ASHESI UNIVERSITY COLLEGE

A TASK SCHEDULING SYSTEM FOR SMALL & MEDIUM SCALE ENTERPRISES

ELYSIA URANIA AMARTEIFIO

2013

Applied Project

ASHESI UNIVERSITY COLLEGE

A TASK SCHEDULING SYSTEM FOR SMALL & MEDIUM SCALE ENTERPRISES

By

ELYSIA URANIA AMARTEIFIO

Applied Project submitted to the Department of Computer Science

Ashesi University College

In partial fulfillment of Science degree in Management Information Systems

APRIL 2013

DECLARATION

I hereby declare that this dissertation is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:……

Candidate's Name:…….

Date:……

I hereby declare that the preparation and presentation of the Applied Project Report were supervised in accordance with the guidelines on supervision of Applied Projects laid down by Ashesi University College.

Supervisor's Signature:……

Supervisor's Name:……

Date:……

ACKNOWLEDGEMENTS

First and foremost I would express my gratitude to my supervisor, Aelaf Dafla, for his direction and constructive feedback, which helped immensely in the development of this project. Your patience was amazing and you always encouraged me to think outside the box. For this, I am very grateful.

I would also like to express my appreciation to my colleagues, Emmanuel Nkansah, Sarpomaa Agyei-Boahene, Frank Anamuah-Koufie, Edem Diaba, and Emmanuel Odoi for being there throughout the difficult stages of the project and for keeping me on track.

Most especially, I would like to thank God for the grace and strength he provided me to embark on this project. Last but not the least, I wish to thank my family, for believing in me through out my study at Ashesi University. It has not been an easy journey but your support has kept me going.

ABSTRACT

High variety industries are characterized by low volume and high product variety. As such their production processes grow more complex with the increase in product variations. Hence, the use of tools such as project management and scheduling applications are important in ensuring efficient production runs. Though SMEs are present in such industries, they have been slow in the adoption of technology due to varied reasons.

This slow adoption is often attributed to high set up costs and the complex nature of such tools. This project involves the development of a task scheduling application that will be simple enough to incorporate into their operations and can help make the operations of SMEs in such industries more efficient. Additionally, this application employs a user-friendly approach in order to eliminate the initial cost of training and to help ease their adoption of technology.

Table of Contents

Chapter 1: Introduction

1.1 Introduction

The use of application software in businesses is increasingly becoming an essential component in its management and its success across varied industries. Generally, application software is designed to address the needs of end users and organisations. It can also be integrated into business operations to provide the necessary support and efficiency [1]. Some classifications of application software include application suites (e.g. Microsoft Office) or educational applications. In a world where customer satisfaction is key in a business' success, application software has proven to be a major factor in improving one's ability to effectively meet a customer's need. One of such ways is through the implementation of project management software designed to ensure timely delivery of products within a set of predefined constraints.

"With project management software, you have a central database for all your project information, rather than spreadsheets scattered over different employees' local hard drives" [2]. However a common practice in some business operations, especially small and medium scale enterprises (SMEs), is the use of guessing, based on past projects, to estimate delivery times. Though larger companies have embraced project management technology, the same cannot be said for SMEs. This is because of high initial costs involved in implementing such software [3]. This project first reviews Enterprise and Information Worker software that deals with managing organisations and individuals respectively. It also

looks at the development of an application that can merge some functionality from these two distinct application groups such that it can be easy enough for SMEs to adopt.

1.2 Industry Profile

My project focuses on small and medium scale industries that have a high variety in the range of products they provide. "In general, volume and variety have an inverse relationship. If a firm caters to a wide range of products then the production volume of each variety is likely to be low" [4]. Variety in this sense refers to "the number of alternative products and variants of each product produced in a manufacturing system" [4]. To illustrate further, a furniture company may produce only sofas and armchairs but there may be several variations of armchairs and sofas produced. As such orders must be made as requested by the customer. A key characteristic of high variety industries is the use of project management, planning and scheduling tools to manage production.

Since these industries earn their revenue from working on orders, these tools are of importance to them [4]. "Planning and scheduling production may become more complex on account of the variance in products offered and may call for better operations management practices [4]". Furthermore, it requires more production processes and a wider range of human skill set to make all the different products that have been ordered. Such industries include the fashion, building or construction and furniture industries, amongst others. It is imperative that deliverables are on time and meet specifications in order to retain clients. "UNIDO's Definition for

Developing Countries defines medium scale firms as those with 20 - 99 workers whereas a small scale firm has between 5-19 workers [5]. However, most SMEs in Ghana lack the necessary technology to manage the complexity that arises with producing variable products. Thus they are also marked by delayed orders and dissatisfied customers [6]. This project also looks at how the issue of project management for SMEs can be addressed by introducing a simpler scheduling application to help manage their productions.

1.3 Problem Description

According to Beth Walker et al. on the low use of technology by SMEs, "little has been progressed for very small businesses in the way of keeping up to date with current technological trends" [3]. Focusing on Ghana, smaller companies as compared to larger corporations have been slow in embracing the use of technology (specifically application software). "Fear of the unknown and lack of skills have also posited as reasons why the uptake of technology is less for small businesses " [3]. The notion that "size does effect the up-take of technology" [3] cannot be refuted because larger companies have shown to be more capable, in terms of cost and human resource, of handing such transitions. This slow adjustment is compounded further by the fact that it is cost intensive for SMSs to train their employees to use complex software that will help improve operations.

To steer more companies into embracing application software, it is imperative that simpler softwares are designed to ease their transition

into incorporating technology. Again, the nature of the high variety industry requires the use of project management and scheduling software to be able to provide timely orders. However, the hindrance of SMEs in doing so is compounded further by the complex nature of PM software that requires many decision variables to be carefully assigned in order to make valid assumptions and recommendations. The use of application software with SMEs generally ends with stocktaking. Most often, guessing is employed by the enterprises in estimating completion times of orders. From my personal experience, such techniques often result in delayed orders and this gives a clear indication of its unreliability. Underestimated production time results in delayed orders and unhappy customers. My project looks at the possibility of developing a simplified task scheduling application that is more appealing and cost effective for SMEs in Ghana to implement whilst being able to incorporate technology in their operations to help meet their production deadlines.

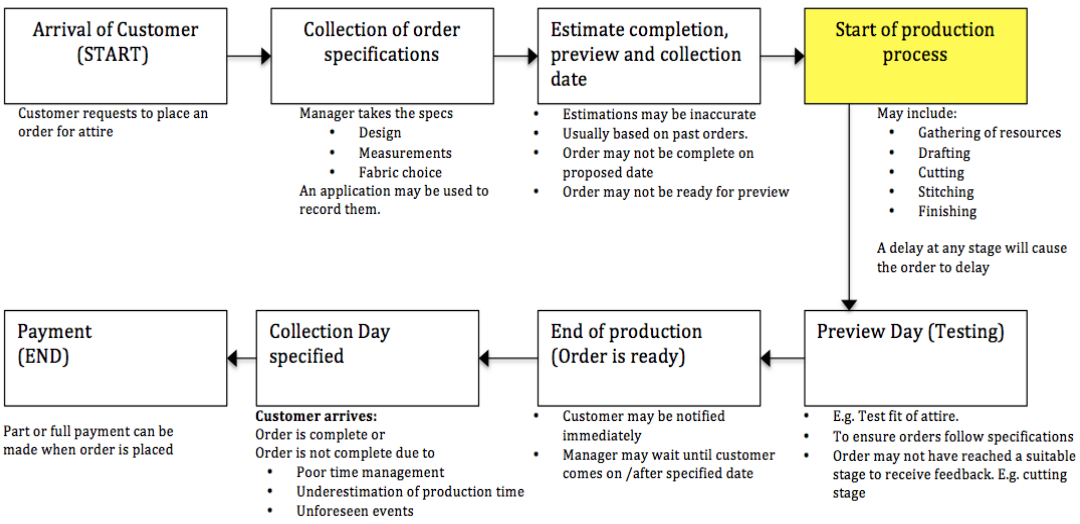


Fig. 1.1 General overview of operations of high-variety production industries.

Fig. 1.1 gives a general overview of the operations and activities that occur in high variety industries once an order has been placed. The comments under each operation/activity are examples of possible outcomes in the fashion industry. This sequence of activities is common to all industries characterized by high levels of customization. Software applications may be used for the collection of specifications however they are hardly used to estimate completion and collection dates of orders. The problematic phase is the production process, which tends to vary in duration with respect to a particular order. This stage determines the timeliness of an order because it involves the time-bound completion of each production process (e.g. Drafting and Cutting) in order to meet delivery deadlines. Further problems arise as workers are usually assigned portions of an order to handle as per the manager's discretion but do not have specific time ranges within which to complete such tasks.

As such, it is difficult to estimate which task needs more time or to determine which employees are delaying in completing their task hence delaying the entire production process. Dates for preview are usually not effective as they should be in tracking the progress of an order because the product may not have reached a satisfactory level to be viewed or to receive feedback. Finally, when an order is completed, customers are unaware unless explicitly called to collect their order. They have no means of checking the progress of their order without going in person. These issues call for the implementation of simpler programs to manage such activities and to make SMEs more receptive of technology and hence improve on the timeliness of orders.

1.4 Existing Solutions

Enterprise applications are used in large organisational environments and are heavily integrated into their business operations or large projects. Such applications include financial management, supply chain management or project management systems. "The main goal behind enterprise software is to improve enterprise productivity and efficiency through business logic support functionality." [7] However information worker software tend to focus more on an individual user and his management of information. It offers functionalities such as resource management and time management for smaller individual projects [1]. It is generally used for day-to-day tasks. Examples include word processing tools and spreadsheets like MS-EXCEL. Both software applications manage two extreme sets of users, which are large organisations and individual users. As such, there appears to be a gap with respect to the availability of an application small enough to be used by SMEs whilst having some of the functionalities of both application types stated above.

1.5 Objective

The purpose of this project is to create a simple task scheduling system that mainly monitors the **production process** of small-scale high-variety businesses. Tasks are the individual processes involved in making a product i.e. the production processes. The system will schedule and monitor individual tasks per order and track progress of the employees on each task until the product is complete. Below are the objectives of the project.

- To generate work schedules for tasks involved in each production process.
- To keep customers updated on the progress of their orders.
- To keep employees updated daily on their work schedules.
- To monitor employee progress on a task.
- To monitor progress of orders and ensure timely delivery of products.

It is imperative that the system is simple in its use because it will be targeted at people who do not have any prior experience with project management software. In order to address this issue across the various industries aforementioned, this application will be designed to be as generic as possible to allow for customization by various industries. The fashion and furniture industry will be the use cases for this project.

1.6 Motivation

My motivation for this project stems from personal experiences with delays in the completion of orders that I have placed. I find it increasingly frustrating and usually do not return to the offending vendor. Furthermore, my ambitions of entering the high variety industries discussed have forced me to think of creative ways to address the problems described without increasing my startup cost. Hence, the implementation of this simple scheduling program to monitor the production process and meet delivery deadlines. Also, I wanted to find a more appealing way to introduce management software to small-scale business that does not require professional training to use. My emphasis

is on simplicity in order to make the transition to technology based operations easier. This way I could help bring more business owners to incorporate application software in their work to make their operations more efficient and effective. I believe simpler IT solutions to business problems could make the use of technology in businesses much simpler, more appealing and cost effective for SMEs especially start-up companies.

1.7 Overview of Report

This report will highlight the implementation of key features of the application. Chapter 2 focuses on the system requirements for the development of this application and looks at various uses cases in relation to the requirements discussed. Chapter 3 looks at the application design, system architecture as well as the user interface design. Chapter 4 discusses the implementation of the application and the tools used in the development of the application. It also explains the reasons for using the selected implementation methods and tools. The chapter is concluded with the testing of the system and analysis of the test results. Chapter 5 concludes the report and gives recommendations for future work.

Chapter 2: Systems Requirements Specifications

2.1 User Requirements

The main users of the system are the managers and the employees. Customers may periodically access the system to check on the progress of their orders. The user requirements for the application are as follows

- The application must efficiently allocate the individual production tasks to be completed over an estimated range of production days.
- The application must indicate when the order will be ready and produce a production schedule to that effect.
- The application must assist in monitoring the progress of employees on the tasks they have been assigned and indicate the level of completion of the product.

2.2 System Requirements

The functional requirements were derived from the user requirements specified above. The system requirements are grouped as functional system requirements and non-functional system requirements.

2.2.1 Functional Requirements

The functional requirements indicate what the application should be able to perform

- The system should be able to perform an entire ordering process including recording order specifications and management of product catalogues.
- The application should estimate duration of an order and generate work schedules for the tasks involved in making the order.
- The application should assign tasks to employees as the orders are placed and give daily notifications of tasks to be completed.

- The system should allow for adjustments to the schedule to accommodate unforeseen events.
- The system should respond to customers' queries for progress reports and updates on orders.
- Users should be able to access their account via mobile web technology.
- The application should broadcast notifications of upcoming important activities to users e.g. collection and preview days.

2.2.2 Use Cases And Diagrams

The use cases and diagrams are used to further illustrate the functionality of the system and how the users will interact with it.

SCENARIO 1

A customer requests to place an order. The manager must collect the details of the customer's order in order to generate a production schedule for the order.

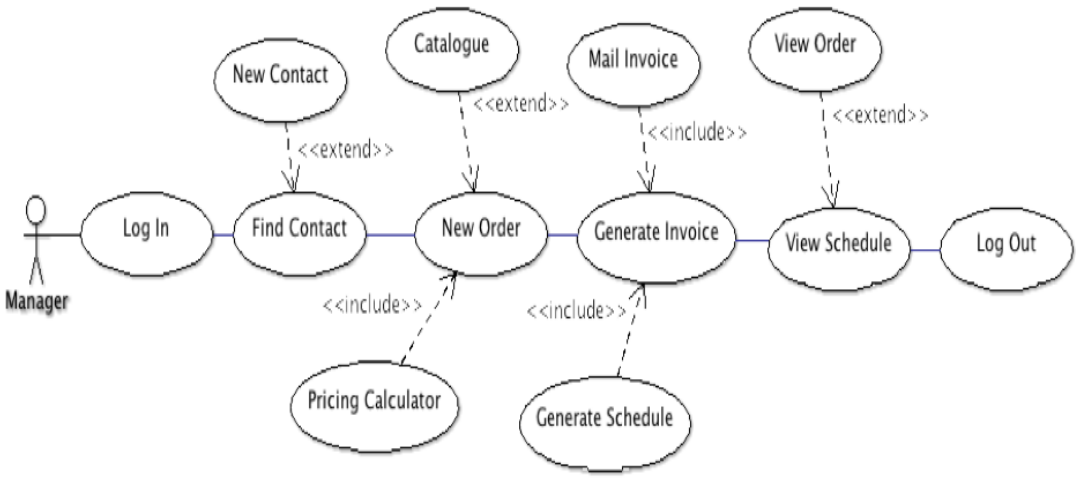


Fig. 2.1 Use Case: Scenario 1 for Managers.

USE CASE 1:

- Log In session: It involves the start session and validation process for the manager.
- Find Contact: The manager must either create a new customer profile or find an already existing one to begin an order-placement process.
- New Order: This option collects the order specifications and generates a work schedule based on the product type. Managers must assign employees to each production task.
- Invoice: This is generated once the order is finalized and is sent by SMS and email to the customer.
- Pricing calculator: This is an extension of the order book. It calculates the cost of the garment, including profits, based on predefined values.
- Production schedule option: Once the order is saved and the invoice is logged, the production schedule is displayed showing the employees assigned to a particular production phase.
- Log out: The session is ended and all transactions and updates are saved.

SCENARIO 2

The manager views the tasks to be completed for the day and checks on the progress for each one. He has approved some of the tasks done and wants to monitor the progress of the order and the employees assigned to it.

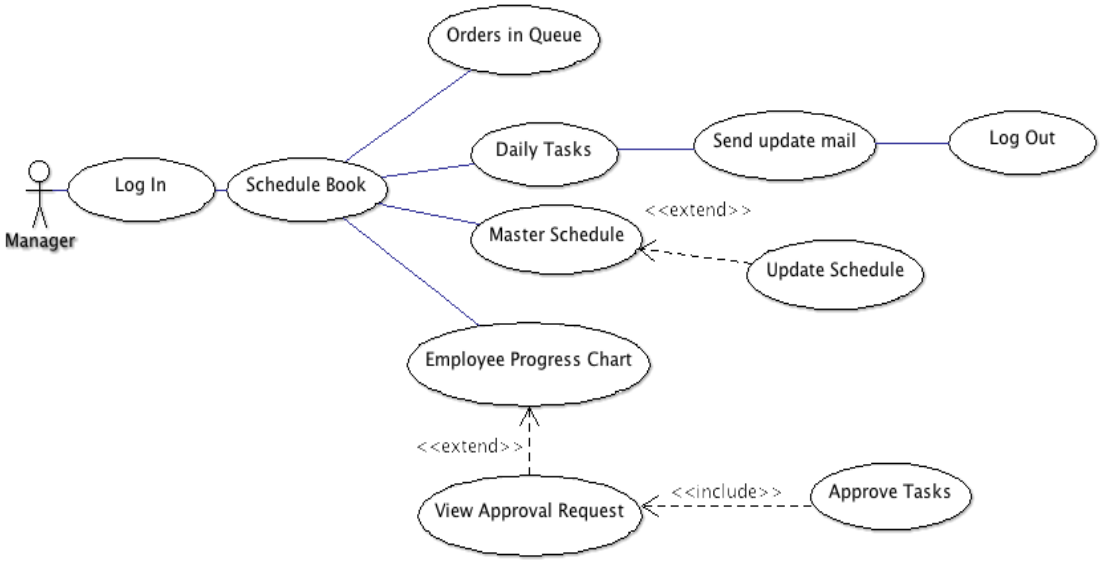


Figure 2.2 Use Case: Scenario 2 for Managers.

USE CASE:

- Work Schedule: This option allows the user to view the tasks for the day, the master schedule, orders in queue and the employee progress sheet.
- Daily task option: This gives an overview of the tasks each day. It can be sorted based on the employees or order numbers. It indicates the ones that have been completed.
- Jobs in queue: This shows the jobs to be done with the one nearest to its completion date having the highest priority and indicates completion rates.
- Approvals: This displays a list of request for task approvals[1] sent by employees. The manager must have physically inspected work done before giving approval. Once approval of all phases is completed an

[1] Approval refers to the manager giving consent that a task has been completed well. This triggers a schedule updates and notifies the employee assigned to the next process to commence. To ensure that preceding tasks are completed.

automated message is sent to the customer before the day of collection.

SCENARIO 3

The employees want to view the details of their tasks and request for approval.

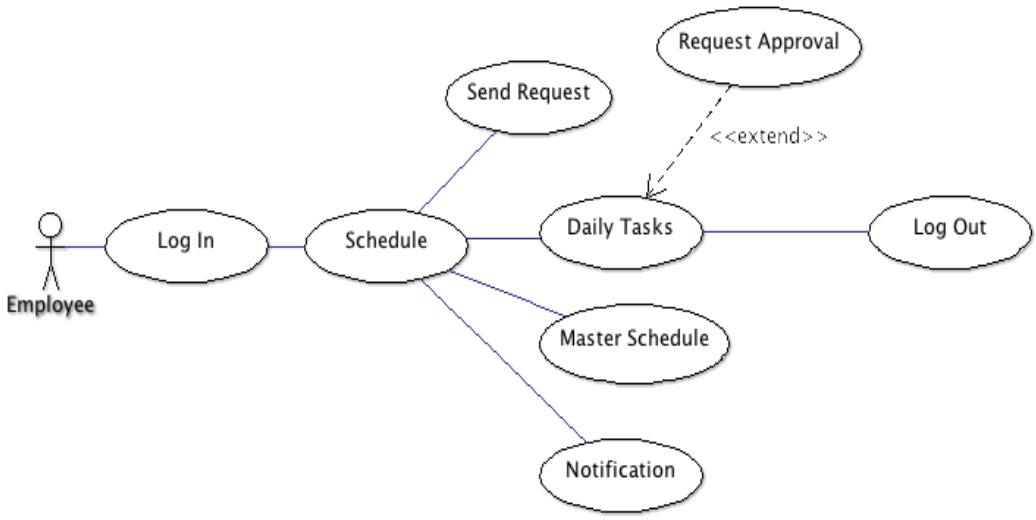


Figure 2.3: Use Case: Scenario 3 for Employees.

USE CASE:

- Schedule: Employees can view the tasks pertaining to them and cannot make any changes. Daily-automated SMS notifications of tasks assigned will be sent to them.
- Approval: Once a task is completed the employee is required to send a request for approval.
- Request for absence: Because the employee can view all tasks they have been assigned, they can also see the days they are free. This option allows employees to request for days off. However, the manager would have to approve these requests at his or her

discretion. Approval of leave exempts the employee from being assigned work on the said date.

SCENARIO 4

The customer would like to know the progress of their order and make requests.

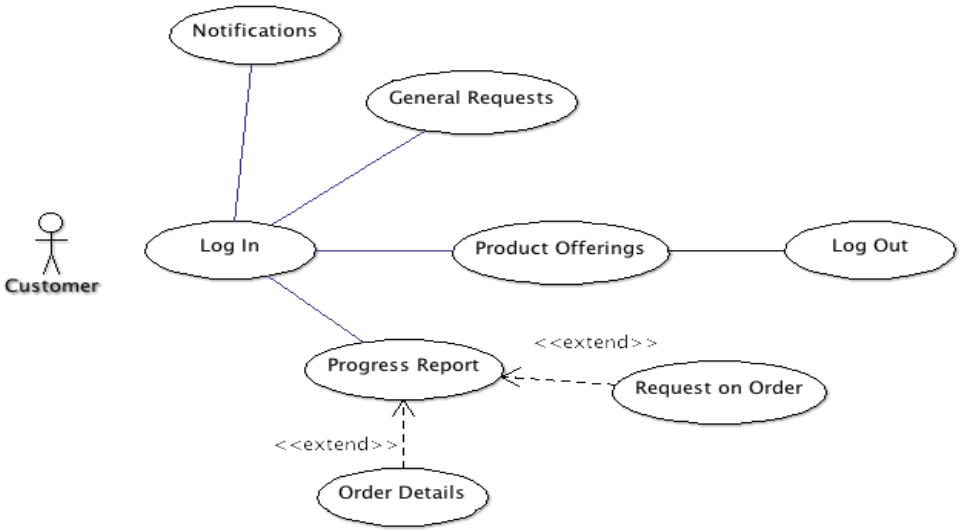


Figure 2.4: Use Case: Scenario 4 for Customers.

USE CASE:

- Schedule: Online schedule indicates test fit days, collection dates, and progress report for the customer's order. Short SMS can be sent to the system to retrieve such information.
- Product Offerings: This indicates available packages and offers by the company.
- Requests: The customer can make general enquires and request via SMS or the web application.

2.2.3 Non-Functional Requirements

These requirements are not related directly to how the system functions but it may pose as a constraint

- The application will be a web based application
- The application must have a user friendly interface and be easy to use

Chapter 3: Design

3.1 System Architecture

The system was designed in three layers, which include (from the bottom) the database layer, the application layer and the user interface layer. The database layer mainly deals with data management. This includes storing information keyed in by the users, performing update functions (i.e. updating, insertion and deleting) and providing data for queries run by the application. MYSQL Workbench was used to create the database and perform the functions aforementioned. The application layer serves as the bridge between the user interface and the database.

It runs on the web server and handles all the algorithms for various computations such as the generation of work schedules and cost calculations. The server side script is PHP and the AJAX model was used to implement this architecture. The final layer is the user interface layer that serves as the front end of the system and handles all of the inputs keyed in by the user. It also presents output when the user triggers specific events. The user interface is designed to be easy to use and has inbuilt validation procedures to ensure that the right kind of information is entered. For a more appealing interface, Twitter Bootstrap was used to design the interface.

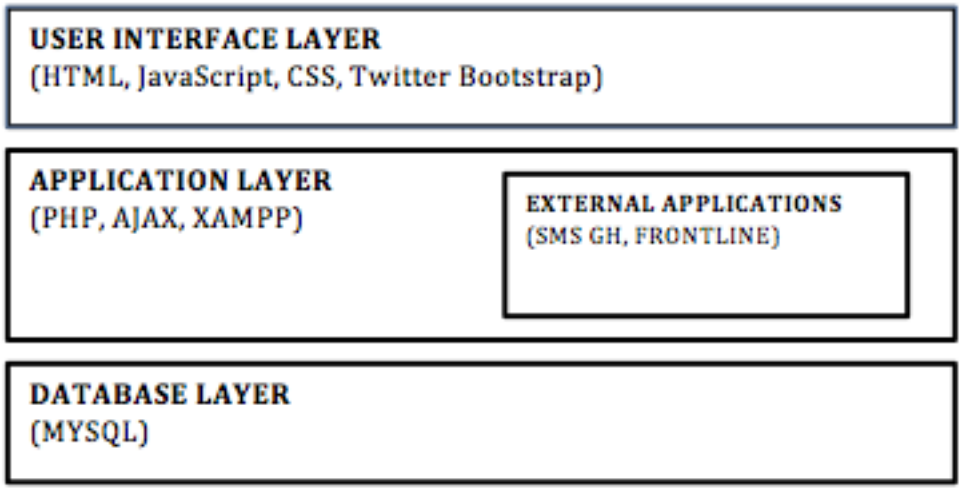


Fig. 3.1 Database stack implemented.

3.2 Database Architecture

In designing a suitable database, entity relationship models and enhanced entity relationship models were used to clearly map out the system requirements and develop a database that will efficiently manage the information it was fed. The main objective was to create a well-normalized database that would reduce update anomalies, reduce redundancies and data inconsistency.

3.2.1 Entity Relationship Model

The entity relation model helped to identify the main entities within the system and their relationships. It was necessary in developing the database because linkages between the entities needed to be established in order to clearly define and uniquely identify records. The diagram below show the relationships identified between the main entities.

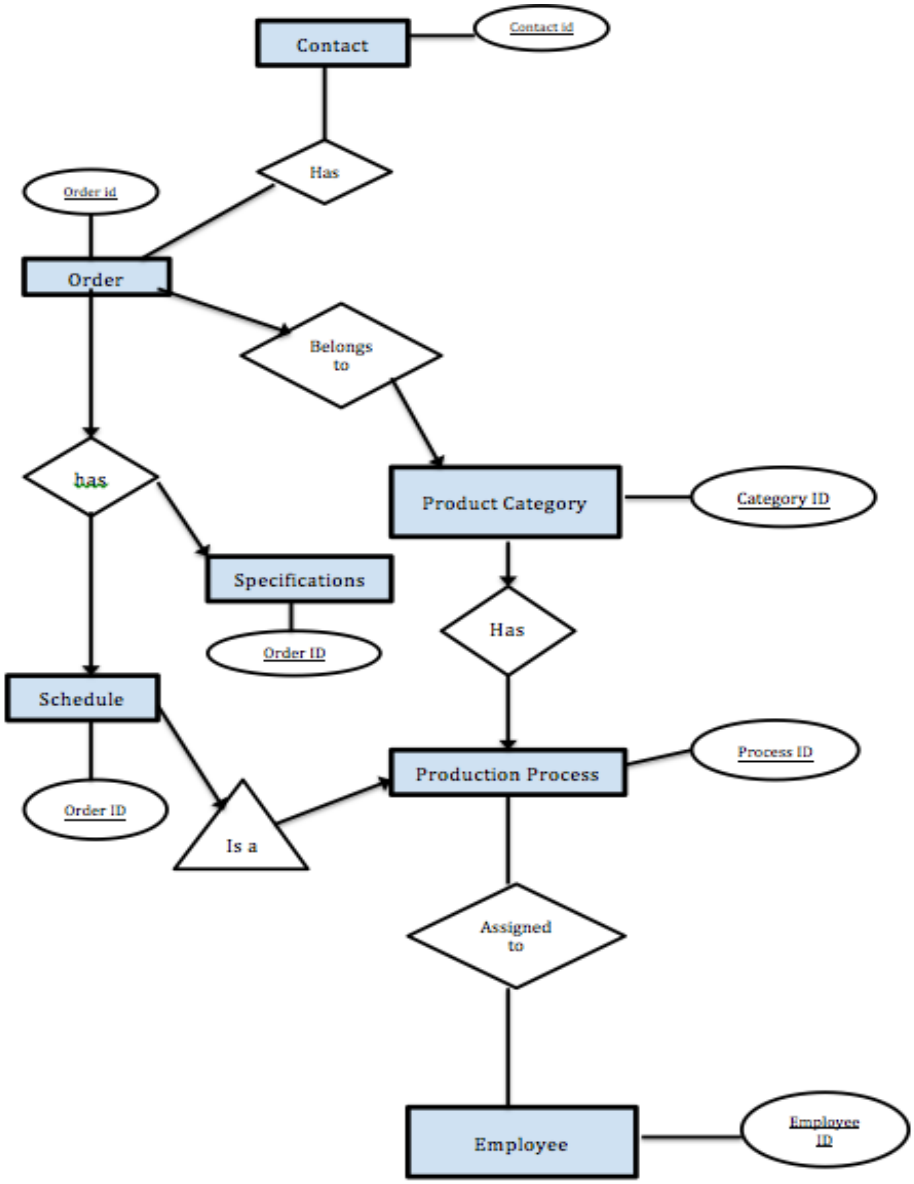


Fig. 3.2 Entity Relationship Model of the system

3.2.2 Database Schema

The database schema was based on the logic from the ER model and served as the blueprint for the implementation of the database layer. The design of the database schema was essential in helping to define the primary and foreign keys. At this stage the links between the various entities are established and are displayed graphically by the MYSQL

Workbench. Normalization was done at this stage to reduce anomalies and redundancies through the linkage of the tables. The decomposition helped to streamline the database to ensure absolute data consistency.

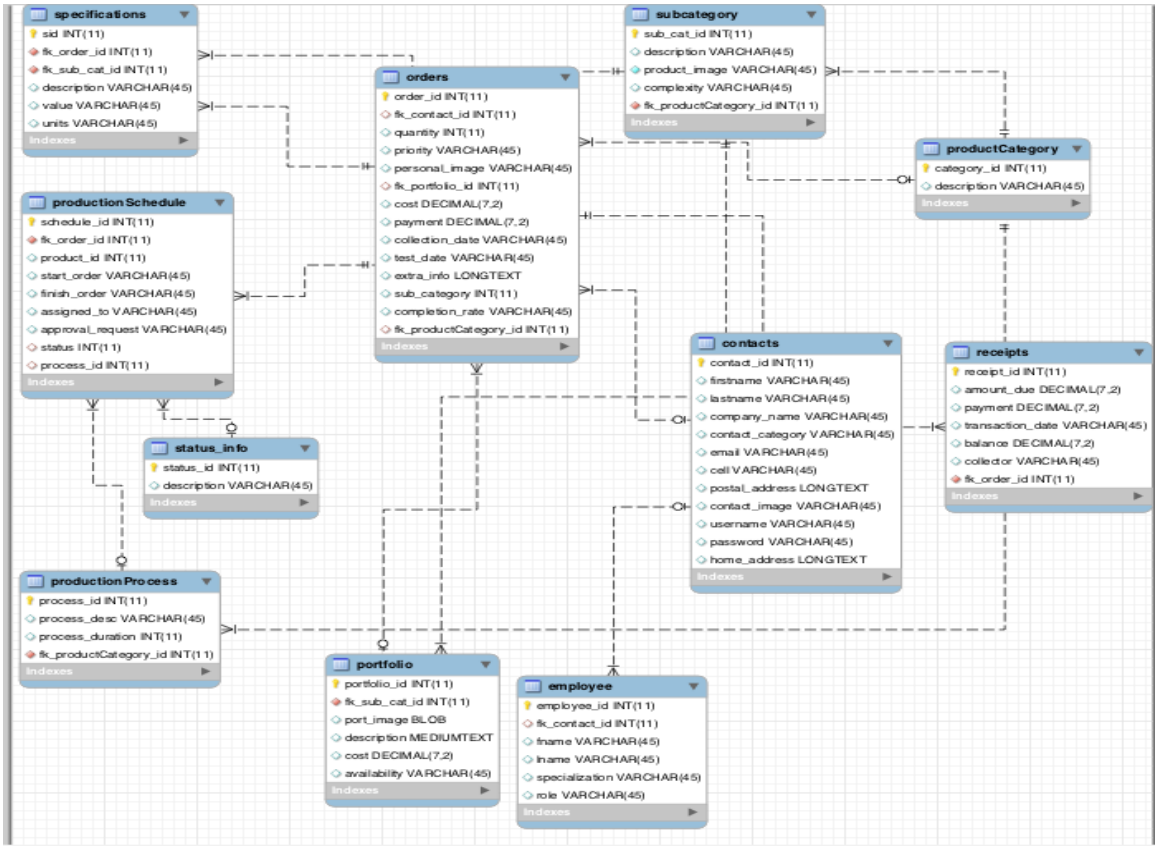


Fig. 3.3 Database Schema (Enhanced Entity Relationship Model)

Several changes were made to the original design in order to reduce redundancies. On of such changes was to the *productionProcess* table. The initial design was to hold every kind of production task in a record for a product (e.g. (process1_name, p1_duration, process2_name, p2_duration etc.). This proved to be inefficient because the production processes vary between products thus some fields would be null. This resulted in a redesign of the relation to give each production process a unique id and assign them to the ID of the product they belong to. Thus

to get the production processes of a product, the table will only have to be queried based on the product id.

3.3 User Interface Design

This application is mainly designed to manage and monitor the progress of orders. The users of this program are in three main groups, the managers, employees and customers. A bulk of the application is designed for the manager's use.

3.3.1 Manager View

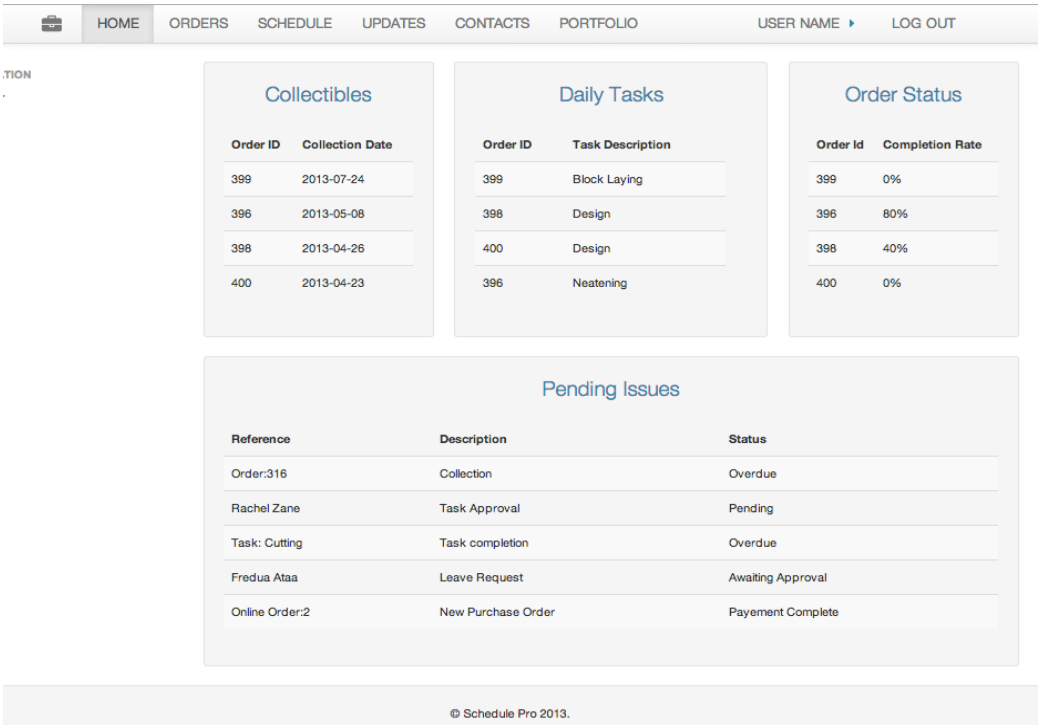


Fig. 3.4 Manager Home (Index).

At log in, the user gains access to the homepage according to the user group they belong in. The figure above shows the homepage of the manager's view after log in. This view gives brief updates on the progress of multiple orders that are on going. The manager can click on any row for more information. Four update previews where put on this page so that

the manager need not move through several pages to access them. It provides first hand view of the progress of orders and is suitable for monitory purposes. A simple interface was implemented inline with the objective of making the application easy to use.

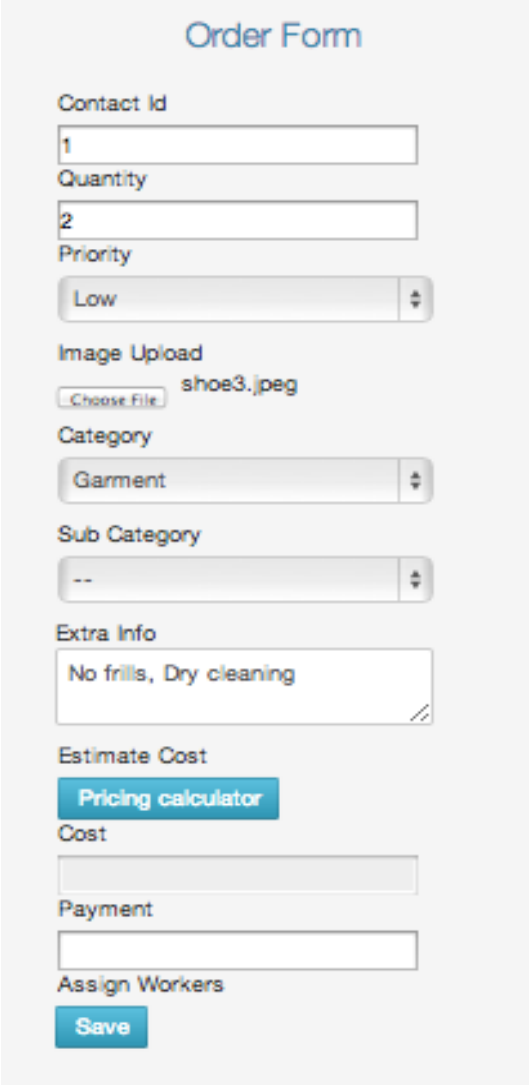


Fig. 3.5 User Interface for a New Order.

This interface takes the customers details and computes the cost of the order. Orders must be linked to a contact before it is taken (recorded). Images of preferred designs can be uploaded or selected from the

company portfolio. Once an order process ends, including specification collection, a schedule is generated.

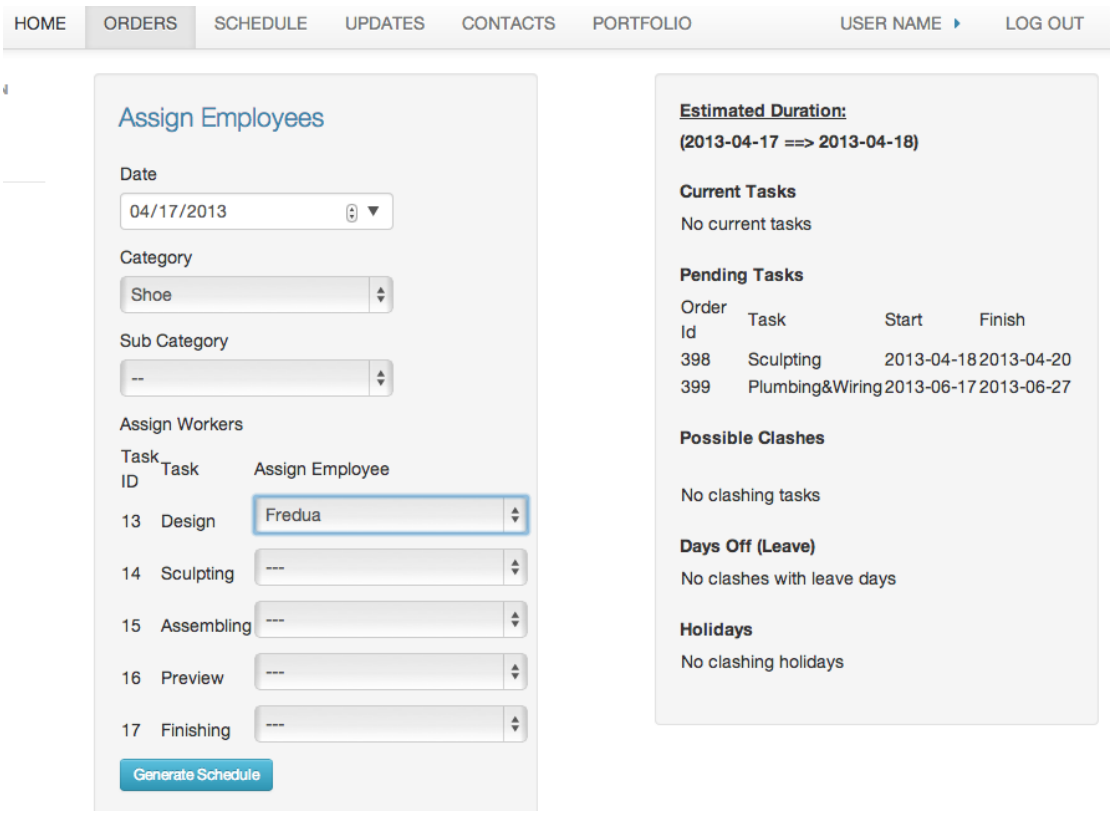


Fig. 3.6 User Interface for Assigning Employees

This page is included in the order process. On this page, the manager is required to specify the start date of the order and the general product group that it belongs to. Subsequently he must assign employees to each task. When an employee is selected, information regarding the current tasks assigned and possible clashes appears. This is to help the manager make a more informed decision about which task to assign to an employee whilst taking into consideration the possibility of clashes with other tasks and holidays.

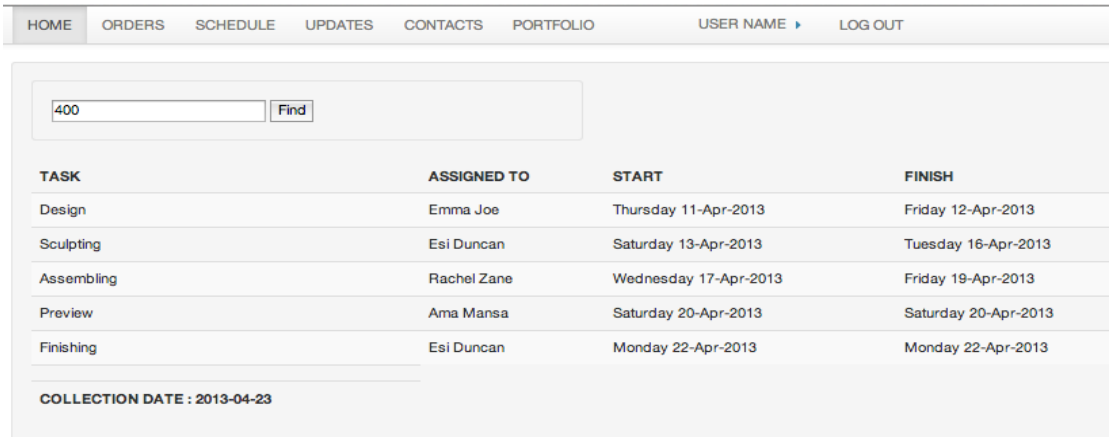


Fig. 3.7 User Interface for Viewing Work Schedules

The page above shows an important feature of the application. The work schedule of an order is immediately generated when an order is placed and it indicates working duration for the task. It also indicates which employee is assigned to each task. The purpose of this page is to start the monitoring processes for each task to ensure that they are done in time so that the customer is served on time.

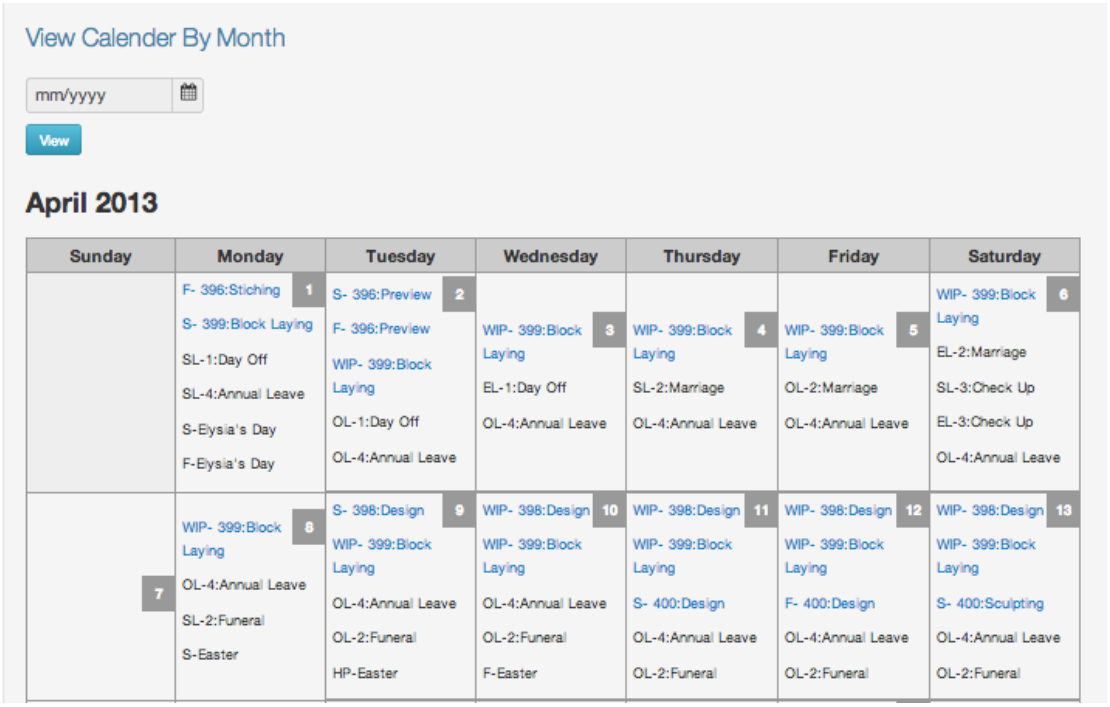


Fig. 3.8 Calendar view of tasks assigned (Manager View)

This interface was designed to give both managers and employees a view of their monthly schedules. In so doing, both users know the free days available at any given time. This view helps managers to decide when to edit the schedule in order to free up some days.

3.3.2 Employee View

Fig. 3.9 User interface to view tasks assigned to an employee

Upon log in, the employee gets to view the task assigned to them. Only tasks that have any preceding tasks completed will display. A request for approval can only be made when a task is complete. A simple button is used to send a request to the manager. This design is selected to minimize the amount of interaction needed to send messages to the system.

3.4 Software Quality Attributes

SIMPLE INTERFACE:

A simple interface design gives the application the professional outlook of project management software. However it also takes away the daunting and complex outlook of such software thus making users more willing to try it.

USER FRIENDLINESS:

The application requires minimal input from user and little assistance or training to run the application. This is to ensure ease of use for first time users.

INTEROPERABILITY:

This can be run on all browsers but functions at its best on Google Chrome and Firefox. It also runs well on both Windows and IOS operating systems.

Chapter 4: Implementation

4.1 Tools And Approach

The tools used in building this application were chosen based on the nature of my project. Being a web application, these tools were the most convenient for the development of this project.

(PHP) Hypertext Processor:

This tool is commonly used for server side scripting in developing web applications. Though I am conversant with it, PHP also has a lot of online support, which helps extensively in the development of this project. I chose this language because it was suitable for developing web applications and works well with the XAMPP server.

MYSQL

I used this tool in conjunction with PHP to manage and populate my database. I chose MySQL because of its ease of use and simplicity. The MySQL Workbench handled all the queries that were used in the functions during the project development. It also works well in conjunction with PHP.

Hyper Text Mark-Up Language (HTML) & JavaScript

HTML was used in the basic designs of the webpages. This included the use of various document elements such as tables, forms etc. JavaScript was used in conjunction with HTML to make the pages more dynamic and

interactive. As such changes made can be seen without having to reload the pages.

Twitter Bootstrap (CSS, JQUERY, JS)

This toolkit was selected to make the interface more appealing and interactive by using predefined styles in conjunction with JavaScript and jQuery. I chose to use the twitter bootstrap because it made the designing of the interface very easy in comparison with using just CSS. All that was required to implement or use the Twitter Bootstrap was to add the toolkit folders to the server files and call the predefined style using the class of the HTML elements. For example, to add an onmouseover hover effect to a table, the class attribute is declared as <table class="table table-hover">. The toolkit can be downloaded from http://twitter.github.io/bootstrap/

SMS GH & Frontline SMS

Frontline SMS is a text-messaging system that was primarily used to respond to "short codes" sent by users to the system. SMS GH API was also used to send the alert messages from the system to users (primarily employees). SMS GH was chosen for this purpose over Frontline because it was relatively cheaper to send bulk message with its API than through Frontline. Frontline, however, proved to be a faster option in responding to the "short-codes" than the SMS GH API.

4.2 Implementation Of Features

The application layers include external service components that connect the system to other applications as part of its process. External applications used include Frontline SMS and SMSGH Gateway API to handle all SMS queries sent to and from the system. The Apache web server was used to handle interactions between the database and the user interface.

4.2.1 Generation of a Work Schedule for an Order

The work schedules for an order are generated after the placement of an order is completed. The order process includes collection of contact details and the design specifications for the order and saving them into the *contacts* and *specifications* table respectively. During the order process the type of product must be selected so that the corresponding processes initialize the generation of the schedule. On the selection of a product (e.g. a garment, furniture), the list of production tasks is displayed. Similarly, the list of employees is generated into a dropdown for the manager to select which employee should be assigned to a particular process. The manager is required to specify the start date of production.

The working assumption for this program is that only Sundays are excluded from the working days and working hours are between 9am-5pm. In computing the schedule, two tables, *productionProcess* and *employees*, are used. The query is used to pick the duration of each task needed a particular order. For the first process, the duration is added to the date specified in the calendar. A check is implemented to ensure that the starting and completion dates do not fall on a Sunday. If a Sunday

falls within a production period (e.g. Saturday to Monday) the system counts the number of times its occurs and adds it to the completion date of the task. All subsequent processes start a day after the end of the previous one. This also includes the collection date, which is a day after the final task is completed. The employees selected for an order are assigned to the task they were allocated to by the manager. The schedule then displays for the manager to view. The code snippet shows how the computations were made.

```
            $sDate=$_REQUEST['startDate'];

        }
        //get start date entered
        $year =substr($sDate,0,4);
        $month =substr($sDate,5,2);
        $day =substr($sDate,8,2);

        //if the start date is a sunday, start from monday
            $check=date("w",mktime(0,0,0,$month,$day,$year));
            if($check==0){
                $day=$day+1;
        }
        // assumption working hours are  9-5 not 24 hrs
        //get the processes for that type of order e.g processes for a
shoe
        $mydb_schedule->getProcessInfo($fk_productCategory_id);
        $row = $mydb_schedule->getRecord();

        //set starting date. Intial values are entered by the user
        $currentDate =date("Y-m-d",mktime(0,0,0,$month,$day,$year));

        //== get number of processes
        $mydb2_schedule->countProcess($fk_productCategory_id);
        $num=$mydb2_schedule->getRecord();
        $countProcess="{$num['num_processes']}";

        // get assigned workers
        $employee_arr =$_REQUEST['employee'];

        //get order id
        $mydb4_schedule->orderID();
        $row4=$mydb4_schedule->getRecord();
        $oid="{$row4['order_id']}";

        //while fetching each task for an order , assign dates and
employees
        $emp_id=0;
                while($row){

                //query for a process duration
                $x="{$row['process_duration']}";

                //start a new process , the day after one ends
                //the start date is inclusive in calculating the work days
                $newDay=($day+$x-1);

                //generate end date of a process
```

```
                //if the end date is a sunday, end on monday
                $check=date("w",mktime(0,0,0,$month,$newDay,$year));
                    if($check==0){
                        $newDay=$newDay+1;
                    }

                //generate the new date
                $newdate=date("Y-m-d",mktime(0,0,0,$month,$newDay,$year));
                $x=$day;
                $count=0;
                while($x<$newDay){
                    $gg=date("Y-m-d",mktime(0,0,0,$month,$x,$year));
                    $check=date("w",mktime(0,0,0,$month,$x,$year));
                    if($check==0){
                        $count++;
                    }
                    $x++;
                }
                //count the number of sundays falling in between as add
them to the days
                $sundays=+$count;
                $sum_days=$newDay+$count;
                $newdate=date("Y-m-d",mktime(0,0,0,$month,$sum_days,$year)
);


                // number of employee = number of processes
                $employee_assigned=$employee_arr[$emp_id];
                $process_id="{$row['process_id']}";

                // save the start and end date
                $mydb3_schedule->saveSchedule($oid,$fk_productCategory_id,
$process_id,$employee_assigned,$currentDate,$newdate);

                $emp_id++;

                $value= date("Y-m-d",mktime(0,0,0,$month,$sum_days,$year))
;


                $d =substr($value,8,2)+1;
                $m =substr($value,5,2);
                $y =substr($value,0,4);

                //if the start date is a sunday, start from monday
                $check=date("w",mktime(0,0,0,$m,$d,$y));
                    if($check==0){
                        $d=$d+1;
                    }

                    $currentDate =date("Y-m-d",mktime(0,0,0,$m,$d,

                    $day=$d;
                    $month= $m;
                    $year=$y;


                    $row=$mydb_schedule->getRecord();
                    $row2=$mydb2_schedule->getRecord();
                    }


            $check=date("w",mktime(0,0,0,$month,$day,$year));
                if($check==0){
                    $day=$day+1;
                }

            //estimate collection date (+1 after finish day
            $collection=date("Y-m-d",mktime(0,0,0,$month,$day,$yea

            //update order to include preview and collection dates

            $mydb3_schedule->preview($oid);
            $row=$mydb3_schedule->getRecord();
            $begin="{$row['start_order']}";
            $end="{$row['finish_order']}";
        $mydb2_schedule->putDates($oid,$collection,$end);
```

Fig. 4.1 Code Snippet of Scheduling Algorithm

4.2.1 Display of Individual Employee Tasks

As a work schedule is generated, employees are assigned simultaneously. Employees are only allowed to view assigned tasks that have preceding ones completed. When a task is completed the employee sends an approval request to the manager. An approval request is an indication to the manager that a task has been completed and requesting him/her to

view the work and give the employee the go ahead to move on to the next task. Once a task is approved, its status is changed to “Complete” and it allows for the employee to view the next task he has been assigned to. This is to make sure that tasks are given in an orderly manner. The code snippet shows the algorithm used to display the tasks.

```
$mydb->getTasks($workerID);
$row=$mydb->getRecord();
$today = date("Y-m-d");

if(!$row){
    echo"<small><center><h2>Today's Task</h2></center>";
    echo"<table class='table table-hover table-striped'>
    <thead>
        <tr>
            <th><span>No for Today </span></th>

        </tr>
    </thead>
    <tbody>
    </table></small>";

    }
else {

echo"<small>
    <table class='table'>
    <thead>
    <tr>
        <th class='text-info'>Order </th>
        <th class='text-info'>Description</th>
                <th class='text-info'>Start </th>
                <th class='text-info'>Finish </th>
                <th colspan='2' class='centered-content text-
info'>Action</th>
            </tr>
            </thead>
            <tbody>";

        echo"<form action='employee_tasks.php' method='post'>";
        while($row){

            $orderID = $row['fk_order_id'];
            $processID = $row['process_id'];
            $task_desc = $row['process_desc'];
            $start = $row['start_order'];
            $finish = $row['finish_order'];
            $request=$row['approval_request'];

            $mydb2->getTasksByOrderID($orderID);
            $array_process=array();
            $array_status=array();
            $i=0;

            $row2=$mydb2->getRecord();
            while($row2){
                $array_process[$i]=$row2['process_id'];
                $array_status[$i]=$row2['status'];

                $i++;
                $row2=$mydb2->getRecord();
            }

            //if your task is the first in the process, check if it i
incomplete and display
            if($array_process[0]==$processID){
                    if($array_status[0]==0){
                        if($today>=$start){
                        echo "<tr>";
                            echo "<td><a
href='employee_orderDetails.php?order_id=$orderID'>$orderID</a></td>";
                            echo "<td>$task_desc</td>";
                            echo "<td>$start</td>";
                            echo "<td>$finish</td>";


                        if($request==0){
                            echo "<td class='centered-content'>
                            <input id='reqBtn' class='btn btn-info
```

```
btn-mini' type='submit'
                                onclick='onReq($orderID,$processID)'
value='Request Approval'/></td>";
                                echo "</tr>";
                              }
                            if($request==2){
                                echo "<td class='centered-content'>Reque
Sent: Awaiting Approval</td>";
                            echo "</tr>";
                            }
                            if($request==4){
                                echo "<td class='centered-content'>Reque
Denied: Resend Request</td>";
                                echo "<td class='centered-content'><inpu
class='btn btn-danger btn-mini' id='reqBtn' type='submit'
onclick='onReq($orderID,$processID)' value='Request Approval'/></td>";
                            echo "</tr>";
                            }

                        }
                        }
                    }

            for($z=1;$z<count($array_process)+1;$z++){

                    if($array_process[$z]==$processID){
                        if($array_status[$z-1]==1){
                            if($today>=$start){
                                if($array_status[$z]==0){

                                    echo "<tr>";
                                    echo "<td><a
href='employee_orderDetails.php?order_id=$orderID'>$orderID</a></td>";
                                    echo "<td>$task_desc</td>";
                                    echo "<td>$start</td>";
                                    echo "<td>$finish</td>";


                                   if($request==0){
                                    echo "<td><input id='reqBtn' type='submi
onclick='onReq($orderID,$processID)'
                                     value='Request Approval' '/></td></form
;
                                    echo "</tr>";
                                  }
                                if($request==2){
                                    echo "<td>Request Sent: Awaiting
Approval</td></form>";
                                    echo "</tr>";
                                    }
                                    if($request==4){
                                        echo "<td>Request Denied: F
</td>";
                                        echo "<td><input id='reqBtr
onclick='onReq($orderID,$processID)'
                                        value='Request Approval'/>
                                    echo "</tr>";
                                    }


                                        echo "</tr>";
                                    }
                                    }
                                }
                            }
                        }

                        $row=$mydb->getRecord();
                    }
                    echo "</form>";
                    echo"</tbody>
                    </table></small>";
                    }
?>
```

Fig. 4.1 Code Snippets of the Algorithm to Display Daily Tasks.

4.2.2 Processing of Employee Requests

Employees have the opportunity of viewing all the tasks they have been assigned to. However, they cannot access the details of the respective

orders until the system recognizes that a preceding task, if any, has been completed. This gives the employee an overview of his free days and hence the opportunity to request for leave on those days. If a leave date is approved, should the manager attempt to assign him to a task within that period, an alert will popup indicating a clash with the approved leave date.

4.2.3 SMS Notifications

The system makes use of SMS notifications to keep employees updated with their work schedules and to keep customers informed about the progress of their orders.

ORDER PROGRESS:

Customers can send the ID of their order (1: [order id] e.g. 1:315) as a short code and receive an update on the completion rate and the collection date of the order. Senders are verified based on their cell numbers to make sure the message is coming from a valid customer. It is important to use the number saved in the system to send the short code. Completion and preview dates are automatically sent by the system to clients.

ORDER SPECIFICATIONS:

Once an employee begins a task, he or she can send a short code (1: [employee id] e.g. 1:040) to get the specifications of that order. If the SMS cannot send all the specifications, the employee will be directed to the mobile web component of the system to view all. The source of the

SMS is verified to make sure employees only access specifications through SMS.

DAILY TASKS SMS:

Employees receive a text message daily on the tasks for the day. This ensures that all employees can be accessed and they know their schedules. The push SMS is triggered when the system identifies the time to be 8:00am.

4.2.4 Mobile Web Component

This is a limited view of the system and is designed to give the users access to information that cannot be sent by text. For example, some specifications cannot be sent via text. This gives flexibility to the manager and to move around and inspect tasks while remotely approving them and updating the system. It will be mainly used to view schedules and to send requests. The application uses device detection to detect if it is a phone and to switch to the mobile component accordingly.

See Appendix 6.2 for screenshots of the interfaces.

4.2.5 Business Intelligence

Business intelligence is essential in making the monitoring interaction with the systems better. This helps to make the system more useful to the manager. In order to incorporate business intelligence into the system, the manager has an overview of all the tasks to be completed. Once a task is a day from its completion date and has not been approved as a completed task, the manager is alerted to the possibility of its delay so

that the necessary action can be taken. This alert is executed by picking the finish date of a process and setting an alert to go of the day before if the activity has not been logged as complete on that day. If a task delays, the manager receives a red flag until the order is completed.

4.3 Operating Environment

All web browsers support this web application but it runs best on Google Chrome or Firefox. It can be hosted on the local server of the PC.

4.4 Testing

4.4.1 Compatibility Testing

This test was to verify whether the system is compatible with different browsers. The browsers used were Firefox, Chrome and Safari. It was tested across two operating systems, namely IOS and Windows, to ensure its interoperability. These two were selected due to its popularity with users. The system worked successfully on both operating systems. However the use of the html input tag (attribute type='date') to collect dates did not work in the Firefox browser. As such I had to use bootstrap to append the date calendar to the input tag to make it work.

4.4.2 Functional Testing

This test was to ensure that the system meets the requirements specified in Chapter 2. I took each specification defined and checked if it was implemented and whether it met its specifications. I also checked to ensure that the output of these features was valid. This test was specifically carried out on the schedule to ensure its accuracy. All features specified worked well.

4.4.3 Usability Testing

This test checked the user friendliness of the application. To do so I asked a range of students across the various year groups to use the system and give some feedback on their usability of the system. Feedback from the users indicated that it was simple enough to use however more colour could be added to the user interface to make it more appealing.

4.4.4 Scenario Testing

Given the use case scenarios stated in Chapter 2, I wanted to ensure that the users could perform all the functions within the specified scenario. The system was tested against each scenario to ensure that it offered the functionality specified in Chapter 2. The system functioned as expected in all scenarios.

4.5 Challenges

I faced a few challenges especially with regards to functions that depended on the schedule. The first challenge encountered was the processing of request. The system should send an SMS whenever a request from a user or employee is responded to. Initially I wanted to put all the requests on a page with navigation tabs to eliminate the need to move through various pages (using JavaScript). This however proved to be very difficult because the Ajax function just kept generating errors. I was forced to split the code onto four pages. Though it was less efficient, the code run perfectly. A general challenge was with the use of Ajax and JavaScript to eliminate the need to reload pages. The function for passing

information via the url tended to randomly encounter errors. For most instances, the problem was resolved but for some just I had to take out the JavaScript and use only PHP.

Chapter 5: Results, Recommendation and Conclusion

5.1 Benefits (Results)

- The system can record orders and make a production schedule for that order.
- Employees can receive daily updates of their work schedules.
- Managers can monitor the progress of their orders.
- Alerts help managers to stay ahead of production to ensure that all potential causes for delays are addressed.
- The system's simple interface makes it very easy to use.
- Employees are well informed of the progress of their orders.

5.2 Recommendations For Future Work

Currently the project only has a mobile web component. I would suggest the development of a mobile application as an extension of the project. This is because it would make the access to the features easier. For example, accessing your daily schedule would be easier on a mobile application because it can update its self instead of receiving daily SMSs from the system. Also the algorithm for scheduling can be improved to accommodate requests for express orders and to take into account the possibility of two production processes running simultaneously.

5.3 Conclusion

This project sought to provide SMEs with a simpler project management application suitable for their transition into technology based operations and inferences from the test result show that it has been achieved. This

task scheduling application if implemented would offer basic project management assistance to managers in the industry,

Bibliography

[1] Techopedia. (2013, March) Techopedia. [Online]. http://www.techopedia.com/definition/4224/application-software

[2] Miree Le Roy. (2011, March) IPM Global Website. [Online]. http://ipmglobal.net/ blog/The IPM Blog/post/5 advantages of project management software/

[3] Beth Walker, Shirley Bode, Janice Burn, and Beverley Webster, "Small business and the use of technology: why the low uptake?," in *Small Enterprise Association of Australia and New Zealand Annual Conference*, 2003, pp. 1-9. [Online]. http://www.cecc.com.au/programs/resource_manager/accounts/seaanz_papers/72WalkerSmallbusinessandtheuseoftechnologyfinal.pdf

[4] B. Mahadevan, "Design of Manufacturing Proceses," in *Operations Management: Theory and Practice*. New Delhi, India: Pearson Education, 2010, pp. 104-115. [Online]. http://books.google.com.gh/books?id=pq4k3EaFzMYC&pg=PA105&lpg=PA105&dq=characteristics+of+high+variety+low+volume+industries&source=bl&ots=_EsJG4KEWy&sig=gY9lMwei6yY_8XRnTq4i9LIC4B4&hl=en&sa=X&ei=ejVYUeeCL9Gu7Aady4CYCA&ved=0CE0Q6AEwBQ#v=onepage&q=characteristics%20of%20high%20variety%20low%20volume%20industries&f=false

[5] Dalitso Kayanula and Peter Quartey. Business Environment. [Online]. http://www.businessenvironment.org/dyn/be/docs/60/fdwp15.pdf

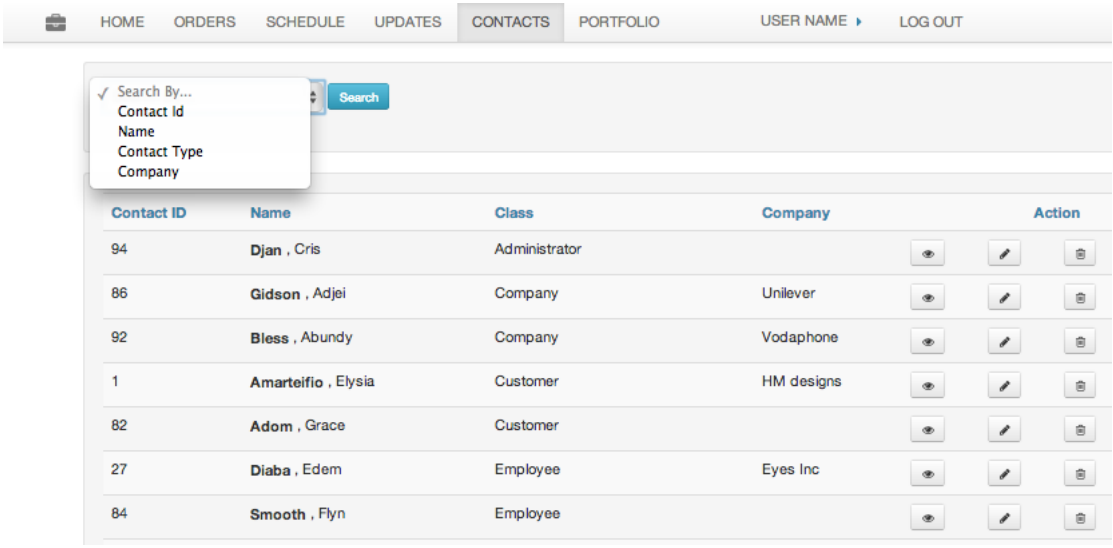
[6] (2005, August) Ghana Cedi Website. [Online]. http://ghanacedi.com/business/business-in-ghana/6648-small-and-medium-enterprises-urged-to-embrace-technology-to-cut-down-cost

[7] Techopedia. (2012, March) Techopedia Website. [Online]. http://www.techopedia.com/definition/7045/enterprise-software

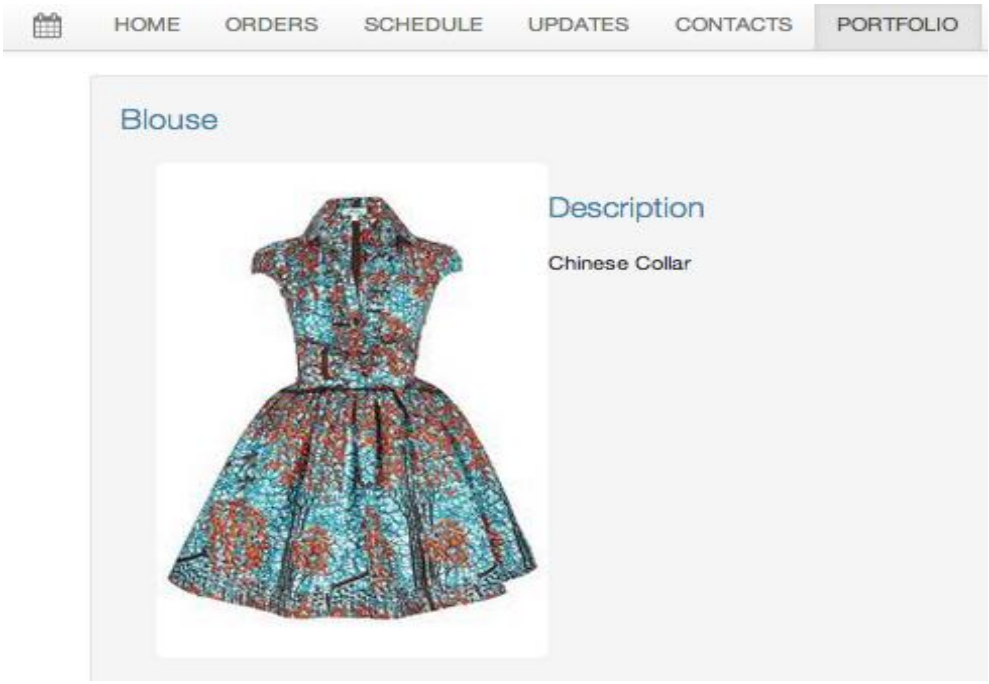
Appendix

6.1 WEB APPLICATION INTERFACES

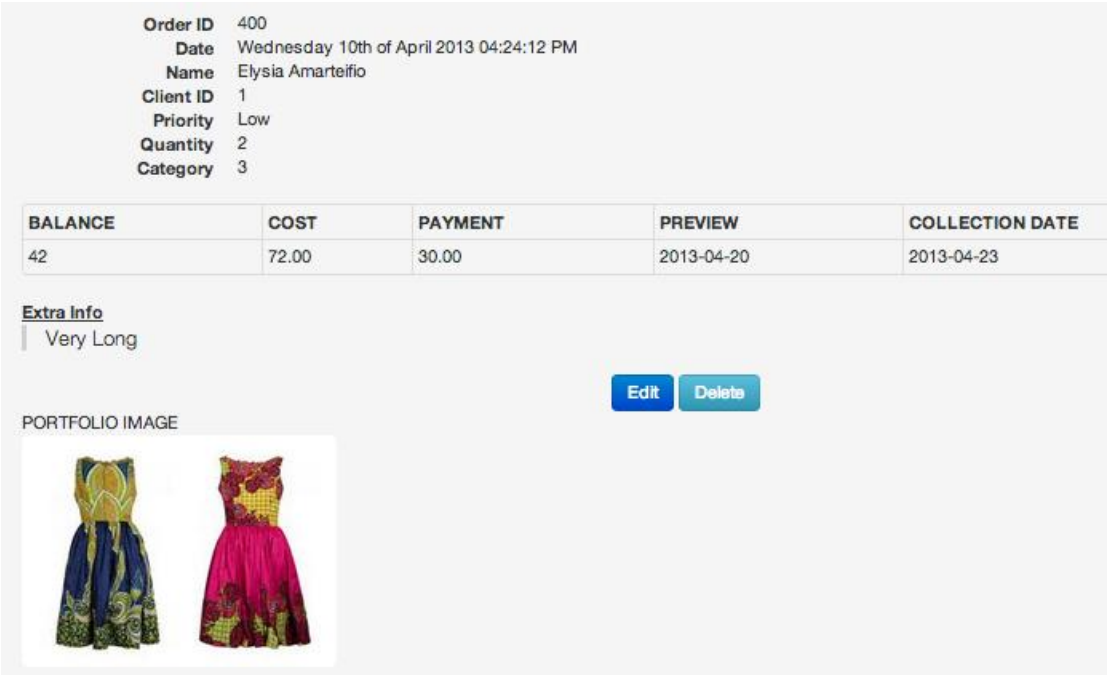
6.1.1 Appendix 1: Contacts book



6.1.2 Appendix 2: Portfolio view for both managers and customers



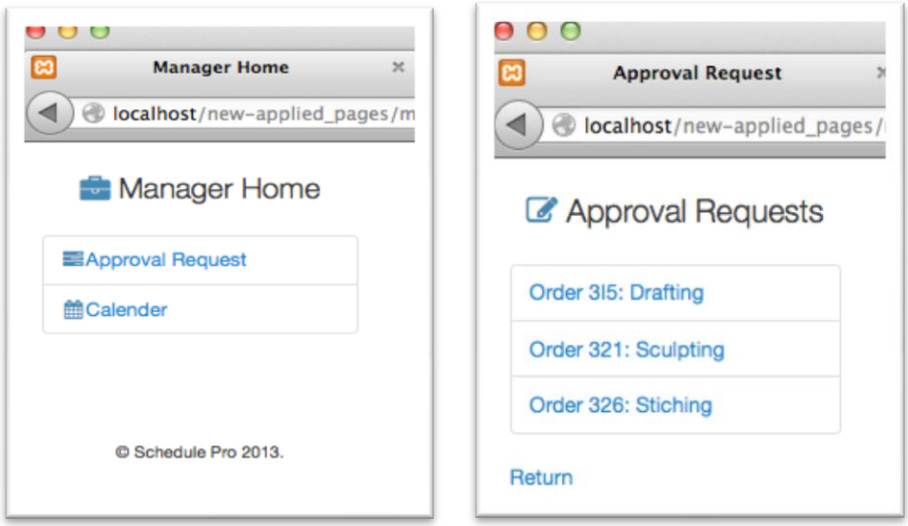
6.1.3 Appendix 3: View order details

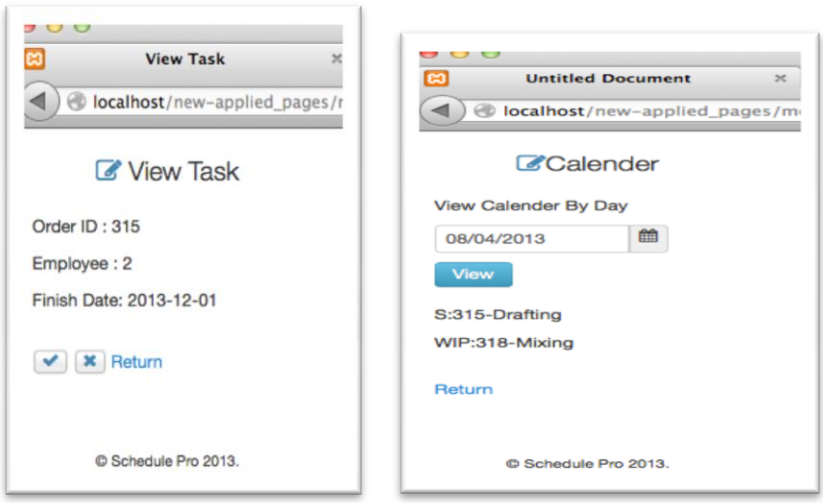


6.2 MOBILE WEB COMPONENTS

6.2.1 Appendix 4: Manager View

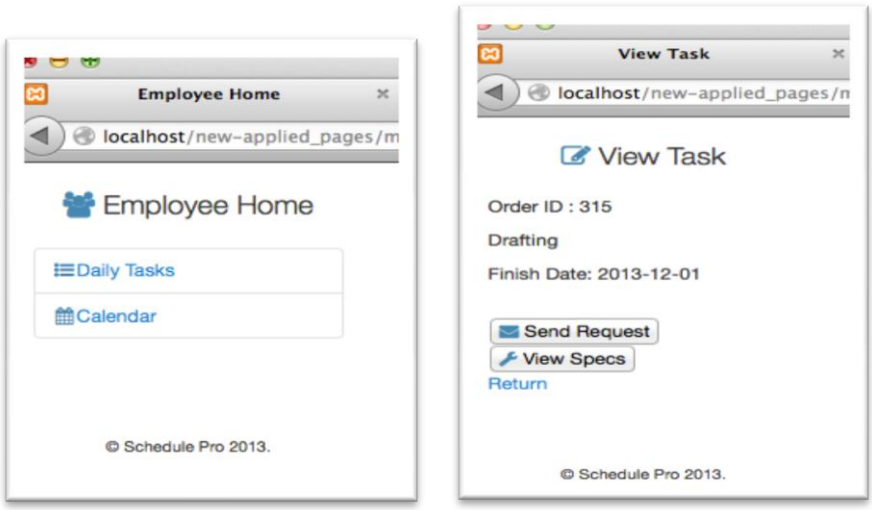
The manager can approve task requests and view the calendar via the mobile web component.

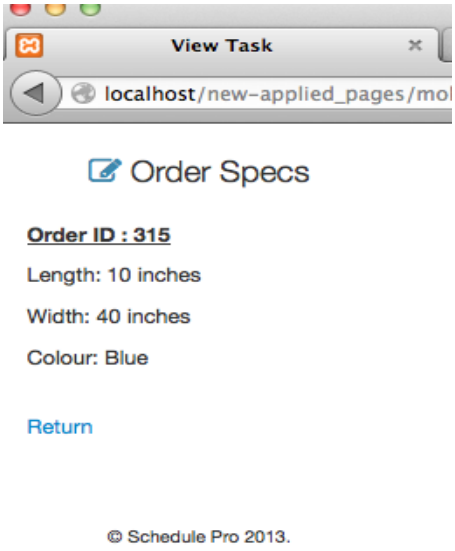




6.2.2 Appendix 5: Employee View

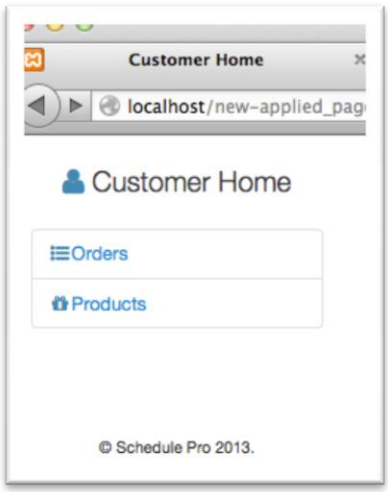
The employee can send approval requests for tasks completed. Similar to the manager, they can view details of the order and its specifications. The calendar is also included in the mobile component.

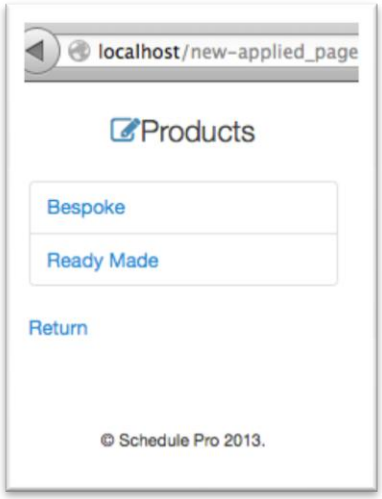


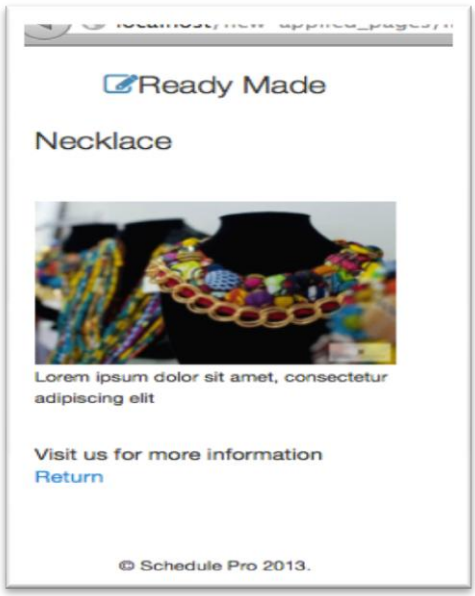


6.2.3 Appendix 6: Customer View

The customer can access the details of all orders placed. This component also allows for the viewing of products available.







A copy of the software, its documentation and supporting applications can be found on the CD submitted with this document.