see from this question that people overwhelming want notifications of changes to their hours.



I can see from this that the current systems do work, but not for everyone, therefore I need to work on improved access for all ages.

# Success Criteria

# Success Requirements

This is a list of requirements for my application, I have sorted them into three categories, essential, preferred and extra.

This is based upon my research into many different options and considered the feedback from the client and the staff.

## Essential

1. Ability to add a staff member to the team, with their hours
2. Ability to create rules/ laws about when staff can work
3. WebUI to manage and view the dashboard
4. Both a mobile and desktop version of the page
5. Different permissions for different staff to change various options, i.e. manager can change hours unlike general staff

## Preferred

1. Ability to create PDF based on various rotas/ timetables
2. Login ability, with hashing of password
3. Auto generation of rota for the week
4. Option to change no. staff in shop for each hour
5. Ability to email changes to staff’s hours
6. Ability to declare when the staff member is flexible, and accept changed hours
7. Option to view hours based daily, weekly or monthly

## Additional

1. Ability to auto assign extra hours based upon preference
2. Budget calculator based upon staff pay
3. Adding comments to the timetable, such as when breaks are etc.
4. Since now completing both primary and secondary research I have drawn conclusions and an outline for the requirements.

## Hardware:

Client Device:

|  |  |
| --- | --- |
|  | Client PC |
| CPU | 1 GHz 32bit or better processor |
| RAM | 1GB |
| Storage | 50KB |
| Screen Size | 10.5” for desktop site, 4” for mobile |
| Inputs | Keyboard (physical or touch), mouse or touch input |
| Connection | 10Kb/s |

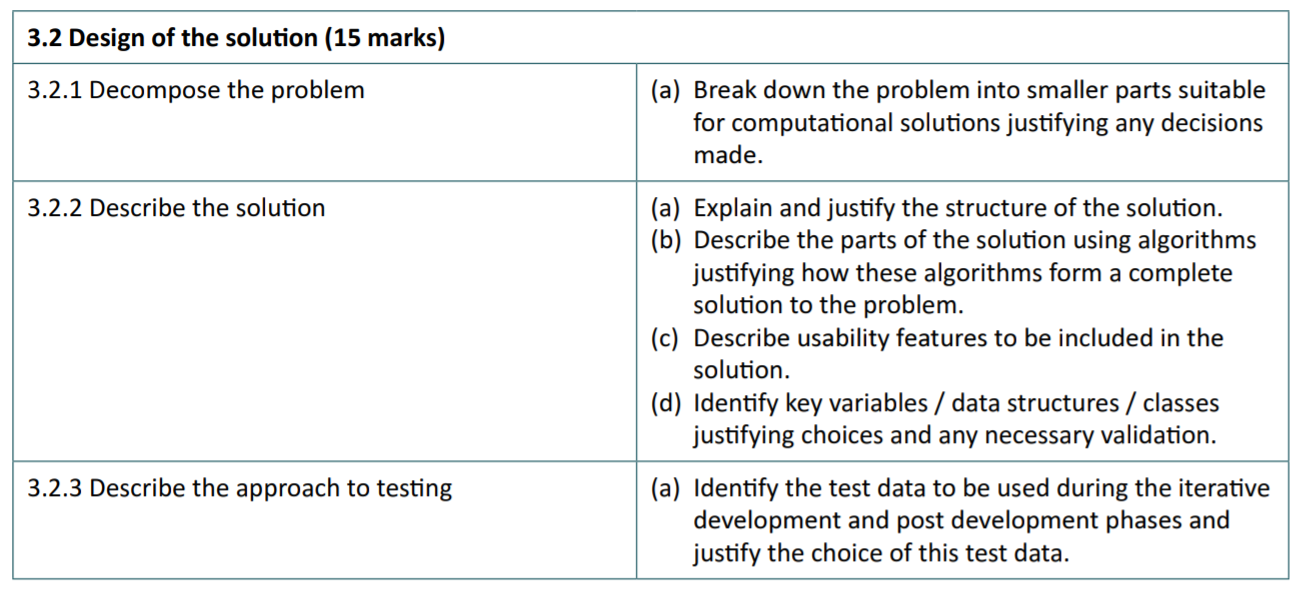
These are based on support for Windows 7 or Android 8, and their minimum specification. The screen size support is based on common usage for browsers on various devices.

## Software:

Any browser that supports HTML 5.

These are based on the support for browsers, and their support for PHP V5.5 as well as HTML 5.

Design of the Solution



# Decomposition of the Problem

I have broken down the problem into its several different parts, this is so that later separate components can be developed and testing individually.

# Database

To store all my data, I am using an SQL database. Within this database there are several essential tables. There are many benefits to using a database, such as automatic backups. More efficient file storage, and the system can be easily imported and exported as a .sql file.

## Users:

|  |  |  |
| --- | --- | --- |
| Column Heading | Example Data | Explanation |
| staffID | 00001 | This is the unique ID for the Staff member. It is used across many other tables as a foreign key. The system does not automatically generate the key as the company may have a method for generating these ID’s. |
| name | Mark Graham | The name of the employee is stored here. |
| passHashed | 30875438957958 | This will be a hash of the users password, it is stored in a the hashed form so that a data breach does not reveal uses passwords. |
| email | mark@gmpauto.co.uk | We will store the staff’s email so that they can receive emails about new, or available hours. |
| positionID | 0 | The position ID is a foreign key for the position of the employee, such as customer assistant. You will see later how this number translates into understandable data. |
| accountLocked | FALSE | If the users account is locked this may be due to being suspended from work, or that they have entered the password incorrect to many times. This can only be unlocked via the admins account. |

## Positions:

This is used so that the users can create their own positions, this means if the company has many different roles that they want to be reflected on the roster, they can do such.

|  |  |  |
| --- | --- | --- |
| Column Heading | Example Data | Explanation |
| positionID | 0 | This is the primary key for positions, it is generated by the system. |
| posName | System Admin | The name of the position. |
| perms | 100 | The permission level, this will be used for restricting access to certain pages. |

# Web UI

This is the layout of the website, as this is the frontend system, I have separated it from any of the backend systems. The website needs to be optimised for devices with small screens such as phones or larger devices such as laptops. It also needs to be lightweight, so it loads quickly. Finally, I need to keep it simple so that everyone can understand the system easily.

The mobile version of the site will always be on an auto redirect to a different subdomain, this stops any older devices not properly requesting the mobile version, or utilising CSS scaling.

## Initial Sketches

[insert initial sketches for mobile and desktop]

## Digital Format

# Login System

This part of the program is where the user logins in and can update their password. I will also include adding users in this section as it is relevant.

## Design:

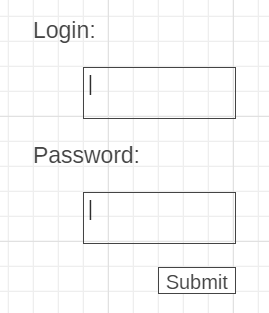


Figure 5 - Login Interface

This is the login interface, it is very simple, the password input will be masked so that you cannot see it. The page includes the same standard elements as above.

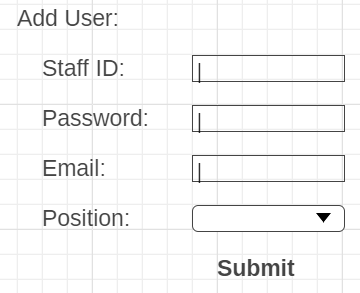


Figure 6 - Adding a User

Only admins can add users, the permission level will be set at 50 out of 100, as described in the permission database. The design is simple and easy to follow so that little training is needed to use the program. When the user presses submit some data validation is performed to ensure that the input data is correct.

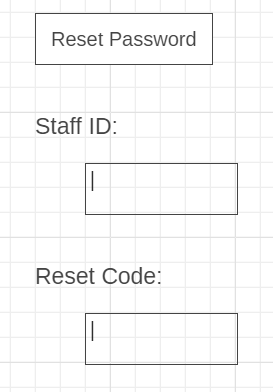


Figure 7 - Reset Password

This interface allows the user to reset their password if they have forgotten it. They start by entering their staffID, then they receive and email with a reset code, when the enter this they can then reset their password.

## Pseudocode:

### Login Form:

staffID = get(staffID)

password = get(password)

SELECT staffID, password FROM USERS

If staffID == staffID AND password == password{

Login = TRUE

GOTO home.php  
}

### Reset Password:

staffID = get(staffID)

SELECT staffID, email FROM users

IF staffID == staffID {

sendEmail(email, randomInt(1111, 9999)

}

resetCode = get(resetCode)

if resetCode == randomInt(1111, 9999){

allowReset = TRUE

}

if allowReset = TRUE {  
 password = get(password)

UPDATE password=password WHERE staffID = staffID

GOTO login.php

} else {  
 Output Error!

GOTO login.php

}

### Adding a User:

staffID = get(staffID)

password = get(password)

email = get(email)

position = get(position)

# Viewing A Daily Rota

## Design: