TECHNICAL REPORT

E-LEUKEMIA WEB APPLICATION

x13747569 Veronica Ancuta
National College of Ireland

BSc in Computing -Cyber Security

(X13747569 Veronica.ancuta1@student.ncirl.ie)

Contents

Declaration Cover Sheet for Project Submission

SECTION 1

Name: Veronica Ancuta
Student ID:x13747569
Supervisor: Kalianee Boodoo

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: Veronica Cristina Ancuta Date:10/05/017

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,

Executive Summary

E-Leukaemia is a web application framework that targets the Leukaemia sector which will aid and benefit children suffering from leukaemia. The application has a responsive web design and aims to adjust the presentation on any mobile devices to allow access to information from anywhere and at any time with an active internet connection.

The proposed solution target kids with leukaemia aged 0-16 and it will help parents to administrate the treatment as kids can’t take medication unsupervised due to safety precautions. The consequences of incorrectly administering medication can range from minor to life-threating.

E-leukaemia is expected to provide the users with various functions such as retrieving patients profile, visibility to weekly treatment and blood tests results, ability to track symptoms when required and create appointments. Medical staff will post patients treatment and lab results on a weekly/monthly basis allowing transparency for the end users on the functions outlined. Administrators can register new accounts only if patients are diagnosed with leukaemia and are attending this hospital. Role based access along with security features will ensure integrity and availability in the system. To use the functions of the E-Leukaemia it is important that all authorised staff in the hospital should have access on the existing terminals in Hospital.

The application will have security features to ensure confidentiality and integrity of sensitive data which are the key points from any security principles.

A dynamic prototype is developed by using ASP.NET, MVC5, CSS3, HTML5, JavaScript, MS SQL Server and bootstrap technologies. The system test and evaluation is made to show the system’s usability.

1 Introduction

The report consists of the following main parts: analysis, design and implementation, and evaluation. The document will outline the purpose and feature of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. The purpose of this document is to set out the requirements for the development of a Web Application designed for kids with leukaemia.

1.1 Background

Health care is one of the most important sectors in any country for its national interest. Hospitals currently use numerous paper work with data stores spread throughout the hospital management infrastructure. Often the information is not complete or does not follow the standards and it's not centralised. Multiple copies of the same information exist in the hospital and may lead to data inconsistency. A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of information. This type of information typically involves patient profile, medical history, lab results, symptoms and so on. Ideally this information must be managed in an efficient way and error free.

In Ireland, there is no electronic system incorporated within hospitals to help patients see their health history, blood results, treatment plans etc. In the frenetic place of A&E Departments in Ireland, important patient details and vital information is handwritten on papers, whiteboards like in schools, memorised by overworked hospital staff and transferred verbally across the medical team. Information is widely used with different meaning and purposes.

("How Technology Could Save Your Life In An Irish A&E")

The proposed solution will develop a E-Leukaemia web application specifically designed for kids with leukaemia that are registered in St John's ward, Crumlin

Hospital. The application will be used and accessed by the parents that will administrate the treatment during leukaemia remission which can last around 3 years for boys and 2 years for girls.

E-Leukaemia has the potential to greatly improve health service efficiency, it should expand or extend treatment delivery for thousands of patients in developing countries and improve patients results. Research suggests that the web users express the strong desire to own an online website which is used for their self-management and directly getting advice from the doctor.

This web application is focusing on self-management treatment, tracking patient symptoms and interoperability. Right information at right time saves lives, so E-Leukaemia web application framework is a part of information and communication technology that supports parents/careers to administrate treatment for kids with leukaemia. The web application can help patients the manage the treatment journey by providing important information through digital technology and scrapping the paper version. All this information is also visible for the medical staff and having the information centralised and accessible 24hours it will save time. The application will be used every week/month by the patients when new treatment will be approved based on individual patient history by the medical team.

1.2 Aims

The main purpose of the project is to develop an E-Leukaemia system basic frame combined with the responsive web design technology incorporated within Crumlin Hospital-St john's Ward. It provides treatment plan for the patients uploaded by medical staff on a weekly/monthly basis, lab blood tests result and it will post the upcoming appointments. Also, patients can track symptoms and side effects after each chemotherapy medication or whenever required.

All the information must be managed in an efficient way and error as it may be fatal for the patients. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies.

Role based control will be implemented for the medical staff so only medical staff can update patient profile. Privacy and confidentiality is an important aspect of this system thus the patient hospital relationship must be confidential and protected.

Database access will be protected and sensitive data encrypted. Using Role Base Access Control will authorize medical staff to update or read data.

Summarised Functionalities

Admin

1. Register users- The admin can register new accounts for existing patients in the hospital system to have access to the application. The registration will require users to provide an email address and a hospital number. The admin will check to see if the patient number is already in the hospital system and it will create the account. The medical staff will require also registration to access the application. The medical staff will be registered by providing same details: hospital number and email address.
2. CRUD functionalities- the admin will have full control over all functionalities and will maintain the database.

Medical staff

1. Create, delete, update appointments- medical staff can create, update and delete appointments
2. Read patient symptoms- Medical staff can read submitted symptoms by the patients
3. Access all patient's data- Medical staff can create, read and update patients profiles
4. Post lab results -Medical staff can post laboratory results for patients

5. Post patient treatments- The application will allow medical staff to submit treatments for patients

Patient(parent)

- Create symptoms- patients can request appointments
- Read its profile- patients can see their details
- Read treatment- patients can access the system and read their posted treatment by the medical staff
- Read lab Results- patients can access the blood tests results

Visitors

- Read home, about, contact pages- unregistered visitors can access only home page, about and contact

1.3 Technologies

E-leukaemia will be developed using the following technologies: MVC, ASP .NET, bootstrap, JavaScript, Azure Cloud for hosting the application and MS SQL. The application will use bootstrap technology for front end to enhance the application to be responsive and accessible from any mobile device. The front page of web application is a modified template to suit E-Leukemia web application. In order to build my project, I used Microsoft Visual Studio that allowed me to test and run the application on the local server.

1.4 Structure

The first section of this report will provide the reader a general overview to the project and it will help readers to understand what the project does exactly. The background section is identifying current problem associated with the project solution. The aims of the project outline the main functionalities that will be achieved for the final product.

System will describe the project requirements, design, the engineering of the software involved, testing plans, GUI layout, customer testing and evaluation.

Conclusions will describe the outcome of the project and a summary of all the obstacles encountered during the implementation and what I have learned during the progress of the project.

Further development or research describes how the system could be further developed such as enhanced functionalities to allow virtual connection between patients and doctors, extend the application to suit different departments and development on different platforms (IOS, Android).

References is the section where a list of all my internal and external learning resources are outlined

Appendix contains monthly reflective journals and other materials used.

2 System

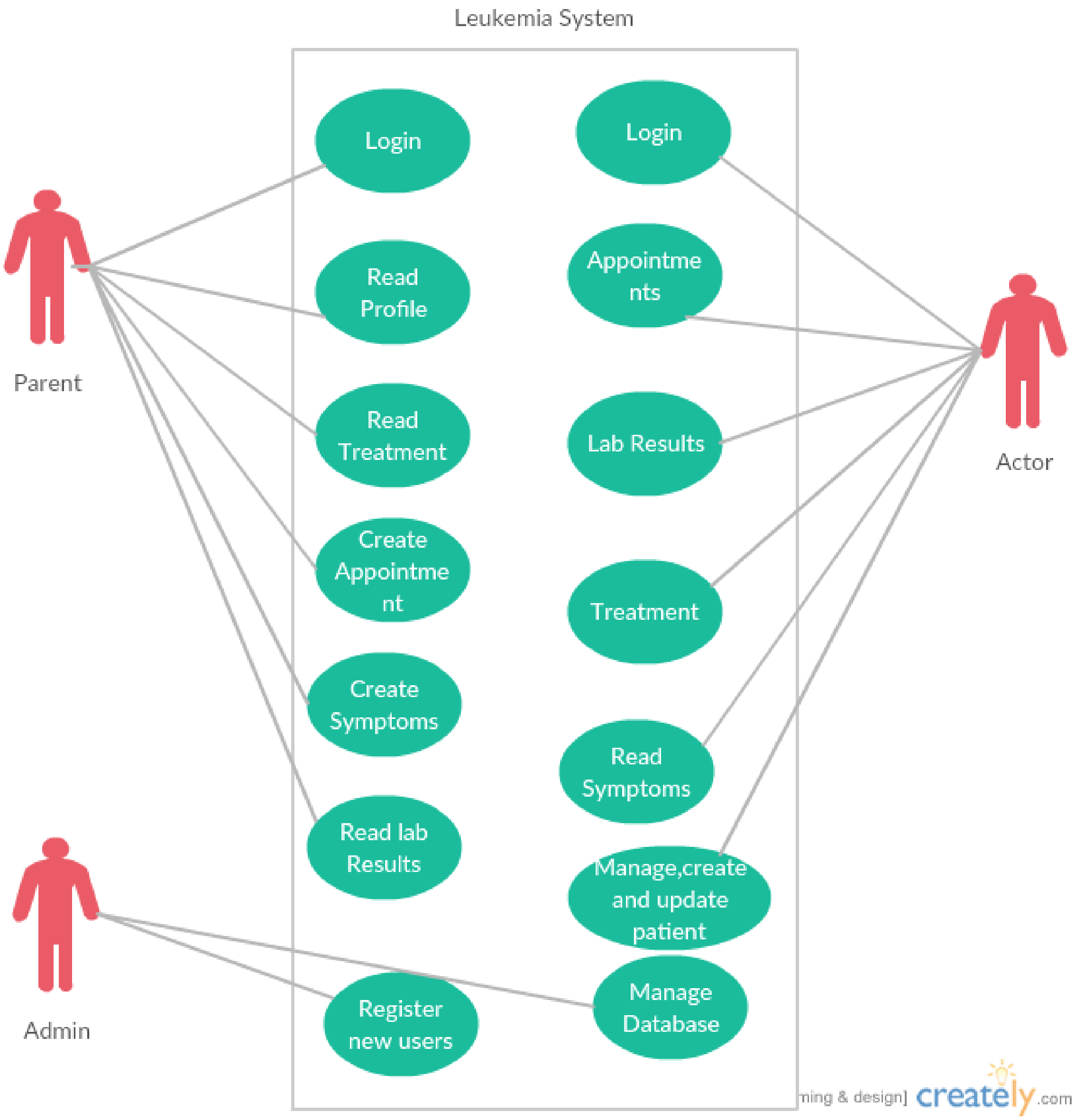
2.1 Requirements

System Use Case Diagram

Use case ID	Use case name	Primary actor	Scope	Complexity	Priority
1	Login	Parent, Medical Staff	In	High	1
2	Read Treatment	Parent	In	High	1
3	Manage Profile	Parent	In	Low	3
4	Track symptoms	Parent	In	Medium	2
5	Get Lab Results	Parent	In	Medium	2
6	Create, read appointments	Parent	In	High	1
7	Update Patient information	Medical Staff	In	High	1
8	Create, delete, update appointments	Medical Staff	In	High	1
9	Post lab results	Medical Staff	In	High	1

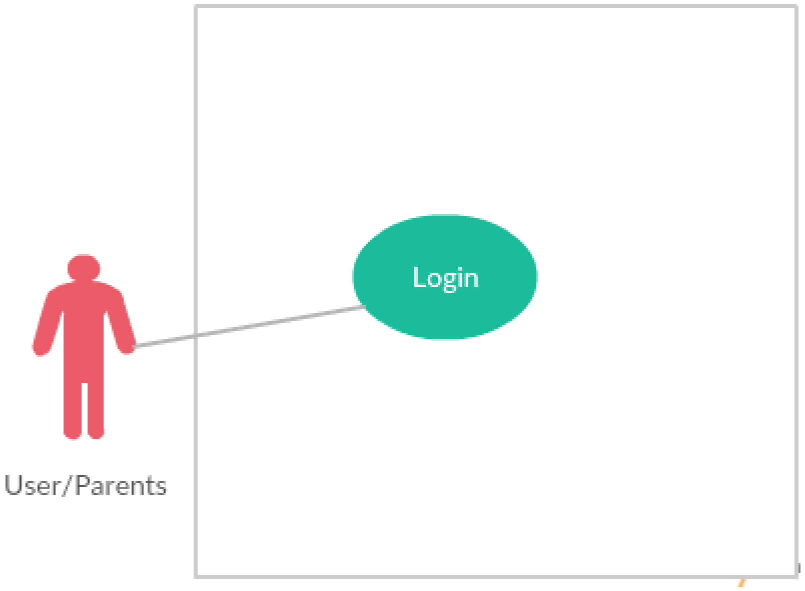
10	Manage database	Admin	In	High	1
11	Add/delete users	Admin	In	Medium	2
12	Register new accounts	Admin	In	High	1

Note: All users require login authorisation and sensitive data will be encrypted in database



2.1.1 Functional requirements

Log-in

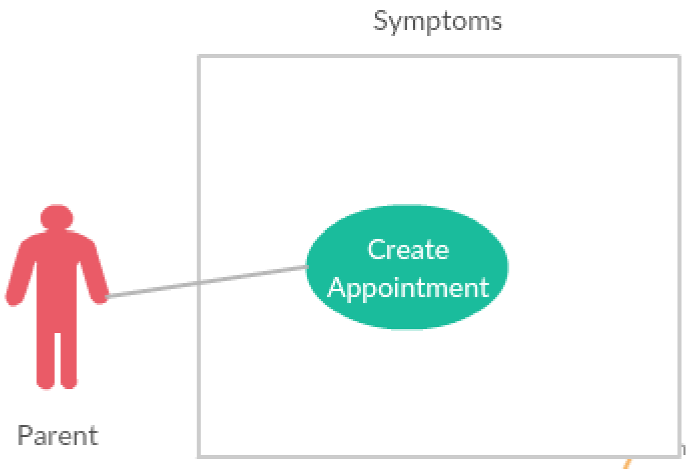


ID:	UC-1
Title:	Login for parent
Description:	Parent accesses the system the system and proceed with login
Primary Actor:	Parent
Preconditions:	Active internet connection
Post conditions:	User is registered
Main Success Scenario:	1. User selects “Login” from the menu.

	2. System prompt for the used id and password 3. Parent enter the password. 4. System sends a token to the parent phone for authentication 5. Parent enter the pin number received via text message 6. System direct the user to the patient portal while verifying use
Extensions:	a. Wrong user name. a1. Parent enter wrong password. a2. Parent either backs out of this use case, or tries again after rectifying the cause. a3. After three password attends system locks the account. a4. System displays the message error
Frequency of Use:	Daily use and sometimes every second day
Termination:	The use case terminates when the authorized user make another selection from the menu
Priority:	P1 – High

Post condition:	The system goes into a wait state

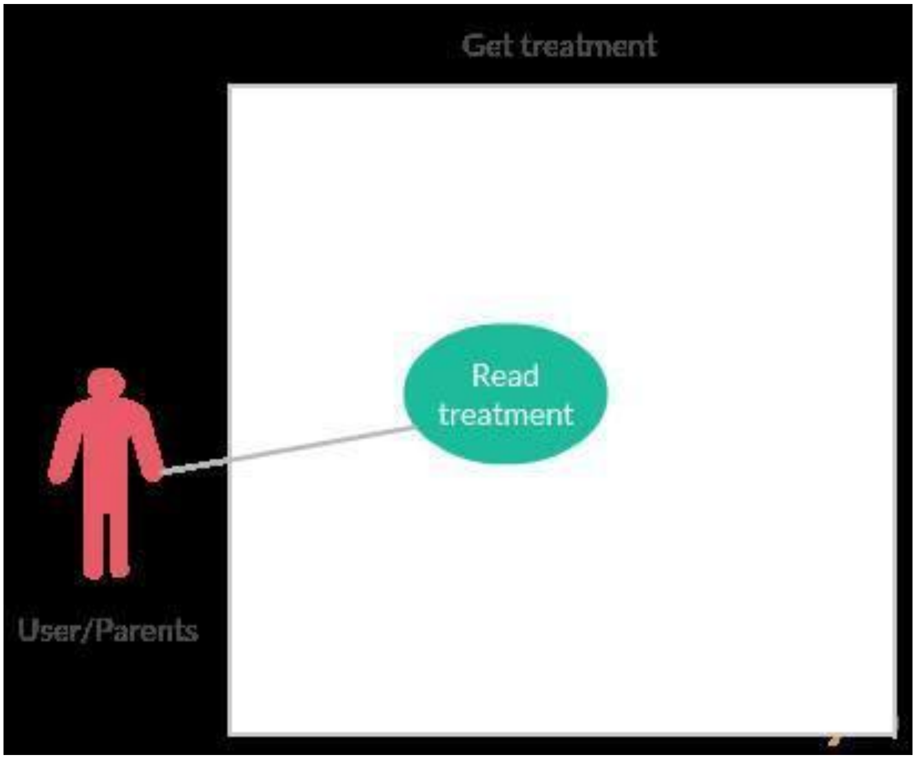
Create Appointment



ID:	**UC-3**
Title:	Parents create appointment
Description:	Parent accesses the system and select appointment tab.
Primary Actor:	Parent
Preconditions:	Parents logged into system with an active internet connection

Post conditions:	Parent is a registered user
Main **Success Scenario:**	1. Parent selects “Appointment” tab from the menu. 2. Parent can select Medical staff name 3. Parent select Department 4. Parent select date for appointment 5. Parent select time slot 6. Parent write the reason for appointment (description)
Extensions:	5.a. Time slot returns an error due to an existing appointment in the system
Frequency of Use:	Weekly/Monthly
Termination:	The use case terminates when the user requests another information from the system
Priority:	P2-High
Post condition:	The system goes into a wait state

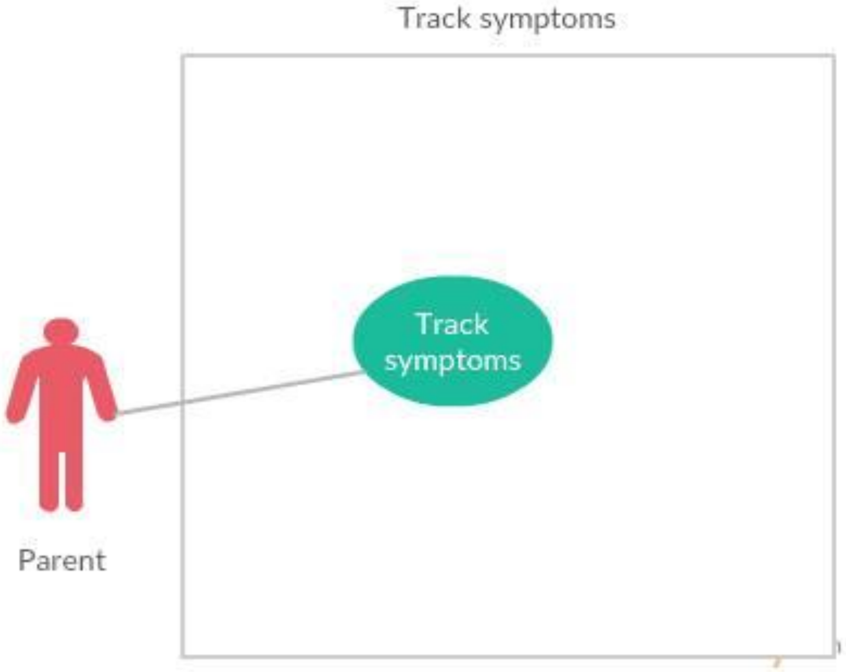
Parents read treatment



ID:	**UC-3**
Title:	Parents read treatment
Description:	Parent accesses the system and views the patient treatment.
Primary Actor:	Parent
Preconditions:	Parents logged into system with an active internet connection
Post conditions:	Parent is a registered user
Main **Success Scenario:**	1. Parent selects “Patient Treatment” tab from the menu. 2. System displays details of the

	updated treatment.
Extensions:	a. No alternate flow or extensions are available for this case
Frequency of Use:	Daily or every week
Termination:	The use case terminates when the user requests another information from the system
Priority:	P2-High
Post condition:	The system goes into a wait state

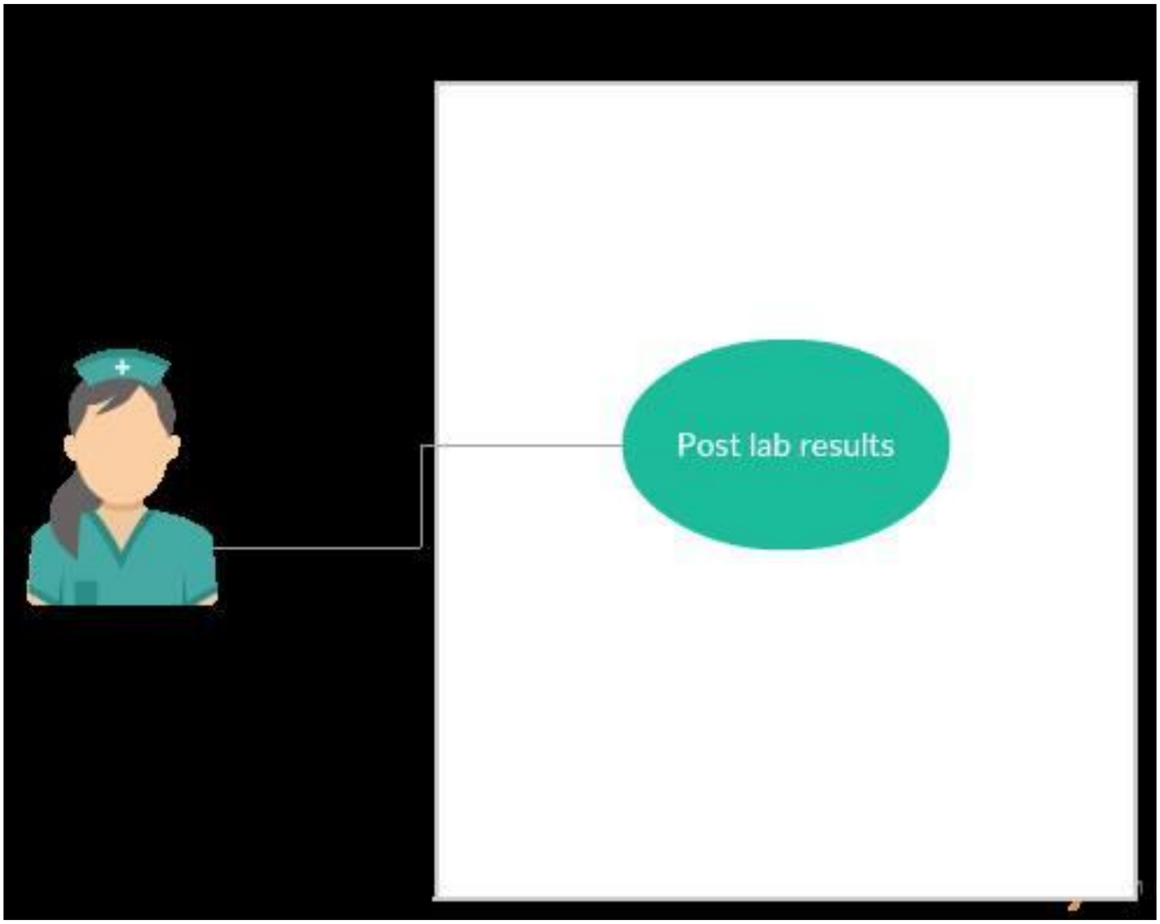
Parent track symptoms



ID:	**UC-3**
Title:	Parent track symptoms
Description:	Parent access the system and views the available tabs. Then he selects the Track symptoms.

Primary Actor:	Parent
Preconditions:	Parent is logged into system
Post conditions:	Parent is a registered user
Main **Success Scenario:**	1. Parent selects “Track symptoms” from the menu. 2. System retrieve the requested tab. 3. Parent enter the symptoms in the text area. 4. Parent clicks “Submit” button when he finished typing 5. System records and display the typed content.
Extensions:	a. No alternate flow or extensions
Frequency of Use:	When required
Termination:	The use case terminates when parent request another information from the menu
Priority:	P3-Medium
Post condition:	The system goes into await state

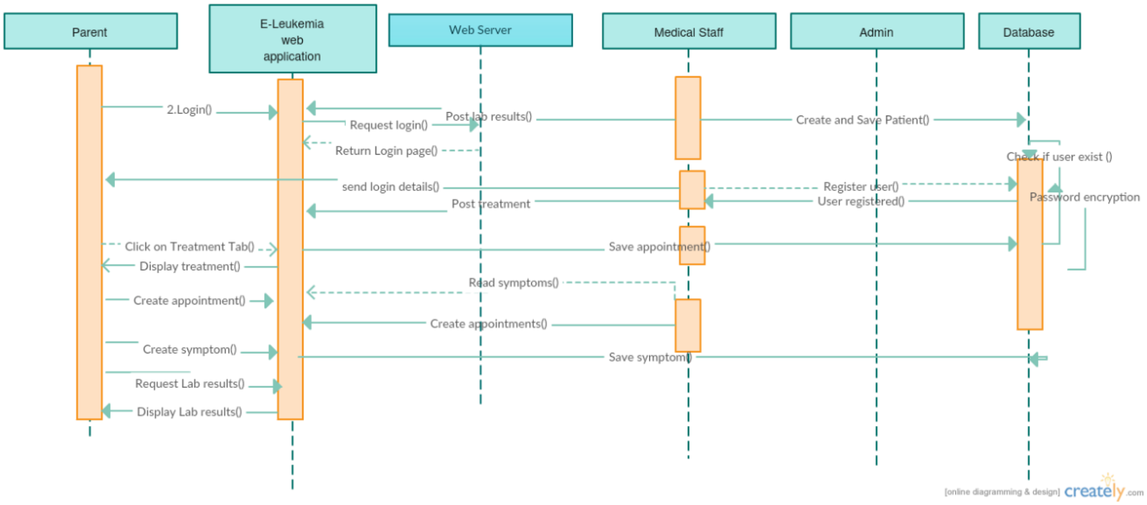
Medical staff post lab results



ID:	**UC-5**
Title:	Medical Staff posts blood tests results
Description:	Medical Staff access the system the system and proceed with login
Primary Actor:	Medical Staff
Preconditions:	Active internet connection
Post conditions:	Lab is authorized to access the system

Main **Success Scenario:**	1. User selects “Login” from the menu. 2. System prompt for the used id and password 3. Medical Staff enter the password. 4. System check if the user has the authorization to access this portal 5. If successful verified system displays a window allowing staff to select patient ID 8. Medical Staff click on submit button after entering the results 9. Medical staff can select another patient or log off
Extensions:	a. Wrong user name. a1. Medical staff enter wrong password. a2. Medical staff is not authorized to access the portal.
Frequency of Use:	Daily use and sometimes every second day
Termination:	The use case terminates when the lab logs out
Priority:	P1 – High
Post condition:	The system goes into a wait state

Sequence Diagram for the System (general overview)



2.1.2 Data requirements

All input data required by the application should be simple and easy to understand by the users when filling in the forms. The system will store the data in the database then presented on the interface allowing the user to revise the entered data ensuring the details are correct. User validation feature will inform users of any issues when inserting data into the fields. (e.g. date of birth validation-DOB cannot be from the future).

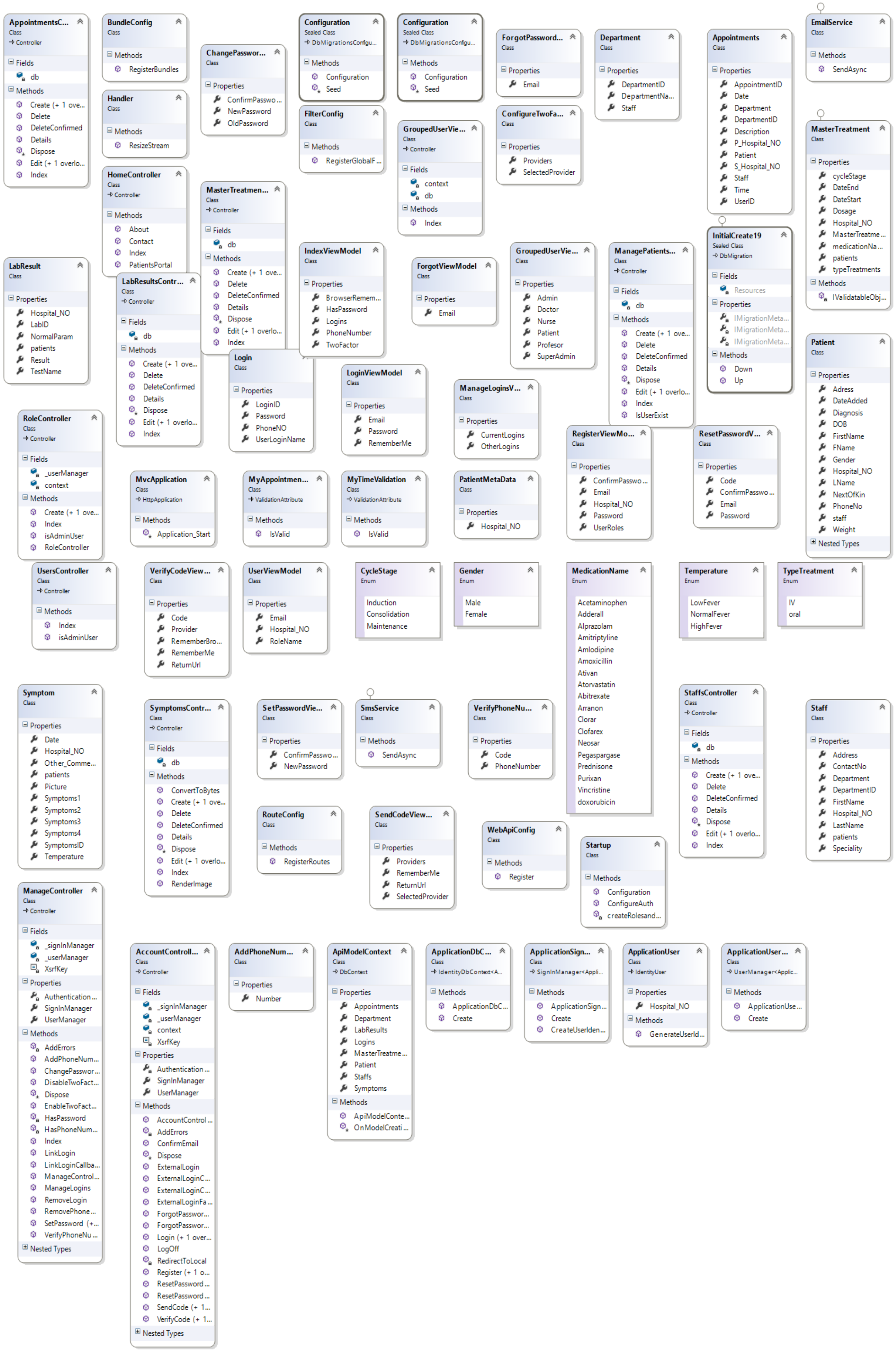

AppointmentsC...
Class
→ Controller
Fields
db
Methods
Create (+ 1 ove...
Delete
DeleteConfirmed
Details
Dispose
Edit (+ 1 overlo...
Index
BundleConfig
Class
Methods
RegisterBundles
ChangePasswor...
Class
Properties
ConfirmPasswo...
NewPassword
OldPassword
Configuration
Sealed Class
→ DbMigrationsConfigu...
Methods
Configuration
Seed
Configuration
Sealed Class
→ DbMigrationsConfigu...
Methods
Configuration
Seed
ForgotPassword...
Class
Properties
Email
Department
Class
Properties
DepartmentID
DepartmentNa...
Staff
Appointments
Class
Properties
AppointmentID
Date
Department
DepartmentID
Description
P_Hospital_NO
Patient
S_Hospital_NO
Staff
Time
UserID
EmailService
Class
Methods
SendAsync
Handler
Class
Methods
ResizeStream
FilterConfig
Class
Methods
RegisterGlobalF...
ConfigureTwoFa...
Class
Properties
Providers
SelectedProvider
GroupedUserVie...
Class
→ Controller
Fields
context
db
Methods
Index
MasterTreatment
Class
Properties
cycleStage
DateEnd
DateStart
Dosage
Hospital_NO
MasterTreatme...
medicationNa...
patients
typeTreatments
Methods
IValidatableObj...
HomeController
Class
→ Controller
Methods
About
Contact
Index
PatientsPortal
MasterTreatmen...
Class
→ Controller
Fields
db
Methods
Create (+ 1 ove...
Delete
DeleteConfirmed
Details
Dispose
Edit (+ 1 overlo...
Index
InitialCreate19
Sealed Class
→ DbMigration
Fields
Resources
Properties
IMigrationMeta...
IMigrationMeta...
IMigrationMeta...
Methods
Down
Up
LabResult
Class
Properties
Hospital_NO
LabID
NormalParam
patients
Result
TestName
LabResultsContr...
Class
→ Controller
Fields
db
Methods
Create (+ 1 ove...
Delete
DeleteConfirmed
Details
Dispose
Edit (+ 1 overlo...
Index
IndexViewModel
Class
Properties
BrowserRemem...
HasPassword
Logins
PhoneNumber
TwoFactor
ForgotViewModel
Class
Properties
Email
GroupedUserVie...
Class
Properties
Admin
Doctor
Nurse
Patient
Profesor
SuperAdmin
ManagePatients...
Class
→ Controller
Fields
db
Methods
Create (+ 1 ove...
Delete
DeleteConfirmed
Details
Dispose
Edit (+ 1 overlo...
Index
IsUserExist
Patient
Class
Properties
Adress
DateAdded
Diagnosis
DOB
FirstName
FName
Gender
Hospital_NO
LName
NextOfKin
PhoneNO
staff
Weight
Nested Types
Login
Class
Properties
LoginID
Password
PhoneNO
UserLoginName
LoginViewModel
Class
Properties
Email
Password
RememberMe
ManageLoginsV...
Class
Properties
CurrentLogins
OtherLogins
RoleController
Class
→ Controller
Fields
_userManager
context
Methods
Create (+ 1 ove...
Index
isAdminUser
RoleController
RegisterViewMo...
Class
Properties
ConfirmPasswo...
Email
Hospital_NO
Password
UserRoles
ResetPasswordV...
Class
Properties
Code
ConfirmPasswo...
Email
Password
MvcApplication
Class
→ HttpApplication
Methods
Application_Start
MyAppointmen...
Class
→ ValidationAttribute
Methods
IsValid
MyTimeValidation
Class
→ ValidationAttribute
Methods
IsValid
PatientMetaData
Class
Properties
Hospital_NO
UsersController
Class
→ Controller
Methods
Index
isAdminUser
VerifyCodeView...
Class
Properties
Code
Provider
RememberBro...
RememberMe
ReturnUrl
UserViewModel
Class
Properties
Email
Hospital_NO
RoleName
CycleStage
Enum
Induction
Consolidation
Maintenance
Gender
Enum
Male
Female
MedicationName
Enum
Acetaminophen
Adderall
Alprazolam
Amitriptyline
Amlodipine
Amoxicillin
Ativan
Atorvastatin
Abitrexate
Arranon
Clorar
Clofarex
Neosar
Pegaspargase
Prednisone
Purixan
Vincristine
doxorubicin
Temperature
Enum
LowFever
NormalFever
HighFever
TypeTreatment
Enum
IV
oral
Symptom
Class
Properties
Date
Hospital_NO
Other_Comme...
patients
Picture
Symptoms1
Symptoms2
Symptoms3
Symptoms4
SymptomsID
Temperature
SymptomsContr...
Class
→ Controller
Fields
db
Methods
ConvertToBytes
Create (+ 1 ove...
Delete
DeleteConfirmed
Details
Dispose
Edit (+ 1 overlo...
Index
RenderImage
SetPasswordVie...
Class
Properties
ConfirmPasswo...
NewPassword
SmsService
Class
Methods
SendAsync
VerifyPhoneNu...
Class
Properties
Code
PhoneNumber
StaffsController
Class
→ Controller
Fields
db
Methods
Create (+ 1 ove...
Delete
DeleteConfirmed
Details
Dispose
Edit (+ 1 overlo...
Index
Staff
Class
Properties
Address
ContactNo
Department
DepartmentID
FirstName
Hospital_NO
LastName
patients
Speciality
RouteConfig
Class
Methods
RegisterRoutes
SendCodeView...
Class
Properties
Providers
RememberMe
ReturnUrl
SelectedProvider
WebApiConfig
Class
Methods
Register
Startup
Class
Methods
Configuration
ConfigureAuth
createRolesand...
ManageController
Class
→ Controller
Fields
_signInManager
_userManager
XsrfKey
Properties
Authentication...
SignInManager
UserManager
Methods
AddErrors
AddPhoneNum...
ChangePasswor...
DisableTwoFact...
Dispose
EnableTwoFact...
HasPassword
HasPhoneNum...
Index
LinkLogin
LinkLoginCallba...
ManageControl...
ManageLogins
RemoveLogin
RemovePhone...
SetPassword (+...
VerifyPhoneNu...
Nested Types
AccountControll...
Class
→ Controller
Fields
_signInManager
_userManager
context
XsrfKey
Properties
Authentication...
SignInManager
UserManager
Methods
AccountControl...
AddErrors
ConfirmEmail
Dispose
ExternalLogin
ExternalLoginC...
ExternalLoginC...
ExternalLoginFa...
ForgotPasswor...
ForgotPasswor...
Login (+ 1 over...
LogOff
RedirectToLocal
Register (+ 1 o...
ResetPassword...
ResetPassword...
SendCode (+ 1...
VerifyCode (+ 1...
Nested Types
AddPhoneNum...
Class
Properties
Number
ApiModelContext
Class
→ DbContext
Properties
Appointments
Department
LabResults
Logins
MasterTreatme...
Patient
Staffs
Symptoms
Methods
ApiModelConte...
OnModelCreati...
ApplicationDbC...
Class
→ IdentityDbContext<A...
Methods
ApplicationDbC...
Create
ApplicationSign...
Class
→ SignInManager<Appli...
Methods
ApplicationSign...
Create
CreateUserIden...
ApplicationUser
Class
→ IdentityUser
Properties
Hospital_NO
Methods
GenerateUserId...
ApplicationUser...
Class
→ UserManager<Applic...
Methods
ApplicationUse...
Create

Figure 1 Class diagram

2.1.3 User requirements

User experience requirements is the most important and if they request special needs or preferences for an application they should be implemented as users come first.

E-Leukaemia came in to use and scrapped the paper based system to record patient's information. These paper files might have errors when data in entered into the documents. By using E-Leukaemia, data is stored electronically and is much simpler, it's more efficient and perhaps has less errors or at least they can be spotted and rectified quickly.

Long term treatments can be placed electronically and this would help patients to manage the medication. Also, E-Leukaemia can help doctors to reduce the time of locating and reading patients information. Patients can begin to become gradually aware of self-care management. Also it is convenient for maintaining with experts in medical and application development.

In order to use the system Parents needs an active internet connection and mobile devices to use the application. The application can be accessed from any device such a laptop, smart phone, tablet etc. as it has a responsive design built in with bootstrap features.

Patients will be registered by the hospital and they will be provided with the login details and password if they want to use the application. The password can be later changed by the user.

The pilot solution is addressed to a small number of users that will be registered by the hospital when they first attend the hospital. The users will sign a document with the hospital about data protection, confidentiality and disclosure policy which will also include a consent to use the application.

Medical staff would like a dynamic system so they can access at anywhere to retrieve the information about patients. Handwritten papers are time consuming and sometimes this can lead to mistakes. Having all the information centralized can help doctors or nurses to focus on patient records and make decisions faster. Also, the system can help in minimizing medical errors and misunderstandings.

The web application need an administrator to maintain and monitor the system. For E-Leukemia the admin will register and delete users and have full access control over the entire system. The admin will not have access to the passwords.

2.1.4 Environmental requirements

The application needs an active internet connection and it's compatible with Windows 7.8,8.1, and 10 operating system. Browsers such as Microsoft Edge, Mozilla Firefox or Google Chrome are essential for the application to run. The user has the option to verify their account by mobile text message, therefore a valid number is required. The application required the following to build:

- Microsoft Visual Studio
- ASP.NET MVC 5 web application
- The system uses a MS SQL database

The application was built, hosted and tested on local sever but it can be hosted on live servers as Azure Microsoft.

2.1.5 Usability requirements

The interface is easy to use and users will have access to specific views eliminating the confusion for unknown functionalities. (e.g. patients won’t have to see medical staff details)

Patients require basic knowledge of computers usage and no manual instructions will be required. The interface is efficient to use and goals are easy to accomplish navigating through the different tabs from the menu at the top of the web application. Help messages and errors guide users to use the application as it was designed. Some dropdown options will assist users to select the appropriate data.

The MVC architecture separates the model from the view (GUI) and controller (input) allowing communication between user and database.

2.2 Design and Architecture

2.2.1 Introduction

This part will describe how to use E-Leukaemia defined above and explain how the model will be implemented. The information will be provided to describe and define architecture and system design. Each user interface will also be included in this part along with the sequence diagrams.

2.2.2 Constraints

- Graphic User Interface is only in English
- Login and password is used for identification of users that are using the application
- Only registered users (patients, doctors and admin) will be granted access to the application

2.2.3 Assumptions

- patients have an active internet connection
- gave permission to use the information electronically (over the internet)

2.2.4 Application type

E-Leukaemia is a rich application designed for kids with leukaemia. Developed in C# for backend and HTML with bootstrap for front end the web application is using MS SQL to store and centralise data.

2.2.5 Architecture Deployment Type

E- Leukaemia functionality and layers of the system resides on a single server apart from data storage which is hosted on MS Sql database. The application is not sharing its business logic with other applications and only the presentation layer will access the business logic. Because data is stored in a centralised system it will allow the implementation process to focus and improve security, reducing the risk of a data breaches. The SSL connection ensure that data transmitted across the internet is protected and hard to be intercepted by an attacker.

2.3 Security Architecture

Interoperability of E-Leukaemia model allow users to store a large amount of important information which requires security protection. Patients profile contain sensitive information like phone number, date of birth, address and so on. Data theft can compromise confidentiality of medical records and even modify records compromising data integrity. To avoid this security features becomes one of the main keys in the implementation phase.

The view will render the data from a model to an interface and the controller will receive the user inputs but it will never talk directly to database.

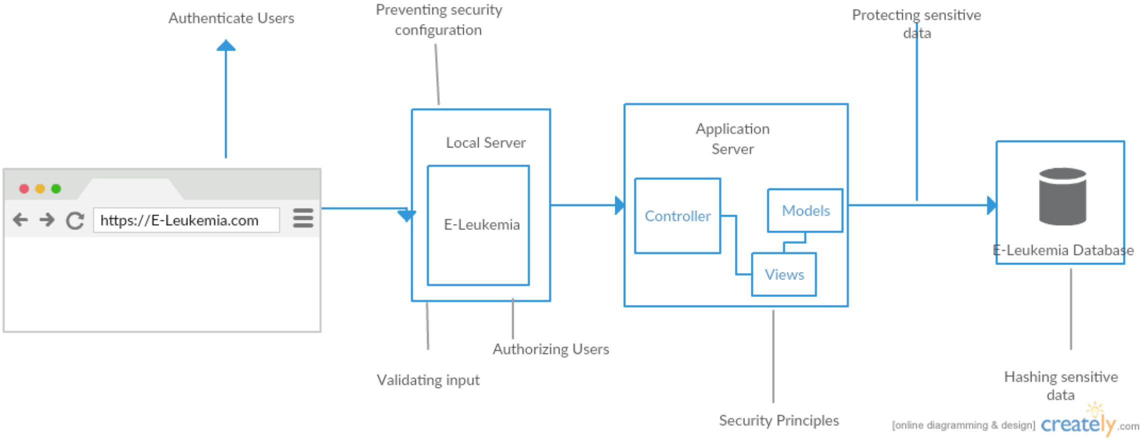
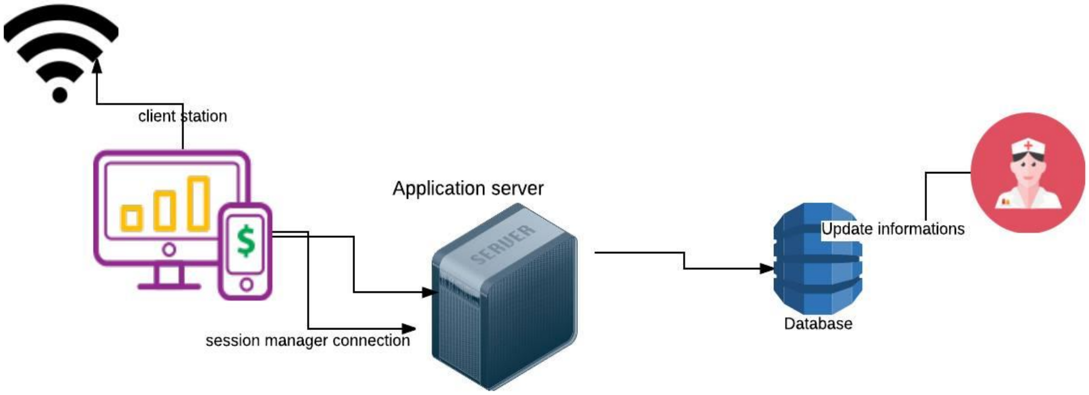


Figure 2 Security Architecture Diagram



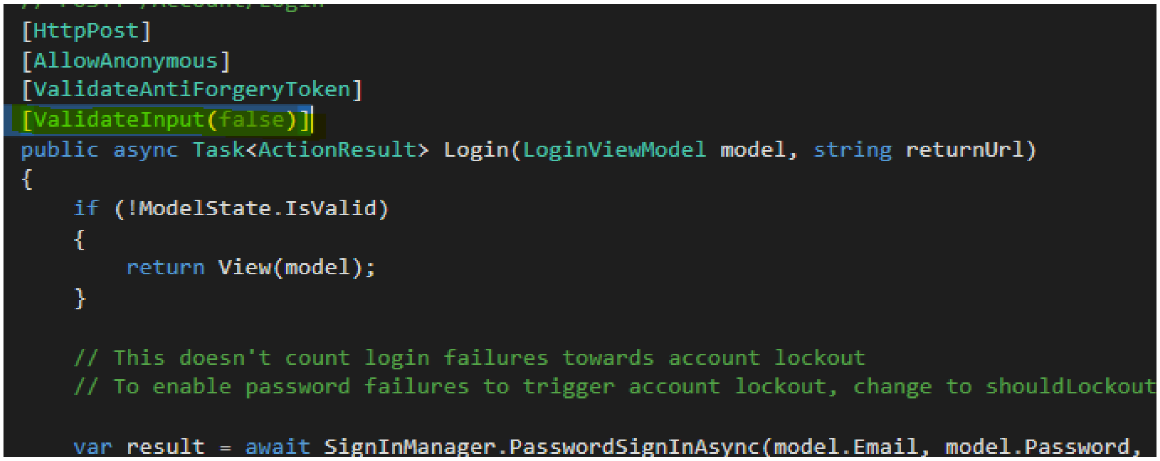
2.4 *Implementation*

2.4.1 Security implementation

Implementation is the most challenging part of this application because security is one of the main concerns outlined from the design phase. Developing a secure application can minimise the risk of a breach that could possible compromise the entire system and in the worse scenario loss of critical data. The main security features implemented in this web application are: input validation, authentication and password management (twillio 2 factor key authentication), hashing sensitive data (password), error handling and logging and users access control.

1. **Preventing Cross Site Scripting**

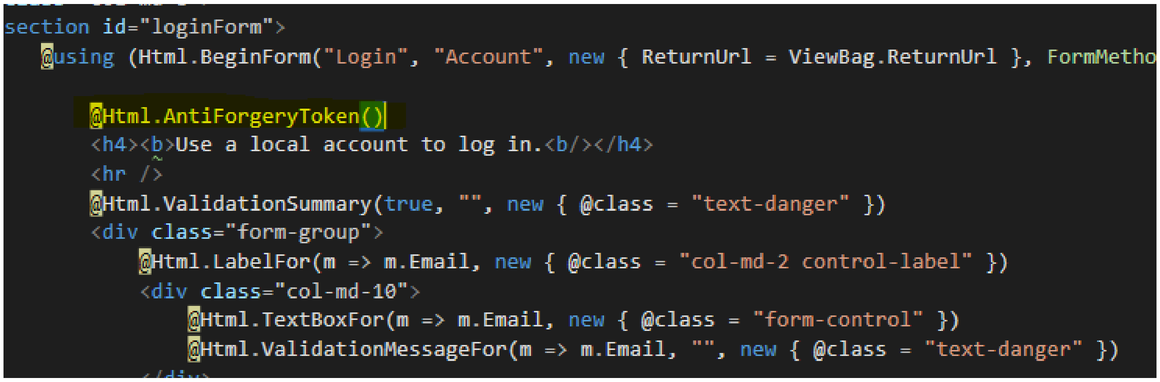
   XXS is an attack performed by an attacker where a malicious code is injected in input fields (text area or text area). To protect the application from these attacks security is implemented to avoid execution of these scripts.



```
// POST: /Account/Login
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
[ValidateInput(false)]
public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    // This doesn't count login failures towards account lockout
    // To enable password failures to trigger account lockout, change to shouldLockout

    var result = await SignInManager.PasswordSignInAsync(model.Email, model.Password,
```

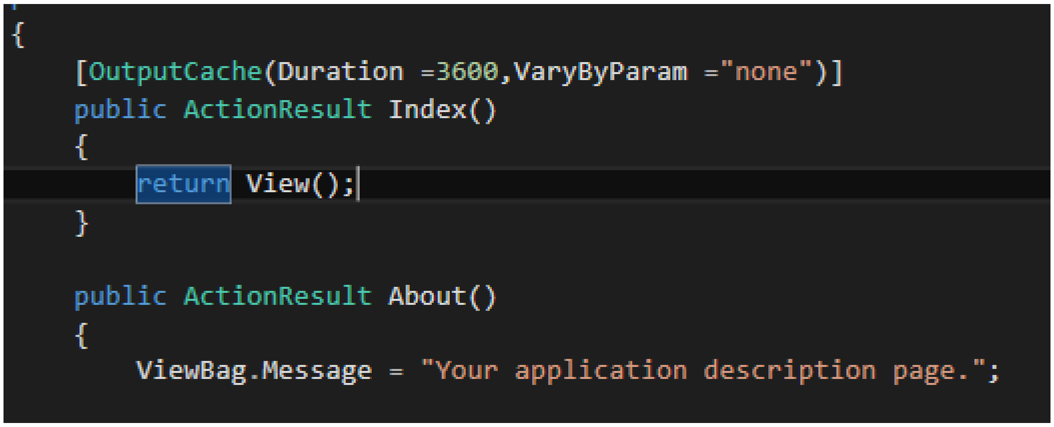
2. **Cross Site Request Forgery(CSRF) Attack**

   CSRF attacks are one of the most popular and they may redirect to some URL and then moves back to your site. Behind this strange behaviour an attacker try to hack the data or attempting to insert invalid data into the web application. Adding the attribute ValidateAntiForgeryToken on the Savedata ActionMethod will validate the post. When submitting the form two tokens, cookie and hidden field are sent over to the server that are validated if both present, otherwise the request is not allowed. ASP.NET includes a set of solutions that will detect and block CSRF using specific tokens features. It’s important to know when Anti forgery tokens are implemented users should accept cookies otherwise the requests are not performed.



```
section id="loginForm">
   @using (Html.BeginForm("Login", "Account", new { ReturnUrl = ViewBag.ReturnUrl }, FormMetho

       @Html.AntiForgeryToken()
       <h4><b>Use a local account to log in.<b/></h4>
       <hr />
       @Html.ValidationSummary(true, "", new { @class = "text-danger" })
       <div class="form-group">
           @Html.LabelFor(m => m.Email, new { @class = "col-md-2 control-label" })
           <div class="col-md-10">
               @Html.TextBoxFor(m => m.Email, new { @class = "form-control" })
               @Html.ValidationMessageFor(m => m.Email, "", new { @class = "text-danger" })
```

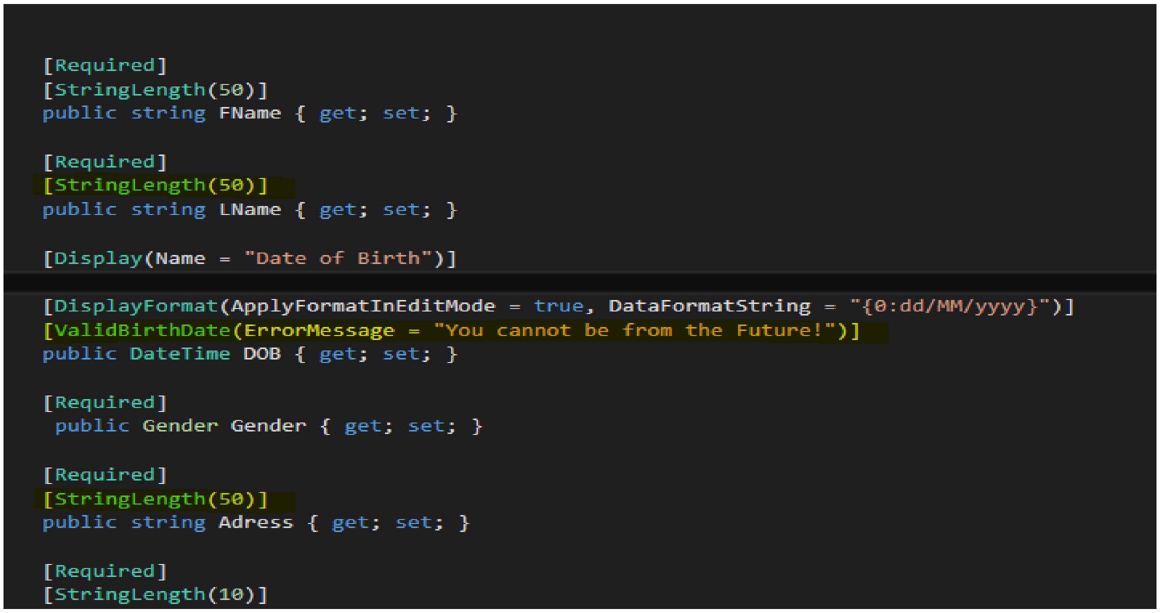
3. The application performance can be improved by catching stale data. The bandwidth between the server and client is reduced and the action of the webpage it can be cached. Cache Pages usually should be applied to pages that contain shared data or are open to public and don’t require authorisation.



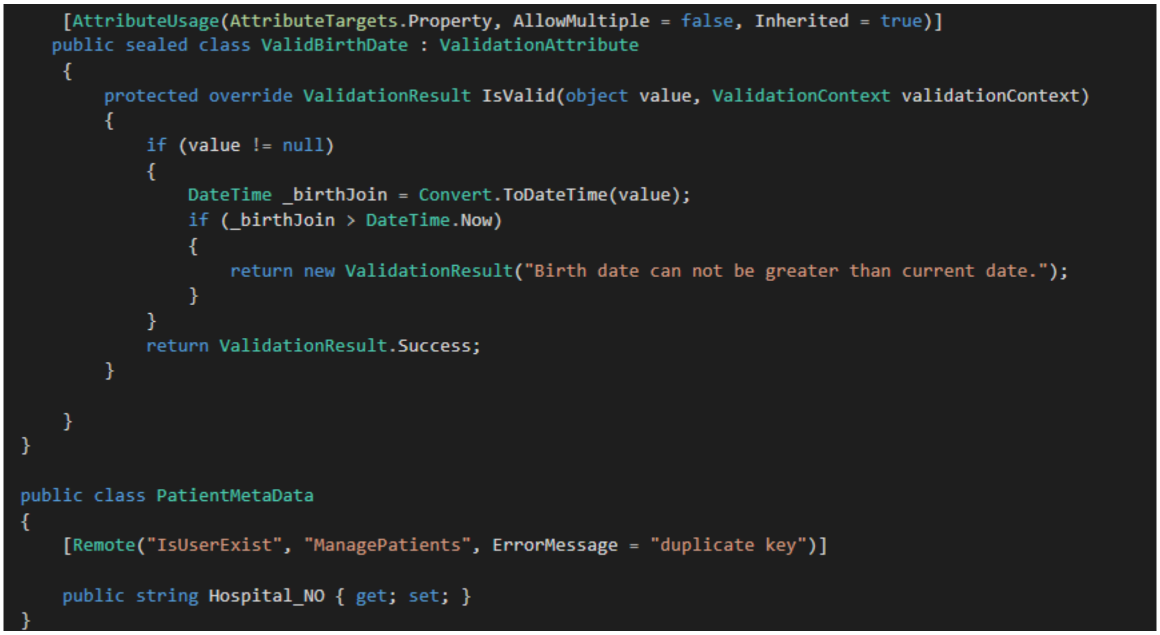
```
{
     [OutputCache(Duration =3600,VaryByParam ="none")]
     public ActionResult Index()
     {
         return View();
     }

     public ActionResult About()
     {
         ViewBag.Message = "Your application description page.";
```

4. **Input Validation**

   Declarative Validation allows adding attributes to the proprieties from the model which it will enable JavaScript to support client-side validation. When the input box requires a string then only strings will be granted to type in. Metadata is another type of attribute associated to a class and the model can be generated while the attribute is in a separate class. This feature is helpful as validation control can be handled safer as it is an alternative way for validation and it’s more secure.



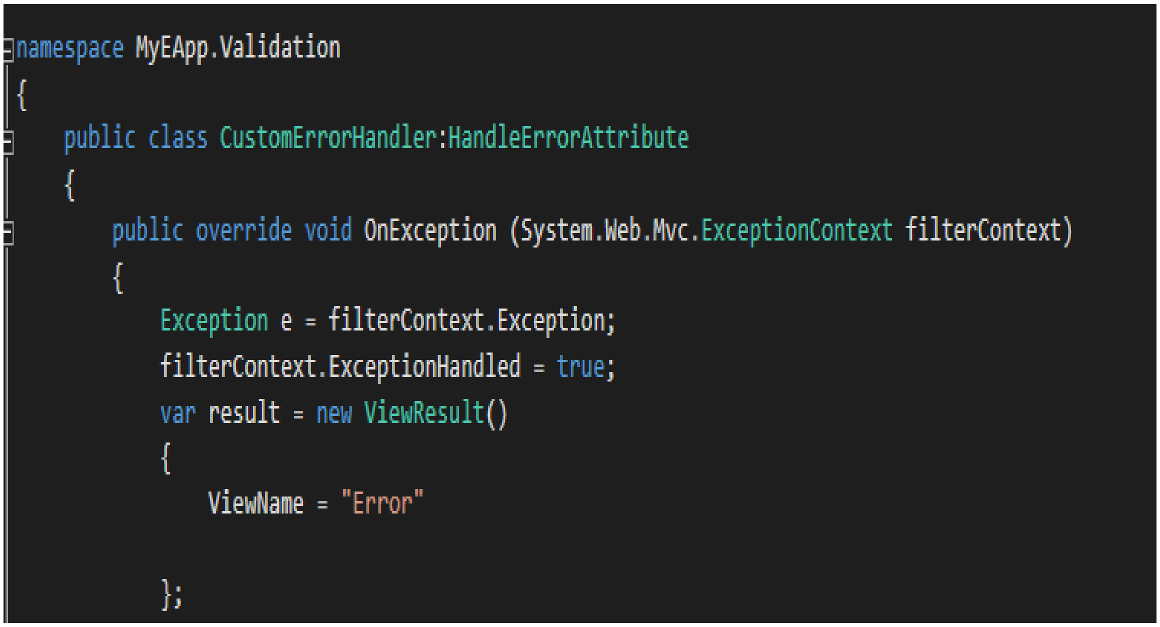
```
[Required]
[StringLength(50)]
public string FName { get; set; }

[Required]
[StringLength(50)]
public string LName { get; set; }

[Display(Name = "Date of Birth")]

[DisplayFormat(ApplyFormatInEditMode = true, DataFormatString = "{0:dd/MM/yyyy}")]
[ValidBirthDate(ErrorMessage = "You cannot be from the Future!")]
public DateTime DOB { get; set; }

[Required]
 public Gender Gender { get; set; }

[Required]
[StringLength(50)]
public string Adress { get; set; }

[Required]
[StringLength(10)]
```



```
    [AttributeUsage(AttributeTargets.Property, AllowMultiple = false, Inherited = true)]
   public sealed class ValidBirthDate : ValidationAttribute
    {
        protected override ValidationResult IsValid(object value, ValidationContext validationContext)
        {
            if (value != null)
            {
                DateTime _birthJoin = Convert.ToDateTime(value);
                if (_birthJoin > DateTime.Now)
                {
                    return new ValidationResult("Birth date can not be greater than current date.");
                }
            }
            return ValidationResult.Success;
        }

    }
}

public class PatientMetaData
{
    [Remote("IsUserExist", "ManagePatients", ErrorMessage = "duplicate key")]

    public string Hospital_NO { get; set; }
}
```

5. Custom Error Handler

Internal error messages in the web applications can be exploited by an attacker. To avoid that internal errors should be hidden by implementing custom error handler. The below image display custom error handler implemented in this application and is applied globally for entire application.



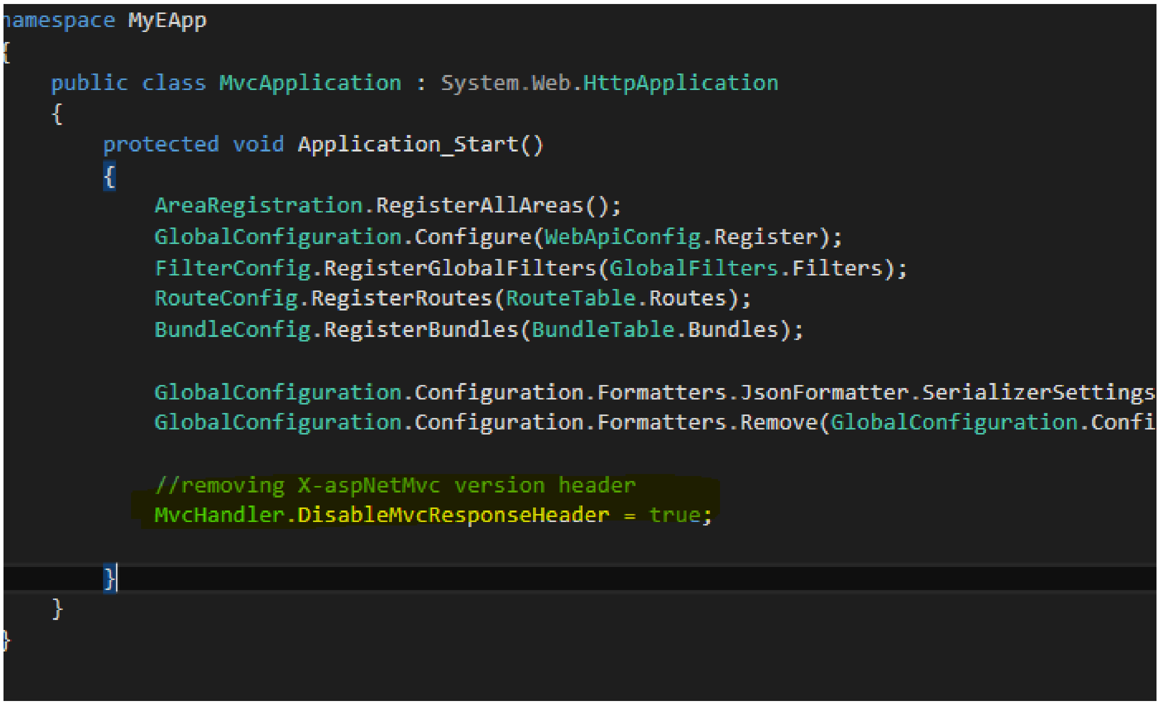
```
namespace MyEApp.Validation
{
    public class CustomErrorHandler:HandleErrorAttribute
    {
        public override void OnException (System.Web.Mvc.ExceptionContext filterContext)
        {
            Exception e = filterContext.Exception;
            filterContext.ExceptionHandled = true;
            var result = new ViewResult()
            {
                ViewName = "Error"

            };
```

6. **Hashing the password**

   Users can register only one account for each person. Admin will have the authority to register users so a password will be provided. System allow the users to change the password and 2 factor key authentication is implemented which it will send a token key to the user phone when they will login. All the passwords are encrypted. Users passwords are protected by using MD5 class which returns an array of bytes.

7. **Version discloser**

Version information can be exploited by an attacker to conduct a specific attack on version discloser. Version discloser shows information about the server, what framework is being used and what ASP.NET MVC version is implemented. The E-Leukaemia web application implemented a security feature to protect disclosing these details as it can be seen below.



```
namespace MyEApp
{
    public class MvcApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            GlobalConfiguration.Configure(WebApiConfig.Register);
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);

            GlobalConfiguration.Configuration.Formatters.JsonFormatter.SerializerSettings
            GlobalConfiguration.Configuration.Formatters.Remove(GlobalConfiguration.Confi

            //removing X-aspNetMvc version header
            MvcHandler.DisableMvcResponseHeader = true;

        }
    }
}
```

8. Users access control

In MVC all controllers by default are accessible to anonymous users either are logged or not. To secure these actions Authorize attribute can be used. The E-Leukaemia web application has 4 types of users: anonymous, patients, medical staff and administrator. Anonymous can access home, about and contact pages and they cannot log in the system if they are not registered. All the registered users can get access to the application and they will access certain pages as in below screenshot. The registration process is restricted to admin and on the top of ActionResult Register the authorize condition is added hence only the admin has the permission to register new users.

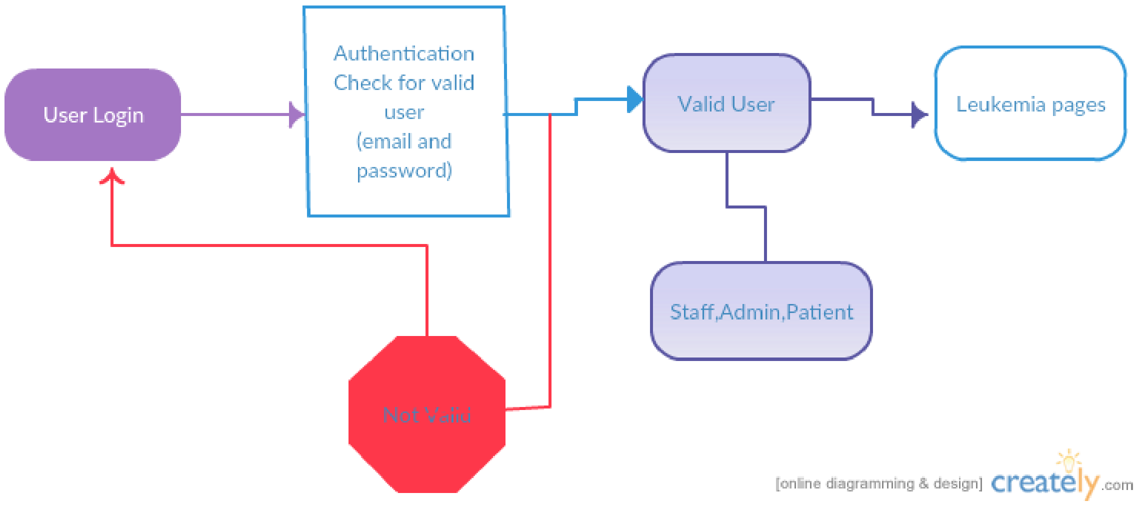
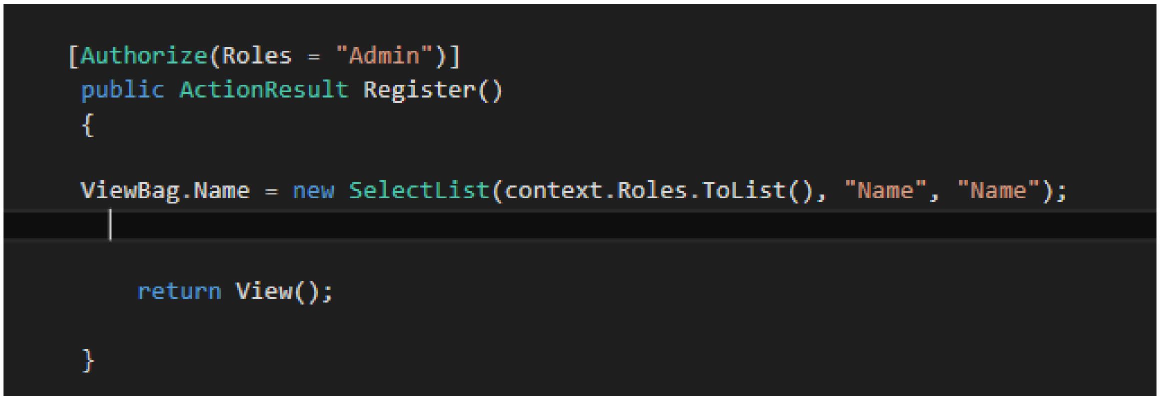
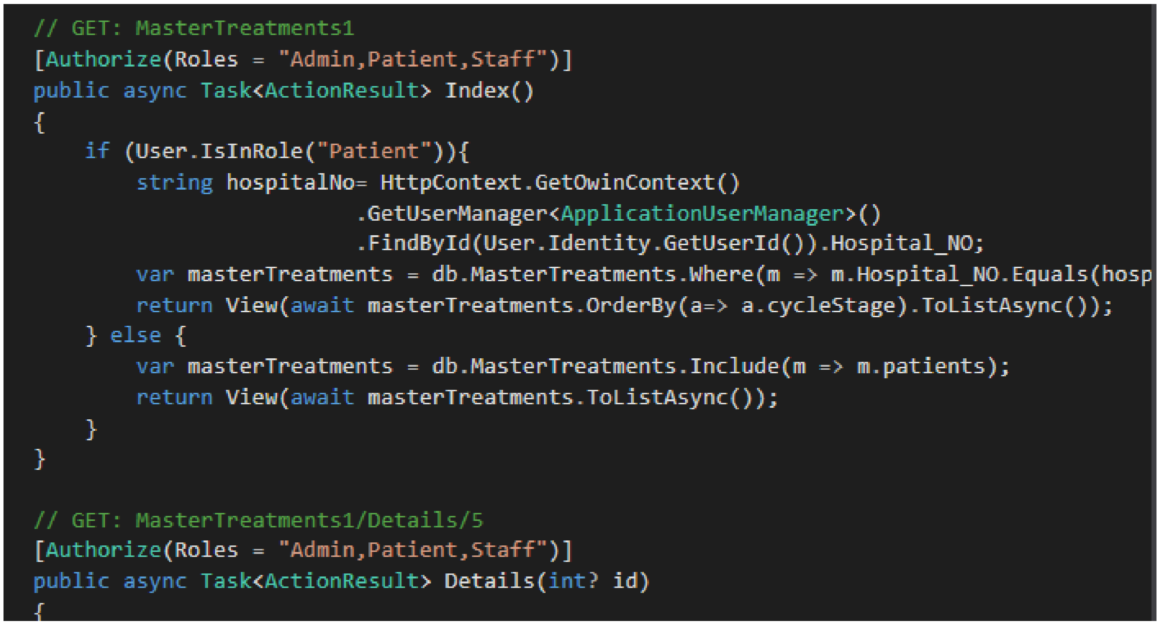

User Login
Authentication Check for valid user (email and password)
Valid User
Leukemia pages
Staff,Admin,Patient
Not Valid
[online diagramming & design] creately.com

Figure 3 Access control



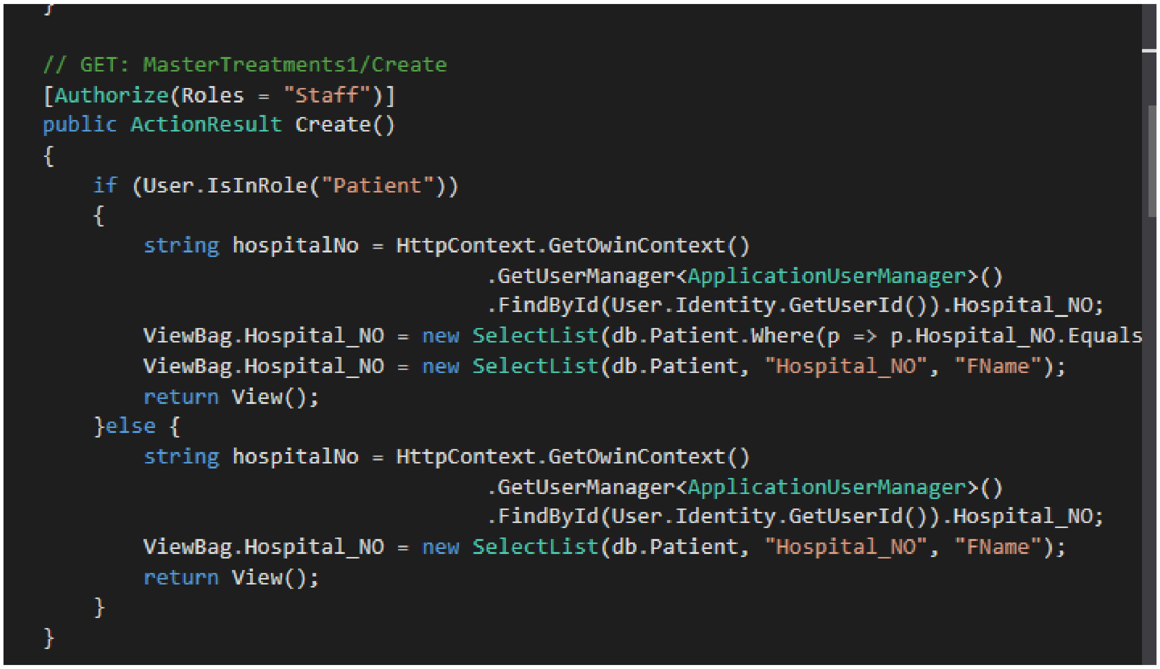
```
[Authorize(Roles = "Admin")]
 public ActionResult Register()
 {

 ViewBag.Name = new SelectList(context.Roles.ToList(), "Name", "Name");


     return View();

 }
```

For Master Treatments action patient, medical staff and admin can read the information.



```
// GET: MasterTreatments1
[Authorize(Roles = "Admin,Patient,Staff")]
public async Task<ActionResult> Index()
{
    if (User.IsInRole("Patient")){
        string hospitalNo= HttpContext.GetOwinContext()
                         .GetUserManager<ApplicationUserManager>()
                         .FindById(User.Identity.GetUserId()).Hospital_NO;
        var masterTreatments = db.MasterTreatments.Where(m => m.Hospital_NO.Equals(hosp
        return View(await masterTreatments.OrderBy(a=> a.cycleStage).ToListAsync());
    } else {
        var masterTreatments = db.MasterTreatments.Include(m => m.patients);
        return View(await masterTreatments.ToListAsync());
    }
}

// GET: MasterTreatments1/Details/5
[Authorize(Roles = "Admin,Patient,Staff")]
public async Task<ActionResult> Details(int? id)
{
```

Only authorised people can update the treatment for the patients as they have the knowledge and access to patient records. This is crucial for patients and restricting the action to medical staff is increasing the security of the application. Patients will not have privileges to some specific actions or other important information (e.g. medical staff details).



```
}

// GET: MasterTreatments1/Create
[Authorize(Roles = "Staff")]
public ActionResult Create()
{
    if (User.IsInRole("Patient"))
    {
        string hospitalNo = HttpContext.GetOwinContext()
                                    .GetUserManager<ApplicationUserManager>()
                                    .FindById(User.Identity.GetUserId()).Hospital_NO;
        ViewBag.Hospital_NO = new SelectList(db.Patient.Where(p => p.Hospital_NO.Equals
        ViewBag.Hospital_NO = new SelectList(db.Patient, "Hospital_NO", "FName");
        return View();
    }else {
        string hospitalNo = HttpContext.GetOwinContext()
                                    .GetUserManager<ApplicationUserManager>()
                                    .FindById(User.Identity.GetUserId()).Hospital_NO;
        ViewBag.Hospital_NO = new SelectList(db.Patient, "Hospital_NO", "FName");
        return View();
    }
}
```

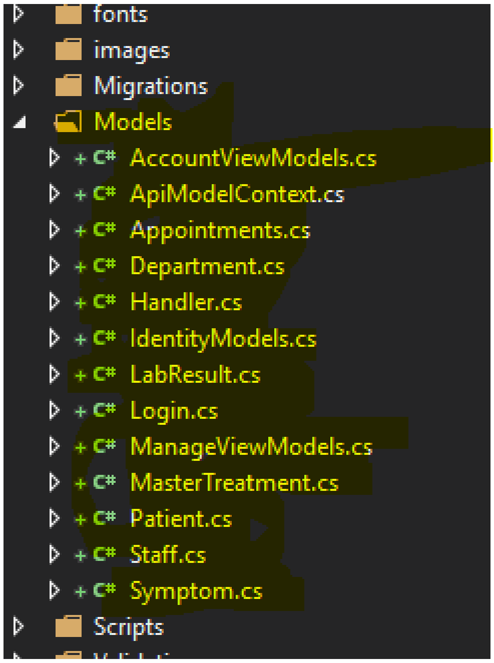
Role based architecture control will restrict and grant permissions and is one of the most successful security schemes in software development area.

9. Other details about security features

The ASP.NET framework is using Identity systems which is concerned with OAuth integration, OWIN and user profiles that are included in NutGet package plugin. This powerful combination of these features is handling the process of authentication for users and is increasing security level in E-Leukaemia web application.

2.4.2 Code Implementation

The project was implemented using Visual studio 2015 editor and I first built on my models that populated the tables in the database.



```
fonts
images
Migrations
Models
    AccountViewModels.cs
    ApiModelContext.cs
    Appointments.cs
    Department.cs
    Handler.cs
    IdentityModels.cs
    LabResult.cs
    Login.cs
    ManageViewModels.cs
    MasterTreatment.cs
    Patient.cs
    Staff.cs
    Symptom.cs
Scripts
```

Figure 4 Model

I designed the models based on user's preferences. The application is using two databases ASPNET Identity, one that's built in and is handling authentication and application cookies, and the second database DBContext is handling the rest of functionalities from the system. By default registration in ASP.NET is using email and password but in this project hospital number is added for registration process.

```
  [HttpPost]

        [Authorize(Roles = "Admin")]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult> Register(RegisterViewModel model)
        {
            if (ModelState.IsValid)
            {
                 var user = new ApplicationUser { UserName = model.Email, Email =
model.Email, Hospital_NO = model.Hospital_NO };
```

```
                var result = await UserManager.CreateAsync(user, model.Password);
                if (result.Succeeded)
                {

                    await SignInManager.SignInAsync(user, isPersistent:false, 
rememberBrowser:false);
                    await UserManager.AddToRoleAsync(user.Id, model.UserRoles);
                    return RedirectToAction("Index", "Home");
                }

                ViewBag.Name = new SelectList(context.Roles.Where(u 
=>!u.Name.Contains("Admin")).ToList(), "Name", "Name");
                AddErrors(result);
            }
```

Users are stored in ApplicationUser which is checking registration model (email, hospital_NO) and after that these details are passed onto UserManager which is storing the user and password model. This method is authorizing admin to register users. Hospital_NO is a foreign key in DBContext that it will allow the application to run queries.

For the appointments implementation in controller the method is checking if the user is a patient allow the user to see data related to his account using UserManager which is holding information about users. Users identity is filtrated by checking their Hospital_NO as this is a unique identifier for each individual user.

```
        // GET: Appointments
        public async Task<ActionResult> Index()
        {
            if (User.IsInRole("Patient"))
            {
                string hospitalNo = HttpContext.GetOwinContext()
                                 .GetUserManager<ApplicationUserManager>()
                                 .FindById(User.Identity.GetUserId()).Hospital_NO;
                var appointments = db.Appointments.Where(a => 
a.P_Hospital_NO.Equals(hospitalNo)).Include(a => a.Department).Include(a => 
a.Patient).Include(a => a.Staff);

                return View(await appointments.ToListAsync());
            }else
            {
                var appointments = db.Appointments.Include(a =>
a.Staff).OrderByDescending(a => a.Date);
                return View(appointments.ToList());
            }
        }
```

The appointments implementation includes a series of check-ups to avoid duplicate appointments or time overlapping for timeslots.

```
  // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [Authorize(Roles = "Admin,Patient,Staff")]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult> Create([Bind(Include =
"AppointmentID,UserID,P_Hospital_NO,S_Hospital_NO,DepartmentID,Date,Time,Descripti
on")] Appointments appointments)
        {
            string userId = User.Identity.GetUserId().ToString();
            appointments.UserID = userId;
            if (ModelState.IsValid)
            {
                List<Appointments> ap = db.Appointments.ToList();
                bool dateUnavailable = false;
                foreach(Appointments a in ap)
                {
                    if((a.Date.CompareTo(appointments.Date) == 0)
                        && (a.Time.CompareTo(appointments.Time)==0)
                        &&((a.S_Hospital_NO.CompareTo(appointments.S_Hospital_NO)
== 0)))
                    {
                        dateUnavailable = true;
                        // break;

                        ModelState.AddModelError("", "errrorrrr appointment
already taken");
                    }

                }

                if (dateUnavailable == false)
                {
                    db.Appointments.Add(appointments);
                    await db.SaveChangesAsync();
                    return RedirectToAction("Index");
                }
            }

            string hospitalNo = HttpContext.GetOwinContext()
                             .GetUserManager<ApplicationUserManager>()
                             .FindById(User.Identity.GetUserId()).Hospital_NO;
            ViewBag.DepartmentID = new SelectList(db.Department, "DepartmentID", 
"DepartmentName");
            ViewBag.P_Hospital_NO = new SelectList(db.Patient.Where(p => 
p.Hospital_NO.Equals(hospitalNo)), "Hospital_NO", "FName");
            ViewBag.S_Hospital_NO = new SelectList(db.Staffs, "Hospital_NO", 
"LastName");
            return View(appointments);
        }
```

This method is creating timeslots of 30 minutes for each patient.

```
public class MyTimeValidation:ValidationAttribute
    {
        public override bool IsValid(object value)
        {
            DateTime dt = (DateTime)value;
            if (dt.Minute == 30 || dt.Minute == 00)
                return true;
            else
                return false;
        }
    }
```

The below method allow medical staff to create new patients records and in order to avoid concurrency in the database a JSON method is implemented to do a check if the patient hospital_NO exists in the database for another user.

```
  // GET: ManagePatients/Create
        [Authorize(Roles = "Staff,Admin")]
        public ActionResult Create()
        {
            ViewBag.FirstName = new SelectList(db.Staffs, "Hospital_NO", 
"FirstName");
            return View();

    }

        public JsonResult IsUserExist(string Hospital_NO)
        {
            if (Hospital_NO != null)
            {
                return Json(!db.Patient.Any(patient => patient.Hospital_NO == 
Hospital_NO), JsonRequestBehavior.AllowGet);
            }
            return Json("name already taken", JsonRequestBehavior.AllowGet);
        }
```

Custom error handles are developed in the application to hide internal errors which it will minimise an attacker to break in the system to discover vulnerabilities and later to exploit them.

```
public class CustomErrorHandler:HandleErrorAttribute
    {
        public override void OnException (System.Web.Mvc.ExceptionContext 
filterContext)
        {
            Exception e = filterContext.Exception;
            filterContext.ExceptionHandled = true;
            var result = new ViewResult()
            {
                ViewName = "Error"

            };

            result.ViewBag.Error = "Error Occur While Processing Your Request 
Please Check After Some Time";
       // ViewBag.Error = "~/Content/images/Error.jpg";
            filterContext.Result = result;
        }
    }
}
```

The master treatment method in view page is checking all the treatments by the expiring dates and if it finds any it's highlighting them in red helping the user to focus on current treatment. On the bottom, another condition is implemented restricting the access to certain actions for a patient.

```
                <tr style=@((item.DateEnd - DateTime.Today).Days >3 ? null : 
"Background-color:Red" )>
                    <td>
                        @Html.DisplayFor(modelItem => item.patients.FName)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.cycleStage)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.DateStart)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.DateEnd)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.medicationNames)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.Dosage)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.typeTreatments)
                    </td>

          @if (!User.IsInRole("Patient"))
          {
                    <td>
                        @Html.ActionLink("Edit", "Edit", new { id =
item.MasterTreatmentId }) |
                        @Html.ActionLink("Details", "Details", new { id =
item.MasterTreatmentId }) |
                        @Html.ActionLink("Delete", "Delete", new { id =
item.MasterTreatmentId })
                    </td>
          }
                </tr>
                }
```

Because the application has an increased number of functionalities only specific links will be visible to users based on their roles.

```
<div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li>@Html.ActionLink("Home", "Index", "Home")</li>
                    <li>@Html.ActionLink("About", "About", "Home")</li>
                    <li>@Html.ActionLink("Contact", "Contact", "Home")</li>

                    @if (User.IsInRole("Admin,Staff")) {
                    <li>@Html.ActionLink("Manage Patient", "Index",
"ManagePatients")</li>
                    <li>@Html.ActionLink("LabResults", "Index", "LabResults")</li>
                    <li>@Html.ActionLink("MasterTreatments", "Index",
"MasterTreatments")</li>
                    <li>@Html.ActionLink("Appointments", "Index",
"Appointments")</li>
                    <li>@Html.ActionLink("LabResults", "Index", "LabResults")</li>

                    }
                    else if (User.IsInRole("Patient"))
                    {

                    <li>@Html.ActionLink("Appointments", "Index", new { Controller
= "Appointments", Action = "Create" })</li>
```

```
                    <li>@Html.ActionLink("Symptoms", "Index", new { Controller =
"Symptoms", Action = "Create" })</li>
                    <li>@Html.ActionLink("ManagePatient", "Index", new {
Controller = "ManagePatients", Action = "Details" })</li>
                    <li>@Html.ActionLink("MasterTreatments", "Index",new {
Controller = "MasterTreatments", Action = "Details" })</li>
                    <li>@Html.ActionLink("LabResults", "Index", new { Controller =
"LabResults", Action = "Details" })</li>
                    }

                </ul>
```

2.5 System description

The aim of E-Leukaemia platform is to design a platform to help users to manage self-care management and health guidance for leukaemia treatment. Additionally, the application will implement different basic roles with different access control actions. The application will also have basic function modules available to public visitors (home, about and contact pages). Another important aspect is collecting information such as appointments booked by the patients and symptoms submitted by the patients. The symptom module will allow patients to track temperature, sides effects during the chemo cycle and moreover take an instant picture (e.g. blisters, skin rash) and upload it to the application. On the other side, medical staff can collect this important information and further analyse the cause of the symptom. This is critical information as some patients can have serious side effects for specific medication and based on that medical staff can make decisions on changing the medication. Medical staff can present information to the patients by creating new treatment plans, new appointments and Lab results for each individual patient which is available at the other end based on individual account. Patients can’t see other patients records or staff details. Personal privacy and data security is implemented throughout the system ensuring the availability and integrity. The system will offer the following functionality:

1. The application offer patient’s self-management for leukaemia treatment where the user interacts with the system to retrieve information they need for the leukaemia treatment.
2. Transparent platform allows the application the exchange and share information based on user’s roles. Both users, patient and medical staff can create blocks of information (e.g. symptom, appointments, lab results) that it will be saved on database then retrieved by the authorised users.
3. Assigning authority can be performed by the administrator. The role management will provide 3 roles which are patient, medical staff and admin.
   - Patients can read only treatments, lab results and their profile. Patients can create and submit symptoms and appointments. For symptoms function they can upload pictures, this is optionally.
   - The medical staff can read symptoms submitted by the patients. They have CRUD authority for master treatments, appointments, lab results and patient’s profiles.
4. The admin has the authority to view all the Patients and medical staff information, they can maintain and update database and register users. The admin will not have access to user’s passwords as the passwords are encrypted.
5. Data collection is focusing on user’s personal data and the application is implementing methods to facilitate users to enter information in easier manner with additional guidance (e.g. drop down selection, date picker)
6. Concise view is important for the users because the information should be presented in a way to be clear with no errors
7. Security is crucial for any user because of data confidentiality. The information is protected by implementing protection measures ensuring is not available to the public

8. Reliability is another key aspect of this application as the app will be used for long term (2-3 or more years) hence the system has a reliable performance.

2.5.1 Local database and cloud database

The application can be hosted on cloud as a service platform and one option would be to use Azure cloud database to host this project.

During the development, the database was hosted on Microsoft SQL Server. Hosting the application on cloud is more reliable and flexible and the users the access it at any time whenever they want from anywhere. In the E-Leukaemia app the number of users can increase and storage will be huge. Is important to take in consideration that if too many users login in the app at the same time may create a bottleneck situation but the in the cloud this will not occur and database will be a lot faster. The application can be accessed online at http://e-leukemia.azurewebsites.net/ and it's faster and reliable. The database is also hosted on the Azure Cloud which is increasing the reliability and computation capability of the system.

2.6 System Diagram

The E-Leukaemia frame relies to browser-server architecture where the browser has the highest priority. Model-view-controller is a software design implemented in E-Leukaemia. Model is the lowest level of the pattern and is storing data. View is displaying data to users. Controller is the middleware between View and Model. These three elements separate the application logic from the user interface. Controller is responsible with all the requests for the web application and collaborates with the model who is preparing the data requested by the view. The

View display the data that was prepared by the controller. The MVC architecture is represented as below.

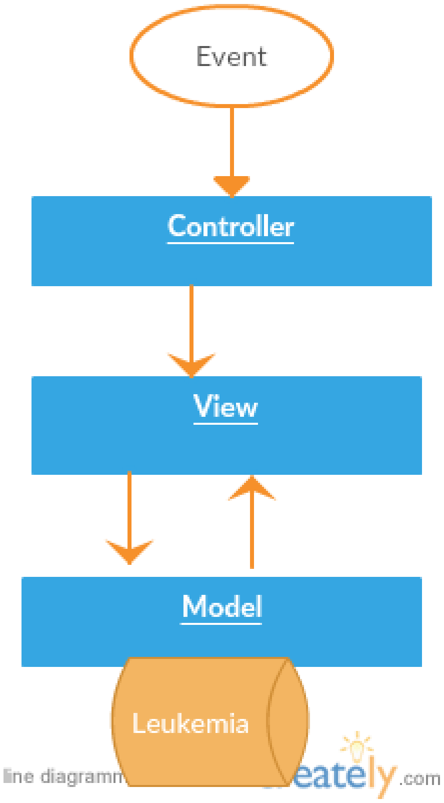


Figure 5 System Architecture

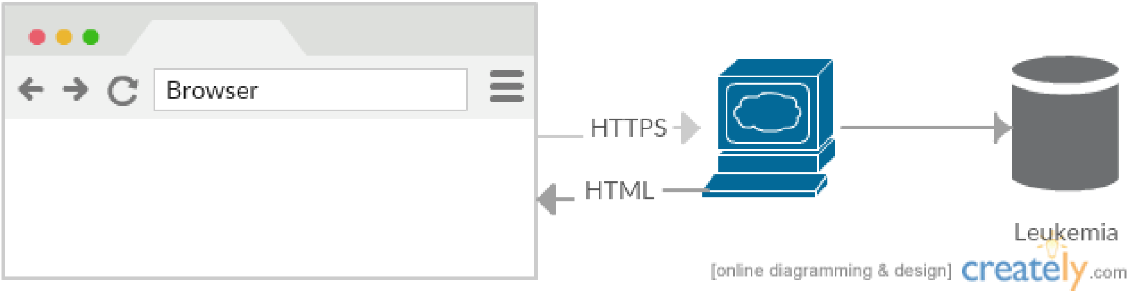
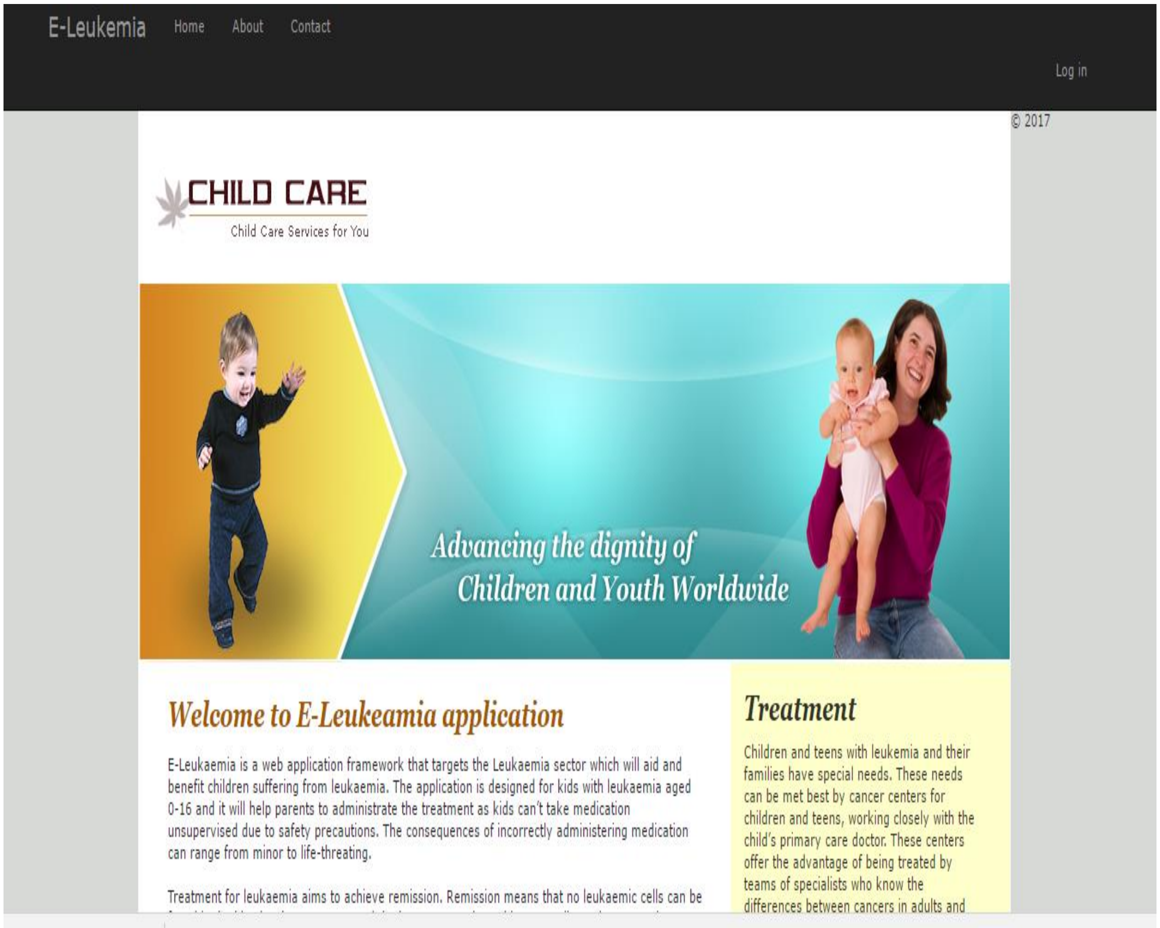


Figure 6 Browser/Server Architecture

2.7 Graphical User Interface (GUI) Layout

The E-Leukaemia home page is available for any visitor and they can access only three tabs from the application (Home, About and Contact). The web application is easy to navigate, comfortable for all users and has a responsive design. The responsive design is allowing the application to easily adapt the screen size on any device due to bootstrap technology implemented. The home page display general information about leukaemia disease and contains three links to about and contact page.



Welcome to E-Leukeamia application

E-Leukaemia is a web application framework that targets the Leukaemia sector which will aid and benefit children suffering from leukaemia. The application is designed for kids with leukaemia aged 0-16 and it will help parents to administrate the treatment as kids can't take medication unsupervised due to safety precautions. The consequences of incorrectly administering medication can range from minor to life-threating.

Treatment for leukaemia aims to achieve remission. Remission means that no leukaemic cells can be

Treatment

Children and teens with leukemia and their families have special needs. These needs can be met best by cancer centers for children and teens, working closely with the child's primary care doctor. These centers offer the advantage of being treated by teams of specialists who know the differences between cancers in adults and

Figure 7 Home page

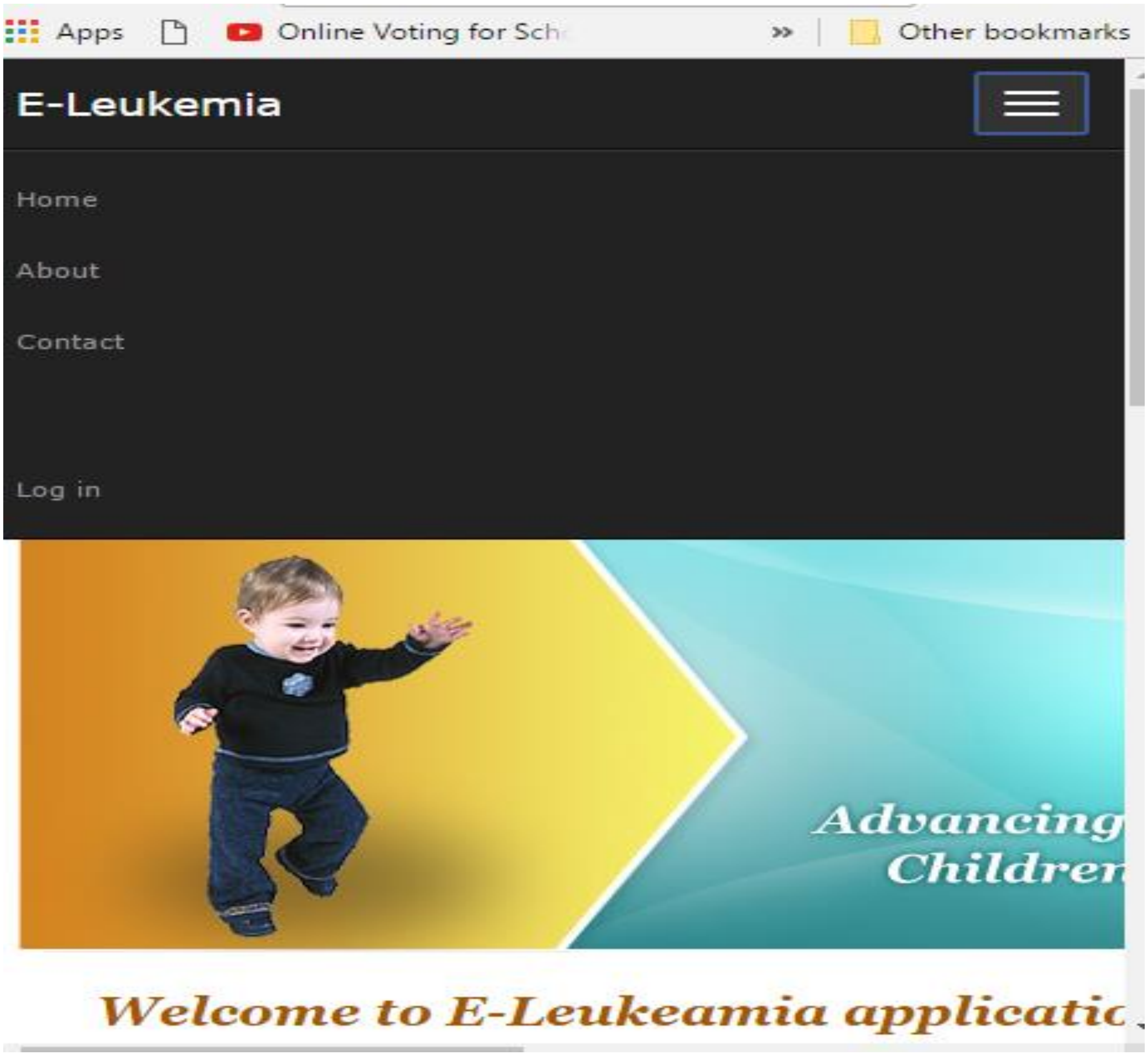
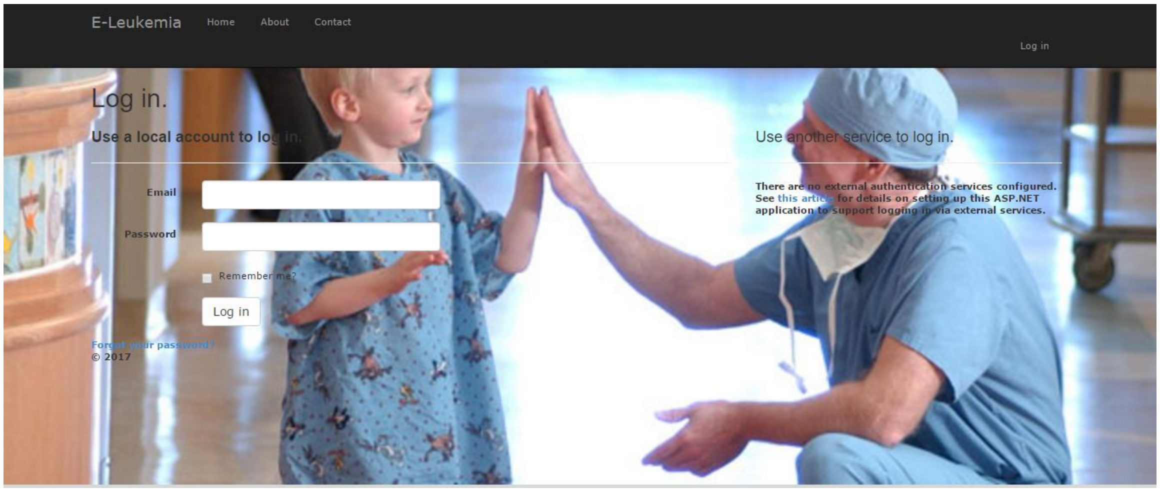


Figure 8 Home page from a mobile device

Login page for registered Users

Login structure will allow users to log into the system using their email address and password. Password can be changed if required and a validation request will be sent to their email for validation hence valid email address is mandatory. The page will redirect users to specific pages after user validation and authentication.

The login page requires users to enter their email address and password. If user forgets his password the application allows users to create a new one but a valid email address is necessary as a reset link will be sent to the email. The password will be encrypted and stored into database.



Log in.

Use a local account to log in.

Use another service to log in.

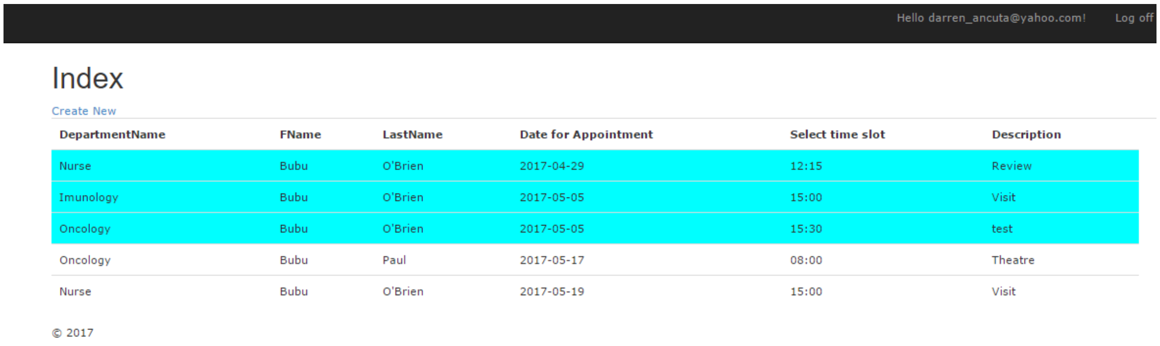
There are no external authentication services configured. See this article for details on setting up this ASP.NET application to support logging in via external services.

© 2017

Figure 9 Login Page

Appointments page

Patients can see the upcoming appointments and they also can request appointments. When an appointment is selected, the system will check if the selected medical staff is available and if there is a free time slot at requested time. The timeslot duration is 30 minutes long and the system will highlight in blue the past appointments.



Hello darren_ancuta@yahoo.com! Log off

Index

Create New

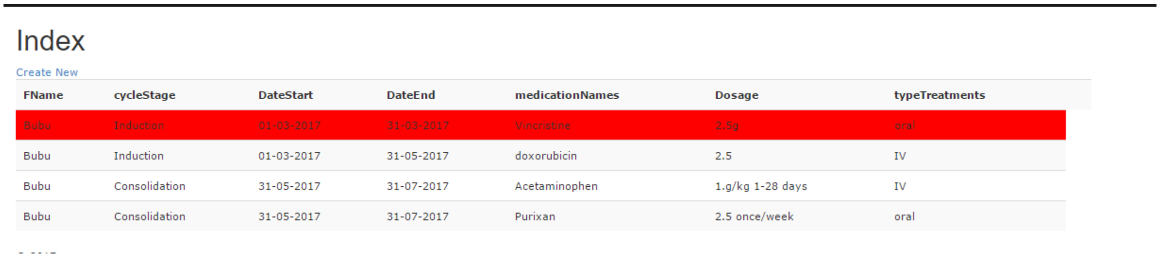
DepartmentName	FName	LastName	Date for Appointment	Select time slot	Description
Nurse	Bubu	O'Brien	2017-04-29	12:15	Review
Imunology	Bubu	O'Brien	2017-05-05	15:00	Visit
Oncology	Bubu	O'Brien	2017-05-05	15:30	test
Oncology	Bubu	Paul	2017-05-17	08:00	Theatre
Nurse	Bubu	O'Brien	2017-05-19	15:00	Visit

© 2017

Figure 10 Appointments

Master treatments

Patients can get information about their treatment using the system and past treatments are highlighted with red. This electronic information is centralized on this page and is replacing the paper version. The patients can't edit, create or deleted treatments.



Index

Create New

FName	cycleStage	DateStart	DateEnd	medicationNames	Dosage	typeTreatments
Bubu	Induction	01-03-2017	31-03-2017	Vincristine	2.5g	oral
Bubu	Induction	01-03-2017	31-05-2017	doxorubicin	2.5	IV
Bubu	Consolidation	31-05-2017	31-07-2017	Acetaminophen	1.g/kg 1-28 days	IV
Bubu	Consolidation	31-05-2017	31-07-2017	Purixan	2.5 once/week	oral

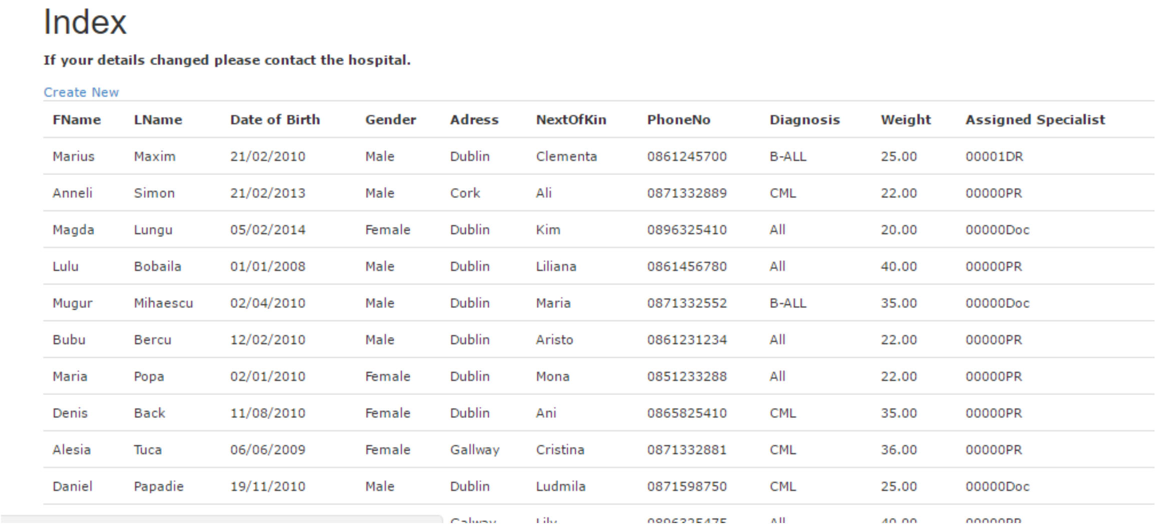
Figure 11 Treatments

Patient profile

This page display a summary of patient profile and patients cannot modify or delete any details from their record, only medical staff can update this information.

Admin control

Admin can have full access to the system and he can register users if they are already attending the hospital. Patients that are attending the hospital are already recorded in hospital database and if parents wish to avail this application they can request a registration account to get access to the application.



Index

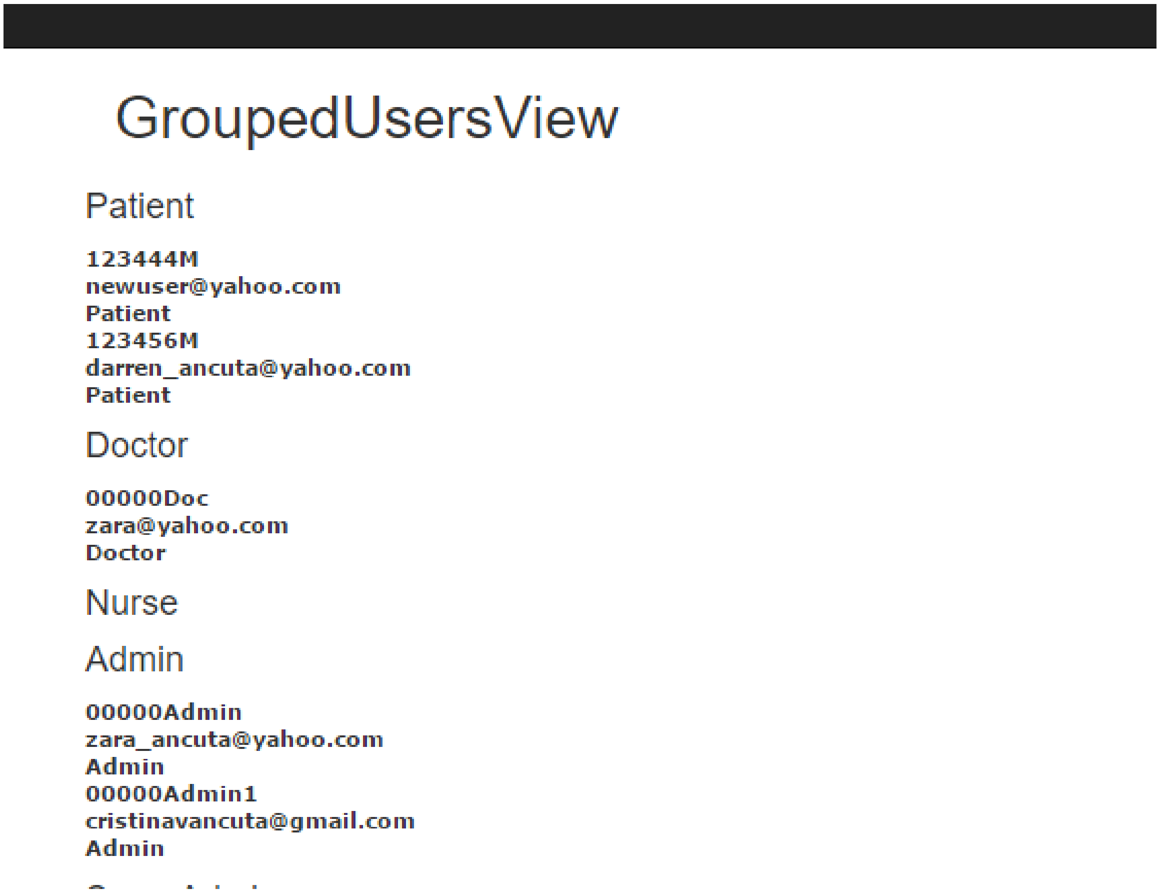
If your details changed please contact the hospital.

Create New

FName	LName	Date of Birth	Gender	Adress	NextOfKin	PhoneNo	Diagnosis	Weight	Assigned Specialist
Marius	Maxim	21/02/2010	Male	Dublin	Clementa	0861245700	B-ALL	25.00	00001DR
Anneli	Simon	21/02/2013	Male	Cork	Ali	0871332889	CML	22.00	00000PR
Magda	Lungu	05/02/2014	Female	Dublin	Kim	0896325410	All	20.00	00000Doc
Lulu	Bobaila	01/01/2008	Male	Dublin	Liliana	0861456780	All	40.00	00000PR
Mugur	Mihaescu	02/04/2010	Male	Dublin	Maria	0871332552	B-ALL	35.00	00000Doc
Bubu	Bercu	12/02/2010	Male	Dublin	Aristo	0861231234	All	22.00	00000PR
Maria	Popa	02/01/2010	Female	Dublin	Mona	0851233288	All	22.00	00000PR
Denis	Back	11/08/2010	Female	Dublin	Ani	0865825410	CML	35.00	00000PR
Alesia	Tuca	06/06/2009	Female	Gallway	Cristina	0871332881	CML	36.00	00000PR
Daniel	Papadie	19/11/2010	Male	Dublin	Ludmila	0871598750	CML	25.00	00000Doc
				Galway	Lily	0896325475	All	40.00	00000PR

Figure 12 List of patients

Admin can retrieve a list of all users registered in the system and the request will return a list with users sorted by user roles.



GroupedUsersView

Patient

123444M
newuser@yahoo.com
Patient
123456M
darren_ancuta@yahoo.com
Patient

Doctor

00000Doc
zara@yahoo.com
Doctor

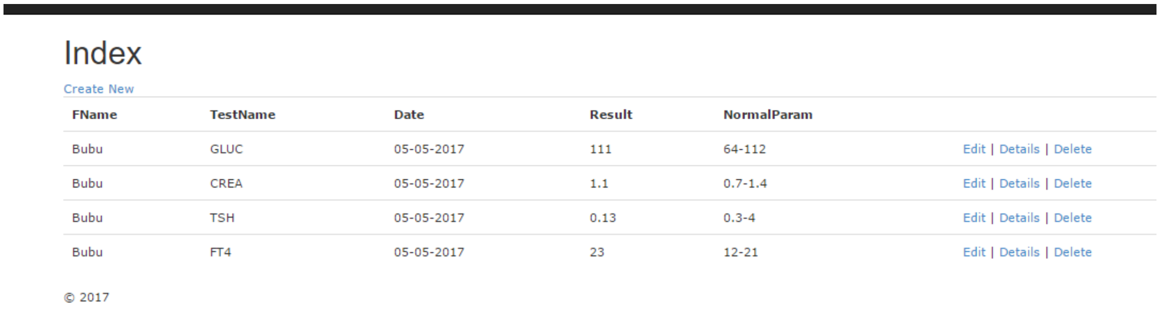
Nurse

Admin

00000Admin
zara_ancuta@yahoo.com
Admin
00000Admin1
cristinavancuta@gmail.com
Admin

Figure 13 List of users

Medical Staff can create, update, edit and delete specific pages such as master treatments, appointments, lab results and patients profile. Symptoms are submitted by patients and medical staff can only read and database will record every form submitted by the patients.



Index

Create New

FName	TestName	Date	Result	NormalParam	
Bubu	GLUC	05-05-2017	111	64-112	Edit | Details | Delete
Bubu	CREA	05-05-2017	1.1	0.7-1.4	Edit | Details | Delete
Bubu	TSH	05-05-2017	0.13	0.3-4	Edit | Details | Delete
Bubu	FT4	05-05-2017	23	12-21	Edit | Details | Delete

© 2017

Figure 14 Create master treatment

2.8 Database ERD

E-Leukemia system is using two databases, Context A and Context B. Context A is built in by ASP.NET and is holding information about users, roles and login. The second database Context B is storing information about Departments, Lab Results, Master Treatment, Patients, Staff and Symptoms.

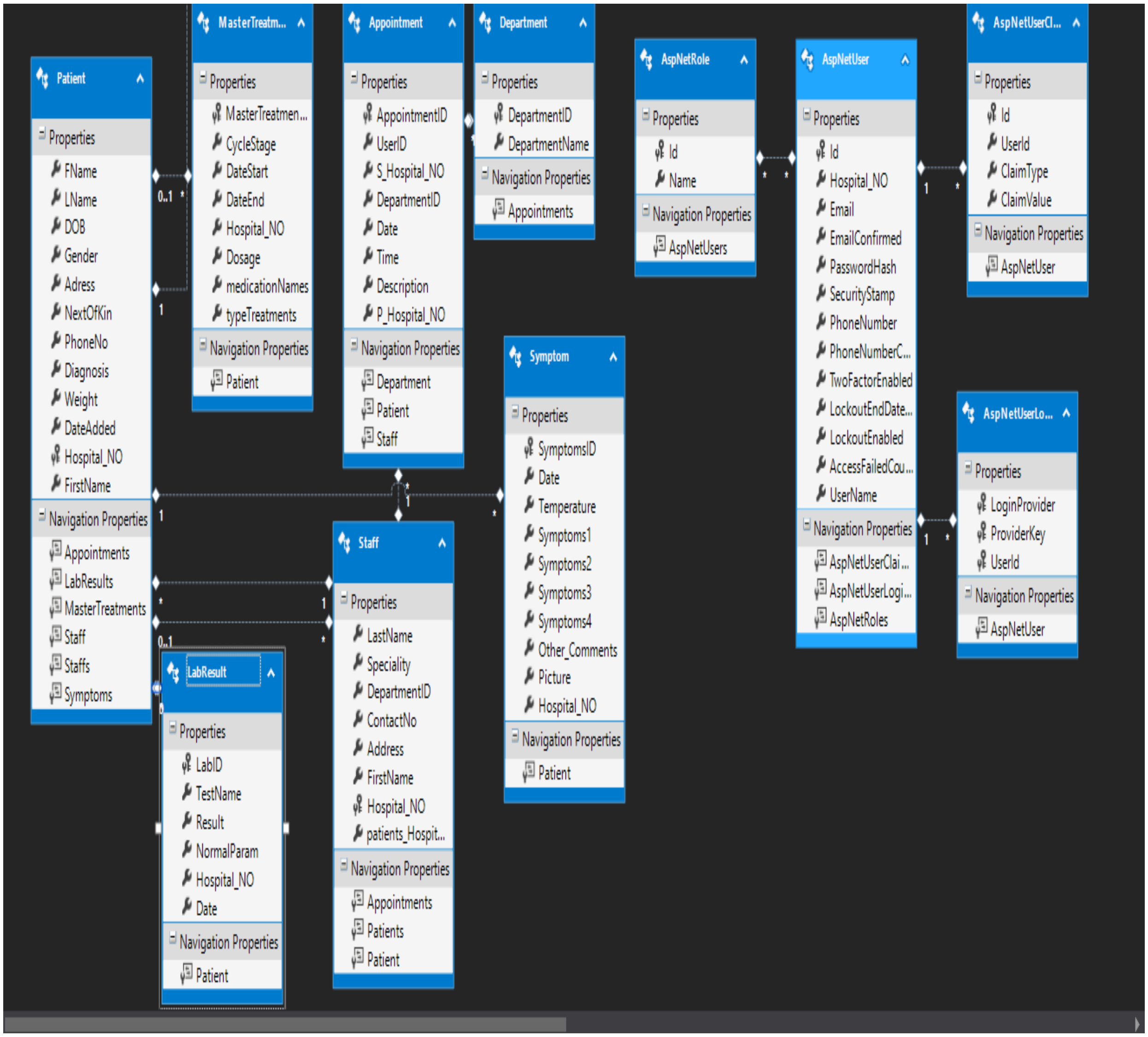


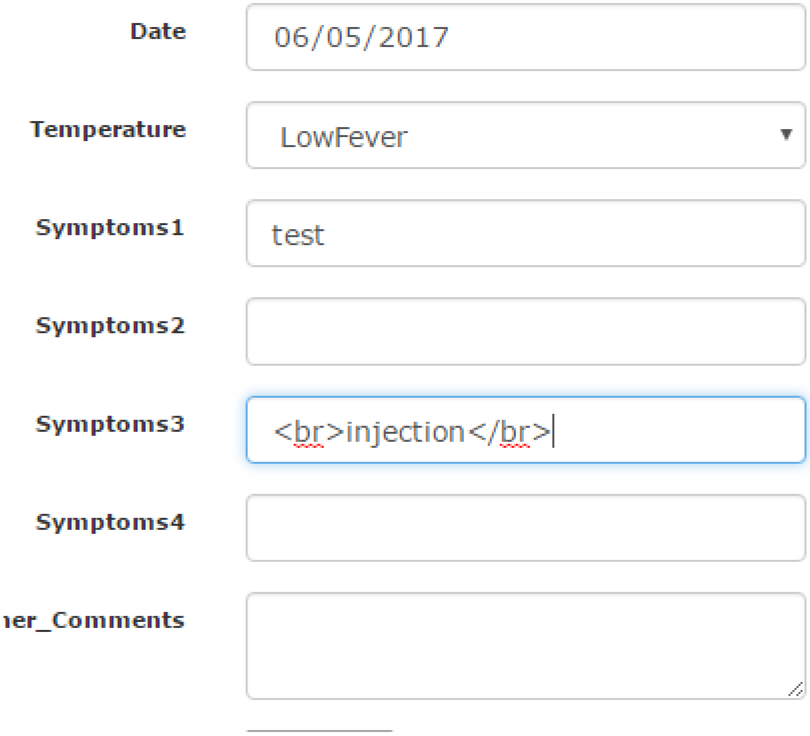
Figure 15 ERD diagram

2.9 Testing

Software testing is a very important key point in software development lifecycle. In order to ensure the system has minimal flows a series of tests were carried out. The purpose of testing is to identify existing flows and vulnerabilities in a system and mitigate against them.

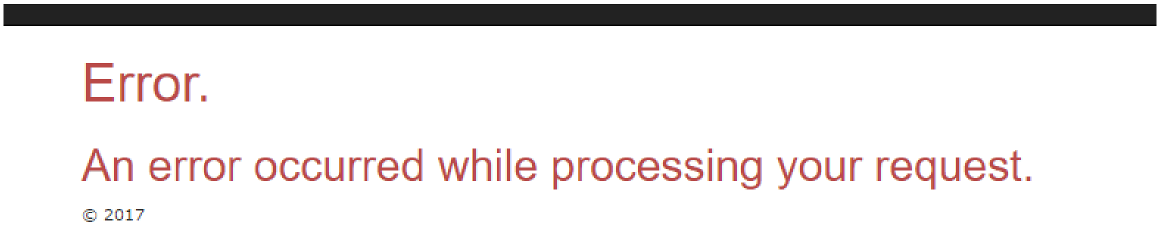
2.9.1 Testing script injection

XSS attacks are type of injections where the attackers insert malicious scripts in the browser. These types of attacks are successful if flaws are identified in a system. For testing purposes a html code is passed in the input field and the action didn’t return any internal error that could potentially lead to an attack. Because custom handle was implemented in the system internal errors will not be displayed.



Date	06/05/2017
Temperature	LowFever ▾
Symptoms1	test
Symptoms2	
Symptoms3	 injection</br>
Symptoms4	
ıer_Comments	

Result



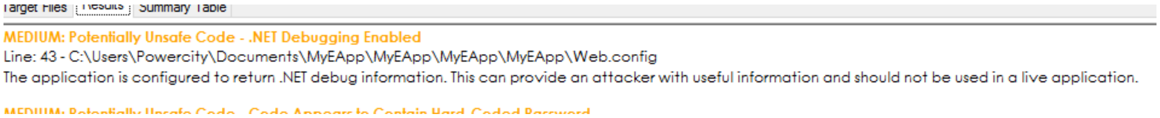
Error.

An error occurred while processing your request.

© 2017

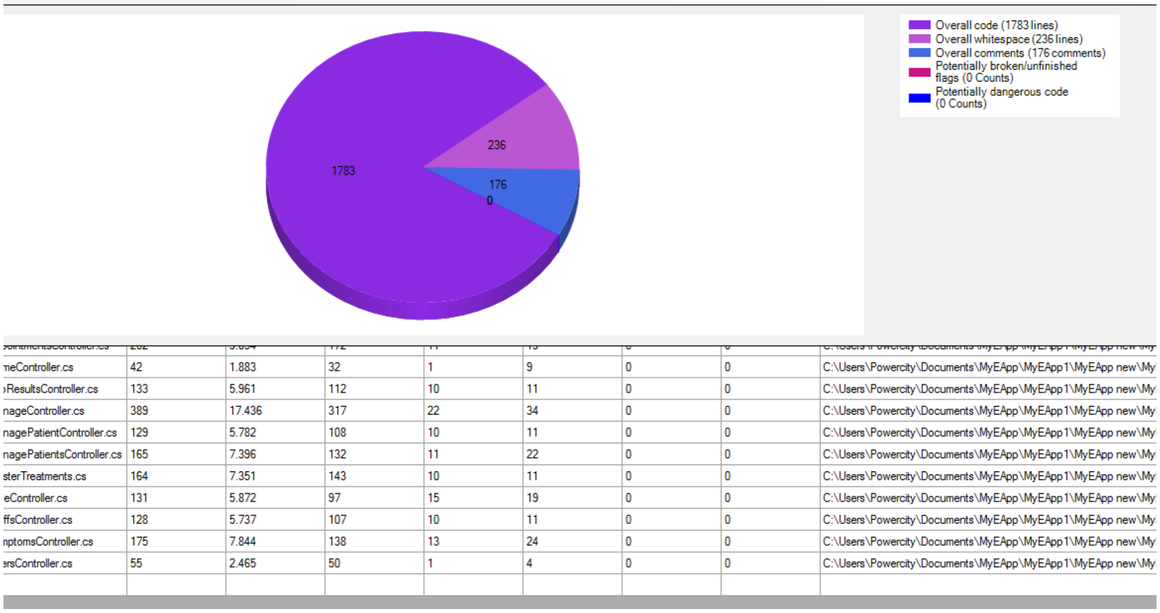
Visual code Grepper is an automated security tool that scan the system to identify vulnerabilities. The output of this automated scan tool identified a potentially unsafe code due to .NET Debugging Enabled. The identified vulnerability is considered a Medium risk and it was enabled on this application for debugging.

After this vulnerability, has been identified by the scanner it was disabled.



```
MEDIUM: Potentially Unsafe Code - .NET Debugging Enabled
Line: 43 - C:\Users\Powercity\Documents\MyEApp\MyEApp\MyEApp\MyEApp\Web.config
The application is configured to return .NET debug information. This can provide an attacker with useful information and should not be used in a live application.

MEDIUM: Potentially Unsafe Code - Code Appears to Contain Hard-Coded Password
```

The automated scan was launched again and no other flaws were detected in the system.


1783
236
176
0
Overall code (1783 lines)
Overall whitespace (236 lines)
Overall comments (176 comments)
Potentially broken/unfinished flags (0 Counts)
Potentially dangerous code (0 Counts)

ointmentsController.cs	[illegible]	[illegible]	172	11	15	0	0	C:\Users\Powercity\Documents\MyEApp\MyEApp1\MyEApp new\My
meController.cs	42	1.883	32	1	9	0	0	C:\Users\Powercity\Documents\MyEApp\MyEApp1\MyEApp new\My
ResultsController.cs	133	5.961	112	10	11	0	0	C:\Users\Powercity\Documents\MyEApp\MyEApp1\MyEApp new\My
nageController.cs	389	17.436	317	22	34	0	0	C:\Users\Powercity\Documents\MyEApp\MyEApp1\MyEApp new\My
nagePatientController.cs	129	5.782	108	10	11	0	0	C:\Users\Powercity\Documents\MyEApp\MyEApp1\MyEApp new\My
nagePatientsController.cs	165	7.396	132	11	22	0	0	C:\Users\Powercity\Documents\MyEApp\MyEApp1\MyEApp new\My
sterTreatments.cs	164	7.351	143	10	11	0	0	C:\Users\Powercity\Documents\MyEApp\MyEApp1\MyEApp new\My
eController.cs	131	5.872	97	15	19	0	0	C:\Users\Powercity\Documents\MyEApp\MyEApp1\MyEApp new\My
ffsController.cs	128	5.737	107	10	11	0	0	C:\Users\Powercity\Documents\MyEApp\MyEApp1\MyEApp new\My
nptomsController.cs	175	7.844	138	13	24	0	0	C:\Users\Powercity\Documents\MyEApp\MyEApp1\MyEApp new\My
ersController.cs	55	2.465	50	1	4	0	0	C:\Users\Powercity\Documents\MyEApp\MyEApp1\MyEApp new\My

2.9.2 Black box testing

Black box testing is executed during the development of project and is mainly used for testing functionality in a system. The main functionalities of E-Leukaemia has been executed as per below table.

Test Number	Description	Expected outcome	Pass/Fail
1	Register users 1. Fill in the form with all info	Message with user name showed on the screen	Pass

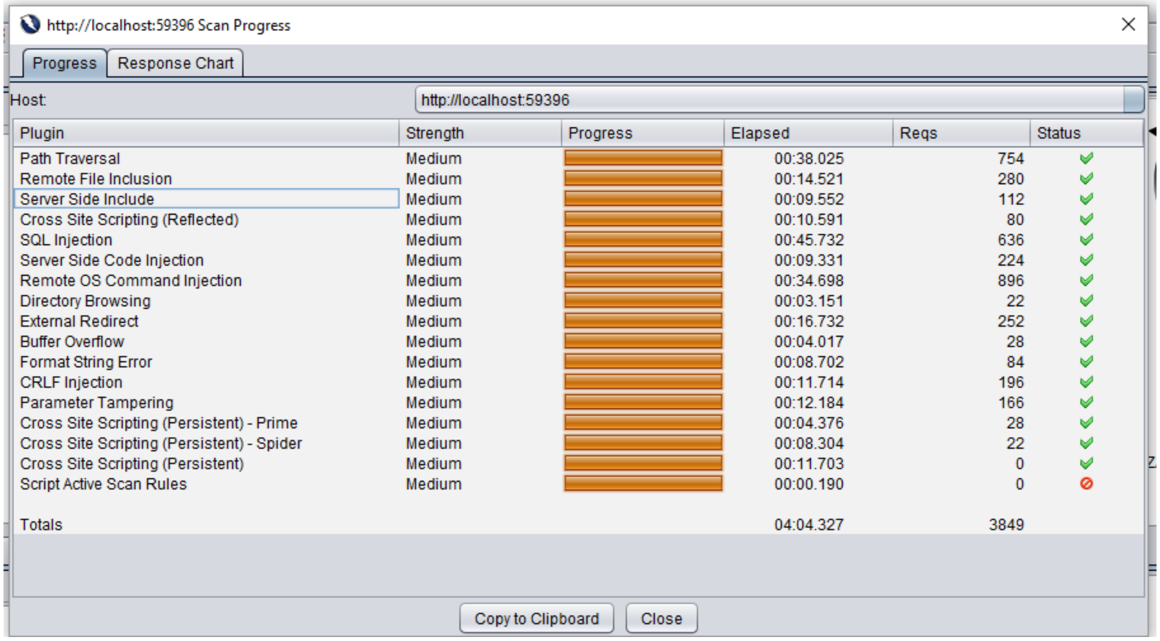
	2. Submit button		
2.	1. Register users without mandatory info 2. Submit form	An error popped up on the screen	Pass
3.	Register users with passwords without capital letter and characters	An error popped on the screen informing the user about incorrect password format	Pass
4.	Change user password with the old password	An error appeared on the screen saying that is a password already used	Pass
5.	Sign in with auth 2FKA enabled and input wrong token	Error message showed up on the screen, login not granted	Pass
6.	Create new user and assign user as a MedicalStaff	The authenticated user will view only authorised pages	Pass

7.	Create a user with invalid hospital_No	Error message return on the screen that hospital_NO is not existing in hospital DB	Pass

2.10 *Penetration scanning*

Owasp ZAP was used to scan and identify the vulnerabilities of E-Leukaemia system. It’s a tool that was developed at OWASP team and it’s used by professional testers to identify vulnerabilities in the system that may lead to a possible attack.

Zap will scan the application and it will output the results on the screen. E-Leukaemia was tested with Zap and it did not detect any vulnerabilities in the application as it can be seen below.



Plugin	Strength	Progress	Elapsed	Reqs	Status
Path Traversal	Medium	▬	00:38.025	754	✓
Remote File Inclusion	Medium	▬	00:14.521	280	✓
Server Side Include	Medium	▬	00:09.552	112	✓
Cross Site Scripting (Reflected)	Medium	▬	00:10.591	80	✓
SQL Injection	Medium	▬	00:45.732	636	✓
Server Side Code Injection	Medium	▬	00:09.331	224	✓
Remote OS Command Injection	Medium	▬	00:34.698	896	✓
Directory Browsing	Medium	▬	00:03.151	22	✓
External Redirect	Medium	▬	00:16.732	252	✓
Buffer Overflow	Medium	▬	00:04.017	28	✓
Format String Error	Medium	▬	00:08.702	84	✓
CRLF Injection	Medium	▬	00:11.714	196	✓
Parameter Tampering	Medium	▬	00:12.184	166	✓
Cross Site Scripting (Persistent) - Prime	Medium	▬	00:04.376	28	✓
Cross Site Scripting (Persistent) - Spider	Medium	▬	00:08.304	22	✓
Cross Site Scripting (Persistent)	Medium	▬	00:11.703	0	✓
Script Active Scan Rules	Medium	▬	00:00.190	0	⊘
Totals			04:04.327	3849	

Figure 16 Owasp Zap scan

2.11 Performance testing

Page loading performance is important for users as they expect to interact with the system fast to collect the information they need. To evaluate the performance of the system on online tool was used to rank the website.

The test estimated a page speed of 93 out of 100 showing that the users get access the application fast with no major delays.

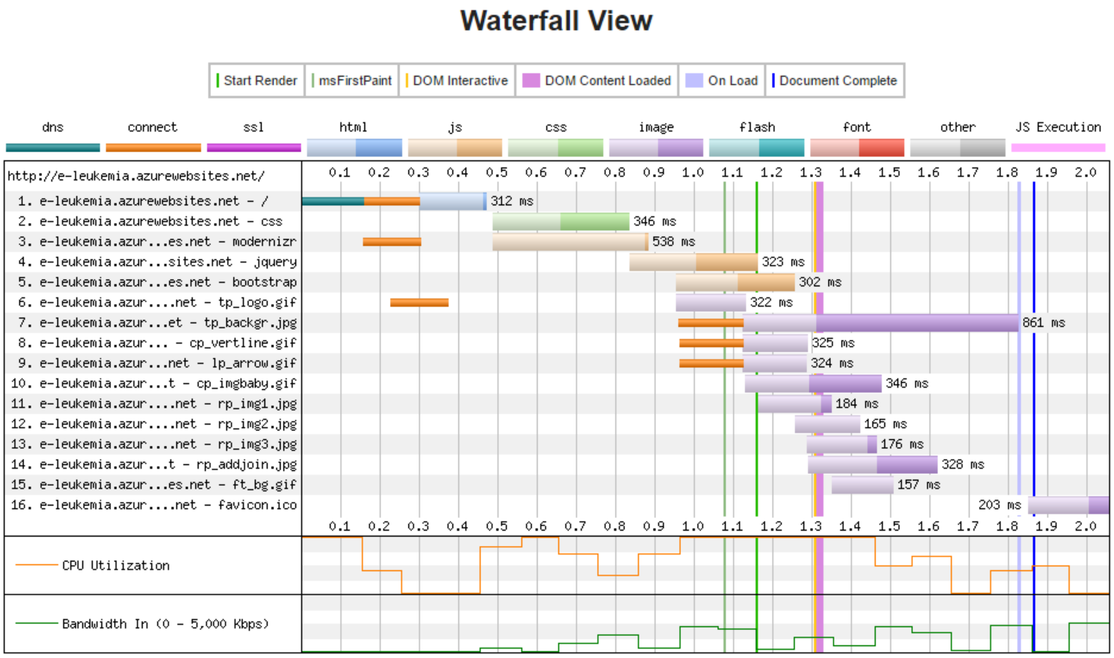


Figure 17 Performance testing

2.12 Customer testing

The idea of creating a web application for kids with Leukaemia arise from best my friend when her child, aged 2 at the time was cruelty diagnosed with leukaemia. Her world fell apart and her life changed to 180 degrees. She never expected something so dramatic to occur in her life and thoughts of guilt and stress were always in her mind so that affected her concentration for her child treatment. She always had loads of notes with the treatment details, or symptoms that happened during a chemo session and it was very hard to keep everything organised and in control. I asked her if an electronic application would be any help to her she agreed that would be great idea. She shared my idea with other mothers in this situation and they all agree that would be very helpful. I went over to Crumlin Hospital and ask Medical Staff what do they think from their side and they also approved my idea. I gathered information's from both sides (patient and hospital) and based on the user's preferences I designed this web application. During the development, I’ve asked my friend to test the application and I encouraged other parents to do it too. They all tested my prototype and based on their feedback I adjusted the application accordingly.

2.12 Development cost

The prototype is developed to help patients to self-manage leukaemia treatment and it will be released for testing purposes to ensure the system meets users’ needs. The application will be launched for a small group of people during a 3 months’ trial. Based on the feedback the system will be re-evaluated and improved. Another trial of will be launched to a bigger number of users for further 3 months then costs will be outlined in the document.

2.13 Conclusions

There are many advantages around E-Leukaemia concepts. A good example is Electronic health records. The application will provide an accurate, up to date and complete information for the users. Parents will get all the information they need by accessing the application at any time and from anywhere. All the information they need will be electronically integrated in this system and it will help them to eliminate any confusion they might have in relation to the treatment details with all complicated medication terms. They can keep tracking important information after each chemotherapy cycle and the doctors can follow up and analyse any details that can help them to decide the right treatment for each individual patient. The application can improve clinical decision making by integrating patient information from multiple sources.

A system must be developed with an interface to allow users to communicate with the system, which E-Leukaemia Leukaemia will introduce. Another important aspect is the electronic language to cross communicate between different healthcare system about the medical data such as patient's treatment and hospital internal record.

Testing process evaluated the security and performance of E-Leukaemia system helping the implementation to be improved creating a secure, fast and reliable product.

.

3 Further development or research

Since the application have been done within limited time the system can be further expanded with new enhanced functionalities and features. The application can be developed on a mobile platform, android or IOs as many people are using the applications from their mobile instead of accessing a web application on their phone.

Video features can be wrapped into the system to allow patients to interact with medical staff in real time. Future enhancements that can be applied to the proposed system can be enhanced to be more user friendly and effective for user registration, log in and requests for consulting doctors. Further, the system can be modified in such a way that patients can communicate their problem to the doctor from their home through internet using an online chat in real times.

More encryption and security can be implemented to add an extra layer of security to the system. Treatments portal can be developed in a better way by displaying a calendar with additional details for the treatment where information can be centralized and collected in an easier manner by the patients.

The application can be adapted to suit different departments in the hospitals such as diabetes or epilepsy categories.

The purpose of further development is to provide users an easier interaction with the system and make it more complex and secure. Technologies are constantly evolving so we need to adapt to these changes.

4 References

How Technology Could Save Your Life In An Irish A&E". *The Irish Times*. N.p., 2017. Web. 2 May 2017. [1]

C-sharpcorner.com. (2017). *Security Feature in ASP.Net MVC*. [online] Available at: http://www.c-sharpcorner.com/UploadFile/cda5ba/security-feature-in-mvc/ [2] [Accessed 4 May 2017].

Csharptools.com. (2016). *C# Tools - Testing*. [online] Available at: http://www.csharptools.com/view/testing

[Accessed 9 Dec. 2016].

Csharptools.com. (2016). *C# Tools - Testing*. [online] Available at: http://www.csharptools.com/view/testing

[Accessed 9 Dec. 2016].

The Irish Times. (2017). *How technology could save your life in an Irish A&E*. [online] Available at: http://www.irishtimes.com/life-and-style/health-family/how-technology-could-save-your-life-in-an-irish-a-e-1.2982016

[Accessed 2 May 2017].

bageri, s. (2017). *10 Points to Secure Your ASP.NET MVC Applications. - CodeProject*. [online] Codeproject.com. Available at: https://www.codeproject.com/Articles/1116318/Points-to-Secure-Your-ASP-NET-MVC-Applications

[Accessed 8 May 2017].

Anwar, W. (2017). *EzzyLearning - Tutorials, HOW TOs, Blog Posts, Sample Code about C#, ASP.NET, AJAX, JQuery, LINQ, VB.NET, ADO.NET, .NET Framework, XML, Web Services, Java and more*. [online] Ezzylearning.com. Available at:

http://www.ezzylearning.com/tutorial/creating-custom-validation-attribute-in-asp-net-mvc-

[Accessed 8 May 2017].

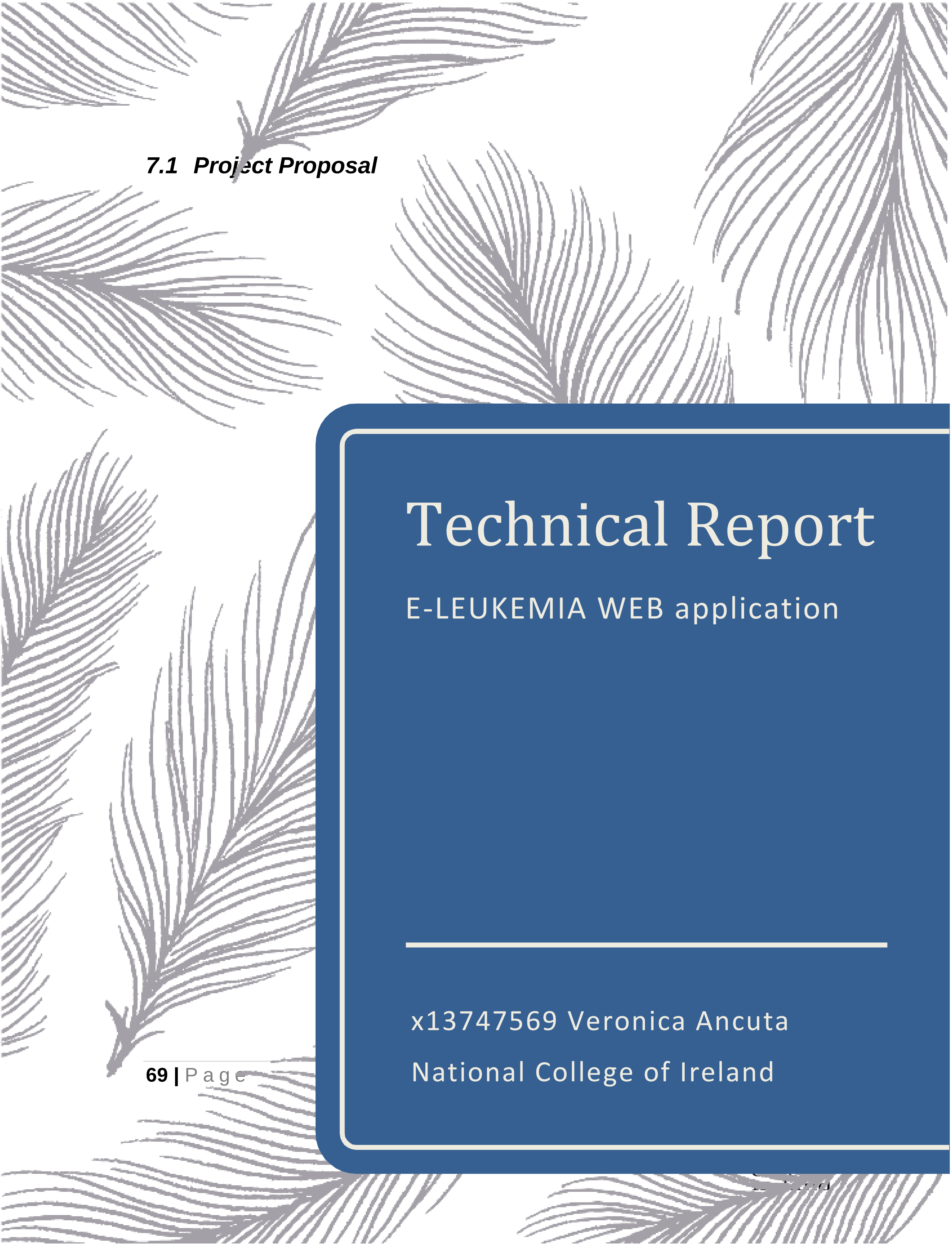
Docs.microsoft.com. (2017). *Updating Related Data with the Entity Framework in an ASP.NET MVC Application (6 of 10)*. [online] Available at: https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions/getting-started-with-ef-5-using-mvc-4/updating-related-data-with-the-entity-framework-in-an-asp-net-mvc-application

[Accessed 8 May 2017].

7 Appendix

The following pages are attachments to the project

7.1 Project Proposal



Technical Report

E-LEUKEMIA WEB application

x13747569 Veronica Ancuta

National College of Ireland

Leukaemia Web Application Proposal

Veronica C Ancuta

X13747569

Veronica.Ancuta@student.ncirl.ie

BSc(Hons) in Computing-Cyber Security

19/10/2016

Objectives

The purpose of the web application is to serve patients with a beneficial service to access medical information's about their illness.

The advantage of Leukemia Web Application can be made full use of use of by making up the time and distance gap between patients and doctors or nurses to provide exact information when needed.

Leukemia require a long plan treatment, 3 years for boys and 2 years for girls and it involves many surgical interventions, different plan medication for each phase of treatment, weekly blood tests and clinics appointments.

Parent/ career can access the web application to read plan treatment for each week with additional details that includes the dosage and optimal time to intake the medication, oral or through Hickman.

On the other side Oncology Department, should upload plan treatment for each cycle of leukemia and notify the patient about the upcoming appointments.

After each chemotherapy patients, might have different reactions or sides effects which they need to be monitored and recorded with careful details. The application should allow parents/career to record these sides effects (e.g. temperature, skin rash, other complications) for each new medication.

The application display appointments for patients, when they are due in for clinics, appointments or theatre interventions.

I choose to build a web application and not a mobile app due to specific clarity of information that a user needs to access, a small screen can diminish details of the treatment which it can be fatal in this case.

The web application will be accessible from a mobile device also as it will be user friendly and it will be responsive.

Background

The use of technology by health care professionals has improved many aspects of clinical practice. Numerous apps and web applications are available to assist hospital staff with many important aspects: information and time management; health record maintenance and access, communication and consulting, patient monitoring and many other. These applications are widely used by health professionals and patients.

Web applications and apps play an important role in patient education, information disease self-management and remote monitoring of patients.

The medical environment has been slow to adapt to technical progress in IT because the IT usage is not regarded as a priority.

The proposed solution is designed for kids between 0-16 years registered in St John's Ward from Crumlin Hospital. The application will be used by parents/career to help them manage the diagnosis and the long treatment plan.

As a part of my research I couldn't find any web application integrated into the hospital to help patients with their treatment.

The terminology and the variety of the medication for this long battle is hard to memorize.

According to the last survey over 400 cases in Ireland are diagnosed with leukemia every year, where over 200 are children under age 19.

Children receiving treatment for their leukemia will go home after a 2-week initial stay and they continue treatment as an outpatient.

Crumlin offer at home chemotherapy which works brilliantly, it saves patients from travelling to the hospital.

The diagnosis of leukemia has a deep effect on the child and especially on parents, family and friends.

When doctors talk about cancer survival statistics, often they say 5-year survival rate and they refer to the patients who live 5 years after they have been diagnosed.

With leukemia children that are free of disease after 5 years are more than likely successfully treated, in rare occasions cancer return after this period.

Cancer treatments are traumatizing and parents might find it very stressful to remember details of the medications and the optimal hours to be administrated.

After the patient’s situation is established and a Hickman is inserted in the heart medications can be administrated oral and through the line. The application will inform parent/career on what medication they must administrate to the sick child.. Details of dosage will be uploaded by the medical staff with additional instructions.

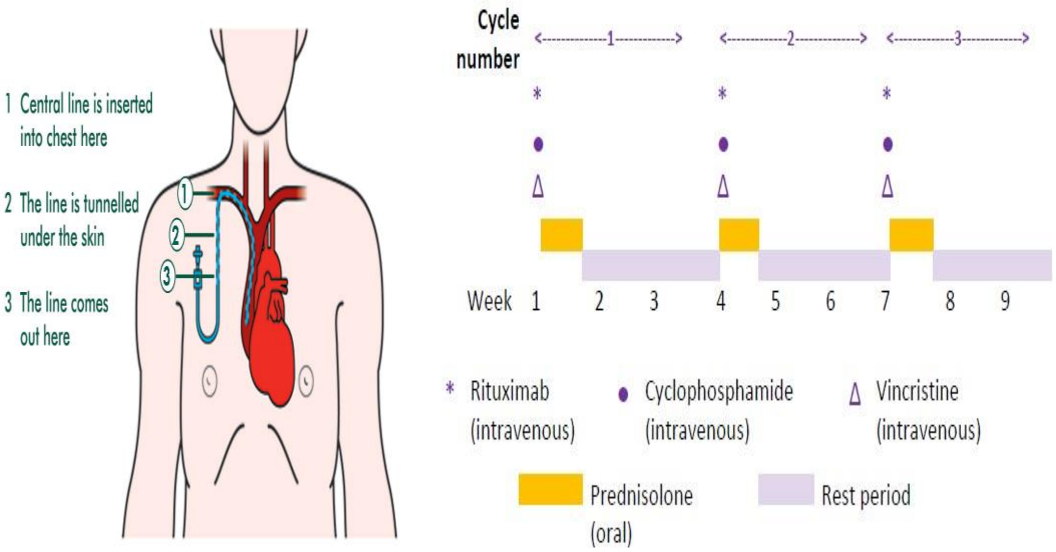

1 Central line is inserted into chest here
2 The line is tunnelled under the skin
3 The line comes out here
1
2
3
Cycle number
<-----1----->
<-----2----->
<-----3----->
Week 1 2 3 4 5 6 7 8 9
* Rituximab (intravenous)
● Cyclophosphamide (intravenous)
Δ Vincristine (intravenous)
Prednisolone (oral)
Rest period

Figure 19-Medication details

Patients will attend the clinic every week and they can read and create new appointments with the hospital when required through the web application.

The web application will be used by patients, medical staff and admin and they will be registered by the admin.

Parents/ career should have access only to personal information, data protection will be implemented in the project.

The medical staff should be able to hide some information that is not relevant to the patient. (e.g. nurse’s notes)

Technologies and technical approach

To be able to develop ASP.NET MVC web application I need a browser, operating system (windows, Mac or android).

In order to make my idea to take shape from development phase up to completion process will make use of Visual Studio, that will enable me to create a powerful web framework. I will use C Sharp for the web service as it's deployed on Windows Azure. I will use Windows Azure as the cloud platform and SQL because is available on Windows Azure.

Why C sharp?

Because it is one of the most versatile programing language in the world, it can be used to develop windows client applications, web applications windows store apps, iOS and android applications and has security features built in. For interface, I will use boot strap to develop a responsive application on the web.

Hardware requirements

Laptop/ PC with operating system windows

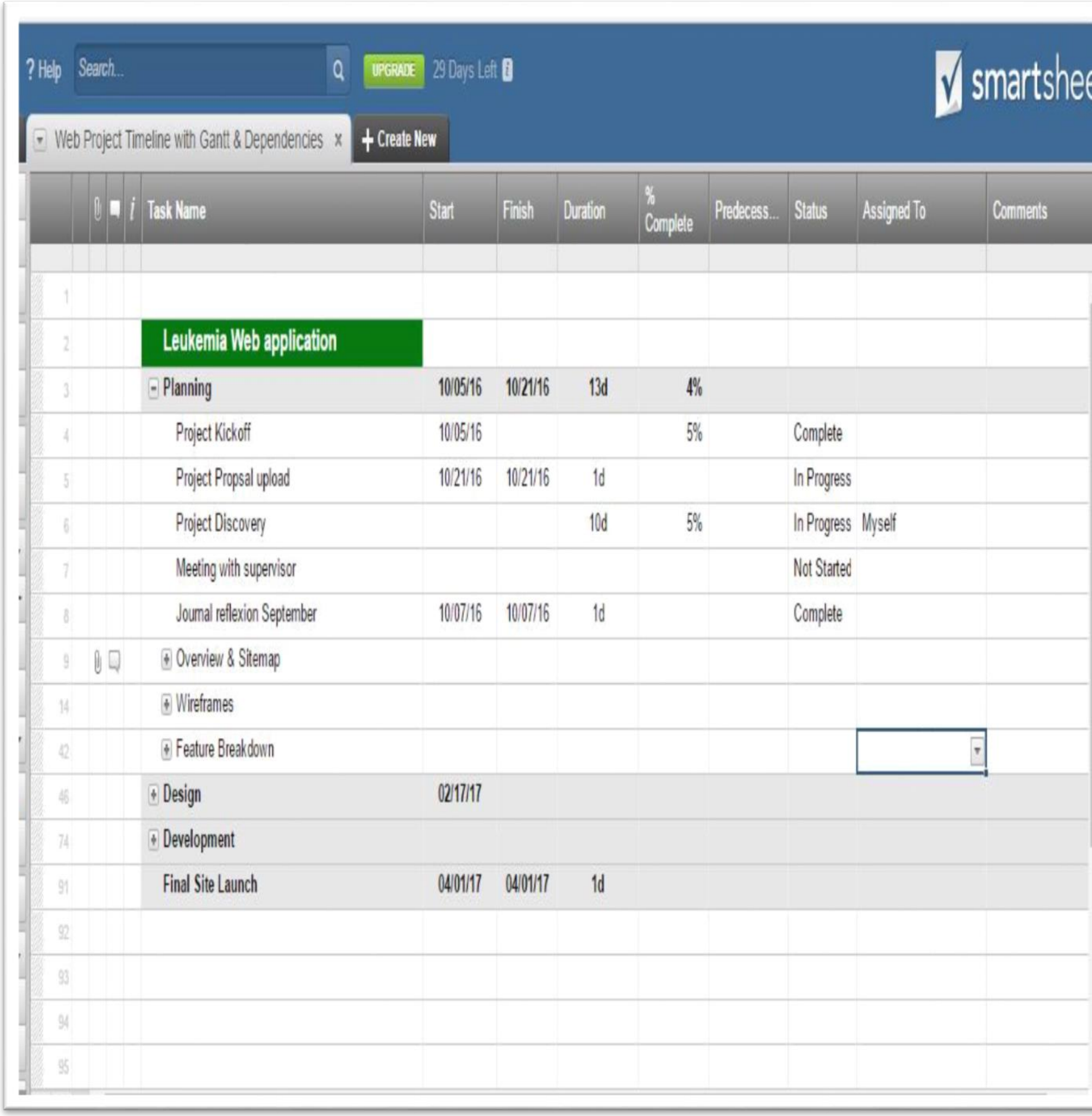
Programing language C sharp

Web-Framework –ASP.NET

Back -end SQL server

Web server -IIS

Gantt Chart



	Task Name	Start	Finish	Duration	% Complete	Predecess...	Status	Assigned To	Comments
1									
2	Leukemia Web application								
3	Planning	10/05/16	10/21/16	13d	4%				
4	Project Kickoff	10/05/16			5%		Complete		
5	Project Propsal upload	10/21/16	10/21/16	1d			In Progress		
6	Project Discovery			10d	5%		In Progress	Myself	
7	Meeting with supervisor						Not Started		
8	Journal reflexion September	10/07/16	10/07/16	1d			Complete		
9	Overview & Sitemap								
14	Wireframes								
42	Feature Breakdown								
46	Design	02/17/17							
74	Development								
91	Final Site Launch	04/01/17	04/01/17	1d					
92									
93									
94									
95									

Figure 20-Gantt chart

Evaluation

I will create a C sharp source code file with full class and run it through the CodeDom provider for C sharp and compile it into an assembly then execute it

Also, I will run tests on a weekly basis and with multiple testers and I will use their feedback to improve my web application.

I will use myself the application, make modification, J-unit testing and I will record all the outcomes in my reflective journal.

FxCop is an application that analyze managed code assemblies so code that targets the .NET framework and reports information about the assemblies such as possible design, localization, performance and security improvements.

Prefast is another tool that identifies defects in C sharp programs.

Google CodeSearchDiggity uses google to spot vulnerabilities in open source code hosted by Github. Security is a major feature in my project idea and this tool can identify SQL injection, cross-site scripting, insecure remote and local files, hard coded password and so on.

For system performance, I will use Timing Analyzer known as profiler, it reports time spent in varies range of code and it will concentrate on areas to improve performance.

7.2 Monthly Journals September, October, November, December, January, February and March

Reflective Journal

Student name: Veronica Ancuta

Programme (BSc in Computing): Cyber Security

Month: September

My Achievements

This month, I didn't achieved too much in relation to my project as I wasn't sure if my idea is feasible or not. I had my heart set on this idea and I'm happy that it went through.

My Reflection

I've realised that I have to maximise security in my project idea otherwise my app will not be successful. I'm thinking of functionality and I need to gather information from the nurse's point of view as my app is related to the hospital.

Intended Changes

No changes yet

I have to discuss with nurses to find out if they would embrace my idea and if they do what would they expect from this app.

Supervisor Meetings

Date of Meeting:

Items discussed:

Action Items:

Reflective Journal

Student name: Veronica Ancuta

Programme (BSc in Computing): Cyber Security

Month: October

My Achievements

This month, I was able to set an account to GitHub. I uploaded my java files into my GitHub account to see how it works.

I am still not sure of what technologies I am going to use for my project as I was advised to use other languages than C sharp. I already start designing database as is very important step in my idea development. For my project technical requirements, I struggled a little bit as I don't know exactly how this web application is going to look like. At the moment I focus on the requirements and I try to keep up with the deliverables.

I am planning to meet with a nurse from Crumlin Hospital to get an idea of what would they expect from an application that I am going to build. Any opinion or recommendation can benefit my project.

Kalianee B was assigned as a supervisor for my project. I emailed her and we agreed to meet in the college to talk about my project requirements. She had some guidelines for me and she asked me to do a dissertation.

My Reflection

I need to decide on technologies I am going to use. I would need some help in relation to that and I hope I can get good advises.

However, I was not successful in completing the document for my supervisor due to huge amount of work required by the college. I know this would be very beneficial to me to help me understand the most important aspects of my project and I am planning to complete the task I was assigned as soon as possible.

Intended Changes

Next month, I will work on the prototype and I will have database fully implemented.

Supervisor Meetings

Date of Meeting: 02/11/2016

Items discussed: PROJECT DISSERTATION

Reflective Journal

Student name: Veronica Ancuta

Programme: BSc in Computing – Cyber Security Stream

Month: November

My Achievements

This month, I was able to get a better picture about my project. For my project requirements I've done additional research to help me find the best approach. I've designed the class diagram for the document and further implemented the database for my application to ensure that I can get a working prototype. My supervisor gave me suggestions on the areas I have to improve and they were very beneficial to me. I also approached some of my lecturers asking for advice and that cleared a lot of confusions I had in relation to my submission. I felt that some part of requirements were difficult to implement as I wasn't sure if they can align with my project idea.

My Reflection

I felt, it worked well to finish the project requirements and they will be a good guide for my next deliverable. I wish I would have had more time to spend on the document because I am obsessed about details and quality of work. Talking to my supervisor and other people around it really eased the pressure I accumulated lately with the projects.

Intended Changes

Next month, I will complete the technical requirements with the documents required for my next milestone and I have to implement a working prototype. My database is already implemented and I need to focus on security features. The most annoying aspect is the Visual Studio that gave me such a difficult time in trying to fix the errors. Hopefully further research and tutorials would help me to have a good prototype for my next presentation.

I realised that I need to get a voucher for hosting my web application on Azure server. I was advised by Michael B that I should be able to get a free voucher but the downside is that the duration of this voucher is only for 3 months hence I will try to obtain this with few months before May.

Supervisor Meetings

Date of Meeting:

Reflective Journal

Student name: Veronica Ancuta

Programme (BSc in Computing): Cyber Security

Month: December

My Achievements

This month I worked on my prototype and I finished my report. It was a though month and involved a lot of college work. I can say it was the hardest month of all these years in NCI. I was able to explain my idea better by using a working prototype in front of my lecturers. They were happy with my working prototype and they also pointed out few aspects to my project. I need to focus more on my project and get things done within the deadline.

My Reflection

Mid-Point presentation helped me to prioritise security features which are very important for my project. I have to make some changes to my project as I got better ideas from my supervisor. I realized that I have to create separate login for admin in order to implement access controls for the users.

Intended Changes

As I mentioned above I need to make some changes to my project and one of them is related to Tracking symptoms for patients. I was suggested to save them as a PDF to ensure that the users can still save the symptoms when they don't have an active internet connection. Also I will use encryption for these details to secure the information. Eugene noticed that my prototype had a small error in the HTML design, the main GUI from Home index wasn't proportionally laid out so I need to calibrate that by making small changes in the code.

Supervisor Meetings

I have met Kalianee in December and we discussed the details that I needed to implement for the mid-point presentation. I also emailed her my report and she gave me feedback with additional changes. The next meeting will be after my exams to discuss further the plan for my project.

Reflective Journal

Student name: Veronica Ancuta

Programme (BSc in Computing): Cyber Security

Month: January

My Achievements

I developed my application further by adding extra functionality as suggested in my last journal. One of the planned functionality was pointed out to me during the presentation from December suggested by Eugene. I followed tutorials on the internet and I asked some developers for an advice as I couldn't find anything related to this and because my project is a web application and one of the main conditions is for users to have active internet connection I could not implement this step.

During my research I found out that mobile applications support this feature when network connection is not available (example: Facebook -user still access information when network connection is not available) but for web application this feature is not available. Hence I will use my initial idea to track symptoms for the patients by filling in a form and store it into database.

Also I implemented the admin role and this was one of the most important aspect of my project. Admin will register users to the system and he will assign different permission based on their role.

This is a security feature that will enhance the web application to be more secure.

My Reflection

I am focused on my project and I'm determined to finish the web application earlier so I can test it in the live environment. Testing it's important for my project and also will increment my marks if I meet the requirements.

Intended Changes

I will not make any changes for next step in my project. I will work on patient portal and I will work on the appointments class for next week.

Supervisor Meetings

I have met Kalianee few weeks ago and I cancelled the meeting on 14th of February as I was sick.

Reflective Journal

Student name: Veronica Ancuta

Programme (BSc in Computing): Cyber Security

Month: February

My Achievements

This month I made progresses with my project and I encountered few challenges because of my database structure. I have realised that I have some inconsistencies in my database and I had to rebuild some tables. Migration and input validation was the biggest challenge this month. I worked on my patient and staff portal that allow users to see different type of data. I implemented admin control with full access to database. Users have different access permissions and only authorised staff can make changes to patient information.

My Reflection

I had my regular meetings with my supervisor and she suggested me to add extra information in my project requirements document. I must focus on more security aspects and project design. At the end of next month, I intend to deploy my project on Azure to enable users to use the web application for testing purposes as this is another important phase of my project.

Intended Changes

I probably need to change some fields from my tables to allow me to create appointments functionality. If I will have spare time I will add extra functionality to my project to make it more complex and attractive. I am focused on technical part as I think is the most important at the moment then I will work on the rest of my documentation.

Supervisor Meetings

I have met Kalianee in February and I had a list of tasks to complete. Our meetings are every Tuesday and I update her regularly with my progress on my project. Kalianee is always suggesting me where to improve and she is also asking me to focus on documentation.

Reflective Journal

Student name: Veronica Ancuta

Programme (e.g., BSc in Computing): BSc in Computing -Cyber Security

Month: March

My Achievements

This month I have to complete my project and I need extra time to test the project. I encountered few challenges and I realised that database it's not designed properly and it's difficult to run joint queries or multiple queries. Implementing appointments was difficult and I still must find a solution to fix this problem.

My contributions to the projects included core functionalities that I was hoping to achieve them by the end of the project. All users are set to see and access certain pages according to their roles. I applied security features on my application and I intend to do some encryption for uploaded photos.

My Reflection

I felt, it worked well to work on my project every week and watching tutorials on YouTube helped me to achieve major steps. I couldn't generate all the registered users from my project as I had to join 3 tables and I spent few days trying to achieve this until I found these videos on YouTube. I must update my documents as well because some functionalities changed during the development.

However, I was not successful in developing a patient portal as I initially planned because I need more time to design it as a separate entity in the navigation system but I managed to achieve that by hiding some information using user roles. Some security features blocked few functionalities in my application throwing me errors and I had to revise my code to avoid these issues.

Supervisor Meetings

Date of Meeting: 27/04/2017

Items discussed: update documents, test the projects, create more records in the project

Action Items: Deadline for document 4/04/2017 (email the document to Kalianee).