DUBLIN INSTITUTE of TECHNOLOGY

Institiúid Teicneolaíochta Bhaile Átha Cliath

Android App and Website to aid in Health Care Management

Project Report

DT265 Higher Diploma in Computing

Student: Ashling Gleeson - D13127846

Supervisor/Second Reader: Andrea Curley

School of Computing

Dublin Institute of Technology

9th June 2015

Abstract

This project comprises research and implementation of an android app and a website. It is has been developed aiming to be of use for management in home health care companies (website) and for lone care workers (app). The completed application will be a communication tool and a scheduling tool to aid in company management. The app is created using Java for Android in the eclipse development environment. The website is created using PHP, developed in Notepad ++. The database is MySQL and stored on the server.

Declaration

I hereby declare that the work described in this project report is, except where otherwise stated entirely my own and has not been submitted as an exercise for a degree at this or any other university.

Signed

Ashling Gleeson

Acknowledgements

I would like to thank my supervisor Andrea Curley for her guidance and support with this project.

Table of Contents

1. Introduction

The aim of this app and website is to aid in communication between staff members and management in home health care.

Through researching technologies available for this user group and through communicating with management at home health care companies, it is clear that there is a niche in the market for this type of app. According to Hagglund, Koch and Scandurra (2010)

"A lack of adequate information access and communication tools hampers healthcare professional's work, sharing information and coordinating work within the care providing team in particular"

This app aims to provide a useful tool to improve this sharing of information and coordination of work in a health care team environment. This in turn aims to improve the daily lives of those who require home assistance.

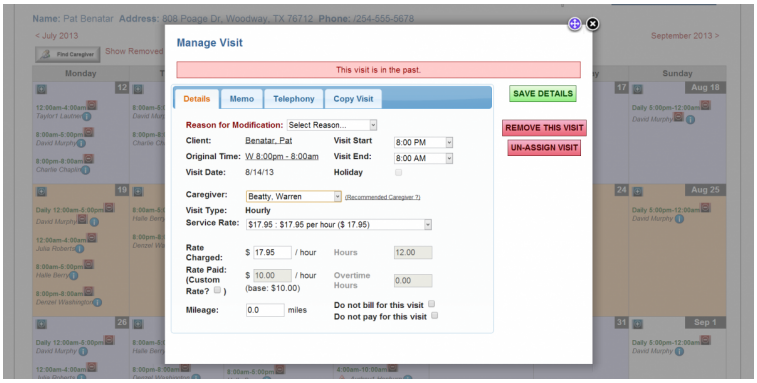
The app aims to consist of a main menu with 3 links- scheduling, communication, and edit profile. Scheduling links to care workers monthly work schedule. From here they can view their daily schedule and associated information. Communication links to a messaging activity where the care worker can speak with their boss on the app. Edit Profile allows the care worker to edit their profile name and picture.

The website is for the manager in the office on desktop. The manager enters login details and is brought to a main menu where they have two options- to update schedules or to communicate with the staff.

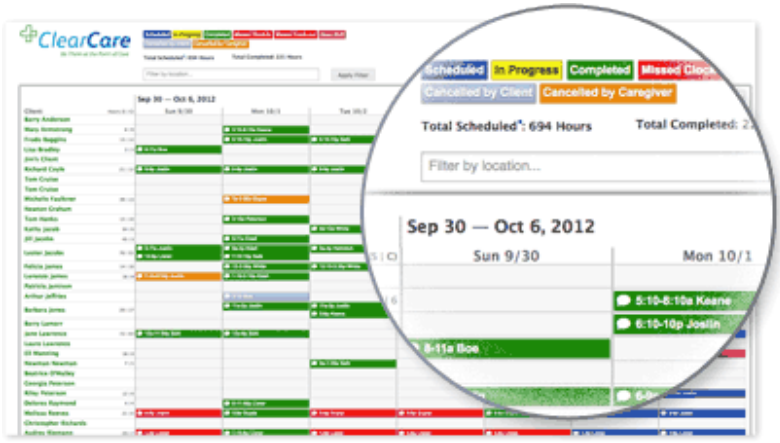
1.1 Overview

There are many websites/apps out there aimed at health care. Compare Home Health Care Software (2015) provides a list of apps considered the most efficient in the home care system. Based on number of reviews recommendations, the top 5 software's are as follows:

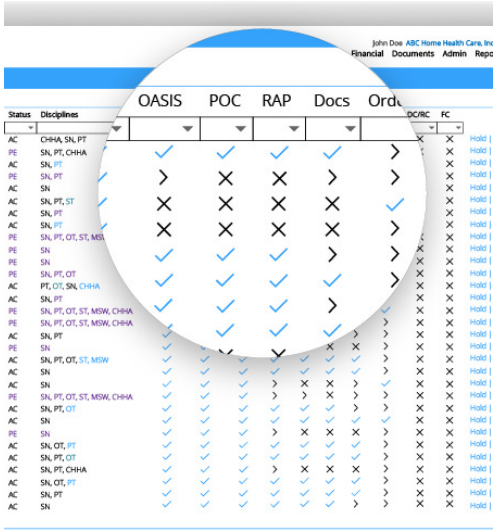
Axiscare



Clearcare

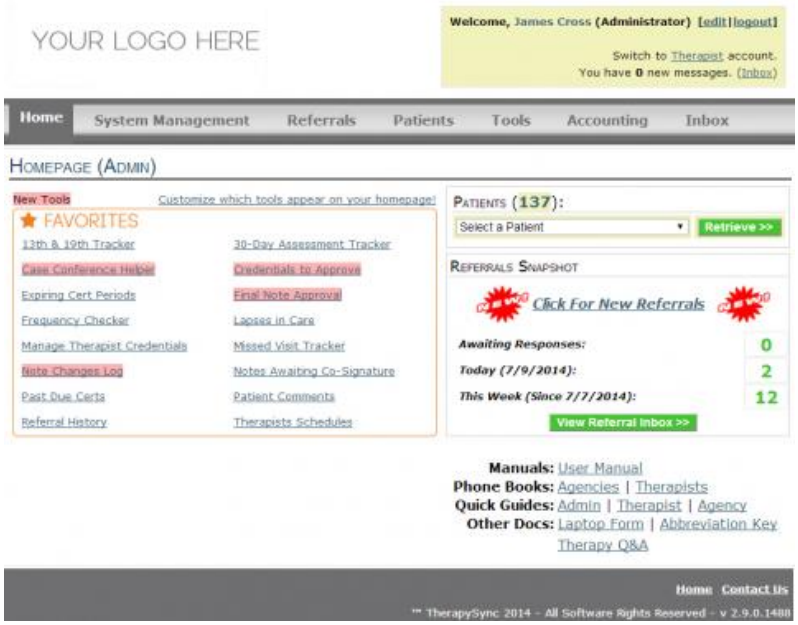


Fasternotes



Axxess Agencycore

TherapySync



Through researching these software's it has been noticed that only one of the above has a mobile application(*Axxess Agencycore)* This is not to say there are not mobile applications available to the home health care industry. This will be further reviewed in section 2.3. However, considering the information above and from speaking to companies and individuals working in care (4.3) it is clear mobile applications are not as popular as they should be in this market. Developing solely an app was considered, however after research it was decided a desktop version would also be developed. This would take into account how home health care companies in Ireland work currently.

A reason this project was chosen is also due to my own personal experience with home care. A family member of mine requires home care, and a sibling of mine works as an operations manager for a private home care company in Ireland. This project has been influenced through gaining knowledge of my sibling's requirements, researching the lack of software available in this field along with a personal interest in home care.

From speaking with home care workers, they are required to use desktop to view their schedules for the day. When it comes to querying with a manager, they usually text or call. In front of the patient these calls can be seen as unprofessional, and if the manager is not always by their phone they may not be answered efficiently. From the manager's point of view, a software enabling communication with the carer on the field would enhance efficiency.

1.2 Project Objectives

+ Mobile App

An initial interface- the care worker is asked to enter name. Following this on opening the app, the care worker will automatically enter to the main screen.

Main Menu interface – containing links:

- Scheduling
- Communication
- Edit profile.

Scheduling – consists of calendar view. When day is clicked- view of tasks for current day is shown

Communication – messaging functionality. Care worker can communicate with management. Patient details will show on this screen (if time permits)

Edit Profile- care worker can adjust name and picture on this interface.

(Timesheet Upload Section will be added if time permits)

+Website

Initial interface – login

Main Menu – containing links:

- Create/Update Schedules
- Communication

Create/Update Schedules- similar to Google Calendar where events are added, updated and deleted.

Communication – similar to mobile app.

(Edit Profile will be added if time permits)

2. Background Research

This section will provide detail on the background of my project through outlining general information on home care through highlighting research into what applications and technologies are currently available to this user group.

2.1 Home Health Care

"Home health care is a wide range of health care services that can be given in your home for an illness or injury. Home health care is usually less expensive, more convenient, and just as effective as care you get in a hospital or skilled nursing facility" (Home health care: what it is and what to expect ,2015)

Home health care, also referred to as domiciliary care, social care or in-home care involves supportive care provided in the home.

The main aim of home care is to provide for the needs of the patient, to allow the patient to remain living at home, regardless of age or disability. After surgery, a patient may require home care services that may range from such homemaking services such as cooking or cleaning to skilled medical care. Some patients require home health aides or personal care attendants to help them with activities of daily living (ADL). Many types of patients require home care.

Listed below are some services provided in home care in Ireland (Comfort Keepers, 2015):

- **General Personal Care** – bathing assistance, incontinence care, meal preparation, mobility and activity
- **Companionship and Home Making Services**- medication reminders, companionship and conversation, laundry, help with errands
- **Palliative Care Services**- relieving the symptoms of disease, supporting the patient and the family
- **Alzheimer's and Dementia Care** - alzheimer's respite care, intellectual activities, personal care assistance
- **Respite Care**- disability respite care, dementia respite care, adult respite care
- **Disability Care**- social support, night time support, educational support, exercise partnerships

Home care benefits include reducing hospital waiting lists, allowing people to stay independent for longer and allowing people to stay in their own surroundings for as long as possible. This requires assistance of a care worker also known as a care support worker or a

healthcare assistant. This care worker is responsible for lone working and delivering, according to a care plan drawn up by a nurse. A care plan is a set of tasks that a care worker must deliver while they are in the patient's home. A care worker may be with a patient from 30 minutes to hours, then with a different client with different needs.

According to Comfort Keepers (2015) benefits of home care include:

- People are happier in their own homes
- Neighbourhood and community support
- Remain independent
- Home care saves money
- Infection and healing
- Tailored care levels
- Your home is your castle

Governance is required as people who need homecare are generally vulnerable in nature and may not have anyone else in their lives. Care being delivered as set out and on time can be a matter of life or death – companies, the state and families need to know these patients are safe and being looked after.

For the care worker it can be hard to remember every detail, and changes that may occur based on the client request. Seeing as these care workers are lone workers they will need to communicate with their supervisors if there has been any issue in relation to delivery of care.

Lone workers are those who work by themselves without close or direct supervision. Anybody who works alone, including contractors, self-employed etc. is classified as a lone worker. The employer holds the main responsibility for protecting the safety and health of lone workers. Nonetheless, lone workers themselves have a responsibility to help their employer fulfil this duty.

Best practice requires documentation and communication. The idea behind this application is to allow for efficiencies in this practice.

2.2 Technology and Health Care

Demand for care services continues to increase due to demographic changes and rising expectations of quality of life. New technologies would assist care workers in delivering care where it is most required and at the most appropriate time. This would support the independence of vulnerable people living in this community.

According to Miller (1998):

"New technology may be able to enhance the range of current care services while also increasing efficiency and reducing the operating costs of those agencies providing the care"

The need for long-term care services is increasing and in future there will not be enough long-term care facilities to accommodate all patients. Many people prefer to grow old at home. This is a concept known as aging-in-place. It is important to take into consideration that the complexity of high-tech homecare, can be striking.

"A great deal depends upon the context of its deployment – upon the "who", the "where", the "why", and the "for how long" as well as the "what" **(**Arras, J & Dubler, N. 1994)

A wide variety of high-tech home care interventions is currently being administered to a very broad spectrum of patients .New innovations offer significant benefits in terms of safety and peace of mind for both patients and families. Technology adoption however could be a key problem. Healthcare is particularly sensitive area with vital social implications. It is important to understand technology acceptance. As indicated in the overview, many home health care systems are used to using their systems on desktop, and moving directly to mobile app may not be feasible, therefore having both desktop and mobile is good to begin with.

The demand on health- and social care personnel will increase. Take for example the elderly population which is growing and the wage-earning population is estimated to decrease. It is becoming more common that elderly people continue to live in their own home instead of moving to old- age care homes to receive care and personal service. Demands on care assistance increase as elderly people become fragile and to take care of elderly people demands competence.

"Through implementing IT, particularly mobile technologies can give elderly and persons with disabilities care and services on equal terms in their own home." (Janson, M & Wikman, A. 2012).

On a day to day basis home care assistants undertake complex work, however using I.T. systems as a supporting tool while they are on the field does not play a big role in the home health care industry in Ireland today. This has been discovered through the research of popular software's (1.1) and through interviews with this user group (appendix).

As a developer, it is important to gain knowledge of the care assistants working context. Understanding this work practice when designing and developing a mobile app is a learning practise and it is a necessity to make the app as usable, accessible and supportive as possible.

Increasing interest in end users reactions to health information technology has elevated the importance of theories that predict and explain health IT acceptance and use.

"As the adoption of information technology has increased, so too has the demands that these systems become more adapted to physicians and nurse environments to make access and management of information easier" (Beuscart, R & Brender, J & Zephir, M. 1997).

While developing software in health care industry it is important to use quality management techniques in order to ensure the product will satisfy the given requirements.

2.3 Mobile Apps and Health Care

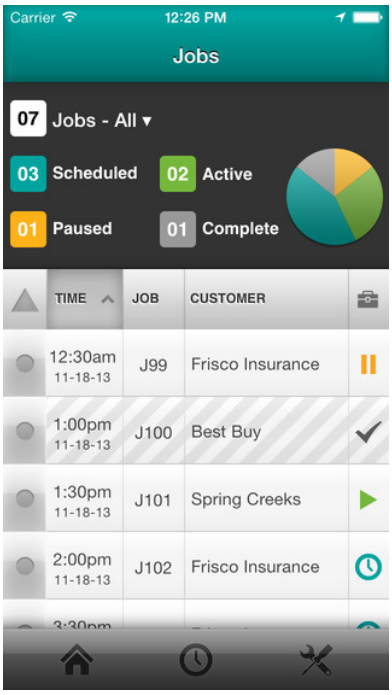
"Pervasive healthcare support through mobile information technology solutions is playing an increasing role in the attempt to improve healthcare and reduce costs" (Cocosila, M & Norm, A & Zhang, H . 2010)

An integrated care consortium called Kaiser Permanente leapfrogged the health industry when it made its entire electronic health care system- the most extensive electronic medical record offering in the world available to its 9 million members via an Android app. Kaiser patients now make appointments, check lab tests, order medicine and communicate with their physicians from the palm of their hand.

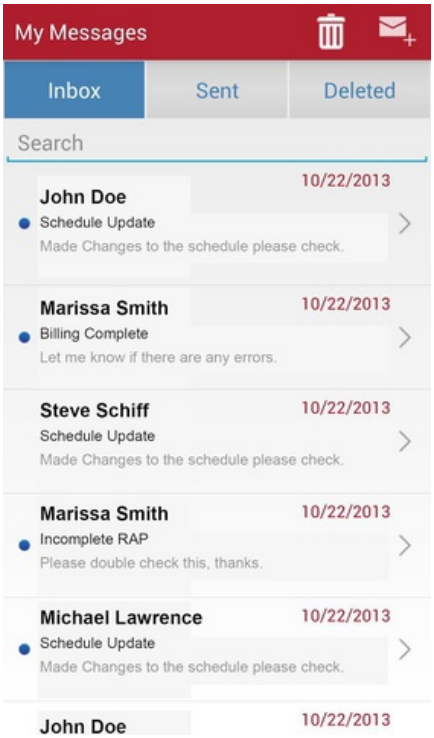
"Global sales of smartphones are expected to hit 1.5 billion units by 2016" (Forbes, 2012).

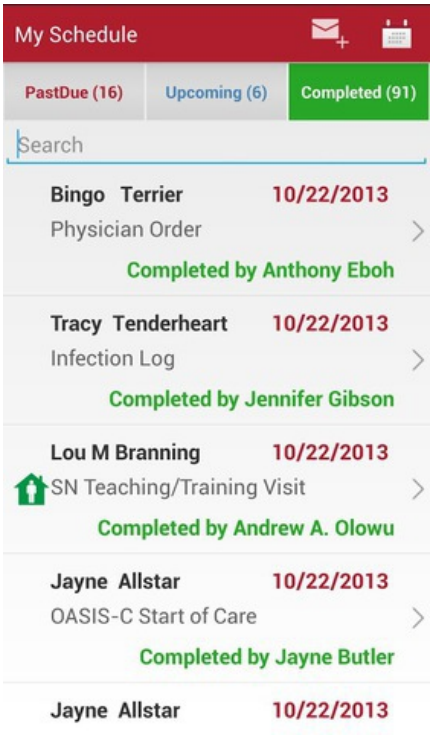
During research some of the best home care mobile apps available online have been considered (recommended from various resources).

FieldAware is a mobile app that considers easy home care scheduling and mobile apps for health care workers where they can review visit information, track completed tasks and much more.



Axxess Agencycore referred to in 1.1 is available on the Android App store. This app helps health agencies document and verify patient visits consistently through compliant messaging, easy intuitive scheduling and electronic visit verification.





3. Technologies Researched

3.1 Mobile App and Website

As outlined above, following research it felt necessary to develop both a website and a mobile app. Home care companies in Ireland work mainly on desktop, therefore developing a website alone would not be much different to what is already in place.

However, as stated in 2.2 moving a company's major functionality to a mobile app would be a drastic change. Therefore, the aim of this project is to develop both. In the future, progress could be made through incorporating both management and staff to the use of mobile app.

3.2 Programming Languages – Mobile App

*"The two most popular mobile operating systems are iOS with roughly 11.7% of market share and Android with 84.4%. Other mobile operating systems such as Windows Phone and Blackberry OS share 4%."(*IDC, 2015)

Deciding which language to use for this project was difficult. Taking into consideration technologies studied:

- Java(basic)
- C#
- Lua with Corona SDK
- SQL
- HTML
- CSS

Taking this into consideration, the choice would be made between writing the application using 'Objective C' for iOS or 'Java' for Android. With the type and the functionality of the app in mind, positives and negatives associated with both languages have been reviewed. Each platform has different software development kits (SDKs) and different integrated development kits (IDEs). Depending on the type and functionality of application

Android:

Positives:

Has 84.4% of market share of smartphones

It is simple to get software approved and posted on Google Play

Provides an open development platform

Can build on any OS – Linux, Mac, and Windows

Negatives:

The range of screen sizes used in android phones means it is more difficult to layout user-interface components

All layout guides must be entered using XML files

iOS:

Positives:

OS has an all-in-one IDE called 'Xcode'. It is compiler aware and syntax aware.

Xcode has its own interface builder with all the Apple User interface components for programmers to access.

Negatives:

iOS is a closed system and only allows users to purchase applications with an iOS operating system

An apple computer is required to programme.

Approval system is much longer than Google Play store.

After researching this, it has been decided that Android would be the application of choice. With no experience with Java for Android, there is a desire to improve this basic skillset.

3.3 Web Development- Reviewing Languages & Databases

After reviewing scripting languages such as Ruby, PHP and ASP it was decided that PHP would be used in the development of the web application.

PHP

Positives:

There is a large community – most problems faced by a web developer have existing solutions.

It has extensive database support

Most PHP scripts run on different operating systems with a minimum of tweaking.

N**egatives:**

Lacks named parameters which have to be simulated with associative arrays or other techniques

Inconsistent function names in the standard library (isset vs is_null)

JQuery and AJAX will also be used in the development of the web application. Prior to this project, I have gained no knowledge in the technologies stated above.

Databases

The 2 databases reviewed for this project were MySQL and SQLite.

MySQL

MySQL is easy to use, yet extremely powerful, secure and scalable. Because of its small size and speed, it is the ideal database solution for websites.

Positives:

Easy to use: while a basic knowledge of SQL is required and most relational databases require the same knowledge- MySQL is very easy to use. With only a few simple SQL statements you can build and interact with MySQL.

Secure: MySQL includes solid data security layers that protect sensitive data from intruders. Passwords are encrypted.

It's inexpensive: Available by free download.

Its fast: MySQL designers made the decision to offer fewer features than other major database competitors.

It's scalable: MySQL can handle any amount of data, up to as much as 50 million rows or more. The default size limit is about 4 GB.

It runs on many operating systems.

Negatives:

It has some stability issues

It suffers from relatively poor performance scaling

Its functionality tends to be heavily dependent on Add-ons

SQLite

Positives:

File Based

Standards Aware

Great for user testing

Negatives:

No user management

Lack of possibility for additional performance

Although SQLite is a very powerful, embedded relational database management system, MySQL was chosen for the purpose of this project. The main reason MySQL was chosen over SQLite is due to various resources rating it as the best available database.

"MySQL is the world's most popular open source database, enabling the cost-effective delivery of reliable, high-performance and scalable Web-based and embedded database applications, including all five of the top five websites"(Oracle,2015)

4 Design

4.1 Accessibility and User Interface Design

In the design phase of this app and website, mental models along with Nielsens Heuristics, have been helpful regarding usability and user interface design.

"Mental models are psychological representations of real, hypothetical, or imagery situations. They were first postulated by the American philosopher Charles Sanders Peirce, who postulated (1896) that reasoning is a process by which a human perceives in the parts of the diagram relations not explicitly mentions in the premises" (Johnson-Laird, P. 2014).

Mental models are one of the most important concepts in HCI. What users believe they know about UI strongly impacts how they use it. They base their predictions about the system on their mental models and thus plan their further actions based on how that model predicts appropriate course. In the design phase of the app and website, a massive goal is to make the user interface communicate the system's basic nature well enough that users form accurate mental models.

The design of the scheduling page on desktop is very similar to Google Calendar where schedules can be created, updated, deleted, dropped and dragged etc.

For the messenger/communication on the mobile app, WhatsApp UI has been taken into consideration especially in relation to the scroll feature and the placement of the send text area.

The 10 most general principles for interaction design are called ‘heuristics’. Evaluators measure the usability, efficiency, and effectiveness of the interface based on these 10 usability heuristics originally defined by Jakob Nielsen in 1994. This app aims to adhere to these principles. Below is an outline:

Visibility of System Status:

This app displays a progress/loading bar, if schedule is taking time to load. This is so the user is aware of the status.

Match between system and the real word:

This app aims to speak the user’s language using terms and concepts that are familiar to the intended audience. Schedules are displayed in a calendar view on the app which care workers would be used to seeing.

User Control and Freedom:

This app aims to allow the user to easily exit the system. Also there is functionality to undo and redo actions on the website. I.e. the manager can add, update or delete a schedule.

Consistency and Standards:

Controls, icons, terminology and error messaging aims to be consistent throughout the interface. For example using icons the audience are familiar with rather than creating new designs, this can be seen on the website when scheduling.

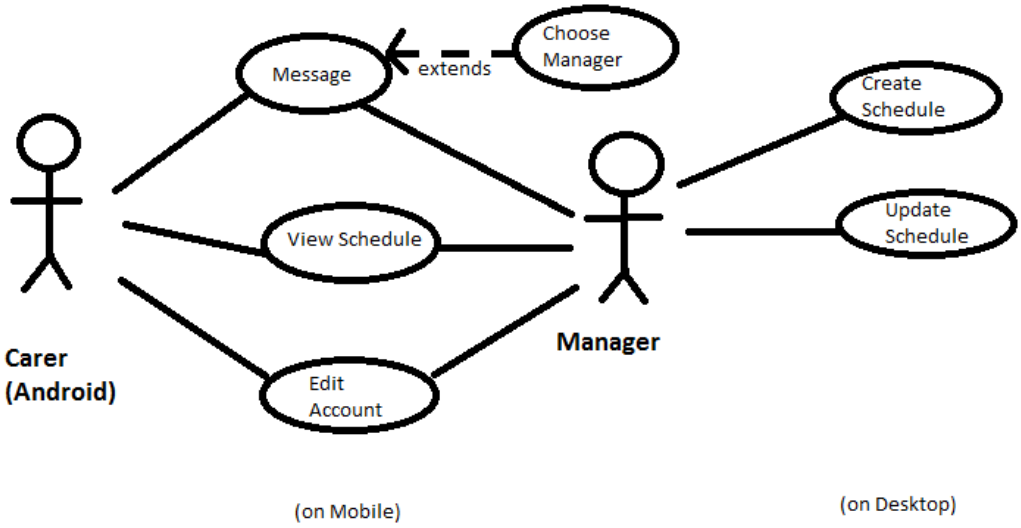
Recognition rather than recall:

This app aims to reduce the memory load of users by presenting familiar actions and options whenever possible. For example back button, calendar icons on website, chat icons etc.

Aesthetic and Minimalist Design:

This app avoids displaying excessive information and design elements, as they will visually compete with more relevant information on the screen.

4.2 Use Case Diagram



Outlined above is the initial use-case developed to propose the functionality of this app. Some of this has changed throughout the course of the app; some of the functionality is beyond the scope of this project.

There are two types of user

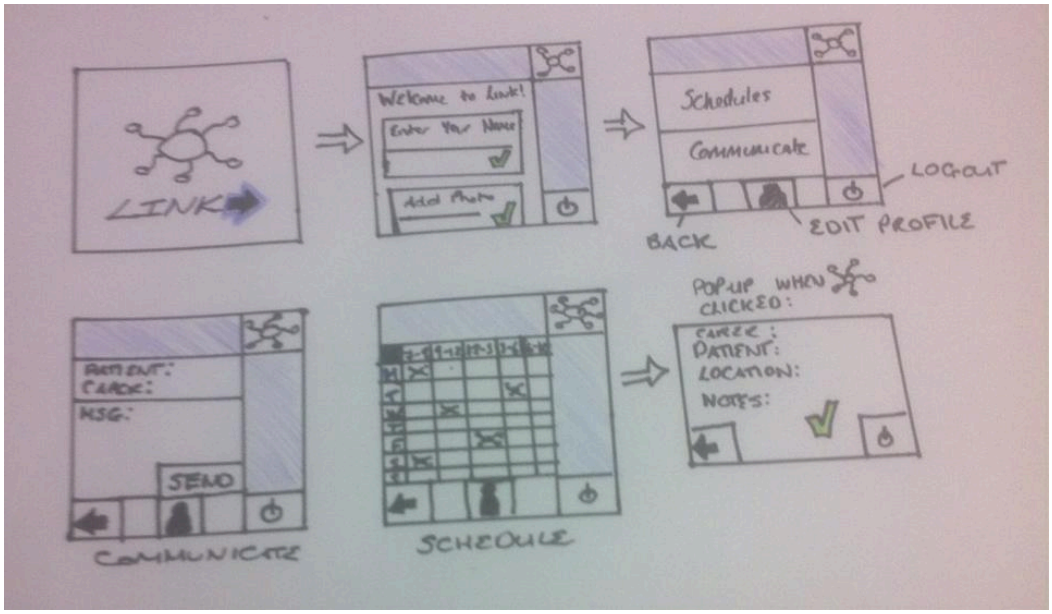
- Care worker(on android)
- Manager(on desktop)

A care worker can send messages, view their daily and monthly schedules and edit their profile details on the mobile app.

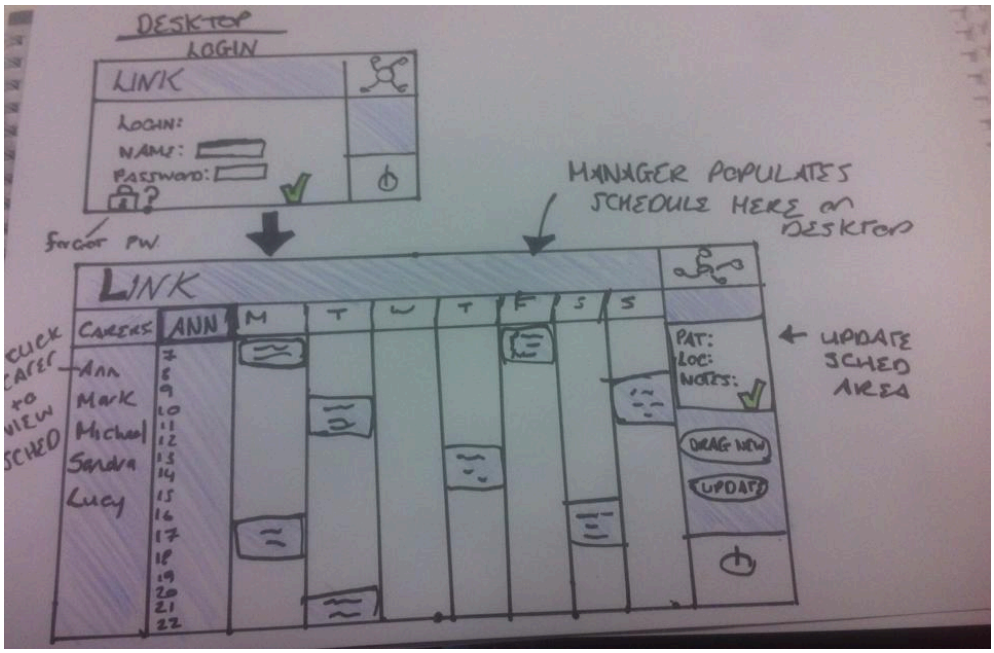
A manager can create, update and delete schedules and can send messages to care workers.

4.3 Design Prototypes

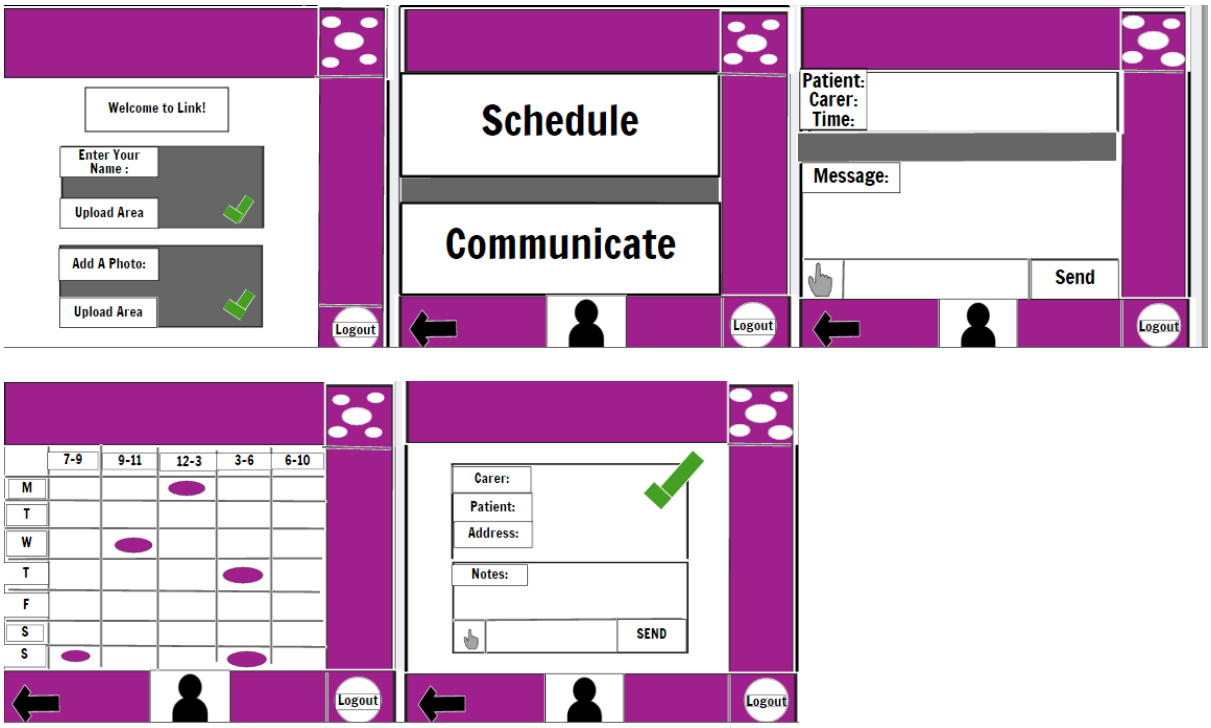
Below are the initial paper prototypes of the app and website. Since these prototypes have been drawn, many changes have been made to both.



The initial paper prototype of the website can be seen below. The final version of the website is much different, there is more functionality added. Styling is quite different for the scheduling as I decided to go with a style more similar to google calendar as mentioned above.



Medium fidelity prototypes of the mobile app:



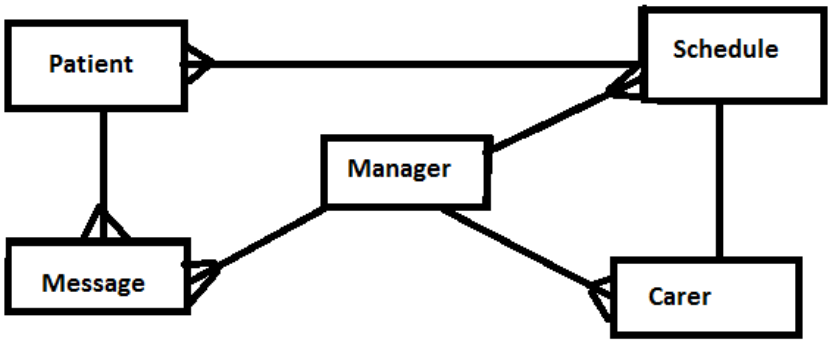
Firstly the user is welcomed with a screen to enter name and add a photo. This brings them to a screen with two choices to schedule or communicate. There is an option to logout, view profile and go back on each screen.

When the user clicks communicate they are brought to a screen where they can message their manager and the patient's name, carers name and time is brought to the screen.

When the user clicks schedule a calendar style feature is shown and when they click on the necessary date, a screen shows with information of the tasks/ patients they have that day.

This has also been updated since the initial design phase. The changes made can be seen in section 5.

4.4 Initial ERD



Above, is an initial ERD Diagram - a representation of an information system that shows the relationship between people, objects, places, concepts, events etc.

The entities in this ERD are Patient, Message, Manager, Schedule and Carer.

The relationship between the entities as follows:

- One patient to many messages
- One manager to many messages
- One manager to many carers
- One manager to many schedules
- One carer to one schedule

4.5 Requirements- Future/Suggested

Timesheet upload area for care worker on Mobile app

Mobile app for management while out of office

Patient documents section for mobile app

Map routing to addresses for care workers—(Google Map inspired)

Payroll on desktop for management

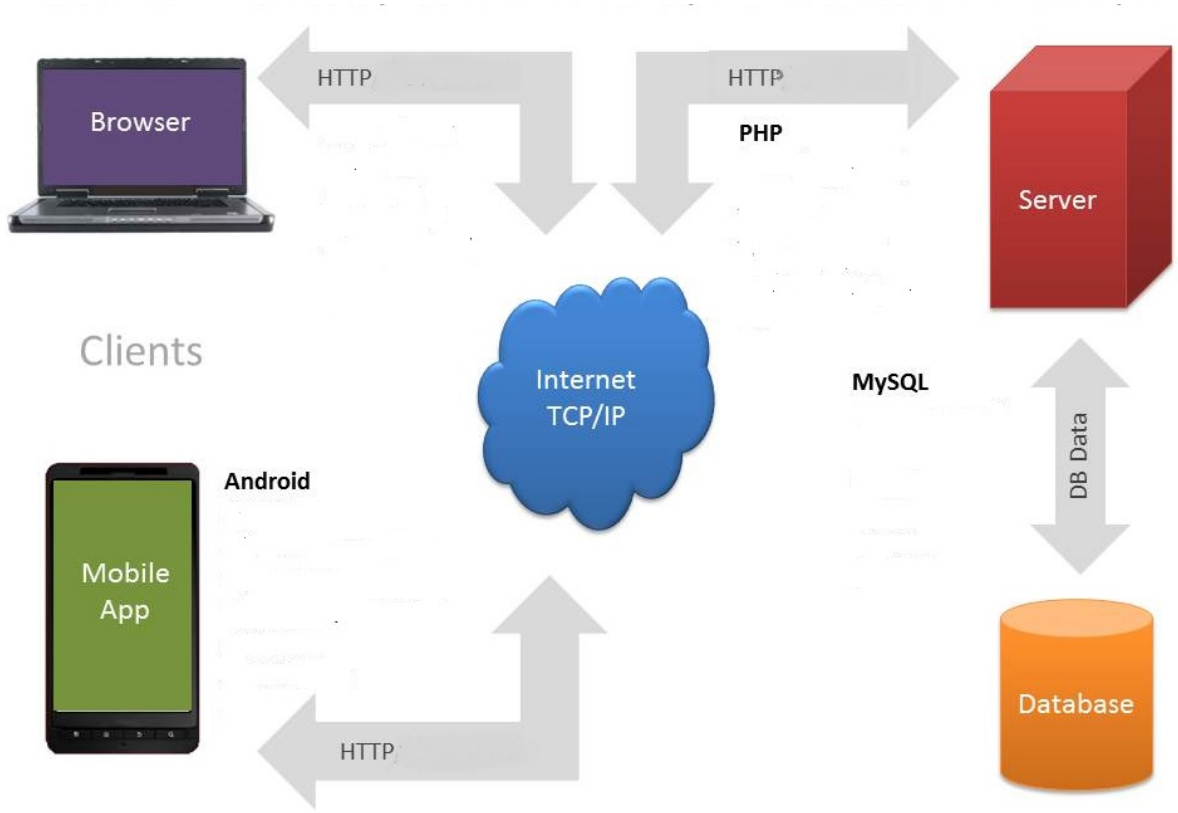
5. Implementation

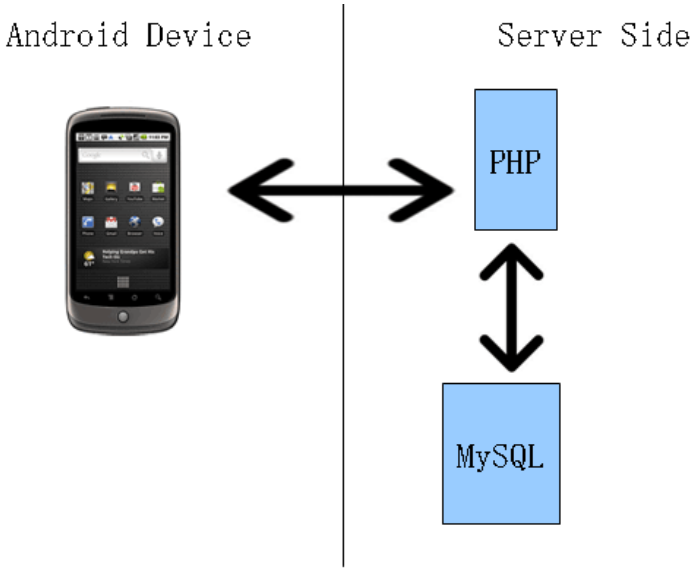
5.1 Software Architecture

Android/PHP/ MySQL Client

Unfortunately a MySQL database and an Android app cannot communicate directly. Therefore PHP can help to make this happen. In this project an Android application is created that communicates with a remote database (MySQL) using PHP. A PHP based web service will be created to all our Android. This project will use a server hosting service. The website is available at http://aislinggleeson.site11.com/.

Wamp Developer Pro's Apache server is also being used to test this locally. First your android app calls a PHP script in order to perform a data operation. The PHP script then connects to your MySQL database to perform the operation. The data flows from your Android App to PHP script then finally is stored in the MySQL database.





The implementation of this project has been split into three phases.

Phase 1 –MySQL and PHP

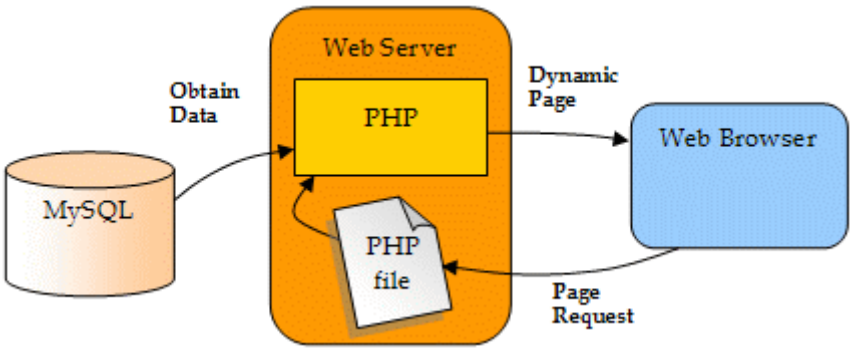
5.2 Creating MySQL Database on the Server

It was decided that MySQL would be used for this project (see 3.3). This section will describe how the database was created, how tables were created, and how to create, update, modify and delete data in the database. Database queries

The data in MySQL databases are stored in tables. A database query is a request to the database for specific information that returns a record set.

5.3 Opening a connection to MySQL with PHP

Before accessing data in the MySQL database, we need to be able to connect to the server.



A connection was opened to MySQL from PHP. PHP provides an inbuilt function ***mysql_connect ()*** which can be used to connect to database.

```php
<?php
if (!$link){
$dbhost='mysql13.000webhost.com';
$dbusername='a8050996_ash';
$dbuserpass='ash123';
$dbname='a8050996_ash';
$link=mysql_connect($dbhost, $dbusername, $dbuserpass);
mysql_select_db($dbname);
}
?>
```

5.4 Creating MySQL tables using SQL

4 MySQL Tables were created in SQL for this project. The CREATE TABLE statement is used to create a table in MySQL. The ERD has been outlined in the previous section and outlined below are the tables created- **USER, MESSAGE, SCHEDULE** and **jqcalendar**.

USER

```
CREATE TABLE `USER` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `USERNAME` varchar(20) COLLATE latin1_general_ci NOT NULL,
  `PASSWORD` varchar(20) COLLATE latin1_general_ci NOT NULL,
  `TYPE` int(1) NOT NULL,
  `DATE` datetime NOT NULL,
  `PHONE_ID` varchar(20) COLLATE latin1_general_ci NOT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=15 ;
```

SCHEDULE

```
CREATE TABLE `SCHEDULE` (
  `ID` int(7) NOT NULL AUTO_INCREMENT,
  `DATE` datetime NOT NULL,
  `NOTES` varchar(240) COLLATE latin1_general_ci NOT NULL,
  `ID_ACC` int(6) NOT NULL,
  `COMPLETE` varchar(240) COLLATE latin1_general_ci NOT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=4 ;
```

MESSAGE

```
CREATE TABLE `MESSAGE` (
  `ID` int(7) NOT NULL AUTO_INCREMENT,
  `TEXT` varchar(255) COLLATE latin1_general_ci NOT NULL,
  `IDFROM` int(6) NOT NULL,
  `IDTO` int(6) NOT NULL,
  `MSGTIME` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`ID`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci AUTO_INCREMENT=88 ;
```

jqcalendar

```
CREATE TABLE `jqcalendar` (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `Subject` varchar(1000) CHARACTER SET utf8 DEFAULT NULL,
  `Location` varchar(200) CHARACTER SET utf8 DEFAULT NULL,
  `Description` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `StartTime` datetime DEFAULT NULL,
  `EndTime` datetime DEFAULT NULL,
  `IsAllDayEvent` smallint(6) NOT NULL,
  `Color` varchar(200) CHARACTER SET utf8 DEFAULT NULL,
  `RecurringRule` varchar(500) CHARACTER SET utf8 DEFAULT NULL,
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=10018 ;
```

5.5 Inserting Data into MySQL with PHP

After the database and tables have been created, data can then be added to them. The INSERT INTO statement is used to add new records to a MySQL table:

The following code taken from *user.php* is a sample of inserting into the database.

The variable $username is being passed in as the only parameter to *isset(),* this will return true or false depending on whether the variable is set. If it is the data($phoneid and $username) will be inserted to the database

```
if (isset($username) && $username!=''){
    mysql_query("INSERT INTO USER(PHONE_ID ,USERNAME) VALUES ('".$phoneid."','".$username."')")
    $id = mysql_insert_id();
    echo 'inserted:'.$id;

}else{
    echo 'new';
}
```

Another example of the INSERT INTO statement taken from *message.php* file

```
]if (isset($from) && $from!='' ){

]if (isset($to) && $to!=''&& isset($text) && $text!=''){
 mysql_query("INSERT INTO MESSAGE(IDFROM ,IDTO, TEXT, MSGTIME) VALUES (".$from.",".$to.",'".$text."',NOW())");
 $id = mysql_insert_id();
 }

 $filter=' AND (M.IDFROM='.$from.' OR M.IDTO='.$from.')';

 }
```

The *mysql_insert_id()* function returns the id (generated with AUTO_INCREMENT) used in the last query.

The variable *$filter* seen in the code above is filtering data from the database with PHP. This features here so users of the mobile app will only receive messages relevant to them.

5.6 Selecting Data from database with PHP

The SELECT statement is used to select data from one or more tables or the * character can be used to select ALL columns from a table.

The code below from *chat.php* is an example of this. Here all columns from the USER table are being selected.

```
$sql = "select * from USER";
$handle = mysql_query($sql);
while ($row = mysql_fetch_object($handle)) {
  $ret['events'][] = array(
    $row->ID,
    $row->USERNAME
  );
}
```

Another example of a select statement can be seen in login.php. The query here is also selecting user information from the USER table.

```
// SQL query to fetch information of users
$resUser = mysql_query("SELECT * FROM USER WHERE password='$password' AND username='$username'");
```

5.7 PHP GET and POST Methods

PHP is also used to fetch the record from the MySQL database once it is created. In order to fetch record some information must be passed to PHP page regarding what record to be fetched.

The first method to pass information is through GET method in which ***$_GET*** command is used. The variables are passed in the URL and the record is fetched. The ***$_GET*** method sends the encoded user information appended to the page request. The page and the encoded information are separated by the **?** character.

Example of the **$_GET** method in *user.php* as follows:

```
$phoneid=$_GET["phoneid"];
$username=$_GET["username"];
```

The second method is to use POST method. Th***e $_POST*** method transfers information via HTTP headers. The information is encoded as described in the ***$_GET*** method and put into a header called **QUERY_STRING.** Example of the ***$_POST*** method in *datafeed.db.php* :

```
    case "adddetails":
        $st = $_POST["stpartdate"] . " " . $_POST["stparttime"];
        $et = $_POST["etpartdate"] . " " . $_POST["etparttime"];
        if(isset($_GET["id"])){
            $ret = updateDetailedCalendar($_GET["id"], $st, $et,
                $_POST["Subject"], isset($_POST["IsAllDayEvent"])?1:0, $_POST["Description"],
                $_POST["Location"], $_POST["colorvalue"], $_POST["timezone"]);
        }else{
            $ret = addDetailedCalendar($st, $et,
                $_POST["Subject"], isset($_POST["IsAllDayEvent"])?1:0, $_POST["Description"],
                $_POST["Location"], $_POST["colorvalue"], $_POST["timezone"]);
        }
        break;
```

The code mentioned above is part of a **switch statement**. The switch statement is used to perform different actions based on different conditions. First we have a single expression, in this case the variable $method. The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed – in this case – the ***$_POST*** method transferring information from database. The break is used to prevent code from running into the next case automatically.

5.8 PHP and Android

PHP selects information from the database and sends it to the Android application. For example in the ***tasks.php*** file PHP is selecting data that has been input from the desktop to

the jqcalendar table and sending this to Android application using ***echo statement***. How PHP and Android work will be explored in more detail in phase 3 of implementation.

```
if (isset($date) && $date!='' && isset($id) && $id!=''){

$resUser = mysql_query("SELECT * FROM  `jqcalendar` WHERE  `StartTime` >=
 '".$date."' and `EndTime` <=  '".$date." 23:59:59' and Location=".$id);

    while($row = mysql_fetch_array($resUser)) {

    $dateFrom=date_create($row["StartTime"]);
    $dateTo=date_create($row["EndTime"]);


        echo "Name: ".$row["Subject"]. "\r\nTime: " . date_format($dateFrom,"H:i:s")." ".
        date_format($dateTo,"H:i:s"). "\r\nAddress: " .     $row["Description"]."\r\n\r\n";
    }
```

Phase 2- Creating Desktop Application

Firstly as a test server, WampDeveloper Pro Server is downloaded. WAMP is acronym for Windows, Apache, MySQL and PHP. WAMP is software which creates a localhost environment for developing PHP, MySQL web application.

5.9 Login Functionality

Login.php and *Index.php* are the PHP files created for the login functionality.

A session is started with the *session_start()* function. This is an error I ran into on many occasions, forgetting to add this to a PHP file.

It is possible to insert the content of one PHP file into another PHP file (before the server executes it) with the include statement. In the code below we are including *conex.php* which connects to the database. The variable $error is set to store the error message.

If the user posts incorrect or empty login details, the error 'Username or Password is invalid' will show.

```
<?php
session_start(); // Starting Session
include ("conex.php");
$error=''; // Variable To Store Error Message
if (isset($_POST['submit'])) {
if (empty($_POST['username']) || empty($_POST['password'])) {
$error = "Username or Password is invalid";
}
```

MySQL Query selects all columns from the user table. The mysql_fetch_array() function fetches a result row as an associative array, a numeric array, or both. The session is

initialised if result is correct. If not error message is displayed and user is prompted to enter details again.

```
$resUser = mysql_query("SELECT * FROM USER WHERE password='$password' AND username='$username'");

if ($row = mysql_fetch_array($resUser)) {

$_SESSION['login_user']=$username; // Initializing Session
$_SESSION['id_user']= $row["ID"];

} else {
$error = "Username or Password is invalid";
}
}
```

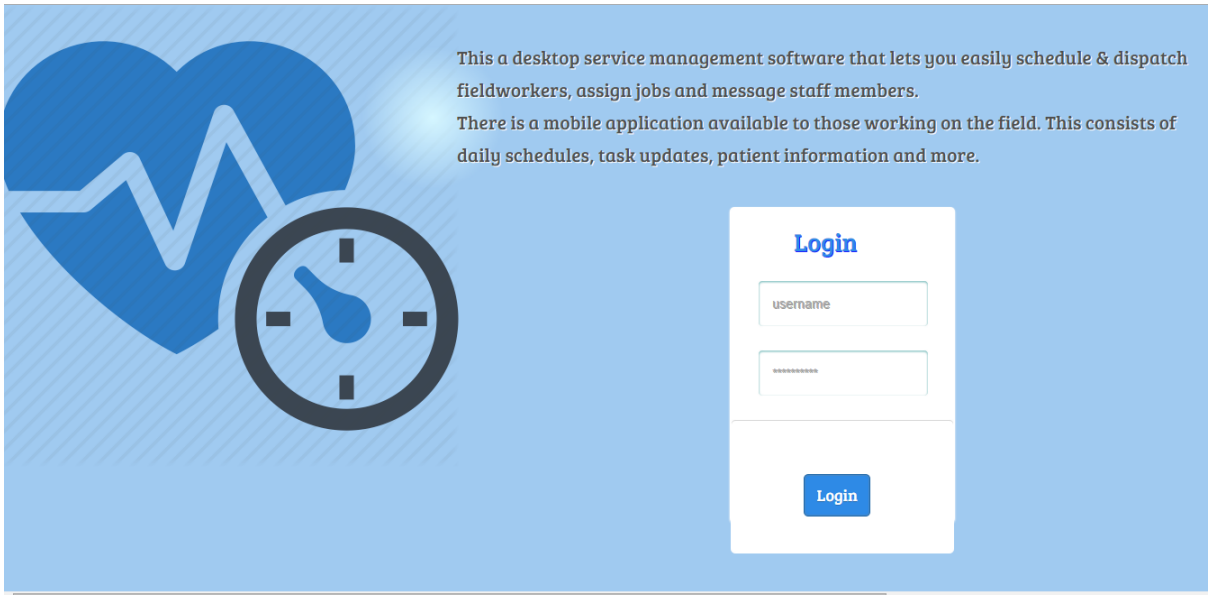
Index.php file includes *login.php.* If the session variable 'login_user' is set and true, the user is directed to *select.php* file. This is the main menu screen where the user can choose to message staff members, update schedules or logout.

```
<?php

include('login.php'); // Includes Login Script

if(isset($_SESSION['login_user'])){
header("location: select.php");
}
?>
```

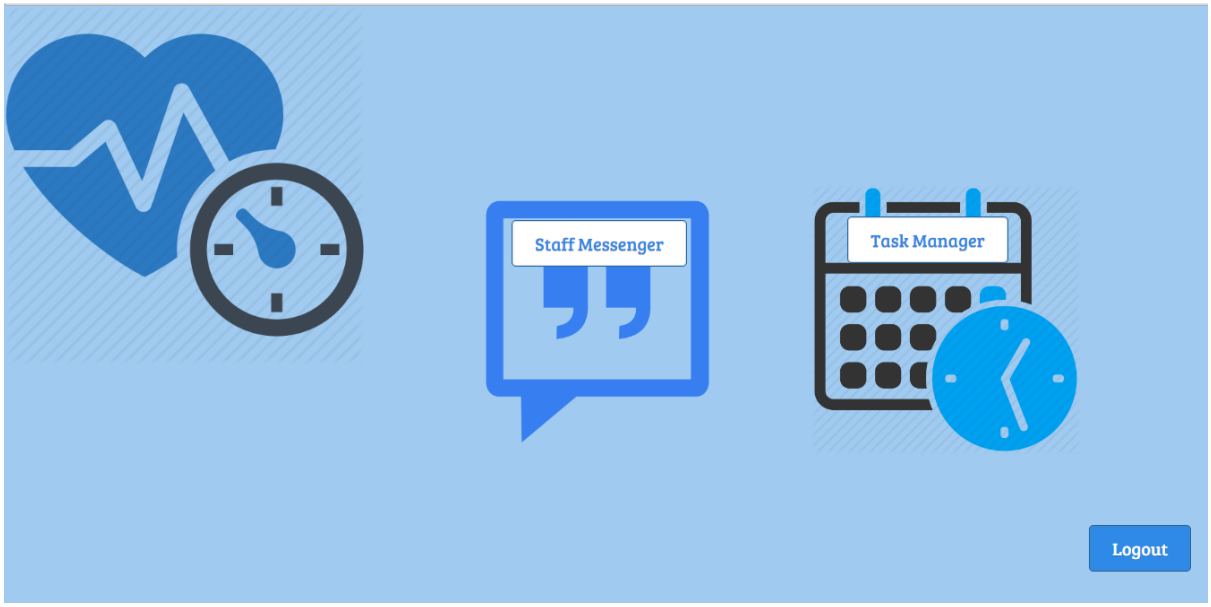
The *index.php* file also includes the html for the login form. Here is a snippet of the HTML used for the form.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Login Form</title>
<link href="css/login.css" rel="stylesheet" type="text/css">
</head>
<body>
 <div id="wrapper">
 <div class="container">

<form name="login-form" class="login-form" action="" method="post">
<div class="header">
<h1>Login</h1>
</div>
```

Below is a screenshot of the login on desktop.



After logging in the user is brought to the main menu screen. Here the user selects whether to use the messenger or to update the task manager. Screenshot of screen below:



5.10 Creating, Updating and Deleting Schedules

As stated in the User Interface Design section (4.4) I wanted the scheduling functionality of this application to be comparable with Google Calendar.

JQuery is a fast, small and feature-rich JavaScript library used for this project. It makes things like HTML document traversal and manipulation, event handling and Ajax must simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

For this project a jQuery plugin called **wdCalendar** was used. This jQuery AJAX event calendar is very similar to google calendar and other desktop calendar applications such as Microsoft Outlook or iCal on Mac OS X. Features of wdCalendar include: creating, updating, or removing calendar events or for the use of this project- schedules. This takes place by drag & drop. The user can also choose to have a daily view, weekly view or monthly view. wdCalendar is very powerful and integrates well with PHP.

After downloading wdCalendar there were 3 folders required. The CSS folder containing the imagery and styling. The JS folder containing the JS files and jQuery plugin files. The PHP folder containing the PHP files. The table associated with this in the database is the jqcalendar explained in 5.3.

These PHP files were connected to the DB through a *dbconfig.php.* The following code was implemented:

```php
<?php
class DBConnection{
    function getConnection(){
      //connecting to database server/user name/password
        mysql_connect("localhost","a8050996_ash","ash123") or
         die("Could not connect: " . mysql_error());
    //database name
        mysql_select_db("a8050996_ash") or
            die("Could not select database: " . mysql_error());
    }
}
?>
```

Using the include statement, I ensured to include *dbconfig.php* in the relevant PHP files. I also included *functions.php*

```php
<?php

session_start(); // Starting Session

include_once("php/dbconfig.php");
include_once("php/functions.php");

?>
```

The *functions.php* file contains functionality to produce a date and time print out for each entry into the DB. This is part of the wdCalendar script. I then uploaded the files to my site. On my site the calendar was populating random data. In order to change this, any reference to the *datafeed.php* file in the code I changed to *datafeed.db.php*. This was so the data from my DB would populate the calendar.

I also changed any functionality pointing to *edit.php* to *edit.db.php* so any information edited would communicate with my database rather than the demo.

Datafeed.db.php consists of the following functions:

function addCalendar(); function addDetailedCalendar(); function removeCalendar(); function updateDetailedCalendar(); function updateCalendar(); function listCalendar(); function listCalendarByRange();

This functionality is very similar to the wdCalendar JQuery plugin. *Datafeed.db.php* also contains ***$_GET*** and ***$_POST*** methods to perform different actions as mentioned in 5.6.

Edit.db.php consists of the following functions:

function getCalendarByRange(); function showerror(); function getHM(date);

Edit.db.php also contains the HTML for when a user clicks on 'New Event' or clicks on the event on the calendar to edit the details.

I made a few changes to this file so it would be relevant to my project. First of all I changed the HTML code so the task being added would involve carer and patient information. I changed headings from subject, location and address to ***Patient Name, Patient Address and Carers Name***.

Following this I created a function called ***getUsers()*** to fetch user information from the DB using PHP. This function allows the desktop user(manager) to select which carer the schedule affects.

The following code is adding events in the calendar using an array retrieved from the MySQL server. Exception handling is used to change the normal flow of the code execution of a specific error condition occurs- exception.

To handle an exception the code below uses try and catch syntax. Try function- if the exception doesn't not trigger, the code will continue as normal. However if the exception triggers, an exception is 'thrown'. The catch block retrieves an exception and creates an objection containing the exception information.

In this case, the code should retrieve the ID and USERNAME of the USERS from the DB.

```
function getUsers(){
$ret = array();
  $ret['events'] = array();

  try{
    $db = new DBConnection();
    $db->getConnection();
    $sql = "select * from USER";
    $handle = mysql_query($sql);
    //echo $sql;
    while ($row = mysql_fetch_object($handle)) {
      $ret['events'][] = array(
        $row->ID,
        $row->USERNAME
      );
    }
    }catch(Exception $e){
     echo $e->getMessage();
  }
  return $ret;
}
```

The ***foreach*** construct is used in the code below as a way to iterate over arrays. This will issue an error if you try to use it on a variable with a different data type or an unitialized variable. Seen in the code below the foreach loops over the array. The value of the current element is assigned to ***$value***. The PHP if statement along with the **echo function()** is then used to create a HTML option list. This HTML list allows the user to choose which carer they want to assign the task to

```
<span>*Carer Name:</span><br>
<select id="Location" name="Location">

<?php

$ret = getUsers();
foreach ($ret['events'] as $value){

    if (isset($event) && $event->Location==$value[0]){
        echo '<option selected value="'.$value[0].'">'.$value[1].'</option>';
    }else{
        echo '<option value="'.$value[0].'">'.$value[1].'</option>';
    }

}

?>
```

Schedule.php contains JQuery functionality and HTML in the PHP file which allows the user to view daily view, weekly view and monthly view of the schedule. Snippet of this code to show month view:

```
//to show month view
$("#showmonthbtn").click(function(e) {
    //document.location.href="#month";
    $("#caltoolbar div.fcurrent").each(function() {
        $(this).removeClass("fcurrent");
    })
    $(this).addClass("fcurrent");
    var p = $("#gridcontainer").swtichView("month").BcalGetOp();
    if (p && p.datestrshow) {
        $("#txtdatetimeshow").text(p.datestrshow);
    }
});
```

Other functionality in *Schedule.php* deals with connecting to and sending requests to the database. When the user clicks on save or close when adding, updating, or deleting a task. The following functionality is implemented.

function ***cal_beforerequest()*** uses a switch statement to perform different actions based on different conditions. If the value of the expression matches the value of the case, that case is executed. In this case when data is loading to database *"Loading data"* is executed. When the data is being processed *"The request is being processed"* is executed.

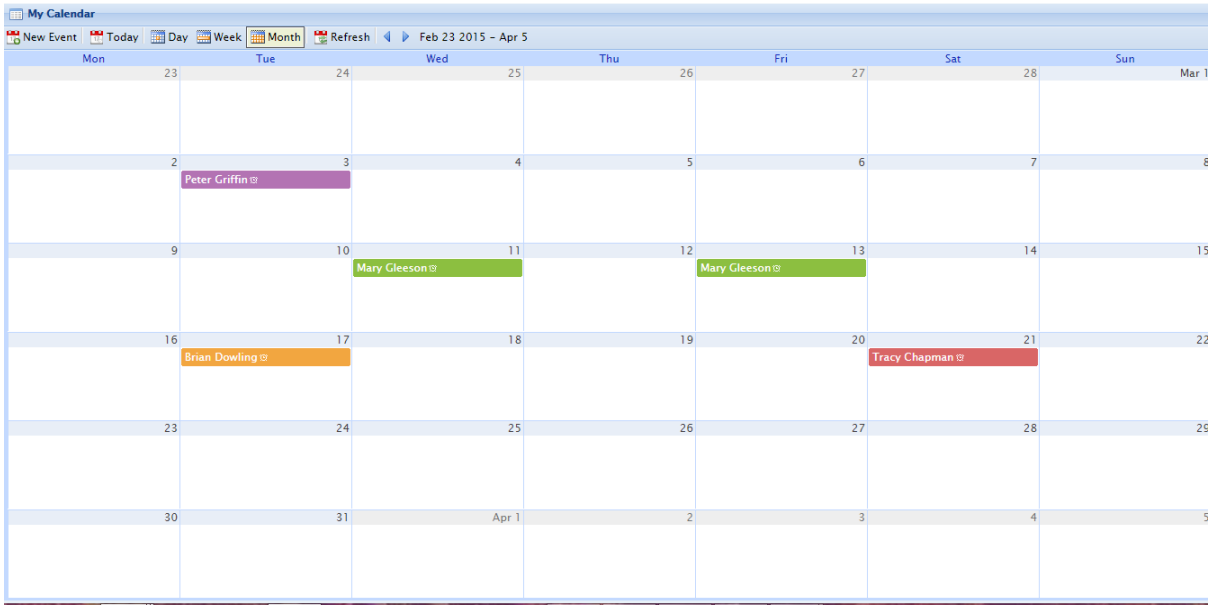
This can be seen in the snippet of code below.

```
function cal_beforerequest(type)
{
    var t="Loading data...";
    switch(type)
    {
        case 1:
            t="Loading data...";
            break;
        case 2:
        case 3:
        case 4:
            t="The request is being processed ...";
            break;
    }
    $("#errorpannel").hide();
    $("#loadingpannel").html(t).show();
}
```

The function ***cal_afterrequest()*** has been created for when the data reaches the database successfully. The HTML shows *"Success".* If this is not successful an *errorpannel* is shown.

```
function cal_afterrequest(type)
{
    switch(type)
    {
        case 1:
            $("#loadingpannel").hide();
            break;
        case 2:
        case 3:
        case 4:
            $("#loadingpannel").html("Success!");
            window.setTimeout(function(){ $("#loadingpannel").hide();},2000);
        break;
    }

}
```

Below is a screenshot of the scheduling page with some tasks added. The Patient Name is viewed on the screen and when you click on it more details show.

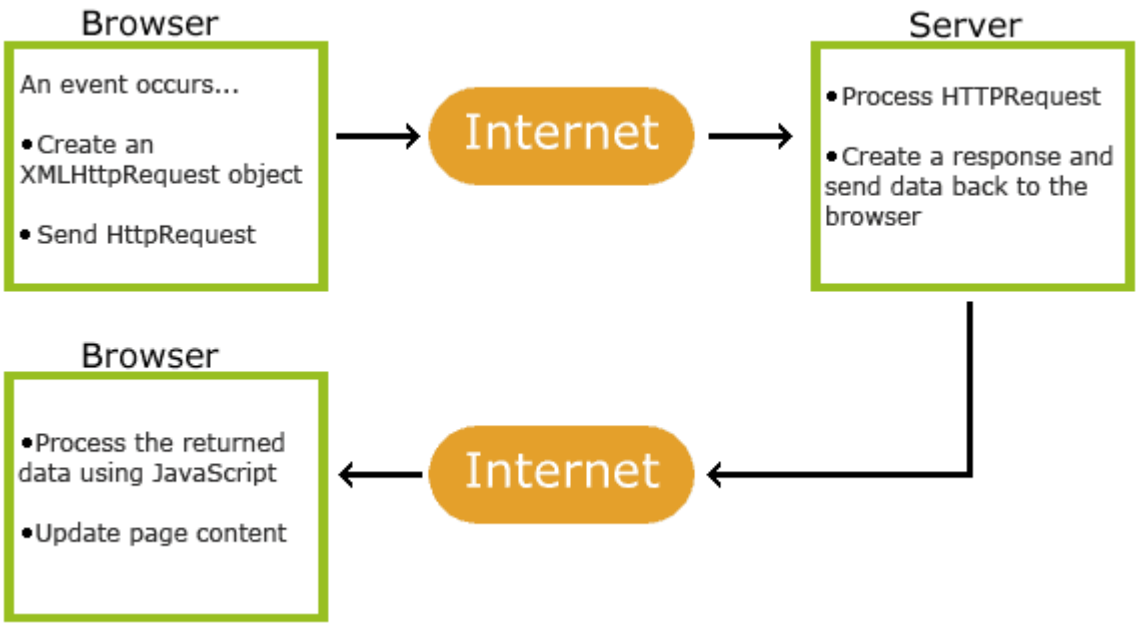


5.11 Messenger

In the ***select.php*** file on the site the user can select to 'message staff'. This brings the user to **chat.php** file.

I have implemented the same ***getUsers()*** function in this file as referred to in 6.2. This code has been implemented to retrieve the information of the users from the DB so the manager/ admin user can select which user they are chatting to from an option menu (see 6.2 for code). The same code has been implemented in the HTML of this file also to allow the manager to do so.

In this file, **AJAX** has been implemented. AJAX is Asynchronous JS and XML. It is a technique for creating fast and dynamic web pages. AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means it is possible to update parts of a web page without reloading the whole page.



AJAX- Send a request to a server

With AJAX to send a method to the server *open()* and *send()* methods are used. Sample of code using the send method in my project:

```
chat.send(text, strUser,'<?php echo $_SESSION['id_user']?>');
$(this).val("");
```

The reason AJAX is being used in this file, is so messages can be updated to the screen without reloading the page. In the ***chat.js*** three chat functionalities have been created.

function **getStateOfChat(); function updateChat(); function sendChat();**

function ***updateChat()*** updates the chat.

```
//Updates the chat
function updateChat(){


        $.get("message2.php", function(data, status){
         $('#chat-area').append($("<p>"+ data +"</p>"));
        document.getElementById('chat-area').scrollTop = document.getElementById('chat-area').scrollHeight;
        });


}
```

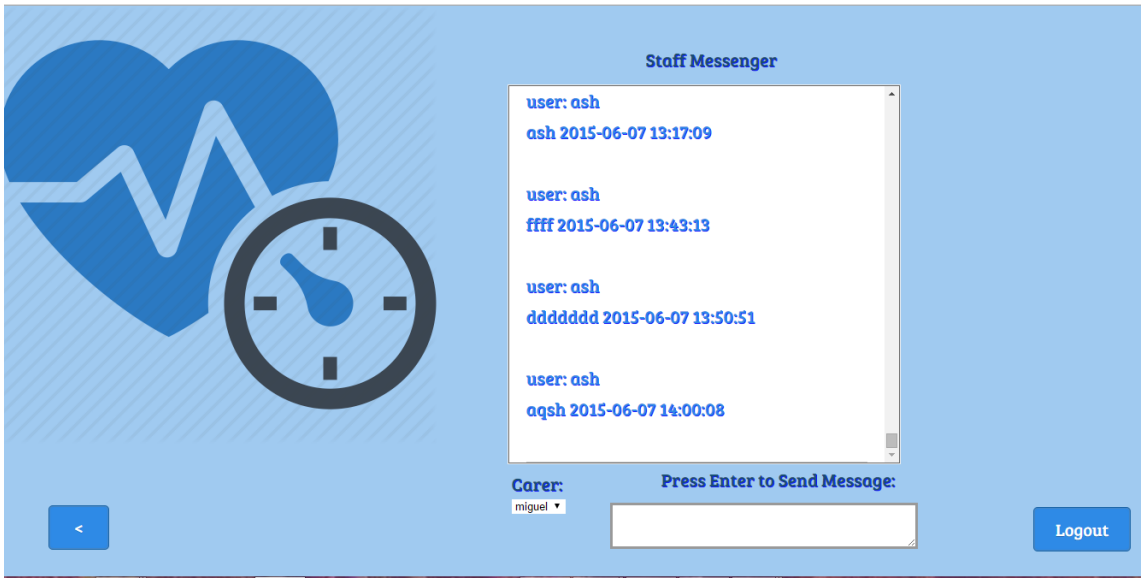
The following is implemented in the HTML of chat.php to update the messaging every 2 seconds.

<body onload="setInterval('chat.update()', 2000)">

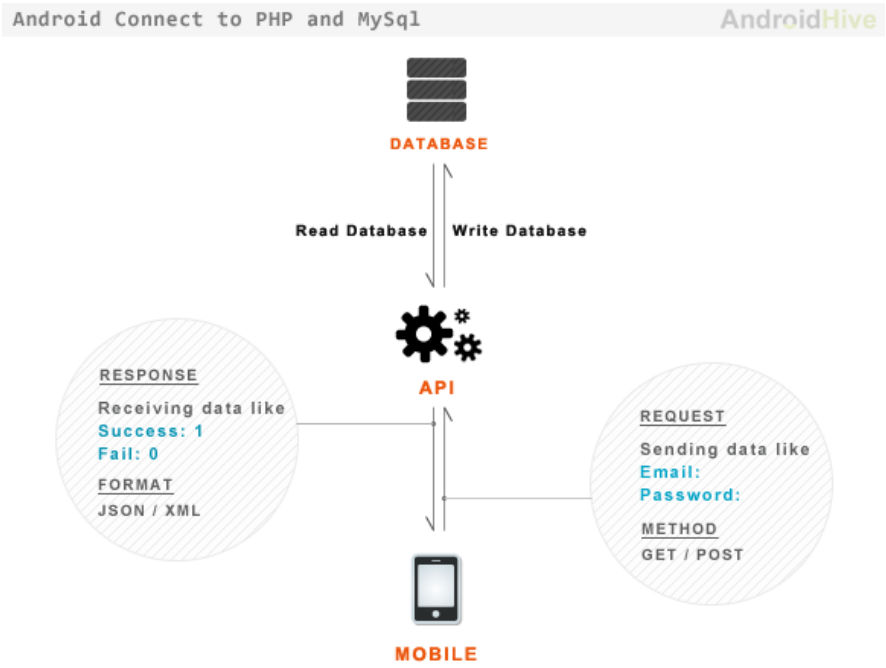
function ***sendChat()*** sends the chat. Code from JS as follows:

```
//send the message
function sendChat(message, nickname,from)
{

        $.get("message2.php?from="+from+"&to="+nickname+"&text="+message, function(data, status){
         $('#chat-area').append($("<p>"+ data +"</p>"));
        document.getElementById('chat-area').scrollTop = document.getElementById('chat-area').scrollHeight;
        });

}
```

Screenshot of messaging page:



Phase 3 – Creating Android Application



In order to create an android application a new project is created in the ECLIPSE ID as the Main Screen Activity. Following this AndroidManifest.xml is updated adding the classes to manifest file, and also adding internet connect permission.

In the Android Manifest all the classes that are created are added here.

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".MainActivity"
        android:label="@string/app_name" >
    </activity>
    <activity
        android:name=".UserActivity"
        android:label="@string/title_activity_user" >
    </activity>
```

The internet connect permission is also added to the Android Manifest as follows:

```
    <uses-permission android:name="android.permission.INTERNET" />

    <application
```

For this project 6 classes have been created to manifest 6 java activity files.

MainMenuActivity.Java

UserActivity.Java

MainActivity.Java

CalendarAdapter.Java

CalendarView.Java

CalendarViewSampleActivity.Java

5.11 MainMenuActivity.Java

When compiling the application, each XML layout file is compiled into a view resource. The layout can be loaded from the application code in the ***Activity.onCreate()*** callback implementation. This can be seen in my code below calling **setContentView().** This passes the reference to the XML layout resource

.***onCreate()***method creates the activity on the screen.

android_id gets the unique identification of the phone.

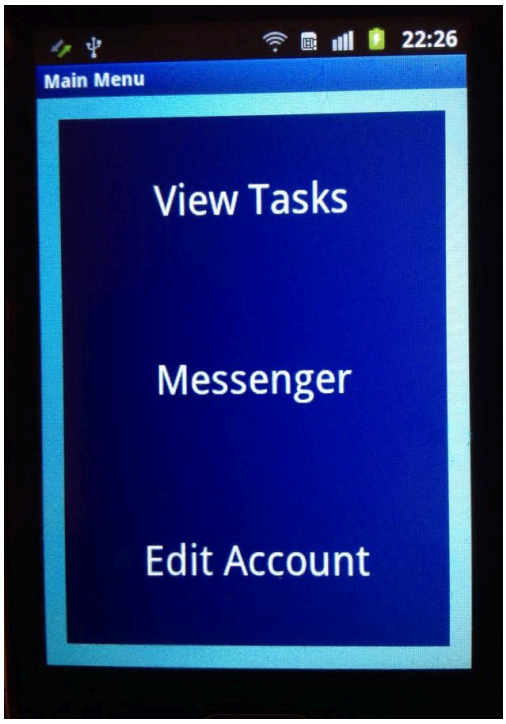
```
    //creates activity - screen
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_menu);
//get android id unique identification for that phone
        android_id = Secure.getString(getApplicationContext()
                .getContentResolver(), Secure.ANDROID_ID);

        System.out.println(">>>>>>>>>>>>" + android_id);
        new ComManager().execute("http://aislinggleeson.site11.com/user.php",
                "phoneid=" + android_id);
```

A loading bar has been added to the MainMenu activity for when data is loading to the screen. This is then dismissed at the end of the activity.

```
        //loading bar
        progress = new ProgressDialog(this);
        progress.setTitle("Loading");
        progress.setMessage("Wait while loading...");
        progress.show(); // open
```

There are 3 buttons displayed on this mainmenu screen as follows:

- View Tasks
- Messenger
- Edit Profile



The functionality of each of these buttons is implemented in the mainmenu activity file. Each button directs to another screen on the application. Snippet of code below indicates how these are developed.

Intent starts the activity. Firstly the button is added to the XML file. In the activity file there is a property at the beginning of the code to hold a reference to the button.

The ***onCreate*** method is overridden, and a listener is attached to the click event for the button. Optional parameters can be passed as per code below.

```
    //messenger button
    button2 = (Button) this.findViewById(R.id.button2);
    button2.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent myIntent = new Intent(MainMenuActivity.this, MainActivity.class);
            myIntent.putExtra("phoneid", ""+android_id); //Optional parameters
            myIntent.putExtra("myId", ""+myId);
            MainMenuActivity.this.startActivity(myIntent);
        }

    });
    //edit account
    button3 = (Button) this.findViewById(R.id.button3);
    button3.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent myIntent = new Intent(MainMenuActivity.this, UserActivity.class);
            myIntent.putExtra("phoneid", ""+android_id); //Optional parameters
            myIntent.putExtra("myId", ""+myId);// open window pass parameters
            MainMenuActivity.this.startActivity(myIntent);
        }
```

5.12 MainActivity.Java

This activity impacts the functionality of messenger on the mobile application. **onCreate()** shows layout for main activity. The content view is set to the activity_main_.xml. In this layout.xml file there is a web view, a send button and an edit text button.

The ***webview*** variable has been implemented in the main activity. The following code is added to access the website.

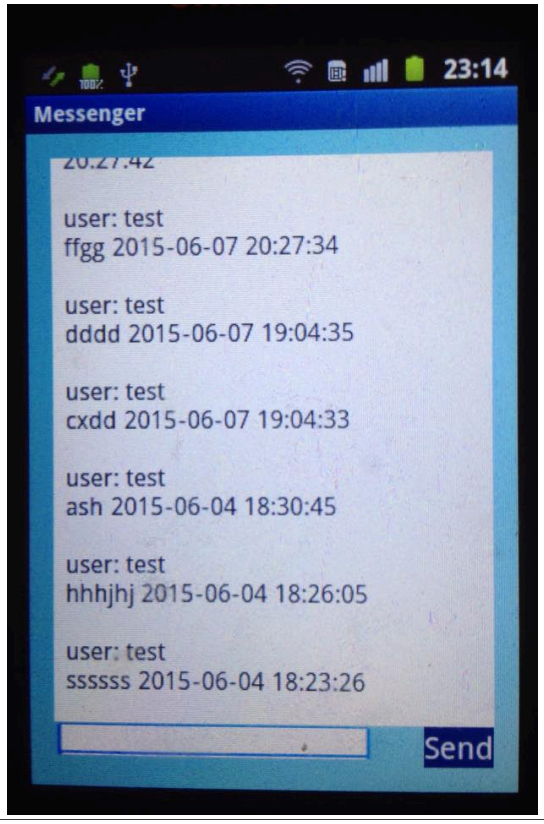
```
webview.setWebViewClient(new WebViewClient() {
    public boolean shouldOverrideUrlLoading(WebView view, String url) {
        System.out.println("Processing webview url click...");
        view.loadUrl(url);
        return true;
    }

    public void onPageFinished(WebView view, String url) {
        System.out.println("Finished loading URL: " +url);
        /*if (progressBar.isShowing()) {
            progressBar.dismiss();
        }*/
    }

    public void onReceivedError(WebView view, int errorCode, String description, String failingUrl) {
        System.out.println("Error: " + description);
        Toast.makeText(getApplicationContext(), "Oh no! " + description, Toast.LENGTH_SHORT).show();
        alertDialog.setTitle("Error");
        alertDialog.setMessage(description);
        alertDialog.setButton("OK", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                return;
            }
        });
        alertDialog.show();
    }
});
webview.loadUrl("http://aislinggleeson.site11.com/message.php");
```

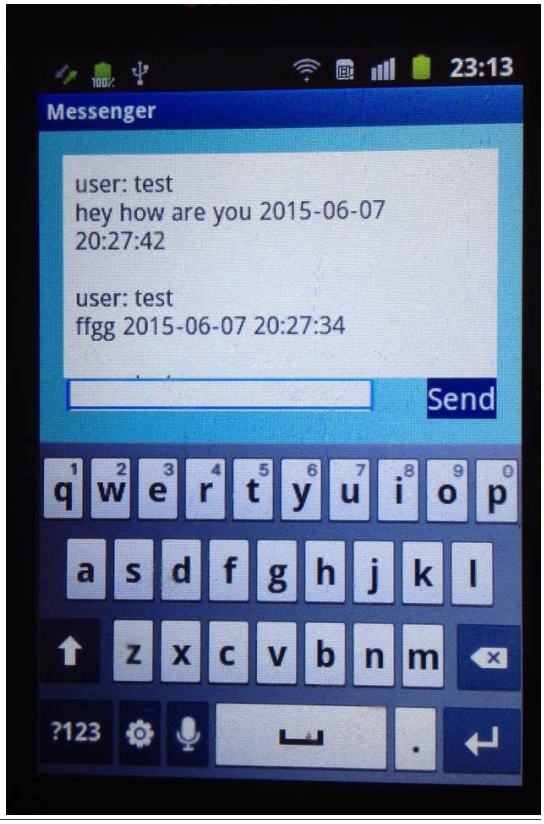
The code below inserts text into the database on the server. In order to request information from or insert data into the database, Android must connect through PHP. This code below is sending the text to the database. When the user clicks the send button, the information is sent to message.php which then connects with the database on the server.

```
msgtexbox =(EditText) this.findViewById(R.id.editText1);
btnsendtext=(Button) this.findViewById(R.id.button2);
btnsendtext.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        String res="http://aislinggleeson.site11.com/message.php?from="+myId+"&to=0&text="+msgtexbox.getText();
        System.out.println("#####BUTTON CLICKED "+res);
        btnsendtext.setEnabled(false);
        msgtexbox.setText("");
        webview.loadUrl(res);
        btnsendtext.setEnabled(true);
```

A scroll bar has been created at the side of the webview for usability. The scroll bar appears when the user taps on the message area, it scrolls downwards to view older messages.

```
webview.setScrollBarStyle(WebView.SCROLLBARS_OUTSIDE_OVERLAY);

final AlertDialog alertDialog = new AlertDialog.Builder(this).create();
```





5.13 UserActivity.Java

User Activity deals with thr user information and the profile features of this application.

The following code, is saving the user's current state of the application – **savedInstanceState.** It is important to always call the superclass implementation of **onSaveInstanceState()** so the default implementation can save the state of the view hierarchy. This can be seen in the code below.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_user);
```

The code below shows the functionality for saving user details. Firstly the button is added to XML file and assigned ID. The + tells Android to generate an ID for this element so can be referenced in Java files. In the activity class there is a property at the start of the code to hold a reference to the button:

Button btnSend;

The onCreate method is overridden and a listener is attached to the click event for this button as seen in the code below.

getIntent() gets phone id from parameters of previous window.

When ***onClick()*** is pressed a call is made to ***user.php***, passing the parameters **username** and **phoneid:**

```
intent = getIntent();
phoneid = intent.getStringExtra("phoneid"); //if it's a string you stored.

btnSend =(Button) this.findViewById(R.id.button1);
nameText =(EditText) this.findViewById(R.id.editText1);

btnSend.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
```

The following code communicates to the site through the ***user.php*** file:

```
//communicates to site--php
new ComManager().execute("http://aislinggleeson.site11.com/user.php", "phoneid="+phoneid+"&username="+nameText.getText());
```

AsyncTask enables easy use of the UI thread. This class allows to perform background operations and to publish results on the UI threads without having to manipulate threads and/or handlers.

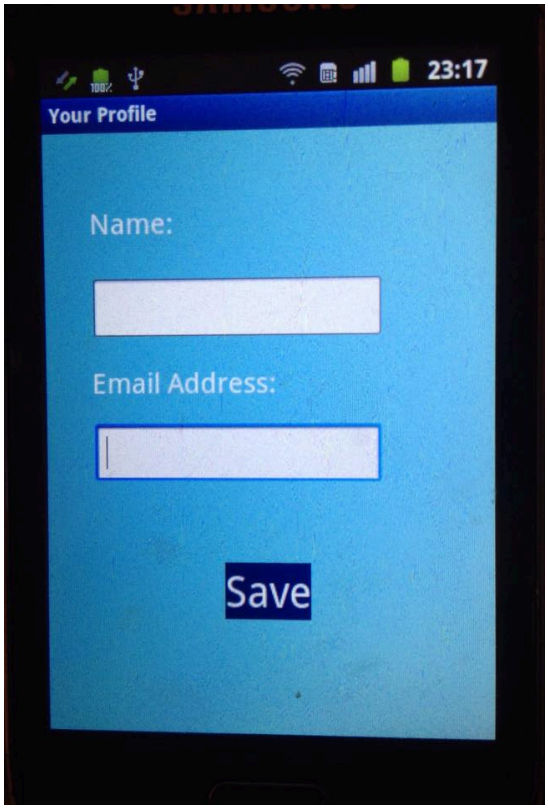
AsyncTask is designed to be a helper class around Thread and Handler and does not constitute a generic threading framework.

An asynchronous task is defined by 3 generic types called Params, Progress and Result and 4 steps called onPreExecute, doInBackground, onProgressUpdate and onPostExecute

AsyncTask must be subclassed to be used as seen below. This communicates with the site in the background without disrupting functionality.

```
//communicates an easy use of UI thread--- operations in background
private class ComManager extends AsyncTask<String, String, String> {
      // boolean changeUser=false;
      @Override
      protected String doInBackground(String... params) {
          String responseString = "";
          AndroidHttpClient client = null;
          try {
              String link = params[0] + "?" + params[1];
              System.out.println(link);
              client = AndroidHttpClient.newInstance("Android");
              HttpResponse response;

              // System.out.println("<==="+link);
              HttpGet myHttpGet = new HttpGet(link);
              response = client.execute(myHttpGet);

              StatusLine statusLine = response.getStatusLine();
              // System.out.println(">===="+statusLine.getStatusCode());
              if (statusLine.getStatusCode() == HttpStatus.SC_OK) {
                  ByteArrayOutputStream out = new ByteArrayOutputStream();
                  response.getEntity().writeTo(out);
                  out.close();

                  responseString = URLDecoder.decode(out.toString(), "UTF-8");

              }
```

On opening the app the first screen displayed to the user is the screen below. The user enters their name and email address.



The code mentioned in the section above pulls the unique phone id to the database. This turns the user from a new user to an existing user. Next time the user opens the app it will open on the main menu screen as their details are now stored on the database.

However, when the app is opened on a different phone or by a different user they will be presented with the 'Your Profile' screen above.

This screen is also accessible through the main menu, if a user wishes to edit their account.

5.14 Implementing a Calendar in Android

In order to implement a calendar in Android a GridLayout is created with 7 columns and a custom BaseAdapter is created to display data correctly. The aim is to create a single calendar view where the user can navigate between months and choose a date.

Two classes are required. One class to display calendar grid and the other to adapt calendar days and items into the grid.

CalendarView.java

This is a class that encapsulates the UI of the calendar- with a header (navigation and name of the month) and a GridView.

onCreate() does some initializations and res/layout/calendar.xml is set for the content view.

The handler attribute is used to populate calendar items onto the calendar. Event listeners are then set.

GridView's OnClickListener returns selected data as a string format as seen in the code below.

```
gridview.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View v, int position, long id) {
        TextView date = (TextView)v.findViewById(R.id.date);
        if(date instanceof TextView && !date.getText().equals("")) {

            Intent intent = new Intent();
            String day = date.getText().toString();
            if(day.length()==1) {
                day = "0"+day;
            }
            // return chosen date as string format
            intent.putExtra("date", android.text.format.DateFormat.format("yyyy-MM", month)+"-"+day);
            setResult(RESULT_OK, intent);
            finish();
        }

    }
});
```

refreshCalendar() method asks calendar adapter to refresh itself and changes month name.

onNewItem() method captures date parameter and sets it as current date

CalendarAdapter.java

This is a class that handles the content of the calendar – how many days to display and whether to display the calendar item icon or not. ***refreshDays()*** method formats source array for the grid. Application assumes that week's start on Monday, thus calculation is needed on how many empty days are before the first day of the month.
getView() method creates a view for each item referenced by the adapter. Layout of the view is defined in *res/layout/calendar_item.xml.*

CalendarViewSampleActivity.java

This is an activity class that opens CalendarView. DatePicker's date is updated by the return value of the Calendar View.

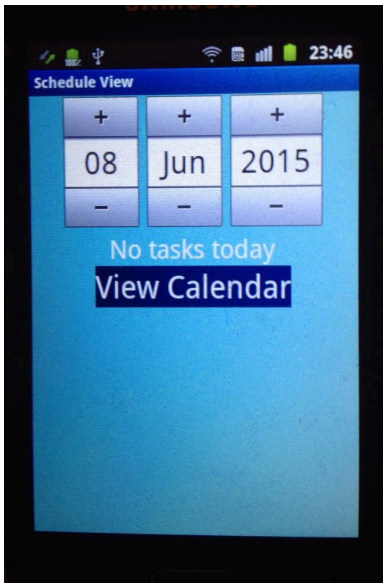
on DateChanged() communicates with the database through PHP. This is to retrieve information about the tasks assigned for the carer on that day.

```
dp.init(dp.getYear(), dp.getMonth(), dp.getDayOfMonth(),new OnDateChangedListener() {
        //calendar view gets the tasks from the php from the database
        @Override
        public void onDateChanged(DatePicker arg0, int arg1, int arg2, int arg3) {

            String dat="date="+arg1+ "-"+ (arg2+1) + "-"+arg3;
            if(!myId.equals("")){

                new ComManager().execute("http://aislinggleeson.site11.com/tasks.php", dat+"&id="+myId);
            }

        }
       } );
```

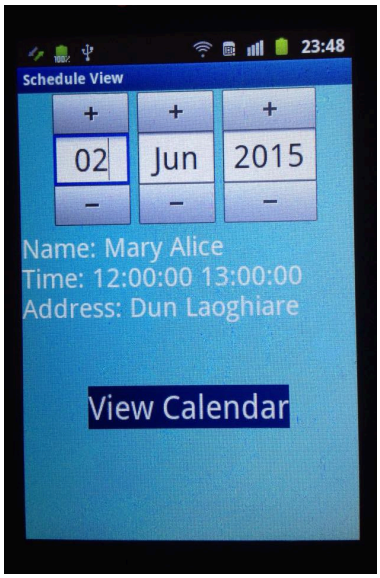
When there is a task in schedule on the chosen day the following code is executed. If there is no data for the chosen day 'No tasks today' will appear.

```
// when there is a task in schedule
        @Override
        protected void onPostExecute(String data) {


            if (!data.equals("")) {
                textView1.setText(data);


            }else{
                textView1.setText("No tasks today");

            }

        }
```

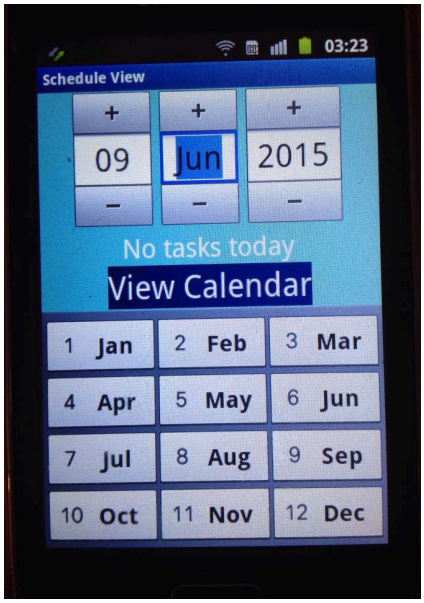
Calendar View when there is no tasks for the user.



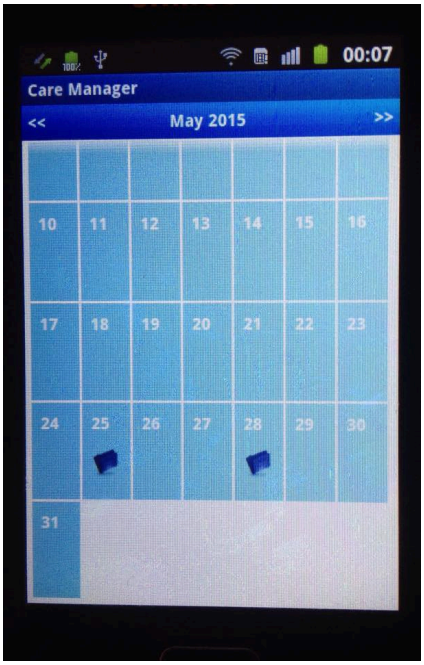
Calendar View when there is tasks in schedule for care worker. Patient Name, Time and Patient Address is pulled from the database as per code above.

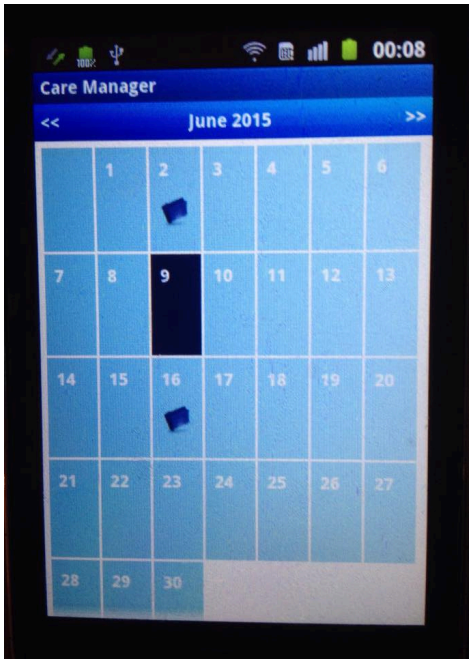


When user clicks on day, month or year, they can change the date from menu below.



Monthly view of calendar, blue files on a date indicates user has tasks that day, dark blue is focus on current day, user can access months with the arrows at the top of the screen for the year 2015.





5.16 Outstanding Issues

The option for the manager on desktop to have different views on the messenger for each carer needs to be implemented fully. Currently all the messages from all the carers are in one screen. This will be achieved in the next development stage.

Adding a logout and back button to each screen on the mobile app is also an outstanding issue. Currently the user can press the back button on the phone to go back to the last screen, however this is a requirement for the next phase of development.

It would also be preferable to update logos and personalise the mobile app more. Styling is quite minimal.

Updating the styling on how the tasks are displayed on mobile. Also updating the styling of the text in the messenger on both desktop and mobile app. Adding user icons is another issue that will be considered in the next development stage of this project.

6 Evaluation

The link to the website was sent to the operations manager of a Health Care company in order to observe how a manager would interact with this web tool. There were some interesting observations that will drive the completion of this application .

One such observation, is with the messenger page. Feedback from the operations manager was it would be beneficial if when the 'carer's name' was clicked a new message screen could show instead of all the messages appearing on the same screen to avoid confusion. This will be implemented in future development.

The manager liked how the scheduling worked and said it was easy to follow. A feature she said she would like to be included was a patient information section in which the patients information can be taken and added to the schedule rather than manually updating, this will also be taken into future development as it would improve efficiency.

The mobile application was presented to care workers on a test phone. Her feedback was that it was very easy to use and to the point. She thought the messenger tool while on the field would be very efficient. Feedback on this section of the mobile app was quite successful,, considering this is not in place in the company at the moment. Also the view schedule was reviewed well also. One aspect for the mobile application would be to add back buttons to the screens , as one of the workers had difficulty using the phone's back button. This will be considered in the future also.

A feature that was not included in this project was a timesheet update section on the mobile application. Care workers said this would be very beneficial to them and this will be the main future requirement for this project.

Overall both management and staff engaged enthusiastically with the website and the mobile app. They said it was basic, easy to follow and a requirement needed in their company. This was the main goal at the beginning of this project. These software's have been developed to make the life of the user group easier in the work place.

In the future, all feedback and testing will be taken into consideration for further development of both the website and the app.

7 Conclusion

7.1 Future development

The aim from here is to develop this website and application to meet the requirements of this company. The outstanding issues mentioned in the implementation section also need to be improved. The feedback taken from the evaluation(6) will also have to be implemented for this software to improve.

Improvements will be made on the styling of both desktop and mobile application in the future, adding logos to the mobile app, personalising it etc. This would increase the usability for the care workers.

Another element to take into future development, is to make the mobile app accessible to management. This would allow management to interact with their employees while they are out of office or away from their desks for a period of time.

7.2 Lessons learned

This project has been a massive learning curve. It has provided a space to put areas of software development studied in the past year to practice. A basic understanding of object oriented programming (Java and C#) has given a good foundation for this project. On completion, I have gained confidence and made improvements on my technical skills from this project. Time management skills have also improved. This is helpful for my role as a project manager in work.

Dealing with development issues that arose throughout the course of the project, has also been a valuable lesson learned. Through exploring design principles, usability and testing, I have gained knowledge on the user group and meeting user requirements.

The majority of functional requirements set out at the beginning of this project have been met; however following evaluation and feedback there are more requirements to take into consideration. This project is a working progress and I have every intention of reaching these requirements and developing this software further.

8. References

8.1 Journals

Arras, J & Dubler, N. (1994). Bringing the Hospital Home Ethical and Social Implications of High-Tech Home Care. The Hastings Center Report. 24 (5), p.19-28.

Beuscart, R & Brender, J & Zephir, M. (1997). How to assess the usability of information technology in healthcare. Computer Methods and Programs in Biomedicine. 54 (1), p19-28.

Cocosila, M & Norm, A & Zhang, H . (2010). Factors of Adoption of Mobile Information Technology by Homecare Nurses: A Technology Acceptance Model 2 Approach. CIN: Computers, Informatics, Nursing:. 28 (1), p49-56.

Janson, M & Wikman, A. (2012). Care assistants experiences, opinions and expectations. *Information and Communication Technology in Homecare* . 1 (1), p3-14.

Hagglund, M. & Koch, S. & Scandurra, I. (2010). Scenarios to capture work processes in shared homecare—From analysis to application.*International Journal of Medical Informatics*. 79 (Abstract), p126-134.

Miller,I. (Aug. 1998). Lifestyle monitoring-technology for supported independence. Computing and Control Engineering. 9 (4), p169-174.

8.2 Websites

Comfort Keepers. (2015). *Getting started with homecare.* Available: http://comfortkeepers.ie/information/step-by-step-guide-to-home-care/. Last accessed 27th June 2015.

Gaby, L. (2015). *Compare Home Health Care Software.* Available: http://www.softwareadvice.com/ie/home-health/. Last accessed 27th May 2015.

IDC Corporate USA. (2015). *Smartphone OS Market Share, Q1 2015.*Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp. Last accessed 08/06/2015.

Johnson-Laird, P. (2014). *What are mental models?.* Available: http://mentalmodels.princeton.edu/about/what-are-mental-models/. Last accessed 27th June 2015.

Forbes. (2012). *5 Ways Mobile Apps Will Transform Healthcare.*Available: http://www.forbes.com/sites/ciocentral/2012/06/04/5-ways-mobile-apps-will-transform-healthcare/. Last accessed 27 June 2015.

ORACLE. (2015). *MySQL The World's Most Popular Open Source Database.* Available: http://www.oracle.com/us/products/mysql/overview/index.html. Last accessed 7th June 2015.

The Official Goverment Site for Medicare. (2015). *Home health care: what it is and what to expect.* Available: http://www.medicare.gov/what-medicare-covers/home-health-care/home-health-care-what-is-it-what-to-expect.html. Last accessed 27th June 2015.

9.Appendix

9.1 Requirements- Interviews with Management and Care Staff

I brought the initial idea of the app/website to an operations manager of one of the leading home care companies in Ireland- Comfort Keepers. I also interviewed a care worker on her team. Before the development stage, I was unsure of whether to add the functionality specific to the manager to the app or to a website. The feedback given in these interviews, along with other research helped me decide.

In the interviews, I asked opinions on thoughts of introducing a mobile app to the company, what benefits would come from this type of app, accessibility of the app, and ideas around other functionality they might like to see added to the app, and if they saw a place for this in their company.

Highlighted below is the interview held with operations manager and care worker:

Interview with Operations Manager:

What are your thoughts on having a mobile app to communicate with the lone care workers in your company?

I think it is a great idea to have an app to communicate with my staff. A lot of time is spent over the phone and on emails communicating with these lone workers. I think an app specific to this would be great, and beneficial to both management and staff. This type of app would be more efficient and professional than what is in place at the minute in our company. I think it would be an efficient way of communicating and easy to use for both managers and care workers.

What is the main thing that would benefit your company from this type of application?

The most beneficial thing in my opinion anyway, is the fact that this app would combine both scheduling and a tool to communicate. At the moment our company does not possess this even on desktop, so to have this combined and available on your phone would be great.

Also, what I like about this is that everything is logged in a professional way, rather than staff members calling me and texting me. Having an application like this, means questions asked previous are available on the application.

Is there any other functionality you would like to see on this app?

What would be great, would be some sort of tracking device. I have so many carers in so many different locations during the day. Sometimes it's easy to forget where staff members

are at a certain point in the day. Normally I have to go onto my laptop, call them and ensure that they are in that place.

Lone workers spend a lot of time with different patients during the day. It would be great to have an update on their location throughout the day.

Do you have any other feedback/recommendations?

In the future I would like to see a mobile app developed for management. Since I've worked here we have scheduled on desktop. Personally I am constantly in meetings and on the road. If there was a mobile app developed for management also I would appreciate this. Sometimes it takes me a while to get back to staff and that can just be down to how busy I am. If I had an app, that I could check if a care worker had completed a job while out and about and could communicate through the app with them, this would save me a lot of time!

Interview with Care Worker:

What are your thoughts on having a mobile app to communicate with management in your company?

To be honest I think this would be great! As a lone worker it is difficult to get in contact with management at times. At the moment if I have a question about something, I would ring a manager, however sometimes this is quite uncomfortable in front of a patient. That is not the case, but when a manager does not respond to a text I have no other choice but to call.

What is the main thing that would benefit you as a lone worker from this type of application?

At the moment I view my schedule for the day online, I either jot this down on my phone or whip out my laptop to get to my next patient. It would be so much quicker, if I had an application to check. Also there is time's during the day where my schedule might change, I get a call off a manager and it rearranges my day.

If this was all available on an app, I think it would work better.

Do you think the scheduling and communication on the app is enough or is there anything else you would like to see on this app?

I think these are very important features. Maybe more information on the patient would be informative. It would be nice if there was a patient section where we could review the patients profile and what type of care they require.

A forum section could work well to, where carers and managers could chat in groups and questions could be posted between all types of staff members

Do you think this app would be easy to use while you are at work?

It would definitely be better than what is at place at the minute. I spend a lot of time on the phone communicating with my boss. Also spend time going through my schedules on my laptop. This seems like it would be much more efficient and easy to use.

9.2 Requirements - Order of Priority

1. *Care Worker viewing schedule on mobile app*

---this involves calendar view for the year

---carer clicks on date to view tasks that day.

---next screen contains date, patient name and start/end time of tasks for that day

2. *Care Worker communicating with management through the mobile app*

---this is a chat feature with a scroll where chat history can be viewed

3. *Manager creating/updating/deleting schedule's on desktop*.

---manager updates task and assigns to specific care worker.

---this information then must update to care workers app.

4. *Messaging system on desktop for Management to communicate with staff while in office/meetings.*

--similar to mobile app.

5. *Login Functionality*

---login username and password for Desktop

---on Mobile care worker to enter name , and from here on main menu will appear as care workers info will be saved to db.