



Technical Report

Graham Gallagher x12443558

Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name: Graham Gallagher
Student ID: x12443558
Supervisor: Frances Sheridan

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: _____

Date: _____

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

Complete the sections above and attach it to the front of one of the copies of your assignment,

What constitutes plagiarism or cheating?

The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a particular piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks

and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

Penalties for Plagiarism

If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend

- that a student's marks shall be reduced
- that the student be deemed not to have passed the assignment
- that other forms of assessment undertaken in that academic year by the same student be declared void
- that other examinations sat by the same student at the same sitting be declared void

Further penalties are also possible including

- suspending a student college for a specified time,
- expelling a student from college,
- prohibiting a student from sitting any examination or assessment.,
- the imposition of a fine and the requirement that a student to attend additional or other lectures or courses or undertake additional academic work.

Table of Contents

Executive Summary	7
Introduction.....	8
1.1 Background	8
1.2 Aims.....	9
1.3 Technologies Used	11
2 System	12
2.1 Requirements	12
2.1.1 Functional requirements	12
2.1.2 Data requirements.....	14
2.1.3 User requirements	15
2.1.4 Performance requirements.....	16
3 Technologies and Architecture	18
3.1 Laravel PHP.....	18
3.1.1 Models	19
3.1.2 Controllers	20
3.1.3 Views.....	24
3.1.4 Security.....	26
3.2 Database Architecture	27
3.3 Graphical User Interface (GUI) Layout.....	29
4 Testing.....	33
4.1 Unit Testing.....	33
4.1.1 Task Testing.....	34
4.2 User acceptance testing	37
4.2.1 Trunk Test	37
5 Conclusions	41

5.1	System Evolution	41
5.2	Further development or research	42
6	Appendix.....	44
6.1	Project Proposal	44
6.2	Monthly Journals	52

Executive Summary

The main objective of JobFlow is to provide a communication and ticketing system for different departments within a large organisation with the end goal of ultimately replacing phone and email workflows with a simple to use web portal.

End users are able to raise requests via the JobFlow portal and workers are able to respond to tickets raised and plan their work better as well as management being able to set and monitor key metrics on the performance of the department using information gathered from the system.

This report outlines the requirements, the design implementation and testing of the system and the future uses and evolution of the system over time.

"End Users" – These are the people who will be raising the tickets

"Workers" – These are the people who will be accepting tickets and completing the work

"Administrators" – These are the supervisors and management levels

"IT Support" – These are the support staff that will provide maintenance and support for the system

Introduction

The purpose of this report is to outline the background and reasoning behind creating this system, the requirements gathered and the technology behind it. It will also outline the testing requirements and potential growth and evolution of the system over time.

This project will be discussed in the context of a case study.

This case study is the painters workshop within a maintenance department in a large organisation which is currently facing a problem keeping track of requests by end users.

Currently, end users (office staff, departmental staff) need to raise requests for painting and decorating work through calling the workshop, emailing the operations manager or by visiting the workshop directly.

JobFlow seeks to resolve this by providing a one stop shop for raising requests through a dynamic web portal.

1.1 Background

Within the organisation to raise an issue with the Maintenance department you needed to either contact a worker directly by speaking to them or raise the issue by phone or via email.

This makes it difficult to track the progress of requests for users and for workers to prioritize effectively as job requests are coming through many different channels and are recorded on paper or in various locations and documents.

This inefficiency has become most prominent in the painter's workshop.

The painter's workshop provides many services. They provide signage, design as well as painting and decorating. End users who raise the requests are mostly unaware of the services available to them and are unsure of how to get in touch with the workshop.

The workshop consists of full time and contracted workers who are assigned tasks based on requests raised throughout the previous day. These requests are given to the secretary via phone or email who then compiles them into a word document and the supervisor must then prioritise and categorise the jobs correctly before delegating them to the painters and decorators. These requests are usually provided without a specified format and typically have missing information or contact details.

Once these tasks have been delegated the end users who raised the request are not effectively notified and as their request is not given a unique identification code it becomes difficult for them to track. This leads to a situation when the painters do arrive the room is not ready to be painted or it is not a convenient time as frequently contact details are not sought or are incorrect.

Workers are also unclear on the workload and requests that the workshop currently has. Workers are only able to read their list of work to be carried out during the day and cannot see the overall workload for the workshop. They are also unaware of the last time a request was raised for the area they are visiting leading to unnecessary repeat work.

All of this has led to drastic inefficiencies and frustration across the board. This lack of transparency, automation and accountability is exactly what JobFlow aims to alleviate.

1.2 Aims

JobFlow seeks to resolve the problems that have been discussed in the previous section. This system will benefit users all through the organisation.

A simplistic portal for users

JobFlow provides a portal for users to simply raise a ticket request. They provide their email address and password, log their contact details and the department they are a part of and summarise their request.

They are able to check the progress of their ticket at any time by quoting the ticket ID or by checking the status on JobFlow itself.

A workable environment for the painters

Workers are able to see a general overview of the workload they currently have through the number of tickets open. They are able to get their requests through a simple system and add notes as needed.

Usable data for administrators

Administrators and supervisors are able to check the efficiency of the workshop by checking the number of open tickets, the number of tickets by department and having a centralized database of tickets to allow data mining as needed.

The aim of JobFlow is to bring the effectiveness of technology to businesses in order provide a solid workflow solution overtaking inefficient processes to bring benefits to a businesses overall efficiency and effectiveness.

1.3 Technologies Used

Pluralsight

Pluralsight has been used as a basis for understanding and training on technologies on all aspects of the project in order to aid development.

Laravel

JobFlow will be primarily based on the MVC Laravel PHP framework using atom editor for development.

Microsoft SQL

I will be using a Microsoft SQL server in order to hold user and worker information for the system. Once in production the database will be converted to PostgreSQL to adapt easily to Heroku deployment structure.

Google Charts

Google charts will be used to provide pre-determined insights into the system for management levels and the system admins.

Python

Python will be used to develop data analytics for management and system admins using information gathered from users and workers.

Bootstrap/Bootstrap Sass

The web application utilizes the Bootstrap framework to provide a responsive contextual front end design.

Heroku

Heroku was used to deploy the application online to its cloud platform allowing independent access from the development device.

2 System

2.1 Requirements

From the first round functional requirements the requirements for the JobFlow system have evolved somewhat since the original requirements spec was developed.

2.1.1 Functional requirements

1. Login
 - 1.1 Users must be able to log in with their existing work email credentials
2. Creating new requests
 - 2.1 - Must require essential information
 - 2.2 - Must allow for the upload of files as needed e.g an image
 - 2.3 - Users must select the correct category their issue falls under
3. Worker view
 - 3.1 - Workers must be able to view all of the fields which the user has filled in on the ticket
 - 3.2 - Workers must be able to edit details of a ticket
 - 3.3 - Workers must be able to add notes to a ticket at any stage
 - 3.4 - Any modification to a ticket must be recorded as a change and the resource which changed it must be captured "e.g John edited ticket @ 5.00pm 12-12-16"
4. Reporting
 - 4.1 - The number of tickets raised at any given time must be recorded for query by system administrators and management at any specified time field.
 - 4.2 - The performance of a worker must be able to be monitored e.g. the number of tickets assigned, number of tickets closed
 - 4.3 - Average ticket resolution time by worker

- 4.4 - Number of requests by department
- 4.5 - Number of requests by category
- 5. End users
 - 5.1 - End users must fill in their contact details upon raising a ticket
 - 5.2 - The system should allow for a notification to the end user once an update or change has been made on the ticket they raised through email or otherwise
Fields must be clear to understand
 - 5.3 - Users must be shown a warning if they haven't filled the required fields
 - 5.4 - Users must be presented with a message once the request is raised e.g "ticket created successfully"
- 6. User Management
 - 6.1 - End users and workers must be able to register themselves on the system using their work email.
 - 6.2 - Users and workers must be able to be edited removed and added through admin access to the JobFlow system.

Non-Functional Requirements

- 1. Instant Messaging
 - 1.1 – The system should allow for end users to be able to query a ticket they raised with a worker if they are online by quoting their ticket number.
 - 1.2 – Workers should be sent an email if there is no worker agents online when an end user tries to contact them via the instant messaging function.

2.1.2 *Data requirements*

In order for the JobFlow system to work correctly it is important that it captures information that can be used in order to effectively complete a task.

The data requirements would be best explained by dividing it into the basis of the users that will be interacting with the system

End users:

Required Fields:

Work email:

Contact Number:

Description:

Attached item: (as required)

From feedback gathered from the interviews and questionnaires it is clear that one of the most prominent problems for workers is that they do not receive enough information from end users prior to creating a ticket.

The previous system which allowed the ability to raise tickets but did not take into account the workers or end users preferred data requirements.

Workers:

Workers should be required to change the ticket status once it has been viewed. One of the issues facing the previous system was that workers could open a ticket, read it and close it without taking any action. This system must require the worker to action a ticket once it has been opened and to add a note regarding that ticket raised e.g "reviewed ticket will add scheduled time today".

Workers should also be required to register their work email address and contact details to the system. This is important as ensures accountability and transparency as well as leaving an avenue open for administrators and end users to contact the person responsible for the ticket

"Resource Assigned"

The system must have a populated list of current working employees. This list will be known as the "Resource" list.

When a worker is assigned to a ticket they become the "assigned resource".

Administrators:

Administrators should have the most open ended access to the system. They must be allowed to add, remove and edit users.

IT support:

IT support should be able to provide comprehensive support to the system and should have front and backend access to the system.

IT support must also be able to add remove and edit users and workers.

2.1.3 *User requirements*

End Users:

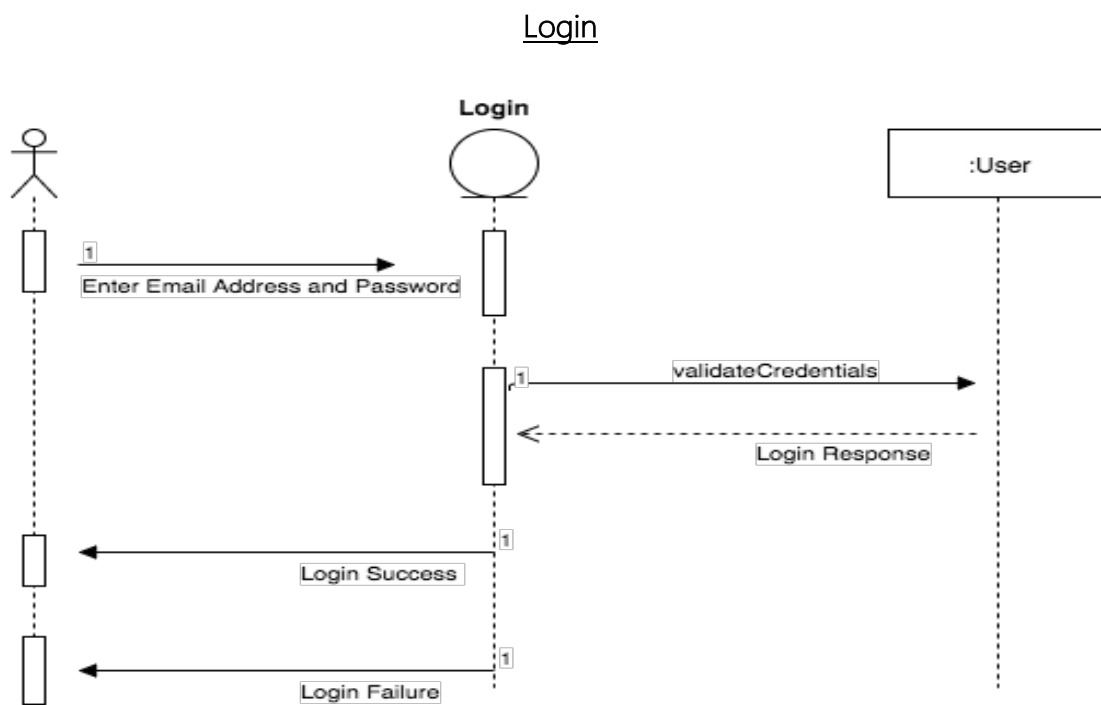
- Users must be registered for a work email address.
- Users must have access to a device with internet access.
- Users must register or be registered on JobFlow.

2.1.4 Performance requirements

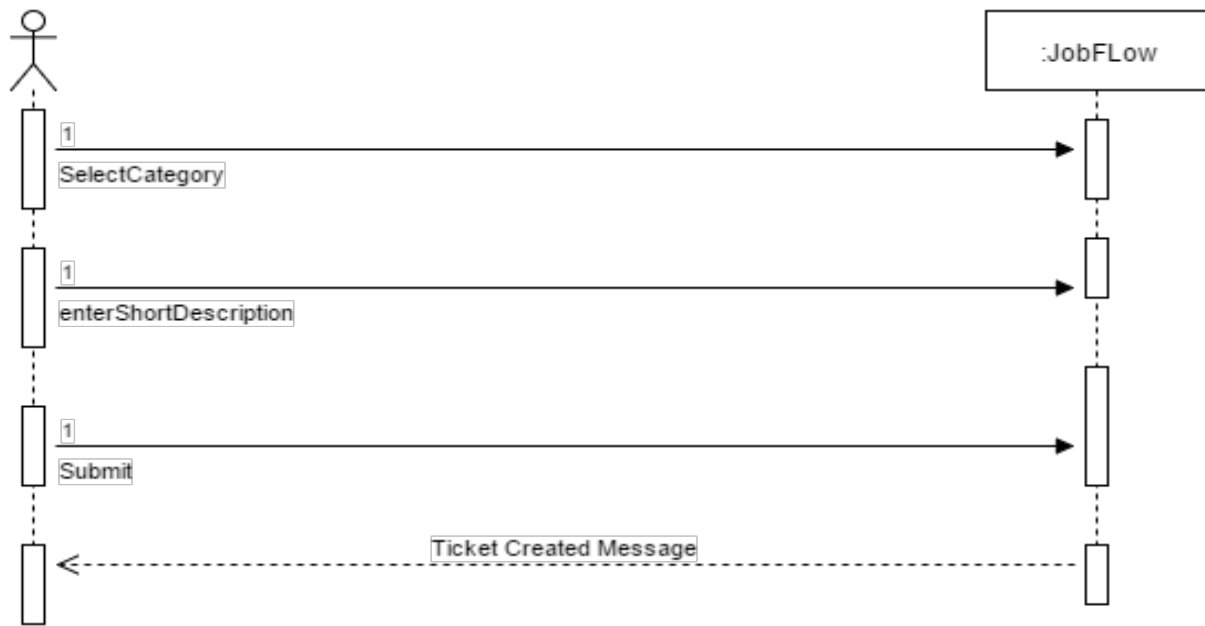
It is essential that JobFlow is easy to use by all actors and that it requires a minimal amount of training.

- JobFlow must create a ticket instantly once the create ticket button is pressed without delay.
- Tickets must update instantly for workers without delay.

Sequence Diagrams



Submit New Ticket



3 Technologies and Architecture



Figure 1 Overview of Technologies

3.1 Laravel PHP

The primary technology of JobFlow is Laravel. It is a PHP based MVC framework which consists of Models Views and Controllers. Laravel was chosen as it is " *accessible, yet powerful, providing powerful tools needed for large, robust applications. A superb inversion of control container, expressive migration system, and tightly integrated unit testing support*". (Laravel, 2017). [1]

Using an MVC framework allows for the greatest flexibility required for a system like this which will handle different types of data and different relations to and from that data.

3.1.1 Models

Laravel comes with Eloquent ORM ActiveRecord implementation which allows simpler interaction with database tables. In Laravel models act as a driver for tables which can be retrieved updated, added and removed without needing to change the database configuration itself.

JobFlow has a number of models which serve to query a table and create relations in the migrations files.

User Model

The user model was created to serve the User table. The user model contains a number of relations. These relations link the different tables and creates the relationship types between these tables.

The user table relations are:

User has many tickets

```
Public function tickets()  
{ return $this->hasMany(App\Ticket');
```

User has many owns

```
Public function owns()  
{return $this->hasMany('App\Own');
```

Laravel 5 Entrust was then used to generate user Role logic.

```
Use EntrustUserTrait;
```

Ticket Model

The ticket model is used to serve the "ticket" table and holds the relations between the users and ownership tables.

Ticket belongs to user

```
Public function user(){  
Return $this->belongsTo('App\User')
```

Ticket is "owned" by an assigned user

```
Public function own(){  
Return $this->belongsTo('App\Own');
```

Own Model

This model is used to serve the "owns" table and holds relationships between users and tickets.

A ownership belongs to a user

```
Public function user(){  
Return $this->belongsTo('App\User')
```

An ownership belongs to a ticket

```
Public function ticket(){  
Return $this->belongsTo('App\Ticket')
```

3.1.2 Controllers

User Controller

The user controller is used to handle the logic behind how a user is handled in the database table and how their information is processed.

The user controller consists of the following functions:

postSignUp – Sign up function

This function is used to firstly validate users input requiring that email addresses must be unique, first name is required and a minimum of three characters, the password is required and must be a minimum of 8 characters and that the user must select their relevant department

for registration. It then passes this information to the user table in the database with the corresponding column.

Once this information is passed on the user is redirected to the dashboard route and is considered a logged in user.

```
public function postSignUp(Request $request)
{
    #This is the validation for the registration of a user. If any of these are not provided the message blade will be
    #returned with an error
    $this->validate($request, [
        'email' => 'required|email|unique:users',
        'first_name' => 'required|min:3',
        'password' => 'required|min:8',
        'department' => 'required',
    ]);
    #This section tells laravel take these columns out of the database because we want to work with them
    $email = $request['email'];
    $first_name = $request['first_name'];
    #Passwords are encrypted with bcrypt before being passed to database.
    $password= bcrypt($request['password']);
    $department = $request['department'];
    $phone_number = $request['phone_number'];

    # The columns are retrieved and we can now tell the database consider the above logic a new user
    $user = new User();
    $user->email = $email;
    $user->first_name = $first_name;
    $user->password = $password;
    $user->department = $department;
    $user->phone_number = $phone_number;

    $user->save();
    # If the credentials are filled in correctly, save the new user and consider them logged in and return them to the
    #dashboard
    Auth::login($user);

    return redirect()->route('dashboard');
}
```

Figure 2 User Sign Up Function

Most of the functions operate in a similar way and the code can be found in the User controller (App/Http/Controllers/UserController.php).

The other functions within the User Controller are:

postSignIn

This handles user sign in requests from an already registered email address returning them to the dashboard on successful sign in.

getLogout

This handles logout requests, expires the user token and redirects the user to the registration/login page

getAccount

This function retrieves only the current users details when they click on the my account tab.

postSaveAccount

This function is used for updating user details in the my account page. It handles requests for updating information and requires the user to have a non-null first name for the form.

It also handles the saving of the profile picture images and instructs Laravel to save them in local storage with the name "first_name-id.jpeg". This makes for much easier management of images going forward.

getUserImage

This function is used to handle profile picture updates and instructs the user table to remove the old profile picture from storage when a new one is uploaded preventing conflicts and saving on storage.

Ticket Controller

The ticket controller is one of the most important controllers next to the user controller as it holds most of the logic behind tickets in the JobFlow system.

The ticket controller consists of the following functions:

getDashboard

This function returns all of the user tickets to the dashboard view and orders them by descending order by date of creation.

ticketCreateTicket

This function contains the logic for creating a new ticket. Similar to the user sign up function it has validation protection which returns an error if any field in the ticket form is left blank or the body goes above 1000 characters. It then prepares the database by telling it these fields correspond to a column in the table. It also holds the error message if there is an issue with the request the error message blade is fired and the user is returned a red contextual error message box with the error causing fields in red.

If the Ticket is created successfully a success message is passed back to the dashboard view and a Green contextual success message block appears at the top of the page advising the user the creation was successful.

ticketEditTicket

This section holds the logic for editing a ticket. This allows the user to edit the body of the text only, passing a message back using JSON to the view a success message on successful editing.

getDeleteTicket

This function allows users to delete tickets and upon deletion they are returned to the dashboard with a success message which pops up in a red contextual box in the view to let the user know the ticket has been deleted successfully.

Home Controller

The home controller is used to route the index page and to handle middleware for the index.

The home controller consists of the following functions:

The function to call the Auth middleware for the form

```
Public function _construct(){
```

```
$this->middleware('auth');

}
```

The function to set the index as the Home view

```
Public function index(){

    Return view('home');

}
```

3.1.3 Views

The views are the pages the user sees and interacts with on the front end. These files contain the calls for the controllers to perform certain tasks as well as styling and bootstrap components.

There are a number of views in JobFlow they are:

Dashboard View:

The dashboard view holds the logic for the form presented to the user as well as the ticket stream below the new ticket form.

```
@foreach($tickets as $ticket)
<div class="col-md-4">
<article class="ticket">
    <div class="panel panel-primary">
        <div class="panel-heading"><h4 align="left">TICKET #{{ $ticket->id}}</h4><h3 align="center">{{ $ticket->title}}</h3></div>
        <div class="panel-body"><strong>Ticket ID </strong><h4>{{ $ticket->id}}</h4></div>
        <div class="panel-body"><strong>Department</strong><h4>{{ $ticket->user->department}}</h4></div>
        <div class="panel-body"><strong> Phone Number </strong><h4>{{ $ticket->user->phone_number}}</h4></div>
        <div class="panel-body"><strong>Alternative Contact Number:</strong> {{ $ticket->contact}}</div>
        <div class="panel-body"><strong>Desscription: </strong>{{ $ticket->body}}</div>
        <div class="info">
        <div class="panel-body"> <strong> Raised by</strong> {{ $ticket->user->first_name}} <br></div><strong>Created </strong> {{ $ticket->created_at}}</div>
        <div class="panel-body"> <strong> Email </strong> <a href="mailto: {{ $ticket->user->email}} " > {{ $ticket->user->email}}</a> </div>
    </div>
```

From the code above you can see ticket info can be called and used within the view. The ticket stream also uses Bootstrap styling to lay out the tickets correctly and style them using the

“panel” component. Users information is called in the action div to allow other users to get in touch with the user who raised the ticket.

The Dashboard view is explained in more detail in *2.3 Graphical User Interface (GUI) layout* section.

Account View:

The account view is used to display information for the users contact details and department. Like the dashboard view it pulls in user information from the users table specific to the current user that’s logged in.

The Account view is explained in more detail in *2.3 Graphical User Interface (GUI) layout* section.

About View:

The about view is used to explain the breakdown of how JobFlow works and how to raise a ticket. It’s a simple HTML page with bootstrap styling to explain to users how to raise a ticket, edit their account details, edit tickets and delete them.

My Tickets View:

The My Tickets view is a menu item in the nav bar. This view iterates through all the tickets raised by the user and returns them in the view. Users can then action the tickets they have raised.

3.1.4 Security

Route Protection

All of the routes passed the index route are protected from being accessed by guest users. If a user attempts to access a page without first logging in or register they are returned to the index page.

Bcrypt Encryption

The Bcrypt middleware was used to protect passwords from being saved raw onto the database. When a user submits a password on the registration form the password they have chosen is first encrypted before being passed to the database.

```
$password= bcrypt($request['password']);
```

Figure 3 bcrypt in User Controller

first_name	password
Tony	\$2y\$10\$M.ifihWPRRaEy/x/YMExEeBA5UEWpFYRdZMIAYIODd3...

Figure 4 Encrypted password in Database

Validation & Timeouts

All fields are validated to ensure that they are passed to the database correctly. User registration and sign in requests are also limited. If a user attempts to log in with incorrect credentials 5 times they are locked out from logging in to the system for 60 seconds.

3.2 Database Architecture

The database is SQL based and is constructed using Laravel migrations to create the table schema.

The migrations are created using the command:

```
php artisan make:migration create_name_table
```

This creates a skeleton for the table file which you then add in the elements for the table. The

following is a diagram of the tables in the JobFlow database and how they relate to each other.

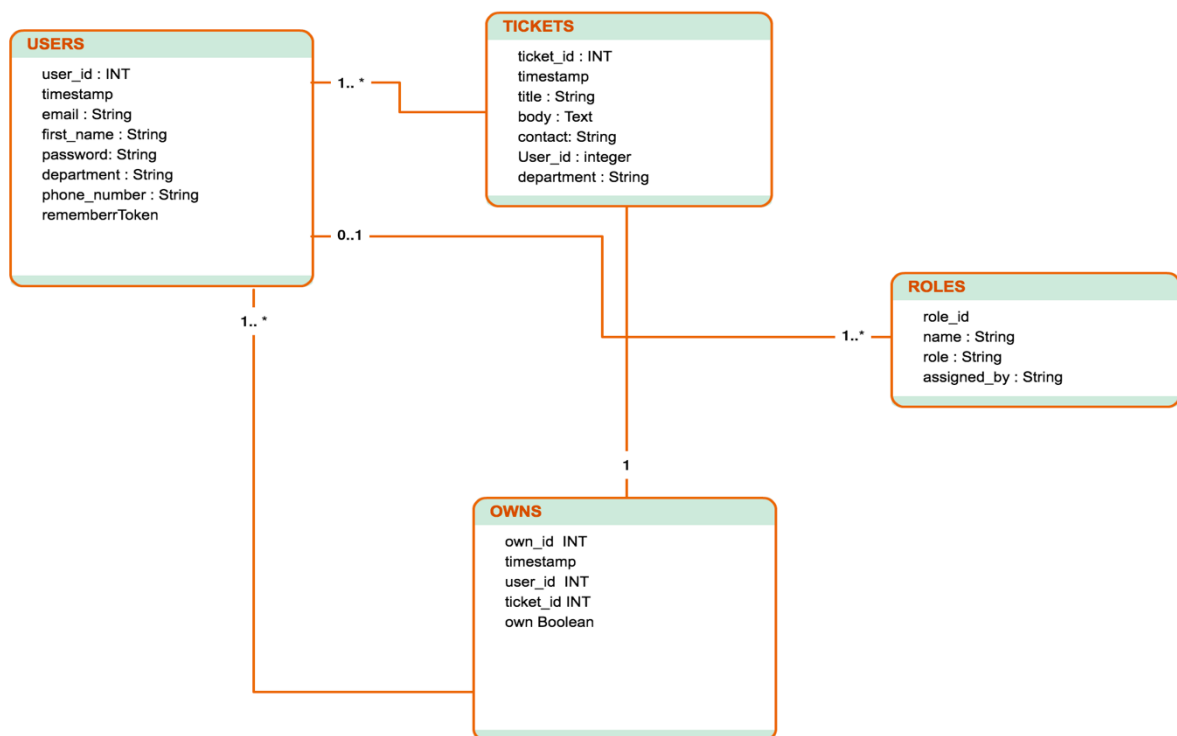


Figure 5 Database Architecture

DFD - Login

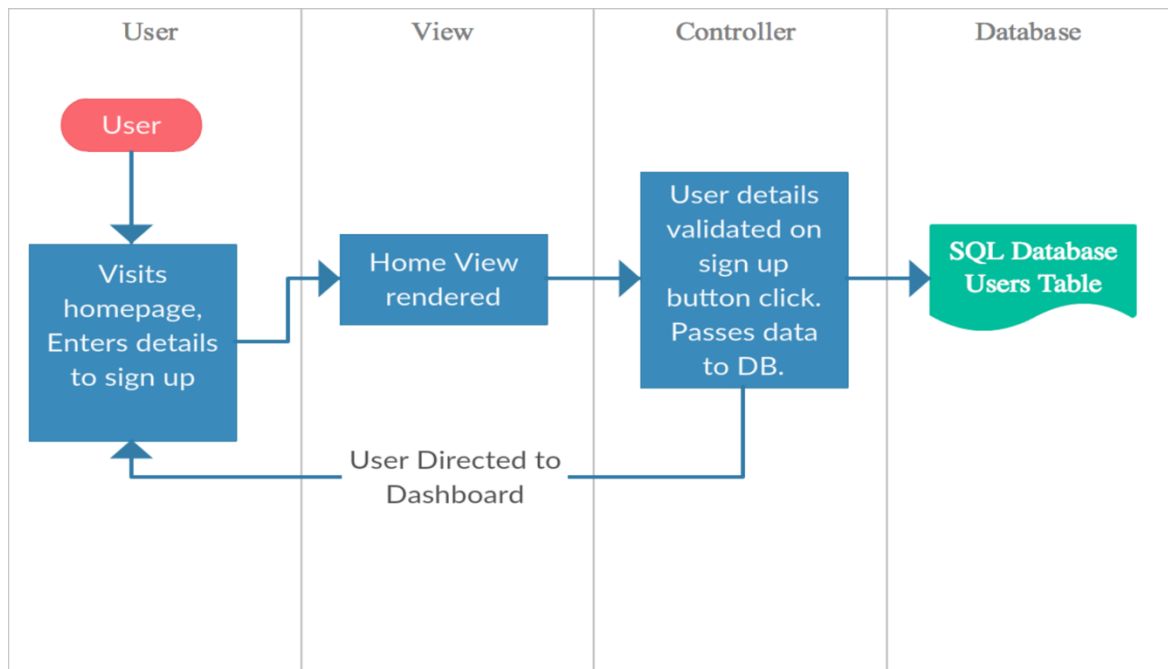


Figure 6 Login DFD

New Ticket Submission

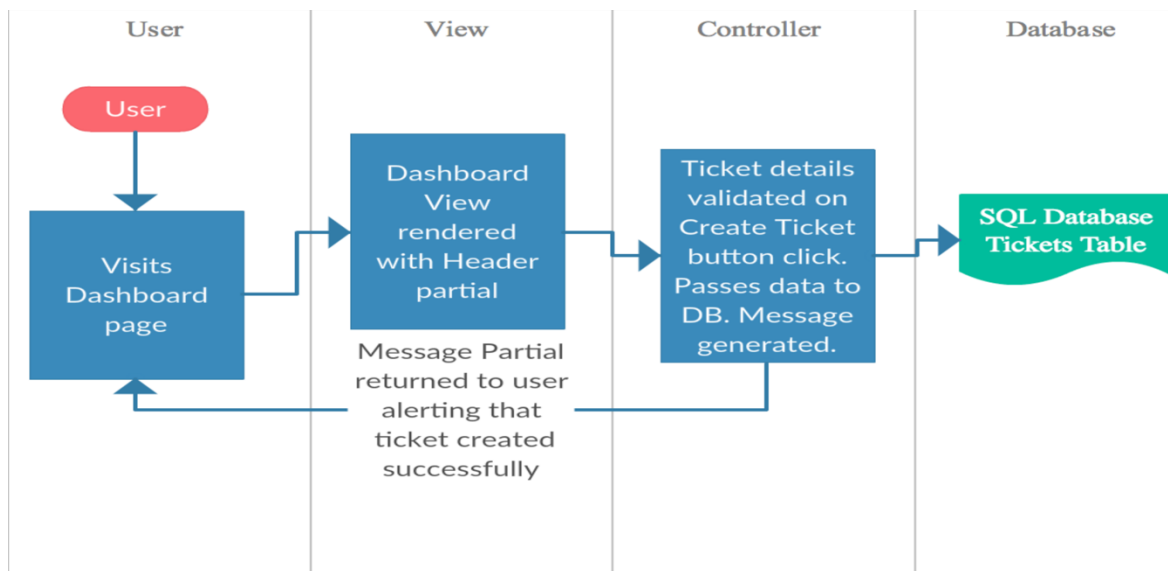


Figure 7 New Ticket Creation DFD

3.3 Graphical User Interface (GUI) Layout

The GUI of JobFlow was designed with simplicity in mind.

Throughout the site the green and blue of the logo are used to keep the site uniform throughout and to allow users to focus on important elements.

The GUI was created using HTML, Javascript and Bootstrap.

Login Screen

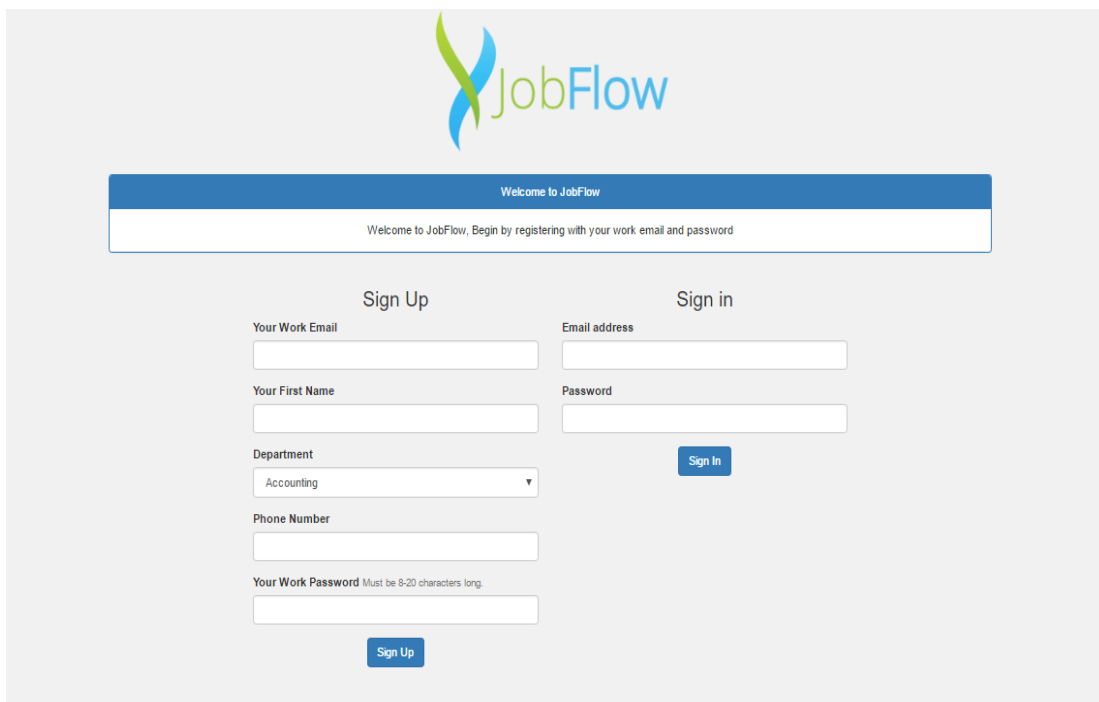
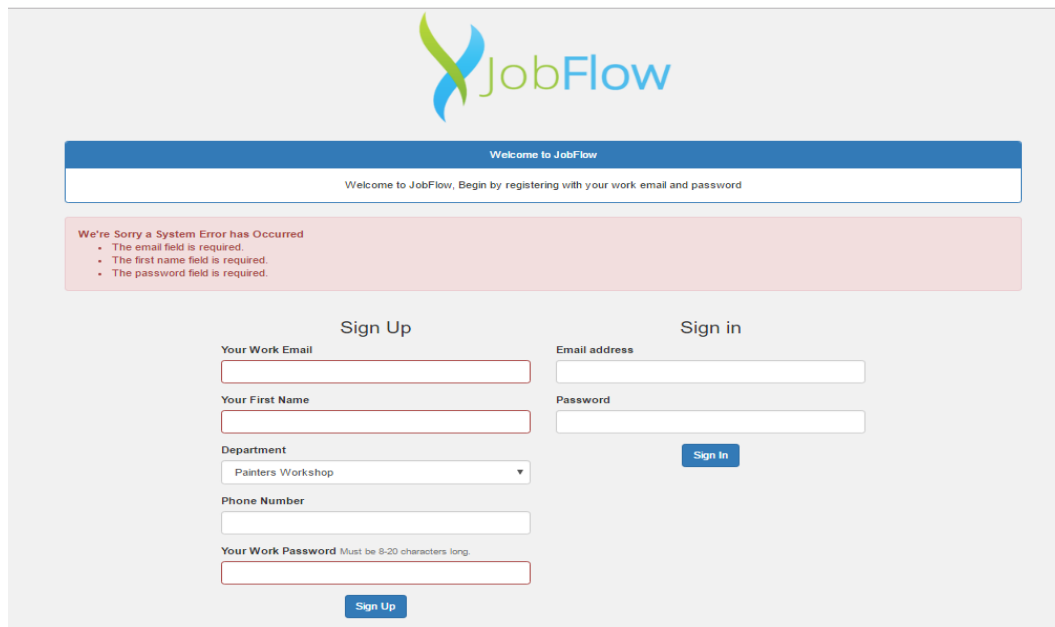


Figure 8 Login/Register Screen

This is the screen that will be presented to all users once they open the JobFlow system. It is simplistic with clean lines and prominent buttons.

The login page utilizes Bootstrap gridding and styling elements.

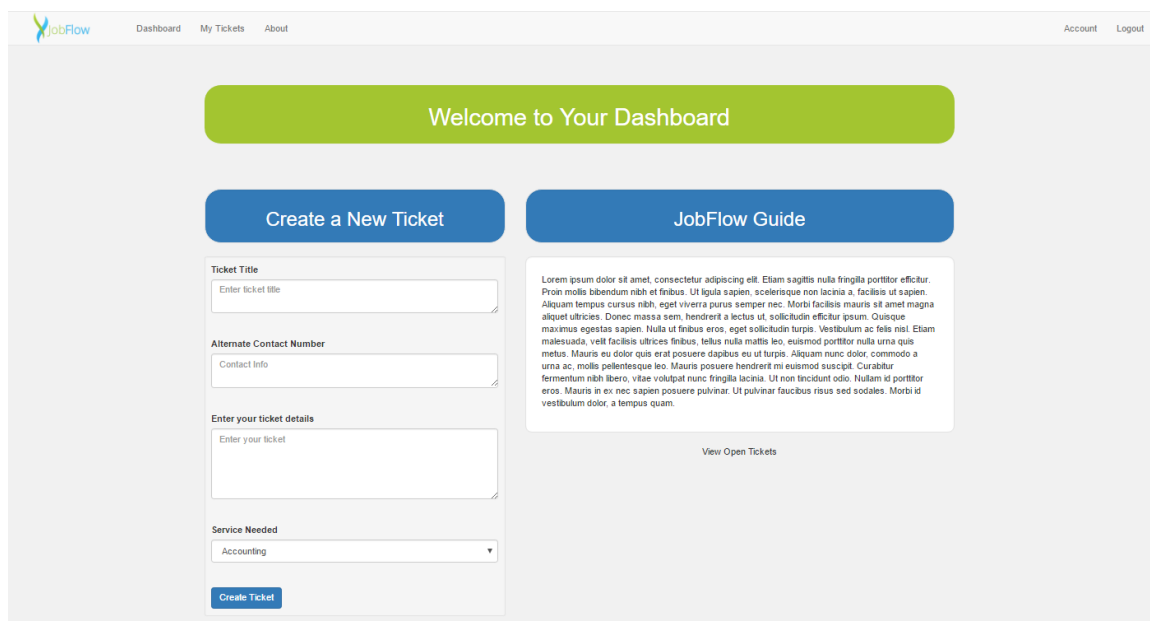
If a user enters incorrect information the message blade is called to return a red contextual error message box. In addition the field which is causing the error is highlighted in red.



The image shows the JobFlow login/register screen. At the top is the JobFlow logo. Below it is a blue banner with the text "Welcome to JobFlow". Underneath is a white box with the text "Welcome to JobFlow, Begin by registering with your work email and password". A red error message box is displayed, stating "We're Sorry a System Error has Occurred" and listing three bullet points: "The email field is required.", "The first name field is required.", and "The password field is required." Below the error message are two columns of form fields. The left column is for "Sign Up" and includes fields for "Your Work Email", "Your First Name", "Department" (a dropdown menu with "Painters Workshop" selected), "Phone Number", and "Your Work Password" (with a note "Must be 8-20 characters long"). The right column is for "Sign in" and includes fields for "Email address" and "Password". Both columns have a "Sign In" button at the bottom.

Figure 9 Login/Register Screen with error message

User Dashboard



The image shows the JobFlow user dashboard. At the top is the JobFlow logo and navigation links: "Dashboard", "My Tickets", and "About". On the right side, there are links for "Account" and "Logout". A green banner at the top of the main content area says "Welcome to Your Dashboard". Below this are two blue buttons: "Create a New Ticket" and "JobFlow Guide". The "Create a New Ticket" button is active and leads to a form. The form has four sections: "Ticket Title" with a text input field, "Alternate Contact Number" with a text input field, "Enter your ticket details" with a text input field, and "Service Needed" with a dropdown menu showing "Accounting". A "Create Ticket" button is at the bottom of the form. To the right of the form is a white box containing placeholder text (Lorem ipsum) and a "View Open Tickets" button at the bottom.

Figure 10 User Dashboard

This is the first screen end users will see when they log in or click on new ticket. Here they must select their category of pre-defined fields and describe their issue. They then fill in an alternate contact number enter clicks Create Ticket.

They can view their raised tickets by clicking “My Tickets” or viewing the ticket stream

The screenshot displays the 'Ticket Stream' interface with three tickets listed side-by-side. Each ticket card includes a title, Ticket ID, Department, Phone Number, Alternative Contact Number, Description, Raised by, Created timestamp, and Email. At the bottom of each card are 'Add Note' and 'Assign to Me' buttons. The footer contains 'Terms of Service | Privacy' and '© 2017 JobFlow Project x12443558'.

Ticket #4	Ticket #3	Ticket #2
Need report for Wages	New Ticket	Door needs to be repainted
Ticket ID 4	Ticket ID 3	Ticket ID 2
Department Admin Office	Department Accounting	Department Accounting
Phone Number 01 801 8109000	Phone Number 01 801 810 810	Phone Number 089 909 50 59
Alternative Contact Number: 081 9020203432	Alternative Contact Number: 01	Alternative Contact Number: 01 808 808 808
Desscription: Need a wages report for January	Desscription: This is a test ticket	Desscription: Main door is marked badly and needs to be repainted
Raised by Robert	Raised by Erika	Raised by Test
Created 2017-05-06 13:34:01	Created 2017-05-05 17:58:05	Created 2017-05-05 16:58:46
Email robert@work.ie	Email erika@work.ie	Email test@test.ie
Add Note Assign to Me	Add Note Assign to Me	Add Note Assign to Me

Terms of Service | Privacy © 2017 JobFlow Project x12443558

Figure 11 Ticket Stream

The Ticket Stream

The ticket stream allows users to see the currently raised tickets quickly and easily and take action as needed. Users cannot edit or delete tickets unless they own the ticket and cannot close tickets unless they own the ticket.

JobFlow

Dashboard My Tickets About

Account Logout

Your Account

First Name

Ronan

Department

Accounting

Phone Number

01 801 801 810

Profile Picture (only .jpg)

Choose File No file chosen

Save Account

Figure 12 User Account Page

Account Page

The account page allows users to change their details say for example a new phone number or if they change their name after marriage. The department field is read only as only an administrator can change an employee's department.

Beneath the form there is a "Profile Preview" which allows the user to see the profile picture they uploaded and a brief card of their details as it is on their User account.

4 Testing

Testing of the JobFlow system was be conducted in two ways. Firstly, Functional evaluation testing and secondly user evaluation testing.

4.1 Unit Testing

Unit testing was performed during the development of JobFlow and are run whenever a significant new feature or change in styling is added.

The JobFlow system utilizes Laravels inbuilt testing capabilities.

These tests are created using the command:

php artisan make:test TestName

This generates a test skeleton for you to add your own testing logic which adapts to the PHPUnit testing element of Laravel. These tests serve as the unit testing for the system.

```
<?php

use Illuminate\Foundation\Testing\WithoutMiddleware;
use Illuminate\Foundation\Testing\DatabaseMigrations;
use Illuminate\Foundation\Testing\DatabaseTransactions;

class SignupTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */
    public function testExample()
    {
        $this->visit('/')
        ->type('testaccount@work.ie','email')
        ->type('test','first_name')
        ->select('accounting','department')
        ->type('011111111','phone_number')
        ->type('password01','password')
        ->press('Sign Up')
        ->seePageIs('/dashboard');
    }
}
```

Figure 13 User Registration Test

The testing files can be found under the “tests” folder.

Manual testing was also conducted based on a number of tasks. The tasks have been recorded in the following table with the expected result and actual result.

4.1.1 *Task Testing*

Task		Expected Outcome	Actual Outcome	Success/Failure
1	Sign up using Email address and password	User enters 8 character password. Once sign up button is clicked user is directed to their dashboard.	User signs up with unique email. User returned to dashboard on clicking sign up	Successful
2	User signs up with already registered email address	User is shown a red warning context dialog box to say email is not unique.	Error message displayed and email account not added to table.	Successful
3	User enters a blank password or too short password	User is presented with red warning contextual dialog box.	Error message displayed. User information not captured in database.	Successful
4	User signs in with registered email and password	User is returned to their dashboard	User directed to their Dashboard	Successful

5	User Creates a ticket using field.	Ticket created and present in database. Green success contextual dialog box appears to notify ticket created.	Ticket appears in database with user ID, Green "Ticket Created" dialog box appeared on top of dashboard page.	Successful
6	User leaves ticket title or body blank	User is returned an error and information is not captured in DB	User is returned red contextual error message. Input not recorded in DB.	Success
7	Check newly created ticket in ticket view	Ticket number, title, body and users contact number, name, ticket created at date and department displayed on ticket view	All fields present.	Success
8	User edits ticket in ticket view	User is displayed a modal box which they can enter the update in the body text. On save the ticket updates	Ticket body is updated on clicking "save changes"	Success

9	User closes ticket	User clicks "close ticket" message displayed to indicate ticket deleted, ticket removed from ticket stream and DB	Ticket removed from database, red context message displayed indicating ticket deletion. Ticket no longer appears in stream	Success
10	User clicks JobFlow logo	User is returned to their dashboard.	User returned to dashboard.	Success
11	User clicks account	User is displayed their respective information from their user record (name (read only), phone number, dept).	Information displayed correctly.	Success
12	User Uploads profile picture	User clicks upload file. User uploads JPEG image.	Image appears however might be useful to add in message for dimensions so it appears better.	Success (Add in comment to profile picture for dimensions)

13	User clicks logout tab on menu	User is directed to login page and cannot access /dashboard URL	User is logged out and returned to login page. User is returned to login page when /dashboard URL entered	Success
----	--------------------------------------	--	---	---------

Subsequently to these results I have added in a restriction on the dimensions of a user's profile picture to ensure it displays correctly without the need to resize the image before upload.

4.2 User acceptance testing

In addition to the functional and unit testing user acceptance testing was conducted.

4.2.1 **Trunk Test**

I will be conducting a "*trunk test*" with 3 non-tech savvy members of the workshop after with no training on the system prior to the test. The trunk test is used to evaluate the ease of navigation and the time taken to preform basic tasks on a system.

The test is conducted by showing a user a page of the website and asking them a number of questions based on the page they are currently on.

Instructions Transcript

"Hello ____ thank you for joining me today and helping out with this project. I am looking to develop a web app and need to evaluate how users will interact with it. To do this we will be conducting a trunk test. I will present you with a random page from the site and will ask you a few questions based on what you find."

Thanks again for your support

Respondent 1

Q #	Question	Answer	Correct/Incorrect
1	What site is this? (User is on login page)	I'm not sure it just says welcome to JobFlow	Incorrect.
2	What page are you on currently? (User is on Dashboard page)	I'm on the dashboard page	Correct
3	What are the main sections of this page?	Creating a ticket, the tickets section at the bottom and the menu at the top	Correct
4	What are the navigation options on this page?	There's a menu at the top, and a menu at the bottom	Correct
5	How did I get here?	The menu option on top for dashboard or the logo	Correct

Respondent 2

Q #	Question	Answer	Correct/Incorrect
1	What site is this? (User is on login page)	A work system of some sort.	Incorrect.
2	What page are you on currently? (User is on Account page)	I'm on the account page	Correct
3	What are the main sections of this page?	Name, number, email address uploading a picture and my picture	Correct
4	What are the navigation options on this page?	There's a menu at the top and can use the back button	Correct
5	How did I get here?	The account tab on the menu	Correct

Respondent 3

Q #	Question	Answer	Correct/Incorrect
1	What site is this? (User is on login page)	A payroll system.	Incorrect.
2	What page are you on currently? (User is on Account page)	I'm on the My Tickets page	Correct

<u>3</u>	What are the main sections of this page?	A list of tickets that were raised.	Correct
<u>4</u>	What are the navigation options on this page?	There's a menu at the top and can use the back button	Correct
<u>5</u>	How did I get here?	The "My Tickets" tab on the menu	Correct

Summary of results

From the trunk test it is clear that users can easily navigate and understand the page they are currently on however all respondents failed when it came to the registration page to understand what the JobFlow system is.

It is clear that there needs to be a guide or an instruction on the registration page to ensure that people understand the purpose of the system better before logging in.

5 Conclusions

By gathering feedback from the workers and admin staff directly rather than speaking to an IT manager JobFlow was designed from the users up.

JobFlow does not apply pre-defined criteria that IT think that they need but rather they have a direct say in the system that works for them.

JobFlow will lead to happier end users and much more effective planning and an overall more productive environment for all simply and easily.

5.1 System Evolution

Advantages:

The advantages of this system are primarily providing a much more efficient business process than would previously be in place.

- > One stop access for resources for users.
- > Better oversight on workload of departments.
- > Metrics are much easier to monitor and develop.

Disadvantages:

The system presumes end users have internet access in order to raise a ticket, this may not be possible in all environments.

The system requires end users and workers to have training in basic web use which may not be the case.

Opportunities:

The system provides a good basis for the potential rollout of the system for other companies to replace old inefficient and chaotic phone and email workflows that currently exist.

Limitations:

There are many CRM and ticketing systems available which integrate into other functionalities such as lead generation and marketing. E.g Microsoft Dynamics, Salesforce.

5.2 Further development or research

Azure AD

In the future it would be beneficial if JobFlow can be integrated into Azure AD much like many cloud based applications so that it is easier for businesses to integrate JobFlow within their organisation without too much disruption. If they are registered for Office 365 they will be able to access JobFlow easily through the office 365 portal if the organisation uses azure AD connect.

CRM System

The ideal situation would be as the system develops it can become a custom XRM platform for businesses to create a custom CRM system for themselves and to not have to rely on the huge costs of other systems which do not listen to user requirements effectively.

PowerBI

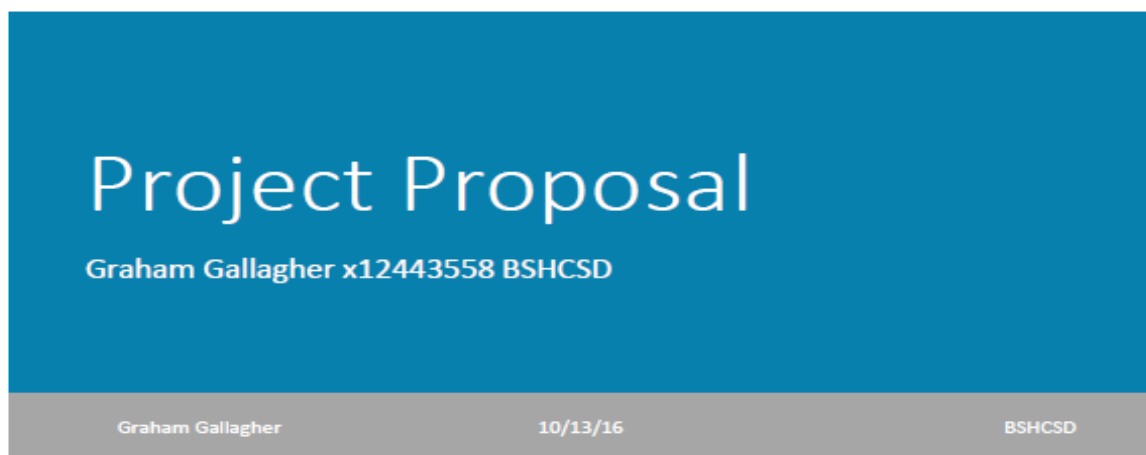
In future developments it would be beneficial to submit the system for review to the Microsoft team in order to develop an API which will allow users to connect directly with JobFlow to gain further insights into their data.

Outlook Integration

In future versions of JobFlow it is envisaged that an add in for outlook would be available for workers and end users alike to be able to track requests on a side bar. This will allow users to easily check their tickets without having to log in to JobFlow.

6 Appendix

6.1 *Project Proposal*



Introduction/Objectives

My proposed project is called "JobFlow". The objective of this project to resolve a real world problem within large organizations.

While working in a large company I found that it could be very difficult to contact or get details for a department you were not in regular contact with or would not usually be in contact with. While working in the IT department I found it became near impossible to preform your job when there is no direct line between you and the person you are looking to support. They would find it difficult to find your details on the system in order to raise an issue with IT support and we would find it difficult to keep track of the end user who raised the issue as well as being able to track support call levels and the performance of the support department.

I found that this was a prevalent issue among many departments and was not just exclusive to IT support.

On many occasions, there has been an attempt to implement a system, which will resolve this problem all with varying degrees of success. These systems were extremely expensive some charging up to €80 p/m per user and yet would still be massively ineffective. This has led to a situation where there are four different systems to log different aspects of work and to find details on work completed or raised and none of these systems would communicate with each other. This took up a lot of time for people looking to perform their work and it became impossible for it to be used effectively.

My objective is to encompass all the requirements of the end users and the workers under one umbrella system. This system will be cost efficient, flexible, easy to use and most importantly actually meets the user's requirements.

The person who requires support or is looking to raise an issue with a department can simply log on to an online portal using their work email address and password, chose the department they require, log their details and it will be logged and tracked in the relevant department. That department can then handle their workload, create KPIs for their workers and track progress all from one system saving them time, improving efficiency and removing frustration so they can get back to the work they need to do and leave the rest to JobFlow.

To demonstrate how I wish to propose the system I will use a use case.

Large Organisation with a painting and decorating department

From the users perspective

This organisation has hundreds of employees with multiple buildings. A worker has noticed that their office is looking grubby and needs to be painted.

Currently there is no clear defined way for them to communicate their request directly to the painter's workshop.

The following is a demonstration of the channel of communication between the office worker and the end painter who will be working on the job

Current Work Flow

Office Worker - > Office Manager -> Building Manager -> Head of Operations -> Head of Maintenance -> Maintenance foreman -> Painter foreman -> Worker

Proposed Work Flow

Office Worker -> Office Manager -> JobFlow System -> Painter Foreman/Workers

This use case is a perfect example of why a workflow system is required

As you can see there are so many channels of communication in the current workflow which

- Take up time and energy of management staff that could be better used elsewhere
- Creates huge frustration for the end customer and the worker
- As there is so many channels to communicate through the message could be miscommunicated or lost completely
- There is no clear responsive communication back to the customer to indicate that their job has been recognized, assigned and scheduled. The end user may or may not know if the request has even reached its destination

The JobFlow system aims to alleviate these problems by eliminating the chaotic and dysfunctional communication flow currently in place.

The worker simply communicates to their office manager and the office manager decides if they believe the office needs to be painted. Once they decide it does they log their ticket on JobFlow. The painter supervisor or worker can now see the request for painting of that building and can check their documents to see when it was last painted. They can then check if it is even necessary for this job to be completed or they work it into their schedule and reply to this request directly from the raised request to tell the office manager that their job has been scheduled and is planned on a certain date and time or rejected and a reason why.

From the workers perspective

The current workflow for workers is equally ineffective

Painter Foreman > Creates job list according to jobs raised through different channels emails, phone and written request > Jobs are not effectively prioritized > Workers are unaware of the number of jobs and the progress, priority or location > More jobs are assigned ad-hoc > Jobs raised ahead of new jobs are pushed down the que again > As there is no effective scheduling and the end user is unaware of when the job is scheduled for the worker arrives and the room is not prepared to be painted.

This workflow is extremely frustrating for both end user and worker. There is no clear system for logging tickets that can be organized, prioritized and scheduled. This means that workers cannot plan effectively and work in the most efficient manner.

The JobFlow system seeks to resolve the problems on the worker end by providing a dynamic dashboard, which will display all of the jobs raised by customers. These jobs can then effectively be reviewed and prioritized, workers can be assigned and they can communicate directly with the user to seek clarity if it is required, advise them when their job is scheduled to be completed and to advise them how they can prepare the room for it to be ready. This provides the workers a clear display of their workload, who is responsible for the task when this job has been raised.

Worker/Foreman > JobFlow System > Communicates to end User

Foreman/Admin > JobFlow System > Stats and Job organization

Administrator/Management

Currently there is no effective way of monitoring employees, productivity and workload, which makes it extremely difficult for administration and management to monitor the progress of the department they oversee.

This hampers productivity across the entire organization as it makes it near impossible to create KPIs for departments to meet as there is no way to effectively monitor what the current workload is, the anticipated workload, and the productivity per employee.

My aim is using information generated from usage data from the system I will be using Microsoft PowerBI to pull data from the SQL server in order to show useful data which can be used to monitor how workers are using the system, the number of requests per week, department etc...

Resource Requirements

My aim is to have this system deployed fully in the cloud so that it can give the flexibility required when demonstrating both in college and at an end user demo

My current plan is to roll out the system primarily for desktop however to have it built with good mobile design practice in mind. Once a solid working foundation is built for the system and it has been user tested, I can work on developing the app to work with mobile more effectively. I believe this is a crucial point, as I do not want to spend a huge amount of time on mobile if it means sacrificing building a good quality system.

Deploying to an Azure web service is going to give me the best flexibility for deployment and integration into the website. It will also make it easier to code for PowerBI to retrieve data from the SQL server.

Technical Approach

In order for this project to be a success, it is essential that system requirements will be met with the users in mind and that their input be given into the development of the system.

My approach to this system is to build from the requirements up with the end user in mind the whole time.

Requirements Building Approach:

Much like the system, my aim is to be as clear as possible regarding system expectations. Many times proposing a system to a large organization is met with hundreds of different opinions, expectations and attitudes.

For me it is crucial that my requirements gathering approach is clearly communicated and the results of the gathering process is communicated effectively in return.

Functional Foundations – Phase 1 Gathering

I will be collecting preliminary functional requirements from an end user by this process

This requirements gathering phase will first be addressed to the users for the gathering of functional requirements. This will be clearly explained to the feedback group so that we do not have unnecessary requirements being requested at this point.

Non-Functional Gathering – Phase 2

This feedback gathering session will be taken after the gathering of functional requirements and will take into account the requirements gathered previously. This phase will be working on the aesthetic and layout functions of the system and how the functional requirements will work. This I believe is an important aspect to have the end users involved in so that they are more comfortable about how the system will perform once developed and it's not going to be my own vision of how they should use it.

Conclusions/Additions Phase 3

Once all of the non-functional and functional requirements have been gathered from the user base I will be presenting them with the opportunity to provide feedback on possible additions to the system in the future, which could be developed after the system has been deployed.

Categorize and Prioritize

Once the requirements have been gathered I plan to categorize the requirements according to the section that the requirement falls under. This will make it easier to follow when I am working on different sections I can look back and develop the requirements accordingly.

Once they have been categorized, I will be prioritizing them according to need "Low" "Medium" and "High". This will provide clarity on importance so I do not spend a disproportionate amount of time on a section that isn't high priority.

Technical Details

This project requires the ability to handle users and access requirements; it needs to be flexible and easily accessed by all staff with little training.

From my previous experience in my past modules, I had created a similar system using Microsoft MVC ASP.NET to create a ticketing system for an IT department.

My aim is to create this system on the same platform using an SQL database and ADO.NET for authentication purposes.

The reason I have chosen this form of technology is there is a vast bank of information on ASP development available online from Microsoft directly as well as PluralSight and other training websites.

It also addresses the need for flexibility as this app can be deployed as an Azure web service, which allows for the flexibility of attaching directly into office 365 apps.

Evaluation

In order to evaluate the end system I will be performing extensive user testing as well as providing feedback sessions between each major release. This will ensure that the end users are aware of the progress of the system and what to expect upon release.

6.2 *Monthly Journals*

Reflective Journal

Student name: Graham Gallagher x12443558

Programme (e.g., BSc in Computing): BHCSD

Month: September

This month we began our new Semester.

Before starting back in college I was working as a web developer and software support agent during my internship. While I was working there I found there was a major issue with our ability to track and notify the finance department of software and warranty subscriptions and renewals which lead to a serious issue of software or warranty's expiring without the finance department's knowledge. This reminded me of when I worked in a large hospital for the summer after I finished school. There was consistent issues of departments not being able to communicate with each other causing inefficiencies and frustration.

Having attended the software project class and reviewing the suggested projects I've decided I would like to tackle this problem with my own system.

My Achievements

This month, we had our project proposal presentation.

We had to present in front of a panel of three lecturers and try to convince them that our chosen idea was worth pursuing.

During my presentation I felt I was able to well defend the purpose of taking on this project. It responds to a real world problem and is practical.

One of the lecturers noted that it might be a good feature to add in room booking and scheduling. I think although that would be a nice addition to the system I would rather have a solid foundation of what I had originally set out to do as this would be far too time consuming and difficult to implement.

My Reflection

I believe it was really important that we found a project that we are passionate about and that this passion will drive us forward to complete the work to the best of our ability. Particularly this year as these are our final year projects and will follow us on for the rest of our careers.

What I've noticed over the past years in NCI is that if you're not passionate about a project or are just doing something for the sake of "it'll do" it becomes incredibly difficult to find the drive to complete your work to the best of your ability when it becomes difficult and to produce something you can be proud of.

I'm a very logic orientated person and this is why my idea "JobFlow" really appealed to me. It's in an area that I am interested in of business automation and it has a purpose.

I think time management is crucial this year. I think I could have done more in the days leading up to the presentation to research competing software so I could have answered better questions put to me about them in the presentation.

Intended Changes

Over the coming month I will be looking to nail down a rough outline of the requirements for the project and to begin work on the project proposal

Reflective Journal

Student name: Graham Gallagher x12443558

Programme (e.g., BSc in Computing): BHCSD

Month: October

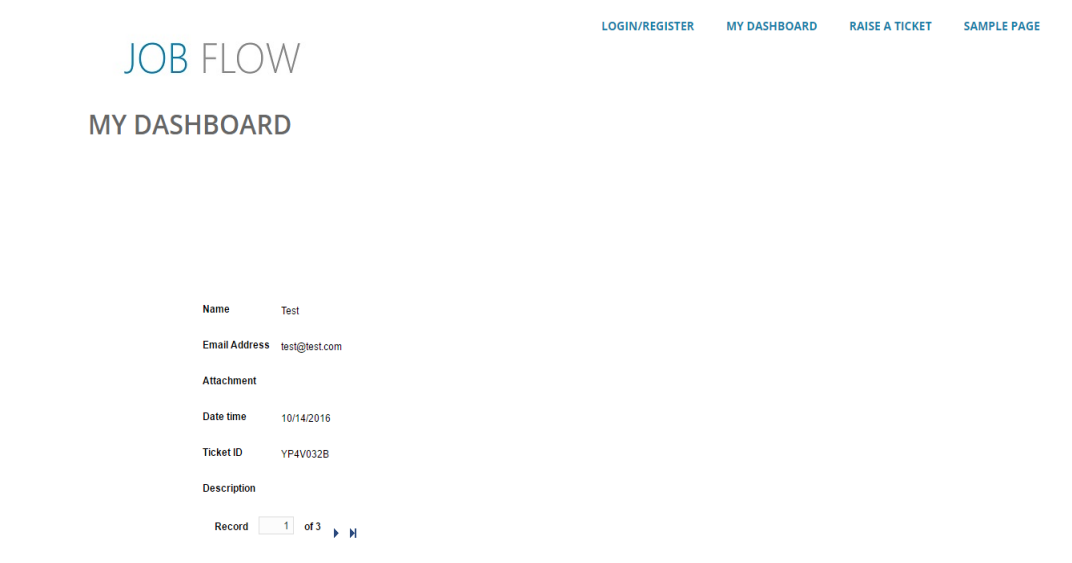
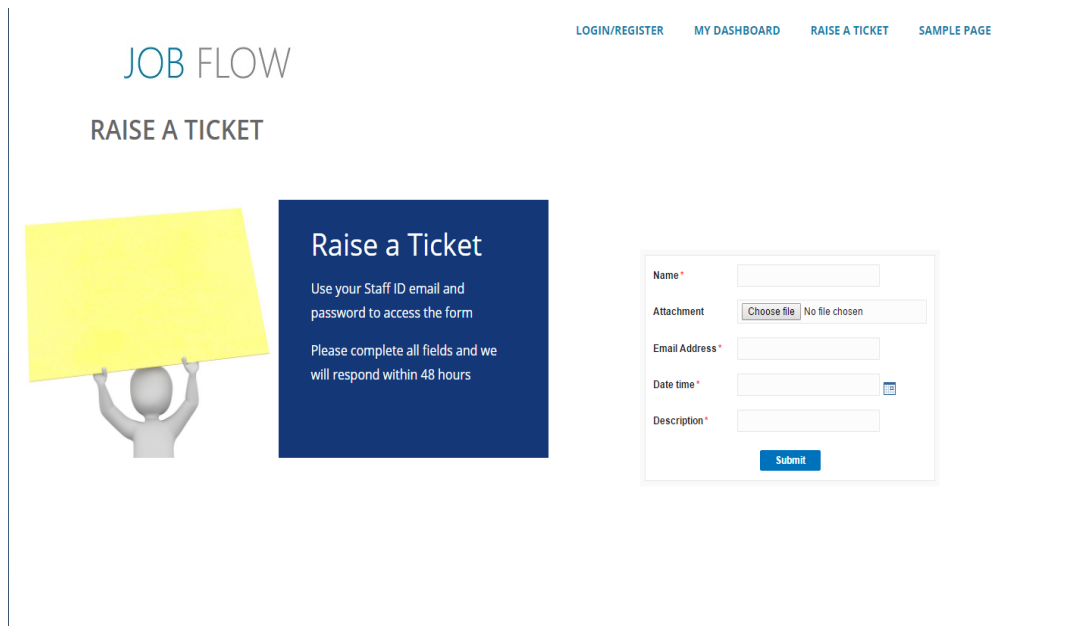
My Achievements

I had my first group meeting with my supervisor Frances.

I had gotten in touch with a contact from the maintenance department and arranged a meeting for the 19th.

For this meeting, I created a rough draft of the system on my domain. This was created using Caspio, Wordpress and Firebase. I think it was important to visualise what the end system will do so that we were both on the same page of what this system is proposing to do. The draft system was very simple it had 3 pages. The login page allowed one test user to log in, the raise a ticket page which allowed you to raise a ticket and the dashboard page which allowed you to view raised tickets. A user could raise a ticket once they logged in. They needed to enter their email and password and these credentials must have been registered onto the system previously. Once the user did this they were able to leave a brief description of their query and were able to attach documents or photos if necessary. The tickets raised were pushed to Caspio and were visible in the database. The user could view and change the tickets on the dashboard. The tickets were given a unique ID reference number which could be used in future developments for data manipulation.

I've attached screenshots of the prototype below.



Although this was a basic overview of the system it was useful for explaining how it would work and setting expectations for the meeting.

I completed my project proposal and uploaded it early. I used a use case to describe how the system will work. I felt this was easier than trying to describe the system in detail.

My Reflection

The project proposal brought home to me the scale of the job that will be at hand. I think I have let other modules consume my time more than necessary. Frances pointed out in our meeting that this module is worth more credits than any other module and it should be priority.

I held a meeting with a member of the maintenance department in the organisation I had based my original idea on the 19th and we had discussed the broad requirements of the system and what issues they are currently facing. My original intention for the meeting was to gather feedback from the end users as well as the maintenance staff but he advised that this wouldn't be possible for several weeks.

We had discussed the systems intended purpose and as we discussed it he began to add in potential add ons and features that he thought would be beneficial. I felt that we were getting off topic of the intention of the system itself and I advised him that I will take these on board as potential developments in the second phase but that we should focus on creating a good foundation system to build on.

I was quite frustrated that I couldn't have a meeting with the end users as my requirements specification is due in November and I knew it was crucial to get the feedback of the people who were going to use the end system because they would be make or break for it. I will get in touch with him again to see when this would be possible.

I'm becoming more and more aware of how time is getting on quickly and its becoming difficult to balance my time. I'm trying to more effectively plan my time by taking advantage of the independent study sessions on Mondays and not getting caught into the trap of thinking of working at home will be as productive.

Intended Changes

Over the coming month I plan on completing my requirements spec with details gathered from the meeting with the end users. If I cannot get a meeting with the end users in time I will try get another meeting with the maintenance department contact to go over the requirements, we discussed and finalise what we envisage the end system to be and do.

I will get in touch with the projects manager in the company I did my internship in and see if he would be able to provide me with some feedback on my requirements spec before I hand it over.

Requirements spec due 11/11/16

Supervisor Meetings

Date of Meeting: 26-10-16

Items discussed:

I was assigned Frances Sheridan as my supervisor.

Frances sent out an email to all the students that had her to arrange a group meeting. We all met Frances as a group in the 4th floor boardroom.

We first went around the table and discussed our ideas and the technologies we intended to use.

Frances was a great help as she questioned our use of technologies and asked how would it work and what you intended to use them for. She said a lot of people were considering Firebase and a few people in our group were going to use it too and suggested that we try work

together when developing our different parts on firebase and to share with the group any difficulties we may be having or any features that could be useful for others.

Frances told us that she will be holding group meetings every week and to hold individual meetings when required. The group meetings I feel are a great help to me as we can see how other people work on their projects and can provide help where needed and that we also have the support of being able to request individual meetings when we need them.

Action Items:

Work on requirements specification

Reflective Journal

Student name: Graham Gallagher x12443558

Programme (e.g., BSc in Computing): BHCSD

Month: November

My Achievements

I began working on my midpoint presentation and prototype to be ready for the mid-point.

I originally based the project on WordPress and caspio however going forward I have began developing the project using visual studio and have gotten to grips with SQL integration

My Reflection

I found this month to be extremely stressful. We have 5 projects in all due around the same time. Software Project prototype, Mobile and Multimedia Project, Data Application Development project, web services project and artificial intelligence project. This has severely distracted me from the software project and has been disappointing for me as I really would have liked to have been able to spend more time on it and to work out a better plan.

Because of the CA schedule for this year I think I really need to reevaluate my work schedule for this project and set myself better targets and deadlines.

Our technical report is due on Sunday so I am aiming to have a high quality document ready in time.

Intended Changes

I intend to have regular meetings with Frances with the run up to the mid-point and get some guidance on planning the project development and implementation going forward.

Tech Report due 11/12/16

Supervisor Meetings

Date of Meeting: 26-10-16

Items discussed:

Action Items:

Work on the technical report and conduct regular meetings with Frances in order to keep better in touch with my supervisor.

Reflective Journal

Student name: Graham Gallagher x12443558

Programme (e.g., BSc in Computing): BHCSD

Month: December

My Achievements

I had my midpoint presentation with my supervisor Frances Sheridan and lecturer Lisa Murphy.

I had also completed my technical report.

My Reflection

This month was stressful with the end of our lectures approaching and the mid point coming up.

I had completed my technical document and sent it on to Frances for review as well as my PowerPoint before my presentation.

I was very happy with how the presentation went. I was able to clearly explain the background and the reasoning for the project as well as what I'm trying to achieve without going overboard.

I was also able to get a demo up and running to show them the basic functionalities of the system and they gave me great feedback (I was feeling a bit lost as to where to go next or what I should focus on improving).

Frances gave me great points on improving the user experience of the system and noticed something that stuck with me. She said that users of this system would typically be moving around a lot and wouldn't be sitting in front of a desk as much so to make sure that the system is mobile friendly. This is something I hadn't really thought of and could have been a big miss.

Lisa gave me valuable feedback on the overall look and design of the project. She advised me to focus on getting a clear brand and colour and style scheme and to make it as colourful and clean looking in order for it to be more user friendly and impressive if it is to be a viable commercial project.

Working on my android project gave me an idea to possibly add in an android app on the side for this project but I won't focus too much on this.

Intended Changes

Im going to look into the style and branding of the system and try make it clearer and more concise.

Create a project plan in the run up to semester 2.

Supervisor Meetings

Date of Meeting: 19-12-16 (Mid Point)

Items discussed: Style, Branding, User friendliness, responsiveness and future actions

Action Items:

Create a better branding and styling plan for the project and a project timeline for semester 2.

Reflective Journal

Student name: Graham Gallagher x12443558

Programme (e.g., BSc in Computing): BHCSD

Month: January

My Achievements

Met with Frances to discuss progress to date and future work

Began development in Laravel

My Reflection

This month I'm very aware of the workload that will be placed on us with our upcoming projects so I wanted to make sure I had a solid foundation built so I wasn't leaving it all to the last minute.

I previously had planned to develop the system based on Visual Studio MVC however as I developed it became more and more difficult to add functionality and it was also noted during the presentation that it's important to have mobile functionality for the nature of the project.

I took the decision to completely overhaul the project and begin with Laravel php. I'm finding the language much easier to use and I'm able to integrate bootstrap into it simply and effectively without hassle.

I met with Frances to discuss progress to date and she advised to get to the point where I was with the previous development with laravel before meeting her again. I began following a pluralsight tutorial which showed how to create a blog and to attach a database. I developed this and found it so much easier to code in rather than though C#.

I also set up my GitHub for the laravel development so I can backtrack If the need be as well as to be able to see the progress to date.

Intended Changes

Im planning on writing up a new system design strategy to keep myself on track with development rather than developing pages on the fly.

Meet with Frances on a regular basis to dicuss progress and get advise on how I should proceed moving forward.

Supervisor Meetings

Date of Meeting: 30-1-17 (Mid Point)

Items discussed: Change of language and progress to date

Action Items:

Get back to the point I was with C#

Date of Meeting: 6-2-17

Items discussed: Progress to date and system design

Action Items:

Create a working database

Reflective Journal

Student name: Graham Gallagher x12443558

Programme (e.g., BSc in Computing): BHCSD

Month: February

This month I had completed my usability project and began development on my cloud application development project. I found that the cloud application project was really useful as Ruby seems to be very similar to Laravel. Unfortunately I found that this has distracted me a lot from my software project but I still managed to make a lot of progress.

I had a meeting with Frances to discuss developments so far and showed her the database that holds the records for when users register and the encryption features that were implemented. She was happy with the progress so far and recommended I continue working on it and email her when I have a full workflow so she can recommend improvements. She also noted that it's Important that I implement testing code that was covered in the seminar and to gather feedback for the project so that I can show evidence of it solving a problem in the real world.

Currently users can register using their email address and password. The password is encrypted on the database end and the field is data type of password so it doesn't display when users enter it. Users can create and post tickets as well as being able to view their "ticket stream" to see a list of open tickets.

My Achievements

This month, we had our project proposal presentation.

Attended a meeting with Frances to discuss progress so far.

Created a fully functional database.

Created ticket stream and ticket logging.

Implemented security features that prevent users from accessing dashboards without an authorized login.

My Reflection

I feel like I keep on going off and developing features that I like without working to a plan and then having to remove them or rearrange them. I think going forward I really need to nail down my development process to get the workflow fully working so I can spend more time on improving the system rather than developing it at a later stage.

Intended Changes

I need to document exactly what I have done so far and the features of the system as well as creating an elevator pitch for the project for the showcase.

Creating a workflow that does exactly what the project intends.

Arrange a meeting with Frances.

Complete showcase material.

Supervisor Meetings

Date of Meeting: 06-02-17, 20-02-17

Items discussed: Progress so far, database connection, future developments

Action Items: Create a workflow and come back for a meeting, discuss testing with business contact, create development plan.

Reflective Journal

Student name: Graham Gallagher x12443558

Programme (e.g., BSc in Computing): BHCSD

Month: March

My Achievements

I developed an application using Ruby on rails for my cloud module which helped me a lot in my software project because I was able to trial and error publishing my code to heroku. It's still not quite there but It's another milestone out of the way.

I finished a report for my Usability Design and this really helped me develop some usability techniques that I could use when testing my project for user acceptance.

My database had completely wiped out which was a huge shock to me and caused a major setback in the development of the project but thankfully I was able to repair it and it is now holding user data again.

We have been given the spec for our software showcase and the materials required to complete the final upload.

My Reflection

It's hard to believe its almost over. I found that setbacks really knock my confidence in the project. The difficulty with the database after working so hard to get it up and running made it really frustrating to work on and took the joy out of it. I found that with every setback though once It's repaired and I've learned what caused it and stop it from happening again I start enjoy working on it again.

I was looking over the marking scheme and I think I need to spend some time working on a manual as well as the testing and usability strategies as there is a good chunk of marks going

for it. I think as well with the testing of my project its important that I get some testing code in to validate the code.

Intended Changes

Over the coming weeks I plan on getting the app deployed to heroku and have my user testing complete in time before the software showcase. I also plan on creating documentation material for users of the system and to have a final meeting with Frances to go over the next steps forward before the showcase and upload.