

Submitted for the Degree of B.Sc. in Software Engineering, 2015-2016.

# **Employee Performance Management System**

*Registration Number: 201115069  
Name: Aidan Smith*

Except where explicitly stated all the work in this report, including appendices, is my own and was carried out during my final year. It has not been submitted for assessment in any other context.

Signature: \_\_\_\_\_

I agree to this material being made available in whole or in part to benefit the education of future students.



UNIVERSITY OF STRATHCLYDE

## *Abstract*

BSc (Hons) Software Engineering

### **EMPLOYEE PERFORMANCE MANAGEMENT SYSTEM**

by Aidan SMITH

Most companies conduct reviews with employees to monitor and measure their performance over a period of time. There are many Employee Performance Management systems available on the market to do this but most fail to provide a truly continuous feedback process.

The software that has been produced as a result of this project has formed a system that makes use of a continuous feedback process for employees. The EPMS web application allows employees to submit their work carried out every three months. Managers are then able to log in and view this work, grading it appropriately. Feedback on the work is provided in the form of an email notification and a performance page returned to the employee. The performance page includes aims and objectives for the next three months.

The following report explores the design and development of the system discussing the practices and techniques used throughout.

## *Acknowledgements*

I would like to express my utmost gratitude to my supervisor Dr Alex Coddington for her invaluable guidance throughout the project.

I am also extremely grateful to those who took part in my user evaluation and survey. They provided some great insight and advice that helped make the EPMS the best it could be.

Also, to the staff at Police Scotland who were able to spare their time to show me their HR system and answer my questions at related work interviews.

Finally, I would also like to thank all my friends and family for their love and support throughout this project and my time at University.

# Contents

<b>Acknowledgements</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Task . . . . .	1
1.2 Objectives . . . . .	1
1.3 Outcome . . . . .	1
1.4 Structure . . . . .	2
<b>2 Background/Related Work</b>	<b>3</b>
2.1 Existing Systems . . . . .	3
2.1.1 Police Scotland . . . . .	3
2.1.2 Other Existing Systems - Trakstar . . . . .	4
2.2 System Usability . . . . .	4
2.2.1 Mobile Development . . . . .	4
2.2.2 Responsive Web App vs Native Mobile App . . . . .	5
2.2.3 Other Usability Features . . . . .	6
2.3 Anticipated Benefits . . . . .	7
<b>3 Problem Description &amp; Specification</b>	<b>8</b>
3.1 Problem Outline . . . . .	8
3.2 Functional Requirements and Specification . . . . .	8
3.2.1 Original Requirements and Specification . . . . .	8
3.2.2 Final Revised Requirements and Specification . . . . .	8
3.3 Non-Functional Requirements . . . . .	10
3.4 Approach . . . . .	10
3.4.1 Original Project Plan . . . . .	11
3.4.2 Final Revised Project Plan . . . . .	11
<b>4 System Design</b>	<b>12</b>
4.1 Database Design . . . . .	12
4.2 Design Methodologies . . . . .	15
4.3 System Use Cases . . . . .	17
4.3.1 Employees . . . . .	17
4.3.2 Managers . . . . .	17
4.4 User Interface Design . . . . .	18
4.4.1 About . . . . .	18
4.4.2 Home Page . . . . .	18
4.4.3 Datatables . . . . .	19
4.4.4 View Records . . . . .	19
4.4.5 Open Records . . . . .	20
4.4.6 Submitted Records . . . . .	21
4.4.7 Graded Records . . . . .	22
4.4.8 View Performance Page . . . . .	23
4.4.9 Forms . . . . .	24

<b>5 Detailed Design &amp; Implementation</b>	<b>25</b>
5.1 Submit Work Feature . . . . .	25
5.1.1 Records . . . . .	25
5.1.2 Work . . . . .	27
5.1.3 Submission . . . . .	29
5.2 Feedback Process . . . . .	30
5.3 Technologies Used . . . . .	32
5.3.1 Visual Studio . . . . .	32
5.3.2 Razor/Entity Framework . . . . .	32
5.3.3 Bootstrap & CSS . . . . .	33
5.3.4 SVN/Git . . . . .	33
5.3.5 C#.NET . . . . .	34
5.3.6 JavaScript . . . . .	35
5.3.7 MVC 5 . . . . .	35
5.3.8 SQL Server . . . . .	36
5.3.9 MySQL . . . . .	36
5.3.10 Microsoft Azure . . . . .	37
5.4 Security . . . . .	38
5.5 Error Handling . . . . .	39
5.6 Database . . . . .	40
5.6.1 Database Challenges . . . . .	41
5.7 Implementation Challenges . . . . .	41
5.7.1 Bootstrap Datepickers . . . . .	41
5.7.2 New Employees . . . . .	41
5.7.3 IsManager Flag . . . . .	42
<b>6 Verification &amp; Validation</b>	<b>43</b>
6.1 Testing Strategy . . . . .	43
6.1.1 Black Box Testing . . . . .	43
6.1.2 Automated Browser Testing . . . . .	43
6.1.3 User Testing . . . . .	44
6.2 Debugging . . . . .	44
6.3 Other Testing . . . . .	44
6.3.1 Testing on Devices . . . . .	44
6.3.2 Testing on Browsers . . . . .	45
<b>7 Results &amp; Evaluation</b>	<b>46</b>
7.1 Final Result . . . . .	46
7.2 Changes & Risks . . . . .	51
7.2.1 Version Control . . . . .	51
7.2.2 Programming Languages and Technologies . . . . .	51
7.3 User Evaluation . . . . .	51
7.3.1 Plan . . . . .	51
7.3.2 Likes . . . . .	52
7.3.3 Dislikes . . . . .	52
7.3.4 Suggestions/Comments . . . . .	52
7.4 Summary . . . . .	53

<b>8 Summary &amp; Conclusions</b>	<b>54</b>
8.1 Summary . . . . .	54
8.2 Possible Future Features . . . . .	54
8.2.1 Further Authentication Development . . . . .	54
8.2.2 Further Notification Development . . . . .	54
8.2.3 Peer Assessment . . . . .	55
8.3 Conclusion . . . . .	55
<b>References</b>	<b>56</b>
<b>A Appendix A - Working System Screenshots</b>	<b>58</b>
<b>B Appendix B - Use Case Descriptions</b>	<b>66</b>
<b>C Appendix C - Test Cases and Results</b>	<b>77</b>
<b>D Appendix D - SVN/GitHub Commits</b>	<b>94</b>
<b>E Appendix E - High Level Class Diagrams</b>	<b>100</b>
<b>F Appendix H - User Evaluation Documents</b>	<b>106</b>
<b>G Appendix G - User Guide</b>	<b>112</b>
<b>H Appendix H - Program Listing</b>	<b>123</b>
<b>I Appendix I - UI Mockups and Ideas</b>	<b>124</b>
<b>J Appendix J - Trakstar Email Trail</b>	<b>133</b>
<b>K Appendix K - Trello board screenshot</b>	<b>135</b>
<b>L Appendix L - Selenium Automated Test Results</b>	<b>136</b>

## List of Abbreviations

<b>EPMS</b>	Employee Performance Management System
<b>TDD</b>	Test Driven Development
<b>GUI</b>	Graphical User Interface
<b>RWD</b>	Responsive Web Design
<b>SCOPE</b>	System Coordinate Personal Establishment
<b>DBA</b>	Database Analyst

# Chapter 1

## Introduction

This chapter will discuss the motivation for the project with the main objectives and outcomes. The overall structure of this report will then be outlined.

### 1.1 Task

The task was to build a continuous feedback system to measure employee performance within an organisation.

### 1.2 Objectives

The objectives are to build a continuous feedback system that allows employees to very easily record their work on a day-to-day basis and receive notifications from "scorers" when work is graded. This could be shown in an alert mechanism, and scores could be recorded in a dashboard.

### 1.3 Outcome

This EPMS (Employee Performance Management System) is a responsive web application that aims to reduce the amount of time between performance reviews and provide a more continuous feedback system for employees. The system will enable staff to easily record completed projects and tasks assigned by their managers inside records which will last for three months.

At the end of each three month period the records will be submitted to the manager and the work will be inspected. Then the work will be graded and a performance page will be created for the record. This will include a score out of ten, an overall grade (unsatisfactory, satisfactory or very good), justification for the grade and aims set out for the next three months.

When a record is graded, a notification system is put in place for employees to receive an email notification with their grade and a link to the system. Employees can now log into the system and view the feedback they have received and create the next record and start creating work for it.

Due to the deliverable being a responsive web application, it can be used over a number of platforms and devices.

For more information on how to use the system see Appendix G - User Guide.

## **1.4 Structure**

Initially, the report will discuss background and related work - this will include results from market research carried out with Police Scotland and Trakstar. The background information also includes system usability and anticipated benefits of the system. In chapter 3, the problem description and specification is outlined - the chapter will include initial and finalised functional and non functional requirements as well as the approach to the whole project. Chapter 4 is where the system design will be detailed - database design, design methodologies, system use-cases and the user interface are discussed in this chapter. Chapter 5 focuses on the more detailed design - gives insight into the implementation of the main features of the system and highlights the technologies used and justifies their use. The security, error handling and database challenges are also discussed here. Chapter 6 summarises and discusses how the project was verified and validated throughout the testing stages. Chapter 7 explains the final result and discusses the evaluation carried out throughout. The final chapter summarises the project as a whole, and reflects on the project providing some possible future features that could be added.

# Chapter 2

## Background/Related Work

Background/Related Work describes existing systems that were analysed before the design stage of the project. Background information such as usability of the system and anticipated benefits are also discussed.

### 2.1 Existing Systems

The Existing Systems section discusses the results revealed from background research carried out on current similar systems. The features included in similar type of systems were used as a reference when this Employee Performance Management System was designed. Existing systems that will be discussed include "SCOPE"(System to Coordinate Personnel and Establishment) - a HR program used by Police Scotland. Stakeholders that use these systems were interviewed in advance and were then involved in the user evaluation stage of the project.

#### 2.1.1 Police Scotland

SCOPE is a HR system that is used by Police Scotland to manage employees, appraisals and performances. This has been used by the whole organisation fully since Police Scotland merged to one organisation in 2013.

The system itself, is a large process and has many different opportunities for good functionality. Personal records are stored in the system containing personal details of each employee and it is the responsibility of the employees to update this information. An example of this information being used in the system, is the annual/sick leave stored in personal records, which are used to populate calendars and shift rotas. The system also includes links to other software such as staffing applications which generate documents such as shift rotas for Police officers.

SCOPE's area of functionality that is most relevant to this EPMS is the appraisals. These are carried out every twelve months (or when moving role within the organisation) and is when employees meet with their line managers to discuss performance over the past twelve months. Within the twelve month period, employees and managers will both add "evidence" to the appraisal page. Evidence is proof of work being carried out (or not being carried out) and could include skills developments such as training courses. This is what determines the level of performance of the employee for the twelve-month period. It is essential that this appraisal page is managed properly by both employees and management for the system to function effectively.

Background/related work involved interviews and meetings with a number of different stakeholders that use the system. These users were at different ranks within the organisation - one at the manager's end, one at employee's end and one which was ranked as both an employee and a manager. They were able to give different views and insight into the SCOPE system. Speaking to more than one stakeholder and in different roles has been vital to extract as much relevant information as possible and from different angles of the system.

Research results showed that SCOPE, as a system has potential to be very useful to Police Scotland. One of the issues that emerged was that no member of IT staff owned the application or the data. This resulted in some "loose ends" within the system.

### **2.1.2 Other Existing Systems - Trakstar**

Trakstar[1] is a market leading performance appraisal software based in Seattle - this system was used as a reference when carrying out market research for this system. The aim of the trakstar is to provide software that is easy to use and be as intuitive as possible. The system they provide is able to adapt easily to any business of any size.

When an appraisal is created every twelve months in Trakstar, competencies are added. These are good (and bad) tasks that have been carried out by employees throughout the year. When the managers are scoring the appraisals they are able to refer to these competencies as evidence and justify the scores. Overall appraisal scores are formatted and customized to what the managers prefer ie as a number or string value. Automatic email reminders and notifications are also provided in the application.

EPMS used this type of system for research and analysed it's features when at design stage. Some features in this application inspired decisions that were made throughout and contact was made with Trakstar via email to investigate their system further and request that they take a look at the final EPMS result. They were extremely helpful with regards to providing information about their system. Glenn Martin was the point of contact from Trakstar who is their HR Performance Officer. Please see Appendix J for the email trail with Trakstar.

## **2.2 System Usability**

This section shows how research aided decisions made regarding the usability of the EPMS. One of the most important decisions made was to implement a responsive web application which would be readily available on several different devices.

### **2.2.1 Mobile Development**

Many systems in different industries today, target several platforms and devices to allow their end-users to gain the best possible experience when the system is functioning. Research carried out at Police Scotland, produced results indicating that users of current systems would like to be able to use their EPMS/Appraisal systems while on the move. They stated that they believe their SCOPE HR System would be embraced more if they were able

to use it on multiple devices and not just their traditional desktop computer. This inspired the decision that the project would be mobile friendly and include software which would be able to be used on different desktop and mobile platforms. Benefits of having software which is mobile friendly include[2]:

- Software becomes more convenient to use, as people are able to use it in different ways on different platforms
- Mobile devices provide an enhanced user experience - touch-screen, accelerometer etc
- Wireless handsets can be used any time and any place - more portable than traditional desktop software

### 2.2.2 Responsive Web App vs Native Mobile App

Native mobile applications are specific types of software that are designed and developed to be used only on mobile devices. These devices include smart phones and tablets from many different operating systems such as iOS and android. Native applications are designed for particular platforms and this allows them many benefits such as[2]:

- They do not have to be connected to the Internet to run
- Ability to take advantages of features that are OS specific. An example of this is push notifications within the application
- Responsiveness within a native mobile application tends to be much more smoother than a web application

Responsive web applications are web applications that respond to the user's behaviour and (or) environment. This approach to web development is becoming more and more popular and as findings have shown in the existing systems, more people would like to embrace mobile technologies. Responsive Web Design (RWD) allows the whole design of the web application to be more flexible and be used on a number of different devices and environments for example, if a user switches from a desktop to a tablet, the web application will automatically adapt to the new device's preferences increasing the quality of the whole user experience. The user can also resize the browser window in use and the responsive web app reacts to the change. Benefits of RWD include[3]:

- Can offer web applications over several different platforms
- Responsive web applications do not need to be installed on any device as they are simply run from a browser
- Reduced maintenance is required as there is only one version of the project and changes will be made in one centralised place
- Time and cost are the two main advantages. Time is saved in that only one web application needs to be created and money is saved in that only one web application needs to be hosted

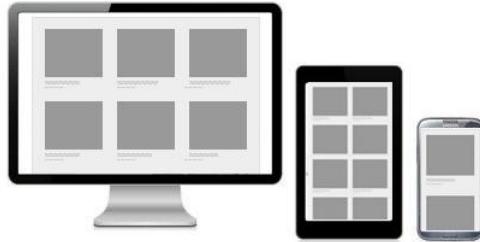


FIGURE 2.1: Responsive Example

[3]

It is apparent that both types of applications considered would be very beneficial and in an ideal world, both would be implemented. In terms of this project, however, the EPMS was written as a responsive web application.

When considering project development, value for money and time constraints are essential considerations. In view of this, the most cost-effective and time efficient method was to write the application as a responsive web application. Furthermore, this allowed for easy distribution when approaching evaluation stages as the application did not need to be installed onto a platform.

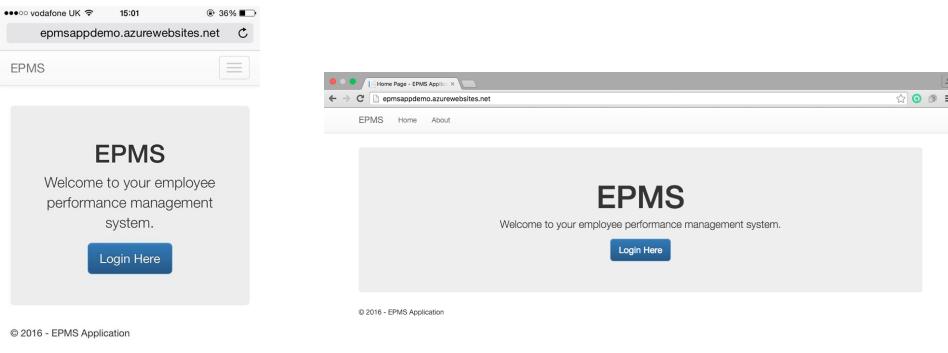


FIGURE 2.2: EPMS - Mobile and Desktop

### 2.2.3 Other Usability Features

It was important when designing software to consider all types of users that may come across the system. One of the aspects of the EPMS web application that aids usability is that it reaches out to people that may have a disability or colour blindness.

It is extremely difficult, if not impossible to design a system that addresses every possible disability. However, it is important to understand the users and account for as much as possible. When designing the EPMS web application, buttons were made larger and "clickable". This helped users with mobility issues using the mouse/trackpad. Another aspect of the system which helps users with disabilities is that it has been kept as simple as possible. It has been made clear throughout that an aim would be

for the system to be easy to use and simple. Through user evaluations and testing (see chapter 7) it can be seen that this was a success.

Using smart colour choices is good practice for any user interface as this could be useful for users who are visually impaired or suffer from colour blindness. This was considered during the design stage of the EPMS to improve overall usability. EPMS avoids pairing similar colours and simple text is displayed on simple backgrounds where possible to improve the colour scheme[4].

### 2.3 Anticipated Benefits

- It is anticipated that the system will be easy to use and intuitive (users will be able to pick up the functionality very quickly).
  - The system will be very user friendly and be aimed at all kinds of people from different backgrounds.
  - Aimed at all sizes of business from small teams to large organisations with several teams.
  - The system will be available on most platforms and devices. This is something that was requested when interviewing existing users of the HR system at Police Scotland.
  - A continuous beneficial feedback system will be set in place meaning users will receive their feedback for previous record and aims for next record will be set out clearly.
  - Managers will be able to see submitted work clearly in tables and queues, making it easier to provide feedback for records.

## Chapter 3

# Problem Description & Specification

### 3.1 Problem Outline

The main aim of this project was to design and implement software that would be used as an employee performance management system. There is an abundance of these systems currently in the market. However, many suffer from the issue of failing to deliver a continuous feedback process for employees regarding their performance and aims. An example of this, can be seen clearly in the background/related work section where the SCOPE system that Police Scotland use is discussed.

### 3.2 Functional Requirements and Specification

This section discusses the functional requirements of the system, initial and finalised requirements and how the project evolved.

#### 3.2.1 Original Requirements and Specification

Initially, the aims and objectives of the project were as follows:

To produce a system that allows employers to continuously provide feedback to employees based on work submitted. This would be achieved via an alert mechanism supported by a mobile phone native app. The system would allow peer groups, clients and other third parties, to record their feedback - this could be achieved by the introduction of a social ratings system. Peers and the client can often be closer to the appraisee's performance than line and project managers, and the introduction of a social "rating/comment/discussion" system that is integral to the process could provide better evidence of the appraisee's performance over the course of the year.

#### 3.2.2 Final Revised Requirements and Specification

Things changed throughout the software development cycle and in the end it was decided that performances would be recorded every three months to allow grades and aims to be more accurate and specific. Also, only direct line managers would be able to view and grade work completed by employees.

Final Revised Requirements:

The system has two different types of users - Employees and Managers. Employees will be able to:

- Login/Logout using the login details provided by their line manager.
- View all of their own records that are created every three months
- Create a new record every three months. These will be used to store each piece of work completed within the three months.
- Have access to a performance page for previous records, where they will see the feedback they have received from their line manager and be able to see aims and objectives within the next 3-month period.
- Use the performance pages as a reference when creating new work.
- Open up each record to view what pieces of work are included in it.
- Create a piece of work within the record to act as evidence that a piece of work has been completed and describe the work. Work includes projects finished or training courses completed.
- Edit and(or) delete pieces of work.
- Submit a record to their line manager to view and grade the work and return the results in the form of a performance page.

Managers will have the same full permissions that employees have. The means that they are able to create work and records for themselves and submit them to their line managers. In addition to this, they will be able to:

- Add new employees to their team in the system
- Edit existing employees to their team in the system
- View full details of employees in their team
- Create new login accounts on the system
- View records for each employee and inspect the work within each record
- Grade records and provide feedback by creating a performance page for each record - This is where managers are able to justify the grade and set aims and objectives for the next three months
- View all records that have been submitted by their team members
- Add and edit projects in the system
- Managers can submit their own work to their line managers that may be in charge of another team. This feature increases the ability to expand the system for a larger organisation

### **3.3 Non-Functional Requirements**

The system should be able to:

- Be fully functional on any platform including mobile devices
- Be fully functional on any browser
  - For limited time, will be accessible via the following link:  
<http://epmsappdemo.azurewebsites.net>
  - Have user login authentication
  - Be an attractive responsive web application that is simple and easy to use

### **3.4 Approach**

For the project to be a success, the problem had to be broken down into a number of stages and this involved using a good structured and disciplined plan. The early stages involved some research and background work in order to gain a better knowledge and understanding of existing employee performance management systems in the market. This included research of existing applications online that do provide this and also interviewing stakeholders that are involved in the current HR system at Police Scotland.

Interviews were carried out with three different stakeholders in the system - ICT Manager, Chief Inspector and Divisional Commander. This was important in that it gave an insight into the system from different angles. Information collected helped to correlate ideas and features that could be used to create the best possible employee performance management system.

After interviews were carried out to help gain an understanding of these systems, the next step was the design stage. This included decision making for which technologies, techniques, design methodologies and UI layouts would be used in the system. Some of the most important decisions made were that the system would be written in c#.NET and using MVC5. Agile design methodology was chosen to be used and UI screen mockups were created on "Moqup"<sup>[21]</sup> to show how the system would look. Weekly project meetings helped implement the Agile methodology and new features were added sequentially and tested after the addition of each feature. Full details of the design of the system and decisions made can be found in chapters 4 and 5.

The implementation of the application was completed by using test driven development as much as possible. This involved prioritising features into "must haves", "could haves" and "would haves"(known as the MOSCOW technique)<sup>[22]</sup>. This helped with the ordering of which features would be added to the system first. After the development of each feature they were tested and published onto the live environment on azure.

The next stage was to test the system and this was very important. Black box testing was used throughout for each use case to confirm the functionality for each one. Another type of testing that was used was automated

testing. This was carried out using software called Selenium HQ. This is a web browser automation tool that can be used for testing purposes. This was used to automate features and prove that the test cases passed. User testing was also carried out providing feedback and evaluation of the system. For full information on testing see section 6.1.

Evaluation was carried out throughout the whole project as much as possible. This was completed from interviews to user evaluations and self evaluations. User evaluations were completed in different groups of people from different backgrounds including IT and medicine students. Police Scotland also helped with user evaluations and this can be seen in more detail in chapter 7.

### 3.4.1 Original Project Plan

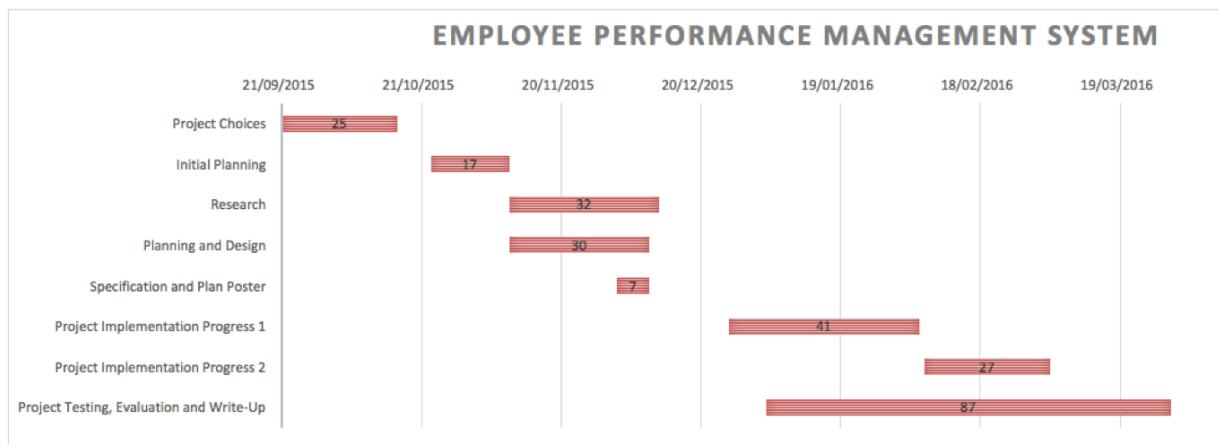


FIGURE 3.1: EPMS - Initial Plan Gantt

### 3.4.2 Final Revised Project Plan

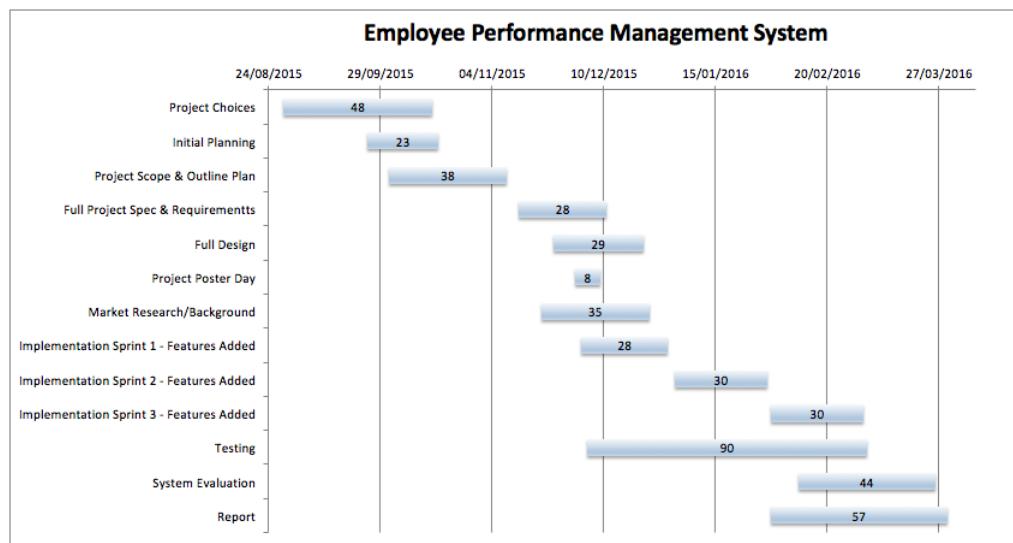


FIGURE 3.2: EPMS - Revised Plan Gantt

# Chapter 4

# System Design

## 4.1 Database Design

The design of the database tables within the system was an important part of the system design. The data was normalised and the final database tables can be seen below and the full structure can be seen in figure 4.7:

**Employees Table**

Column	Type	Key	Extra
ID	Int	Primary	AUTO INCREMENT
First Name	Nvarchar(255)		
Last Name	Nvarchar(255)		
User Name	Nvarchar(255)		
Row Version	TimeStamp		
Employee_Manager	Int		
IsActive	Bit		
Team	Int	Foreign(Team ID)	

FIGURE 4.1: EPMS - Db Structure - Employees

**Records Table**

Column	Type	Key	Extra
ID	Int	Primary	AUTO INCREMENT
Status	Nvarchar(100)		
SubmittedDate	DateTime		
TimePeriodBegin	DateTime		
TimePeriodEnd	DateTime		
Row Version	Int		
Record_Employee	Int	Foreign (Employee ID)	

FIGURE 4.2: EPMS - Db Structure - Records

**Work Table**

<b>Column</b>	<b>Type</b>	<b>Key</b>	<b>Extra</b>
ID	Int	Primary	AUTO INCREMENT
DateCompleted	DateTime		
DateDue	DateTime		
WorkItem	Nvarchar(100)		
Description	Nvarchar(500)		
HoursWorked	Int		
HoursRemaining	Int		
SubmittedDate	DateTime		
Row Version	TimeStamp		
Work_Record	Int	Foreign Key (Record ID)	
Work_Project	Int	Foreign Key (Project ID)	
Work_Category	Int	Foreign Key (Category ID)	
Late	Int		

FIGURE 4.3: EPMS - Db Structure - Work

**Teams Table**

<b>Column</b>	<b>Type</b>	<b>Key</b>	<b>Extra</b>
ID	Int	Primary	AUTO INCREMENT
TeamName	Nvarchar(100)		
Manager	Int		
IsActive	Bit		

FIGURE 4.4: EPMS - Db Structure - Teams

**Categories Table**

<b>Column</b>	<b>Type</b>	<b>Key</b>	<b>Extra</b>
ID	Int	Primary	AUTO INCREMENT
Row Version	TimeStamp		
Name	Nvarchar(255)		
IsActive	Bit		

FIGURE 4.5: EPMS - Db Structure - Employees

### Projects Table

Column	Type	Key	Extra
ID	Int	Primary	AUTO INCREMENT
Name	Nvarchar(100)		
Description	Nvarchar(500)		
Row Version	TimeStamp		
IsActive	Bit		
DevTeam	Bit		
Bit	Bit		
Operations	Bit		
Communications	Bit		
LocalSupportWest	Bit		
LocalSupportEast	Bit		
LocalSupportNorth	Bit		
Admin	Bit		
Trainers	Bit		
BusinessServicesDelivery	Bit		
BusinessServicesAnalysis	Bit		

FIGURE 4.6: EPMS - Db Structure - Projects

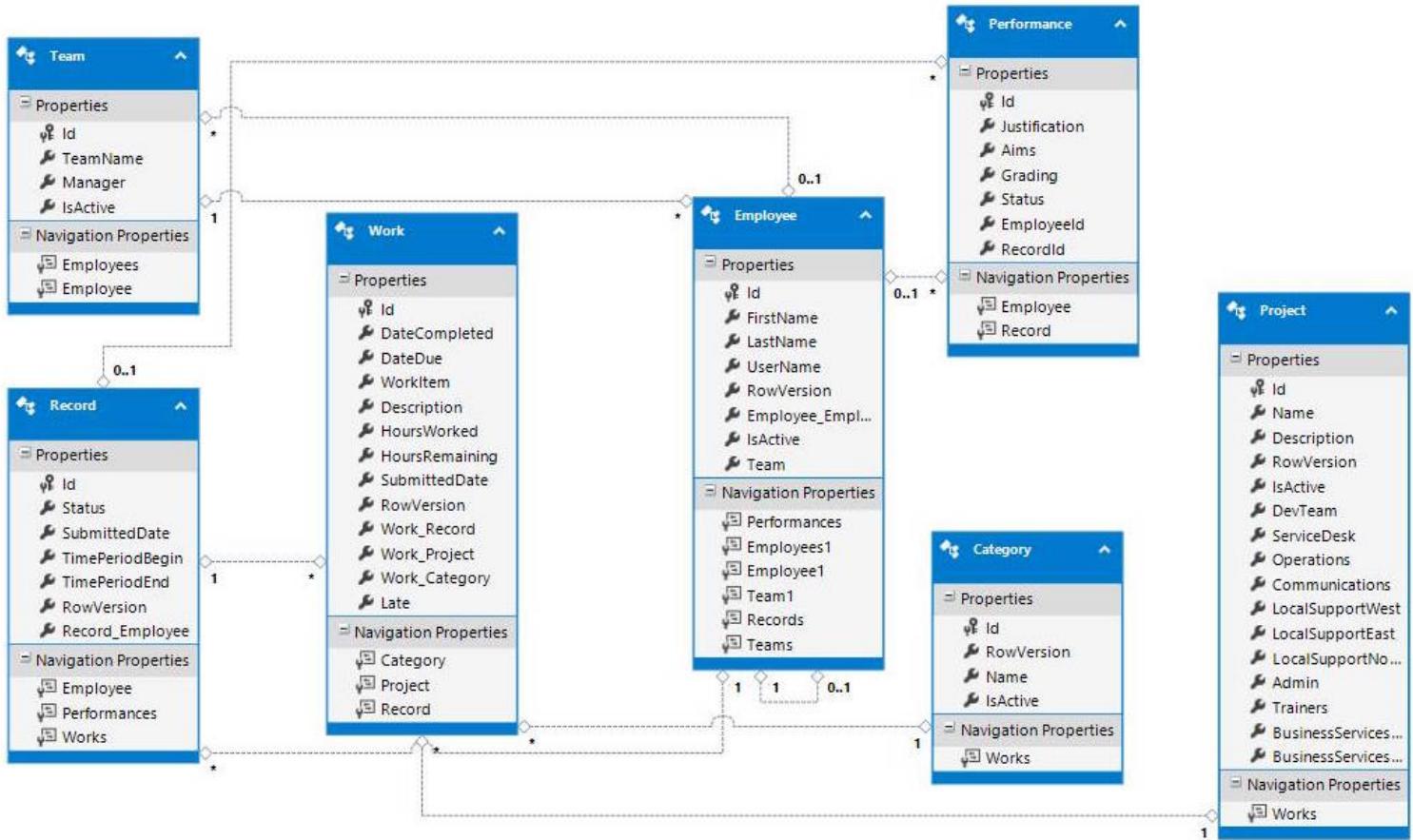


FIGURE 4.7: EPMS - Model Class Diagram

## 4.2 Design Methodologies

The project was designed using an Agile Methodology approach wherever possible. The EPMS is a complete Software Engineering project and Agile allows for an overall better quality of project at the end of the software development life cycle. Benefits of choosing Agile as a design methodology include[6]:

- Agile allows for an iterative design process
- High speed of production as there are regular releases of software throughout
- High quality of released software as each time functionality is added it is tested
- Good error management as small incremental releases allow issues to be seen early instead of approaching the end of the software development life cycle and finding many errors
- Good flexibility as change is always accepted by an Agile Methodology. This means that the requirements may change throughout and are discussed frequently at stand-up meetings (meetings with marker).

The system was developed in sprints and these can be seen on the project's trello board (Appendix K - Trello Board Screenshot). Each area of functionality in the EPMS was designed, implemented and built on for new functionality. The project also touched on some test driven development which meant that after each feature was incrementally added it was then tested. This was incorporated into the agile approach and worked particularly well and efficiently.

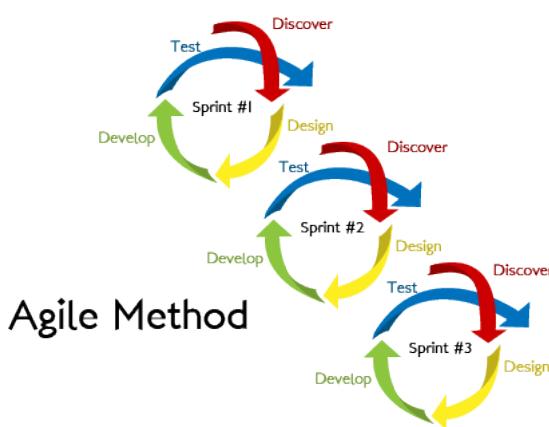


FIGURE 4.8: Agile

[5]

An alternative to Agile Design Methodology that was considered, is Waterfall Methodology. This involves a more sequential approach where each stage of the software development life cycle is carried out one by one, one

after the other. This requires each participant in the project to know accurate functional requirements and plan exactly what is going to be carried out at each stage. In waterfall, the end user's only involvement is at the beginning and if requirements change throughout then the whole process must start again.

This methodology is simple to understand and can be easy to use within small projects and works well. However, waterfall does not allow for any flexibility or change within the requirements limiting its capability for more complex and unpredictable projects.

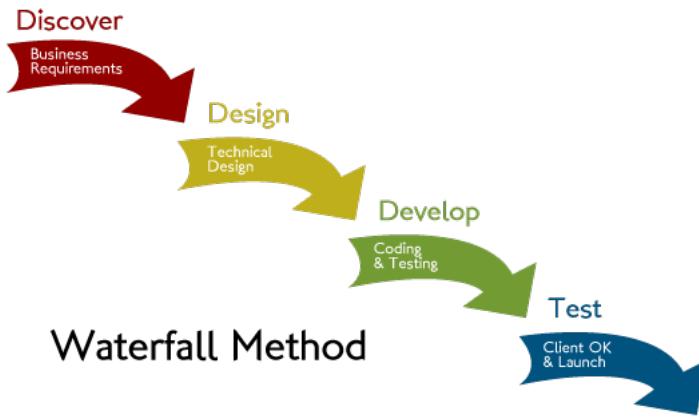


FIGURE 4.9: Waterfall

[5]

## 4.3 System Use Cases

The UML use-case diagrams below provide a high-level overview of the usage requirements of the system. A use-case is an action (or a step) that can be carried out by a user of the system. These are then used to create test cases in the testing stage of the system. In each diagram below you can see the use-cases that can be carried out by each type of user - Employees and Managers. Each use-case can be found in full detail in Appendix B - Use Case Descriptions.

### 4.3.1 Employees

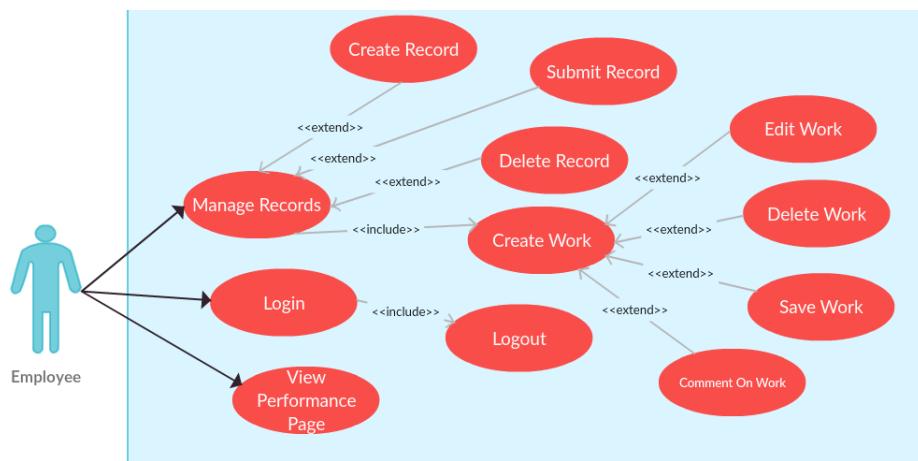


FIGURE 4.10: EPMS - UML Use-Case Diagram (Employees)

### 4.3.2 Managers

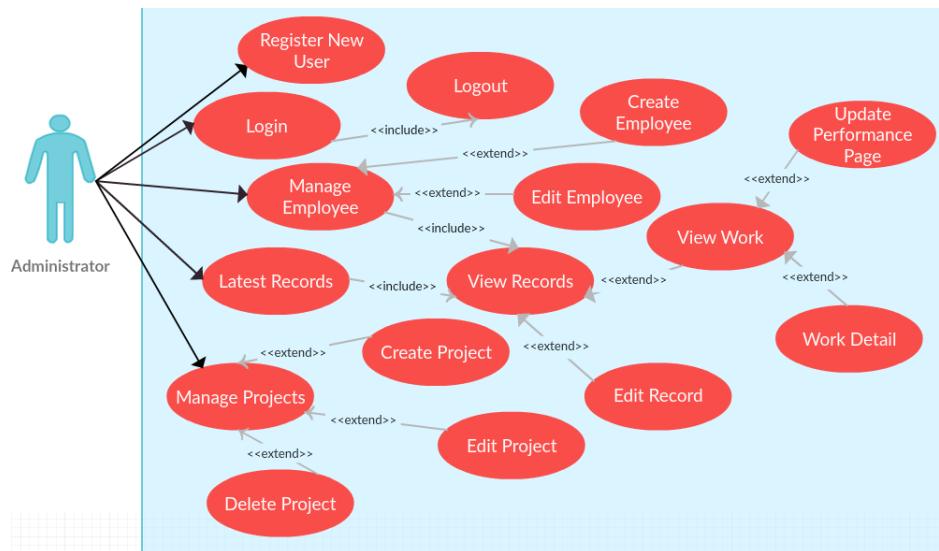


FIGURE 4.11: EPMS - UML Use-Case Diagram (Administrator)

## 4.4 User Interface Design

The user interface was developed using HTML5 and JavaScript making use of a front-end framework called Bootstrap. The user interface was designed to be as simple and appealing as possible. It has excellent performance over most platforms including mobile, but performs best in Google Chrome. See alternatives and ideas discussed in Appendix I - UI Mockups and Ideas.

### 4.4.1 About

An about page is included in the web application to give the user some information about the system and how to operate it.

### 4.4.2 Home Page

The home screen when a user logs in provides the opportunity to view work records via the button in the centre of the screen:

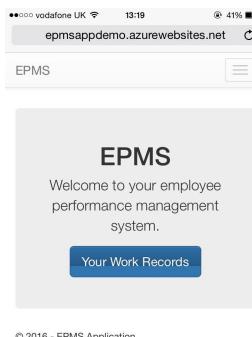


FIGURE 4.12: EPMS - Home Mobile

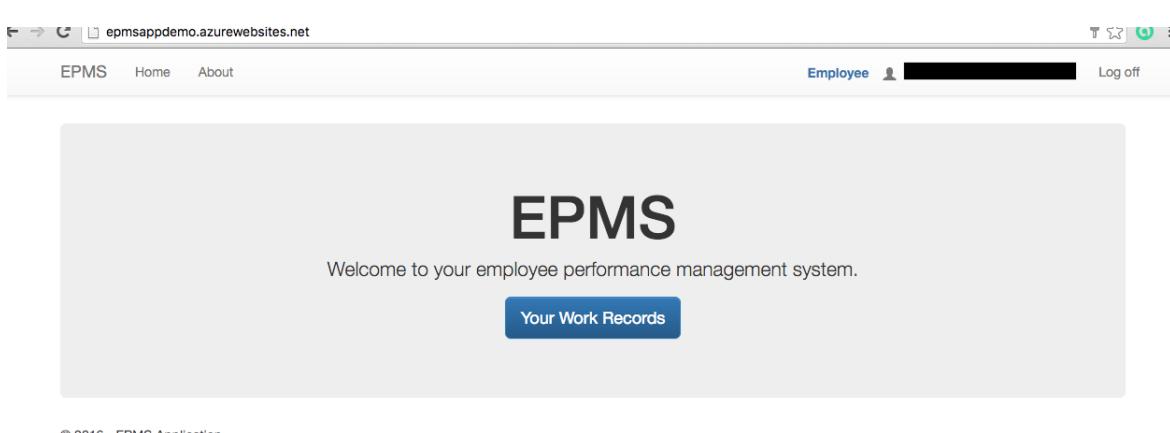


FIGURE 4.13: EPMS - Home

### 4.4.3 Datatables

The tables seen below in the views which show records and work are frequent throughout the system. When a manager views the team's employees or views the records in their submitted records page, they are presented with a datatable[7] that holds the data.

### 4.4.4 View Records

Records can be viewed in a datatable implemented using jQuery and JavaScript. This table can be sorted in terms of status. If a user wants to view all records that are "Satisfactory", they are able to do this by checking the radio button:

The screenshot shows a mobile browser interface for the EPMS application. At the top, there is a header bar with the URL 'epmsappdemo.azurewebsites.net' and a battery icon showing 39%. Below the header is a navigation bar with the text 'EPMS'. The main content area is titled 'My Records'. It features a search bar with the placeholder 'Search:' and a dropdown menu showing filter options: 'All' (selected), 'Open', 'Satisfactory', 'Unsatisfactory', 'Very Good'. Below the search is a table with three columns: 'TimePeriodBegin', 'TimePeriodEnd', and 'Status'. The table contains three rows of data:

TimePeriodBegin	TimePeriodEnd	Status
2016-03-11	2016-06-11	Unsatisfactory
2016-06-11	2016-09-11	Satisfactory
2016-09-11	2016-12-11	Open

At the bottom of the table, there are links for 'Create Next Record' and 'Back To Home'. A copyright notice at the very bottom reads '© 2016 - EPMS Application'.

FIGURE 4.14: EPMS - Employee Records Mobile

The screenshot shows a desktop browser interface for the EPMS application. The address bar displays the URL 'epmsappdemo.azurewebsites.net/Records'. The top navigation bar includes links for 'Employee' (with a user icon) and 'Log off'. Below the navigation is a header titled 'My Records'. It features a search bar with the placeholder 'Search:' and a dropdown menu showing filter options: 'All' (selected), 'Open', 'Satisfactory', 'Unsatisfactory', 'Very Good'. Below the search is a table with four columns: 'TimePeriodBegin', 'TimePeriodEnd', 'Status', and 'Work Created'. The table contains three rows of data:

TimePeriodBegin	TimePeriodEnd	Status	Work Created
2016-03-11	2016-06-11	Unsatisfactory	1
2016-06-11	2016-09-11	Satisfactory	2
2016-09-11	2016-12-11	Open	0

At the bottom of the table, there are links for 'Create Next Record' and 'Back To Home'. A copyright notice at the very bottom reads '© 2016 - EPMS Application'.

FIGURE 4.15: EPMS - Employee Records

#### 4.4.5 Open Records

Users can open records to view and manage all work inside. They will have the buttons/links to be able to create new work inside the record and/or edit/delete existing work. They will also have the option to submit the record for grading using the long blue submit button. This information is also shown in a datatable:

FIGURE 4.16: EPMS - Open Record Mobile

FIGURE 4.17: EPMS - Open Record

#### 4.4.6 Submitted Records

Users then submit records to be graded by managers. When a record has been submitted a lock is put onto it. Users can no longer amend the work in the record and the submit button is disabled:

The screenshot shows a mobile browser displaying a list of work items. At the top, there's a header bar with signal strength, battery level (12%), and the URL 'epmsappdemo.azurewebsites.net'. Below the header, the page title is 'EPMS'.

The main content area is titled 'Work (From 2016-09-22 To 2016-12-22)'. A message indicates 'This record has been submitted to admin.' Below this, there's a search bar and a table with the following columns: Workitem, Description, DateCompleted, Project, Category, and Late.

Workitem	Description	DateCompleted	Project	Category	Late
Supervisor at Tulliallan	As instructed I have been helping in a 2- month project to improved skill levels of service desk staff. Supervising the whole team went very well and we feel an improvement has been made already. Especially by Craig Heron.	2016-08-14	Tulliallan	Training	N

At the bottom of the table, there's a blue button labeled 'Submitted'.

FIGURE 4.18: EPMS - Submitted Record Mobile

The screenshot shows a desktop browser displaying the EPMS application. The address bar shows the URL 'epmsappdemo.azurewebsites.net/Works/Index/58'. The top navigation bar includes links for 'EPMS', 'Home', and 'About', along with a user profile and 'Log off'.

The main content area is titled 'Work (From 2016-09-22 To 2016-12-22)'. A message indicates 'This record has been submitted to admin.' Below this, there's a search bar and a table with the same columns as Figure 4.18.

Workitem	Description	DateCompleted	Project	Category	Late
Supervisor at Tulliallan	As instructed I have been helping in a 2- month project to improved skill levels of service desk staff. Supervising the whole team went very well and we feel an improvement has been made already. Especially by Craig Heron.	2016-08-14	Tulliallan	Training	N

At the bottom of the table, there's a blue button labeled 'Submitted'.

FIGURE 4.19: EPMS - Submitted Record

#### 4.4.7 Graded Records

After the record has been viewed by a manager, it is then assigned a performance page and a grade. The performance page can be viewed using the long button across the screen. Depending on the grading of the record the button may have a different colour. Satisfactory and very good records are the colour green, whereas unsatisfactory records are the colour red:

Work Item	Description	Date Completed	Project	Category	Late
Service Manager	Training course in Glasgow to help wit...	2016-03-22	Balsamiq Training	Project Work	N
HR Testing	Testing the new HR system using the pro...	2016-05-01	HR System	Evaluation Category	N

FIGURE 4.20: EPMS - Graded Record Mobile

WorkItem	Description	DateCompleted	Project	Category	Late
Service Manager	Training course in Glasgow to help wit...	2016-03-22	Balsamiq Training	Project Work	N
HR Testing	Testing the new HR system using the pro...	2016-05-01	HR System	Evaluation Category	N

FIGURE 4.21: EPMS - Graded Record

#### 4.4.8 View Performance Page

Users are able to view their performance grading for previous records which will contain all the information they need to continue work for the next three months:

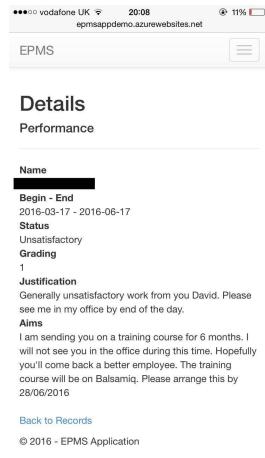


FIGURE 4.22: EPMS - Performance Page Mobile

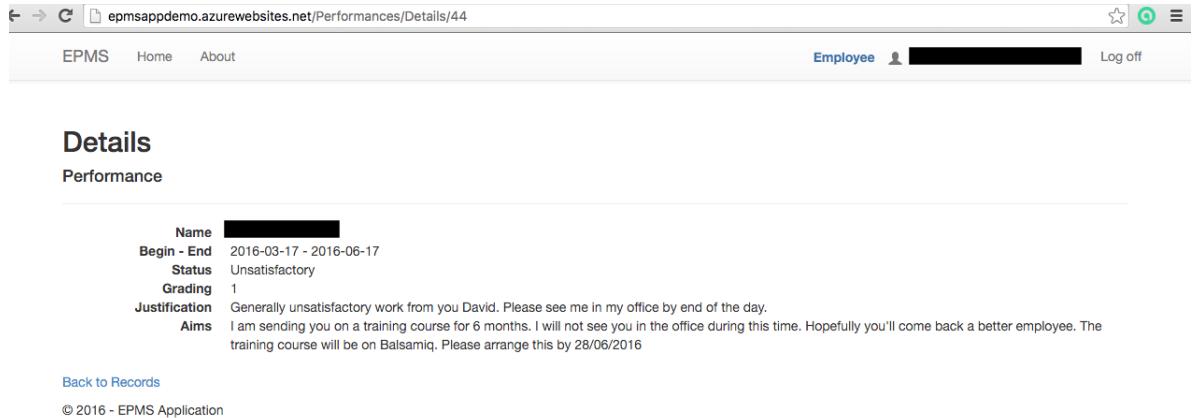


FIGURE 4.23: EPMS - Performance Page

#### 4.4.9 Forms

Forms throughout the system are created with similar styles and validation. This can be seen below in the create performance page form and the login form. These are the same types of forms used throughout for amending work, employees and projects:

The screenshot shows a mobile browser displaying the EPMS login page. The URL in the address bar is 'epmsappdemo.azurewebsites.net'. The page title is 'EPMS'. Below it, the text 'Log in.' is displayed, followed by the instruction 'Use your EPMS account to login.' There are two input fields: 'Email' containing a placeholder email address and 'Password' containing a masked password. A 'Remember me?' checkbox is present, followed by a blue 'Log in' button. At the bottom, the copyright notice '© 2016 - EPMS Application' is visible.

FIGURE 4.24: EPMS - Login Form Mobile

The screenshot shows a web browser displaying the 'Create Performance' form. The URL in the address bar is 'epmsappdemo.azurewebsites.net/Performances/Create/40'. The page title is 'EPMS'. The top navigation bar includes links for 'Home', 'About', 'Administration', 'Team Manager' (logged in as 'manager1@hotmail.co.uk'), and 'Log off'. The main form has sections for 'Grading' (with a dropdown menu showing '0'), 'Status' (a dropdown menu), 'Justification' (a large text area), and 'Aims' (another large text area). At the bottom is a blue 'Create' button.

FIGURE 4.25: EPMS - Create Performance Form

# Chapter 5

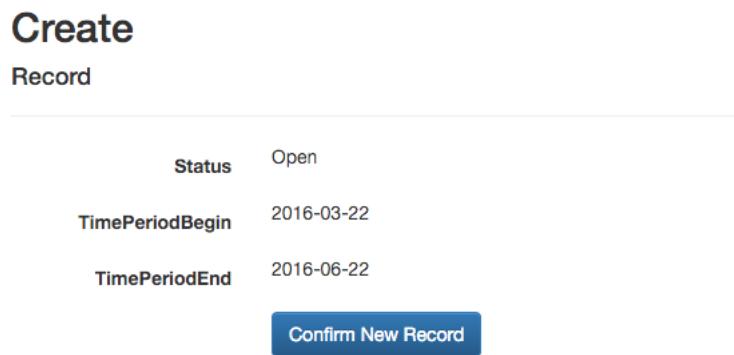
## Detailed Design & Implementation

### 5.1 Submit Work Feature

In order to allow managers to provide feedback on the work completed by employees, it is required that employees must submit their work to be graded. This section focuses on how this happens.

#### 5.1.1 Records

Firstly, a record is created. This record will last for three months and is where all the completed work will be held. When a link is clicked on the UI to create a new record, a method in the Records Controller is called. The "Confirm New Record" button calls the "Create" method in the records controller (see below):



The screenshot shows a web-based application interface for creating a new record. At the top, it says "Create Record". Below that is a table with three rows of data:

Status	Open
TimePeriodBegin	2016-03-22
TimePeriodEnd	2016-06-22

At the bottom of the form is a blue "Confirm New Record" button. At the very bottom of the page, there are links for "Back to List" and copyright information: "© 2016 - EPMS Application".

FIGURE 5.1: EPMS - Create Record

```

// GET: Records/Create
public ActionResult Create()
{
    //get the current user's id
    var getUser = db.Employees.First(i => i.UserName == User.Identity.Name).Id;
    //get the records for the current user
    var records = db.Records.Where(i => i.Record_Employee == getUser);
    //Count the number of records
    var noRecords = db.Records.Count(i => i.Record_Employee == getUser);
    //Get the date begin for the record (now)
    var startDate = DateTime.Now;
    //If no records exist then the start date is now
    if (noRecords == 0)
    {
        startDate = DateTime.Now;
    }
    //If there are already records for the user then the start date must be for the next three months
    else if (noRecords > 0)
    {
        var getFirstRecord = records.Max(i => i.Id);
        var getPreviousRecordTimeBegin = records.First(i => i.Record_Employee == getUser && i.Id == getFirstRecord).TimePeriodBegin;
        var getPreviousRecordTimeEnd = records.First(i => i.Record_Employee == getUser && i.Id == getFirstRecord).TimePeriodEnd;
        startDate = getPreviousRecordTimeBegin.AddMonths(3);
    }
    //Create the new record that will be passed to the post method
    Record record = new Record()
    {
        //Set all values of new record
        Id = db.Records.Max(i => i.Id + 1),
        Status = "Open",
        TimePeriodBegin = startDate,
        TimePeriodEnd = startDate.AddMonths(3),
        Record_Employee = getUser,
        SubmittedDate = DateTime.Now
    };
    return View(record);
}

```

FIGURE 5.2: EPMS - Get Method Create Record

```

// POST: Records/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create(Record record)
{
    record.Status = "Open";
    //Start of try catch method to add the record to the database
    try
    {
        if (ModelState.IsValid)
        {
            db.Records.Add(record);
            db.SaveChanges();
            return RedirectToAction("Index");
        }
    }
    //If the error is caught then display meaningful error message (as well as custom error page)
    catch (DbEntityValidationException e)
    {
        // Retrieve the error messages as a list of strings.
        var errorMessages = e.EntityValidationErrors
            .SelectMany(x => x.ValidationErrors)
            .Select(x => x.ErrorMessage);

        // Join the list to a single string.
        var fullErrorMessage = string.Join("; ", errorMessages);

        // Combine the original exception message with the new one.
        var exceptionMessage = string.Concat(e.Message, " The validation errors are: ", fullErrorMessage);

        // Throw a new DbEntityValidationException with the improved exception message.
        throw new DbEntityValidationException(exceptionMessage, e.EntityValidationErrors);
    }
    return View(record);
}

```

FIGURE 5.3: EPMS - Post Method Create Record

### 5.1.2 Work

Each record lasts for three months and includes each piece of work completed in this time. One of the challenges that occurred with creating work was that when a new piece of work was created, the system was always trying to give it an ID of 0 in the database. This threw an error almost every time (except the first time) as there was an existing piece of work with the ID of 0. This issue was solved by finding the ID of the last entry in the database and incrementing this to be the new ID. The work to be included inside the record is created in a similar way to the records with much more user input. Employees are created in the system in the same way (see below):

**Create**

Work

---

DateDue	25/06/2016 (If not specified, DateDue = today's date)
DateCompleted	28/06/2016
WorkItem	Organisation OS Windows 10
Description/Comments	This project involved working with John Smith in the Operations team rolling out 2 new operating systems to the organisation. The project did not go smoothly. John and I completed the piece of work but was three days late.
Project	New Operating System
Category	Project Work

**Create**

[Back to List](#)  
© 2016 - EPMS Application

FIGURE 5.4: EPMS - Create Work

```

// GET: /EPMSDev4/Create

public ActionResult Create(int id = 0)
{
    //Call method to populate the dropdown menus
    ProjectList();
    ViewBag.Work_Category = new SelectList(db.Categories.Where(i => i.IsActive == true).OrderBy(x => x.Name), "Id", "Name");
    ViewBag.Work_TimeSheet = new SelectList(db.Records, "Id", "Status");

    //Get the record values for time begin and time end
    timebegin = db.Records.First(i => i.Id == id).TimePeriodBegin;
    timeend = db.Records.First(i => i.Id == id).TimePeriodEnd;

    //Create the new work
    Work work = new Work()
    {
        tempTimeBegin = timebegin,
        tempTimeEnd = timeend,
        Id = -1,
        Work_Record = id,
        HoursWorked = 0,
        SubmittedDate = DateTime.Now
    };

    //Pass the new work to the post method
    return View(work);
}

```

FIGURE 5.5: EPMS - Post Method Create Work

```

// POST: /EPMSDev4/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create(Work work)
{
    if (ModelState.IsValid)
    {
        //Check for late flag
        if (work.DateCompleted > work.DateDue)
        {
            work.Late = 1;
        } else
        {
            work.Late = 0;
        }

        work.HoursWorked = 0;
        work.Id = db.Works.Max(i => i.Id + 1);
        db.Works.Add(work);
        //Try to save the changes in the db and if not display meaningful error message
        try
        {
            db.SaveChanges();
        }
        catch (DbEntityValidationException ex)
        {
            StringBuilder sb = new StringBuilder();

            foreach (var failure in ex.EntityValidationErrors)
            {
                sb.AppendFormat("{0} failed validation\n", failure.Entry.Entity.GetType());
                foreach (var error in failure.ValidationErrors)
                {
                    sb.AppendFormat("- {0} : {1}", error.PropertyName, error.ErrorMessage);
                    sb.AppendLine();
                }
            }
        }
    }
}

```

FIGURE 5.6: EPMS - Post Method Create Work

When the "Create" button is clicked to create a new piece of work (see above), the "Create" method is called in the work controller which then creates the piece of work and adds it to the database.

### 5.1.3 Submission

When the end of the three month period approaches (and the record is full of completed work), the next step is to submit the record to the line manager. The user can do this by simply clicking the long submit button along the bottom of record table:

WorkItem	Description	DateCompleted	Project	Category	Late
Supervisor	As instructed I have been helping in a 2- month project to improved skill levels of service desk staff. Supervising the whole team went very well and we feel an improvement has been made already. Especially by Craig Heron.	2016-06-25	Tulliallan	Training	Y

FIGURE 5.7: EPMS - Open Record

Clicking the button in the view calls the following method in the record controller. The method takes in the record ID and searches the database for the record. It then changes the status of the record to submitted:

```
// GET: Records/Edit/5
public ActionResult SubmitRecord(int id)
{
    Record record = db.Records.Find(id);

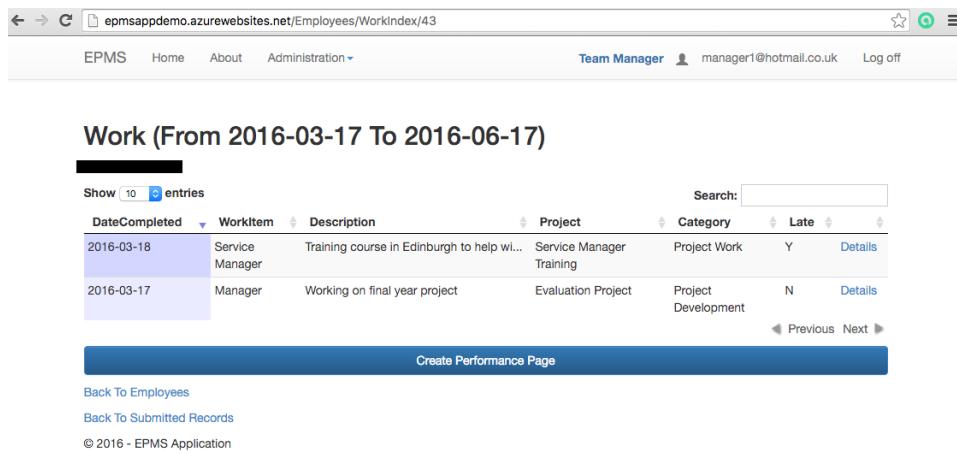
    record.Status = "Submitted";

    if (record == null)
    {
        return HttpNotFound();
    }
    db.Entry(record).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("Index");
}
```

FIGURE 5.8: EPMS - Submit Record Method

## 5.2 Feedback Process

The feedback process begins with managers accessing employee's records and inspecting the work that has been submitted. They will view the submitted record via "Managing Employees" screen or "Submitted Records" queue. When managers open a submitted record they click on the button to create a performance page and, when completed (see form above), this calls the "Create" methods in the performance controller:



The screenshot shows a web browser window for the EPMS application at the URL [epmsappdemo.azurewebsites.net/Employees/WorkIndex/43](http://epmsappdemo.azurewebsites.net/Employees/WorkIndex/43). The page title is "Work (From 2016-03-17 To 2016-06-17)". The header includes links for EPMS, Home, About, Administration, Team Manager, and Log off. The main content displays a table of work items with columns: DateCompleted, WorkItem, Description, Project, Category, Late, and Details. Two entries are shown: one for 2016-03-18 (Service Manager) and another for 2016-03-17 (Manager). Below the table is a blue button labeled "Create Performance Page". At the bottom of the page are links for Back To Employees, Back To Submitted Records, and a copyright notice: © 2016 - EPMS Application.

FIGURE 5.9: EPMS - Grade Record

```
// GET: Performances/Create
public ActionResult Create(int id)
{
    ViewBag.EmployeeId = new SelectList(db.Employees, "Id", "FirstName");
    //Get the employee id
    var empId = db.Works.First(i => i.Work_Record == id).Record.Record_Employee;
    Performance performance = new Performance();
    performance.RecordId = id;
    performance.EmployeeId = empId;

    return View(performance);
}

// POST: Performances/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Id,Justification,Aims,Grading,Status,EmployeeId,RecordId")] Performance performance)
{
    //Create the new performance record in the db and redirect to change the status of the record
    if (ModelState.IsValid)
    {
        performance.Id = db.Performances.Max(i => i.Id + 1);
        db.Performances.Add(performance);
        db.SaveChanges();
        return RedirectToAction("UpdateRecord", "Records", new { id = performance.RecordId });
    }

    ViewBag.EmployeeId = new SelectList(db.Employees, "Id", "FirstName", performance.EmployeeId);
    return View(performance);
}
```

FIGURE 5.10: EPMS - Create Performance Method (Performance Controller)

When the create method in the performance controller completes execution, control is then passed back to the record controller to update the record with the new grade - "Update Record" method. This method takes in the ID of the record and pulls the new status from the performance page just created. This is then assigned to the record and the database changes are saved:

```
// GET: Records/Edit/5
public ActionResult UpdateRecord(int id)
{
    var getNewStatus = db.Performances.First(i => i.RecordId == id).Status;
    Record record = db.Records.Find(id);

    record.Status = getNewStatus;

    if (record == null)
    {
        return HttpNotFound();
    }
    db.Entry(record).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("email", "Performances", new { id = record.Id });
}
```

FIGURE 5.11: EPMS - Update Record Method (Record Controller)

Next, at the end of this method get and post "Email" methods are called in the performance controller which sends an email notification to the employee with their grade and a link to the system[8]:

```
//Method that is triggered when a manager grades a record. This creates the email to be sent
public async Task<ActionResult> email(FormCollection form, int Id)
{
    var getRecordEmployee = db.Records.First(i => i.Id == Id).Record_Employee;
    var getGrade = db.Performances.First(i => i.RecordId == Id).Status;
    var getEmployeeName = db.Employees.First(i => i.Id == getRecordEmployee).FirstName;
    var name = form["sname"];
    var email = form["semail"];
    var messages = "Dear " + getEmployeeName + ",\n\nA recent record has been graded as " + getGrade + " by your manager. Please see http://epmsappde";
    var phone = form["sphone"];
    var x = await SendEmail(name, email, messages, phone, Id);
    if (x == "sent")
        ViewData["esent"] = "Your Message Has Been Sent";
    return RedirectToAction("RecordIndex", "Employees", new { id = getRecordEmployee });
}
```

FIGURE 5.12: EPMS - Email Get Method (Performance Controller)

```
//Method sends the email
private async Task<String> SendEmail(string name, string email, string messages, string phone, int Id)
{
    var getRecordEmployee = db.Records.First(i => i.Id == Id).Record_Employee;
    var getEmployeeEmail = db.Employees.First(i => i.Id == getRecordEmployee).UserName;
    var getEmployeeName = db.Employees.First(i => i.Id == getRecordEmployee).FirstName;
    var message = new MailMessage();
    message.To.Add(new MailAddress(getEmployeeEmail)); // replace with receiver's email id
    message.From = new MailAddress("EPMSdonotreply@outlook.com"); // replace with sender's email id
    message.Subject = "EPMS Email Notification";
    message.Body = messages;
    message.IsBodyHtml = true;
    using (var smtp = new SmtpClient())
    {
        var credential = new NetworkCredential
        {
            UserName = "EPMSdonotreply@outlook.com", // sender's email id
            Password = "Txtekgsn1" // Sender password
        };
        smtp.Credentials = credential;
        smtp.Host = "smtp.live.com";
        smtp.Port = 587;
        smtp.EnableSsl = true;
        await smtp.SendMailAsync(message);
        return "sent";
    }
}
```

FIGURE 5.13: EPMS - Email Post Method (Performance Controller)

## 5.3 Technologies Used

This section will focus on discussing the main technologies used to produce the project, their functionality and the role they had to play in the project.

### 5.3.1 Visual Studio

Visual Studio is an integrated development environment from Microsoft that includes a number of different development tools to create many computer programs including C#.NET applications.

Visual Studio was used broadly in the creation of this project and helped with the integration of entity framework, GitHub and many more useful features.

### 5.3.2 Razor/Entity Framework

Entity framework is a mapper that was created to eliminate friction that occurs when .NET developers are working with a web application and relational data. The Microsoft ADO .NET Entity Framework makes it easier for developers to treat relational data as domain specific objects.

Developers use LINQ[9] (as discussed below) to query and manipulate data from the database and treat them as strongly typed objects. Entity Framework is an enhancement to .NET that gives developers an automated mechanism for accessing and storing data in the database.

In the case of the EPMS, entity framework was useful as the database for this project had been created in advance of the web application. This meant that after the creation of the database the entity framework was also able to generate classes from the database objects.

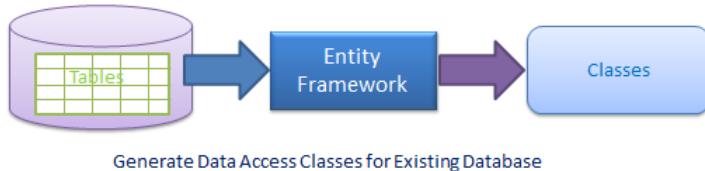


FIGURE 5.14: Entity Framework Overview

### 5.3.3 Bootstrap & CSS

Bootstrap is a front-end framework that is very popular among web application developers. It is an open-source modular collection of style sheets. The CSS/JS/HTML files are all customizable but keep the web application design responsive and this allows it to function on a number of different devices, including ones which are mobile.

Bootstrap[11] is used throughout the design of the front-end of the EPMS web application and can be seen clearly in the forms. JQuery integrates nicely with the Bootstrap UI to make forms look more attractive. Examples of this can be seen in the date pickers and dropdown menus throughout the system. Bootstrap also provides good styling of the form validation. Bootstrap combined with some JavaScript provided the form validation used in this project.

### 5.3.4 SVN/Git

Version control is essential in any software development project. This is a repository of files containing the source code from a project which is updated with every change made to the code. Version control is able to keep track of who made the changes and why they were made.

There were two options considered for version control during the planning and design stage for this project - Subversion (SVN) and GitHub. However, both were used in this case due to an issue with SVN. Initially, it was chosen that SVN would be used to track changes in the source code but after a few commits, an issue arose with the repository in that when the code was checked out, some of the references included in the project were missing. A GitHub repository[12] was then set up to include the source code and this was used until the end of the project with no issues. The Visual Studio Github plugin was also installed which allowed straight forward commits and pulls to take place within the IDE.

An issue occurred late in the project where the "csproj" file became corrupt and the project was unable to load. This meant that the project could not load at all when deploying the system, azure would fail. Given the time left before the hand-in date, there was very little (if anything) that could be done to rebuild the file from scratch. The problem was caused by an unexpected crash that took place in the operating system and the file became corrupt. The problem was solved by restoring the last working commit from the Git repository. This issue shows the importance of version control in software development and the vital role it plays in the cycle.

Commit history from both SVN and Git repositories can be found in Appendix D.

### 5.3.5 C#.NET

This is what is used to write most of the EPMS web application. C# is an object-oriented programming language that enables developers to write applications that run on Microsoft's .NET framework. The syntax of C# .NET is simple and easy to learn, especially to developers that are familiar with languages such as C++ and Java.

In terms of this EPMS, C# is used to write each of the controllers, the models and also used in conjunction with other languages in this section to produce the views in the solution.

```
public partial class Work
{
    public int Id { get; set; }
    [Required]
    [DataType(DataType.Date)]
    [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
    public Nullable<DateTime> DateCompleted { get; set; }
    [Required]
    [DataType(DataType.Date)]
    [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
    public Nullable<DateTime> DateDue { get; set; }
    [Required]
    public string WorkItem { get; set; }
    [Required]
}
```

FIGURE 5.15: EPMS - Code Extract From Work Model

In order to pull data into the application, LINQ is used within C# to query the database. LINQ (Language-Integrated Query) is one of many features provided that allow developers to query and update databases. Visual Studio allows for communication between LINQ, .NET framework collections and SQL Server databases, like the one used for storing data for the EPMS. Using reference, "System.Linq" SQL Server databases can be accessed. An example of LINQ can be seen below querying the employees table to return their records:

```
// GET: Records
public ActionResult Index()
{
    //Get the current logged in user from the db
    var username = @System.Web.HttpContext.Current.User.Identity.Name;
    //See if they exist in the db
    var searchForUser = db.Employees.Count(i => i.UserName == username);

    if (username != "" && searchForUser != 0)
    {
        //get the current user's id
        var getUser = db.Employees.First(i => i.UserName == username).Id;

        //grab all records for this user
        var records = db.Records.Where(i => i.Record_Employee == getUser);

        //return a list of these records
        return View(records.ToList());
    }
}
```

FIGURE 5.16: EPMS - LINQ Example

### 5.3.6 JavaScript

JavaScript is a cross-platform scripting language that is very small and light-weight. JavaScript can be connected to objects within the browser and can be used to control and manipulate them.

The data throughout the EPMS web application is mostly displayed through jQuery datatables which are created and controlled in JavaScript (see below). JavaScript is also used throughout for extra form validation. Another example of JavaScript being used in the EPMS is setting the max and min values on the datepickers.

## Work (From 2016-06-11 To 2016-09-11)

This record has been submitted to admin.

WorkItem						Description	DateCompleted	Project	Category	Late
Windows 10 Training						As instructed in previous performance page, I attended the Windows 10 training course in Glasgow City Centre. This went very well and I will be able to bring these skills into our work place.	2016-06-22	New Operating System	Training	N
Office Work						Tidied up the office	2016-08-20	Other	General Maintenance	N

Submitted

[Back to Records](#)

© 2016 - EPMS Application

FIGURE 5.17: EPMS - JQuery DataTable

### 5.3.7 MVC 5

MVC (Model View Controller) is a design pattern that separates an application into three main components - Models, Views and Controllers.

This project uses the ASP .NET MVC 5 framework to produce a structured web application. The MVC framework is designed using the "System.Web.Mvc" assembly in which, the user interaction with the view raises events in the controller which, in turn, updates the model.

Both inheritance and composition are also used within the application. Composition occurs more frequently.

Inheritance can be seen in the controller classes where they inherit their methods from an abstract class "Controller".

Composition is used throughout where methods in different controllers are accessed. This means that code is not used over and over making the solution cleaner.

[13]

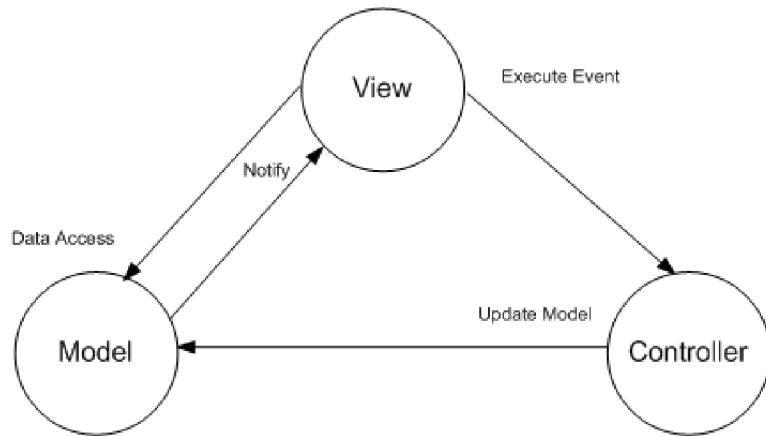


FIGURE 5.18: MVC Overview

One of the many good reasons for choosing this design pattern when designing the EPMS web application was that it supports test-driven development. As discussed in chapter 4, TDD was used throughout and the MVC framework decouples components, making it easier to test individual components in isolation from the rest of the application.

For class diagrams, see Appendix E.

### 5.3.8 SQL Server

SQL Server is a Microsoft-based relational database management system used to store, manage and maintain information for a number of different purposes. The relational aspect of the database management system is that it allows for tables to be linked, or "Related" to one another.

SQL Server 2014 was used in the creation of this project and provided the storage for most of the data that the application uses. Tables that were created in the SQL Server database "EPMSDev" included - Employees, Records, Work, Performances etc. This SQL database is hosted on Microsoft Azure alongside the application and the MySQL database "IdentityMySQLDatabase" which is used to store the login and authentication information. SQL Server Management Studio 2014 was used in conjunction with Visual Studio to integrate the database and web application together allowing them to communicate efficiently and manage the data properly.

Although, using a SQL Server database for most of the data used by the EPMS web application was the likely outcome - other options had to be considered. MySQL was also considered to hold this data. Both are relational databases that are very popular with web application developers. SQL Server was chosen for this project mainly due to its easy integration with Visual Studio.

### 5.3.9 MySQL

MySQL is an open source relational database management system and is based on SQL. Running on most platforms, MySQL can be used within a wide range of applications. However, it is most popular for being used as

part of web applications due to its many useful attributes and is one of the reasons it was chosen for this project.

All the account and authentication data for the EPMS is held in a MySQL database that is hosted on Microsoft Azure (as you can see below). Authentication information stored in the MySQL database includes Email, Username and Password and this is created in the code using "AspNet.Identity.MySQL"[14]. This authentication reference generates the ASP .Net Identity Store and uses a MySQL database to hold the data. In order to manage and maintain the data, MySQL Workbench was also used.

Other options considered for this were MongoDB (non-relational database) and Microsoft SQL Server. MySQL was chosen to store this data over MongoDB due to previous experience and familiarity with relational databases. Furthermore, the dataset to be held for authentication and account datasize was relatively small and datatypes weren't too complex. These attributes favour relational databases.

### 5.3.10 Microsoft Azure

Microsoft Azure[17] is a cloud computing platform designed by Microsoft used for building, deploying and managing applications held in data centres all over the world.

The EPMS web application itself, is hosted on Microsoft Azure alongside a MySQL database (for authentication) and a SQL Server database (for all other data related to the system). These were deployed directly from Visual Studio via an Azure SDK[19] provided by Microsoft to make it as easy as possible to deploy to the cloud from Visual Studio. Other options considered when deciding where the project would be hosted were Digital Ocean and Strathclyde's DevWeb2015. Both of these are Linux servers. Therefore, the .NET application and SQL Server database (both Microsoft products) were unable to be hosted on them. It would have been possible to hold the MySQL database on either of these Linux servers due to it being open-source. However, it was decided that everything would be hosted in the one place and that Azure was to be that place[18].

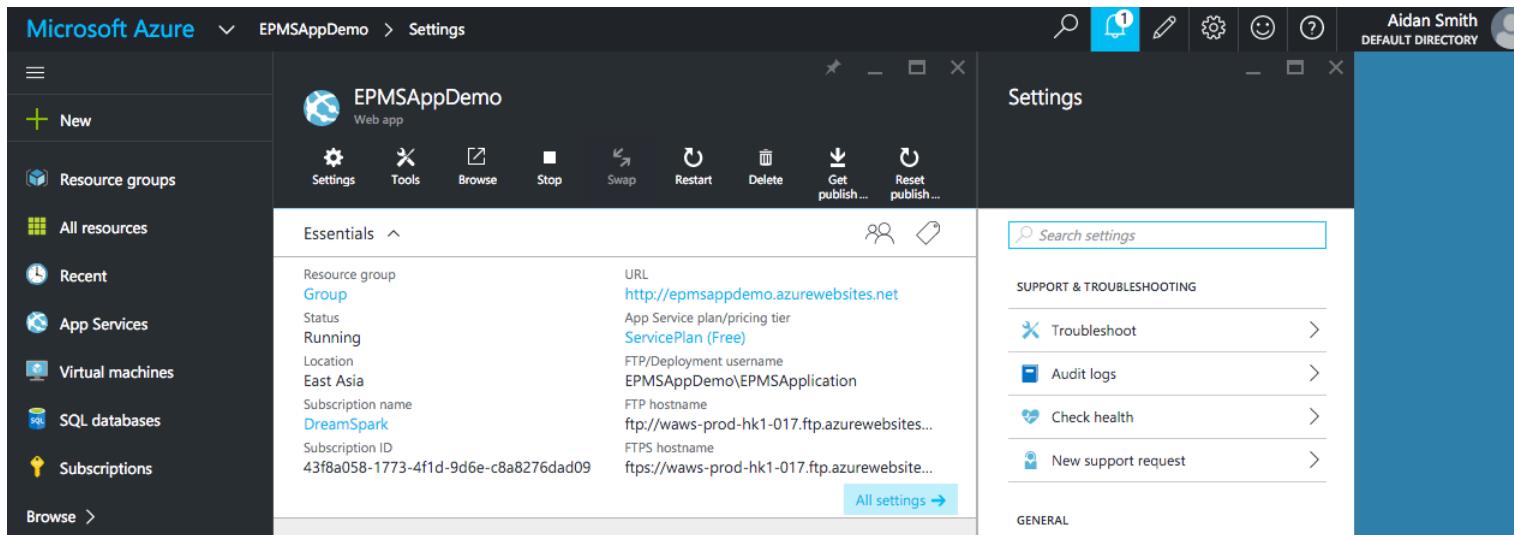


FIGURE 5.19: Azure - EPMS Subscription

[18]

Hosting data in the cloud involves placing applications and services on a virtual storage area. Benefits of hosting in the cloud over an in-house server include[20]:

- There is no need for own hardware expenses due to no physical server components needed
- Microsoft support is available with Azure storage
- Easily scalable - cloud space can be increased/decreased very easily to meet any changes in demands
- Hosting in the cloud removes any stress of physically managing the server - In this case, Microsoft are responsible for this

## 5.4 Security

There are many security features implemented throughout the EPMS web application. Security is important in any system as it involves detecting and preventing unauthorised use of the software. The main aim for security within this system was to make sure that all users that are logged in are only exposed to the data/pages that they should be.

Authentication in the EPMS web application is implemented throughout with the help of ASP.Net Identity[14]. This is a security feature provided by Microsoft to help implement authentication that integrates with entity framework. This has been used here to implement secure authentication for users but has many other features that could be used but weren't in this case. Some of these can be seen in section 3.5. Accounts for the system can only be created by line managers, adding an extra layer of security so that users are not able to create their own accounts which prevents any fraud or falsified users. If a manager wants to create a new user on the system it is also required for them to add the user to their team in the "Manage Employees" section. This also adds to the security of the system.

After user authentication has been checked and approved the system searches the database to check if the logged-in user is an employee or a manager. This then determines what they will be able to do in the system. For example, if a user is an employee then they will only have access to the basic features provided by the system. However, if a user is a manager in the database, then they will have access to more features such as managing employees and grading records.

Form validation is implemented throughout the system to prevent users entering wrong datatypes and throwing errors. Ideally, users fill out all forms correctly with no mistakes. However, this is not always the case and this is where web form validation is important. EPMS implements both server and client side validation. Server-side is used when a field in the model is created as being "Required". This means that it must be entered when the information is being submitted. The validation error below appears in the EPMS web application if the form is submitted without entering a value for the work item.

The screenshot shows a user interface element labeled "WorkItem" with a red border around it. Below the input field, the text "The WorkItem field is required." is displayed in red, indicating a validation error.

FIGURE 5.20: EPMS - Server Side Required

Another type of form validation used is client side validation. This involves validation that is created via scripting languages such as JavaScript meaning that errors are shown as the users type. This is very responsive and produces visually good validation. The main limitation of client side validation is that it relies on JavaScript and if users disable JavaScript, then they are able to bypass the validation. This is the main reason server side validation was chosen - the web application is more secure using server side validation. A better option would have been to implement both server side and client side which would have given the EPMS advantages of both types of validation. However, this was something that was not essential (see section 3.5 - Future Features) and was classed as a "could have" feature in the gathering requirements stage.

To improve the security of the application, protection against cross-site scripting could have been implemented. Cross-site scripting is an attack in which malicious scripts are injected into forms on a website. This kind of protection would be essential for this web application should it be released into a professional environment in which users would be using it on a day-to-day basis.

## 5.5 Error Handling

It is inevitable that some errors will arise in any software system and this case is no different. Error handling involves anticipating, detecting and handling errors appropriately. Some errors are simple and can be dealt with relatively easily. An example of a small error in EPMS is an invalid login attempt and can be seen in the figure below.

The screenshot shows a login form with the heading "Log in.". Below the heading is the instruction "Use your EPMS account to login.". A red error message "• Invalid login attempt." is displayed above the "Email" input field. The "Email" field contains "mfb11127@uni.strath.ac.uk". The "Password" field is empty. A "Remember me?" checkbox is present but unchecked. A blue "Log In" button is at the bottom.

FIGURE 5.21: EPMS - Invalid Login

These are smaller validation errors that are implemented in the backend of the application through the assembly of the ASP.Net Identity and can be changed easily in the bootstrap CSS.

Error messages in any application alerts users of any problems that have occurred when trying to execute instructions. If an error is caught then it is important to display a meaningful message to the user that they will be able to react to and make sense of. In the EPMS web application, a custom error page was created that is more attractive and is displayed to the user when an error is caught. The error page shown in the figure below is one which is shown when an error is caught in the backend for example a database connection error or if the email server fails:

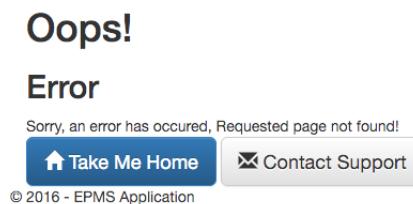


FIGURE 5.22: EPMS - Custom Error Message

This is a custom error page created using HTML5 and Bootstrap that gives users the option to go to the homepage or contact support via email. The custom error pages can change depending on the type of error. In this case, there was no need to check for the type of error but in other systems it would be required to check the error number and produce the appropriate error message for the user.

## 5.6 Database

For users of the system to be able to operate it effectively it was essential to use a database to hold its data. There are two types of database used - Microsoft SQL Server and MySQL.

Data used by the system itself can be found on a SQL Server database hosted on Microsoft Azure and the structure of the tables on the database can be seen in section 4.1 - Database Design. This data was managed and manipulated using Microsoft SQL Server Management Studio and an integration language called LINQ (Read more about this in section 5.3 - Technologies used). A smaller MySQL database is used to store authentication data such as usernames/emails and passwords. To manage this data, MySQLWorkbench was used. A number of database techniques were used in order to organise the information that was kept. One of these was normalisation. Normalisation involves organising the data by creating tables and establishing relationships between these tables to increase flexibility and eliminate redundant dependencies.

### 5.6.1 Database Challenges

A few of the challenges faced throughout the design and implementation of the EPMS web application were due to database issues. One of the biggest problems was that when deploying the SQL Server database to Azure, the server was incompatible and there was an error when trying to access the live database. This was quickly solved by installing SQL Server 2014 on Azure and manipulating the project in Visual Studio to access the 2014 version of SQL Server.

Another issue faced was that when users created a performance page, there was no way of displaying which record it was that had been graded. After investigation it became obvious that there was no connection between the performance pages and records in the database. This was solved by creating a foreign key in the performance table which referenced the primary key of the records table. This meant that the two tables were now related and the system could easily pull the record ID from the performance page created.

Fortunately, there has been no major issues with the databases that haven't been solved. This does not always happen but in this case the databases are both live and fully functioning.

## 5.7 Implementation Challenges

This section will summarise the issues that were faced during the implementation stage of the project.

### 5.7.1 Bootstrap Datepickers

Bootstrap combines with jQuery to provide a flexible attractive datepicker that allows users to select a date in a form easily. The date format for these datepickers are set to stay in dd/mm/yyyy and this became an issue as SQL formatted all dates in the form yyyy-mm-dd. This meant that no dates in the database could be manipulated using the standard datepickers that bootstrap provides. The solution was clear that for the forms to be able to submit and data to be entered, the format of the datepickers had to be changed. It is relatively straight forward to change the format using "/" in between months, days and years. However, requiring "-" was much more of an issue.

This problem was solved through reading in the datepicker's input as a string, parsing it and re-formatting the date. This had to be done each time a form was submitted with a date in it.

### 5.7.2 New Employees

After the addition of the create new employee feature, testing was carried out to confirm functionality. It was discovered that the team of the new employee was not being set properly. When a new employee was being created, the team they were being created in was being duplicated in the team table of the database. This proved to be an issue as employees should be created and assigned to just one team. They should not be assigned to a separate team.

This was solved in the employee controller in the create method. When the get method is executing, the database is queried to find the team ID of the manager creating the employee. This same value is then set to the new employee's team ID in the create method, preventing the system from adding a new team ID to the table. This avoided having duplicate entries for the one team.

### **5.7.3 IsManager Flag**

A static method had to be created in the employee controller also, in order to carry out a check on the logged in user and return a boolean value to check if they are a manager in the database. This is used throughout the EPMS web application to check when to display certain information.

# Chapter 6

## Verification & Validation

### 6.1 Testing Strategy

Software testing is the process of running a piece of software to validate or verify its functionality.

The EPMS web application was tested in a number of different ways in order to confirm the correct functionality. Test-driven development took place as much as possible throughout the software development cycle as the web application was created. This involved testing each feature following its addition to the system.

#### 6.1.1 Black Box Testing

During the implementation stage, each feature added to the system was prioritised and developed. Black box testing was carried out each time a feature was added to ensure the features were performing as expected. Black box testing involves testing that the correct output occurs after a specific input is put into the system. The tests were carried out using a number of different browsers over several devices including laptops, iPads and iPhones. The system performed best on Google Chrome but is fully functional in all other web browsers including Safari, Safari (mobile), FireFox and Internet Explorer.

There are almost endless possibilities in terms of the number of devices and platforms that the EPMS could have been tested on. One popular platform that the system was not tested on is Android. There was no Android device available to test on, however if time had permitted, an Android emulator could have been installed to test this. If the project was to be continued and put into a professional capacity then this would be done.

Test cases were produced for each of the Use Cases created for the system - Test cases can be seen in Appendix C and Use Cases can be seen in Appendix B. The process of black box testing was to have Use Cases created each time a feature was added and then create a test case to go with this. Each test case passed or failed (in this case, every test was passed).

#### 6.1.2 Automated Browser Testing

A browser automation tool called Selenium HQ[15] was used in order to carry out some automated system testing to prove that some of the test cases worked. Selenium is a plugin for Mozilla Firefox that is able to record actions carried out in the browser and playback in order to prove that test cases have passed. The system's features can be broken down into six main test cases. Selenium was used to test that the test cases passed for

all main functionality ie Login, Logout, Create Records, Create Work, Submit Records, Grade Records. (The results after the tests were carried out can be seen in Appendix L - Selenium Automated Test Results)

Although it was only used as playback tool for authoring and confirming tests in this instance, Selenium HQ is a very powerful tool that would be used in a larger capacity if the project was to grow to a live environment. It can also provide a testing language that can be used to write the tests. This runs on most operating systems and is open source.

### 6.1.3 User Testing

User tests were carried out amongst other students where they were instructed to carry out tasks on the web application that stretched the functionality of it as much as possible. The behaviour of the application was then analysed and any errors were noted.

## 6.2 Debugging

It was frequently the case, that the application had to be debugged throughout the development and testing stages. Debugging is the process that is carried out when developers come across errors or bugs in their programs. There are many different ways to debug a system. When errors or bugs occurred throughout the testing of the EPMS web application there were two techniques that were carried out to help debug the system:

- Alert or Print statements were used in the JavaScript code of a view to display the runtime values of variables. This was for more of the front end errors.
- Breakpoints were inserted into the code to stop the program execution at certain points to check values of variables etc. Breakpoints can also be used to allow developers to closely inspect the flow and step through each part of the code.

## 6.3 Other Testing

### 6.3.1 Testing on Devices

One of the proposals that emerged from related work interviews was that users of some appraisal systems would like to see EPMS on a range of different devices. They suggested that EPMS should be a system to be used on mobile devices as well as stand alone desktop computers. The project was eventually designed as a responsive web application meaning it can be used on these devices. Testing for this was carried out during the development. At the user evaluation stage, some of the users also used iPads and mobile phones to complete the evaluation and this was completed without any problems. Positive feedback was also given.

### 6.3.2 Testing on Browsers

It is essential to ensure that a system is able to work fully on a number of browsers as it is impossible to know which browser each user will be using. Due to the EPMS being a web application that runs in a browser, this type of testing is vital to ensure project success. Testing was carried out as each feature was added to ensure full functionality across as many browsers as possible.

Browser compatibility includes - Chrome, Safari, Firefox and IE.

The system is able to detect which browser is being used through JavaScript[16] and display an alert to the user to give instructions accordingly such as date formats etc. This is not best practice due to users being able to turn off JavaScript. They will then either bypass browser detection or have no instructions to complete forms with data types or formats. However, for the purpose of this web application the decision was made to use JavaScript for browser detection.

# Chapter 7

## Results & Evaluation

### 7.1 Final Result

The final product is a responsive web application that can be used in a professional environment by an organisation to measure employee performance. The application allows employees to create each piece of completed work inside records on the system and submit each record every three months.

Managers can log in and view each piece of work submitted from their team for the three months. They are required to provide feedback for each record submitted. The system is designed for this to be a continuous process (every three months). A notification system is in place that means when a record is graded, the employee receives an automated email to notify them and confirm the grade.

Managers are also able to access an administration dropdown menu, giving them access to their employee's information and projects that are entered into the system. They also have access to a "Submitted Records" queue which shows all most recent submitted records due. See screenshots below for main functionality of the working system (All other screenshots can be found in Appendix A).

Status	Open
TimePeriodBegin	2016-06-17
TimePeriodEnd	2016-09-17

[Confirm New Record](#)

[Back to List](#)  
© 2016 - EPMS Application

FIGURE 7.1: EPMS - Employee Creates Record

The screenshot shows a web browser window for the EPMS application at the URL [epmsappdemo.azurewebsites.net/Works/Index/60](http://epmsappdemo.azurewebsites.net/Works/Index/60). The page title is "Work". At the top right, there is a user menu with "Employee" and a log-off link. Below the title, there is a search bar and a table header with columns: WorkItem, Description, DateCompleted, Project, Category, and Late. A message "No data available in table" is displayed below the header. At the bottom of the page, there is a blue banner with the text "No Work". Navigation links "Back to Records" and "© 2016 - EPMS Application" are at the bottom.

FIGURE 7.2: EPMS - Employee Empty Record

The screenshot shows a web browser window for the EPMS application at the URL [epmsappdemo.azurewebsites.net/Works/Create/60](http://epmsappdemo.azurewebsites.net/Works/Create/60). The page title is "Create". At the top right, there is a user menu with "Employee" and a log-off link. The main content area contains a form with fields: DateDue (30/06/2016), DateCompleted (30/06/2016), WorkItem (Organisation OS Windows 10), Description/Comments (a text area containing a project summary), Project (New Operating System), and Category (Project Work). A "Create" button is at the bottom of the form. Navigation links "Back to List" and "© 2016 - EPMS Application" are at the bottom.

FIGURE 7.3: EPMS - Employee Creates Work

**Work (From 2016-06-17 To 2016-09-17)**

Create New

Show 10 entries

WorkItem	Description	DateCompleted	Project	Category	Late
Organisation OS Windows 10	This project involved working with John Smith in the operations team rolling out 2 new operating systems to the organisation. The project went very smoothly with no issues and the work was completed on time.	2016-06-30	New Operating System	Project Work	N
Gizmoball group	Group project to produce a user-friendl...	2016-07-15	Other	Evaluation Category	N

Search:

Submit This Record

Back to Records

© 2016 - EPMS Application

FIGURE 7.4: EPMS - Employee Populated Record

**Work (From 2016-06-17 To 2016-09-17)**

Create New

Show 10 entries

WorkItem	Description	DateCompleted	Project	Category	Late
Organisation OS Windows 10	This project involved working with John Smith in the operations team rolling out 2 new operating systems to the organisation. The project went very smoothly with no issues and the work was completed on time.	2016-06-30	New Operating System	Project Work	N
Gizmoball group	Group project to produce a user-friendl...	2016-07-15	Other	Evaluation Category	N

epmsappdemo.azurewebsites.net says:  
Are you sure you want to submit this record?

Cancel OK

Submit This Record

Back to Records

© 2016 - EPMS Application

FIGURE 7.5: EPMS - Employee Submits Record

The screenshot shows a web-based application interface for managing work items. At the top, there's a header with navigation links: EPMS, Home, About, Administration, Team Manager, email (yyb11164@uni.strath.ac.uk), and Log off. Below the header, the title "Work (From 2016-06-17 To 2016-09-17)" is displayed, followed by a subtitle "Kevin Paton". A search bar is present with placeholder text "Search: [ ]". Below the title, there's a table with columns: DateCompleted, WorkItem, Description, Project, Category, Late, and Details. Two rows of data are visible:

DateCompleted	WorkItem	Description	Project	Category	Late	Details
2016-07-15	Gizmoball group	Group project to produce a user-friendly gizmoball game. Other Team working skills were required and this was a successful project as was completed really early.		Evaluation Category N		<a href="#">Details</a>
2016-06-30	Organisation OS Windows 10	This project involved working with John Smith in the operations team rolling out 2 new operating systems to the organisation. The project went very smoothly with no issues and the work was completed on time.	New Operating System	Project Work N		<a href="#">Details</a>

At the bottom of the page, there are links: "Create Performance Page", "Back To Employees", "Back To Submitted Records", and copyright information: "© 2016 - EPMS Application".

FIGURE 7.6: EPMS - Manager Views Record

The screenshot shows a "Create" form for a "Performance" record. The top navigation bar includes EPMS, Home, About, Administration, Team Manager, email (yyb11164@uni.strath.ac.uk), and Log off. The main form has several fields:

- Grading:** A dropdown menu set to "9".
- Status:** A dropdown menu set to "Very Good".
- Justification:** A text area containing the text: "Excellent work on these projects. You carried out exactly what was required and everything was on time."
- Aims:** A text area containing the text: "Next record, I want you to complete all data analysis for the SCOPES system database. Please complete this for 4th October. Also please complete Java training course that I have arranged for 3rd September."

At the bottom of the form is a blue "Create" button.

FIGURE 7.7: EPMS - Manager Grades Record

**Create**  
Project

Name: Java Training

Description: Training course in Fife for developers.

IsActive:

DevTeam:

ServiceDesk:

Operations:

**Create**

[Back to List](#)  
© 2016 - EPMS Application

FIGURE 7.8: EPMS - Manager Creates Projects (From Manage Projects) For Next Record

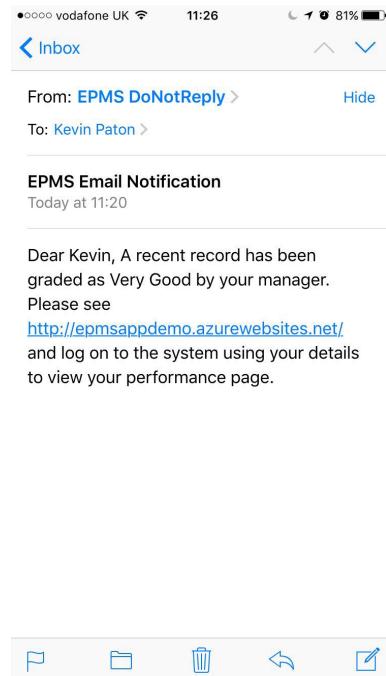


FIGURE 7.9: EPMS - Employee Email Notification

The screenshot shows a web browser displaying the EPMS application at the URL <http://epmsappdemo.azurewebsites.net/Performances/Details/60>. The page title is "Details" under the "Performance" section. The user is logged in as "Employee". The main content area displays the following information:

Name	Kevin Paton
Begin - End	2016-06-17 - 2016-09-17
Status	Very Good
Grading	9
Justification	Excellent work on these projects. You carried out exactly what was required and everything was on time.
Aims	Next record, I want you to complete all data analysis for the SCOPE system database. Please complete this for 4th October. Also please complete Java training course that I have arranged for 3rd September.

At the bottom of the page, there are links for "Back to Records" and "© 2016 - EPMS Application".

FIGURE 7.10: EPMS - Employee Views Grade

## 7.2 Changes & Risks

### 7.2.1 Version Control

Version control for the project caused an issue at the beginning when code was being committed to repositories. Initially, it was decided that SVN would be used, however due to issues with assemblies and references (See section 5.3 - SVN/Git) this was not going to be possible. The project's source code was then pushed and pulled from a GitHub repository. This presented no problems and everything from that point onwards was put on Git.

### 7.2.2 Programming Languages and Technologies

The programming languages and technologies eventually used were seen as a risk when decisions were being made. This was due to lack of previous exposure to the technologies. Throughout the Software Engineering degree, Java is predominately used. Therefore, using C#.NET and Visual Studio was not the most likely choice. This decision was made due to experience gained working with the technologies when on an industrial placement with a third party organisation.

## 7.3 User Evaluation

This section will discuss the user evaluation carried out - how this was done and the results that were produced from it.

### 7.3.1 Plan

User evaluation was carried out in sessions. Due to the EPMS web application being aimed at organisations split up into teams, there were three evaluation sessions with three different teams. Each team were from a different background - development team were software engineering students, service desk were a mixture of students from several courses and operations were made up of people who currently work in a professional environment.

One member of each team would use the system as the team manager with the rest acting as employees. Each manager would also submit work to an overall line manager that would be the head of all three teams.

The sessions were carried out by creating an instruction sheet for each individual member to have them use all functionality of the system and guide them through the feedback process. In the end, an evaluation survey was given to the participants (for all document layout see Appendix H).

### 7.3.2 Likes

Of the fourteen participants, 92% of users gave the UI a minimum of eight out of ten for attractiveness and 86% of users scored it a minimum of eight out of ten for ease of use. The software was not complicated in terms of usability but provided useful features, and users appreciated this. Users initial impressions of the EPMS web application were that the system's simplistic layout was "pleasing and inviting". Furthermore, whilst simplistic, the system was also seen to be "reliable and sophisticated in terms of its processes". Other reasons given for such high scores for UI were that it allowed the system to be as intuitive as possible and this is positive to see in any system. The layout and display of the UI was also complimented to be "professional" and some users indicated they would embrace this particular system in their work place in the future.

Participants were then asked some questions based on the feedback practices carried out by the system, in which 79% of the answers scored at least eight out of ten for how satisfying the process is. In addition to this another two people scored the question ten out of ten suggesting the feedback process could not be any better. Participants commented on the feedback process being "prompt and clear". Most users particularly enjoyed the email notification given to users as part of the process.

Overall, participants found the system very "intuitive" and "easy to use". They also commented on the project looking "professional" and providing a "sophisticated" system for users.

### 7.3.3 Dislikes

Interestingly, a common UI features mentioned that were not favored were the create and back buttons for the datatables. It was mentioned four times in the evaluation results that these buttons should be more prominent and stand out.

Some users commented on the links and particularly the "Log Off" link. This is just plain text currently and could possibly be changed to look more like the other links in the web application.

### 7.3.4 Suggestions/Comments

Security was a feature of the application that was spoken about during evaluation and some people had different views on the subject. It was discussed that managers knowing and distributing people's log in information is a security risk. One suggestion to improve this is to prompt users to reset their password when they first log into the system (See section 8.2).

Another useful suggestion provided by participants was to implement an email notification to be sent to managers when users submit records. This would improve the flow of the system and would not be difficult to implement.

## 7.4 Summary

As mentioned previously, participants found the system very "intuitive" and "easy to use". They also commented on the project looking "professional" and providing a "sophisticated" system for users.

The user evaluation of the EPMS web application has been extremely useful to the project. It allowed for great reflection on the system and produced some excellent suggestions for future features to be added.

## Chapter 8

# Summary & Conclusions

This chapter will discuss the eventual conclusions reached from the EPMS project and then summarise other features that could have been added to the system and may be added in the future.

### 8.1 Summary

Over the course of this project a responsive web application has been designed, implemented, tested and evaluated. Through the use of software engineering technologies and practices the EPMS project has been successfully built. Other features were considered for the system but were deemed unfeasible due to time availability and other reasons.

### 8.2 Possible Future Features

#### 8.2.1 Further Authentication Development

A feature which could have been added into the system is an authentication feature. This would involve users logging into the system via a third party application such as Google, Facebook or Twitter. This is implemented using an assembly called "OAuth" - an open source authentication tool that gives clients (secure) access to third party applications and their servers. This can be very useful for many web applications as it eliminates the need for users to create accounts on applications. Through research it is clear that this kind of authentication would be relatively simple to implement due to its easy integration with Visual Studio and .Net applications. However, this was something that was deemed inappropriate to include as the EPMS is a system that would be implemented in a professional working environment. A suggestion that was given in user evaluation and a feature that would have been implemented given more time is for users to be able to change their own passwords. This would help with any administration issues that the DBAs would have with people forgetting passwords. Furthermore, prompting users to change their passwords the first time they log into the application would increase security within the system.

#### 8.2.2 Further Notification Development

Notifications are currently sent to employees to notify them when a record of their's has been graded. A feature to add in the future could be to implement the same type of notification to be sent to managers when a record is

submitted. This would be straight forward to implement and would have been added had more time been available.

### 8.2.3 Peer Assessment

A future feature of the system is to allow peer assessment within the feedback process. This would involve using peers who work alongside employees and allow them to enter feedback into submitted records as well as managers. This is a feature which could take more time to implement and was also not included due to the time available. This feature, through use of relational databases and programming techniques would certainly be possible, and would improve the EPMS feedback process.

## 8.3 Conclusion

In conclusion, the EPMS project has provided a successful system that could be used in any professional work place to measure, monitor and record employee's performances. Although all aims and objectives of the project were met (and more), the system could easily be developed further to produce an even better feedback process for businesses. Overall, the project also provided an extremely enjoyable learning curve.

# References

- [1] Trakstar, Technology Company, <http://www.trakstar.com> [Accessed 2nd January 2016]
- [2] Responsive web design, info resource, <https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/> [Accessed 23rd February 2016]
- [3] Responsive web design, advantages, <http://www.rapidvaluesolutions.com/whitepapers/responsive-web-design.html> [Accessed 23rd February 2016]
- [4] Software usability, info resource, <http://mashable.com/2014/04/22/website-disability-friendly/#K.nutPY3TZqu> [Accessed 11th March 2016]
- [5] Agile and Waterfall, info resource, <http://www.commonplaces.com/blog/agile-v-waterfall-how-to-approach-your-web-development-project/> [Accessed 13th March 2016]
- [6] Agile, benefits, <http://www.allaboutagile.com/10-good-reasons-to-do-agile-development/> [Accessed 13th March 2016]
- [7] JQuery datatables, webpage, <https://datatables.net/> [Accessed 18th March 2016]
- [8] SMTP email server, webpage, [https://msdn.microsoft.com/en-us/library/system.net.mail.smtpclient\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.net.mail.smtpclient(v=vs.110).aspx) [Accessed 18th March 2016]
- [9] Introduction to LINQ, webpage, <https://msdn.microsoft.com/en-us/library/bb397897.aspx> [Accessed 19th March 2016]
- [10] Entity framework, webpage, <http://www.entityframewortutorial.net/what-is-entityframework.aspx> [Accessed 19th March 2016]
- [11] Bootstrap, webpage, <http://getbootstrap.com/> [Accessed 22nd March 2016]
- [12] GitHub, webpage (repository), <https://github.com/GitSmudj1/EPMS> [Accessed 22nd March 2016]
- [13] MVC, webpage, [http://www.codeproject.com/KB/architecture/MVC\\_MVP\\_MVVM\\_design/MVC\\_MVP\\_MVVM\\_figure4.gif](http://www.codeproject.com/KB/architecture/MVC_MVP_MVVM_design/MVC_MVP_MVVM_figure4.gif) [Accessed 23rd March 2016]

- [14] Authentication, microsoft article, <http://www.asp.net/identity/overview/getting-started/aspnet-identity-using-mysql-storage-with-an-entityframework-mysql-provider> [Accessed 23rd March 2016]
- [15] Selenium HQ, webpage, <http://www.seleniumhq.org/> [Accessed 24th March 2016]
- [16] Browser compatibility, article, <http://mashable.com/2014/02/26/browser-testing-tools/#.KbO15W7TGqE> [Accessed 24th March 2016]
- [17] Azure, webpage, [https://azure.microsoft.com/en-gb/overview/what-is-azure/?WT.srch=1&WT.mc\\_ID=SEM\\_WS1bucqG](https://azure.microsoft.com/en-gb/overview/what-is-azure/?WT.srch=1&WT.mc_ID=SEM_WS1bucqG) [Accessed 24th March 2016]
- [18] Azure, webpage (portal), <https://portal.azure.com/?pageVersion=5.0.302.31545.160328-1527&l=en.en> [Accessed 24th March 2016]
- [19] Azure Tools, Visual Studio Webpage, <https://www.visualstudio.com/en-us/features/azure-tools-vs.aspx> [Accessed 24th March 2016]
- [20] Cloud hosting, Webpage, <http://www.interoute.com/what-cloud-hosting> [Accessed 25th March 2016]
- [21] Moquup, Webpage, <https://moqups.com/> [Accessed 27th March 2016]
- [22] Moscow technique, article, <http://businessanalystlearnings.com/ba-techniques/2013/3/5/moscow-technique-requirements-prioritization> [Accessed 28th March 2016]

## Appendix A

# Appendix A - Working System Screenshots

This section demonstrates the flow of the system/process through use of screenshots.

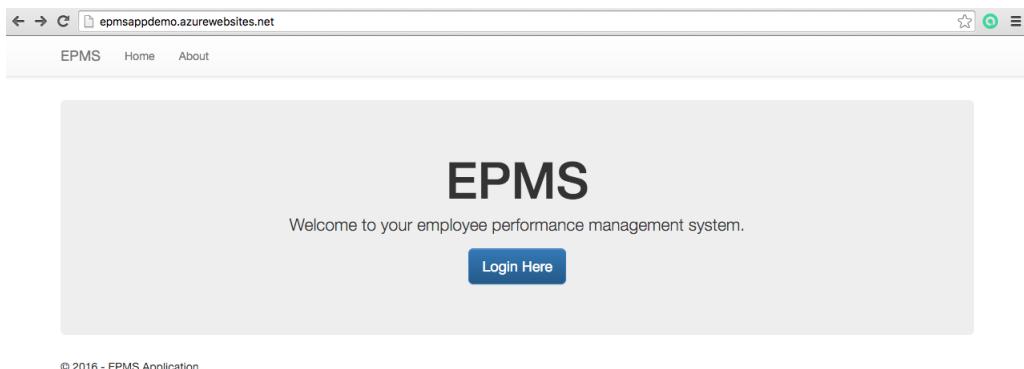


FIGURE A.1: EPMS - Home Page

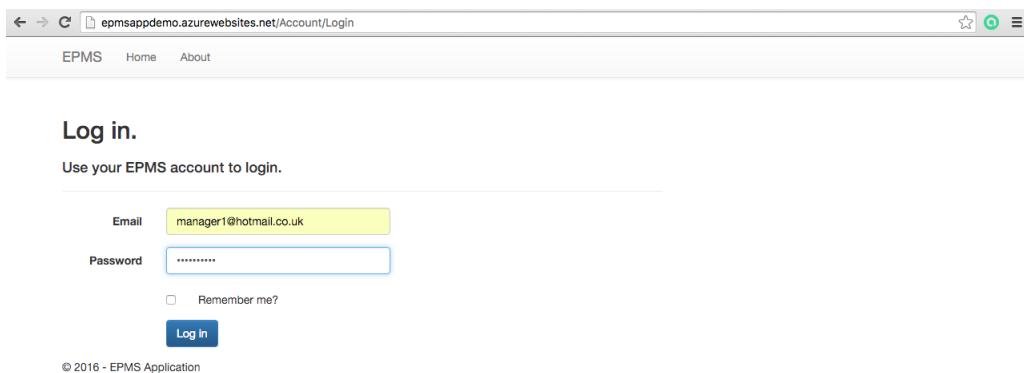


FIGURE A.2: EPMS - Employee Logs In

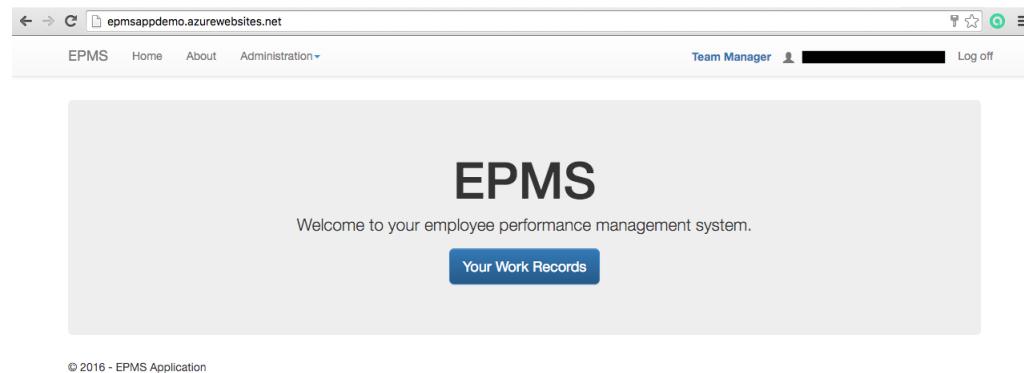


FIGURE A.3: EPMS - Logged In

The screenshot shows a list of employee records. At the top, there is a navigation bar with links for 'EPMS', 'Home', and 'About'. On the right side of the navigation bar, there are icons for 'Employee' and 'Log off'. Below the navigation bar, the main content area has a title 'My Records'. It includes a search bar and a table with columns: TimePeriodBegin, TimePeriodEnd, Status, and Work Created. The table shows one record: TimePeriodBegin is 2016-03-17, TimePeriodEnd is 2016-06-17, Status is Very Good, and Work Created is 2. Below the table, there are links for 'Create Next Record' and 'Back To Home'. At the bottom, a small copyright notice says '© 2016 - EPMS Application'.

FIGURE A.4: EPMS - Employee Records

The screenshot shows a form for creating a new record. At the top, there is a navigation bar with links for 'EPMS', 'Home', and 'About'. On the right side of the navigation bar, there are icons for 'Employee' and 'Log off'. Below the navigation bar, the main content area has a title 'Create' followed by 'Record'. There are four input fields: 'Status' (set to 'Open'), 'TimePeriodBegin' (set to '2016-06-17'), 'TimePeriodEnd' (set to '2016-09-17'), and a 'Confirm New Record' button. At the bottom, there is a link 'Back to List' and a small copyright notice '© 2016 - EPMS Application'.

FIGURE A.5: EPMS - Employee Creates Record

The screenshot shows a web browser window for the EPMS application at the URL [epmsappdemo.azurewebsites.net/Works/Index/60](http://epmsappdemo.azurewebsites.net/Works/Index/60). The page title is "Work". At the top, there is a navigation bar with links for "EPMS", "Home", and "About". On the right, it shows the user "Employee" with a profile icon, a blacked-out email address, and a "Log off" link. Below the navigation is a search bar labeled "Search:" with a placeholder "No data available in table". A table header row includes columns for "WorkItem", "Description", "DateCompleted", "Project", "Category", and "Late". A message "No data available in table" is displayed below the header. At the bottom of the page, there is a blue button labeled "No Work". Navigation links "Previous" and "Next" are also present. The footer contains links "Back to Records" and "© 2016 - EPMS Application".

FIGURE A.6: EPMS - Employee's Empty Record

The screenshot shows a web browser window for the EPMS application at the URL [epmsappdemo.azurewebsites.net/Works/Create/60](http://epmsappdemo.azurewebsites.net/Works/Create/60). The page title is "Create". At the top, there is a navigation bar with links for "EPMS", "Home", and "About". On the right, it shows the user "Employee" with a profile icon, a blacked-out email address, and a "Log off" link. Below the navigation is a form for creating a new work item. It includes fields for "DateDue" (set to 30/06/2016), "DateCompleted" (set to 30/06/2016), "WorkItem" (set to "Organisation OS Windows 10"), "Description/Comments" (containing a detailed project summary), "Project" (set to "New Operating System"), and "Category" (set to "Project Work"). A "Create" button is located at the bottom of the form. The footer contains links "Back to List" and "© 2016 - EPMS Application".

FIGURE A.7: EPMS - Employee Creates Work

The screenshot shows a web browser window for the EPMS application at the URL [epmsappdemo.azurewebsites.net/Works/Index/60](http://epmsappdemo.azurewebsites.net/Works/Index/60). The page title is "Work (From 2016-06-17 To 2016-09-17)". At the top, there is a navigation bar with links for "EPMS", "Home", and "About". On the right, it shows the user "Employee" with a profile icon, a blacked-out email address, and a "Log off" link. Below the navigation is a table showing two completed work items. The first item is "Organisation OS Windows 10" with a description of "This project involved working with John Smith in the operations team rolling out 2 new operating systems to the organisation. The project went very smoothly with no issues and the work was completed on time.", assigned to "New Operating System" under "Project Work". The second item is "Gizmoball group" with a description of "Group project to produce a user-friendl...", assigned to "Other" under "Evaluation Category". Both entries have a status of "N" and "Edit | Delete" links. A "Submit This Record" button is located at the bottom of the table. The footer contains links "Back to Records" and "© 2016 - EPMS Application".

FIGURE A.8: EPMS - Employee's Record Ready To Be Submitted

The screenshot shows a web browser window for the URL [epmsappdemo.azurewebsites.net/Works/Index/60](http://epmsappdemo.azurewebsites.net/Works/Index/60). The page title is "Work (From 2016-06-17 To 2016-09-17)". The main content area displays two work items:

WorkItem	Description	Category	Late
Organisation OS Windows 10	This project involved working with the operations team rolling out new software to the organisation. The project faced some initial issues and the work was completed on time.	Operating System	N
Gizmoball group	Group project to produce a user-friendly...	Project Work	N

A modal dialog box is centered over the table, displaying the message: "epmsappdemo.azurewebsites.net says: Are you sure you want to submit this record?". It contains "Cancel" and "OK" buttons. Below the table, there are "Edit" and "Delete" links for each row. At the bottom of the page, there is a "Submit This Record" button and a link to "Back to Records". The footer includes the copyright notice "© 2016 - EPMS Application".

FIGURE A.9: EPMS - Employee Submits Records To Manager

The screenshot shows a web browser window for the URL [epmsappdemo.azurewebsites.net/Records](http://epmsappdemo.azurewebsites.net/Records). The page title is "My Records". The main content area displays a table of records:

TimePeriodBegin	TimePeriodEnd	Status	Work Created	Action
2016-03-17	2016-06-17	Very Good	2	<a href="#">See Record</a>   <a href="#">Performance Page</a>
2016-06-17	2016-09-17	Submitted	2	<a href="#">See Record</a>

Below the table, there are links for "Create Next Record" and "Back To Home". The footer includes the copyright notice "© 2016 - EPMS Application".

FIGURE A.10: EPMS - Record Now Submitted To Manager

The screenshot shows a web browser window for the URL [epmsappdemo.azurewebsites.net](http://epmsappdemo.azurewebsites.net). The page title is "EPMS". The main content area displays the message: "Welcome to your employee performance management system." and a "Your Work Records" button. On the left side, there is a vertical navigation menu with the following options: "Manage Employees", "Manage Projects", "Submitted Records", and "Register New Account". The top navigation bar includes links for "EPMS", "Home", "About", "Administration", "Team Manager", "Log off", and the user's name "uni.strath.ac.uk". The footer includes the copyright notice "© 2016 - EPMS Application".

FIGURE A.11: EPMS - Manager Access To Admin Menu

The screenshot shows a web browser window for the URL [epmsappdemo.azurewebsites.net/Employees](http://epmsappdemo.azurewebsites.net/Employees). The page title is "Development Employees". The header includes links for EPMS, Home, About, Administration, Team Manager (logged in as [REDACTED]), and Log off. Below the header is a search bar labeled "Search: [REDACTED]". The main content is a table listing employees with columns: FirstName, LastName, Email/UserName, Manager, IsActive, Team, and Records. Each row has "Edit | Details | Records" links. At the bottom left is a copyright notice: "© 2016 - EPMS Application".

FirstName	LastName	Email/UserName	Manager	IsActive	Team	Records
Aidan	Smith	mfb11127@uni.strath.ac.uk	10	<input checked="" type="checkbox"/>	Development	3
Craig	Irving	craig.heron.2013@uni.strath.ac.uk	10	<input checked="" type="checkbox"/>	Development	2
David	Thomson	nfb11170@uni.strath.ac.uk	10	<input checked="" type="checkbox"/>	Development	3
James	Hendrie	james.hendrie.2013@uni.strath.ac.uk	10	<input checked="" type="checkbox"/>	Development	2
Jordan	Wilson	yhb11164@uni.strath.ac.uk	4	<input checked="" type="checkbox"/>	Development	2
Kevin	Paton	kevin.paton.2013@uni.strath.ac.uk	10	<input checked="" type="checkbox"/>	Development	2

© 2016 - EPMS Application

FIGURE A.12: EPMS - Manager Views Team

The screenshot shows a web browser window for the URL [epmsappdemo.azurewebsites.net/Employees/WorkIndex/60](http://epmsappdemo.azurewebsites.net/Employees/WorkIndex/60). The page title is "Work (From 2016-06-17 To 2016-09-17)". The header includes links for EPMS, Home, About, Administration, Team Manager (logged in as yhb11164@uni.strath.ac.uk), and Log off. Below the header is a search bar labeled "Search: [REDACTED]". The main content is a table listing work items with columns: DateCompleted, WorkItem, Description, Project, Category, and Late. Each row has "Details" links. At the bottom left is a copyright notice: "© 2016 - EPMS Application".

DateCompleted	WorkItem	Description	Project	Category	Late
2016-07-15	Gizmoball group	Group project to produce a user-friendly gizmoball game. Other Team working skills were required and this was a successful project as was completed really early.	Evaluation	Category N	<a href="#">Details</a>
2016-06-30	Organisation OS Windows 10	This project involved working with John Smith in the operations team rolling out 2 new operating systems to the organisation. The project went very smoothly with no issues and the work was completed on time.	New Operating System	Project Work	N <a href="#">Details</a>

[Create Performance Page](#)

Back To Employees  
Back To Submitted Records  
© 2016 - EPMS Application

FIGURE A.13: EPMS - Manager Views Employee's Work

The screenshot shows a web browser window for the URL [epmsappdemo.azurewebsites.net/AdminRecords](http://epmsappdemo.azurewebsites.net/AdminRecords). The page title is "Submitted Records". The header includes links for EPMS, Home, About, Administration, Team Manager (logged in as [REDACTED]), and Log off. Below the header is a search bar labeled "Search: [REDACTED]". The main content is a table listing submitted records with columns: Name, TeamName, TimePeriodBegin, TimePeriodEnd, Status, and View Record. Each row has "View Record" links. At the bottom left is a copyright notice: "© 2016 - EPMS Application".

Name	TeamName	TimePeriodBegin	TimePeriodEnd	Status	View Record
Kevin Paton	Development	2016-06-17	2016-09-17	Submitted	<a href="#">View Record</a>
Craig Irving	Development	2016-06-17	2016-09-17	Submitted	<a href="#">View Record</a>

© 2016 - EPMS Application

FIGURE A.14: EPMS - Manager Can Access Submitted Records Queue

The screenshot shows a web browser window for the EPMS application at the URL <http://epmsappdemo.azurewebsites.net/Performances/Create/60>. The page title is 'Create' under the 'Performance' section. The user is logged in as 'Team Manager' with the email 'yyb11164@uni.strath.ac.uk'. The form fields include:

- Grading:** A dropdown menu set to '9'.
- Status:** A dropdown menu set to 'Very Good'.
- Justification:** A text area containing the text: "Excellent work on these projects. You carried out exactly what was required and everything was on time."
- Aims:** A text area containing the text: "Next record, I want you to complete all data analysis for the SCOPEx system database. Please complete this for 4th October. Also please complete Java training course that I have arranged for 3rd September."

At the bottom right of the form is a blue 'Create' button.

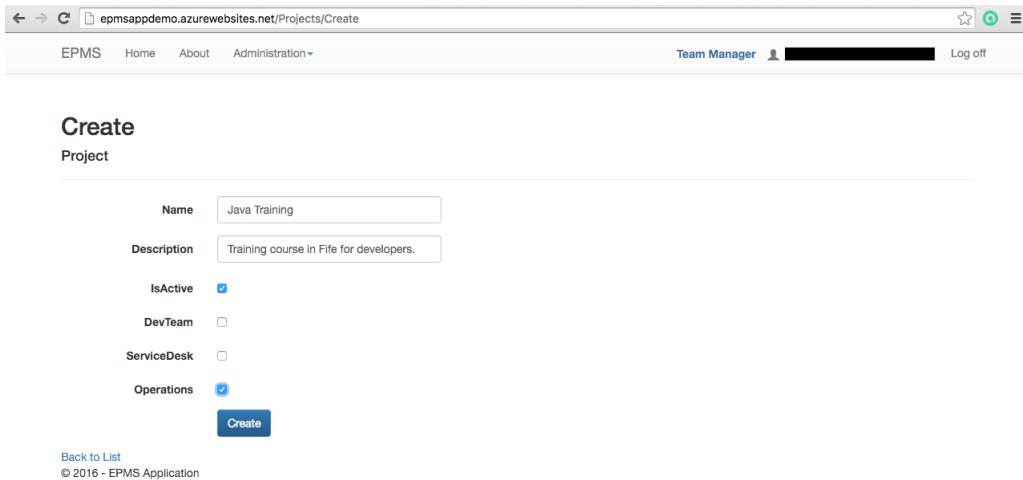
FIGURE A.15: EPMS - Record Is Graded

The screenshot shows a web browser window for the EPMS application at the URL <http://epmsappdemo.azurewebsites.net/Projects>. The page title is 'Projects'. The user is logged in as 'Team Manager' with a blurred email address. The table displays a list of projects:

Name	Description	IsActive	DevTeam	ServiceDesk	Operations	Action
Balsamiq Training	Balsamiq Training	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
Business Services Mobile App	Fix the initial bugs on the mobile app.	True	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
Dev Team Website	New team website for collaborating ideas	True	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Edit
Evaluation Project	Created for evaluation purposes	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
General company applications		True	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
HR System	New HR system to be rolled out	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
New banking system	Design and implementation of brand new banking system	True	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Edit
New Operating System	Rolling out Windows 10 to the organisation	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
Other	Other projects	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit
Scope Training	Scope Training	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit

At the bottom right of the table are navigation buttons: 'Previous' and 'Next'.

FIGURE A.16: EPMS - Manager Can Manage Projects In System



The screenshot shows the 'Create Project' form. The 'Name' field contains 'Java Training'. The 'Description' field contains 'Training course in File for developers.'. The 'IsActive' checkbox is checked. The 'DevTeam' and 'ServiceDesk' checkboxes are unchecked. The 'Operations' checkbox is checked. A 'Create' button is visible at the bottom.

FIGURE A.17: EPMS - Manager Adds New Project To System

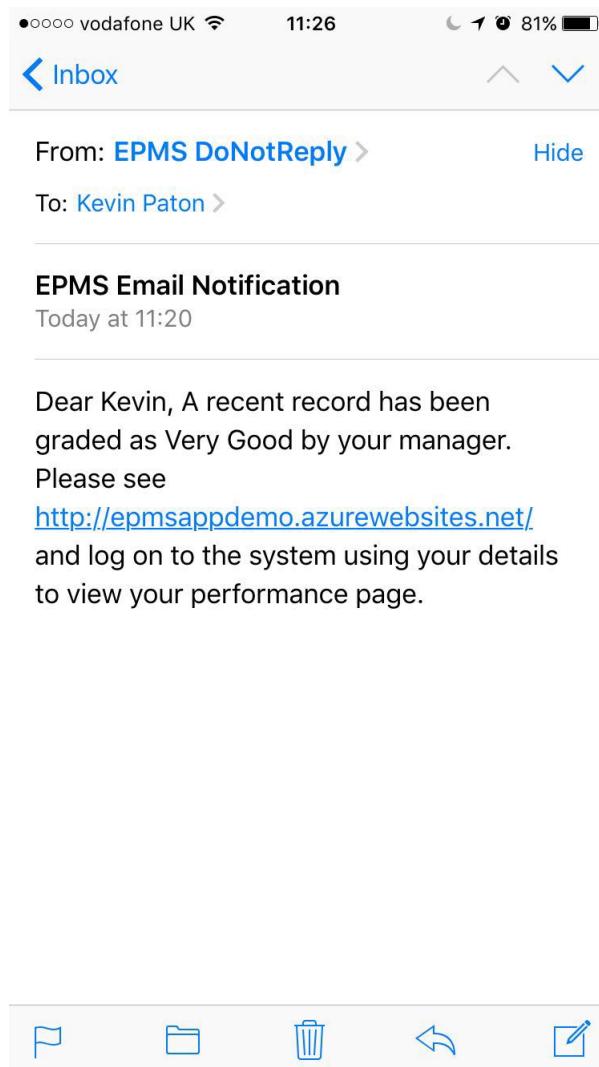


FIGURE A.18: EPMS - Email Sent After Grading

The screenshot shows a web browser displaying the EPMS application at the URL [epmsappdemo.azurewebsites.net/Works/Index/60](http://epmsappdemo.azurewebsites.net/Works/Index/60). The page title is "Work (From 2016-06-17 To 2016-09-17)". A message at the top states "This record has been submitted to admin.". Below this is a table with columns: WorkItem, Description, DateCompleted, Project, Category, and Late. Two entries are listed:

WorkItem	Description	DateCompleted	Project	Category	Late
Organisation OS Windows 10	This project involved working with John...	2016-06-30	New Operating System	Project Work	N
Gizmoball group	Group project to produce a user-friendl...	2016-07-15	Other	Evaluation Category	N

At the bottom of the table, there are links for "Previous" and "Next". A green banner at the bottom right says "Performance Page Created: View Here!". Navigation links at the bottom include "Back to Records" and "© 2016 - EPMS Application".

FIGURE A.19: EPMS - Record Is Now Graded

The screenshot shows a web browser displaying the EPMS application at the URL [epmsappdemo.azurewebsites.net/Performances/Details/60](http://epmsappdemo.azurewebsites.net/Performances/Details/60). The page title is "Details". A section titled "Performance" contains the following information:

Name	Kevin Paton
Begin - End	2016-06-17 - 2016-09-17
Status	Very Good
Grading	9
Justification	Excellent work on these projects. You carried out exactly what was required and everything was on time.
Aims	Next record, I want you to complete all data analysis for the SCOPE system database. Please complete this for 4th October. Also please complete Java training course that I have arranged for 3rd September.

Navigation links at the bottom include "Back to Records" and "© 2016 - EPMS Application".

FIGURE A.20: EPMS - Performance Page Given To User

## Appendix B

# Appendix B - Use Case Descriptions

This section holds all use case descriptions.

### Employee Login:

**ID:** EMUC - 1

**Title:** Login

**Description:** Employee/Manager logs into the system using username and password.

**Primary Actor:** Employee/Manager.

**Preconditions:** Employee/Manager opens the system.

**Postconditions:** Employee/Manager is logged into the system and is free to start using.

**Main** 1. Employee/Manager opens up the system via the webpage.

**Success Scenario:** 2. Employee/Manager enters username and password.

3. System logs employee/manager in.

**Extensions:** Employee/Manager's login details are invalid as they are not in the database.

**Frequency of Use:** Every time the system is used.

**Status:** Design Stage.

FIGURE B.1: Login Use-Case-1

### Employee Logout:

**ID:** EMUC - 2

**Title:** Logout

**Description:** Employee/Manager logs out of the system.

**Primary Actor:** Employee/Manager

**Preconditions:** Employee/Manager closes the system.

**Postconditions:** Employee/Manager is logged out of the system.

**Main** 1. Employee/Manager clicks the 'logout' button.

**Success Scenario:** 2. System logs Employee/Manager out.

**Extensions:** Employee/Manager does not click the button and is not logged out.

**Frequency of Use:** Every time the system is used.

**Status:** Design Stage.

FIGURE B.2: Logout Use-Case-2

**Manage Records:**

<b>ID:</b>	EUC-3
<b>Title:</b>	Manage Records
<b>Description:</b>	The employee is able to see all their current work records and manage them. There are four records each year (every three months).
<b>Primary Actor:</b>	Employee
<b>Preconditions:</b>	The employee has logged in and is at the homepage. The option to view/manage records is in the centre of the screen.
<b>Postconditions:</b>	The manage records screen is displayed. The employee can see this with a table in the centre of the screen with all created records.
<b>Main Success Scenario:</b>	<p>1. The employee logs onto the system.</p> <p>2. Clicks 'Your Work Records'.</p> <p>3. The 'Manage Records' view is shown with a table of employee's records.</p>
<b>Extensions:</b>	If the employee does not have any records, then an empty table is shown. Or if the employee does not exist in the 'Employee' database, then an error will be thrown.
<b>Frequency of Use:</b>	Each time the employees want to access their records.
<b>Status:</b>	Design Stage.

FIGURE B.3: Manage Records Use-Case-3

**Create Record:**

<b>ID:</b>	EUC-4
<b>Title:</b>	Create Record
<b>Description:</b>	Employee has option to view, create or delete records. When selected, create new record creates a new record for next three months.
<b>Primary Actor:</b>	Employee
<b>Preconditions:</b>	The employee has selected to manage records from the home screen and clicked 'Create Next Record' and confirmed this.
<b>Postconditions:</b>	A new (empty) record for the next three months is added to the table.
<b>Main Success Scenario:</b>	<p>1. The employee logs onto the system.</p> <p>2. Clicks 'Your Work Records'.</p> <p>3. Clicks 'Create Next Record'.</p> <p>4. The new record is added to the table.</p>
<b>Extensions:</b>	N/A
<b>Frequency of Use:</b>	Each time the employee wants to create a new record.
<b>Status:</b>	Design Stage.

FIGURE B.4: Create Record Use-Case-4

**Delete Record:**

<b>ID:</b>	AUC-27
<b>Title:</b>	Delete Record
<b>Description:</b>	The administrators can see all created records. They also have the option to view or delete existing records. When selected, the delete function removes the record and all pieces of work associated with it.
<b>Primary Actor:</b>	Admin
<b>Preconditions:</b>	The employee has informed the manager of some sort of mistake and requested to have a record removed. The manager has brought up employee's records in order to do this.
<b>Postconditions:</b>	The delete link is selected.
<b>Main Success Scenario:</b>	<p>1. Administrator logs onto the system.</p> <p>2. Clicks 'Administration' dropdown.</p> <p>3. Clicks 'Manage Employees'.</p> <p>4. Clicks 'View Records'.</p> <p>5. Clicks 'Delete'.</p> <p>6. Record is removed with all other work associated.</p>
<b>Extensions:</b>	The employee does not have any records to remove, an empty table is displayed. The manager refuses to delete the record, nothing is removed from the system.
<b>Frequency of Use:</b>	Each time the employee (or manager) wants to delete a new record.
<b>Status:</b>	Design Stage.

FIGURE B.5: Delete Record Use-Case-5

**Create Work:****ID:** EUC-6**Title:** Create Work**Description:** When the user opens a record, they will create work within this record. This will be done using the 'Create Work' link.**Primary Actor:** Employee**Preconditions:** The employee has logged in successfully and opened a record for the current three-month period. They have clicked on the link to create a new piece of work within this record.**Postconditions:** A new piece of work is created and posted within the table for the user to see.**Main** 1. Employee has logged in.**Success Scenario:** 2. Created a new record or opened existing record.  
3. They have clicked on the link to create a new piece of work within the record.  
4. They put in the details of the piece of work and confirm them.  
5. The table of work is now shown with the newly created piece of work included in it.**Extensions:** The employee does not enter the correct datatype (or not enough information) into the details of the work. An error message is shown to request the correct information. In a good record, the pieces of work will be identical to what was entered in the performance page by the manager at the previous record.**Frequency of Use:** Each time the employee wants to create a new piece of work.**Status:** Design Stage.

FIGURE B.6: Create Work Use-Case-6

**Edit Work:****ID:** EUC-7**Title:** Edit Work**Description:** After the employee successfully creates a piece of work, it will be added into the table that holds all work within the current record. They will then be given the option to 'edit' which will allow them to change some aspects of the work.**Primary Actor:** Employee**Preconditions:** The employee has logged in successfully and opened a record for the current three-month period. They then select the current record and wish to edit a piece of work they have created. They have clicked on the link to edit the piece of work and will then have to edit the fields accordingly.**Postconditions:** The work is changed to show the changes made by the employee and the database is updated.**Main** 1. Employee has logged onto the system.**Success Scenario:** 2. Entered the current record.  
3. They have clicked on the link to edit the piece of work from the table.  
4. They change the details that they want to change and click save.  
5. The table of work is now shown with the updated piece of work included in it.**Extensions:** The employee does not enter the correct datatype (or not enough information) into the details of the work. An error message is shown to request the correct information. Validation error**Frequency of Use:** Each time the employee wants to edit a piece of work.**Status:** Design Stage.

FIGURE B.7: Edit Work Use-Case-7

**Delete Work:**

<b>ID:</b>	EUC-8
<b>Title:</b>	Delete Work
<b>Description:</b>	After the employee successfully creates a piece of work, it will be added into the table that holds all work within the current record. They will then be given the option to delete the work which will allow them to remove any of the work created and this will delete from the database.
<b>Primary Actor:</b>	Employee
<b>Preconditions:</b>	The employee has logged in successfully and selected the record for the current three-month period. They have clicked on the link to delete the piece of work they want to remove.
<b>Postconditions:</b>	The work is removed by the employee and the database is updated.
<b>Main Success Scenario:</b>	<p>1. Employee has logged onto the system.</p> <p>2. Entered the current record.</p> <p>3. They have clicked on the link to delete the piece of work from the table.</p> <p>4. The table of work is now shown with the piece of work not included in it.</p>
<b>Extensions:</b>	N/A
<b>Frequency of Use:</b>	Each time the employee (or manager) wants to delete a piece of work.
<b>Status:</b>	Design Stage.

FIGURE B.8: Delete Work Use-Case-8

**Save Work:**

<b>ID:</b>	EUC-9
<b>Title:</b>	Save Work
<b>Description:</b>	The employee will save a piece of work after carrying out one of the following tasks: 1) Creating a new piece of work 2) Editing a piece of work.
<b>Primary Actor:</b>	Employee
<b>Preconditions:</b>	The employee has logged in successfully and opened a record for the current three-month period. They then select the current record and wish to save a piece of work they have created or edited. They have clicked on the link to create or edit the piece of work and then save.
<b>Postconditions:</b>	The work is saved and the database is updated.
<b>Main Success Scenario:</b>	<p>1. Employee has logged in.</p> <p>2. They have created a new piece of work and saved the piece of work.</p> <p>3. Or they have clicked on the link to edit an existing piece and saved that.</p> <p>4. The table of work is now shown with the piece of work included in it.</p>
<b>Extensions:</b>	If the employee hasn't completed all of the required fields then they will be given a validation error message.
<b>Frequency of Use:</b>	Each time the employee wants to save a piece of work.
<b>Status:</b>	Design Stage.

FIGURE B.9: Save Work Use-Case-9

**Comment on Work:**

<b>ID:</b>	EUC-10
<b>Title:</b>	Comment on Work
<b>Description:</b>	The employee will always need to comment on a piece of work that they create. This will be so that when their manager is completing the performance/appraisal page, they can see the employee's views on the piece of work. They will comment on it when they create it.
<b>Primary Actor:</b>	Employee
<b>Preconditions:</b>	The employee has logged in successfully and opened a record for the current three-month period. They have clicked on the link to create a new piece of work within this record. This is where they will be given the opportunity to comment.
<b>Postconditions:</b>	A new piece of work is created and posted within the table for the user to see.
<b>Main Success Scenario:</b>	<p>1. Employee has logged in.</p> <p>2. They have clicked on the link to create a new piece of work within this record.</p> <p>3. They put in the comments and details of the piece of work and confirm them.</p> <p>4. The table of work is now shown with the newly created piece of work included in it.</p>
<b>Extensions:</b>	The employee does not enter the correct datatype (or not enough information) into the details of the work. An error message is shown to request the correct information.
<b>Frequency of Use:</b>	Each time the employee (or manager) wants to comment on a new piece of work.
<b>Status:</b>	Design Stage.

FIGURE B.10: Comment On Work Use-Case-10

**Submit Record:**

<b>ID:</b>	EUC-11
<b>Title:</b>	Submit Record
<b>Description:</b>	When it comes towards the end of the three-month working period, the employee will be required to submit their record for this period. This will be done via a 'Submit Record' button within the record view. The status of the record will then be changed from 'Open' to 'Submitted' and the record will be ready for the employee's manager to review and update the performance page.
<b>Primary Actor:</b>	Employee
<b>Preconditions:</b>	The employee has added all pieces of work to the record and is ready to submit the record to line manager. They click on the submit button which will add it to the latest record's list in the manager's view.
<b>Postconditions:</b>	The record's status is updated to show as 'Submitted' and it is now ready to be reviewed.
<b>Main Success Scenario:</b>	<p><b>Main</b></p> <ol style="list-style-type: none"> <li>Employee has logged in.</li> <li>They have the submit button enabled within the current record.</li> <li>They have clicked on the button to submit the record.</li> <li>This will change the status of the record and the manager will now go in and review it.</li> <li>Manager will see everything that was completed within the three months and update the performance page.</li> </ol>
<b>Extensions:</b>	The employee can only submit records within five days of the next record beginning.(Button will be disabled).
<b>Frequency of Use:</b>	Each time the employee has to submit record.
<b>Status:</b>	Design Stage.

FIGURE B.11: Submit Record Use-Case-11

**View Performance Page:**

<b>ID:</b>	EUC-12
<b>Title:</b>	View Performance Page
<b>Description:</b>	Each record submitted by an employee will be accompanied by a performance page that will be completed by their line manager. This will be completed at the end of every record and this will give comments, feedback and a grading (out of 10) according to how well the employee has performed (satisfactory or unsatisfactory). The employee's will also be given tasks/projects to work on for the next record so will have access to this page at all times.
<b>Primary Actor:</b>	Employee
<b>Preconditions:</b>	The employee has submitted record. The line manager has picked up the record and graded it.
<b>Postconditions:</b>	The manager completes the performance page, the link to this completed information is then made available for the employee to refer to.
<b>Main Success Scenario:</b>	<p><b>Main</b></p> <ol style="list-style-type: none"> <li>Employee has logged in.</li> <li>They are viewing records and click to view performance page for a selected record.</li> <li>Or the employee is in the record and the links to each performance page are visible.</li> <li>The employee clicks on the link and can see the information in the performance page.</li> </ol>
<b>Extensions:</b>	Performance page might not yet be available if the manager hasn't completed it.
<b>Frequency of Use:</b>	Each time the employees want to see their performance pages. They can only view performance pages when the line manager has completed them.
<b>Status:</b>	Design Stage.

FIGURE B.12: View Performance Page Use-Case-12

**Manage Employee:**

<b>ID:</b>	AUC-13
<b>Title:</b>	Manage Employee
<b>Description:</b>	Administrators of the system (line managers) have the ability to manage their employees. They will be given the chance to open up the 'Manage Employees' screen. This will be a table of all the employees in the team and their usernames. This is where they will be able to add and edit employees, as well as access all of the employee's records, work and performance pages. They will be able to update their performance rating this way also.
<b>Primary Actor:</b>	Administrator
<b>Preconditions:</b>	The manager has logged in and clicked on the admin menu. Clicked 'Manage Employees'.
<b>Postconditions:</b>	The manager is displayed with all of the employee's in their team and the option to Create New, Edit, IsActive, View Records – for each employee.
<b>Main Success Scenario:</b>	<p><b>Main</b></p> <ol style="list-style-type: none"> <li>Manager has logged in.</li> <li>The manager will have the 'Admin' menu available to them in the bar across the top of the screen.</li> <li>In this menu, they will click 'Manage Employees'</li> <li>This will show a screen of all employees in the team and give the different options stated above.</li> </ol>
<b>Extensions:</b>	Managers will only have the 'Administration' menu available to them if they are a team manager and have admin permissions within the system.
<b>Frequency of Use:</b>	Each time a manager wants to manage their employees or access individual records.
<b>Status:</b>	Design Stage.

FIGURE B.13: Manage Employees Use-Case-13

**Create Employee:**

<b>ID:</b>	AUC-14
<b>Title:</b>	Create Employee
<b>Description:</b>	Whilst managers are in the 'Manage Employees' view, they will be able to add and edit employees, as well as access all of the employee's records, work and performance pages.
<b>Primary Actor:</b>	Administrator
<b>Preconditions:</b>	The manager has logged in and clicked on the administration menu. Clicked 'Manage Employees' and then 'Create New'. The manager is given a screen that allows them to enter details of new employee such as Name, UserName and IsActive.
<b>Postconditions:</b>	The user creates a new employee and the 'Manage Employee' screen is again shown including the new employee.
<b>Main Success Scenario:</b>	<p><b>Main</b></p> <ol style="list-style-type: none"> <li>Manager is logged in.</li> <li>Enters the 'Manage Employees' view.</li> <li>Clicks 'Create New' and fills each of the fields with the new employee's details.</li> <li>Clicks 'Create'.</li> <li>This adds the employee to the database and shows the updated team in the table.</li> </ol>
<b>Extensions:</b>	The manager does not complete all fields or use the wrong data-type and they will be thrown a validation error.
<b>Frequency of Use:</b>	Each time a manager wants to add a new employee.
<b>Status:</b>	Design Stage.

FIGURE B.14: Create Employee Use-Case-14

**Edit Employee:****ID:** AUC-15**Title:** Edit Employee**Description:** Whilst managers are in the 'Manage Employees' view, they will be able to add and edit employees, as well as access all of the employee's records, work and performance pages.**Primary Actor:** Administrator**Preconditions:** The manager has logged in and clicked on the admin menu. Clicked 'Manage Employees' and then 'Edit' for the selected employee. The manager is given a screen that allows them to update details of new employee such as Name, UserName and IsActive.**Postconditions:** The manager updates employee information and the 'Manage Employee' screen is again shown including the updated employee.**Main**  
**Success Scenario:** 1. Manager is logged in.

2. Enters the 'Manage Employees' view.

3. Clicks 'Edit', fills each of the fields with the new employee's details and clicks 'Save'.

4. This updates in the database and shows on the view.

**Extensions:** If the manager does not complete all fields or use the wrong data-type and they will be thrown a validation error.**Frequency of Use:** Each time a manager wants to edit any employee information.**Status:** Design Stage.

FIGURE B.15: Edit Employee Use-Case-15

**View Records:****ID:** AUC-16**Title:** View Employee Records**Description:** Whilst managers are in the 'Manage Employees' view, they will be able to add and edit employees, as well as access all of the employee's records, work and performance pages.**Primary Actor:** Administrator**Preconditions:** The manager has logged in and clicked on the admin menu. Clicked 'Manage Employees' and then 'View Records' for the selected employee. The manager is given a view with a table containing each of the employee's submitted records.**Postconditions:** The manager is presented with a view with a table containing all of the pieces of work carried out by the employee within the time period of the record.**Main**  
**Success Scenario:** 1. Manager is logged in.

2. Enters the 'Manage Employees' view.

3. The manager clicks 'View Records' (from the 'Manage Employees' screen).

4. Manager is then able to inspect each of the pieces of work submitted.

**Extensions:** N/A**Frequency of Use:** Each time a manager wants to view any employee records submitted.**Status:** Design Stage.

FIGURE B.16: View Records Use-Case-16

**Edit Record:****ID:** AUC-17**Title:** Edit Employee Record**Description:** When the manager accesses the employee's records they will have the ability to edit a record if there have been any mistakes when creating them.**Primary Actor:** Administrator**Preconditions:** The manager has logged in and clicked on the admin menu. Clicked 'Manage Employees' and then 'View Records' for the selected employee. The manager is given a view with a table containing each of the employee's submitted records. Then, manager clicks 'Edit' for selected record.**Postconditions:** The manager is presented with an 'Edit Record' view that allows him/her to amend the record and once they click save the record is changed.**Main**

- Success Scenario:**
1. Manager is logged in.
  2. Enters the 'Manage Employees' view.
  3. Views the selected employees record.
  4. Clicks 'Edit' and then enters the information that he wants to be amended.
  5. After clicking save, the record is amended and the table of records re-appears.

**Extensions:** Managers will only be able to access these pages if they are admin users (team leaders) in the system.**Frequency of Use:** Each time a manager wants to edit any employee records submitted.**Status:** Design Stage.

FIGURE B.17: Edit Record Use-Case-17

**View Work:****ID:** AUC-18**Title:** View Work**Description:** When the manager accesses the employee's records they will have the ability to open a record and this will allow them to view the work within it.**Primary Actor:** Administrator**Preconditions:** The manager has logged in and clicked on the administration menu. Clicked 'Manage Employees' and then 'View Records' for the selected employee. The manager is given a view with a table containing each of the employee's submitted records. Then, the manager clicks 'View Work' for selected record and sees a table with all the work that was submitted to the record.**Postconditions:** The manager is clicks 'View Work' and sees all the work submitted within that record.**Main**

- Success Scenario:**
1. Manager is logged in.
  2. Enters the 'Manage Employees' view.
  3. Clicks 'View Records' for selected employee and then 'View Work' for selected record.
  4. The manager can then see all the work that has been submitted for that record.

**Extensions:** The link will only be available to admin users and if pieces of work exist.**Frequency of Use:** Each time a manager wants to view the work that has been submitted by the employee.**Status:** Design Stage.

FIGURE B.18: View Work Use-Case-18

**Work Detail:****ID:** AUC-19**Title:** Work Detail**Description:** When the manager enters the 'View Work' screen for the selected record they will have the option to view each piece of work in more detail. They will use this when grading and updating performance pages.**Primary Actor:** Administrator**Preconditions:** The manager has logged in and clicked on the admin menu. Clicked 'Manage Employees' and then 'View Records' for the selected employee. The manager is given a view with a table containing each of the employee's submitted records. Then, manager clicks 'View Work' for selected record and sees a table with all the work that was submitted to the record. Then clicks 'Details' for the work details of the selected piece of work.**Postconditions:** The manager is clicks 'Details' and the selected piece of work fully..**Main Success Scenario:** 1. Manager is logged in.

2. Enters the 'Manage Employees' view.

3. Clicks 'View Records' for selected employee and then 'View Work' for selected record.

4. The manager can then see all the work that has been submitted for that record

5. Clicks 'Details' for the selected piece of work they want to see.

**Extensions:** The link will only be available to admin users and if pieces of work exist.**Frequency of Use:** Each time a manager wants to view the work (fully) that has been submitted by the employee.**Status:** Design Stage.

FIGURE B.19: Work Detail Use-Case-19

**Update Performance Page:****ID:** AUC-20**Title:** Update Performance Page**Description:** Managers will be expected to update a performance page at the end of every three-month period. This will be based on the record submitted by employees for the three-month period and will contain a grading, aims and objectives for the coming three months.**Primary Actor:** Administrator**Preconditions:** The manager has logged in and clicked on the admin menu. Clicked 'Manage Employees' and then 'View Records' for the selected employee. The manager is given a view with a table containing each of the employee's submitted records. Then, manager clicks 'View Work' for selected record and sees a table with all the work that was submitted to the record. Then clicks 'Update Performance Page' for this record.**Postconditions:** The manager completes the performance page for this record and grades it. Then the user will be able to view the grading.**Main Success Scenario:** 1. Manager is logged in.

2. Enters the 'Manage Employees' view.

3. Clicks 'View Records' for selected employee and then 'View Work' for selected record.

4. The manager can then see all of the work that has been submitted for that record and clicks 'Update Performance Page', which will then allow the manager to complete the performance page.

**Extensions:** The link will only be available to admin users and if the record has been submitted by the user.**Frequency of Use:** Each time a manager wants to update the performance page for the user**Status:** Design Stage.

FIGURE B.20: Update Performance Page Use-Case-20

**Latest Record Screen:**

<b>ID:</b>	AUC-21
<b>Title:</b>	Latest Record Screen
<b>Description:</b>	The latest record screen will allow managers a quick look at all outstanding records that have been submitted. Starting with the ones which have been submitted for the longest. They can then from this screen, go back to the 'View Record' screen for these records and manage them from there.
<b>Primary Actor:</b>	Administrator
<b>Preconditions:</b>	The manager has logged in and clicked on the admin menu. Then clicked 'Latest Records'.
<b>Postconditions:</b>	The manager can then view a table with all of the team's outstanding records that haven't been dealt with.
<b>Main Success Scenario:</b>	<p><b>Main</b></p> <ol style="list-style-type: none"> <li>Manager is logged in.</li> <li>Clicks 'Administration' dropdown.</li> <li>The manager enters the 'Latest Record' screen allowing him/her to quickly see which records have to be dealt with and can manage them accordingly.</li> </ol>
<b>Extensions:</b>	The link will only be available to admin users if they have the correct permissions in the system.
<b>Frequency of Use:</b>	Each time a manager wants to view all outstanding records from their team members.
<b>Status:</b>	Design Stage.

FIGURE B.21: Latest Screen Page Use-Case-21

**Register New User:**

<b>ID:</b>	AUC-22
<b>Title:</b>	Register New User
<b>Description:</b>	Administrator users will have the option to create new users for the system. This is going to be through a 'Register' link where they will be asked to enter the email and password for the employee and they will use this to log in.
<b>Primary Actor:</b>	Administrator
<b>Preconditions:</b>	The manager has logged in and clicked on the 'Register' link. The manager enters the details and creates the user.
<b>Postconditions:</b>	A new user is created in the database and can now log onto the system using the details entered by the manager.
<b>Main Success Scenario:</b>	<p><b>Main</b></p> <ol style="list-style-type: none"> <li>The manager logs in.</li> <li>Clicks on the 'Register' link.</li> <li>Enters the details for the user and clicks 'Register'.</li> <li>A new user is created.</li> </ol>
<b>Extensions:</b>	The link will only be available to admin users if they have the correct permissions in the system.
<b>Frequency of Use:</b>	Each time a manager wants to add a new user to the system.
<b>Status:</b>	Design Stage.

FIGURE B.22: Register Use-Case-22

**Manage Projects:**

<b>ID:</b>	AUC-23
<b>Title:</b>	Manage Projects
<b>Description:</b>	Administrator users will have the option to manage what projects are entered into the system. This is going to be through a 'Manage Projects' link in the 'Administration' dropdown menu.
<b>Primary Actor:</b>	Administrator
<b>Preconditions:</b>	The manager has logged in and clicked on the 'Administration' dropdown. The manager then clicks on the 'Manage Projects' link.
<b>Postconditions:</b>	The manager can see a screen with a table of all the projects in the system.
<b>Main Success Scenario:</b>	<p><b>Main</b></p> <ol style="list-style-type: none"> <li>The manager logs in.</li> <li>Clicks on the 'Administration' dropdown.</li> <li>Clicks on the 'Manage Projects' link.</li> <li>'Manage Projects' view is displayed.</li> </ol>
<b>Extensions:</b>	The link will only be available to admin users if they have the correct permissions in the system.
<b>Frequency of Use:</b>	Each time a manager wants to see what projects are currently in the system or 'Add', 'Edit' or 'Delete' any projects.
<b>Status:</b>	Design Stage.

FIGURE B.23: Manage Projects Use-Case-23

**Add Project:**

<b>ID:</b>	AUC-24
<b>Title:</b>	Add Project
<b>Description:</b>	In the 'Manage Projects' view, managers will be able to add new projects to the system.
<b>Primary Actor:</b>	Administrator
<b>Preconditions:</b>	The manager has logged in and clicked on the 'Administration' dropdown. The manager then clicks on the 'Manage Projects' link and clicks 'Create New'. The manager enters the details of the new project.
<b>Postconditions:</b>	The project is added to the system and will appear in all of the project dropdown menus throughout the system.
<b>Main Success Scenario:</b>	<p>1. The manager logs in.      2. Clicks on the 'Administration' dropdown.      3. Clicks on the 'Manage Projects' link.      4. 'Manage Projects' view is displayed.      5. Clicks the 'Create New' link and enters information.      6. New project is added to the system.</p>
<b>Extensions:</b>	The link will only be available to admin users if they have the correct permissions in the system.
<b>Frequency of Use:</b>	Each time a manager wants to add new projects to the system.
<b>Status:</b>	Design Stage.

FIGURE B.24: Add Project Use-Case-24

**Edit Project:**

<b>ID:</b>	AUC-25
<b>Title:</b>	Edit Project
<b>Description:</b>	In the 'Manage Projects' view, managers will be able to edit existing projects in the system.
<b>Primary Actor:</b>	Administrator
<b>Preconditions:</b>	The manager has logged in and clicked on the 'Administration' dropdown. The manager then clicks on the 'Manage Projects' link and clicks 'Edit'. The manager enters the details of the project they want to change.
<b>Postconditions:</b>	The project is saved in the database and changed in the system.
<b>Main Success Scenario:</b>	<p>1. The manager logs in.      2. Clicks on the 'Administration' dropdown.      3. Clicks on the 'Manage Projects' link.      4. 'Manage Projects' view is displayed.      5. Clicks the 'Edit' link and enters information.      6. The project is updated in the system.</p>
<b>Extensions:</b>	The link will only be available to admin users if they have the correct permissions in the system.
<b>Frequency of Use:</b>	Each time a manager wants to edit an existing project in the system.
<b>Status:</b>	Design Stage.

FIGURE B.25: Edit Project Use-Case-25

**Delete Project:**

<b>ID:</b>	AUC-26
<b>Title:</b>	Delete Project
<b>Description:</b>	In the 'Manage Projects' view, managers will be able to delete existing projects in the system.
<b>Primary Actor:</b>	Administrator
<b>Preconditions:</b>	The manager has logged in and clicked on the 'Administration' dropdown. The manager then clicks on the 'Manage Projects' link and clicks 'Delete'. The selected project will be removed from all dropdowns and removed completely from the system.
<b>Postconditions:</b>	The project is removed from the database and the system.
<b>Main Success Scenario:</b>	<p>1. The manager logs in.      2. Clicks on the 'Administration' dropdown.      3. Clicks on the 'Manage Projects' link.      4. 'Manage Projects' view is displayed.      5. Clicks the 'Delete' link and confirms.      6. The project is removed from the system.</p>
<b>Extensions:</b>	The link will only be available to admin users if they have the correct permissions in the system.
<b>Frequency of Use:</b>	Each time a manager wants to delete an existing project in the system.
<b>Status:</b>	Design Stage.

FIGURE B.26: Delete Project Use-Case-26

## Appendix C

# Appendix C - Test Cases and Results

This section holds all test cases.

### **Employee/Manager Login:**

**ID:** EMTC-1

**Title:** Employee/Manager Login

**Description:** Testing that all users are able to log into the system

**Primary Actor:** Employee/Manager

**Preconditions:** The user navigates to the login view and is prompted to enter login credentials

**Postconditions:** The user is logged in successfully and is taken to the home page of the web application

**Overall Test Result:** Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to login page		User is on the login screen	User is on the login screen	Pass
2.	Provide valid email	Mfb11127@uni.strath.ac.uk	User types in email with no validation errors	User types in email with no validation errors	Pass
3.	Provide valid password	*****	User types in password with no validation errors	User types in password with no validation errors	Pass
4.	Provide invalid/incorrect email	Mfb11127 or Mfb11127@virginmedia.com	Validation error or incorrect error is displayed above	Validation error or incorrect error is displayed above	Pass (both)
5.	Provide invalid/incorrect password	****	Validation error or incorrect error is displayed above	Validation error or incorrect error is displayed above	Pass (both)
4.	Click the login button		User is logged in and taken to homepage	User is logged in and taken to homepage	Pass

FIGURE C.1: Login Test-Case-1

**Employee/Manager Logout:**

<b>ID:</b>	EMTC-2
<b>Title:</b>	Employee/Manager Logout
<b>Description:</b>	Testing that all users are able to log out of the system
<b>Primary Actor:</b>	Employee/Manager
<b>Preconditions:</b>	The user is logged into the system and has navigated to any page and the logout link is available
<b>Postconditions:</b>	The user is successfully logged out of the system
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Login to the system	Mfb11127@uni.strath.ac.uk *****	User is logged into the system	User is logged into the system	Pass
2.	Click logout link		User is logged out of the system	User is logged out of the system	Pass

FIGURE C.2: Logout Test-Case-2

**Manage Records:**

<b>ID:</b>	ETC-3
<b>Title:</b>	Manage Records
<b>Description:</b>	Testing that all users are able to view and manage their own records
<b>Primary Actor:</b>	Users
<b>Preconditions:</b>	The user navigates to the home screen just after logging in and has the link to view their records in the middle of the page in a blue button
<b>Postconditions:</b>	The user's records load successfully in the form of a data-table
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to home page		User is on the home/welcome screen	User is on the home/welcome screen	Pass
2.	Click on blue button – "Your Work Records"		User enters the records view and can see all their records in a table	User enters the records view and can see all their records in a table	Pass

FIGURE C.3: Manage Records Test-Case-3

**Create Next Record:**

<b>ID:</b>	ETC-4
<b>Title:</b>	Create Next Record
<b>Description:</b>	Testing that all users are able to create next record
<b>Primary Actor:</b>	Users
<b>Preconditions:</b>	The user navigates to the manage records view, clicks on the "Create New Record" link and is shown a "Confirm New Record" view
<b>Postconditions:</b>	After confirmation, the user creates the new record and it is shown in the table
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Records" view		User is on the "Manage Records" view	User is on the "Manage Records" view	Pass
2.	Click on "Create New Record"	Auto-populated with an "Open" and time period is next three months	Confirmation page appears	Confirmation page appears	Pass
3.	Click on "Confirm New Record"	Auto-populated with an "Open" and time period is next three months	New record is created and shown in table	New record is created and shown in table	Pass

FIGURE C.4: Create New Record Test-Case-4

**Delete Record:**

<b>ID:</b>	ETC-5
<b>Title:</b>	Delete Record
<b>Description:</b>	Testing that managers can delete an employee's record
<b>Primary Actor:</b>	Manager
<b>Preconditions:</b>	The employee has an (Open AND Empty) record that they wish to remove. They ask their manager to delete it. The manager navigates to "Manage Employees" view clicks on the employee's records and clicks the "Delete" link to remove the record
<b>Postconditions:</b>	The record is removed and the employee is informed
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate "Manage Employees" view		Manager is on the "Manage Employees" screen	Manager is on the "Manage Employees" screen	Pass
2.	Click "View Records" for employee		Manager can see all records created by this employee	Manager can see all records created by this employee	Pass
3.	Click "Delete" link for the record		Confirm Delete view is shown	Confirm Delete view is shown	Pass
4.	Click "Confirm Delete"		Record is removed and no longer shown in table	Record is removed and no longer shown in table	Pass
5.	Employee tries to remove record		Links are not available (permission issues)	Links are not available (permission issues)	Pass

FIGURE C.5: Delete Record Test-Case-5

**Create Work:****ID:** ETC-6**Title:** Create Work**Description:** Testing that all users can create piece of work**Primary Actor:** Users**Preconditions:** The user navigates to "Manage Records" view and opens up current record. Clicks the link to create new work and enters details of piece of work (using previous performance page as a reference).**Postconditions:** The new piece of work is created and can be seen in the record**Overall Test Result:** Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Records" view		User is on the "Manage Records" screen	User is on the "Manage Records" screen	Pass
2.	Click "See Record"		Opens up record and can see any work that's been created	Opens up record and can see any work that's been created	Pass
3.	Click "Create New" link	10/03/2016, 09/03/2016, HR System, "Building and deploying new system. Delivered a day early and overall a success", HR System, Project Work	User can now enter details of created work	User can now enter details of created work	Pass
4.	Click "Create"		Piece of work is created and shown in record	Piece of work is created and shown in record	Pass
5.	Missing a field out		Validation error – "[field] is required"	Validation error – "[field] is required"	Pass

FIGURE C.6: Create Work Test-Case-6

**Edit Work:**

<b>ID:</b>	ETC-7
<b>Title:</b>	Edit Work
<b>Description:</b>	Testing that all users can edit an existing piece of work
<b>Primary Actor:</b>	Users
<b>Preconditions:</b>	The user navigates to "Manage Records" view and opens up current record. Clicks the link to edit the relevant piece of work and changes the information
<b>Postconditions:</b>	The piece of work is saved and can be seen in the record
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Records" view		User is on the "Manage Records" screen	User is on the "Manage Records" screen	Pass
2.	Click "See Record"		Opens up record and can see any work that's been created	Opens up record and can see any work that's been created	Pass
3.	Click "Edit" link	Change date completed from 09/03/2016 to 18/03/2016. Work is now late so add Late into description.	User can now enter the changed information	User can now enter the changed information	Pass
4.	Click "Save"		Piece of work is amended and shown in record	Piece of work is amended and shown in record	Pass
5.	Missing a field out		Validation error – "[field] is required"	Validation error – "[field] is required"	Pass

FIGURE C.7: Edit Work Test-Case-7

**Delete Work:**

<b>ID:</b>	ETC-8
<b>Title:</b>	Delete Work
<b>Description:</b>	Testing that all users can delete an existing piece of work
<b>Primary Actor:</b>	Users
<b>Preconditions:</b>	The user navigates to "Manage Records" view and opens up current record. Clicks the link to delete the relevant piece of work and confirms this
<b>Postconditions:</b>	The piece of work is removed and can now not be seen in the record
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Records" view		User is on the "Manage Records" screen	User is on the "Manage Records" screen	Pass
2.	Click "See Record"		Opens up record and can see any work that's been created	Opens up record and can see any work that's been created	Pass
3.	Click "Delete" link		User can now confirm this the work to be removed	User can now confirm this the work to be removed	Pass
4.	Click "Confirm"		Piece of work is removed and not shown in record	Piece of work is removed and not shown in record	Pass

FIGURE C.8: Delete Work Test-Case-8

**Submit Record:**

<b>ID:</b>	ETC-11
<b>Title:</b>	Submit Record
<b>Description:</b>	Testing that all users can submit a record to be graded by the manager
<b>Primary Actor:</b>	Users
<b>Preconditions:</b>	The user is logged in and navigates to the open record that they want to submit. Clicks on the submit button.
<b>Postconditions:</b>	The record is now submitted and can be graded by the manager
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to relevant record		User's record is shown on the screen with work created in it	User's record is shown on the screen with work created in it	Pass
2.	Click "Submit" button		The user must now confirm dialog box that they want to submit	The user must now confirm dialog box that they want to submit	Pass
3.	Click "OK"		The record is submitted and can now be graded by the manager	The record is submitted and can now be graded by the manager	Pass

FIGURE C.9: Submit Record Test-Case-9

**View Performance Page:**

<b>ID:</b>	ETC-12
<b>Title:</b>	View Performance Page
<b>Description:</b>	Testing that all users can view performance pages created for them by managers
<b>Primary Actor:</b>	Users
<b>Preconditions:</b>	The user navigates to "Manage Records" view. Clicks the link to view performance page for the relevant record. Managers can navigate to "Manage Employees", view records and see performance page this way
<b>Postconditions:</b>	The performance page is displayed and this can be used as a reference for the next record
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Records" view		User is on the "Manage Records" screen	User is on the "Manage Records" screen	Pass
2.	Click "Performance Page"		Opens up performance page for that record	Opens up performance page for that record	Pass
3.	(Managers) Navigate to "Manage Employees" view		Manager is on the "Manage Employees" screen	Manager is on the "Manage Employees" screen	Pass
4.	(Managers) Click "Records"		All Records are displayed for the employee	All Records are displayed for the employee	Pass
5.	(Managers) Click "View Record"		Record is opened	Record is opened	Pass
6.	(Managers) Click button to view performance page		Opens up performance page for that record	Opens up performance page for that record	Pass

FIGURE C.10: View Performance Page Test-Case-10

**Manage Employees:**

<b>ID:</b>	ETC-13
<b>Title:</b>	Manage Employees
<b>Description:</b>	Testing that all managers can "Manage Employees"
<b>Primary Actor:</b>	Manager
<b>Preconditions:</b>	The manager clicks on the "Administration" dropdown menu and selects "Manage Employees"
<b>Postconditions:</b>	The "Manage Employees" view is shown with a table of the employees in the manager's team
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Employees" view via "Administration" dropdown		User is on the "Manage Employees" screen	User is on the "Manage Employees" screen	Pass

FIGURE C.11: Manage Employees Test-Case-11

**Create Employee:**

<b>ID:</b>	ETC-14
<b>Title:</b>	Create Employee
<b>Description:</b>	Testing that all managers can create employee
<b>Primary Actor:</b>	Managers
<b>Preconditions:</b>	The manager is logged in and navigates to "Manage Employees" view via "Administration" dropdown. Manager clicks on create new and is able to now enter the new employee's information and clicks "Create"
<b>Postconditions:</b>	The employee is created and can now be seen in the team – The manager must now register new employee on the system via "Register New Account" (If the new user is to be a manager then the system's database administrator must be informed)
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Employees" view		Manager is on the "Manage Employees" screen	Manager is on the "Manage Employees" screen	Pass
2.	Click "Create New"	Aidan, Smith, mfb11127@uni.strath.ac.uk, IsActive - Checked	Manager can now enter details of created employee	Manager can now enter details of created employee	Pass
3.	Click "Create"		Employee is created and shown in the team	Employee is created and shown in the team	Pass
4.	Missing a field out		Validation error – "[field] is required" or	Validation error – "[field] is required"	Pass

FIGURE C.12: Create Employee Test-Case-12

**Edit Employee:****ID:** ETC-15**Title:** Edit Employee**Description:** Testing that all managers can edit an existing employee**Primary Actor:** Managers**Preconditions:** The manager is logged in and navigates to "Manage Employees" view via "Administration" dropdown. Manager clicks on edit and is able to now change the current information that is associated with employee**Postconditions:** The employee is changed and changes can now be seen in the team**Overall Test Result:** Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Employees" view		Manager is on the "Manage Employees" screen	Manager is on the "Manage Employees" screen	Pass
2.	Click "Edit"	Changed IsActive to unchecked	Manager can now change the details	Manager can now change the details	Pass
3.	Click "Save"		Employee details are changed and shown in the team	Employee details are changed and shown in the team	Pass
4.	Missing a field out		Validation error – "[field] is required" or	Validation error – "[field] is required"	Pass

FIGURE C.13: Edit Employee Test-Case-13

**View Employee Records:****ID:** ETC-16**Title:** View Employee Records**Description:** Testing that all managers can View Employee Records**Primary Actor:** Managers**Preconditions:** The manager is logged in and navigates to "Manage Employees" view via "Administration" dropdown. Manager clicks "Records"**Postconditions:** The view is now a table showing all records created by this employee**Overall Test Result:** Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Employees" view		Manager is on the "Manage Employees" screen	Manager is on the "Manage Employees" screen	Pass
2.	Click "Records"		The employee's records are shown on screen	The employee's records are shown on screen	Pass

FIGURE C.14: View Employee Records Test-Case-14

**Edit Record:**

<b>ID:</b>	ETC-17
<b>Title:</b>	Edit Record
<b>Description:</b>	Testing that all managers can edit records created by an employee – Done by removing the existing record and employee can create new record
<b>Primary Actor:</b>	Managers
<b>Preconditions:</b>	Employee must have no work currently in the record and it must still be open. The employee will request that the manager removes the record and he/she can now re-create the record amending the information.
<b>Postconditions:</b>	The record is changed and changes can now be seen in the table
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to “Manage Employees” view		Manager is on the “Manage Employees” screen	Manager is on the “Manage Employees” screen	Pass
2.	Click “Records”		The employee’s records are shown on screen	The employee’s records are shown on screen	Pass
3.	Click “Delete”		Confirm delete view is shown	Confirm delete view is shown	Pass
4.	Confirm the details		Record is removed and can be re-created	Record is removed and can be re-created	Pass

FIGURE C.15: Edit Record Test-Case-15

**View Record:**

<b>ID:</b>	ETC-18
<b>Title:</b>	View Record
<b>Description:</b>	Testing that all managers can view work that has been created within each record
<b>Primary Actor:</b>	Managers
<b>Preconditions:</b>	The manager is logged in and navigates to "Manage Employees" view via "Administration" dropdown. Manager clicks on "Records" for the relevant employee and then "View Record" for the relevant record
<b>Postconditions:</b>	The record is now opened and each piece of work created can be seen
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Employees" view		Manager is on the "Manage Employees" screen	Manager is on the "Manage Employees" screen	Pass
2.	Click "Records"		The employee's records are shown on screen	The employee's records are shown on screen	Pass
3.	Click "View Record"		The work within the record can now be seen in the table	The work within the record can now be seen in the table	Pass

FIGURE C.16: View Record Test-Case-16

**Work Detail:**

<b>ID:</b>	ETC-19
<b>Title:</b>	Work Detail
<b>Description:</b>	Testing that all managers can view work (in full detail) that has been created within each record
<b>Primary Actor:</b>	Managers
<b>Preconditions:</b>	Manager clicks on "Records" from "Manage Employees" for the relevant employee and then "View Record" for the relevant record. Then clicks on "Details"
<b>Postconditions:</b>	The piece of work is viewed fully with all fields displayed
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Employees" view		Manager is on the "Manage Employees" screen	Manager is on the "Manage Employees" screen	Pass
2.	Click "Records"		The employee's records are shown on screen	The employee's records are shown on screen	Pass
3.	Click "View Record"		The work within the record can now be seen in the table	The work within the record can now be seen in the table	Pass
4.	Click "Detail"		The piece of work is shown fully with all fields displayed	The piece of work is shown fully with all fields displayed	Pass

FIGURE C.17: Work Detail Test-Case-17

**Create Performance Page:**

<b>ID:</b>	ETC-20
<b>Title:</b>	Create Performance Page
<b>Description:</b>	Testing that all managers can create a performance page for submitted records
<b>Primary Actor:</b>	Managers
<b>Preconditions:</b>	Manager clicks on "Records" from "Manage Employees" /"Submitted Records" for the relevant employee and then "View Record" for the submitted record. Manager then inspects the work and clicks button to create performance page and enters the correct information
<b>Postconditions:</b>	The performance page is created and a notification email is sent to the employee
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Employees" view		Manager is on the "Manage Employees" screen	Manager is on the "Manage Employees" screen	Pass
2.	OR Navigate to "Submitted Records" view		Manager is on the "Submitted Records" screen	Manager is on the "Submitted Records" screen	Pass
3.	Click "Records"		The employee's records are shown on screen	The employee's records are shown on screen	Pass
4.	Click "View Record"		The work within the record can now be seen in the table	The work within the record can now be seen in the table	Pass
5.	Click button to create performance page	7 (Satisfactory). "Work was delivered on time and to a satisfactory standard. The reason it is not very good is that I would have expected more documentation to be provided.", "Please provide documentation for this and email me it by 20 <sup>th</sup> March. Also please update the active directory database by the end of the 3 month period."	Manager is able to enter the information	Manager is able to enter the information	Pass
6.	Click "Create"		Performance page is created and employee is notified	Performance page is created and employee is notified	Pass

FIGURE C.18: Create Performance Page Test-Case-18

**Submitted Records:**

<b>ID:</b>	ETC-21
<b>Title:</b>	Submitted Records
<b>Description:</b>	Testing that all managers can access all submitted records within their team
<b>Primary Actor:</b>	Managers
<b>Preconditions:</b>	Manager navigates to the "Submitted Records" view via "Administration" dropdown
<b>Postconditions:</b>	They can now view and access all submitted records that have come from their team members
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Submitted Records" view via "Administration" dropdown		Manager is on the "Submitted Records" screen	Manager is on the "Submitted Records" screen	Pass

FIGURE C.19: Submitted Records Test-Case-19

**Register New User:**

<b>ID:</b>	ETC-22
<b>Title:</b>	Register New User
<b>Description:</b>	Testing that all managers can register a new user on the system
<b>Primary Actor:</b>	Managers
<b>Preconditions:</b>	Manager navigates to the "Register New Account" view via "Administration" dropdown after adding the user to their team
<b>Postconditions:</b>	The new user is created successfully and they can now log in with the new credentials
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Register New Account" view via "Administration" dropdown	Mfb11127@uni.strath.ac.uk ***** *****	Manager is able to enter details for new account	Manager is able to enter details for new account	Pass
2.	Click "Create"		The new user is created and logged into the system	The new user is created and logged into the system	Pass
3.	Missing a field out		Validation error – "[field] is required" or	Validation error – "[field] is required"	Pass
4.	Enter invalid email or password type	Mfb11127 ***	Error – Email is not a valid email address, password must be at least 6 chars long, number, non-numeric	Error – Email is not a valid email address, password must be at least 6 chars long, number, non-numeric	Pass
5.	Create new account without adding the employee into the team in "Manage Employees"	Yyb11165@uni.strath.ac.uk ***** *****	User friendly error – Request is not possible. Contact support or Homepage	User friendly error – Request is not possible. Contact support or Homepage	Pass

FIGURE C.20: Register New User Test-Case-20

**Manage Projects:**

<b>ID:</b>	ETC-23
<b>Title:</b>	Manage Projects
<b>Description:</b>	Testing that all managers can access the "Manage Projects" view
<b>Primary Actor:</b>	Managers
<b>Preconditions:</b>	Manager navigates to the "Manage Projects" view via "Administration" dropdown
<b>Postconditions:</b>	They can now view and access all the projects that are worked on in the system
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Projects" view via "Administration" dropdown		Manager is on the "Manage Projects" screen	Manager is on the "Manage Projects" screen	Pass

FIGURE C.21: Manage Projects Test-Case-21

**Create Project:****ID:** ETC-24**Title:** Create Project**Description:** Testing that all managers can create projects**Primary Actor:** Managers**Preconditions:** The manager is logged in and navigates to "Manage Employees" view via "Administration" dropdown. Manager clicks on create new and is able to now enter the project's information and clicks "Create"**Postconditions:** The project is created and can now be seen in the table – Can also be seen in the dropdown menus for creating work**Overall Test Result:** Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Projects" view		Manager is on the "Manage Projects" screen	Manager is on the "Manage Projects" screen	Pass
2.	Click "Create New"	Team website, "New team website for collaborating ideas", IsActive – Checked, Dev – Checked, Service Desk – Checked, Ops - Checked	Manager can now enter details of created project	Manager can now enter details of created project	Pass
3.	Click "Create"		Project is created and shown in the table	Project is created and shown in the table	Pass
4.	Missing a field out		Validation error – "[field] is required" or	Validation error – "[field] is required"	Pass

FIGURE C.22: Create Project Test-Case-22

**Edit Project:**

<b>ID:</b>	ETC-25
<b>Title:</b>	Edit Project
<b>Description:</b>	Testing that all managers can edit projects
<b>Primary Actor:</b>	Managers
<b>Preconditions:</b>	The manager is logged in and navigates to "Manage Employees" view via "Administration" dropdown. Manager clicks on "Edit" and is able to now enter the new information and clicks "Save"
<b>Postconditions:</b>	The project is amended and can now be seen in the table – Can also be seen in the team dropdown menus for creating work
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Projects" view		Manager is on the "Manage Projects" screen	Manager is on the "Manage Projects" screen	Pass
2.	Click "Edit"	Change to "Dev Team Website" and uncheck Service Desk and Ops	Manager can now change the details	Manager can now change the details	Pass
3.	Click "Save"		Project is saved and shown in the table	Project is saved and shown in the table	Pass
4.	Missing a field out		Validation error – "[field] is required" or	Validation error – "[field] is required"	Pass

FIGURE C.23: Edit Project Test-Case-23

**Delete Project:**

<b>ID:</b>	ETC-26
<b>Title:</b>	Delete Project
<b>Description:</b>	Testing that all managers can delete projects
<b>Primary Actor:</b>	Managers
<b>Preconditions:</b>	The manager is logged in and navigates to "Manage Employees" view via "Administration" dropdown. Manager clicks on "Delete" and confirms this is the project to delete
<b>Postconditions:</b>	The project is removed and cannot be seen in the table or dropdowns
<b>Overall Test Result:</b>	Pass

Step	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Navigate to "Manage Projects" view		Manager is on the "Manage Projects" screen	Manager is on the "Manage Projects" screen	Pass
2.	Click "Delete"		Confirm screen is shown to confirm	Confirm screen is shown to confirm	Pass
3.	Click "Confirm"		Project is removed	Project is removed	Pass

FIGURE C.24: Delete Project Test-Case-24

## Appendix D

# Appendix D - SVN/GitHub Commits

This section demonstrates a list of commits to version control repositories.

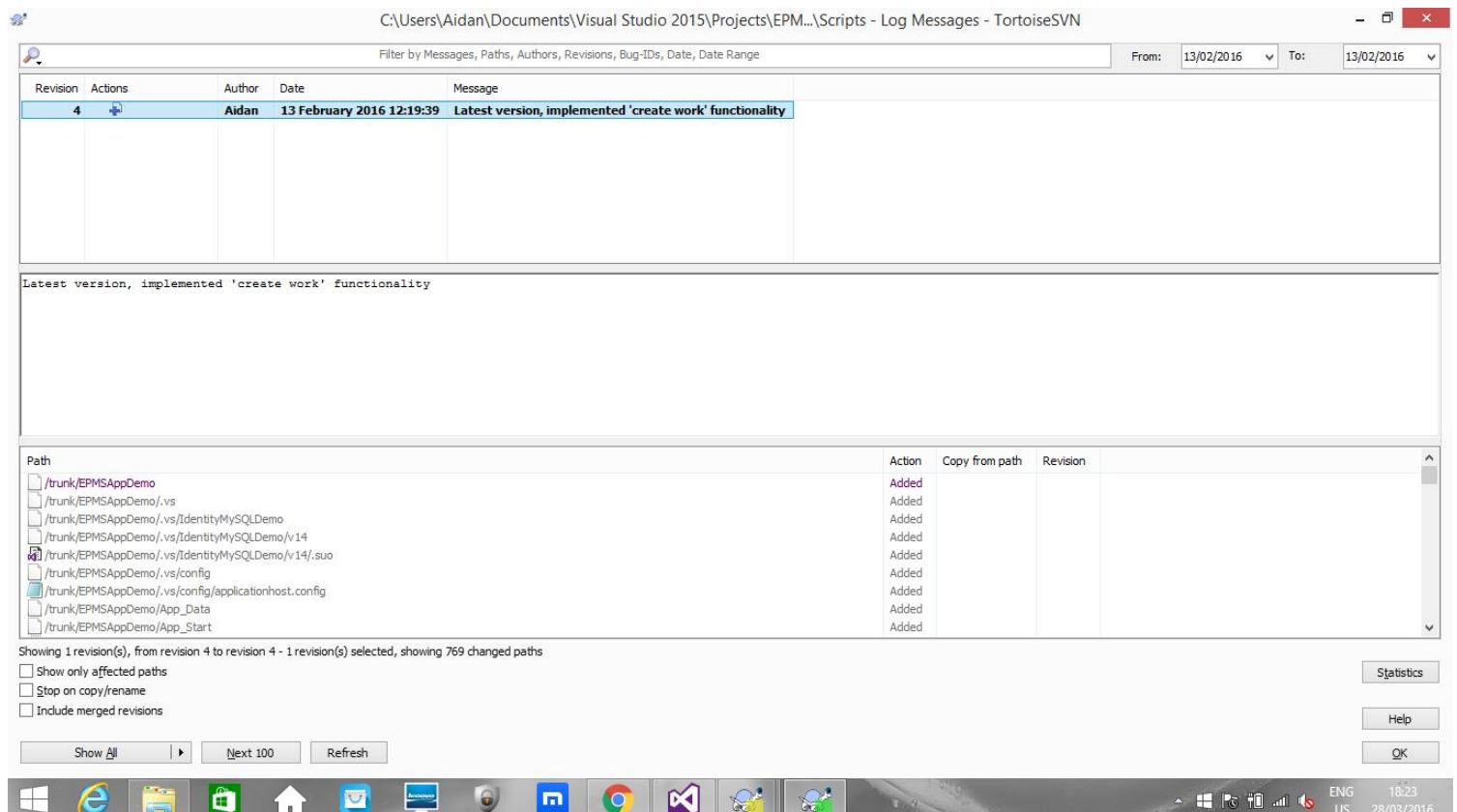


FIGURE D.1: EPMS - SVN Commits

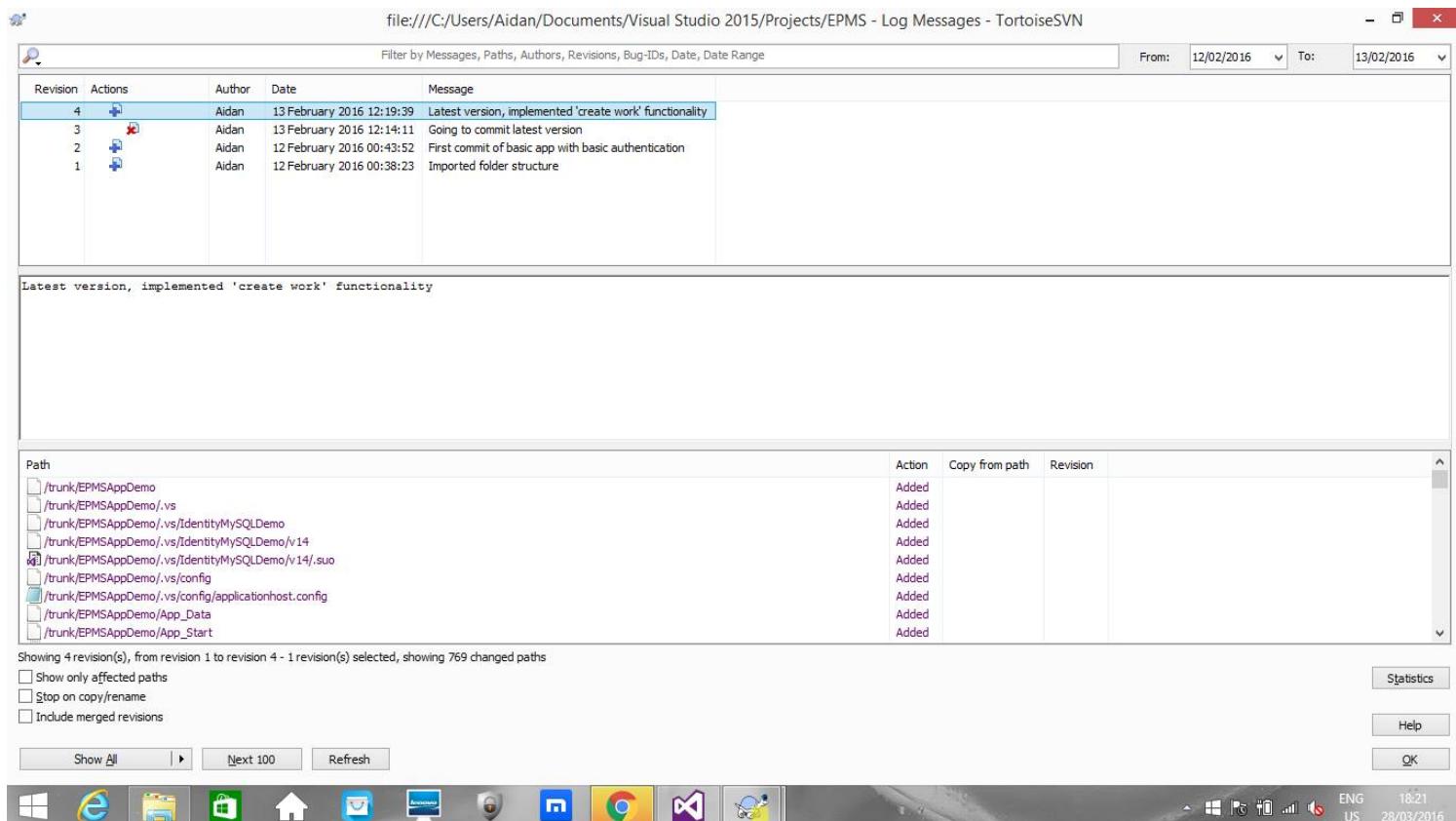


FIGURE D.2: EPMS - SVN Commits

Branch: master ▾

- ⌚ Commits on Mar 22, 2016
  -  **Added in comments throughout to improve documentation**  
Aidan Smith committed 5 days ago [copy] [0cc45d3] [diff]
- ⌚ Commits on Mar 21, 2016
  -  **Made some small changes before final user evaluation session.**  
Aidan Smith committed 6 days ago [copy] [4775d9a] [diff]
- ⌚ Commits on Mar 17, 2016
  -  **Last commit before second evaluation. Changed a couple of things in t...** [copy] [5a1d0fc] [diff]  
Aidan Smith committed 10 days ago
- ⌚ Commits on Mar 15, 2016
  -  **Worked on the 'About' page which will tell the user exactly what the ...** [copy] [b5714f5] [diff]  
Aidan Smith committed 12 days ago
- ⌚ Commits on Mar 9, 2016
  -  **Tidied up code, commenting out things that are unused. Will need to g...** [copy] [b028e63] [diff]  
Aidan Smith committed 18 days ago
- ⌚ Commits on Mar 8, 2016
  -  **Created and implemented custom error page, making it more user friend...** [copy] [5808bbd] [diff]  
Aidan Smith committed 19 days ago
- ⌚ Commits on Mar 7, 2016
  -  **Almost completed my test cases. Need to add in one custom error messa...** [copy] [f6e95b1] [diff]  
Aidan Smith committed 20 days ago

FIGURE D.3: EPMS - Git Commits

Commits on Mar 7, 2016			
 Almost completed my test cases. Need to add in one custom error messa... ...	<a href="#">f6e95b1</a>	<a href="#">Copy</a>	<a href="#">🔗</a>
Aidan Smith committed 20 days ago			
Commits on Mar 4, 2016			
 Fixed small iOS-DatePicker bug. Checkin if its an ipad/iphone user. Aidan Smith committed 23 days ago	<a href="#">7a4755d</a>	<a href="#">Copy</a>	<a href="#">🔗</a>
 Fixed bug when managers deleted employee's records. Aidan Smith committed 24 days ago	<a href="#">1c41c3f</a>	<a href="#">Copy</a>	<a href="#">🔗</a>
Commits on Mar 2, 2016			
 Completed the manage project functionality which will now add project... Aidan Smith committed 25 days ago	<a href="#">24bb4f1</a>	<a href="#">Copy</a>	<a href="#">🔗</a>
 Added 'Team Manager' flag top right to make it more clear when a mana... Aidan Smith committed 26 days ago	<a href="#">730a828</a>	<a href="#">Copy</a>	<a href="#">🔗</a>
Commits on Mar 1, 2016			
 Added the ability to rate a performance 'Very Good'. This is also now... Aidan Smith committed 26 days ago	<a href="#">889b44a</a>	<a href="#">Copy</a>	<a href="#">🔗</a>
Commits on Feb 29, 2016			
 Finished all of the 'must haves' in terms of features of my web appli... Aidan Smith committed 27 days ago	<a href="#">b6eafdf</a>	<a href="#">Copy</a>	<a href="#">🔗</a>
 Employees and managers can now access performance pages for each of t... ...	<a href="#">e2d392f</a>	<a href="#">Copy</a>	<a href="#">🔗</a>

FIGURE D.4: EPMS - Git Commits

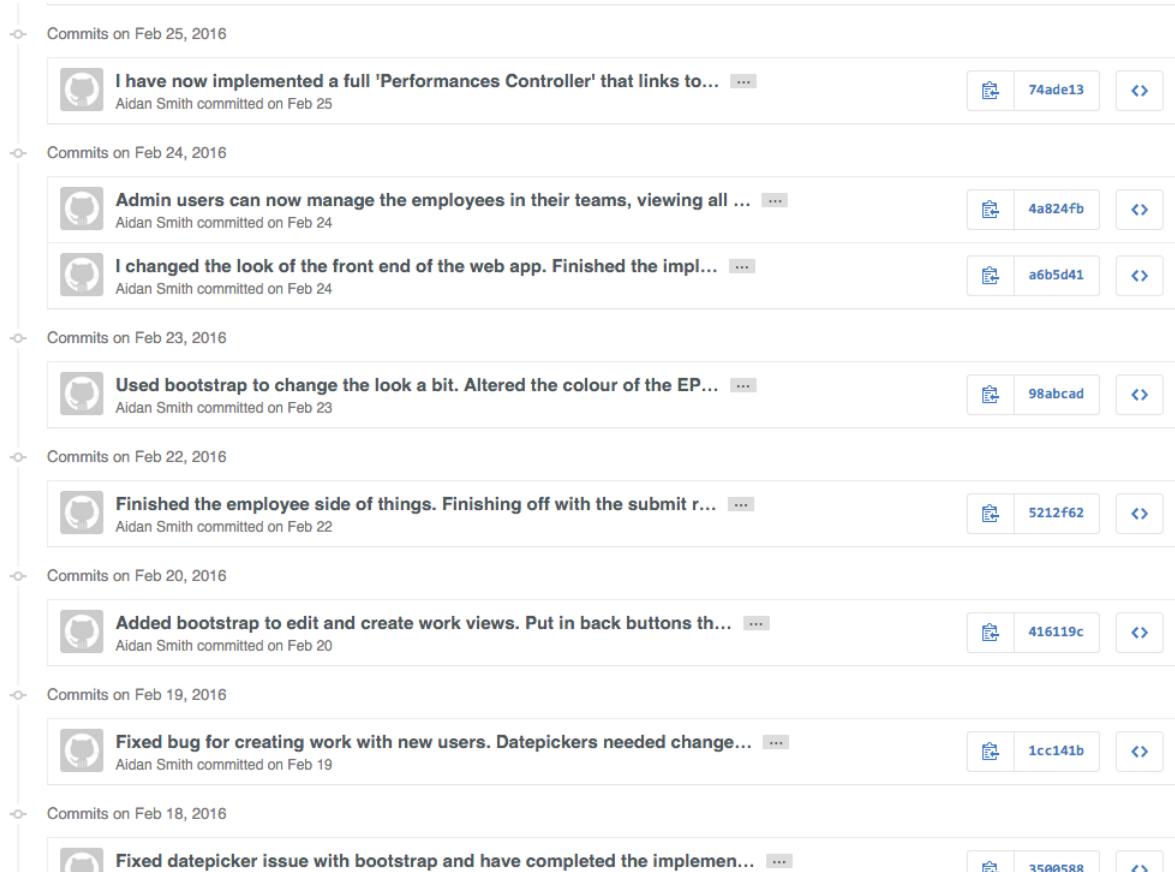


FIGURE D.5: EPMS - Git Commits

The screenshot shows a GitHub commit history for the EPMS repository. The commits are listed by date, with each commit represented by a card. The commits are as follows:

- Commits on Feb 22, 2016:
  - Finished the employee side of things. Finishing off with the submit r...** Aidan Smith committed on Feb 22. SHA: 5212f62.
- Commits on Feb 20, 2016:
  - Added bootstrap to edit and create work views. Put in back buttons th...** Aidan Smith committed on Feb 20. SHA: 416119c.
- Commits on Feb 19, 2016:
  - Fixed bug for creating work with new users. Datepickers needed change...** Aidan Smith committed on Feb 19. SHA: 1cc141b.
- Commits on Feb 18, 2016:
  - Fixed datepicker issue with bootstrap and have completed the implemen...** Aidan Smith committed on Feb 18. SHA: 3500588.
- Commits on Feb 17, 2016:
  - First commit to git. Currently at the stage where I have created full...** Aidan Smith committed on Feb 17. SHA: d44a3a4.
  - Initial commit to add default .gitignore and .gitAttribute files.** Aidan Smith committed on Feb 17. SHA: f070a22.

FIGURE D.6: EPMS - Git Commits

## Appendix E

# Appendix E - High Level Class Diagrams

This section displays some high level class diagrams illustrating the system.



FIGURE E.1: EPMS - High Level Class Diagram Of All Classes

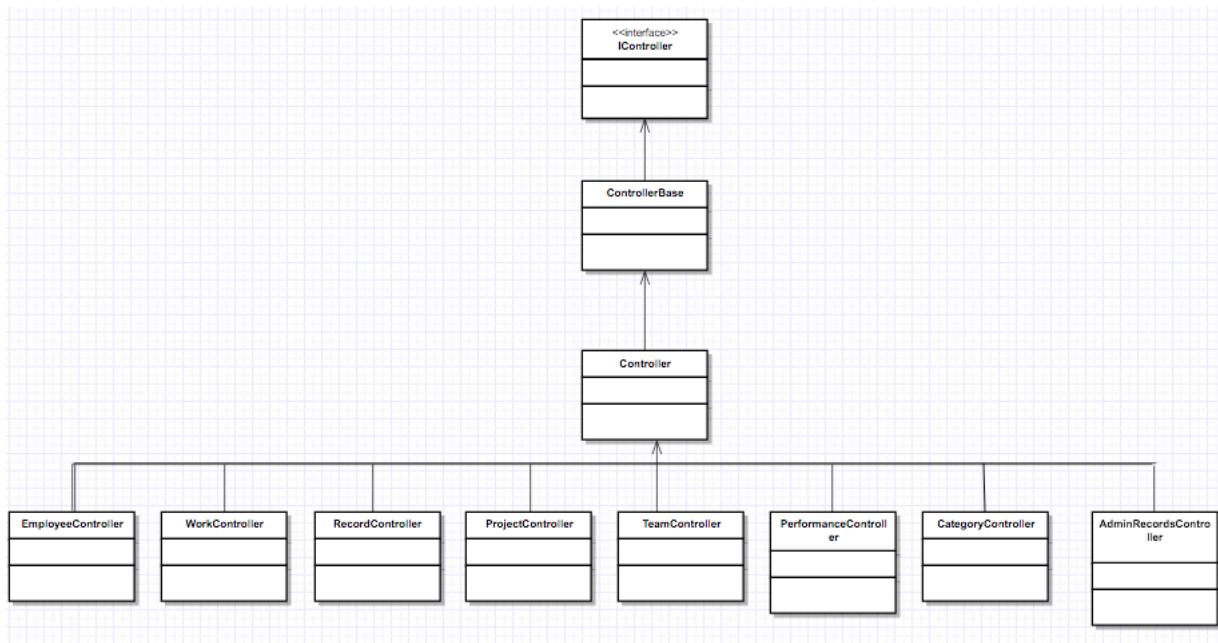


FIGURE E.2: EPMS - Controllers Class Diagram

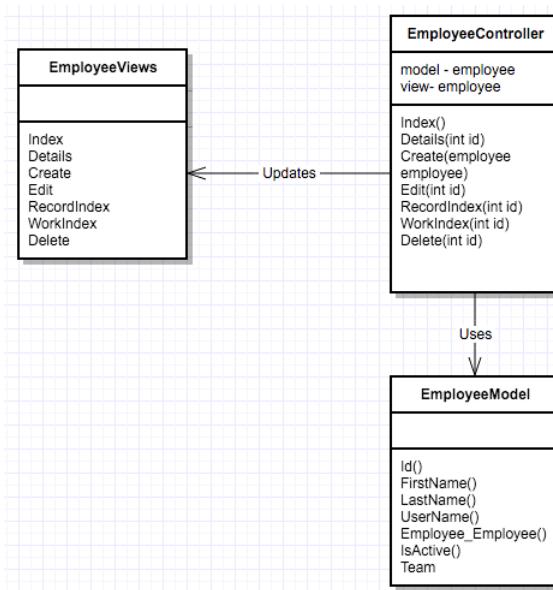


FIGURE E.3: EPMS - Employee Class Diagram

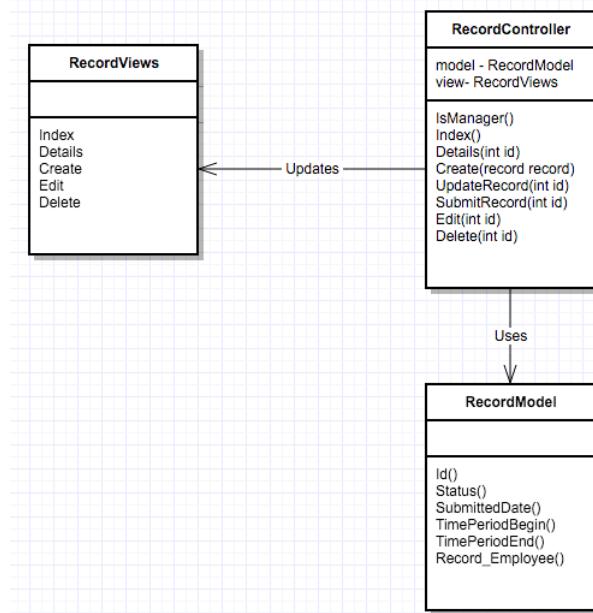


FIGURE E.4: EPMS - Record Class Diagram

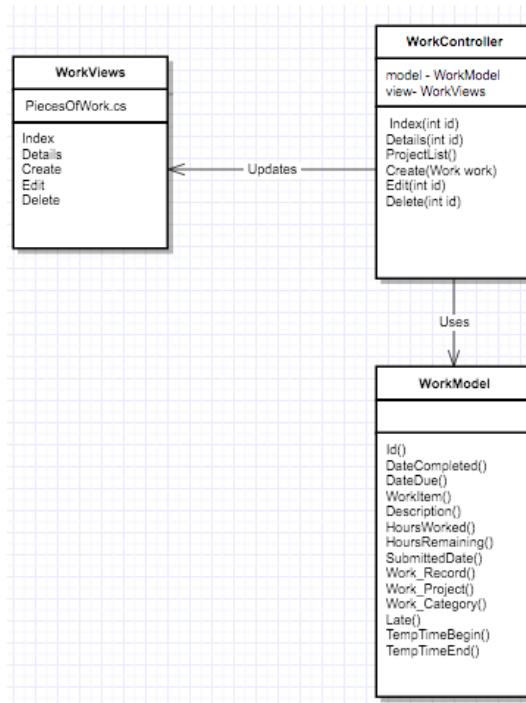


FIGURE E.5: EPMS - Work Class Diagram

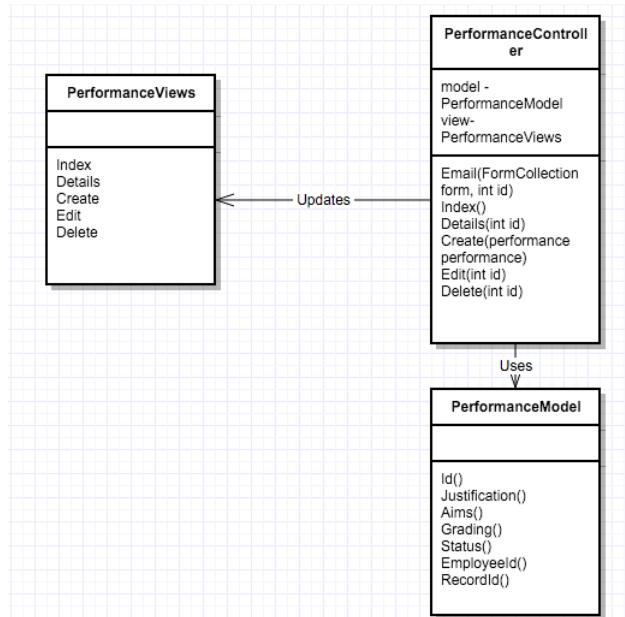


FIGURE E.6: EPMS - Performance Class Diagram

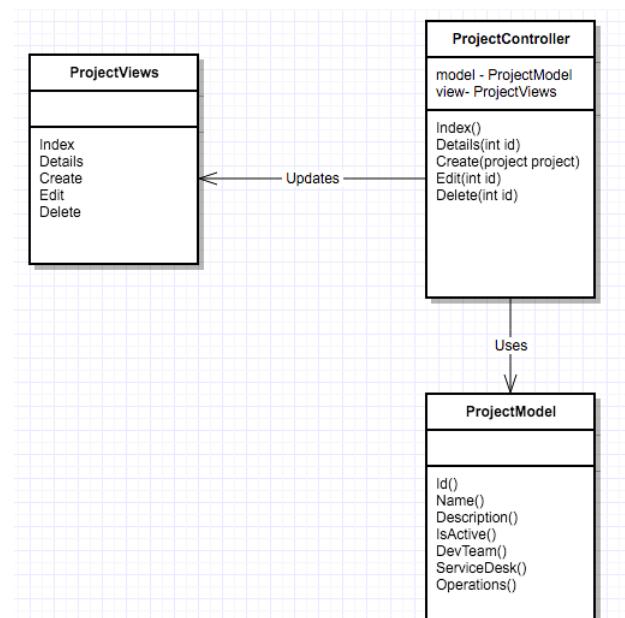


FIGURE E.7: EPMS - Project Class Diagram

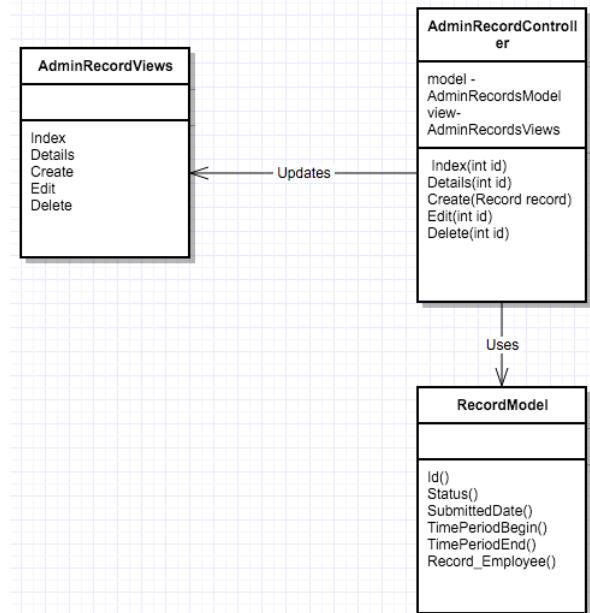


FIGURE E.8: EPMS - Admin Record Class Diagram

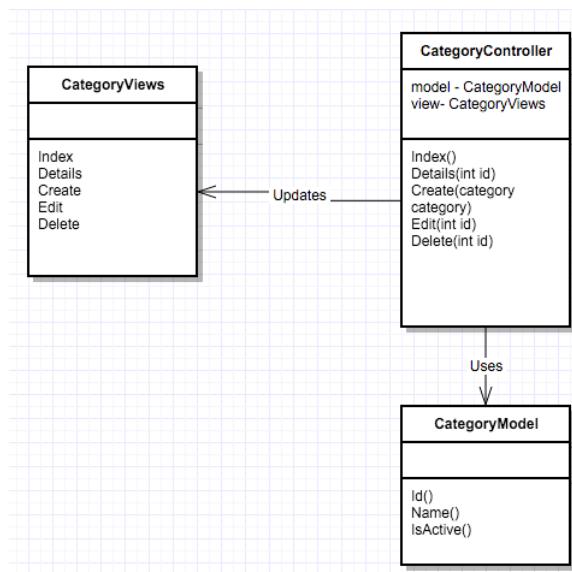


FIGURE E.9: EPMS - Category Class Diagram

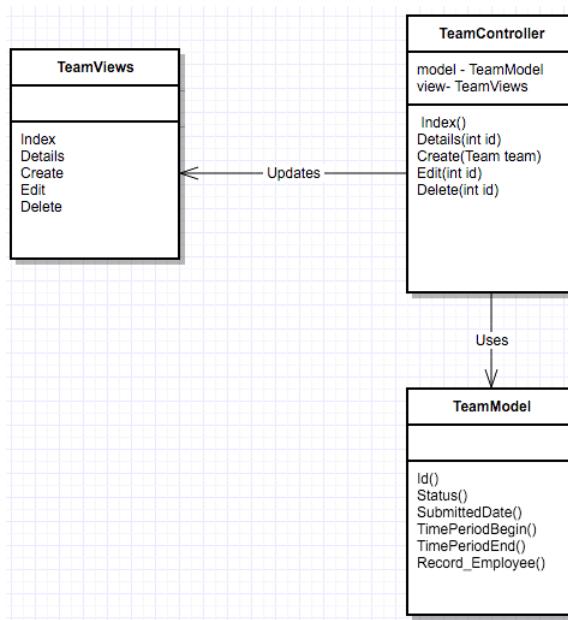


FIGURE E.10: EPMS - Team Class Diagram

## Appendix F

# Appendix H - User Evaluation Documents

This section outlines the content in documents used in the user evaluation stage of the project.

The following instruction sheet is the instructions that were carried out by the participants acting as managers:

**Evaluation Instruction Sheet: [REDACTED] (Manager) - Ops**

- Please click on the following link to access the web application: <http://epmsappdemo.azurewebsites.net/>
- Click 'Login Here' and enter login details provided to begin.
- Before you begin, you will need to create the accounts for your team members (See Aidan)

**Create a work record**

- To do this, open "Your Work Records" at the homepage and click create next work record and confirm. (This creates an open work record to hold your work for the next three months)

**Create work in the record**

- To do this open the record using "See Record" link. Click "Create New" and enter the following test data:

**DateDue:** Today's Date  
**DateCompleted:** Tomorrow's Date  
**WorkItem:** Service Manager  
**Description/Comments:** Training course in Edinburgh to help with the use of the new Service Manager system that I will be using on a day-to-day basis. This was beneficial.  
**Project:** Service Manager Training  
**Category:** Project Work
- Click "Create" when you are happy with the work that you have entered (You wont be able to create the work successfully unless you complete all fields on the page)

**Create another piece of work in the record**

- Repeat steps above creating another piece of work.
- This time use your own example from a project you have worked on recently in university or work.
- Test data will be your own but for project and category use the 'evaluation' option.

**Add SCOPE training to projects**

- From "Administration" dropdown and select "Manage Projects"
- Create New and use test data:

**Name:** SCOPE Training  
**Description:** SCOPE Training  
**IsActive:** True  
**Dev:** True  
**Service Desk:** True  
**Operations:** True
- Click create and this will now be added into the dropdown list and table for projects

**Grade Gerry Findlay's first record**

- Click the "Administration" dropdown and select "Submitted Records" to open up all submitted records for your team
- Select Gerry's "View Record", the record you will be grading

FIGURE F.1: EPMS - Manager Instruction Sheet

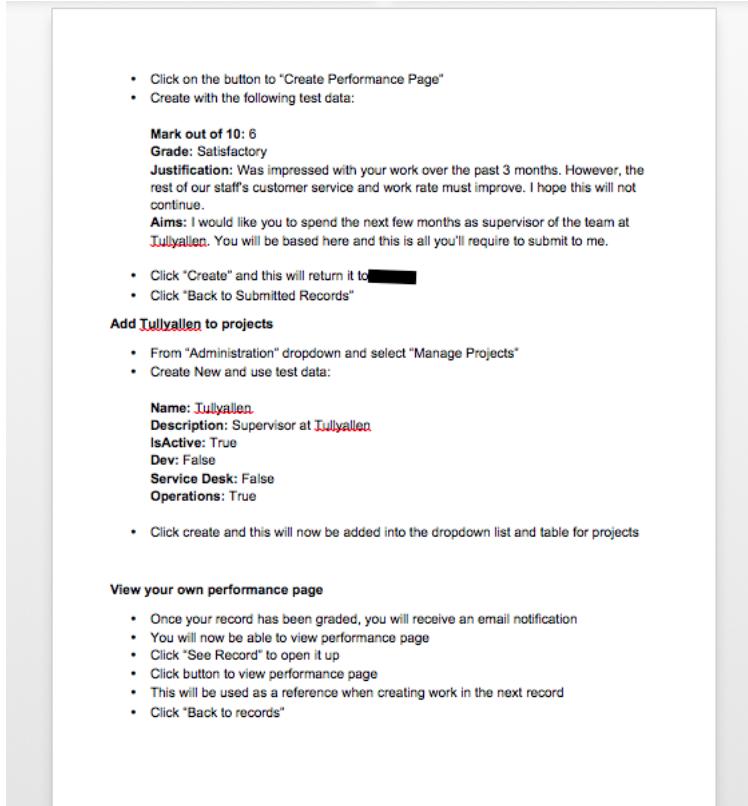


FIGURE F.2: EPMS - Manager Instruction Sheet 2

The following instruction sheet is the instructions that were carried out by the participants acting as employees:

**DEPARTMENT OF COMPUTER & INFORMATION SCIENCES**

**University of Strathclyde**

**Employee Performance Management System**

**Evaluation Instruction Sheet:** [REDACTED]

- Please click on the following link to access the web application:  
<http://epmsappdemo.azurewebsites.net/>
- Click "Login Here" and enter login details provided to begin.

**Create a work record**

- To do this, open "Your Work Records" at the homepage and click create next work record and confirm. (This creates an open work record to hold your work for the next three months)

**Create work in the record**

- To do this open the record using "See Record" link. Click "Create New" and enter the following test data:

**DateDue:** Today's Date  
**DateCompleted:** Tomorrow's Date  
**WorkItem:** Service Manager

**Description/Comments:** Training course in Glasgow to help with the use of the new Service Manager system that I will be using on a day-to-day basis. Very worthwhile.

**Project:** Service Manager Training  
**Category:** Project Work

- Click "Create" when you are happy with the work that you have entered (You wont be able to create the work successfully unless you complete all fields on the page)

**Edit the piece of work**

- Now, edit the work via the "Edit" link
- Change the DateCompleted from Tomorrow's to today's

**REF** UK TOP 20 RESEARCH-INTENSIVE UNIVERSITY  
**THE** UK UNIVERSITY OF THE YEAR WINNER  
**THE** UK ENTREPRENEURIAL UNIVERSITY OF THE YEAR WINNER

The place of useful learning

**Create another piece of work in the record**

- Repeat steps above creating another piece of work.
- This time use your own example from a project you have worked on recently in university or work.
- Test data will be your own but for project and category use the "evaluation" option.

**Create another piece of work in the record**

- Click "Create New" and enter the test data:

**DateDue:** 02/05/2016  
**DateCompleted:** 01/05/2016  
**Workitem:** HR Testing

**Description/Comments:** Testing the new HR System using the provided test data. System is fully functional with no problems.

**Project:** HR System  
**Category:** Evaluation Category

**Submit the record to your manager**

- Click the button to submit this record
- Confirm in the dialog box and record will be submitted

**View performance page**

- Once your record has been graded, you will receive an email notification
- You will now be able to view performance page
- Click "See Record" to open it up
- Click button to view performance page
- This will be used as a reference when creating work in the next record
- Click "Back to records"

**Create next record**

- Follow steps above for creating the next record

**Create new piece of work**

- Follow above steps again to create new piece of work in this record
- Use the following test data:

**DateDue:** 14/08/2016

FIGURE F.3: EPMS - Employee Instruction Sheet

DateCompleted: 14/08/2016

WorkItem: Supervisor at [Tulyallen](#)

Description/Comments: As instructed, I have been helping in a 2-month project to improved skill levels of service desk staff. Supervising the whole team went very well and we feel an improvement has been made already. Especially by [REDACTED].

Project: [Tulyallen](#)

Category: Training

Submit the record to your manager

- Click the button to submit this record
- Confirm in the dialog box and record will be submitted

Date: \_\_\_\_\_ Signed: \_\_\_\_\_

FIGURE F.4: EPMS - Employee Instruction Sheet 2

The following survey is the survey that was provided to the participants to complete after they had carried out all of the instructions:

**DEPARTMENT OF COMPUTER & INFORMATION SCIENCES**

**University of Strathclyde**  
Science

**Employee Performance Management System**

**Evaluation Survey:**

1. What were your initial impressions of the EPMS?

2. On a scale of 1 to 10, how attractive did you find the user interface? (How it looks?)

3. On a scale of 1 to 10, how satisfied are you with this software's ease of use?

4. On a scale of 1 to 10, how satisfied are you with the feedback process provided when a record is graded?

5. On a scale of 1 to 10, how satisfied are you with the account security within the software?

6. Any feature that you particularly like or dislike?

7. Any thoughts on how to improve the web application?

**VERY UNATTRACTIVE**                           **VERY ATTRACTIVE**

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

**VERY UNSATISFIED**                           **VERY SATISFIED**

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

Reasons:

The place of useful learning

**REF** UK TOP 20 RESEARCH-INTENSIVE UNIVERSITY  
**THE** UK UNIVERSITY OF THE YEAR WINNER  
**THE** UK ENTREPRENEURIAL UNIVERSITY OF THE

FIGURE F.5: EPMS - Evaluation Survey Questions

Ethics approval:



DEPARTMENT OF COMPUTER & INFORMATION SCIENCES

**Employee Performance Management System**

**Declaration of Participation:**

I understand that:

1. I am under no obligation to complete the research questions and do not have to participate fully with everything I am asked.
2. The aim of the project is to produce an employee performance management system which could be used in any professional environment to monitor and maintain the performance of employees within an organisation.
3. Data collected will only be used to aid understanding of the project and for inspiration when collating ideas.
4. There will be no personal/sensitive data collected from the research.
5. All data collected from the interviews will be completely confidential and will be disposed of once the project comes to an end.
6. The data will be stored electronically on hard disk and on paper copy.
7. The data may be published in a final report and/or presentation.
8. I will be asked to participate in evaluation of the project. This may include physically testing the system and completing survey questions.
9. Ethical consent has been demonstrated from this document.

Date: \_\_\_\_\_ Signed: \_\_\_\_\_

FIGURE F.6: EPMS - Ethics Consent Form

## Appendix G

# Appendix G - User Guide

This section is a guide on how to use the EPMS web application. Software is compatible on Chrome, FireFox, IE and Safari and can be used on mobile devices. You are required to have an Internet connection for use.

### EMPLOYEES:

Welcome to your EPMS! Click "Login Here" to get started.

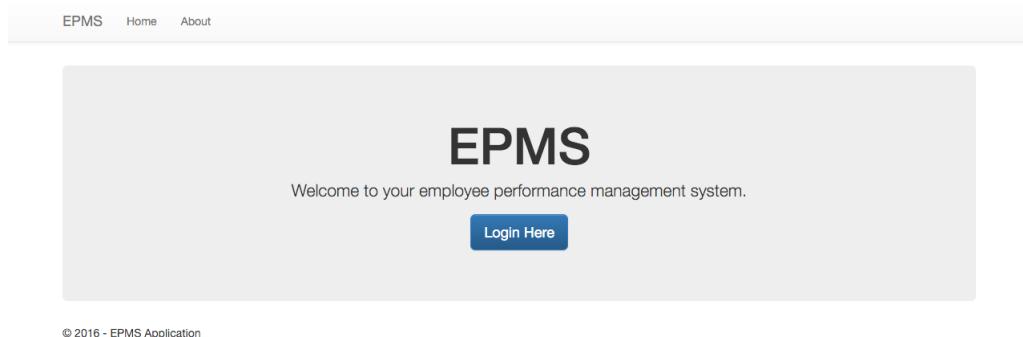


FIGURE G.1: EPMS - Home Page

Enter your login details.

 A screenshot of a web browser showing the EPMS login page. The URL bar shows the address as epmsappdemo.azurewebsites.net/Account/Login. The page has a light gray header with the EPMS logo and navigation links for Home and About. Below this, a 'Log in.' heading and a 'Use your EPMS account to login.' message are displayed. There are two input fields: 'Email' containing a placeholder email address and 'Password' containing a masked password. A 'Remember me?' checkbox is present, followed by a 'Log in' button. At the bottom, there is a copyright notice for 2016.

FIGURE G.2: EPMS - Login Page

First step is to open up your records, click on "Your Work Records".

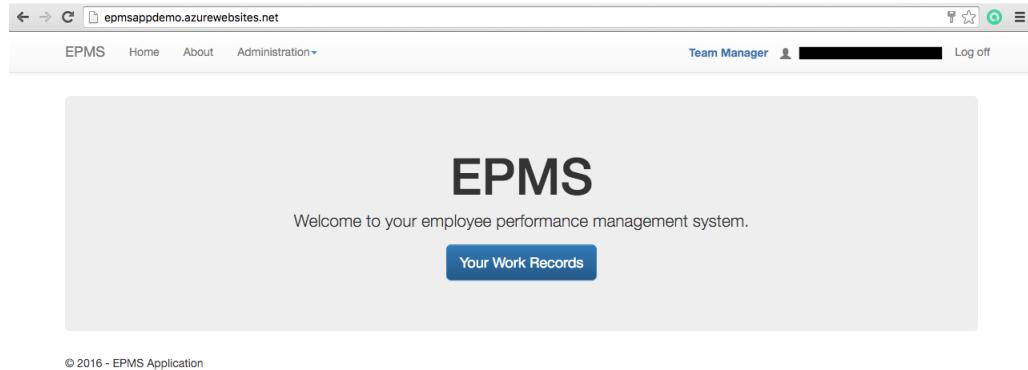


FIGURE G.3: EPMS - Logged In

To create a work record, click on "Create Next Record" and confirm the dates for the next three months.

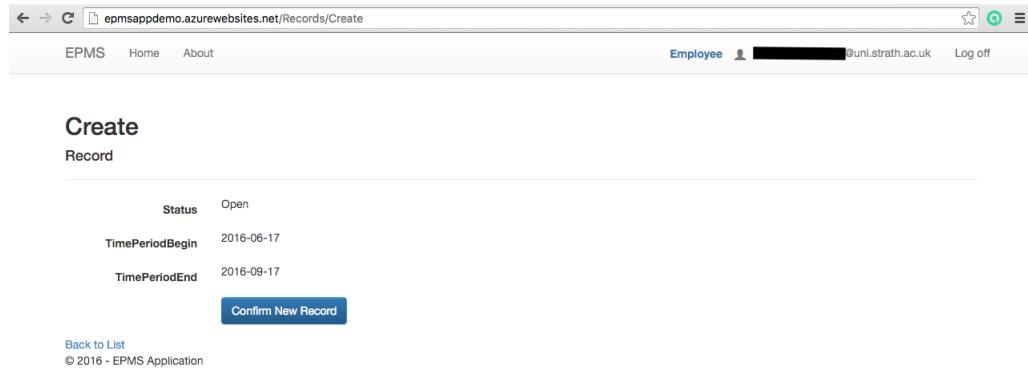


FIGURE G.4: EPMS - Create Record

Next you will be able to create work within the new record. Click "See Record" to open it up and manage the work for this three months.

The screenshot shows a web browser window for the EPMS application at the URL [epmsappdemo.azurewebsites.net/Records](http://epmsappdemo.azurewebsites.net/Records). The page title is "My Records". At the top, there are navigation links for "EPMS", "Home", and "About", and a user session info bar with "Employee" and "Log off". Below the title, there is a search bar labeled "Search:" and a link to "Performance Page". The main content area displays a table with two rows of data:

TimePeriodBegin	TimePeriodEnd	Status	Work Created	
2016-03-17	2016-06-17	Very Good	2	<a href="#">See Record</a>   <a href="#">Performance Page</a>
2016-06-17	2016-09-17	Open	0	<a href="#">See Record</a>

Below the table, there are links for "Create Next Record" and "Back To Home". A copyright notice at the bottom states "© 2016 - EPMS Application".

FIGURE G.5: EPMS - Record Created

The screenshot shows a web browser window for the EPMS application at the URL [epmsappdemo.azurewebsites.net/Works/Index/60](http://epmsappdemo.azurewebsites.net/Works/Index/60). The page title is "Work". At the top, there are navigation links for "EPMS", "Home", and "About", and a user session info bar with "Employee" and "Log off". Below the title, there is a search bar labeled "Search:" and a link to "No Work". The main content area displays a table with columns: "WorkItem", "Description", "DateCompleted", "Project", "Category", and "Late". A message "No data available in table" is displayed above the table. Below the table, there is a blue button labeled "No Work". A link "Back to Records" is visible at the bottom, along with a copyright notice "© 2016 - EPMS Application".

FIGURE G.6: EPMS - Empty Record

Create a new piece of work by clicking on "Create New" link and enter the details of the completed work.

**Create**  
Work

DateDue: 30/06/2016  
(If not specified, DateDue = today's date)

DateCompleted: 30/06/2016

WorkItem: Organisation OS Windows 10

Description/Comments:  
This project involved working with John Smith in the operations team rolling out 2 new operating systems to the organisation. The project went very smoothly with no issues and the work was completed on time.

Project: New Operating System

Category: Project Work

**Create**

[Back to List](#)  
© 2016 - EPMS Application

FIGURE G.7: EPMS - Create Work

**Work (From 2016-06-17 To 2016-09-17)**

[Create New](#)

Show 10 entries	Search:					
WorkItem	Description	DateCompleted	Project	Category	late	
Organisation OS Windows 10	This project involved working with John Smith in the operations team rolling out 2 new operating systems to the organisation. The project went very smoothly with no issues and the work was completed on time.	2016-06-30	New Operating System	Project Work	N	<a href="#">Edit   Delete</a>
Gizmoball group	Group project to produce a user-friendl...	2016-07-15	Other	Evaluation Category	N	<a href="#">Edit   Delete</a>

[Submit This Record](#)

[Back to Records](#)  
© 2016 - EPMS Application

FIGURE G.8: EPMS - After Work Is Added

You are able to manage the work in this view using the "Edit" and "Delete" links for each piece of work.

Click the long button to submit your work record to line manager for grading.

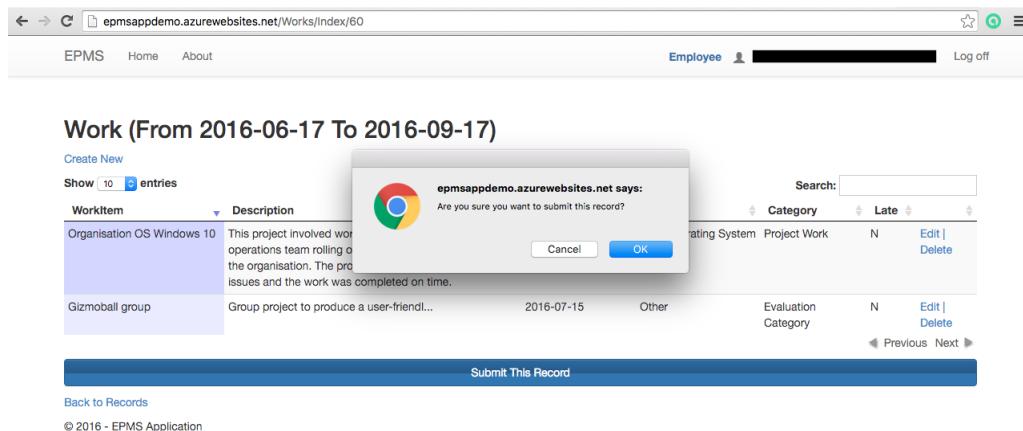


FIGURE G.9: EPMS - Submit Record

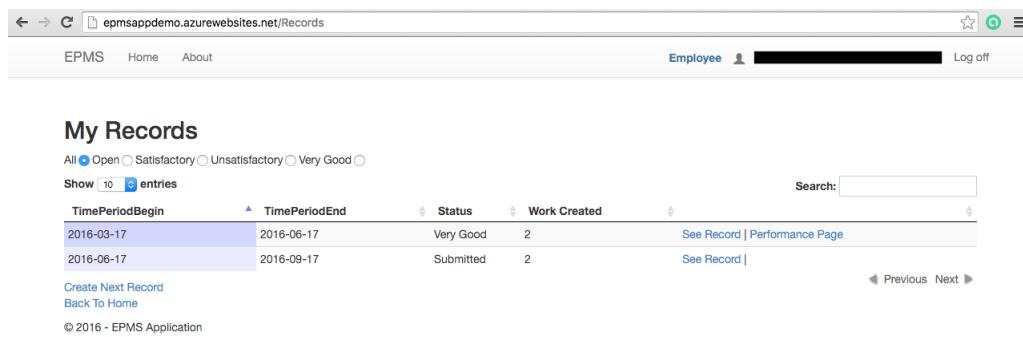


FIGURE G.10: EPMS - Updated Record

When a record has been graded a large button will appear to view the performance page.

The screenshot shows a web browser displaying the EPMS application at [epmsappdemo.azurewebsites.net/Works/Index/60](http://epmsappdemo.azurewebsites.net/Works/Index/60). The page title is "Work (From 2016-06-17 To 2016-09-17)". A message at the top states "This record has been submitted to admin.". Below this, there is a table with columns: WorkItem, Description, DateCompleted, Project, Category, and Late. Two entries are listed:

WorkItem	Description	DateCompleted	Project	Category	Late
Organisation OS Windows 10	This project involved working with John...	2016-06-30	New Operating System	Project Work	N
Gizmoball group	Group project to produce a user-friendl...	2016-07-15	Other	Evaluation Category	N

At the bottom of the table, there is a green bar with the text "Performance Page Created: View Here!". Navigation links "Back to Records" and "© 2016 - EPMS Application" are also present.

FIGURE G.11: EPMS - Graded Record

The screenshot shows a web browser displaying the EPMS application at [epmsappdemo.azurewebsites.net/Performances/Details/60](http://epmsappdemo.azurewebsites.net/Performances/Details/60). The page title is "Details". Under "Performance", it shows the following details:

Name	Kevin Paton
Begin - End	2016-06-17 - 2016-09-17
Status	Very Good
Grading	9
Justification	Excellent work on these projects. You carried out exactly what was required and everything was on time.
Aims	Next record, I want you to complete all data analysis for the SCOPE system database. Please complete this for 4th October. Also please complete Java training course that I have arranged for 3rd September.

Navigation links "Back to Records" and "© 2016 - EPMS Application" are also present.

FIGURE G.12: EPMS - View Performance

## MANAGER:

Managers are able to login and have the same permissions as all employees. For administration click on the "Administration" dropdown menu.

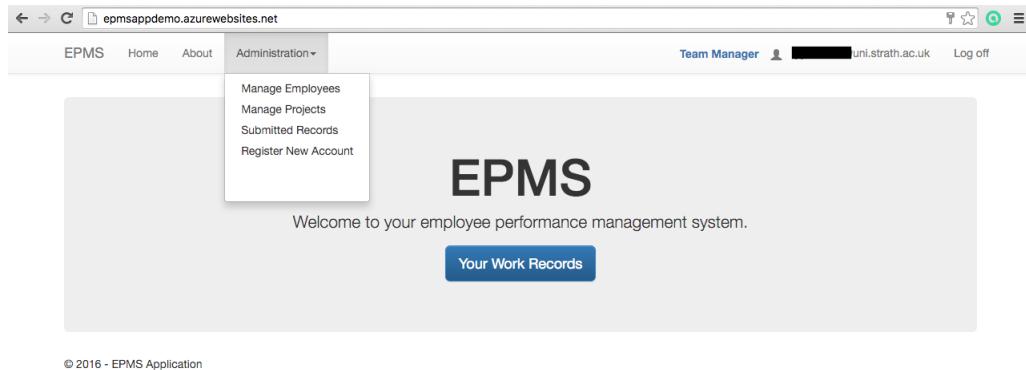


FIGURE G.13: EPMS - Administration Menu

To manage all employees in your team, click "Manage Employees" link. This will also give you access to all their work and details. Manage your employee details with the "Edit" and "Details" link.

The screenshot shows the "Development Employees" page. At the top, there is a "Create New" button and a search bar. Below the search bar is a table listing six employees. The columns are FirstName, LastName, Email/UserName, Manager, IsActive, Team, and Records. Each row contains a set of three links: Edit, Details, and Records. At the bottom of the table, there are navigation links for "Previous" and "Next".

FirstName	LastName	Email/UserName	Manager	IsActive	Team	Records	
Aidan	Smith	mfb11127@uni.strath.ac.uk	10	✓	Development	3	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Records</a>
Craig	Irving	craig.heron.2013@uni.strath.ac.uk	10	✓	Development	2	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Records</a>
David	Thomson	ntb11170@uni.strath.ac.uk	10	✓	Development	3	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Records</a>
James	Hendrie	james.hendrie.2013@uni.strath.ac.uk	10	✓	Development	2	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Records</a>
Jordan	Wilson	yjb11164@uni.strath.ac.uk	4	✓	Development	2	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Records</a>
Kevin	Paton	kevin.paton.2013@uni.strath.ac.uk	10	✓	Development	2	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Records</a>

© 2016 - EPMS Application

FIGURE G.14: EPMS - Manage Employees

There is also a "Submitted Records" queue available to managers that contains all submitted records.

Submitted Records						
Show 10 <input type="button" value="View All"/> entries <input type="text" value="Search:"/>						
Name	TeamName	TimePeriodBegin	TimePeriodEnd	Status		
Kevin Paton	Development	2016-06-17	2016-09-17	Submitted	<a href="#">View Record</a>	
Craig Irving	Development	2016-06-17	2016-09-17	Submitted	<a href="#">View Record</a>	

© 2016 - EPMS Application

FIGURE G.15: EPMS - Submitted Records

You can view any employee's work by selecting their records from the "Records" link (either from "Records" in managing employees or "View Records" in submitted records queue).

Work (From 2016-06-17 To 2016-09-17)						
Kevin Paton <input type="text" value="Search:"/>						
Show 10 <input type="button" value="View All"/> entries <input type="text" value="Search:"/>						
DateCompleted	WorkItem	Description	Project	Category	Late	
2016-07-15	Gizmoball group	Group project to produce a user-friendly gizmoball game. Other Team working skills were required and this was a successful project as was completed really early.	Evaluation Category	N	<a href="#">Details</a>	
2016-06-30	Organisation OS Windows 10	This project involved working with John Smith in the operations team rolling out 2 new operating systems to the organisation. The project went very smoothly with no issues and the work was completed on time.	New Operating System	Project Work	N	<a href="#">Details</a>

[Create Performance Page](#)

[Back To Employees](#)  
[Back To Submitted Records](#)  
 © 2016 - EPMS Application

FIGURE G.16: EPMS - Inspecting Work

To grade work submitted by employees you create a performance page. To do this click on the button "Create Performance Page". An email notification will be sent to the employee when their record is graded.

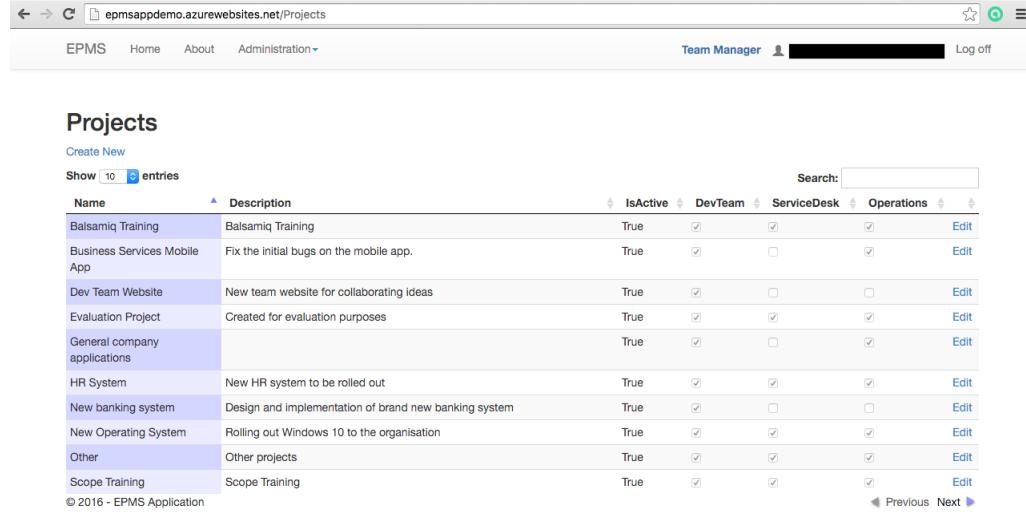
The screenshot shows a web browser window for the EPMS application. The URL in the address bar is `epmsappdemo.azurewebsites.net/Performances/Create/60`. The page title is "Create" under "Performance". The top navigation bar includes links for EPMS, Home, About, Administration, Team Manager (with email yyb11164@uni.strath.ac.uk), and Log off. The main content area contains the following form fields:

- Grading:** A dropdown menu set to "9".
- Status:** A dropdown menu set to "Very Good".
- Justification:** A text area containing the text: "Excellent work on these projects. You carried out exactly what was required and everything was on time."
- Aims:** A text area containing the text: "Next record, I want you to complete all data analysis for the SCOPÉ system database. Please complete this for 4th October. Also please complete Java training course that I have arranged for 3rd September."

At the bottom of the form is a blue "Create" button.

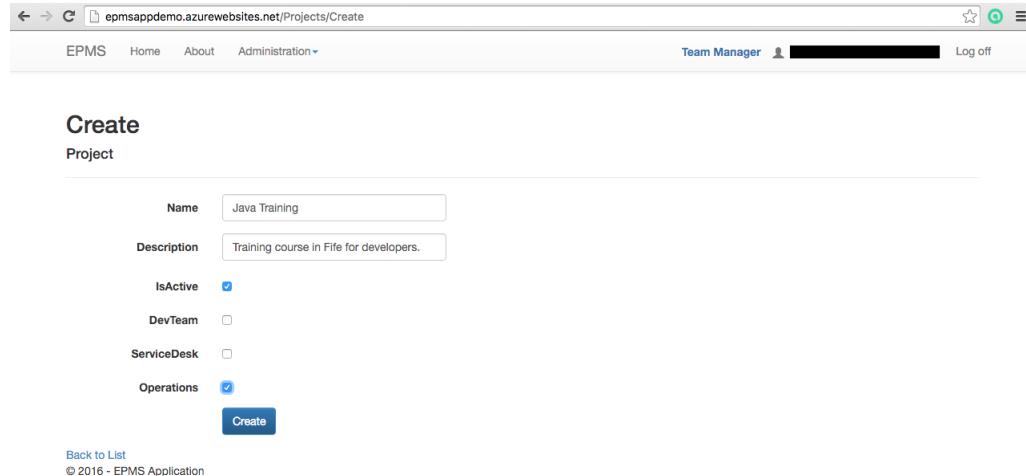
FIGURE G.17: EPMS - Grading Record

Next, the new projects have to be added. Access "Manage Projects" from the "Administration dropdown". You can create and edit projects here.



Name	Description	IsActive	DevTeam	ServiceDesk	Operations	
Balsamiq Training	Balsamiq Training	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>
Business Services Mobile App	Fix the initial bugs on the mobile app.	True	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>
Dev Team Website	New team website for collaborating ideas	True	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<a href="#">Edit</a>
Evaluation Project	Created for evaluation purposes	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>
General company applications		True	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>
HR System	New HR system to be rolled out	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>
New banking system	Design and implementation of brand new banking system	True	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<a href="#">Edit</a>
New Operating System	Rolling out Windows 10 to the organisation	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>
Other	Other projects	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>
Scope Training	Scope Training	True	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>

FIGURE G.18: EPMS - Manage Projects



Name	Java Training
Description	Training course in Fife for developers.
IsActive	<input checked="" type="checkbox"/>
DevTeam	<input type="checkbox"/>
ServiceDesk	<input type="checkbox"/>
Operations	<input checked="" type="checkbox"/>
<a href="#">Create</a>	

FIGURE G.19: EPMS - Create New Project

**Create New Users (MANAGER):**

To create a new user on the system two steps have to be carried out:

- Create new employee in "Manage Employees"
- "Register New Account" in the "Administration" dropdown

The screenshot shows a web-based application interface titled 'Create Employee'. It contains fields for 'FirstName' (Employee), 'LastName' (One), and 'Email/UserName' (employee1@uni.strath.ac.uk). A checkbox labeled 'IsActive' is checked. At the bottom is a blue 'Create' button. Below the form, there are links for 'Back to List' and copyright information: '© 2016 - EPMS Application'.

FIGURE G.20: EPMS - Create Employee In Team

The screenshot shows a web-based application interface titled 'Register'. It has fields for 'Email' (employee1@uni.strath.ac.uk), 'Password' (redacted), and 'Confirm password' (redacted). Below these is a blue 'Register' button. At the bottom, there are links for 'Back To Home' and copyright information: '© 2016 - EPMS Application'.

FIGURE G.21: EPMS - Register Account

## Appendix H

# Appendix H - Program Listing

Program listing for the project can be found on the electronic submission on MyPlace. The system is also currently available at <http://epmsappdemo.azurewebsites.net/> (for limited time only).

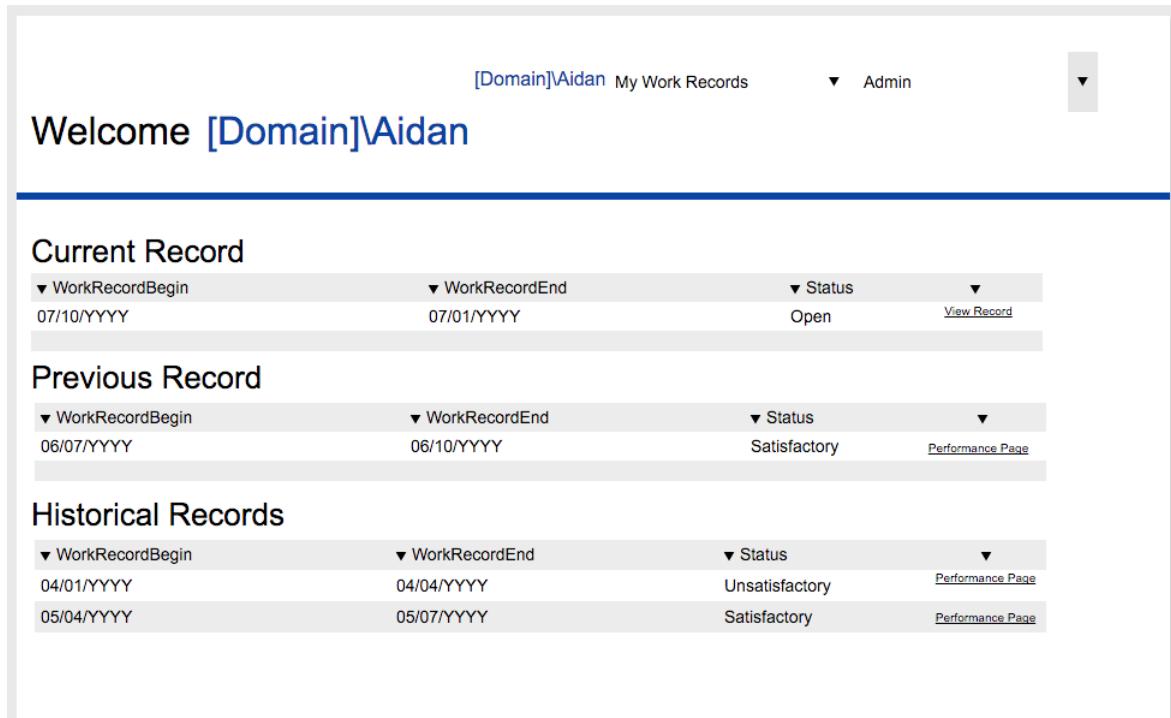
Test account has been created to be used by markers when inspecting the system.

Email - employee@uni.strath.ac.uk Password - Password.1

The application will be demonstrated fully (including email notifications) at the demo day.

## Appendix I

# Appendix I - UI Mockups and Ideas



The screenshot shows a user interface for a system named EPMS. At the top, there is a header bar with the text "[Domain]\Aidan My Work Records" on the left and "Admin" on the right, accompanied by a dropdown arrow. Below the header, a welcome message "Welcome [Domain]\Aidan" is displayed. The main content area is divided into three sections: "Current Record", "Previous Record", and "Historical Records", each containing a table with three columns: "WorkRecordBegin", "WorkRecordEnd", and "Status". In the "Current Record" section, the values are 07/10/YYYY, 07/01/YYYY, and Open, with a "View Record" link next to the status. In the "Previous Record" section, the values are 06/07/YYYY, 06/10/YYYY, and Satisfactory, with a "Performance Page" link next to the status. In the "Historical Records" section, the first row has values 04/01/YYYY, 04/04/YYYY, and Unsatisfactory, with a "Performance Page" link next to the status. The second row has values 05/04/YYYY, 05/07/YYYY, and Satisfactory, also with a "Performance Page" link next to the status.

WorkRecordBegin	WorkRecordEnd	Status	
07/10/YYYY	07/01/YYYY	Open	<a href="#">View Record</a>
WorkRecordBegin	WorkRecordEnd	Status	
06/07/YYYY	06/10/YYYY	Satisfactory	<a href="#">Performance Page</a>
WorkRecordBegin	WorkRecordEnd	Status	
04/01/YYYY	04/04/YYYY	Unsatisfactory	<a href="#">Performance Page</a>
05/04/YYYY	05/07/YYYY	Satisfactory	<a href="#">Performance Page</a>

FIGURE I.1: EPMS - Home

▼ WorkRecordBegin	▼ WorkRecordEnd	▼ Status	▼
04/01/YYYY	04/04/YYYY	Satisfactory	<a href="#">Edit</a>   <a href="#">Delete</a>   <a href="#">View Record</a>
05/04/YYYY	05/07/YYYY	Satisfactory	<a href="#">Edit</a>   <a href="#">Delete</a>   <a href="#">View Record</a>
06/07/YYYY	06/10/YYYY	Unsatisfactory	<a href="#">Edit</a>   <a href="#">Delete</a>   <a href="#">View Record</a>
07/10/YYYY	07/01/YYYY	Submitted	<a href="#">Edit</a>   <a href="#">Delete</a>   <a href="#">View Record</a>

FIGURE I.2: EPMS - Employee Records

[Domain]\Aidan My Work Records      ▼ Admin

## Work - Aidan Smith

---

### Create New Piece of Work

Work Item

Project  
 Options ▾

Category  
 Options ▾

Date Due      Date Completed

April	20	2011
-------	----	------

April	20	2011
-------	----	------

Date Started

April	20	2011
-------	----	------

[Back](#)

FIGURE I.3: EPMS - Employee Create Work

The screenshot shows a web-based application interface for EPMS. At the top, there is a header bar with the text "[Domain]\Aidan My Work Records" and "Admin". Below the header, the title "Work - Aidan Smith" is displayed. Underneath the title, there is a sub-header "(From 04/01/2015 To 04/04/2015)". A link "Create New" is visible. The main content area displays a list of work records with columns: "WorkItem", "WorkCompleted", "Project", and "Category". Each record includes a "Edit" and "Open Work" button. The records listed are:

WorkItem	WorkCompleted	Project	Category
Migrating SQL	04/02/YYYY	SQL2012 Migration	Maintenance
Built HR System	18/03/YYYY	New HR System	Project Development
Harassment Course	01/04/YYYY	Training Course	Non- Project Related

At the bottom of the list, there are links "Back to Records" and "Performance Page", and a prominent "Submit Record" button.

FIGURE I.4: EPMS - Employee Ready To Submit Record

This screenshot shows the same EPMS interface as Figure I.4, but with an additional "Admin" menu option in the top right corner of the header. The main content area is titled "My Work Records" and displays a list of work records with columns: "WorkRecordBegin", "WorkRecordEnd", and "Status". Each record includes "Edit", "Delete", and "View Record" buttons. The records listed are:

WorkRecordBegin	WorkRecordEnd	Status
04/01/YYYY	04/04/YYYY	Satisfactory
05/04/YYYY	05/07/YYYY	Satisfactory
06/07/YYYY	06/10/YYYY	Unsatisfactory
07/10/YYYY	07/01/YYYY	Submitted

FIGURE I.5: EPMS - Records With Admin Menu

The screenshot shows the EPMS Admin Manage Employees interface. At the top, there is a header bar with the text "[Domain]\Aidan My Work Records" and "Admin". Below the header, the title "[Team Name] Employees" is displayed, followed by navigation links: "Manage Employees", "Latest Work Records", and "Manage Projects". A "Create New" button is located on the left, and a search bar with a magnifying glass icon and an 'x' is on the right. A dropdown menu is visible on the far right. The main content area displays a table of employee records with columns: FirstName, LastName, UserName, IsActive, Team, and actions (Edit | View Records). The table contains three rows of data:

FirstName	LastName	UserName	IsActive	Team	
Aidan	Smith	[Domain]\Aidan	True	[Team Name]	<a href="#">Edit</a>   <a href="#">View Records</a>
Jordan	Wilson	[Domain]\Jordan	True	[Team Name]	<a href="#">Edit</a>   <a href="#">View Records</a>
Kris	Hunter	[Domain]\Kris	True	[Team Name]	<a href="#">Edit</a>   <a href="#">View Records</a>

FIGURE I.6: EPMS - Admin Manage Employees

The screenshot shows the EPMS Admin Create Employee interface. At the top, there is a header bar with the text "[Domain]\Aidan My Work Records" and "Admin". Below the header, the title "[Team Name] Employees" is displayed, followed by navigation links: "Manage Employees", "Latest Work Records", and "Manage Projects". The main content area has a heading "Create New Employee". It includes four input fields: "First Name" (with a text input box), "Last Name" (with a text input box), "User Name" (with a text input box), and "Is Active" (with a checked checkbox). Below these fields is a "Create" button and a link "Back to Employees".

FIGURE I.7: EPMS - Admin Create Employee

The screenshot shows the 'Edit Employee' form. At the top, there are navigation links: [Domain]\Aidan My Work Records, Admin, Manage Employees, Latest Work Records, and Manage Projects. Below the header, the title '[Team Name] Employees' is displayed. The main form fields include 'First Name' (Aidan), 'Last Name' (Smith), 'User Name' ([Domain]\Aidan), and 'Is Active' (checked). A 'Save' button is at the bottom left, and a 'Back to Employees' link is at the bottom right.

FIGURE I.8: EPMS - Admin Edit Employee

The screenshot shows the 'Records - Aidan Smith' page. At the top, there are navigation links: [Domain]\Aidan My Work Records, Admin, Manage Employees, Latest Work Records, and Manage Projects. Below the header, the title 'Records - Aidan Smith' is displayed. The main content area displays a table of work records:

WorkRecordBegin	WorkRecordEnd	Status	Action
04/01/YYYY	04/04/YYYY	Satisfactory	<a href="#">Open Record</a>
05/04/YYYY	05/07/YYYY	Satisfactory	<a href="#">Open Record</a>
06/07/YYYY	06/10/YYYY	Unsatisfactory	<a href="#">Open Record</a>
07/10/YYYY	07/01/YYYY	Submitted	<a href="#">Open Record</a>

A 'Back to Employees' link is located at the bottom left.

FIGURE I.9: EPMS - Admin Records

The screenshot shows the EPMS Admin Open Records interface. At the top, there is a header bar with the text "[Domain]\Aidan My Work Records" and "Admin". Below the header, the title "Work - Aidan Smith" is displayed, along with navigation links for "Manage Employees", "Latest Work Records", and "Manage Projects". A date range "(From 04/01/2015 To 04/04/2015)" is shown. The main content area displays a table of work items:

WorkItem	Date	Project	Category	Action
Migrating SQL	04/02/2015	SQL2012 Migration	Maintenance	<a href="#">Open Work</a>
Built HR System	18/03/2015	New HR System	Project Development	<a href="#">Open Work</a>
Harassment Course	01/04/2015	Training Course	Non- Project Related	<a href="#">Open Work</a>

At the bottom left is a link "Back to Records", and at the bottom right is a link "Performance Page".

FIGURE I.10: EPMS - Admin Open Records

The screenshot shows the EPMS Admin View Employee's Work interface. At the top, there is a header bar with the text "[Domain]\Aidan My Work Records" and "Admin". Below the header, the title "Work - Aidan Smith" is displayed, along with navigation links for "Manage Employees", "Latest Work Records", and "Manage Projects". A date range "(From 04/01/2015 To 04/05/2015)" is shown. The main content area displays a table of work item details:

Work Item:	Comments:
Migrating SQL	We completed the project within the deadline, submitting on the final day. There was no issues. Jordan helped a lot with the project.
Project:	
SQL2012 Migration	
Category:	
Maintenance	
Date Due:	Description:
04/02/2015	Through the previous performance page, I was given this project to work on with Jordan in migrating our current development instance of SQL2012 to SQL2015.
Date Started:	
04/01/2015	
Date Completed:	
04/02/2015	
Late?	
N	

At the bottom left is a link "Back", and at the bottom right is a link "Performance Page".

FIGURE I.11: EPMS - Admin View Employee's Work

The screenshot shows the 'Latest Work Records' section of the EPMS Admin interface. At the top, there are navigation links: 'Manage Employees', 'Latest Work Records' (which is the active tab), and 'Manage Projects'. Below the header, a table displays four rows of work records. Each row contains the employee name, start date, end date, status, and an 'Open Record' link.

Employee	WorkRecordBegin	WorkRecordEnd	Status
Aidan Smith	04/01/YYYY	04/04/YYYY	Submitted <a href="#">Open Record</a>
Kris Hunter	05/04/YYYY	05/07/YYYY	Submitted <a href="#">Open Record</a>
Jordan Wilson	06/07/YYYY	06/10/YYYY	Submitted <a href="#">Open Record</a>
Jordan Wilson	07/10/YYYY	03/01/YYYY	Submitted <a href="#">Open Record</a>

[Back to Employees](#)

FIGURE I.12: EPMS - Admin Latest Records

The screenshot shows the 'Performance Page - Aidan Smith' section of the EPMS Admin interface. At the top, there are navigation links: 'Manage Employees', 'Latest Work Records' (active), and 'Manage Projects'. The main content area is divided into two columns. The left column contains a 'Justification' box with text about Aidan's performance and a 'Grading' section with dropdown menus for satisfaction level (Satisfactory) and score (8). The right column contains a 'New Aims & Objectives' box with text about future goals and training requirements. At the bottom, there are 'Save' and 'Back' buttons.

(From 04/01/2015 To 04/05/2015)

**Justification:**

A very good quarter for Aidan. Completing all but one pieces of work. The new HR system must now be priority but overall, satisfactory performance. Keep it up.

**New Aims & Objectives:**

For the next quarter, I would like the HR system to be priority as this was an aim last quarter and was not completed correctly. Work alongside Kris in completing this. Please continue with the maintenance of all SQL databases and new mobile apps.

Please apply for following training courses:  
-Service Manager

**Grading:**

Satisfactory  8

FIGURE I.13: EPMS - Admin Create Performance Page

The screenshot shows the 'Manage Projects' section of the EPMS application. At the top, there are navigation links for 'Manage Employees', 'Latest Work Records', and 'Manage Projects'. A search bar is also present. The main area displays a table with the following data:

Name	IsActive	Team1	Team2	Team3	
Blackberry Enterprise	False	True	True	True	<a href="#">Edit</a>
SQL2012	True	False	True	False	<a href="#">Edit</a>
HR System	True	True	True	True	<a href="#">Edit</a>

FIGURE I.14: EPMS - Admin Manage Projects

The screenshot shows the 'My Work Records' section of the EPMS application. At the top, there are navigation links for 'Manage Employees', 'Latest Work Records', and 'Manage Projects'. The main area displays a table with the following data:

WorkRecordBegin	WorkRecordEnd	Status	
04/01/YYYY	04/04/YYYY	Satisfactory	<a href="#">Edit</a>   <a href="#">Delete</a>   <a href="#">View Record</a>
05/04/YYYY	05/07/YYYY	Satisfactory	<a href="#">Edit</a>   <a href="#">Delete</a>   <a href="#">View Record</a>
06/07/YYYY	06/10/YYYY	Unsatisfactory	<a href="#">Edit</a>   <a href="#">Delete</a>   <a href="#">View Record</a>
07/10/YYYY	07/01/YYYY	Submitted	<a href="#">Edit</a>   <a href="#">Delete</a>   <a href="#">View Record</a>

FIGURE I.15: EPMS - Employee Graded Records

The screenshot shows a web-based performance management system. At the top, there is a header bar with the text "[Domain]\Aidan My Work Records" and "Admin". Below the header, the title "Performance Page - Aidan Smith" is displayed. A blue horizontal bar follows, containing the text "(From 04/01/2015 To 04/05/2015)". Under this bar, there is a section titled "Justification:" which contains a paragraph about Aidan's performance. To the right of this, under the heading "New Aims & Objectives:", there is a text block with instructions for the next quarter. Below these sections, there is a "Grading:" section showing "Satisfactory 8" and a "Manager:" section showing "Richard Coleman". At the bottom left, there is a link labeled "Back".

[Domain]\Aidan My Work Records      Admin

Performance Page - Aidan Smith

(From 04/01/2015 To 04/05/2015)

**Justification:**  
A very good quarter for Aidan. Completing all but one pieces of work. The new HR system must now be priority but overall, satisfactory performance. Keep it up.

**New Aims & Objectives:**  
For the next quarter, I would like the HR system to be priority as this was an aim last quarter and was not completed correctly. Work alongside Kris in completing this. Please continue with the maintenance of all SQL databases and new mobile apps.

Please apply for following training courses:  
-Service Manager

**Grading:**  
Satisfactory 8

**Manager:**  
Richard Coleman

[Back](#)

FIGURE I.16: EPMS - Employee View Performance Page

## Appendix J

# Appendix J - Trakstar Email Trail

During background work and related research stages of the project contact was made to an American company Trakstar - a company who provide their own employee performance software. They were very helpful in the information they provided. This section details the email trail and conversations that took place.

Hi Aidan,  
 I hope the attachment is helpful.  
 Why you need a Performance Management System.  
 Glenn Martin  
 HR Performance Management  
[gmartin@trakstar.com](mailto:gmartin@trakstar.com)/[gmartin@promantek.com](mailto:gmartin@promantek.com)  
 719-395-9101 | Direct  
 303-995-8551 | Mobile  
 877.489.5651 | Technical Support Line  
[www.trakstar.com](http://www.trakstar.com)  
 Promantek DBA Trakstar  
 for-email-signature

Hi Aidan,  
 I hope this is helpful.  
 Glenn Martin  
 HR Performance Management  
[gmartin@trakstar.com](mailto:gmartin@trakstar.com)/[gmartin@promantek.com](mailto:gmartin@promantek.com)  
 719-395-9101 | Direct  
 303-995-8551 | Mobile  
 877.489.5651 | Technical Support Line  
[www.trakstar.com](http://www.trakstar.com)  
 Promantek DBA Trakstar  
 for-email-signature

Hello Aidan,  
 Thank your interest in our Trakstar software.  
 At Trakstar our mission is to revolutionize the world of employee evaluations by creating user-friendly yet powerful performance management tools.

We offer the most user-friendly, customizable and affordable product for our customers.

Today customers worldwide use Trakstar to streamline the performance and appraisal process.

We know what it takes to implement and tailor our solution to your business needs.

If you'd like we can schedule a time to explore Trakstar together and look at the areas that are important to you.

Please see the attached information about our products and services.

Glenn Martin

HR Performance Management

gmartin@trakstar.com/gmartin@promantek.com

719-395-9101 | Direct

303-995-8551 | Mobile

877.489.5651 | Technical Support Line

[www.trakstar.com](http://www.trakstar.com)

Promantek DBA Trakstar

for-email-signature

Hi Adrian,

I will send you some information.

You can go to our home page ([www.trakstar.com](http://www.trakstar.com)) and click Get and Instant Free Trial.

Glenn Martin

HR Performance Management

gmartin@trakstar.com/gmartin@promantek.com

719-395-9101 | Direct

303-995-8551 | Mobile

877.489.5651 | Technical Support Line

[www.trakstar.com](http://www.trakstar.com)

Promantek DBA Trakstar

for-email-signature

From: Aidan Smith [mailto:[aidan.smith1@hotmail.co.uk](mailto:aidan.smith1@hotmail.co.uk)] Sent: Wednesday, March 23, 2016 4:24 AM To: [hello@trakstar.com](mailto:hello@trakstar.com) Subject: Software Engineering Student

Dear Sir/Madam,

My name is Aidan Smith and I am a BSc Software Engineering student in the United Kingdom.

As part of my final year project, I have been designing an Employee Performance Management System. The aim of the project is to design a performance measuring system that could possibly be used in a live professional environment in the future. As part of my market research section, I have came across your software and it has been very beneficial in my report to talk about your system and it's features (all very good stuff). I would really like to reference you guys in my report and if anyone could have a look at my system and give me feedback on my system this would be great?

If this is the case I can provide the URL and a login for one of you guys to have a look?

I look forward to hearing from you guys.

Thanks,

Aidan

## Appendix K

### Appendix K - Trello board screenshot

Trello was used as a project management tool throughout to assist in the managing of the whole project. It was important to keep updating the trello board as agile methodologies were being used as much as possible. This section shows a screenshot of the trello board used.

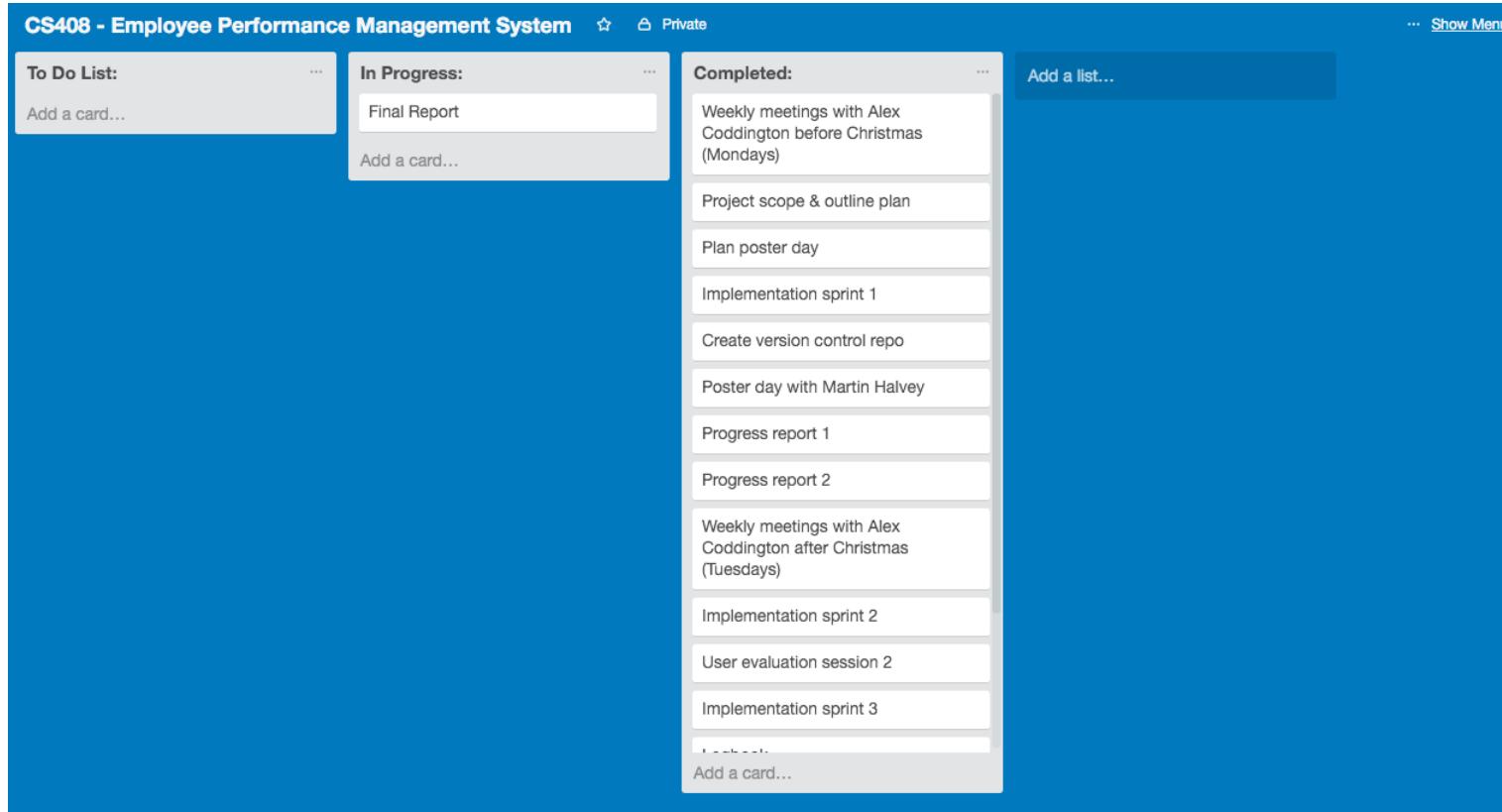


FIGURE K.1: EPMS - Trello Board

## Appendix L

# Appendix L - Selenium Automated Test Results

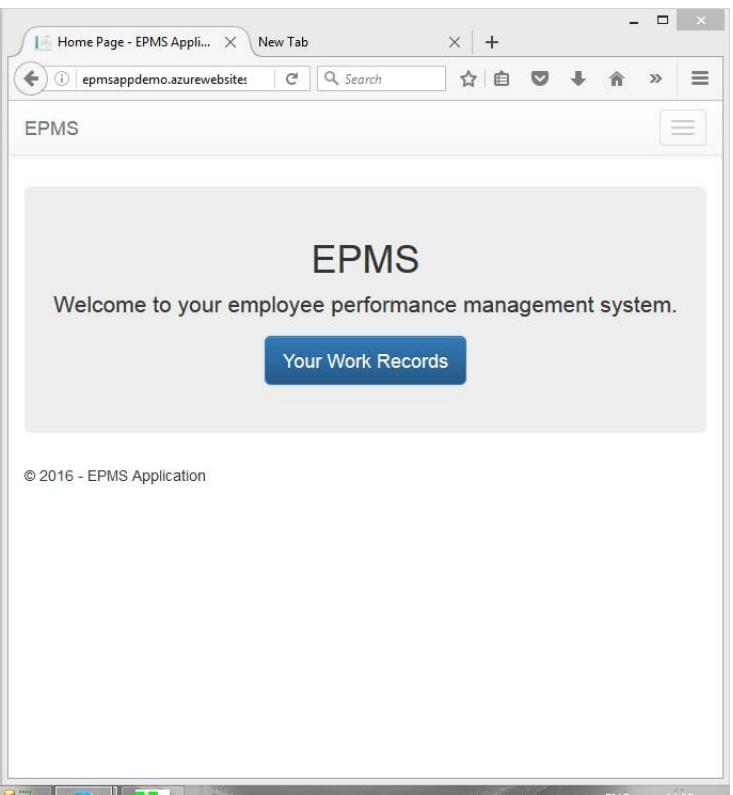
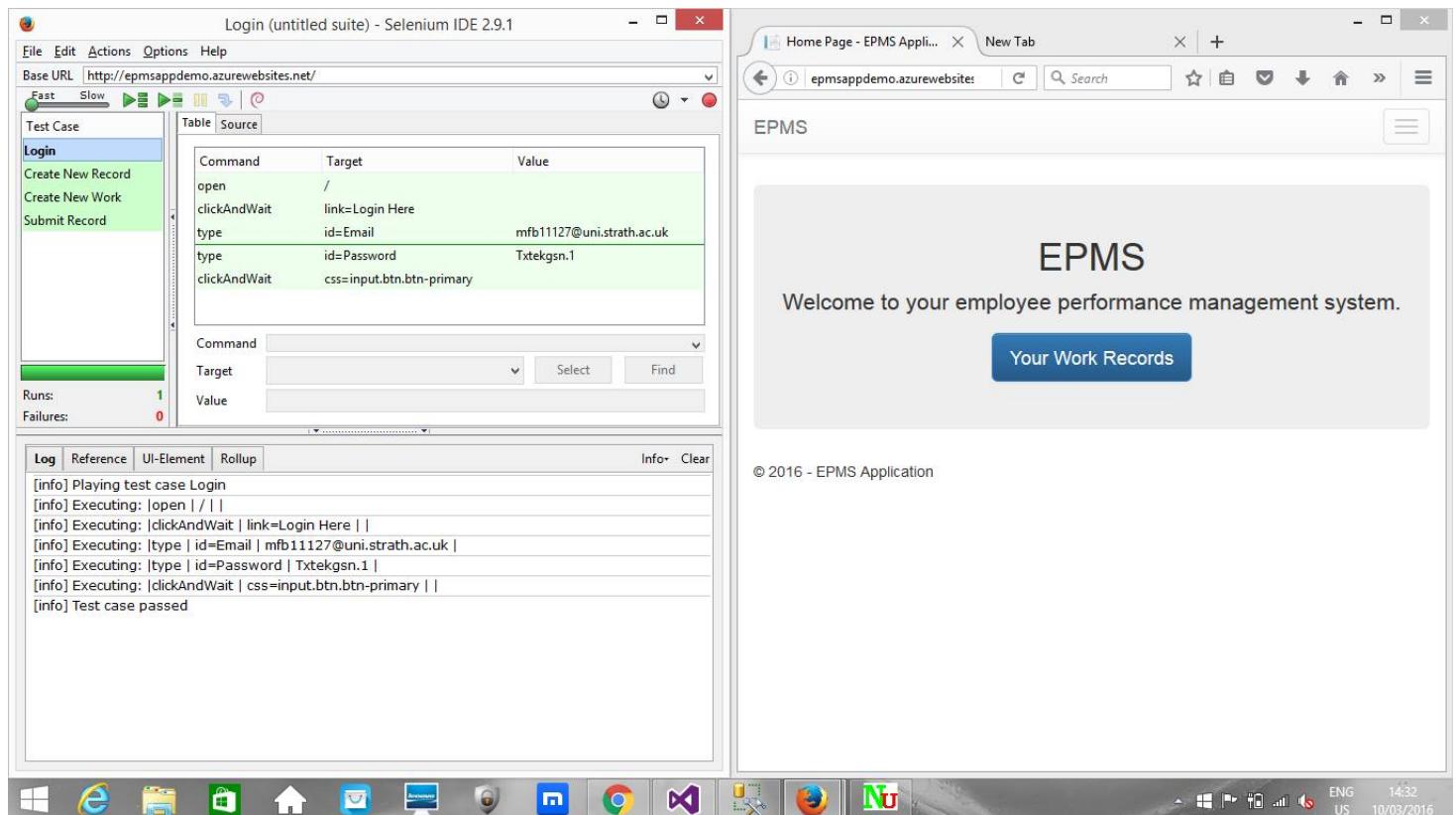


FIGURE L.1: Selenium - Login Test Case

The screenshot shows the Selenium IDE interface with the following details:

- Selenium IDE Window:**
  - File: Create New Record (untitled suite) - Selenium IDE 2.9.1
  - Base URL: http://epmsappdemo.azurewebsites.net/
  - Test Case: Create New Record
  - Log:
    - [info] Playing test case Create New Record
    - [info] Executing: |open| /Records| |
    - [info] Executing: |clickAndWait| link=Create Next Record| |
    - [info] Executing: |clickAndWait| css=input.btn.btn-primary| |
    - [info] Test case passed
- EPMS Application Window:**
  - Records - EPMS Application
  - Search: epmsappdemo.azurewebsites.net
  - EPMS
  - My Records
  - All Open Satisfactory Unsatisfactory
  - Show 10 entries
  - TimePeriodBegin: 2016-03-10, TimePeriodEnd: 2016-06-10, Status: Open, Work Created: 0
  - Create Next Record, Back To Home
  - © 2016 - EPMS Application

FIGURE L.2: Selenium - Create Record Test Case

The screenshot shows the Selenium IDE interface with the following details:

- Selenium IDE Window:**
  - File: Create New Work (untitled suite) - Selenium IDE 2.9.1
  - Base URL: http://epmsappdemo.azurewebsites.net/
  - Test Case: Create New Work
  - Log:
    - [info] Playing test case Create New Work
    - [info] Executing: |open| /Works/Index/35| |
    - [info] Executing: |clickAndWait| link=Create New| |
    - [info] Executing: |type| id=DateDue| 2016-03-10| |
    - [info] Executing: |type| id=DateCompleted| 2016-03-11| |
    - [info] Executing: |type| id=WorkItem| HR System| |
    - [info] Executing: |type| id=Description| Deploying and building a new system. Delivered a day late but overall, a successful project.| |
    - [info] Executing: |select| id=Work\_Project| label=HR System| |
    - [info] Executing: |select| id=Work\_Category| label=Project Work| |
    - [info] Executing: |clickAndWait| css=input.btn.btn-primary| |
    - [info] Test case passed

- EPMS Application Window:**
- Work - EPMS Application
- Search: epmsappdemo.azurewebsites.net
- EPMS
- Work (From 2016-03-10 To 2016-06-10)
- Create New
- Show 10 entries
- WorkItem Description DateCompleted Project Category Late
- HR System Deploying and building a new system. Delivered a day late but overall, a successful project. 2016-03-11 HR System Project Work Y
- Submit This Record, Back to Records
- © 2016 - EPMS Application

FIGURE L.3: Selenium - Create Work Test Case

The screenshot shows the Selenium IDE interface with a test case named "Submit Record". The test steps include opening a URL, clicking a link, and asserting a confirmation message. The log shows the execution of these steps. To the right, a browser window displays the EPMS application's "Work" page, showing a table of work items with one entry for "HR System" dated 2016-03-11.

WorkItem	Description	DateCompleted	Project	Category	Late
HR System	Deploying and building a new system. De...	2016-03-11	HR System	Project Work	Y

FIGURE L.4: Selenium - Submit Record Test Case

The screenshot shows the Selenium IDE interface with a test case named "Create Performance". The test steps involve navigating to a page, clicking a link, selecting dropdowns, entering text, and clicking a button. The log shows the execution of these steps. To the right, a browser window displays the EPMS application's "Records" page for "Aidan Smith", showing a table of performance records with one entry for the period 2016-03-10 to 2016-06-10.

TimePeriodBegin	TimePeriodEnd	Status	Work Created
2016-03-10	2016-06-10	Satisfactory	1

FIGURE L.5: Selenium - Create Performance Test Case  
(Managers)