National College of Ireland

BSc in Business Information Systems

2016/2017

Ryan Lawlor

BSHBISE4

x13379761

x13379761@student.ncirl.ie

Ryan’s Rosters

Table of Contents

Executive Summary

Ryan's Rosters is a system designed for desktop use across all browsers with one main aim. The main aim of this project is to be the best rostering system available for public use and be of the highest possible quality. From research performed on the market, the competitor's solutions all require manual input from the user in order to create a roster for the company and its workers. Ryan's Rosters will allow for manual creation of rosters but the main Unique Selling Point (USP) of this system is that user who more than likely will be the manager can automatically generate rosters for their workers that are stored on the system and then send the rosters to the employees through email which will allow the users to access rosters on their mobile phones for quicker access to their weekly and monthly rosters. These email addresses and phone numbers will also be stored on the system so security is a big requirement to ensure only users who should see this information see it! In order to implement this security, strict authentication will need to be in place in order to grant access to the appropriate users.

Ryan's Rosters will also have a feature implemented to allow shift swaps as we know that employees may not always be able to cover the shift that was assigned to them. The shift swap will allow the employee to request a shift swap with a colleague and the request will be sent to the manager who can approve or reject the request.

The premise of the application is simple but the execution won't be so simple. The system must be easy to use, high quality and do the job intended. It will be designed using ASP.NET MVC to get a good feel of the UI and UX. It must be quick and the roster auto-generation must be fast and stable! There are other rostering and shift schedulers available to Ryan's Rosters aims to be of the highest quality with the features such as auto roster generation and shift swap available from the get go.

1 Introduction

1.1 Purpose

The purpose of this document is to set out the requirements for the development of company rostering system application. The purpose of the application is to be used by companies to create rosters for weekly or monthly use. The intended customers are Irish companies that operate on a rostering system such as local supermarkets or pubs as they usually have a small number of employees so tracking their rostering needs through Ryan's Rosters will be efficient and effective. The application will eliminate the use of physical rosters that are hung on noticeboards as the application will send the rosters to employees via email to employee's phones for easy access.

1.2 Project Scope

The scope of the project is to develop a software system to generate and upon generation give the option to email employees their roster for the week, fortnight or month depending on which way the customer operates their rostering system.

The system shall have an inviting home screen but also be as professional as possible as this product is intended for companies and small businesses. The more professional and easy to use the application is, the more the company and the user will enjoy the system which will ultimately lead to a more enjoyable experience for everyone.

The requirements elicitation for the application involved several activities such as initially brainstorming ideas for the system and performing some initial design work which would then later become a GUI for the overview of the system. Analysis was also conducted to see if a similar application or system existed on the market and there are currently systems similar to this idea but some are not well designed and can be improved upon. A sample prototype i.e. GUI was created based on the brainstorming ideas and on ideas of how to improve existing systems while implementing a business rules engine behind it.

Vigorous research was also conducted into the best software to use to create the application. Both Java and ASP.NET were great options but ASP.NET edged Java as it is a better and easier software to implement the ideas of the project. ASP.Net is also free so this meant it would not cost any money to develop the system which was a huge benefit. Focus was first put on designing the system before coding even began as a comprehensive plan needs to be in place before jumping into developing the project. Fail to plan and you plan to fail!

1.3 Background

At the minute I currently work full time so that means I work designated hours of 9-5:30 Monday to Friday so I do not operate on a roster. However, my mother who works in retail does work different hours every week as she operates on a roster. She has often been off work at the weekend or had a week off work and then been completely unaware of her hours for next week. This would be mean she would either have to contact someone from work or go to work while she was on holidays just to view the roster for next week which to me seems a bit ridiculous. I thought to myself surely there must be an easier way than relying on other people or having to go in to view a noticeboard to see your hours.
I was also in my nanny’s house last weekend and my uncle was down who was off work for a few days and my granddad asked him what time he was back in work Monday to which he replied “I don’t know” so I knew this idea definitely had practicality as it seems most people operating on a rostering system are unsure about their hours at some point or another. Of course, there is also the possibility that whoever creates the rosters for the company could be sick or on holidays and someone forgets to create the roster for the following week which puts employees in a corridor of uncertainty which I want my system to eliminate!

I was also having a pint one day and a girl who worked in the pub came in to see her roster and she proceeded to take a picture of her roster which to me didn’t seem ideal or efficient. I then asked the man who was working how their rostering solution currently worked and he told me they operate on an excel sheet with the shifts in it and that the excel is printed and hung up in the back room. This is when I had the idea to integrate generating the rosters from an excel sheet. This meant that employers could still fill out excel sheets and then generate a better-looking roster on the UI and then send the employees a notification that their rosters are now available. I thought this seemed like quite a good idea and was really glad I went to the pub that day!

When it came to thinking of an idea for my final project I had a few vague ideas but this idea of the rostering system came into my head and from there I just ran with it. I thought to myself that most companies probably operate on this rostering system basis so if I made an application to generate rosters with the designated rules behind it and it worked for one company then why can't it work for multiple companies? This meant there was opportunity in the market especially as when I did my research I mainly found calendar apps which stated they served a rostering system purpose but really, they didn't. I wanted to improve on this and create a quality rostering system scheduler that can be used globally.

I had many thoughts around the idea and it seemed a practical idea so I went ahead with pitching that idea for my ratified project proposal.
I proposed this as a mobile application which was then pointed out to me that it would probably not be the best solution to making an application like this to which I agreed so I steered towards a desktop solution. Once my proposal was accepted I knew there was going to have to be a lot of research done behind this which I did and will continue to do but I started to believe in this project. I think it can be a great success and I hope my vision comes to life.

1.4 Aims

The main objectives of this software project:

- To create an online company rostering system to the best of my ability for small and local businesses to manage their rostering needs efficiently and effectively.

- For the application to be fully functioning and easy to use. The application needs to be intuitive and not frustrating to use for end users.

- A main objective is to have a system that has depth and this means developing multiple sections in the application such as:

  1. *Register/Login* – Employers should be able to register and login to the system to create rosters for their employees to use.

  2. *Add/Edit/Remove Employees* – Employees don't always stay in a business so the system should allow the manager/admin to easily manage their employees on the UI.

  3. *Create a roster to send to employees* – The system should allow for generation of rosters but also for managers to be able to manually add/remove or alter shifts if needs be.

  4. *An integrated notification system* – Managers should be able to send email notification to employees that their roster is ready for the week.

  5. *An employee request shift swap section* – Employees should be able to request a shift swap through the system as they may not always be able to work the shift assigned by the manager.

- The application should be a quality application and achieve success through creation.

- An important objective of this project is to complete it on time and stay organized. Time management will be key as time is limited and this project has a lot of complexity behind it.

- Emphasis will be put on testing as working as a QA the creator of Ryan's Rosters feels end to end testing is required to maintain a quality, functioning and well-designed application.

1.5 Technologies

To create the project, it will be developed by using ASP.NET MVC which uses the C# language to create an online application that will run locally on my machine and then be deployed to Azure for public use.

I will also be using GitHub to store my project which will be helpful for version control and to see progress in terms of development.

Backups of the project will also be made as you can never be too careful when it comes to backing up projects so backups will be available on OneDrive, Google Drive, and Dropbox as well as the local machine.

“The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller. The ASP.NET MVC framework provides an alternative to the ASP.NET Web Forms pattern for creating Web applications.”

In terms of a technical approach for this project the key is planning. The plan is to gather the requirements and complete the requirements specification document efficiently with the help of the supervisor to gather a good idea of time in which to complete the project.

The project will be developed in Visual Studio and making an ASP.NET application as this is the best option for creating the project. The application will work on mobile but it is mainly a desktop application

Mockups will be developed using Balsamiq Mockups to get a vision of the application which will ultimately make implementation easier and quicker as it will eliminate the process of over thinking design.

From a research perspective, there will be a vast amount of research around rostering and shift swap systems currently available from other vendors so there will be a lot of research conducted to ensure it is done correctly and better than other competitor solutions. The application will have the potential to be used worldwide as companies worldwide use employees on a shift basis so Ryan’s Rosters will allow them to create rosters easily and will be of huge benefit to them.

2 System

2.1 Requirements

2.1.1 Functional requirements

Management Requirements

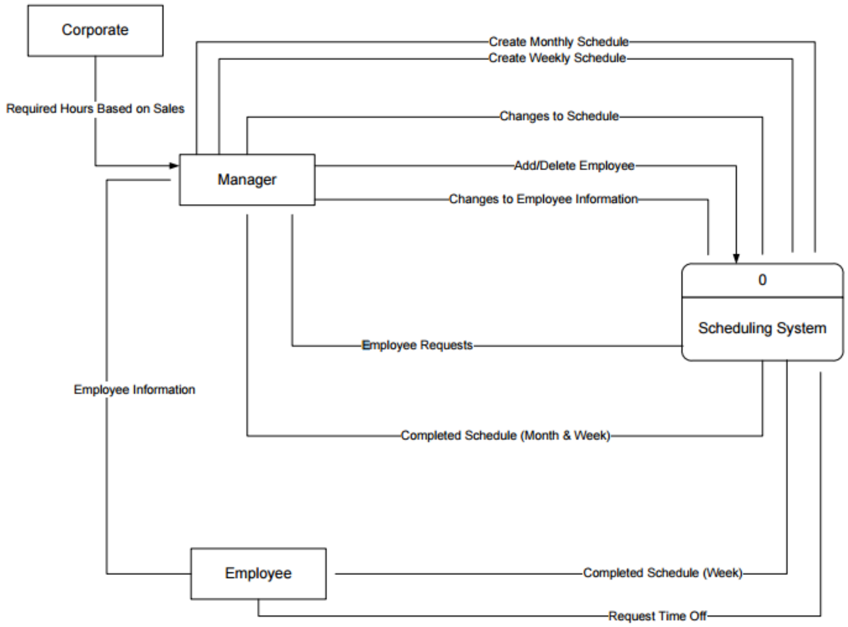
- The system will allow managers to create a weekly schedule
- The system will allow managers to create a monthly schedule
- The system will allow managers to input employee information (i.e. name, phone, email, hours per week)
- Managers will have the ability to edit employee details
- Managers will be able to remove employees that have left the company
- The manager can upload an excel file with employee shifts to generate a roster on the UI
- The manager can add shifts to the roster manually
- The system will allow managers to adjust shifts on the rosters
- The system will allow shifts to be manually removed from the roster
- Managers will receive shift swap requests from employees that have been entered through the system

Employee Requirements

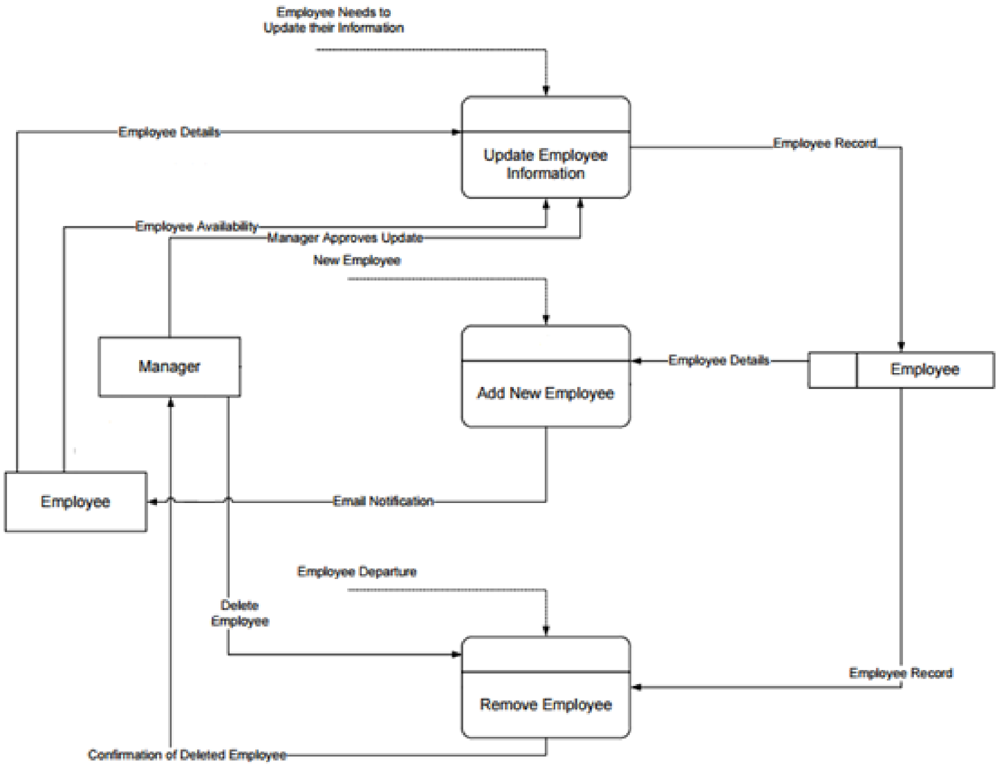
- The system will allow employees to check their roster for the upcoming week/month
- The system will allow employees to request a shift swap with a colleague
- The system will allow employees to view other employees details i.e. email and phone number in case they need to get in contact with them

2.1.2 Data Flow Diagrams

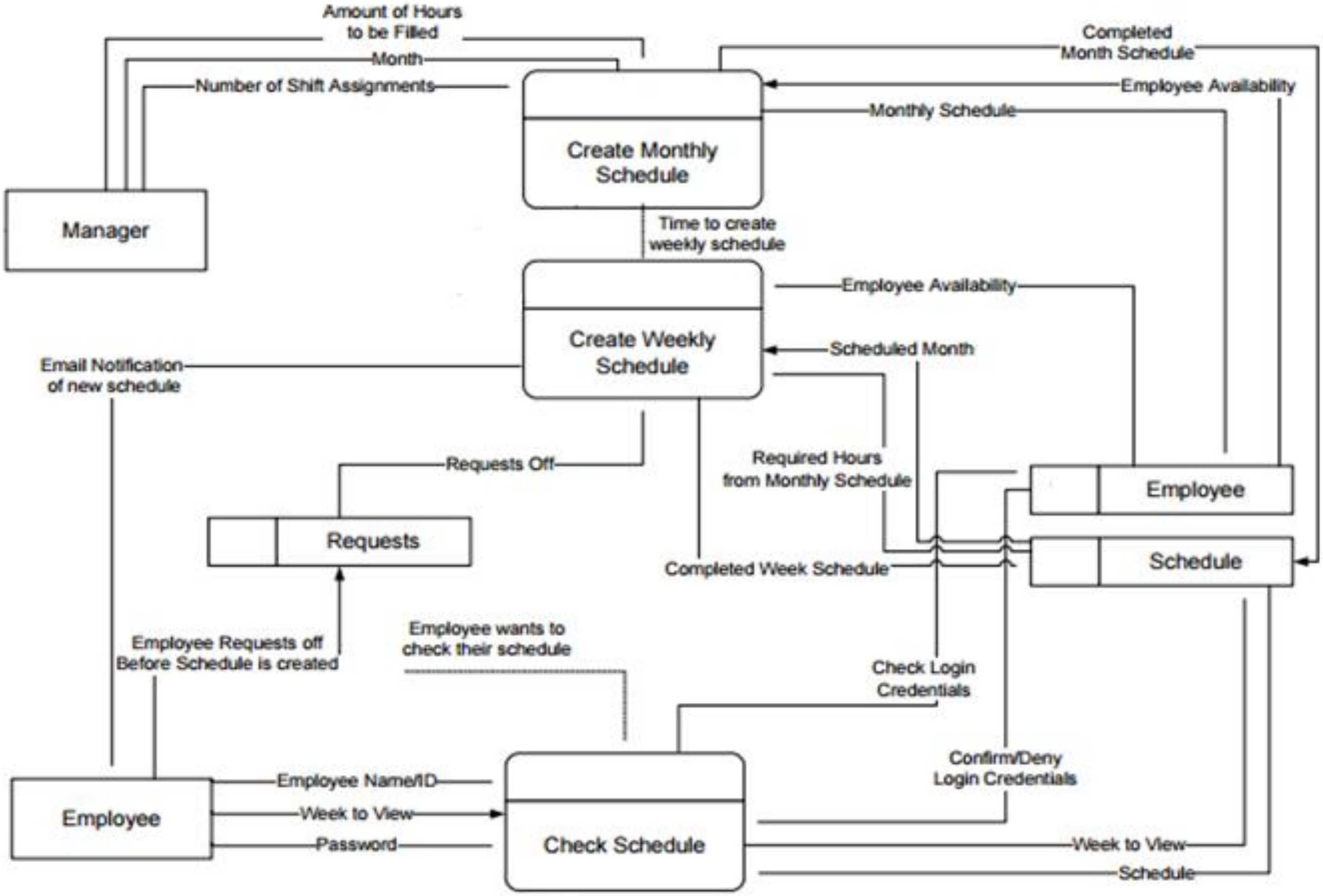
Context Diagram



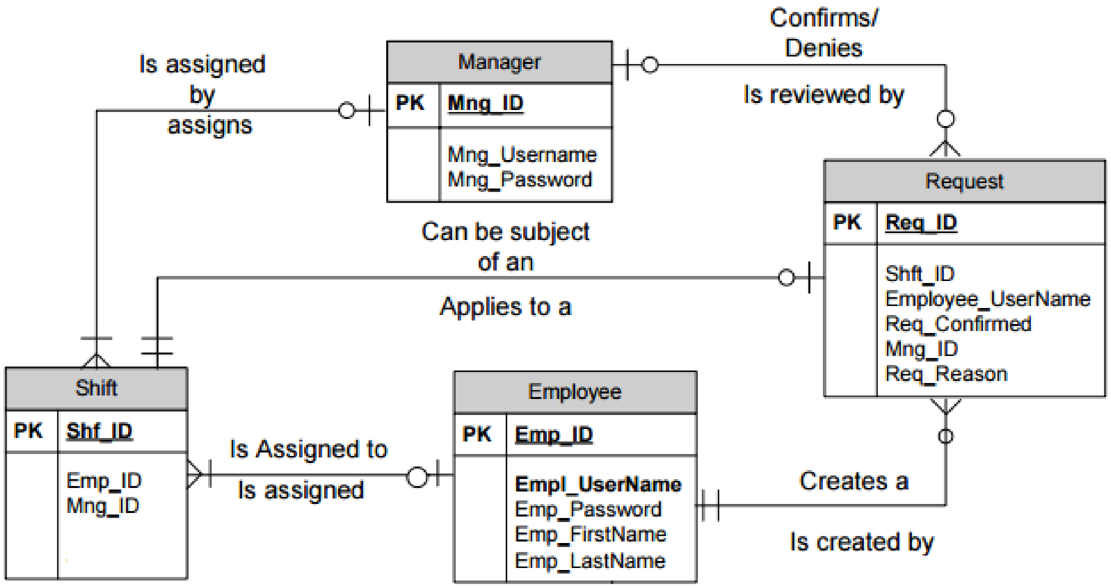
Data Flow Diagram – Level 0



Data Flow Diagram – Level 1



Entity Relationship Diagram



The Employee entity is uniquely identified by an employee ID. The Employee entity contains important information about an employee such as name, username, and password.

Request

The Request entity is uniquely identified by a request ID. It needs to keep track of the Shift ID and the Employee who wants a shift swap. It also needs to keep track of who approved the request and whether it was approved. Only managers can approve a request.

Manager

The Manager entity is uniquely identified by a manager ID. It also contains information about a user name and password to log into a system. Their names are kept track of for the user side. Managers generate rosters and review shift swap requests.

Shift

The Shift entity is uniquely identified by a shift ID. The entity needs to keep track of the manager who created it as well as the employee who will fill it. Each shift is part of the overall schedule.

2.1.3 Requirement 1 <Generating rosters>

2.1.3.1 Description & Priority

This requirement is essential to the system as it is the main feature. This requirement is incorporated to allow the user who in this case is the manager or administrator to generate the appropriate rosters based on the employee’s hours per week that is stored in the employee’s database.

2.1.3.2 Use Case

Scope

The scope of this use case is to allow the user to generate rosters for employees within the company to access.

Description

This use case describes the process of the user generating the rosters within the system and sending them to the employees that are stored within the system.

Flow Description

Activation

This use case starts when the Admin signs in to the system and then navigates to the rosters section of the system

Main flow

1. The Admin logs in to the system
2. The system validates the log in
3. The Admin navigates to the generate section of the system
4. The Admin clicks the import button to generate the roster
5. The system generates the roster based on file uploaded
6. The admin reviews the roster to ensure it is correct
7. The admin clicks the send to email button to send a notification that the roster is ready for the week/month

Post condition

The system sends the roster to the email provided and stored in the database.

2.1.4 Requirement 2 <Shift Swap>

2.1.4.1 Description & Priority

This requirement will allow employees to request a shift swap with another employee as an employee may not always be able to cover the shift provided to them. The shift swap is a high priority as employees will often request shift swaps within their business and this will create an organised rules engine behind requesting a shift swap. This will ultimately help organisation and help to provide email records of shift swaps within the company.

2.1.4.2 Use Case

Scope

The scope of this use case is to outline the shift swap process

Description

This use case describes the process of the user which in this case is the employee requesting a shift swap with another employee.

Flow Description

Precondition

The system is running on the local host

Activation

This use case starts when an employee logs in to the system

Main flow

1. The employee logs in to the system
2. The system validates the log in
3. The employee navigates to the shift swap section of the system
4. The employee clicks the shift swap image
5. The employee is navigated to an email composition to request a swap with a colleague
6. The system sends the request to the admin via email
7. The admin reviews the request
8. The admin can then approve or reject the request

Post condition

The user will be informed if their request is accepted or rejected.

The admin can update the roster accordingly.

2.1.5 Data requirements

A database will need to be connected to the application through SQL Server as the application will be developed in ASP.Net. The database will be required to store data and will have tables such as:

Employees - Employee’s Id, Name, Email, Phone, Location, and Hours.
Events (Shifts) – Event ID, Title, Description, Start At, End At, IsFullDay.
Users – Id, Email, Password, User Name, First Name, Last Name.

These are the main data connections and tables required for the application to work effectively. The users table is required to store user’s login information as security is a high priority for this system and passwords must be encrypted for higher security within the system.

2.1.6 User requirements

The user requirements for the system can be quantified as:

- User can register and log in to the system
- User can view the rosters on the system
- Users can request a shift swap on the system
- User can access the system on all browsers
  (The supported browsers are IE11, Chrome, Firefox, and Microsoft Edge.)
- User details and system data will be stored in the database
- User data will be encrypted
- All user’s security details will be protected

2.1.7 Usability requirements

Understandability

- Interface elements (e.g. menus) should be easy to understand and use
- The purpose of the system should be easily understandable
- The system should be intuitive for users to operate

Learnability

- The documentation and help guides should be easy to follow and to understand for end users of the system
- The system should be easy to learn

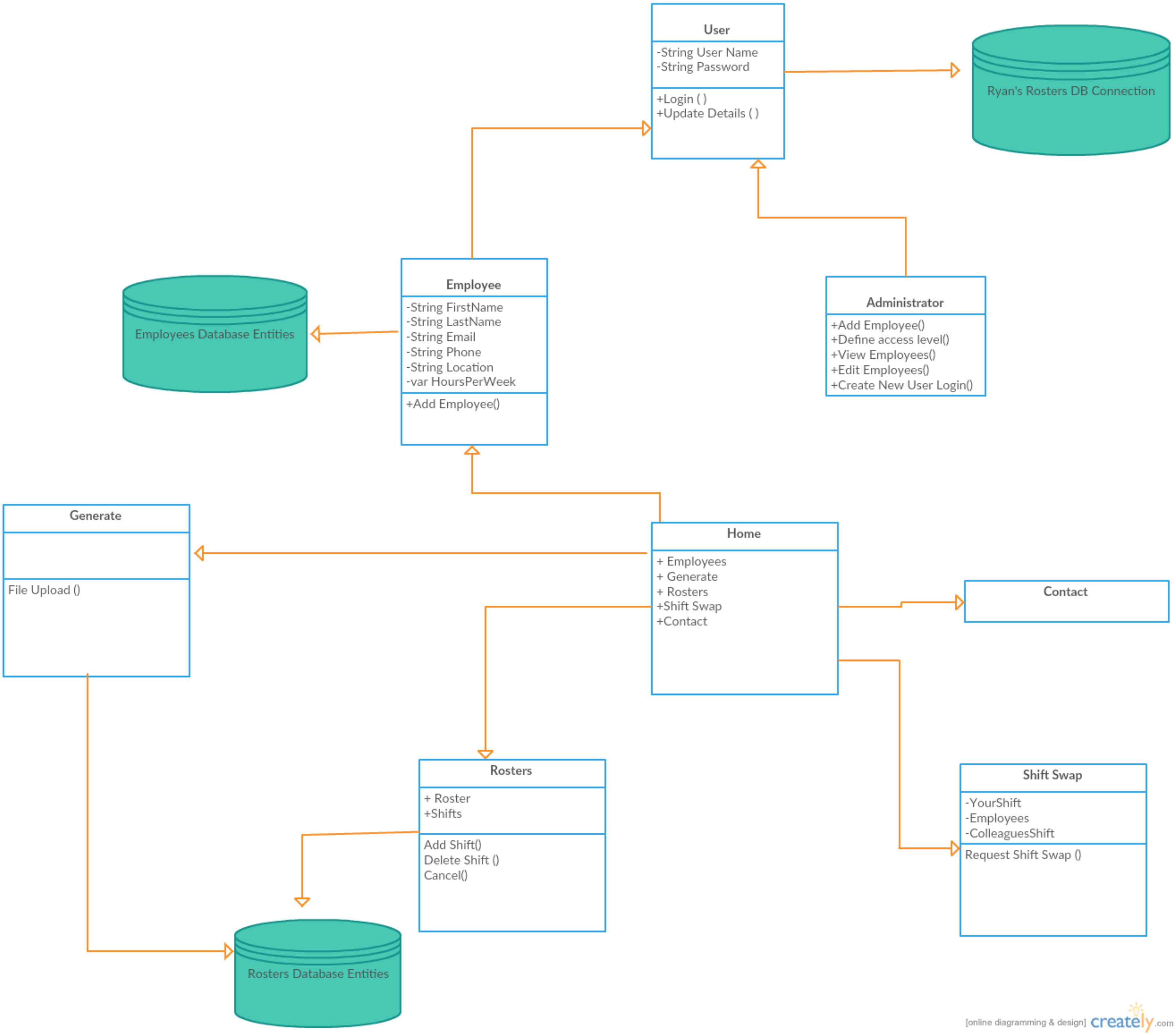
Operability

- The interface actions and elements should be consistent
- Error messages should be displayed where appropriate
- Back should be available for most actions
- A navigation ribbon should be available at all times
- The system can be customisable to meet specific user needs where appropriate

Attractiveness

- The screen layout and colour should be appealing

2.2 Design and Architecture



2.3 Implementation

Generate:

```
@{
    ViewBag.Title = "Generate";
}
<h2>Generate Rosters</h2>
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" 
rel="stylesheet" />
    <script src="https://cdnjs.cloudflare.com/ajax/libs/xlsx/0.8.1/xlsx.full.min.js"></script>
    <script src="http://oss.sheetjs.com/js-xlsx/jszip.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.6.1/angular.min.js"></script>
    <script src="~/Scripts/app.js"></script>
</head>
<body ng-app="MyApp">
    <div ng-controller="MyController" class="container" style="margin-top:50px;">
        <div class="form-inline">
            <input type="file" name="file" class="form-control"
                   onchange="angular.element(this).scope().UploadFile(this.files)" />
            <input type="button" value="Import" class="btn btn-success" ng-
disabled="!SelectedFileForUpload"
                   ng-click="ParseExcelDataAndSave()" />
            <br />
            <span style="color:red">
                {{Message}}
            </span>
        </div>
    </div>
</body>
</html>
```

Generate JavaScript file:

```
var app = angular.module('MyApp', []);
app.controller('MyController', ['$scope', '$http', function ($scope, $http) {
    $scope.SelectedFileForUpload = null;

    $scope.UploadFile = function (files) {
        $scope.$apply(function () { //I have used $scope.$apply because I will call this function from File input type control 
which is not supported 2 way binding
            $scope.Message = "";
            $scope.SelectedFileForUpload = files[0];
        })
    }

    //Parse Excel Data
    $scope.ParseExcelDataAndSave = function () {
        var file = $scope.SelectedFileForUpload;
        if (file) {
            var reader = new FileReader();
            reader.onload = function (e) {
                var data = e.target.result;
                //XLSX from js-xlsx library , which I will add in page view page
                var workbook = XLSX.read(data, { type: 'binary' });
                var sheetName = workbook.SheetNames[0];
                var excelData = XLSX.utils.sheet_to_row_object_array(workbook.Sheets[sheetName]);
                if (excelData.length > 0) {
                    //Save data
                    $scope.SaveData(excelData);
                }
                else {
                    $scope.Message = "No data found";
                }
            }
            reader.onerror = function (ex) {
                console.log(ex);
            }

            reader.readAsBinaryString(file);
        }
    }

    // Save excel data to our database
    $scope.SaveData = function (excelData) {
        $http({
            method: "POST",
            url: "/home/SaveData",
            data: JSON.stringify(excelData),
            headers: {
                'Content-Type': 'application/json'
            }
        }).then(function (data) {
            if (data.status) {
                $scope.Message = excelData.length + " record inserted";
            }
            else {
                $scope.Message = "Failed";
            }
        }, function (error) {
            $scope.Message = "Error";
        })
    }
}])
```

The generate class is referencing the app.js JavaScript file to process to the uploaded excel file for generating shifts on the roster page. This JavaScript is angular JavaScript and from the code we can see how the excel is processed.
If a file is uploaded with the correct the data the UI will display a message saying “excelData.length + “record inserted”. This means it will display 1 record inserted for example. If the file does not meet the criteria it will be failed and if the solution is not running then an error state will be thrown to indicate the file upload is not working. The file must meet the criteria by containing values that map to the database for the roster so it must contain Title, Description, StartAt, EndAt, IsFullDay values to be processed successfully.
The import button is also unavailable until a file has been selected for upload.

Rosters:

```
<div ng-app="myApp" ng-controller="myNgController">
  @* Angular template for modal dialog*@
  <script type="text/ng-template" id="modalContent.html">
    <center>
      <div float right>
        <div class="modal-header">
          <h3 class="modal-title">Shift</h3>
        </div>
      </div>
      <div class="modal-body">
        <div class="error">{{Message}}</div>
        <div class="form-group">
          <label>Shift Title: </label>
          <input type="text" ng-model="NewEvent.Title" autofocus class="form-control" />
        </div>
        <div class="form-group">
          <label>Description: </label>
          <input type="text" ng-model="NewEvent.Description" class="form-control" />
        </div>
        <div class="form-group">
          <label>Time: </label>
          <span>{{NewEvent.StartAt}} - {{NewEvent.EndAt}}</span>
        </div>
      </div>
      <div class="modal-footer">
        <button class="btn btn-primary" type="button" ng-click="ok()">Save</button>
        <button class="btn btn-danger" type="button" ng-show="NewEvent.EventID > 0" ng-click="delete()">Delete</button>
        <button class="btn btn-warning" type="button" ng-click="cancel()">Cancel</button>
      </div>
    </center>
  </script>
  <div class="row">
    <div class="col-md-12">
      <div id="calendar" ui-calendar="uiConfig.calendar" ng-model="eventSources" calendar="myCalendar"></div>
    </div>
```

This is the rosters view which also references an angular JavaScript file (MyApp.js) and uses modal popup for users to be able to add and delete shifts and cancel the action they have taken from the roster which is displayed above. The modal popup shows the title, description and the shift time when displayed by clicking on a record on the rostering calendar. If a user clicks on a date to add to add a shift manually the delete button will not be available as there has been no record added yet but once a shift is added they can navigate back into the shift added and they will be able to delete it. Below are the JavaScript references for the roster itself and the modal popup used.

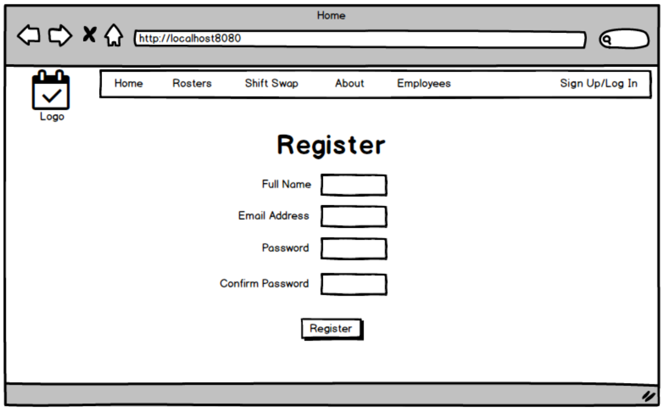
```
@* JS *@
<script src="~/Scripts/moment.js"></script>
<script src="~/Scripts/jquery-1.11.3.js"></script>
@*<script src="~/Scripts/angular.js"></script>*@
@* Will use latest angularjs *@
<script src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.5.0/angular.js"></script>
<script src="~/Scripts/calendar.js"></script>
<script src="~/Scripts/fullcalendar.js"></script>
<script src="~/Scripts/gcal.js"></script>
@* OUR ANGULAR COMPONENTs *@
<script src="~/Scripts/MyApp.js"></script>
@* Script for modal popup *@
<script src="https://cdnjs.cloudflare.com/ajax/libs/angular-ui-bootstrap/1.3.2/ui-bootstrap-tpls.min.js"></script>
```

2.4 Graphical User Interface (GUI) Layout

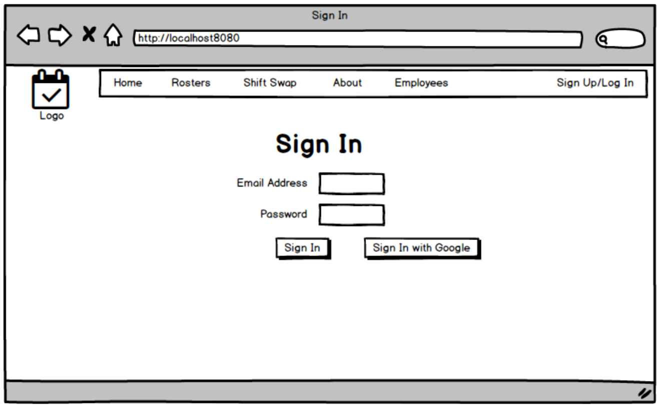
This section describes how the software interfaces with other software products or users for input or output.

2.5 GUI

Register & Login

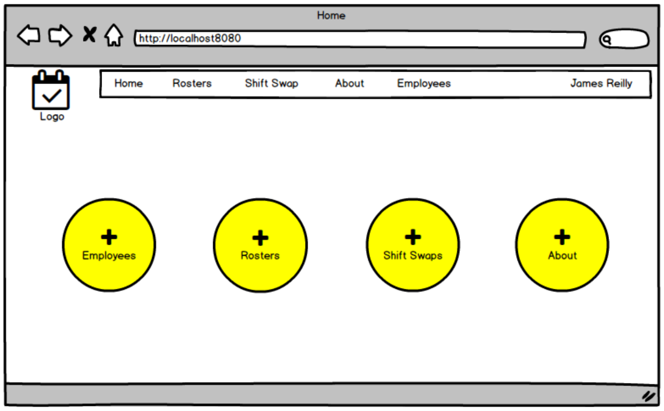


This is the register screen where users i.e. the administrator/manager will register employees employees to be added to the system in order for them to access their rosters.



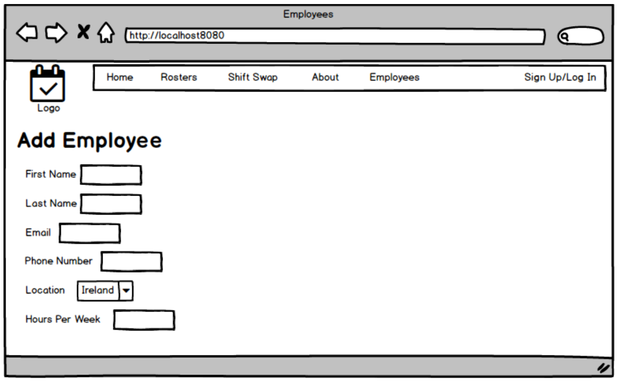
Once users are registered and their details are saved to the database they will then be directed to the login page when they click the Sign Up/Log In link in the top right corner of the page. Users will then input their information to access the system. System Admins will have more rights than normal empoyees i.e being able to add an employee to the system.

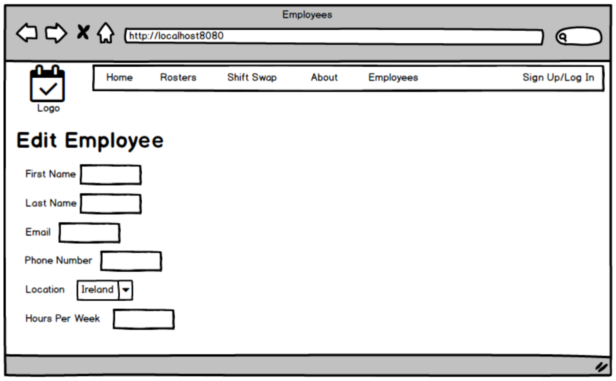
Home

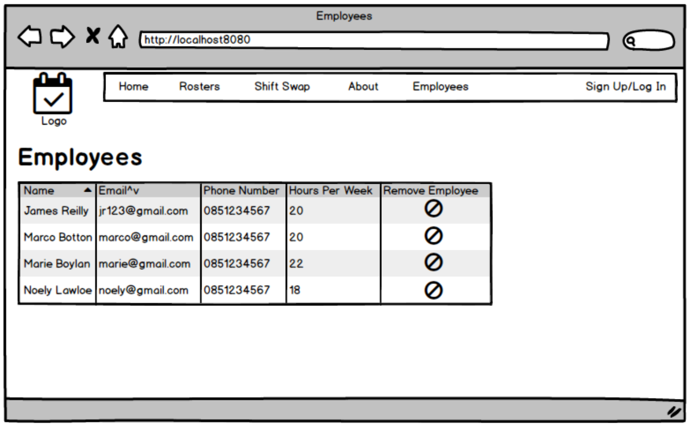


This is the home page where users will be able to access the available sections of the system such as employees, rosters, shift swap and the about page. Employees/users who do not have admin access will be able to access the whole system except for the employees page as they are not permitted to add or remove employees from the system. The system admin is the only person who can access the employees database and make the appropriate changes.

Employees





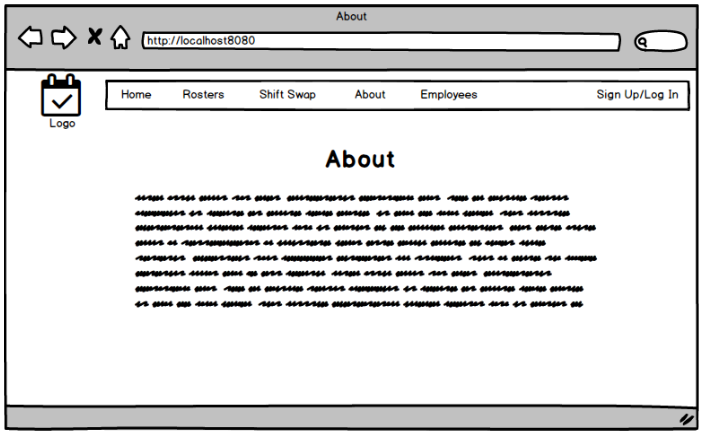


Employees

Name ▲	Email^v	Phone Number	Hours Per Week	Remove Employee
James Reilly	jr123@gmail.com	0851234567	20	⊘
Marco Botton	marco@gmail.com	0851234567	20	⊘
Marie Boylan	marie@gmail.com	0851234567	22	⊘
Noely Lawloe	noely@gmail.com	0851234567	18	⊘

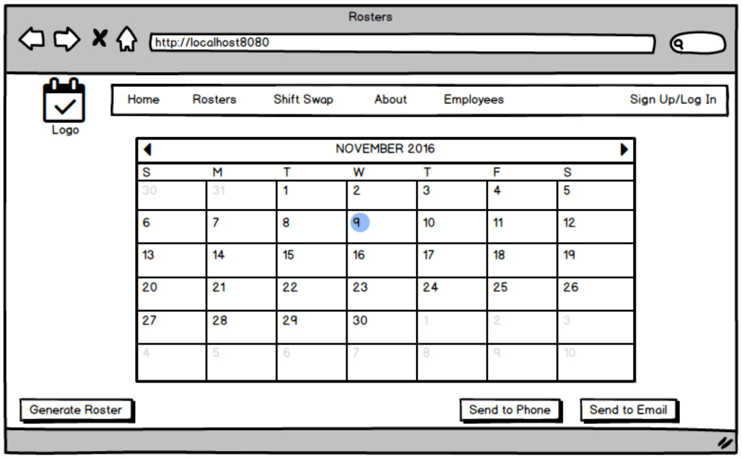
This is the employee screen where the system admin will add users which will save the user and their details to a database to use for generating the roster and also for sending the roster to the appropriate email and phone number they supply. The admin will also have the ability to edit employees. As employees are known to leave companies the administrator will be able to remove employees from the database. The employees will be displayed in a grid with the most relative information on display.

About



This will be a simple about page for the system.

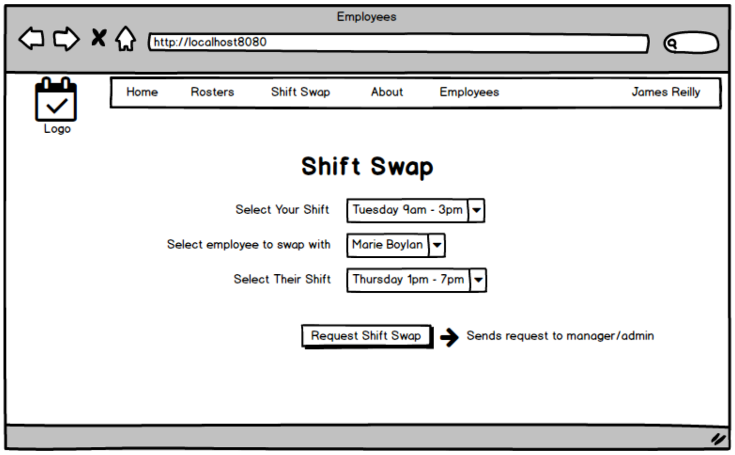
Rosters



S	M	T	W	T	F	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

This is the rosters page where employees rosters will appear once the generate roster button is clicked. The user will be able to send the roster to the employees via phone and email so the rosters are easily seen and accessible.

Shift Swap



Select Your Shift: Tuesday 9am - 3pm

Select employee to swap with: Marie Boylan

Select Their Shift: Thursday 1pm - 7pm

This is where employees can request a shift swap with a colleague via the system. They will select their shift, the employee they want to swap with and select the other employees shift they would like to take. The request will be sent by email to the manager and they will have the option to appove or reject the request

2.6 Testing

For testing Microsoft Test Manager, will be used to store tests that will be performed manually on the systems front end. The tests will be on the system to ensure there are no bugs or errors. To ensure overall quality and full coverage with testing these tests will be executed multiple times through an agile methodology of development. Once an element of the application has been implemented it will be tested throughout to ensure overall operational excellence. Exploratory tests will also be performed to get maximum coverage of the system. Testing will need to be sustainable and ran frequently. Testing API calls on the backend is also a viable method of testing.
Testing will be performed across all browsers i.e. Chrome, Internet Explorer, Firefox, and Microsoft Edge. The main browser will be Chrome as most internet users use Chrome but the application will be supported on other browsers for users who prefer Firefox to Chrome for example.

2.7 Customer testing

As Ryan's Rosters does not have any defined customer's customer testing with people who wanted to purchase the product was not viable. This would be something to invest time in upon further development to tweak design to how the specific user would like the system especially once they have tried the system.

However, the Ryan's Rosters project solution was sent to 6 Quality Analysts in Fenergo (an Irish software company) that have high QA knowledge to be tested and evaluated through exploratory testing for an hour as part of a QA focus group. These people gladly agreed to help testing to uncover bugs and give honest feedback based on the application. Some of the feedback included:

- The design was simple but looked professional
- The system was fast and responsive
- The security in place worked as designed
- The homepage was cluttered
- The application worked well and as intended

This process of asking other professional QA's to help testing worked well and helped to uncover bugs such as typos on the employee page.

Another bug uncovered was how the ribbon changed size when changing between views i.e. Home to Rosters for example.

All bugs uncovered were noted and fixed and then retested to ensure the fix worked.

In this session it was said that the home page was too cluttered and this led to a design change of the home page to simply be the Ryan's Rosters logo so the user is greeted to the logo on the home page and can navigate through the navigation ribbon above.

2.8 Evaluation

Test Case 1: Register Testing

Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Click the register option in the ribbon	Ensure the user is navigated to the register page
Do not fill any fields and click Register	Ensure the appropriate error messages are displayed
Fill in the details i.e. Email, First Name, Last Name, and Password	The password should be at least 6 characters long and contain a symbol
Create a password with 6 characters and a symbol and click register	Ensure you are registered and granted access to the system

Test Case 2: Log in Testing

Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Enter an invalid email and password	Ensure the system returns an error message “Invalid login attempt.”
Enter a valid email and password	Ensure the user is navigated to the home page

Test Case 3: Log in Security Testing

Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Click on the links to the pages in the navigation ribbon i.e. Home etc.	Ensure the system redirects the user to the log in page until they login or register
Login to the system	Ensure you can now access these pages

Test Case 4: Swaps Testing

Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Login to the system	Ensure you are brought to the home page

Navigate to the swaps page	Ensure you are brought to the swaps page
Click on the image to request a swap	Ensure you are brought to the swap request email composer
Fill in the details and click send	Ensure you receive the email and can reply to it to approve/reject the swap

Test Case 5: Add Employee

Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Login to the system	Ensure you are brought to the home page
Navigate to the employee’s page	Ensure you are brought to the employee’s page
Click Create New	Ensure you are brought to the add employee screen
Click the Add Employee button	Ensure the correct error messages appear
Fill in the data correctly and click Add Employee	Ensure you are returned to the employees page your new entry appears

Test Case 6: Edit Employee

Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Login to the system	Ensure you are brought to the home page
Navigate to the employee’s page	Ensure you are brought to the employee’s page
Click Create New	Ensure you are brought to the add employee screen
Click the Add Employee button	Ensure the correct error messages appear
Fill in the data correctly and click Add Employee	Ensure you are returned to the employees page your new entry appears
Click Edit on the employee you added	Ensure you are brought to the edit employee page
Edit some details and click Save	Ensure you are returned to the employee’s page and your updated employee information appears

Test Case 7: Remove Employee

Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Login to the system	Ensure you are brought to the home page
Navigate to the employee’s page	Ensure you are brought to the employee’s page
Click Create New	Ensure you are brought to the add employee screen
Click the Add Employee button	Ensure the correct error messages appear
Fill in the data correctly and click Add Employee	Ensure you are returned to the employees page your new entry appears
Click delete on the employee you added	Ensure you are brought to the remove employee page
Click the Remove button	Ensure you are returned to the employees page and the employee has been removed

Test Case 8: Generate Rosters - Success

Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Login to the system	Ensure you are brought to the home page
Navigate to the generate page through the ribbon	Ensure you are brought to the generate page and the import button is disabled
Click the Choose File option	Ensure the file explorer is opened
Select a file that matches the database table for the rosters (20 entries)	Ensure the file name appears in the upload section and the import button is now active
Click Import	Ensure it says “20 records inserted”

Test Case 9: Generate Rosters - Failure

Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Login to the system	Ensure you are brought to the home page
Navigate to the generate page through the ribbon	Ensure you are brought to the generate page and the import button is disabled
Click the Choose File option	Ensure the file explorer is opened

Select a file that does not match the database	Ensure the file name appears in the upload section and the import button is now active
Click Import	Ensure it says “Error” as the file is not correct

Test Case 10: Rosters – UI/UX Testing

Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Login to the system	Ensure you are brought to the home page
Navigate to the rosters page through the ribbon	Ensure you are brought to the rosters page
Ensure the current date is highlighted with a yellow box on the roster	
Click on the month, week, and day options	Ensure they display the shifts in the roster as appropriate
Click the right arrow on the right of the screen	Ensure the user is taken to the next month and the today option appears
Click the today button	Ensure you are returned to the current date

Test Case 11: Rosters – Add Shift (Full Day)

Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Login to the system	Ensure you are brought to the home page
Navigate to the rosters page through the ribbon	Ensure you are brought to the rosters page
Click on the date in which you want to add the shift	Ensure the modal popup appears
Click Save	Ensure error message “Shift title required appears!”
Enter a shift title and description if required and click save	Ensure the shift appears in the roster and has a full day value

Test Case 12: Rosters – Add Shift (Week View)

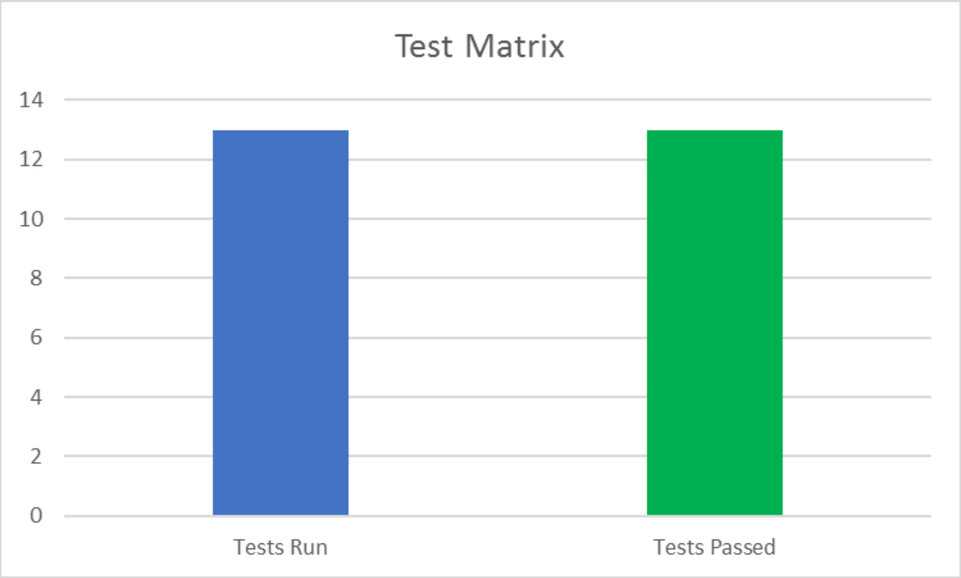
Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Login to the system	Ensure you are brought to the home page
Navigate to the rosters page through the ribbon	Ensure you are brought to the rosters page
Click the week button	Ensure you are brought to the week view of the roster
Click and drag from a time slot i.e. 7am to 1pm	Ensure the modal popup appears
Click Save	Ensure error message "Shift title required appears!"
Fill in the title and description and click Save	Ensure the shift is now visible in the roster

Test Case 13: Rosters – View and Delete

Action	***Expected result***
Run the application	The application should start on the chosen browser and bring the user to the login page
Login to the system	Ensure you are brought to the home page
Navigate to the rosters page through the ribbon	Ensure you are brought to the rosters page

Click on a shift already visible on the roster	Ensure the modal popup appears
Ensure the following buttons are available:	Save Delete Cancel
Click Delete	Ensure the shift is removed from the roster

Test Results:

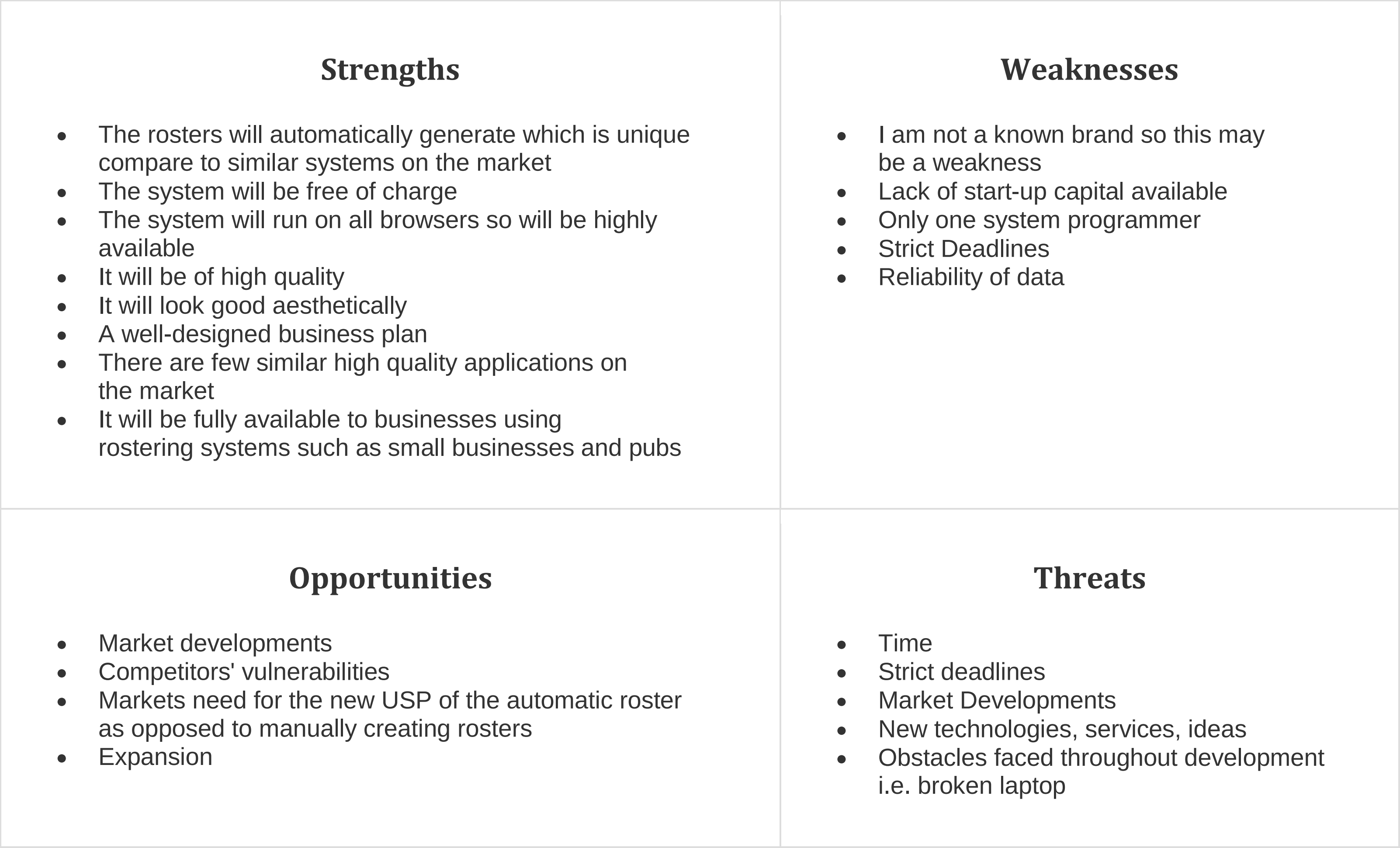


Test Analysis:

As well as these scripted tests that were run against the application, exploratory testing was also carried over for maximum system coverage through an agile approach to software development. As a feature was developed it would be tested and tested again when a new feature was added to ensure the new code did not break the existing code and features already in place. This approach to testing and software development in general worked well and allowed for maximum coverage of the system to ensure error and bug free code. A waterfall method would have been too static when it came to testing approach.

3 Conclusions

Overall, the project was a success in terms of what was set out and what has been delivered. With more time the system could have been better and included better features but the application has potential to be used by businesses due to its ease of use. The chart below sums up the main conclusions of the project through a SWOT analysis:



Strengths

- The rosters will automatically generate which is unique compare to similar systems on the market
- The system will be free of charge
- The system will run on all browsers so will be highly available
- It will be of high quality
- It will look good aesthetically
- A well-designed business plan
- There are few similar high quality applications on the market
- It will be fully available to businesses using rostering systems such as small businesses and pubs

Weaknesses

- I am not a known brand so this may be a weakness
- Lack of start-up capital available
- Only one system programmer
- Strict Deadlines
- Reliability of data

Opportunities

- Market developments
- Competitors' vulnerabilities
- Markets need for the new USP of the automatic roster as opposed to manually creating rosters
- Expansion

Threats

- Time
- Strict deadlines
- Market Developments
- New technologies, services, ideas
- Obstacles faced throughout development i.e. broken laptop

4 Further development or research

There are numerous possibilities as to how this system can evolve over time. The system is solely focused on Irish companies and their rostering needs. If over time the system was a success then the possibility to expand into other countries such as the United States of America, Canada, the UK and many more countries which have companies which are still using physical rosters. The demand would need to be there for expansion but expansion in to other countries would be an easy and achievable prospect for the future of Ryan’s Rosters. Evolution and expansion are something this application should strive on for the future.

Another way Ryan’s Rosters could be developed is to target students as this would be a great application for them to manage their studying needs and to create, store and manage their study plans for exams.

There are also many features which could be added to the system such an integrated time management system to manage employee timesheets to help the company keep track of employee hours and to ensure it matches up with the given roster. This will help ensure the company does not underpay but more importantly for the company overpay their employees as employees may lie on timesheets saying they completed more hours than are part of their roster for the week.

Further research could help expose a hole in the market for these other countries and once the bones of the application are built there is no reason the code cannot be tweaked to coincide with the legislation of the business who is looking for the system. Of course, this would mean that development of the system would be wanted so there would have to be a price outlined for this development. If customers are happy with the price and the overall quality of the service provided, then Ryan’s Rosters will gain a reputation and popularity within the industry meaning more exposure and more potential users which is always a good thing! There are definitely financial prospects and other ways in which this system can evolve in the future!

5 References

THE WORKING WEEK

In-text: ("The Working Week")

Bibliography: "The Working Week". *Citizensinformation.ie* [Online]

Available at:

http://www.citizensinformation.ie/en/employment/employment_rights_and_conditions/hours_of_work/working_week.html

Accessed: 6th December 2016

ASADUZZAMAN, M.

In-text: (Asaduzzaman)

Bibliography: Displaying User Full Name instead of User Email in AspNet Identity 2.0 – CodeProject [Online]

Available at: https://www.codeproject.com/tips/991663/displaying-user-full-name-instead-of-user-email-in

Accessed: 3rd February 2017.

MONDAL, S.

In-text: (Mondal)

Bibliography: Mondal, Sourav. "Part 2 - CRUD Operation On Fullcalender With Angularjs". Demo.dotnetawesome.com [Online]

Available at: http://demo.dotnetawesome.com/part-2-crud-operation-on-fullcalender-with-angularjs

Accessed: 7 May 2017.

IMPORT EXCEL DATA TO DATABASE USING ASP.NET MVC ENTITY FRAMEWORK FUNCTIONALITY

In-text: ("Import Excel Data To Database Using ASP.NET MVC Entity Framework Functionality")

Bibliography: "Import Excel Data To Database Using ASP.NET MVC Entity Framework Functionality". C-sharpcorner.com.

Available at: http://www.c-sharpcorner.com/article/import-excel-data-to-database-using-Asp-Net-mvc-entity-frame/

Accessed: 7 May 2017.

6 Appendix

6.1 Project Proposal

Project Proposal

COMPANY ROSTERING SYSTEM

Ryan Lawlor, x13379761
x13379761@student.ncirl.ie
BSHBISE
14/10/2016

Objectives

The main objectives of this software project:

- To create an online company rostering system to the best of my ability.
- For the application to be fully functioning and easy to use. For me the application needs to be intuitive and not frustrating to use for end users.
- A main objective is to have a system that has depth and this means developing multiple sections in the application such as:

    6. Register/Login
    7. Add/Edit/Remove Employees
    8. Create a weekly roster to send to employees
    9. An integrated email and SMS system to send the roster to employees
    10. An employee request shift swap section

- I want to create a quality application and achieve success through creating this
- An important objective of this project is to complete it on time and stay organized. Time management will be key as time is limited and this project has a lot of complexity behind it.
- Integrating the required rules as outlined by “Rosters - Employee entitlements - Fair Work Ombudsman” to ensure the project has knowledge behind it and is not just meaningless.
- An important objective is also to put emphasis on testing as working as a QA myself I feel end to end testing is required in order to maintain a quality, functioning and well-designed application.
- To be involved with my supervisor as much as possible and continuously gather opinions and feedback from them

Background

At the minute, I currently work full time so that means I work designated hours of 9-5:30 Monday to Friday so I do not operate on a roster. However, my mother who works in retail does work different hours every week as she operates on a roster. She has often been off work at the weekend or had a week off work and then been completely unaware of her hours for next week. This would be mean she would either have to contact someone from work or go to work while she was on holidays just to view the roster for next week which to me seems a bit ridiculous. I thought to myself surely there must be an easier way than relying on other people or having to go in to view a noticeboard to see your hours.
I was also in my nanny's house last weekend and my uncle was down who was off work for a few days and my granddad asked him what time he was back in work Monday to which he replied "I don't know" so I knew this idea definitely had practicality as it seems most people operating on a rostering system are unsure about their hours at some point or another. Of course, there is also the possibility that whoever creates the rosters for the company could be sick or on holidays and someone forgets to create the roster for the following week which puts employees in a corridor of uncertainty which I want my system to eliminate!

When it came to thinking of an idea for my final project I had a few vague ideas but this idea of the rostering system came into my head and from there I just ran with it. I thought to myself that most companies probably operate on this rostering system basis so if I made an application to generate rosters with the designated rules behind it and it worked for one company then why can't it work for multiple companies? This meant there was opportunity in the market especially as when I did my research I mainly found calendar apps which stated they served a rostering system purpose but really, they didn't. I wanted to improve on this and create a quality rostering system scheduler that can be used globally.

I had many thoughts around the idea and it seemed a practical idea so I went ahead with pitching that idea for my ratified project proposal.
I proposed this as a mobile application which was then pointed out to me that it

would probably not be the best solution to making an application like this to which I agreed so I steered towards a desktop solution. Once my proposal was accepted I knew there was going to have to be a lot of research done behind this which I did and will continue to do but I started to believe in this project. I think it can be a great success and I hope my vision comes to life.

Technical approach

In terms of a technical approach for this project the key is planning. The plan is to gather the requirements and complete the requirements specification document efficiently with the help of the supervisor to gather a good idea of time in which to complete the project.

The project will be developed in Visual Studio and making an ASP.NET application as this is the best option for creating the project.

Mockups will be developed using either Visio or Balsamiq Mockups to get a vision of what I am developing towards which will ultimately make implementation easier and quicker as it will eliminate the process of over thinking.

From a research perspective, there will be a vast amount of research around rostering and shift rules and code the necessary rules into the application so there will be a lot of research conducted to ensure it is done right. The application will be focusing mainly on Irish legislation as even though this has potential to be used worldwide, it is best to focus on one area especially as this is being coded from scratch and it does not need to be over complicated.

SPECIAL RESOURCES REQUIRED

Books and online resources around rostering and shift work will be required to fully complete this project successfully.

Online resources around Visual Studio and ASP.NET MVC will be helpful to complete this project. These resources include YouTube videos and online sites such as Asp.net MVC and w3schools for example.

TECHNICAL DETAILS

To create the project, it will be developed by using ASP.NET MVC which uses the C# language to create an online application that will run locally on my machine.

I will also be using GitHub to store my project which will be helpful for version control and to see progress in terms of development.

The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller. The ASP.NET MVC framework provides an alternative to the ASP.NET Web Forms pattern for creating Web applications.

EVALUATION

As I currently work as a Quality Assurance Analyst within a company I am well used to preparing and running tests on a designed system. As I am designing this system I know how it should function which will make it easy to prepare tests. In terms of tests that will need to be run, there must be workflow tests so the user follows a designated path and the system should output the correct result. Negative testing integration tests and unit tests will also be prepared and ran against the system by me to ensure it is behaving correctly.

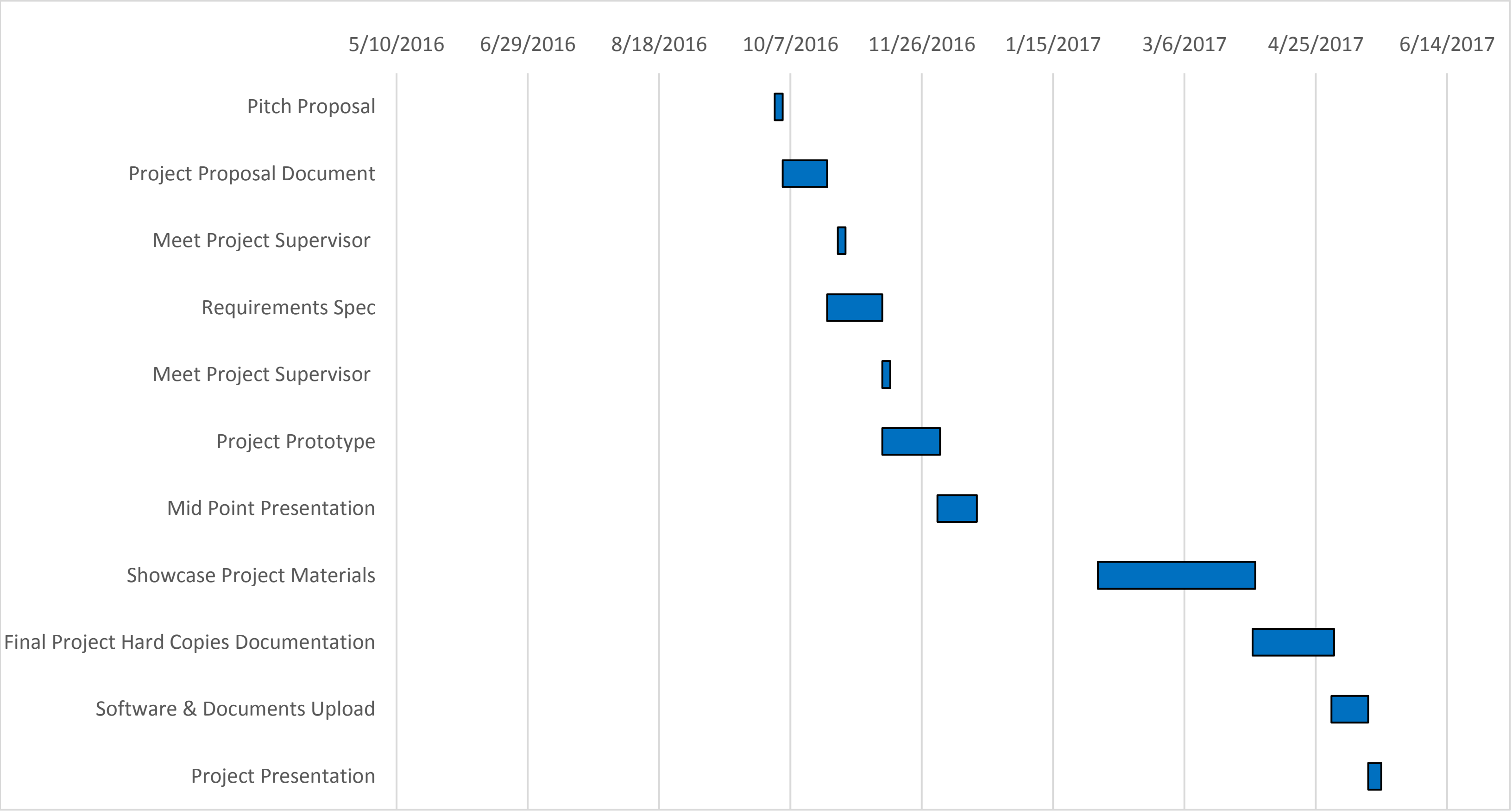
However, I don't feel me running tests will be 100% unbiased or fair as I know the system so I plan to ask a few of my work QA work colleagues who all have different ranges of experience with some having only 6 months and some having 10+ years of experience. I will not tell them how the system works so there testing will be exploratory and it will help me as if they uncover any bugs then I know that area needs to be fixed but also, I will know how easy the system is to use as an end user which ultimately is the goal of any system. I also work with Java developers so I could also get them involved in evaluating the system as it wouldn't hurt to have both a functional and technical evaluation of the system.

Once the end user can use the system easily and there are no errors then I think it is a success. There will be rigorous testing of this system throughout the development of the project as I plan to follow an agile methodology of development.

I think this will be the best approach to development as from my own personal experience it has worked well and I feel the waterfall approach is outdated and not practical. I want to be as thorough as possible with my testing and evaluation of the system which is why I will create a test suite in Microsoft Test Manager (MTM) to store my vast number of tests to be ran against my application.

6.2 Project Plan

Task	Start Date	End Date	Duration
Pitch Proposal	01/10/2016	04/07/2016	3
Project Proposal Document	04/10/2016	21/10/2016	17
Meet Project Supervisor	25/10/2016	26/10/2016	3
Requirements Spec	21/10/2016	11/11/2016	21
Meet Project Supervisor	11/11/2016	12/11/2016	3
Project Prototype	11/11/2016	02/12/2016	22
Mid-Point Presentation	02/12/2016	17/12/2016	15
Showcase Project Materials	01/02/2017	01/04/2017	60
Final Project Hard Copies Documentation	01/04/2017	01/05/2017	31
Software & Documents Upload	01/05/2017	15/05/2017	14
Project Presentation	15/05/2017	16/05/2017	5



6.3 Monthly Journals

Reflective Journal

Student name: Ryan Lawlor

Programme: BSHBIS Evening

Month: September (Month 1)

My Achievements

As this month was the first month of the project it was mainly about coming up with an idea for my project and developing the idea in order to have an overall successful idea that might be reputable within the industry.

My contributions to the projects included sitting down and thinking about ideas for my project. I wanted to come up with a project idea that related to my stream so I wanted a project idea that businesses would love to use and increase their efficiency as a company overall.

First, I had an idea to make a mobile application that had all the user's email accounts in one i.e. Outlook, Gmail etc. I then did some research and found applications that were very similar and well done so I went back to the drawing board.

I was at home then and my mam came home from work and I remembered she'd often been at home and told me she did not know her roster for next week so I thought to myself surely there must be an easier way of knowing your roster instead of having to go in to work or ask a friend for the roster.

I then thought it would be a good idea to have some sort of system or application where an admin/manager can add employees to the system and add their hours as per their contract for the week and generate a roster. The system would then need to have functionality to send an email or text so the employees get their roster instantly.

I pitched the idea and it went well and I got some good feedback which I feel will help me in the long run of developing the application. I am now waiting to get assigned my supervisor and once I get my supervisor I can meet up with them and talk about the project because I feel the more feedback I get, the better and easier it will be to develop this project. Somebody with industry experience would be ideal but any feedback will be appreciated. I’m hoping to make the best project I can and I will start by doing the project proposal that’s due on the 21st of October.

I’m hoping to have my project proposal done and uploaded, do some research around rostering and working regulations in Ireland and create some mock-ups.

Reflective Journal

Student name: Ryan Lawlor

Programme: BSHBIS Evening

Month: October (Month 2)

My Achievements

This month, I was able to complete my project proposal in a timely manner. As we had a reading week I took time off work to have a full week to myself to be able to focus on college work which meant I could get a good start on my requirements specifications document as well other assignments but the requirements specifications document was my main priority. I got a good start on the document and conducted some very valuable research about how I was going to code my project with the business rules engine required behind it.

There is still more work that needs to be put in to the requirements specifications documents but I got a good start on it and will be able to get it done on time for the upload on Friday the 11th of November. The requirements may change over time especially as I must meet my supervisor and they may have some ideas or input but once I get the requirements needed for now I will be happy with my document that will be presented at the midpoint presentation in December.

My Reflection

I felt, I worked well to get a good start on my requirements specifications over the reading week which ultimately would help in my time management later in the semester. I want to finish the document and when I am finished start developing a prototype for the midpoint in December. I feel if I prepare now it will go well when I present so I will put the work in now to make life easier for the future.

Supervisor Meetings

I was assigned my supervisor before the reading week and contacted my supervisor to set up a meeting. Hoping to meet him one of the days after the reading week to discuss my project and future actions.

Reflective Journal

Student name: Ryan Lawlor

Programme: BSHBIS Evening

Month: November (Month 3)

My Achievements

This month, I completed my requirements specification in a timely manner. I conducted some very valuable research about how I was going to code my project with the business rules engine required behind it and included this information in my requirements specification and then my technical report.

I also worked on my technical report for the midpoint presentation before Christmas alongside working on my prototype. In terms of working on the technical report I feel it went well but can always improve especially come towards the final project upload and as I code the application a bit more. In terms of a prototype I began to code the roster part of the project, I have not yet implemented the main algorithm for generating the rosters but that is in the plan and I plan to develop my application after the exams so hopefully the application will come along and come together with the more development time I have after the exams. If I can get the algorithm working to generate the roster and send the roster through the email and phone I will be very happy and it will put me in a good position for May! Using tutorials has helped me develop the roster part of the system but there are few resources available as to how to generate what I am doing so this will be a challenge but I am looking forward to it!

My Reflection

I felt, I worked well to get my requirements and tech report done on time which ultimately will help in my time management and give me more time develop a prototype for the midpoint presentation. I want to get the presentation slides ready and prepare well for the presentation to try get the best marks available and get some feedback to go away over the Christmas and tweak my requirements and develop the application more. I feel if I prepare well this will make life easier for the future and any feedback is always welcome!

Supervisor Meetings

I met my supervisor and told me project idea. I then sent him on my proposal and requirements specifications but I am yet to meet him again.

Reflective Journal

Student name: Ryan Lawlor

Programme: BSHBIS Evening

Month: December (Month 4)

My Achievements

There was not much work completed this month as we were on Christmas holidays and I was studying for exams too.

Supervisor Meetings

I met my supervisor to discuss the mid-point presentation process and then I also met him during the mid-point presentation and got some good feedback to use going forward with my project after the exams were completed in mid-January.

Reflective Journal

Student name: Ryan Lawlor

Programme: BSHBIS Evening

Month: January (Month 5)

My Achievements

Studying for and doing exams took over most of the month so I did not complete much work on my project. When the exams were over I reassessed my project as I had completely over complicated the project. I had a good think about it and consulted some people I work with to see whether the ideas and processes were realistic and tangible and they agreed with me that they were so I decided to go ahead with the proposed changes and try implement them in my project. I made some progress in terms of design and coding the project once I decided to commit to the changes I had thought of.

My Reflection

I was struggling with the project as I am not the best at coding and I had completely over complicated the project. I met with Eamon to discuss the project and he was great help to me and gave me some good advice for the project. As I said I went off and had a think about the project and made it much simpler for me to develop but this did eat in to my time a little bit. I did get some of the project developed which was great and it looks like it is starting to come together a bit now and I hope it will be as good as I anticipate with the changes I plan to implement. I feel the new changes and ideas will work well overall within the application

Supervisor Meetings

I have not met my supervisor this month due to exams and only starting back college but am hoping to regularly meet him to discuss the project going forward.

Reflective Journal

Student name: Ryan Lawlor

Programme: BSHBIS Evening

Month: February (Month 6)

My Achievements

Other important CA’s took over most of the month so I did not complete much work on my project. However, I did implement an employee database and set up other vital pages for the project. I also tried to implement the shift swap/forum/chat system but it did not go to well and the code did not work so I had to go back to an old backup. The next step is trying to get the employees database talking to the scheduler/roster in order to autogenerate tasks and allow them to appear on the roster. I will make another attempt at implementing the shift swap system and will get it implemented even if I must implement it a different way to the way I would like.

My Reflection

I feel I need to get a move on with my project now before exams start because they are not that far away. I have one major 30% left so I plan to complete that as soon as possible and leave my project as the only project and task I must do meaning I can focus solely on the project and get it completed and fully tested and then write up my final documentation. Exams at the end of April will eat into my time especially with study but I feel like there is sufficient time to get the project done to the quality I expect.

Supervisor Meetings

I met with my supervisor to discuss changes with my project I had decided to make and he agreed they were viable changes and I should implement them going forward. He also gave me good advice on how to implement them which was a great help. He also told me to come to him if I needed help with ERD’s and other diagrams going forward.

Reflective Journal

Student name: Ryan Lawlor

Programme: BSHBIS Evening

Month: March (Month 7)

My Achievements

Other important CA's took over most of the month so I did not complete much work on my project. I still must try to get the employees database talking to the scheduler/roster to autogenerate tasks and allow them to appear on the roster. I made another attempt at implementing the shift swap system and struggled a bit so I will need to look at an easier way to implement but ensure the same functionality works to a high quality even though the implementation is simpler. I also worked on my documentation and completed my poster for the showcase which I'm excited for.

My Reflection

I completed all my CA's and prepared notes for my exams in order to have some free time before exams to work on the project within the week leading up to them. Lectures also ended meaning I have a bit more time, not a whole lot but a little more than I had! Once exams are over I also have no other college commitments so I can focus on my project then and adding the final functionalities, touches and styling. I feel I should be okay time wise and look forward to presenting my project and participating in the showcase. I'm excited to be finished!

Supervisor Meetings

I have not met my supervisor this month but the good advice he gave me the month before on how to implement them which was a great help and will help going forward.