Industrial Training Report on VMS in MEAN

A Thesis Submit
In Partial Fulfilment of the Requirement
For the Degree of

MASTER OF COMPUTER APPLICATION

by

Sandeep Singh
(1900290149086)

Under the Supervision of

Prof. Ankit Verma
(Assistant Professor)
Department of Computer Applications,
KIET Group of Institutions,
Delhi-NCR, Ghaziabad

to the

Faculty of Computer Application
DR. APJ ABDUL KALAM TECHNICAL UNIVERSITY
LUCKNOW
(Formerly Uttar Pradesh Technical University, Lucknow)June
July, 2021

DECLARATION

I hereby declare that the work presented in this report entitled “VMS IN MEAN", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name : Sandeep Singh

Roll. No. : 1900290149086

Course/Branch : MCA

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

VMS in MEAN is a lightweight data grid control designed on a flexible object model. Based on the popular version, MEAN offers many unique features such as unbound mode, flexible cell merging, and multi-cell row and column headers that make it a proven solution for data management and tabulation.

MEAN is a free and open-source JavaScript software stack for building dynamic web sites and web applications. Because all components of the MEAN stack support programs that are written in JavaScript, MEAN applications can be written in one language for both server-side and client-side execution environments

The MEAN stack is JavaScript-based framework for developing web applications. MEAN is named after MongoDB, Express, Angular, and Node, the four key technologies that make up the layers of the stack.

- MongoDB - document database
- Express(.js) - Node.js web framework
- Angular(.js) - a client-side JavaScript framework
- Node(.js) - the premier JavaScript web server

MEAN applications can be used in many ways with a cross platform write once approach. While MEAN is particularly suited to real-time applications, particularly those running natively in the cloud and single-page (dynamic) web applications built in Angular.js, it can be used for other use cases such as:

- Workflow management tools.
- News aggregation sites.
- Todo and calendar applications.
- Interactive forums.

CHAPTER 2

TECHNOLOGIES / SOFTWARE REQUIREMENTS

- Operating System : Windows
- Programming language: Angular, Node.js (Angular Framework)
- Front-End: HTML,CSS,BOOTSTRAP, ANGULAR
- Back-End: Node.js, Express.js
- IDE: Visual Studio 2019

CHAPTER 3

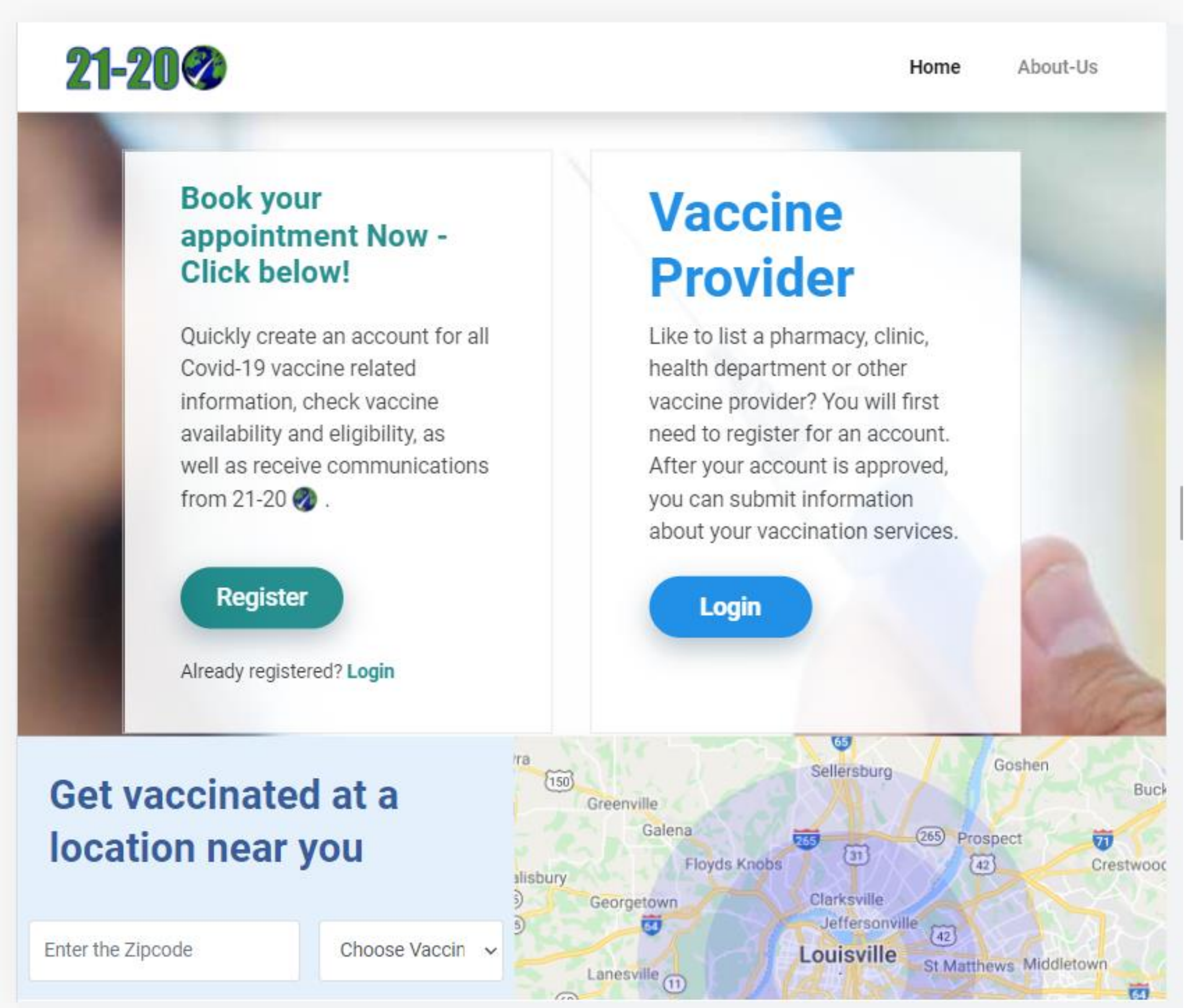
HARDWARE REQUIREMENT / HARDWARE USED

- Windows Version- Windows.
- 2 GB Ram
- 1MB Cache Memory
- External Memory 10 GB

CHAPTER 4

FEATURING AND IMAGES

VMS is free for public health-approved clinics, and can be used on computers, tablets, and other mobile devices. It is not a smartphone app, and no installation or download is required for this web-based platform.



21-20

Home

About-Us

Book your appointment Now - Click below!

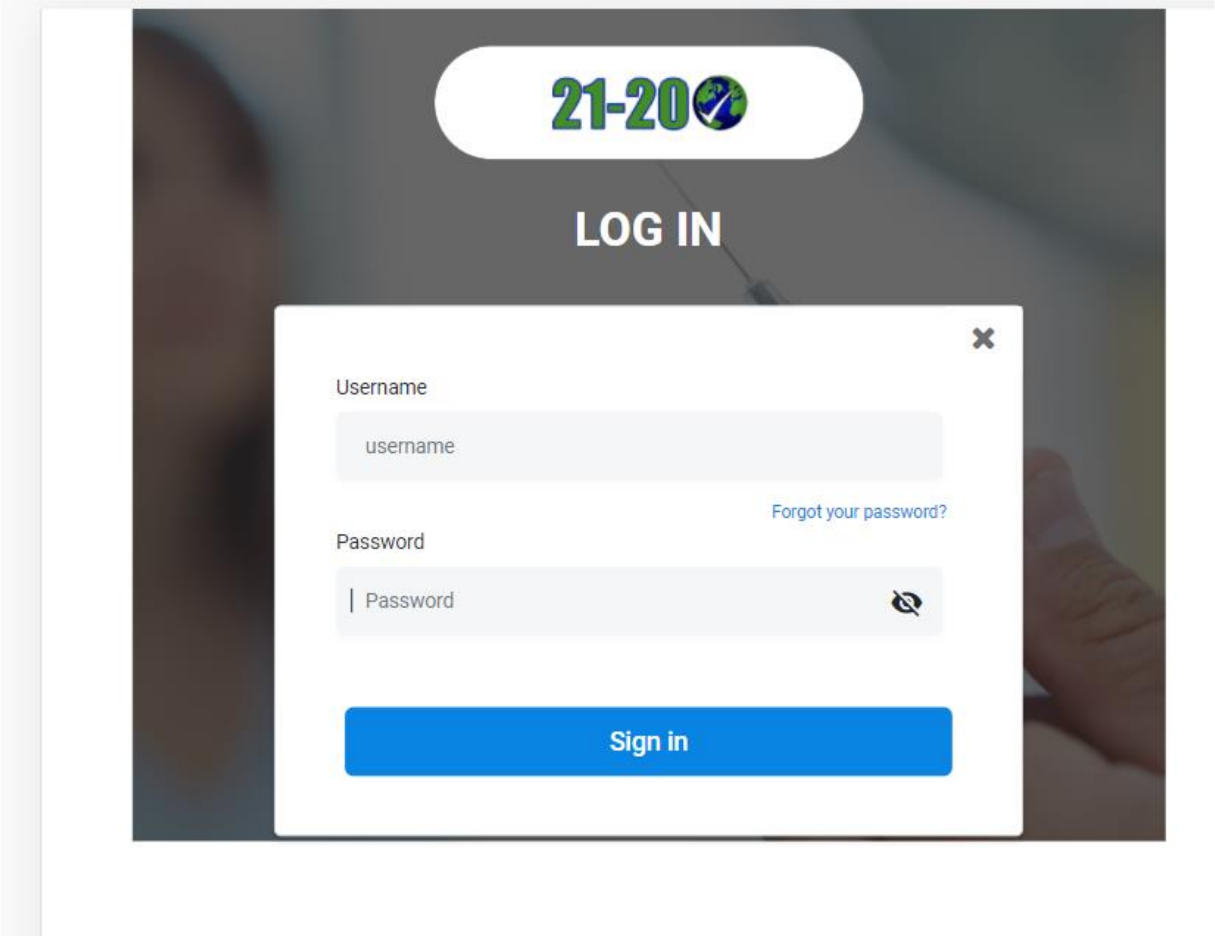
Quickly create an account for all Covid-19 vaccine related information, check vaccine availability and eligibility, as well as receive communications from 21-20 .

Already registered? Login

Vaccine Provider

Like to list a pharmacy, clinic, health department or other vaccine provider? You will first need to register for an account. After your account is approved, you can submit information about your vaccination services.

Get vaccinated at a location near you



21-20

LOG IN

Username

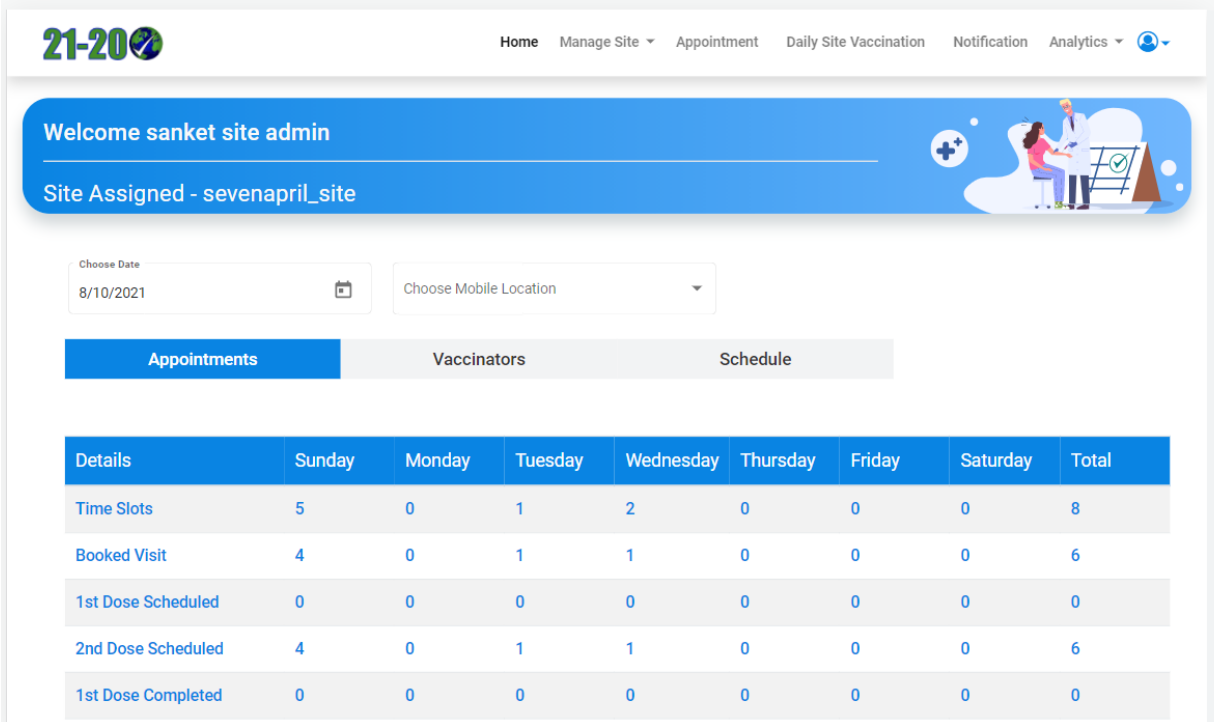
username

Forgot your password?

Password

Password

Sign in



21-20

Home | Manage Site | Appointment | Daily Site Vaccination | Notification | Analytics

Welcome sanket site admin

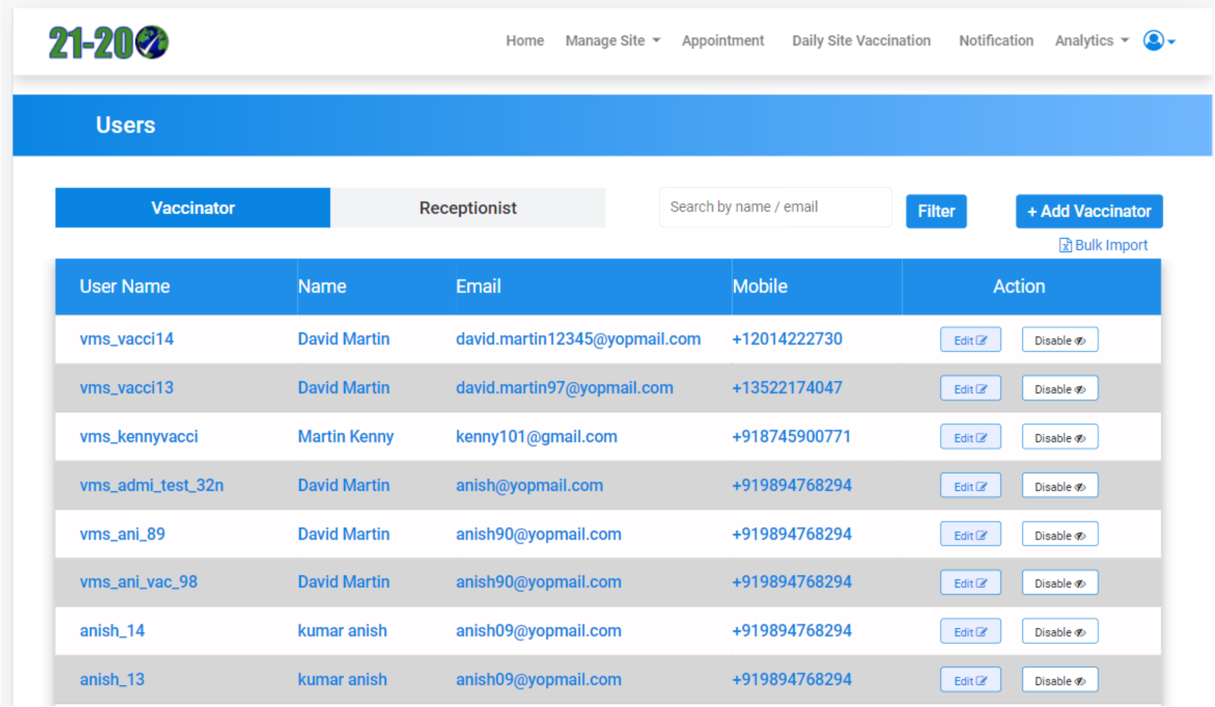
Site Assigned - sevenapril_site

Choose Date: 8/10/2021

Choose Mobile Location

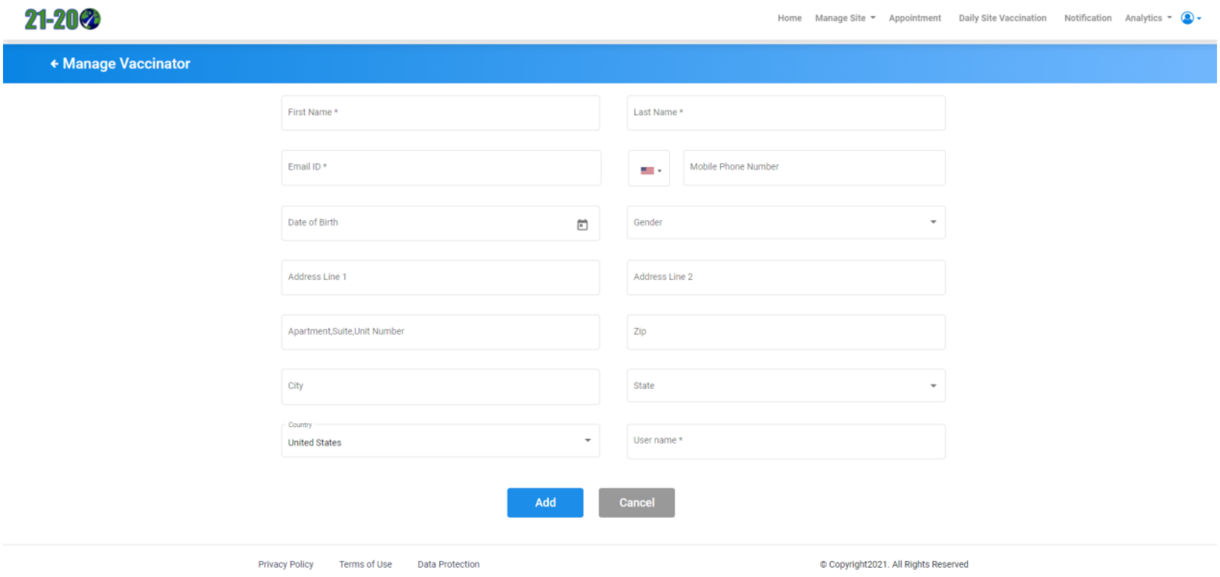
Appointments | Vaccinators | Schedule

Details	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Total
Time Slots	5	0	1	2	0	0	0	8
Booked Visit	4	0	1	1	0	0	0	6
1st Dose Scheduled	0	0	0	0	0	0	0	0
2nd Dose Scheduled	4	0	1	1	0	0	0	6
1st Dose Completed	0	0	0	0	0	0	0	0



Users

User Name	Name	Email	Mobile	Action
vms_vacci14	David Martin	david.martin12345@yopmail.com	+12014222730	Edit Disable
vms_vacci13	David Martin	david.martin97@yopmail.com	+13522174047	Edit Disable
vms_kennyvacci	Martin Kenny	kenny101@gmail.com	+918745900771	Edit Disable
vms_admi_test_32n	David Martin	anish@yopmail.com	+919894768294	Edit Disable
vms_ani_89	David Martin	anish90@yopmail.com	+919894768294	Edit Disable
vms_ani_vac_98	David Martin	anish90@yopmail.com	+919894768294	Edit Disable
anish_14	kumar anish	anish09@yopmail.com	+919894768294	Edit Disable
anish_13	kumar anish	anish09@yopmail.com	+919894768294	Edit Disable



Manage Vaccinator

First Name *
Last Name *
Email ID *
Mobile Phone Number
Date of Birth
Gender
Address Line 1
Address Line 2
Apartment,Suite,Unit Number
Zip
City
State
Country: United States
User name *

CHAPTER 5

CODING

5.1 HTML PART

LOG-IN PAGE

```
<div class="main-div-bg-conatiner container" style="display: flex; justify-content: center;">

  <div style="flex-direction: column;">

    <div class="main-div-logo-area white-bg cursor-pointer">

      <!-- <a href="javascript:void(0);" [routerLink]="'/'"> -->

        <img (click)="backClicked()" src="{{ logo}} ">

      <!-- </a> -->

    </div>



    <p class="d-flex justify-content-center roboto-bold-33 white-text py-3">LOG IN</p>

    <div class="d-flex justify-content-center  login-container mb-5">



      <form class="sign-in pt-0" [formGroup]="authSignInForm" (ngSubmit)="authSignInFormSub()">

        <!-- <button class="vms-btn-sm light-blue-bg white-text roboto-medium-12 ml-1 mr-1" type="button" (click)="getUserType('vaccinator')">I want Vaccine</button>

        <button class="vms-btn-sm blue-bg white-text roboto-medium-12 ml-1 mr-1" type="button" (click)="getUserType('vaccinator')">Vaccinator</button>
```

```
                <button class="vms-btn-sm nevy-blue-bg  white-text roboto-medium-12 ml-1 mr-1" type="button" (click)="getUserType('sadmin')">Site Admin</button>

                <button class="vms-btn-sm dark-blue-bg white-text roboto-medium-12 ml-1 mr-1" type="button" (click)="getUserType('fadmin')">Facility Admin</button> -->



                <div class="sign-in-div">

                    <a class="list-group-item float-right border-0" href="javascript:void(0);" [routerLink]="'/'">

                        <i class="fa fa-times" aria-hidden="true"></i></a>

                    <!-- <p class="roboto-regular-14 text-left ml-40">Email address</p> -->

                    <div class="mb-3 mx-5">

                        <label for="exampleFormControlInput1"

                            class="form-label d-flex justify-content-start w-100 text-left">Username</label>



                        <!-- <mat-label>Phone Number</mat-label> -->

                        <input type="text" class="form-control sign-in-input-field white-2-bg" placeholder="username"

                            formControlName="user" />



                        <!-- <input type="text" class="form-control sign-in-input-field white-2-bg" id="exampleFormControlInput1" placeholder="Username" formControlName="user"> -->

                    </div>

                    <div class="d-flex justify-content-end">

                        <button type="button" class="btn roboto-regular-12 border-0 shadow-none py-0 mr-5 blue-text"
```

```
        [routerLink]="'/auth/admin-reset-password'">Forgot your password?</button>
    </div>
    <div class="mb-3 mx-5">
      <label for="exampleFormControlInput2" class="form-label w-100 text-left">Password</label>


      <input [type]="passwordToggler ? 'password' : 'text'"
        class="form-control sign-in-input-field white-2-bg" id="exampleFormControlInput2"
        placeholder="Password" formControlName="password" required>


      <mat-icon style="position: relative; top: -35px; right: -180px; cursor: pointer;" matSuffix
        (click)="passwordToggler = !passwordToggler">{{passwordToggler ?
        'visibility_off':'visibility' }}
      </mat-icon>
    </div>
    <!-- <p id="password-lable" class="roboto-regular-14 text-left ml-40">Password</p> -->


    <div class="sign-in-btn-container mb-3 mx-5">
      <button class="sign-in-btn blue-bg white-text roboto-medium-18" type="submit">Sign in</button>
    </div>
    <!-- <div class="d-flex justify-content-center ml-2">
```

```
            <span class="roboto-regular-14">Do not have an account?</span><button type="button" class="btn roboto-regular-14 border-0 shadow-none py-0 blue-text"

              [routerLink]="'/auth/sign-up'" required>Register Now</button>

          </div> -->

        </div>

        <div class="social-media-login">

          <!-- <p>Or Login with</p> -->

          <!-- <ul>

            <li><a href="#" title="google plus" class="gp"><i class="fa fa-google-plus"></i></a>

            </li>

            <li><a href="#" title="facebook" class="fb"><i class="fa fa-facebook"></i></a></li>

            <li><a href="#" title="twitter" class="tw"><i class="fa fa-twitter"></i></a></li>

            <li><a href="#" title="instagram" class="inst"><i class="fa fa-instagram"></i></a>

            </li>

          </ul> -->

        </div>

      </form>

    </div>

  </div>

</div>
```

=

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { SiteAdminComponent } from './site-admin.component';
import { PageDashboardComponent } from './page-dashboard/page-dashboard.component';
import { PageManageSeatComponent } from './page-manage-seat/page-manage-seat.component';
import { PageAddSeatComponent } from './page-manage-seat/page-add-seat/page-add-seat.component';
import { NoSeatComponent } from './page-manage-seat/no-seat/no-seat.component';
import { PageListSeatComponent } from './page-manage-seat/page-list-seat/page-list-seat.component';
import { PageVaccinatorComponent } from './page-vaccinator/page-vaccinator.component';
import { PageAddVaccinatorComponent } from './page-vaccinator/page-add-vaccinator/page-add-vaccinator.component';
import { PageListVaccinatorComponent } from './page-vaccinator/page-list-vaccinator/page-list-vaccinator.component';
import { PageVaccineComponent } from './page-vaccine/page-vaccine.component';
import { PageAddVaccineComponent } from './page-vaccine/page-add-vaccine/page-add-vaccine.component';
import { PageListVaccineComponent } from './page-vaccine/page-list-vaccine/page-list-vaccine.component';
import { PageEditVaccineComponent } from './page-vaccine/page-edit-vaccine/page-edit-vaccine.component';
import { PageManageCalenderComponent } from './page-manage-calender/page-manage-calender.component';
import { EditSeatComponent } from './page-manage-seat/edit-seat/edit-seat.component';
import { EditVaccinatorComponent } from './page-vaccinator/edit-vaccinator/edit-vaccinator.component';
import { PageBulkComponent } from './page-bulk/page-bulk.component';
import { PageBulkRecipientComponent } from './page-bulk/page-bulk-recipient/page-bulk-recipient.component';
import { SiteSettingComponent } from './site-setting/site-setting.component';
import { PageVaccinatorScheduleComponent } from './page-vaccinator-schedule/page-vaccinator-schedule.component';
import { BookAppointmentComponent } from './book-appointment/book-appointment.component';
import { PageVaccineReportComponent } from './page-vaccine/page-vaccine-report/page-vaccine-report.component';
import { PreviewCalendarComponent } from './page-manage-calender/preview-calendar/preview-calendar.component';
import { PageNewRegistrationComponent } from './page-new-registration/page-new-registration.component';
import { PageChangeUserIdPassComponent } from './page-change-user-id-pass/page-change-user-id-pass.component';
import { PageAppointmentComponent } from './page-appointment/page-appointment.component';
import { PageAddReceptionistComponent } from './page-receptionist/page-add-receptionist/page-add-receptionist.component';
import { PageEditReceptionistComponent } from './page-receptionist/page-edit-receptionist/page-edit-receptionist.component';
import { PageNoReceptionistComponent } from './page-receptionist/page-no-receptionist/page-no-receptionist.component';
import { ReceptionistListComponent } from './receptionist-list/receptionist-list.component';
import { ReceptionistCreateComponent } from './receptionist-create/receptionist-create.component';
import { PageEditRecepientComponent } from './page-edit-recepient/page-edit-recepient.component';
import { PageViewSeatComponent } from './page-view-seat/page-view-seat.component';
import { PageViewSeatScheduleComponent } from './page-view-seat-schedule/page-view-seat-schedule.component';
import { DailysitevaccinationComponent } from './dailysitevaccination/dailysitevaccination.component';
import { PageRecipientAppointmentComponent } from './page-recipient-appointment/page-recipient-appointment.component';
import { AdverseEventReportingComponent } from '../../pages/adverse-event-reporting/adverse-event-reporting.component';
import { PagePowerBiComponent } from './page-power-bi/page-power-bi.component';
import { PagePowerBiDashboardComponent } from './page-power-bi-dashboard/page-power-bi-dashboard.component';
import { PagePowerBiReportComponent } from './page-power-bi-report/page-power-bi-report.component';
import { PageCdcCertificateComponent } from './page-cdc-certificate/page-cdc-certificate.component';
import { PageAddUserComponent } from './page-add-user-tabs/page-add-user.component';
const routes: Routes = [
  {
    path: '',
    component: SiteAdminComponent,
    data: { user_type: 'site_admin' },
    children: [
{ path: '', pathMatch: 'full', redirectTo: 'dashboard' },
{ path: 'site-setting', component: SiteSettingComponent },
{
  path: 'manage-seat',
  component: PageManageSeatComponent,
  data: { location: 'seat' },
},
{
  path: 'add-seat/:steps',
  component: PageAddSeatComponent,
  data: { location: 'seat' },
},
{
  path: 'add-seat',
  component: PageAddSeatComponent,
  data: { location: 'seat' },
},
{
  path: 'no-seat',
  component: NoSeatComponent,
  data: { location: 'seat' },
},
{
  path: 'list-seat',
  component: PageListSeatComponent,
  data: { location: 'seat' },
},
{
  path: 'list-seat/:steps',
  component: PageListSeatComponent,
  data: { location: 'seat' },
},
{
  path: 'edit-seat/:steps/:id',
  component: EditSeatComponent
},
{
  path: 'edit-seat/:id',
  component: EditSeatComponent
},
{
  path: 'vaccinator',
  component: PageVaccinatorComponent,
  data: { location: 'vaccinator' },
},
{
  path: 'add-vaccinator/:steps',
  component: PageAddVaccinatorComponent,
  data: { location: 'vaccinator' },
},
{
  path: 'add-vaccinator',
  component: PageAddVaccinatorComponent,
  data: { location: 'vaccinator' },
},
{
  path: 'list-vaccinator',
  component: PageListVaccinatorComponent,
  data: { location: 'vaccinator' },
},
{
  path: 'list-vaccinator/:steps',
  component: PageListVaccinatorComponent,
  data: { location: 'vaccinator' },
},
{
  path: 'edit-vaccinator/:steps/:id',
  component: EditVaccinatorComponent,
  data: { location: 'vaccinator' },
},
{
  path: 'edit-vaccinator/:id',
  component: EditVaccinatorComponent,
  data: { location: 'vaccinator' },
},
{
  path: 'vaccine',
  component: PageVaccineComponent,
  data: { location: 'vaccine' },
},
{
  path: 'add-vaccine/:steps',
  component: PageAddVaccineComponent,
  data: { location: 'vaccine' },
},
{
  path: 'add-vaccine',
  component: PageAddVaccineComponent,
  data: { location: 'vaccine' },
},
    {
      path: 'list-vaccine',
      component: PageListVaccineComponent,
      data: { location: 'vaccine' },
    },
    {
      path: 'list-vaccine/:steps',
      component: PageListVaccineComponent,
      data: { location: 'vaccine' },
    },
    {
      path: 'edit-vaccine/:steps/:id',
      component: PageEditVaccineComponent,
      data: { location: 'vaccine' },
    },
    {
      path: 'edit-vaccine/:id',
      component: PageEditVaccineComponent,
      data: { location: 'vaccine' },
    },
    {
      path: 'vaccine-report',
      component: PageVaccineReportComponent,
      data: { location: 'vaccine' },
    },
    { path: 'manage-calender/:steps', component:
PageManageCalenderComponent },
    { path: 'manage-calender', component:
PageManageCalenderComponent },
    { path: 'preview-calender/:steps', component:
PreviewCalendarComponent },
    { path: 'preview-calender', component: PreviewCalendarComponent },
    { path: 'bulk-upload', component: PageBulkComponent },
    { path: 'bulk-recipient', component: PageBulkRecipientComponent },
    {
      path: 'vaccinator-schedule',
      component: PageVaccinatorScheduleComponent,
    },
    { path: 'change-userid', component: PageChangeUserIdPassComponent
},
    { path: 'new-registration', component: PageNewRegistrationComponent
},
    {
      path: 'appointment',
      loadChildren: () =>
        import('./page-appointment/page-appointment.module').then(
          (m) => m.PageApointmentModule
        ),
    },
    {
      path: 'appointment/:pageIndex',
      loadChildren: () =>
        import('./page-appointment/page-appointment.module').then(
          (m) => m.PageApointmentModule
        ),
    },
    {
      path: 'appointment/:pageIndex/:pageSize',
      loadChildren: () =>
```

```
        import('./page-appointment/page-appointment.module').then(
          (m) => m.PageApointmentModule
        ),
    },
    {
      path: 'appointment/:pageIndex/:pageSize/:searchText',
      loadChildren: () =>
        import('./page-appointment/page-appointment.module').then(
          (m) => m.PageApointmentModule
        ),
    },
    {
      path: 'book-appointment/:id',
      component: BookAppointmentComponent,
      data: { location: 'admin_appointment' },
    },
    {
      path: 'recipient-appointment/:id',
      component: PageRecipientAppointmentComponent,
    },
    {
      path: 'recipient-appointment',
      component: PageRecipientAppointmentComponent,
    },
    { path: 'adverse-event/:id', component:
AdverseEventReportingComponent },
    {
      path: 'add-receptionist',
      component: PageAddReceptionistComponent,
      data: { location: 'admin_appointment' },
    },
    {
      path: 'edit-recepient/:id',
      component: PageEditRecepientComponent,
      data: { location: 'admin_appointment' },
    },

    {
      path: 'edit-receptionist',
      component: PageEditReceptionistComponent,
      data: { location: 'admin_appointment' },
    },
    {
      path: 'no-receptionist',
      component: PageNoReceptionistComponent,
      data: { location: 'receptionist' },
    },
    {
      path: 'list-receptionist/:steps',
      component: ReceptionistListComponent,
      data: { location: 'receptionist' },
    },
    {
      path: 'list-receptionist',
      component: ReceptionistListComponent,
      data: { location: 'receptionist' },
    },
    {
    path: 'receptionist-create/:steps',
    component: ReceptionistCreateComponent
```

```
},
{
path: 'receptionist-create',
component: ReceptionistCreateComponent
},
{
  path: 'receptionist-update/:steps/:id',
  component: ReceptionistCreateComponent,
  data: { location: 'receptionist' },
},
{
  path: 'receptionist-update/:id',
  component: ReceptionistCreateComponent,
  data: { location: 'receptionist' },
},
{ path: 'dailysite', component: DailysitevaccinationComponent },

{
  path: 'notification-setting',
  loadChildren: () =>
    import('./notification-setting/notification-setting.module').then(
      (m) => m.NotificationSettingModule
    ),
},
{
  path: 'analytics',
  component: PagePowerBiComponent,
},
{
  path: 'seat-view/:id',
        component: PageViewSeatComponent,
        data: { location: 'seat' },
      },
      {
        path: 'seat-view',
        component: PageViewSeatScheduleComponent,
        data: { location: 'seat' },
      },
      {
        path: 'dashboard',
        data: { location: 'dashboard_report' },
        loadChildren: () =>
          import('./page-dashboard/site-admin-page-dashboard.module').then(
            (m) => m.SitAdminPageDashboardModule
          ),
      },
      {
        path: 'analatics/dashboard',
        component: PagePowerBiDashboardComponent,
      },
      {
        path: 'analatics/report',
        component: PagePowerBiReportComponent,
      },
      { path: 'cdc-certificate/:id', component: PageCdcCertificateComponent
},
      { path: 'add-user', component: PageAddUserComponent },
    ],
  },
];
```

```
@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class SiteAdminRoutingModule {}lblStatus.Content =
Properties.Resources.SuccessMessage;                lblStatus.Foreground =
Brushes.Green;
            }
else
            {
                lblStatus.Content               =               Properties.Resources.LessItemSelection;
lblStatus.Foreground = Brushes.Red;
            }
        }

        private void HourIncreament(object sender, RoutedEventArgs e)
        {
            userTime.Value += new TimeSpan(1, 0, 0);
        }

        private void HourDecreament(object sender, RoutedEventArgs e)
        {
            userTime.Value -= new TimeSpan(1, 0, 0);
        }

        private void MinuteIncreament(object sender, RoutedEventArgs e)
        {
            userTime.Value += new TimeSpan(0, 1, 0);
        }
        private void MinuteDecreament(object sender, RoutedEventArgs e)
        {
            userTime.Value -= new TimeSpan(0, 1, 0);
        }

        private void AmPmClick(object sender, RoutedEventArgs e)
}
import { DatePipe, Location } from '@angular/common';
import { Component, Input, OnInit } from '@angular/core';
import {
  AbstractControl,
  FormBuilder,
  FormGroup,
  ValidatorFn,
  Validators,
} from '@angular/forms';
import { ActivatedRoute, Router } from '@angular/router';
import { RecipientAuthService } from 'src/app/core/services/auth';
import { CountriesService } from 'src/app/shared/services/countries.service';
import { NotificationService } from 'src/app/shared/services/notification.service';

import {
  GENDER_TYPES
  } from '../../../../shared/helpers/constant';

import { SiteAdminService } from '../../../../core';

@Component({
  selector: 'app-page-add-vaccinator',
  templateUrl: './page-add-vaccinator.component.html',
  styleUrls: ['./page-add-vaccinator.component.scss'],
})
export class PageAddVaccinatorComponent implements OnInit {
  iso2: string = '';
  [x: string]: any;
  isEdit = false;
  addVaccinatorForm: FormGroup;
  @Input() vaccinator: any;
  @Input() id: string;
  dialCode: string = '+1';
  stateInfo: any[] = [];
  countryInfo: any[] = [];
  cityInfo: any[] = [];
  public stateshow: boolean = true;
  // stateInfo: any[] = [];
  dontCallAPi: boolean = false;
  genderData  = GENDER_TYPES;
  steperRoute: boolean = false
  Countries = [
    {
      States: [
        {
          Cities: [
            'Eshkashem',
            'Fayzabad',
            'Jurm',
            'Khandud',
            "Qal'eh-ye Panjeh",
          ],
          StateName: 'Badakhshan',
        },
        { Cities: ['Bala Morghab', "Qal'eh-ye Naw"], StateName: 'Badgis' },
        {
          Cities: [
            'Andarab',
            'Baghlan',
            'Dahaneh-ye Ghawri',
            'Nahrin',
            'Pol-e Khumri',
          ],
          StateName: 'Baglan',
        },
        {
          Cities: [
            'Balkh',
            'Dawlatabad',
            'Mazar-e Sharif',
            'Qarchi Gak',
            'Shulgara',
            'Tash Gozar',
          ],
          StateName: 'Balkh',
        },
        { Cities: ['Bamiyan', 'Panjab', "Qil Qal'eh"], StateName: 'Bamiyan' },
        { Cities: ['Anar Darreh', 'Farah', 'Shindand'], StateName: 'Farah' },
        {
          Cities: ['Andkhvoy', 'Darzi Ab', 'Dawlatabad', 'Maymanah'],
          StateName: 'Faryab',
        },
        { Cities: ['Chaghcharan', 'Shahrak', 'Taywarah'], StateName: 'Gawr' },
        { Cities: ['Ghazni'], StateName: 'Gazni' },
        {
          Cities: [
            'Awbeh',
            "Eslam Qal'eh",
            'Ghurian',
            'Herat',
            'Karukh',
            'Kuhestan',
            'Kushk',
            'Qarabagh',
            'Tawraghudi',
            'Tir Pol',
            'Zendejan',
          ],
          StateName: 'Herat',
        },
        {
          Cities: [
            'Baghran',
            'Darwishan',
            'Deh Shu',
            'Gereshk',
            'Lashkar Gah',
            'Sangin',
          ],
          StateName: 'Hilmand',
        },
        {
          Cities: ['Aqchah', 'Qarqin', 'Sang-e Charak', 'Shibarghan'],
          StateName: 'Jawzjan',
        },
        {
          Cities: [
            'Baghrami',
            'Kabul',
            'Mir Bachchekut',
            'Paghman',
            'Qarabagh',
            'Sarawbi',
          ],
          StateName: 'Kabul',
        },
        { Cities: ['Mahmud-e Raqi', 'Taghab'], StateName: 'Kapisa' },
        { Cities: ['Khawst'], StateName: 'Khawst' },
        { Cities: ['Asadabad', 'Asmar'], StateName: 'Kunar' },
        { Cities: ['Mehtar Lam'], StateName: 'Lagman' },
        {
          Cities: ['Azraw', 'Baraki Barak', 'Pol-e Alam'],
          StateName: 'Lawghar',
        },
        {
          Cities: [
            'Achin',
            'Batsawul',
            'Hugyani',
            'Jalalabad',
            'Nader Shah Kawt',
          ],
          StateName: 'Nangarhar',
        },
        {
          Cities: [
            'Chahar Burjak',
            'Chakhansur',
            'Khash',
            'Mirabad',
            'Rudbar',
            'Zaranj',
          ],
          StateName: 'Nimruz',
        },
        { Cities: ['Nuristan'], StateName: 'Nuristan' },
        {
          Cities: ['Orgun', 'Zareh Sharan', 'Zarghun Shahr'],
          StateName: 'Paktika',
        },
        { Cities: ["'Ali Khayl", 'Ghardez'], StateName: 'Paktiya' },
        { Cities: ['Charikar', 'Jabal-os-Saraj'], StateName: 'Parwan' },
        { Cities: ['Qandahar'], StateName: 'Qandahar' },
        {
          Cities: [
            'Dasht-e Archa',
            'Emam Saheb',
            'Hazart Imam',
            'Khanabad',
            "Qal'eh-ye Zal",
            'Qunduz',
          ],
                StateName: 'Qunduz',
              },
              { Cities: ['Aybak', 'Kholm'], StateName: 'Samangan' },
              { Cities: [], StateName: 'Sar-e Pul' },
              {
                Cities: [
                  'Chah Ab',
                  'Eshkamesh',
                  'Farkhar',
                  'Khwajeh Ghar',
                  'Rostaq',
                  'Taloqan',
                  "Yangi Qal'eh",
                ],
                StateName: 'Takhar',
              },
              {
                Cities: ['Deh Rawud', 'Gaz Ab', 'Tarin Kawt', 'Uruzgan'],
                StateName: 'Uruzgan',
              },
              { Cities: ['Gardan Diwal', 'Maydanshahr'], StateName: 'Wardag' },
              { Cities: ['Qalat-e Ghilzay'], StateName: 'Zabul' },
            ],
            CountryName: 'Afghanistan',
          },
          {
            States: [
              { Cities: ['Berat', 'Polican', 'Ure Vajgurore'], StateName: 'Berat' },
              { Cities: ['Bulqize'], StateName: 'Bulqize' },
              { Cities: ['Delvine'], StateName: 'Delvine' },
        { Cities: ['Bilisht'], StateName: 'Devoll' },
        { Cities: ['Maqellare', 'Peshkopi'], StateName: 'Dibre' },
        { Cities: ['Durres', 'Shijak'], StateName: 'Durres' },
        { Cities: ['Cerrik', 'Elbasan', 'Kerrabe'], StateName: 'Elbasan' },
        { Cities: ['Fier', 'Patos', 'Roskovec'], StateName: 'Fier' },
        { Cities: ['Gjirokaster', 'Libohove'], StateName: 'Gjirokaster' },
        { Cities: ['Gramsh'], StateName: 'Gramsh' },
        { Cities: ['Krume'], StateName: 'Has' },
        { Cities: ['Kavaje'], StateName: 'Kavaje' },
        { Cities: ['Erseke', 'Leskovik'], StateName: 'Kolonje' },
        { Cities: ['Korce', 'Maliq'], StateName: 'Korce' },
        {
          Cities: ['Fushe-Kruje', 'Kruje', 'Mamurras', 'Milot'],
          StateName: 'Kruje',
        },
            'Chocen',
            'Chrudim',
            'Chvaletice',
            'Hermanuv Mestec',
            'Hlinsko',
            'Lanshkroun',
            'Letohrad',
            'Litomyshl',
            'Moravska Trebova',
            'Pardubice',
            'Polichka',
            'Policka',
            'Prelouch',
            'Skutech',
            'Svitavy',
            'Usti nad Orlici',
            'Vysoke Myto',
            'Zhamberk',
          ],
          StateName: 'Pardubicky',
        },
        {
          Cities: [
            'Dobrany',
            'Domazhlice',
            "Horazhd'ovice",
            'Horshovky Tyn',
            'Kdyne',
            'Klatovy',
            'Nyrany',
            'Nyrsko',
            'Plana',
            'Plzen',
            'Preshtice',
            'Radnice',
            'Rokycany',
            'Stribro',
            'Sushice',
            'Tachov',
          ],
          StateName: 'Plzensky',
        },
        { Cities: ['Prague', 'Praha'], StateName: 'Praha' },
        { Cities: ['Rajhrad'], StateName: 'Rajhrad' },
        { Cities: ['Smirice'], StateName: 'Smirice' },
        { Cities: [], StateName: 'South Moravian' },
        { Cities: [], StateName: 'Straz nad Nisou' },
        {
          Cities: [
            'Benatky nad Jizerou',
            'Beneshov',
            'Beroun',
            'Brandys nad Labem-Stara Bolesl',
            'Chaslav',
            'Chavaletice',
            'Chelakovice',
            'Chesky Brod',
            'Dobrish',
            'Horovice',
            'Kladno',
            'Kolin',
            'Kralupy nad Vltavou',
            'Kutna Hora',
            'Lysa nad Labem',
            'Melnik',
            'Mlada Boleslav',
            'Mnichovo Hradishte',
            'Neratovice',
            'Nove Strasheci',
            'Nymburk',
            'Podebrady',
            'Pribram',
            'Rakovnik',
            'Richany',
            'Rousinov',
            'Roztoky',
            'Sedlcany',
            'Slany',
            'Stochov',
            'Vlashim',
            'Zruch nad Sazavou',
          ],
          StateName: 'Stredochesky',
        },
        { Cities: ['Unicov'], StateName: 'Unicov' },
        {
          Cities: [
            'Bilina',
            'Chomutov',
            'Dechin',
            'Dubi',
            'Duchcov',
            'Jilove',
            'Jirkov',
            'Kadan',
            'Klasterec nad Ohri',
            'Krupka',
            'Litomerice',
            'Litvinov',
            'Louny',
            'Lovosice',
            'Mezibori',
            'Most',
            'Osek',
            'Podborany',
            'Roudnice',
            'Rumburk',
            'Shluknov',
            'Shteti',
            'Teplice',
            'Usti',
            'Varnsdorf',
            'Zatec',
          ],
          StateName: 'Ustecky',
        },
        { Cities: ['Valletta'], StateName: 'Valletta' },
        { Cities: ['Velesin'], StateName: 'Velesin' },
        {
          Cities: [
            'Bystrice nad Pernshtejnem',
            'Chotebor',
            'Havlichkuv Brod',
            'Humpolec',
            'Jihlava',
            'Ledech',
            'Moravske Budejovice',
            'Nove Mesto na Morave',
            'Okrisky',
            'Pacov',
            'Pelhrimov',
            'Polna',
            'Svetla nad Sazavou',
            'Telch',
            'Trebich',
            "Tresht'",
            'Velke Mezirichi',
            "Zhd'ar",
          ],
          StateName: 'Vysochina',
        },
        {
          Cities: [
            'Brumov',
            'Bystrice pod Hostynem',
            'Chropyne',
            'Holeshov',
            'Hulin',
            'Kromerizh',
            'Kunovice',
            'Napajedla',
            'Otrokovice',
            'Rozhnov',
            'Roznov pod Radhostem',
            'Slavicin',
            'Slusovice',
            'Stare Mesto',
            'Strazhnice',
            'Uherske Hradishte',
            'Uhersky Brod',
            'Valashske Klobouky',
            'Valashske Mezirichi',
            'Veseli nad Moravou',
            'Vsetin',
            'Zborovice',
                'Zlin',
              ],
              StateName: 'Zlinsky',
            },
          ],
          CountryName: 'Czech Republic',
        },
        {
          States: [
            {
              Cities: [
                'Aarhus',
                'Allingabro',
                'Arhus',
                'Assentoft',
                'Auning',
                'Beder',
                'Brabrand',
                'Ebeltoft',
                'Framlev',
                'Galten',
                'Grenaa',
                'Hadsten',
                'Hammel',
                'Hinnerup',
                'Hjortshoj',
                'Horning',
                'Hornslet',
                'Kolt',
                'Langa',
            'Logten',
            'Lystrup',
            'Malling',
            'Mariager',
            'Marslet',
            'Odder',
            'Randers',
            'Risskov',
            'Ronde',
            'Ry',
            'Ryomgard',
            'Sabro',
            'Silkeborg',
            'Skanderborg',
            'Skovby',
            'Soften',
            'Solbjerg',
            'Spentrup',
            'Stavtrup',
            'Stilling',
            'Svejbak',
            'Tranbjerg',
            'Trige',
            'Virklund',
          ],
          StateName: 'Arhus',
        },
        {
          Cities: ['Aakirkeby', 'Allinge-Sandvig', 'Nexo', 'Ronne'],
          StateName: 'Bornholm',
        },
        {
         Cities: [
           'Allerod',
           'Birkerod',
           'Blovstrod',
           'Espergarde',
           'Farum',
           'Fredensborg',
           'Frederikssund',
           'Frederiksvark',
           'Ganlose',
           'Gilleleje',
           'Grasted',
           'Hellebak',
           'Helsinge',
           'Helsingor',
           'Hillerod',
           'Hornbak',
           'Horsholm',
           'Humlebak',
           'Hundested',
           'Jagerspris',
           'Kvistgaard',
           'Lillerod',
           'Liseleje',
           'Lynge',
           'Niva',
           'Nodebo',
           'Olstykke',
           'Skibby',
           'Slangerup',
           'Stavnsholt',
           'Stenlose',
           'Valby',
           'Vekso',
         ],
         StateName: 'Frederiksborg',
        },
        {
         Cities: [
           'Aarup',
           'Arslev',
           'Assens',
           'Bellinge',
           'Blommenslyst',
           'Bogense',
           'Brenderup',
           'Broby',
           'Bullerup',
           'Ejby',
           'Faaborg',
           'Glamsbjerg',
           'Haarby',
           'Hojby',
           'Kerteminde',
           'Langeskov',
           'Marstal',
           'Middelfart',
           'Munkebo',
           'Neder Holluf',
           'Norre Aaby',
           'Nyborg',
           'Odense',
           'Otterup',
           'Ringe',
           'Rudkobing',
           'Sankt Klemens',
           'Seden',
           'Sonderso',
           'Stige',
           'Strib',
           'Svendborg',
           'Thuro',
           'Tommerup',
           'Ullerslev',
           'Vindeby',
           'Vissenbjerg',
         ],
         StateName: 'Fyn',
        },
        {
         Cities: ['Ballerup', 'Brondby', 'Stenlose', 'Vallensbaek'],
         StateName: 'Hovedstaden',
        },
        {
         Cities: [
           'Dragor',
           'Flong',
           'Gentofte',
           'Glostrup',
           'Herlev',
           'Hvidovre',
           'Ishoj',
           'Kastrup',
           'Lyngby',
           'Malov',
           'Smorumnedre',
           'Taastrup',
           'Trorod',
           'Vanlose',
           'Varlose',
         ],
         StateName: 'Kobenhavn',
        },
        { Cities: [], StateName: 'Kobenhavns Amt' },
        { Cities: [], StateName: 'Kobenhavns Kommune' },
        {
         Cities: [
           'Aabybro',
           'Aalborg',
           'Aars',
           'Arden',
           'Bindslev',
           'Bronderslev',
           'Brovst',
           'Dronninglund',
           'Farso',
           'Fjerritslev',
           'Frederikshavn',
        'Frejlev',
        'Gistrup',
        'Gorlose',
        'Hadsund',
        'Hals',
        'Hirtshals',
        'Hjallerup',
        'Hjorring',
        'Hobro',
        'Kas',
        'Klarup',
        'Logstor',
        'Nibe',
        'Norresundby',
        'NÃƒ Æ’ Ã, Æ’ Ãƒ â€ š Ã, Â¸rresundby',
        'Pandrup',
        'Saby',
        'Sindal',
        'Skagen',
        'Skorping',
        'Storvorde',
        'Stovring',
        'Strandby',
        'Sulsted',
        'Svenstrup',
        'Tars',
        'Tranekaer',
        'Vadum',
        'Vestbjerg',
        'Vester Hassing',
          'Vodskov',
          'Vra',
        ],
        StateName: 'Nordjylland',
      },
      {
        Citi
  private _providerRecipientAuthService: RecipientAuthService,
  private notify: NotificationService,
  private country: CountriesService,
  private _location: Location,
  public datepipe: DatePipe,
  private activatedRoute: ActivatedRoute
) {
  const currentDateObj = new Date();
  const currentYear = currentDateObj.getFullYear();
  const currentMonth = currentDateObj.getMonth();
  const currentDate = currentDateObj.getDate();

  this.minDate = new Date(currentYear - 100, 0, 1);
  this.maxDate = new Date(currentYear - 18, currentMonth, currentDate);
  this.checkSteper();
}

// onCountryChange(event) {
//   this.dialCode = '+' + event.dialCode;
//   this.iso2 = event.iso2;
// }

checkSteper(){
  this.activatedRoute.params.subscribe( params => {
    if(params?.steps && params.steps === 'steps'){
      this.steperRoute = true;
    } else {
      this.steperRoute = false;
    }
  });
}

onLoadCountryChange(event) {
  setTimeout(() => {
    //alert(this.iso2)
    //console.log("event load 1",event);
    event.setCountry(this.iso2);
    let data: any = event.hasOwnProperty('s') ? event.s : event;
    console.log('event load', data, this.iso2);
    if (data && data.hasOwnProperty('dialCode')) {
      data.dial_code = data.dialCode;
      //delete data.dialCode;
    }
    this.addVaccinatorForm.controls.countryFlag.patchValue(data);
    //this.dialCode = data;
  }, 1000);
}

onCountryChange(event) {
  let data = event.hasOwnProperty('s') ? event.s : event;
  console.log('event change', data);
  if (data && data.hasOwnProperty('dialCode')) {
    data.dial_code = data.dialCode;
    //delete data.dialCode;
  }
  this.addVaccinatorForm.controls.countryFlag.patchValue(data);
  this.iso2 = data.iso2;
}

backClicked() {
  this._location.back();
}
ngOnInit(): void {
  if (this.id) {
    this.isEdit = true;
    this.show = true;
    this.stateshow = false;
  }
  this.buildForm();
  this.getCountries();
}

onChangeCountry(countryValues) {
  let countryValue = countryValues.value;
  this.stateInfo = this.countryInfo[countryValue].States;
  // this.cityInfo = this.stateInfo[0].Cities;
}

onChangeState(stateValues) {
  let stateValue = stateValues.value;
  // this.cityInfo = this.stateInfo[stateValue].Cities;
}
buildForm() {
    this.iso2 = !this.isEdit ? 'us' : '';
    this.addVaccinatorForm = this._formBuilder.group({
      firstName: [
        !!this.vaccinator ? this.vaccinator.fname : null,
        [Validators.required],
      ],
      lastName: [
        !!this.vaccinator ? this.vaccinator.lname : null,
        [Validators.required],
      ],
      phone: [
        null,
        [Validators.required],
      ],
      countryFlag: [''],
      email: [
        !!this.vaccinator ? this.vaccinator.email : null,
        [Validators.required,Validators.pattern('^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,4}$')],
      ],
      dob: [this.vaccinator && this.vaccinator.dob !='Invalid date' ?new Date( this.vaccinator.dob.replace(/-/g, '\/')) : ''],
      gender: [!!this.vaccinator ? this.vaccinator.gender : null],
      address1: [!!this.vaccinator ? this.vaccinator.address1 : null],
      address2: [!!this.vaccinator ? this.vaccinator.address2 : null],
      address3: [!!this.vaccinator ? this.vaccinator.address3 : null],
      city: [!!this.vaccinator ? this.vaccinator['city'] : ''],
      // city: [!!this.vaccinator ? this.vaccinator.city : null],
      // state: [!!this.vaccinator ? this.vaccinator.city : null],
      state: [!!this.vaccinator ? this.vaccinator['state'] : ''],
    // zip: [!!this.vaccinator ? this.vaccinator.zip : null],
    zip: [!!this.vaccinator ? this.vaccinator['zip'] : ''],
    country: [
      !!this.vaccinator ? this.vaccinator['country'] : 'United States',
    ],
    userName: [
      !!this.vaccinator ? this.vaccinator.user_name : null,
      [Validators.required],
    ],
  });
 if(this.vaccinator && this.vaccinator.hasOwnProperty('mobile_number')){
  this.patchPhoneValue()
 }  else {
  this.iso2 = 'us';
  this.addVaccinatorForm.controls.phone.patchValue(this.vaccinator?.mobile_number);
 }

}

patchPhoneValue(){
  if (this.vaccinator.mobile_number == null) {
    this.vaccinator.mobile_number = '';
    this.iso2 = 'us';
  } else {
    // for old records
    let iso2: number = this.iso2.toString().trim().length;
    let mob: number = this.vaccinator.mobile_number
      .toString()
      .trim().length;
    let flag =
  'countryFlag' in this.vaccinator &&
  this.vaccinator?.countryFlag?.hasOwnProperty(
    'dial_code'
  )
    ? this.vaccinator.countryFlag
    : '';
console.log('flag', flag);
if (mob !== 0 && !flag) {
  if (
    this.vaccinator.mobile_number.indexOf('+1') > -1
  ) {
    this.iso2 = 'us';
    this.vaccinator.mobile_number =
      this.vaccinator.mobile_number.replace('+1', '');
    console.log('event us');
  } else if (
    this.vaccinator.mobile_number.indexOf('+91') > -1
  ) {
    this.iso2 = 'in';
    this.vaccinator.mobile_number =
      this.vaccinator.mobile_number.replace('+91', '');
    console.log('event in');
  } else {
    this.iso2 = 'us';
  }
} else if (mob !== 0 && flag) {
  if (
    this.vaccinator.mobile_number.indexOf('+') > -1 &&
    this.vaccinator.mobile_number.indexOf(
      flag.dial_code
```

```
        ) > -1
      ) {
        this.vaccinator.mobile_number =
          this.vaccinator.mobile_number
            .replace(flag.dial_code, '')
            .trim();
      }
    } else {
      this.iso2 = 'us';
    }
  }
  let phone =  this.vaccinator.mobile_number
            ? this.vaccinator.mobile_number
            : '';
            console.log(phone);
  let countryFlag = 'countryFlag' in this.vaccinator &&
  this.vaccinator?.countryFlag?.hasOwnProperty('iso2')
              ? this.vaccinator.countryFlag
              : '';
  this.addVaccinatorForm.controls.countryFlag.patchValue(countryFlag);
  this.addVaccinatorForm.controls.phone.patchValue(phone);

}

getCountries() {
  this.country.allCountries().subscribe(
    (data2) => {
      this.countryInfo = data2.Countries;
      this.stateInfo = this.countryInfo['230'].States;
    },
    (err) => console.log(err),
    () => console.log('complete')
  );
}
searchInterest() {
  if (this.isEdit) {
    return;
  }
  let wordSearch = this.addVaccinatorForm.value.userName;
  setTimeout(() => {
    if (wordSearch == this.addVaccinatorForm.value.userName) {
      if (this.addVaccinatorForm.value.userName) {
        this.dontCallApi = false;
        this._siteAdminService.CheckUserName(wordSearch).subscribe(
          (res) => {
            this.dontCallAPi = res.result === 'unavailable' ? true : false;
            res.result === 'unavailable'
              ? this.notify.showNotification(
                  'User Name Alreay Exists',
                  'top',
                  'error'
                )
              : '';
          },
          (err) => {
            if (err.error.err.message == '') {
              this.notify.showNotification(
                'Something went wrong',
                'top',
                'error'
              );
            }
            this.notify.showNotification(
              err.error.err.message,
              'top',
              'error'
            );
          }
        );
      } else {
        this.autoCheckUser = false;
      }
    }
  }, 500);
}
fetchzipdetails(zipvalue) {
  if(zipvalue.value.length > 4){

  let zip = {
    zip: zipvalue.value,
  };

  this.addVaccinatorForm.get('zip').setErrors({ incorrect: true });
  this._providerRecipientAuthService.fetchdetails(zip).subscribe(
    (res) => {
      this.addVaccinatorForm.controls.city.setValue(res.city);
      this.addVaccinatorForm.controls.state.setValue(res.state);
      this.addVaccinatorForm.controls.country.setValue(res.country);
      //this.addVaccinatorForm.controls.county.setValue(res.county);
      this.show = true;
```

```
     this.stateshow = false;
     this.addVaccinatorForm.get('zip').setErrors({'incorrect': null});
     this.addVaccinatorForm.get('zip').updateValueAndValidity();
   },
   (err) => {
     this.getError()
     // this.addVaccinatorForm.get('zip').setErrors({ incorrect: true });
     this.notify.showNotification('Enter valid zip code', 'top', 'error');
   }
  );

}else{
  this.getError()
}
}
getError(){
  this.show = false;
  this.stateshow = true;
  this.addVaccinatorForm.controls.city.setValue('');
     this.addVaccinatorForm.controls.state.setValue('');
}

filterCountryFlag(data: any) {
  let object = { name: 'United States', iso2: 'us', dial_code: '+1' };
  if (data) {
    let checkdialcode = data.hasOwnProperty('dial_code') ? true : false;
    checkdialcode = checkdialcode ? data.dial_code.replace('+', '') : false;
    object.name = data.hasOwnProperty('name') ? data.name : object.name;
    object.iso2 = data.hasOwnProperty('iso2') ? data.iso2 : object.iso2;
    object.dial_code = checkdialcode ? '+' + checkdialcode : object.dial_code;
    }
    console.log('object', object, data);
    return object;
  }

  addVaccinatorFormSub() {
    if (!this.addVaccinatorForm.valid) {
      return;
    }

    if (this.dontCallAPi) {
      this.notify.showNotification('User Name Alreay Exists', 'top', 'error');
      return;
    }

    let formValue = this.addVaccinatorForm.value;
    const countryAndMobile = this.filterCountryFlag(formValue.countryFlag);
    formValue.dob = formValue.dob
      ? this.datepipe.transform(formValue.dob, 'yyyy-MM-dd')
      : '';
    // if (!this.isEdit) {
    //   formValue.phone = this.dialCode + formValue.phone;
    // }
    formValue.phone = countryAndMobile.dial_code + formValue.phone;
    formValue.countryFlag = countryAndMobile;
console.log("formValue",formValue);
//return false;
    this._siteAdminService.addVaccinator(formValue, this.id || null).subscribe(
      (res) => {
        console.log(res);
        if (res.status === 'success') {
          const msg = !!this.id
            ? 'Vaccinator updated successfully'
            : 'Vaccinator added successfully';
          this.notify.showNotification(msg, 'top', 'success');
          this._router.navigate([this.steperRoute ? '/site-admin/list-vaccinator/steps' : '/site-admin/list-vaccinator']);
        }
      },
      (err) => {
        const error = err.error;
        if (error.code === 'UsernameExistsException') {
          this.notify.showNotification(`${error.message}`, 'bottom', 'error');
        }
        if (error.code === 'InvalidParameterException') {
          this.notify.showNotification(`${error.message}`, 'bottom', 'error');
        }
      }
    );
  }
}
```

CHAPTER 6

DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model.

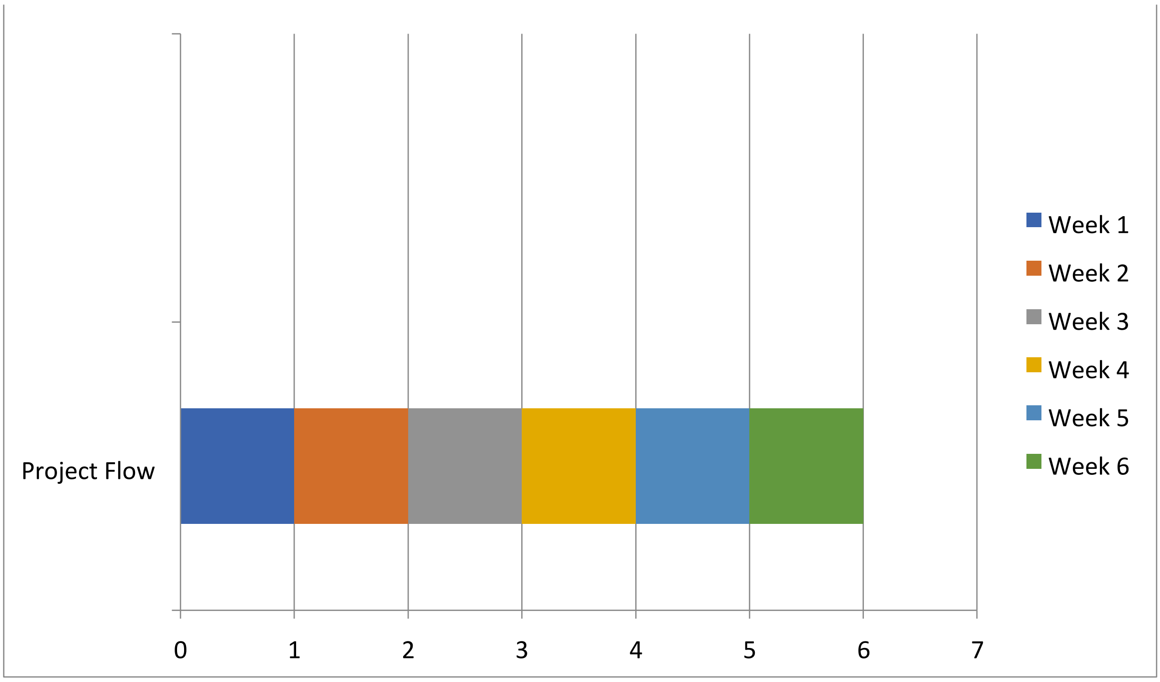
A picture is worth a thousand words. A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both.

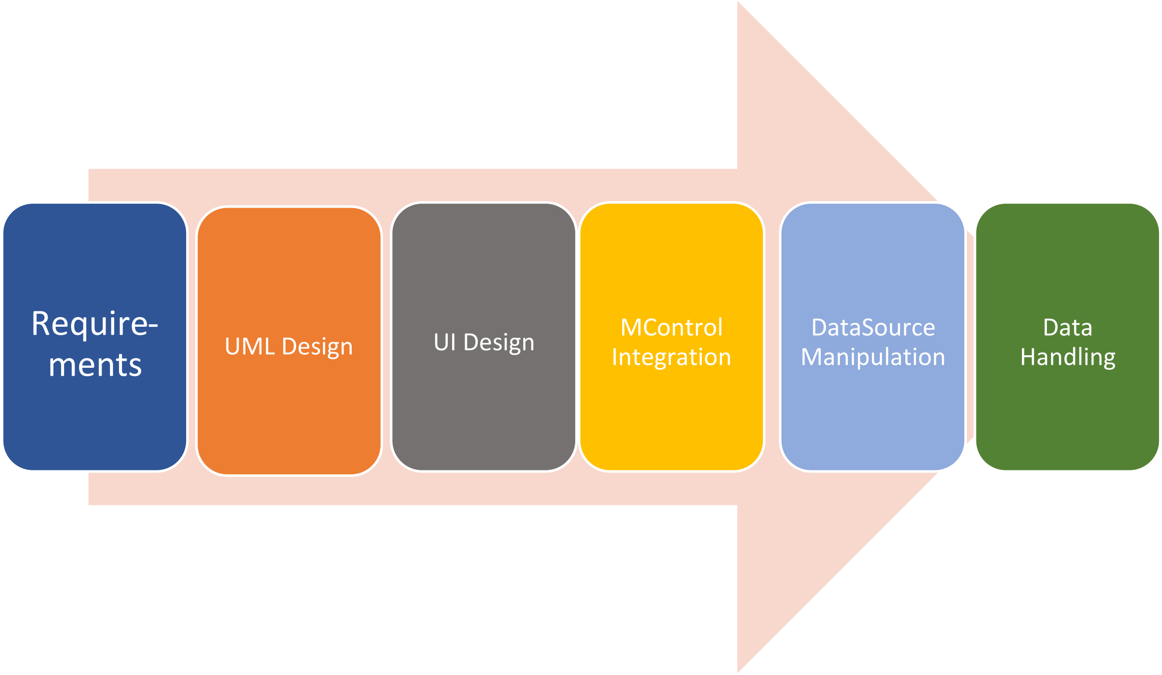
It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

It is usually beginning with a context diagram as level 0 of the DFD diagram, a simple representation of the whole system. To elaborate further from that, we drill down to a level 1 diagram with lower-level functions decomposed from the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required. Progression to levels 3, 4 and so on is possible but anything beyond level 3 is not very common. Please bear in mind that the level of detail for decomposing a particular function depending on the complexity that function.

CHAPTER 7

GANTT CHART





CHAPTER 8

FEASIBILITY STUDY

As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable.

A well-designed study should offer a historical background of the business or project, such as a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such studies precede technical development and project implementation.

8.1 Types of Feasibility Study

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. There are four types of feasibility study—separate areas that a feasibility study examines, described below.

8.1.1 **Economical Feasibility**

System is economical feasible and can be easily implement with minimum hardware and software resources as this is a cloud based application platform is provided by cloud provider only there is a need of Internet connection and a Browser application. It is very important for designer to first analyze the system economically and determines that project is economical feasible or not.

interdependent. Although the systems analyst is trying to propose a system that fulfills various information requirements, decisions to continue with the proposed system will be based on a cost-benefit analysis, not on information requirements. In many ways, benefits are measured by costs, as becomes apparent in the next section.

Systems analysts are required to predict certain key variables before the proposal is submitted to the client. To some degree, a systems analyst will rely on a what-if analysis, such as, "What if labor costs rise only 5 percent per year for the next three years, rather than 10 percent?" The systems analyst should realize, however, that he or she cannot rely on what-if analysis for everything if the proposal is to be credible, meaningful, and valuable.

The systems analyst has many forecasting models available. The main condition for choosing a model is the availability of historical data. If they are unavailable, the analyst must turn to one of the judgment methods: estimates from the sales force, surveys to estimate customer demand, Delphi studies (a consensus forecast developed independently by a group of experts through a series of iterations), creating scenarios, or drawing historical analogies.

If historical data are available, the next differentiation between classes of techniques involves whether the forecast is conditional or unconditional. Conditional implies that there is an association among variables in the model or that such a causal relationship exists. Common methods in this group include correlation, regression, leading indicators, econometrics, and input/output models.

Unconditional forecasting means the analyst isn’t required to find or identify any causal relationships. Consequently, systems analysts find that these methods are low-cost, easyto-implement alternatives. Included in this group are graphical judgment, moving averages, and analysis of time-series data. Because these methods are simple, reliable, and cost effective, the remainder of the section focuses on them.

- **Estimation of Trends**

Trends can be estimated in a number of different ways. One way to estimate trends is to use a moving average. This method is useful because some seasonal, cyclical, or random patterns may be smoothed, leaving the trend pattern. The principle behind moving averages is to calculate the arithmetic mean of data from a fixed number of periods; a three-month moving average is simply the average of the last three months. For example, the average sales for January, February, and March is used to predict the sales for April. Then the average sales for February, March, and April are used to predict the sales for May, and so on.

When the results are graphed, it is easily noticeable that the widely fluctuating data are smoothed. The moving average method is useful for its smoothing ability, but at the same time it has many disadvantages. Moving averages are more strongly affected by extreme values than by using graphical judgment or estimating using other methods such as least squares. The analyst should learn forecasting well, as it often provides information valuable in justifying the entire project.

- **Tangible Costs**

The concepts of tangible and intangible costs present a conceptual parallel to the tangible and intangible benefits discussed already. Tangible costs are those that can be accurately projected by the systems analyst and the business’s accounting personnel.

Included in tangible costs are the cost of equipment such as computers and terminals, the cost of resources, the cost of systems analysts' time, the cost of programmers' time, and other employees salaries. These costs are usually well established or can be discovered quite easily, and are the costs that will require a cash outlay of the business.

- **Intangible Costs**

Intangible costs are difficult to estimate and may not be known. They include losing a competitive edge, losing the reputation for being first with an innovation or the leader in a field, declining company image due to increased customer dissatisfaction, and ineffective decision making due to untimely or inaccessible information. As you can imagine, it is next to impossible to project a dollar amount for intangible costs accurately. To aid decision makers who want to weigh the proposed system and all its implications, you must include intangible costs even though they are not quantifiable.

8.1.2. Technical Feasibility

It is the study of the function performance and constraints that may affect the ability to achieve an acceptable system. The project development requires designer to have technical knowledge of salesforce.com for both application development and database system. Technical feasibility is one of the most important criteria for selecting material for digitisation. The physical characteristics of source material and the project goals for capturing, presenting and storing the digital surrogates dictate the technical requirements. Libraries must evaluate those requirements for each project and determine whether they can be met with the resources available. If the existing staff, hardware and software resources cannot meet the requirements, then the project will need funding to upgrade equipment or hire an outside conversion agency. If these resources are not available, or if the technology does not exist to meet the requirements, then it is not technically feasible to digitise that material.

Considerations for technical feasibility include:

Image capture: Image capture requires equipment, such as a scanner or a digital camera. Different types of material require different equipment, and different equipment produces images of differing quality. When selecting materials for digitising, technical questions that need to be addressed include: does the original source material require high resolution to capture? Are there any oversized items in the collection? Are there any bound volumes in the collection? What critical features of the source material must be captured in the digital product? In what condition are the source materials? Will they be damaged by the digitisation process?

Presentation*:* Presentation refers to how the digitised materials will be displayed online. Consider the following questions to determine the technical feasibility of presenting the digitised material:

Will the materials display well digitally?

How will users use the digital versions?

How will users navigate within and among digital collections?

Do the institutionally supported platforms and networked environment have the capability for accessing the images and delivering them with reasonable speed to the target audience?

Do the images need to be restricted to a specified community?

Do the images need special display features such as zooming, panning and page turning?

Description: Some archival and special collections have been catalogued for public use and contain detailed finding aids with descriptions about each item and the collection as a whole. Other collections may not have been reviewed and documented in detail and do not have much information on individual items. Those collections will require more time, human resources and significant additional expense to research the materials, check the accuracy of the information obtained, and write appropriate descriptions to aid in discovery and use of the digital items. Typewritten documents, like the Drew Pearson columns described above, can have reasonably accurate OCR applied to them to replace, for some uses, the detailed descriptions required for discovery of hand-written or picture materials. The selection criteria should clearly state whether the items and collections that do not contain descriptions should be considered for digitisation.

Human resources: When selecting materials for digitisation, the library should consider whether it has the staff and skill sets to support the digitisation, metadata entry, user interface design, programming and search engine configuration that is required for the project to implement the desired functionality. For large collaborative projects, dedicated staff are usually required from each partner. Digital collections also require long-term maintenance, which needs to be considered and planned for. If a project does not have the necessary staff and skills in-house, but funding is available, outsourcing may be a good choice.

8.1.3. Behavioural Feasibility

In the application domain our system works as an application. There are simple form to fill and service requires no ambiguous entries, all the behavioural entries are simple and GUI based. The system design is very user friendly, interactive. The application should be used by Administrator. People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. [t is common knowledge that computer installations have something to do with turnover, transfers, retraining, and changes in employee job status. Therefore, it is understandable that the introduction of a candidate system requires special effort to educate, sell, and train the staff on new ways of conducting business.

In our safe deposit example, three employees are more than 50 years old and have been with the bank over 14 years, four years of which have been in safe deposit. The remaining two employees are in their early thirties. They joined safe deposit about two years before the study. Based on data gathered from extensive interviews, the younger employees want the programmable aspects of safe deposit (essentially billing) put on a computer. Two of the three older employees have voiced resistance to the idea. Their view is that billing is no problem. The main emphasis is customer service-personal contacts with customers. The decision in this case was to go ahead and pursue the project.

8.1.4 Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.[10]

The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases

An example of an operational feasibility study, or the fourth type, analyzes the inside operations on how a deemed process will work, be implemented, and how to deal with change resistance and acceptance.

Operational feasibility studies are generally utilized to answer the following questions:

- **Process** – How do the end-users feel about a new process that may be implemented?
- **In-House Strategies** – How will the work environment be affected? How much will it change?
- **Adapt & Review** – Once change resistance is overcome, explain how the new process will be implemented along with a review process to monitor the process change.

If an operational feasibility study must answer the six items above, how is it used in the real world? A good example might be if a company has determined that it needs to totally redesign the workspace environment.

After analyzing the technical, economic, and scheduling feasibility studies, next would come the operational analysis. In order to determine if the redesign of the workspace environment would work, an example of an operational feasibility study would follow this path based on six elements:

- **Process** – Input and analysis from everyone the new redesign will affect along with a data matrix on ideas and suggestions from the original plans.
- **Evaluation** – Determinations from the process suggestions; will the redesign benefit everyone? Who is left behind? Who feels threatened?
- **Implementation** – Identify resources both inside and out that will work on the redesign. How will the redesign construction interfere with current work?
- **Resistance** – What areas and individuals will be most resistant? Develop a change resistance plan.
- **Strategies** – How will the organization deal with the changed workspace environment? Do new processes or structures need to be reviewed or implemented in order for the redesign to be effective?
- **Adapt & Review** – How much time does the organization need to adapt to the new redesign? How will it be reviewed and monitored? What will happen if through a monitoring process, additional changes must be made?

The most important part of operational feasibility study is input—from everyone, especially when it affects how or what an organization does as far as processes. If the process were to build a new sports arena for a client, then a study determining how the arena will operate in a way that is conducive to its inhabitants, parking, human flow, accessibility and other elements is a good example of an operational feasibility study.

CHAPTER 9

Test Case

9.1 BLACK BOX TESTING :-

The technique of testing without having any knowledge of the interior workings of the application is called blackbox testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a blackbox test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

9.2 WHITE BOX TESTING:-

Whitebox testing is the detailed investigation of internal logic and structure of the code. Whitebox testing is also called glass testing or openbox testing. In order to perform whitebox testing on an application, a tester needs to know the internal workings of the code.

9.3 GREY BOX TESTING:-

Greybox testing is a technique to test the application with having a limited knowledge of the internal workings of an application. In software testing, the phrase the more you know, the better carries a lot of weight while testing an application.

9.4 UNIT TESTING:-

Unit Testing contains the testing of each unit of Recruitment Application. We have tested each interface by input values and check whether it is working properly working or not we also tested database connectivity. We have entered value in interface and check that the values are properly goes to corresponding tuples or not.

9.5 INTEGRATION TESTING:-

Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. Integration testing can be done in two ways: Bottomup integration testing and Topdown integration testing.

9.6 SYSTEM TESTING:-

System testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified Quality Standards. This type of testing is performed by a specialized testing team.

CHAPTER 10

CONCLUSION

The package was designed in such a way that future modifications can be done easily.

Automation of the entire system improves the efficiency.

It provides a friendly graphical user interface which proves to be better when compared to the existing system.

It gives appropriate access to the authorized users depending on their permissions.

A software project means a lot of experience. We learned a lot through this project. This project has sharpened our concept game engine, animation, and the software-hardware interface. We learned a lot about different documentation. Now I have much wider knowledge of the features Java offers and put into practice various object-oriented methods that learnt last semester.

CHAPTER 11

LITERATURE REVIEW

Abstract

Microsoft.NET Framework 4 security architecture provides additional capabilities for handling and managing code as part of application security. It provides ways to control the code behavior by enforcing stringent security rules before executing the code. In.NET 4, Code Access Security was simplified. The applications access rights are defined using permissions and their transparency.[1] Code access permission, Identity permissions, and Role-based permission are the various permission types available in.NET 4. Security Transparency separates code into "safe," "unsafe," and "maybe safe" categories. Microsoft.NET 4 Framework advances the capabilities of Allowing Partial Trusted Callers (APTCA) by providing control of permissions to hosts instead of a machinewide policy This chapter demonstrates why we need model-based testing, with a small but complete working example. We exhibit a software defect that is not detected by typical

unit tests, but is only exposed by executing more realistic scenarios that resemble actual application program runs [2].
It also looks at building a basic understanding of new additions in.NET Framework 4, such as Dynamic Language Runtime (DLR) and.NET-supported dynamic languages. DLR provides a common framework, a certain set of services, and a runtime execution environment for dynamic languages to function. It provides a pluggable engine to onboard dynamic language compilers through respective binders. DLR is built on top of CLR

11.1 Introduction

The term MEAN stack refers to a collection of JavaScript based technologies used to develop web applications. MEAN is an acronym for MongoDB, ExpressJS, AngularJS and Node.js. From client to server to database, MEAN is full stack JavaScript. This article explores the basics of the MEAN stack and shows how to create a simple bucket list application.

- Node.js is a server side JavaScript execution environment. It’s a platform built on Google Chrome’s V8 JavaScript runtime. It helps in building highly scalable and concurrent applications rapidly.
- Express is lightweight framework used to build web applications in Node. It provides a number of robust features for building single and multi page web application. Express is inspired by the popular Ruby framework, Sinatra.
- MongoDB is a schemaless NoSQL database system. MongoDB saves data in binary JSON format which makes it easier to pass data between client and server.
- AngularJS is a JavaScript framework developed by Google. It provides some awesome features like the two-way data binding. It’s a complete solution for rapid and awesome front end development.

11.2 Related Work

Software tools are a great aid to process engineers, but too much dependence on such tools can often lead to inappropriate and suboptimal designs. Reliance on software is also a hindrance without a firm understanding of the principles underlying its operation, since users are still responsible for devising the design [6]

Times, when programmers sat in dark cellars and tried to solve all problems on their own are over once and for all. In the meantime software engineering has become a very knowledge-intensive [5] and communicative process (not only but also triggered by agile methods for software development) where the actors heavily exchange data (see Google-Code), connect with like-minded (see Google Summer of Code), blog about experiences in their own weblogs, provide code snippets free of charge (see Django-Snippets) or help novices with words and deeds in large mailing lists. is social software engineering-the creation of software and related artifacts within a social network-gained a lot of attention in recent software engineering research [7]

11.3 Design(GUI, analysis, implementatiom, evaluation)

The design method that was chosen toward the current Visual C++ demonstration application development consisted of the following steps

1. Background research on the type of application to be developed 2. Requirements of the intended software application concerning scope and function 3. General use case narratives describing the general workings of the target application 4. Use case scenarios and conversations involving user actions and system responsibilities 5. A noun and verb analysis of the key items and actions concerning the application scope 6. Object analysis involving candidate, responsibilities, and collaborators (CRC) cards 7. Preliminary diagrams reflecting the initial class structure 8. Basic header and source implementing the intentionally bare class structure.

Once the key classes and objects are deigned, a hierarchical class diagram may be drawn to succinctly and visually present the key classes and their association relationships. A preliminary program involving a basic set of source and header Iles organizing and implementing the class structure may then be written, e.g., as a Win32 Console Application, to express the initial set of ideas about how the program should function. This includes basic member methods, variables, and data structures to manage the ¦ow of information throughout the program: this is intentionally very brief and concerns only the developer’s initial structure, rather than any additional systemprovided structure and is designed to be exploratory in nature.[8]

The next step involves the implementation of an initial graphical user interface to display typical Window-based application features, such as child windows, menus with entries, and toolbars with buttons. The Microsoft Visual C++® 6.0, Microsoft Visual Studio Development System (integrated development environment [IDE]), allows the developer to easily set up the interactive features of the application and the key topics covered here are the Application Wizard, menus, icons, toolbars, dialog windows, and attaching functionality to the menu entries and toolbar buttons.

Blocks may be placed on the palette, and connections may be drawn to connect the blocks together. However, before a simulation can be initiated, block parameters need to be set using a dialog window, and the values of these variables need to be updated to the underlying code. Double-left-clicking on any of the blocks should invoke the block dialog window for that particular block and clicking the OK button should update the entered data to the program variables. The following step-by-step instructions indicate how functionality can be added to the code to allow block parameter values to be assigned to a block via a dialog window invoked upon doubleleft-clicking a block. Many different types of items may be moved together by its circumscribing them with what is called a “rubber band” and then moving the whole rubber-band-enclosed group together. This is performed using a CRectTracker object and determining whether the enclosed region of the rubber band intersects or contains items on the palette, e.g., blocks, connection bend points, and connection end points, and then updating the positions of the items to be translated by the same amount as the center point of the whole rectangular rubber band region.

Going where no book on software measurement and metrics has previously gone, this critique thoroughly examines a number of bad measurement practices, hazardous metrics, and huge gaps and omissions in the software literature that neglect important topics in measurement. The book covers the major gaps and omissions that need to be filled if data about software development is to be useful for comparisons or estimating future projects.

Among the more serious gaps are leaks in reporting about software development efforts that, if not corrected, can distort data and make benchmarks almost useless and possibly even harmful. One of the most common leaks is that of unpaid overtime. Software is a very labor-intensive occupation, and many practitioners work very long hours. However, few companies actually record unpaid overtime. This means that software effort is underreported by around 15%, which is too large a value to ignore. Other sources of leaks include the work of part-time specialists who come and go as needed. There are dozens of these specialists, and their combined effort can top 45% of total software effort on large projects.[9]

The book helps software project managers and developers uncover errors in measurements so they can develop meaningful benchmarks to estimate software development efforts. It examines variations in a number of areas that include:

- Programming languages
- Development methodology
- Software reuse

- Functional and nonfunctional requirements
- Industry type
- Team size and experience

11.4 Results

The programmer-centric book is written in a way that enables even novice practitioners to grasp the development process as a whole.

Incorporating real code fragments and explicit, real-world open-source operating system references (in particular, FreeRTOS) throughout[10], the text:

- Defines the role and purpose of embedded systems, describing their internal structure and interfacing with software development tools
- Examines the inner workings of the GNU compiler collection (GCC)-based software development system or, in other words, toolchain.

The chapter looks at some of the software development paradigms in the world of computer science, including both traditional and advanced methodologies. The traditional software engineering paradigm, based on the classic life cycle, includes the waterfall model of development, incremental development, rapid application development, prototype development (throwaway and exploratory models) and spiral development. Advanced development paradigms include agile development, component-based development, aspect-oriented and cleanroom software development.[11] The major findings and critiques of all these paradigms are discussed in detail.

In particular, this chapter provides an introduction to component-based software engineering, together with the four phases of its evolution, and discusses its characteristics

11.5 Discussion

Starting from a basic definition of models, which refer to simpler mappings of relevant attributes of the real world with the intention to reduce the complexity of the real world with respect to modeling objectives, process models can be understood as a homomorphous, time-based mapping of a real-world system focusing a sequence-based, plausible visualization. According to Krallmann et al. [12], a system to be modeled consists of an amount of system elements, that are connected with an amount of system relations. As it is limited by a system border, the system environment and the system are connected with an interface to exchange system input and system output.

Failure of safety critical applications might lead to serious consequences such as significant financial loss or even loss of life.[13] Thus, software quality assurance, also simply called software assurance, has become the focus when certifying a safety critical system. Software assurance includes reliability, security, robustness, safety, and other quality-related attributes, as well as functionality and performance.

11.6 Conclusion

Knowledge production within the field of business research is accelerating at a tremendous speed while at the same time remaining fragmented and interdisciplinary. This makes it hard to keep up with state-of-the-art and to be at the forefront of research, as well as to assess the collective evidence in a particular area of business research. This is why the literature review as a research method is more relevant than ever. Traditional literature reviews often lack thoroughness and rigor and are conducted ad hoc, rather than following a specific methodology. Therefore, questions can be raised about the quality and trustworthiness of these types of reviews.[14] This paper discusses literature review as a methodology for conducting research and offers an overview of different types of reviews, as well as some guidelines to how to both conduct and evaluate a literature review paper. It also discusses common pitfalls and how to get literature reviews published. the process of digitalisation, and accepted standards in these fields are essential for building and exploiting complex computing, communication, multimedia and measuring systems.[15] Standards can simplify the design and construction of individual hardware and software components and help to ensure satisfactory interworking.

11.7 References

[1] .NET 4 for Enterprise Architects and Developers

[2] Model-Based Software Testing and Analysis with C#

[3] PROGRAMMING IN C#

[4] Data Structures and Algorithms Using Visual Basic.NET

[5] Object-Oriented Programming with Visual Basic.NET

[6] Process Engineering and Design Using Visual Basic

[7] Data Structure and Software Engineering

[8] Software Application Development

[9] A Guide to Selecting Software Measures and Metrics

[10] Embedded Software Development

[11] Component-Based Software Engineering

[12] Business Modeling and Software Design

[13] Dependable Software Engineering. Theories, Tools, and Applications

[14] Computer Science Review

[15] Computer Standard & Interfaces