Functional Requirement Specification

Corporate Relationship Management

(CRM)

KIIT University

Review and Approvals

**Prepared By**

|  |  |  |
| --- | --- | --- |
|  | **Date:** |  |
|  | | |

**Reviewed By**

|  |  |  |
| --- | --- | --- |
|  | **Date:** |  |
|  | | |

**Approved By**

|  |  |  |
| --- | --- | --- |
|  | **Date:** |  |
|  | | |
|  | **Date:** |  |
|  | | |
|  | **Date:** |  |
|  | | |

Table of Contents

[1 Introduction 7](#_Toc399431883)

[1.1 Purpose 7](#_Toc399431884)

[1.2 Prerequisite 7](#_Toc399431885)

[1.3 Scope 7](#_Toc399431886)

[1.4 References 7](#_Toc399431887)

[2 Definitions 8](#_Toc399431888)

[3 System Specifications 9](#_Toc399431889)

[3.1 Menu and Menu Items 9](#_Toc399431890)

[3.2 User Logon 11](#_Toc399431891)

[3.3 Page Structure 11](#_Toc399431892)

[4 Functional Requirements 12](#_Toc399431893)

[4.1 Administrator 12](#_Toc399431894)

[4.1.1 Manage User 12](#_Toc399431895)

[4.1.2 Manage Company Type 13](#_Toc399431896)

[4.1.3 Manage Company 14](#_Toc399431897)

[4.1.4 Manage Office Type 15](#_Toc399431898)

[4.1.5 Manage Department 17](#_Toc399431899)

[4.1.6 Manage Stream 18](#_Toc399431900)

[4.1.7 Manage District 19](#_Toc399431901)

[4.1.8 Manage Travel Destination 20](#_Toc399431902)

[4.1.9 Managing Salutation 21](#_Toc399431903)

[4.1.10 Manage Contact Status 23](#_Toc399431904)

[4.1.11 Manage Contact Relationship 24](#_Toc399431905)

[4.1.12 Manage Contact Visibility 25](#_Toc399431906)

[4.1.13 Manage Contact Swapping 26](#_Toc399431907)

[4.2 Contacts 27](#_Toc399431908)

[4.2.1 Add Contact 27](#_Toc399431909)

[4.2.2 View Contact 27](#_Toc399431910)

[4.2.3 Edit Contact 29](#_Toc399431911)

[4.2.4 Delete Contact 29](#_Toc399431912)

[4.2.5 Add Remainder 30](#_Toc399431913)

[4.2.6 View all Remainders 30](#_Toc399431914)

[4.2.7 Edit Remainder 31](#_Toc399431915)

[4.2.8 Add an Activity 31](#_Toc399431916)

[4.2.9 View all Activities 32](#_Toc399431917)

[4.2.10 Edit Activity 32](#_Toc399431918)

[4.2.11 Contact Search 33](#_Toc399431919)

[4.2.12 Contact Edit Status 34](#_Toc399431920)

[4.3 Remainder List 34](#_Toc399431921)

[4.3.1 Add Remainder 34](#_Toc399431922)

[4.3.2 View Remainder 35](#_Toc399431923)

[4.3.3 Close Remainder 36](#_Toc399431924)

[4.4 Calendar 37](#_Toc399431925)

[4.4.1 Calendar 37](#_Toc399431926)

[4.4.2 Travel Plan 37](#_Toc399431927)

[4.5 Reports 38](#_Toc399431928)

[4.5.1 Details by Office 38](#_Toc399431929)

[4.5.2 Activities by Company 39](#_Toc399431930)

[4.5.3 Activities by Contact 40](#_Toc399431931)

[4.5.4 Activities by Member 41](#_Toc399431932)

[4.5.5 Scheduled Items 42](#_Toc399431933)

[4.5.6 Probable Visits 42](#_Toc399431934)

[4.5.7 Employee Performance 43](#_Toc399431935)

[4.5.8 Travel Report 44](#_Toc399431936)

[4.5.9 Contact Report 45](#_Toc399431937)

[5 System Requirements 48](#_Toc399431938)

[5.1.1 Hardware 48](#_Toc399431939)

[5.1.2 Software 48](#_Toc399431940)

[5.1.3 Documentation 48](#_Toc399431941)

[5.1.4 User Capacity 48](#_Toc399431942)

[5.1.5 Location Requirement 48](#_Toc399431943)

[6 Operational / Maintenance Support 48](#_Toc399431944)

[7 Exceptions 48](#_Toc399431945)

[8 Acceptance Criteria 49](#_Toc399431946)

[9 Constraints 49](#_Toc399431947)

[10 Assumptions 49](#_Toc399431948)

[11 Dependencies 49](#_Toc399431949)

[12 Risks 49](#_Toc399431950)

[13 References 50](#_Toc399431951)

[14 Appendix 51](#_Toc399431952)

[14.1 Appendix 1: User Logon 51](#_Toc399431953)

[14.2 Appendix 2: Welcome Screen and Page Structure 51](#_Toc399431954)

[14.3 Appendix - User 52](#_Toc399431955)

[14.4 Appendix - Company Type 52](#_Toc399431956)

[14.5 Appendix - Company 52](#_Toc399431957)

[14.6 Appendix - Office Type 53](#_Toc399431958)

[14.7 Appendix - Department 53](#_Toc399431959)

[14.8 Appendix - Stream 53](#_Toc399431960)

[14.9 Appendix - District 53](#_Toc399431961)

[14.10 Appendix - Travel Destination 54](#_Toc399431962)

[14.11 Appendix - Salutation 54](#_Toc399431963)

[14.12 Appendix - Contact Status 54](#_Toc399431964)

[14.13 Appendix - Contact Relationship 54](#_Toc399431965)

[14.14 Appendix - Swapping Contact 54](#_Toc399431966)

[14.15 Appendix - Contact 55](#_Toc399431967)

[14.16 Appendix - Remainder 56](#_Toc399431968)

[14.17 Appendix – View Remainder 56](#_Toc399431969)

[14.18 Appendix – Close Remainder 56](#_Toc399431970)

[14.19 Appendix - Activity 57](#_Toc399431971)

[14.20 Appendix – Travel Plan 57](#_Toc399431972)

# Introduction

<Introduction about KIIT CRM>

**Business Need:** KIIT University provides placement assistance to its students. To facilitate the placement activity, the university invites various corporate companies to conduct campus drive in different schools (of different streams). The corporate companies are represented by a person or a group of person

The KIIT University requires an online and integrated solution to keep track of various corporate companies visiting the campus for recruitment. For this purpose, the employees of corporate relationship (CR) cell of the university (henceforth regarded as users of the application) keep on trying to contact the companies and establish relationship. The relationship with the corporate company is established via a contact person (henceforth regarded as contacts of the application). The users do telephonic calls, email transactions, personal meeting with the contact person by travelling to the office destination. In doing these activities users need to set remainders for themselves and also for the contacts to respond back. This leads to the need of an application system which shall keep track of the activities that the users do in order to maintain relationship with the corporate bodies.

So the system must provide and easy interface to manage the corporate contact details, employees of the corporate relationship cell, employee travel and call details, as well as reports as per the need of the management.

## Purpose

This document is the Functional Requirement Specification for KIIT CRM Application ([www.kiitcrm.in](http://www.kiitcrm.in)) and explains the different functionalities of the system.

## Prerequisite

The User Requirements Specification (URS) must be approved prior to the approval of this document and must act as input to the Functional Specification.

## Scope

This Functional Requirement Specification (FRS) will define the following functionality (as mentioned in URS) in detail.

# Definitions

|  |  |
| --- | --- |
| **Definition** | **Description** |
| User | The normal user of the application – the corporate relationship cell members of KIIT University |
| Admin | The admin user deals with the static data maintained in the application. Addition to these, the admin user shall act as the approval authority for editing requests (if any) submitted by the users of the application. |
| Company | The corporate houses interested to visit the KIIT university for campus recruitment  Example – Infosys, TCS, Tech Mahindra, Times of India etc |
| Company Type | The type of company depending upon the functional area of  Example – Banking, IT Services, Retails etc |
| Office Type | The type of office  Example – Branch Office, Head Office, Regional Office etc |
| Department | The different functional departments in the company  Example – Finance, HR, Marketing, IT, Purchase etc. |
| District | The different city / district where the corporate office is located  Example – Delhi (New Delhi, India), Bangalore (Karnataka, India) etc |
| Contact | The corporate person who liaisons with the corporate relationship cell member for campus recruitment drive |
| Salutation | The salutations used before the name of the contact  Example – Mr., Mrs., Dr., Colonel etc. |
| Contact Status | The corporate relationship cell member contacts the corporate head via an activity (telephone, email, meeting etc.) The result of the conversation is recoded as status of conversation with contact  Example - +ve (Positive), -ve (Negative), Visited etc. |
| Contact Relationship | The relationship that the corporate contact may have with KIIT University  Example – Alumni, Reference, Others etc. |
| Stream | The different education stream for which the company interested for conducting campus recruitment drive  Example – Bio Tech, B. Tech, Diploma, Law, MBA etc. |
| Travel Destination | The corporate relationship cell member might need to travel to the office location in order to have personal meetings with the corporate contacts.  Example – Bangalore, Chennai, Delhi, Hyderabad etc. |
| Activity | The type of correspondence made by the corporate relationship cell member with the corporate contact person.  Example – Telephonic conversation, personal meeting, Email exchange etc |
| Remainder | The alert of remanding the corporate relationship cell member about the scheduled activity. A remainder shall have a closing date and an opening date that should not precede either current date (e.g. today’s date) or the closing date.  A remainder shall be closed against an activity. |
| Calendar | The monthly calendar where the user shall able to see all the scheduled activities, remainders, and travel plans etc. The calendar shall have the navigation facility to browse the following and previous months in the calendar year. |
| Travel Plan | To record the travels made by the corporate relationship cell member. |

# System Specifications

The system is a single page web application, unlike other web application this needn’t to be refreshed to accomplish a task. The online system can only be accessed by the CR employees of KIIT University (henceforth regarded as user) with proper login. There are two types of users - administrator (admin user) and normal user (user). The admin user shall have the extra privilege access in addition to the normal user and shall be mentioned explicitly.

## Menu and Menu Items

The web application shall have menu items (buttons), on click event the sub-menu items (buttons) shall appear. Access to the menu items shall be privilege based. Functionalities shall be attached with the sub-menu items.

The top-level menu items and their associated sub-menu items are:

|  |  |  |
| --- | --- | --- |
| **Menu Items / Sub Menu Items** | **Access** | **Functional Description** |
| Administrator | Admin | To manage all static and independent data and edit approvals |
| User | Admin | To manage users of the application |
| Company Type | Admin | To manage a company type entity  Example - Banking, Retail, Advertising, BPO, IT etc. |
| Company | Admin | To manage company entity  Example – Infosys, TCS, Tech Mahindra, Airtel etc |
| Office Type | Admin | To manage office type entity  Example – Branch Office, Head Office, Regional Office etc |
| Department | Admin | To manage department entity  Example – Finance, HR, Marketing, IT, Purchase etc. |
| Stream | Admin | To manage education stream entity  Example – Bio Tech, B. Tech, Diploma, Law, MBA etc. |
| District | Admin | To manage address of contacts  Example – Delhi (New Delhi, India), Bangalore (Karnataka, India) etc |
| Travel Destination | Admin | To manage travel destination  Example – Bangalore, Chennai, Delhi, Hyderabad etc. |
| Salutation | Admin | To manage salutation entity.  Example – Mr., Mrs., Dr., Colonel etc. |
| Contact Status | Admin | To manage status of conversation with contact  Example - +ve (Positive), -ve (Negative), Visited etc. |
| Contact Relationship | Admin | To manage the contact relationship  Example – Alumni, Reference, Others etc. |
| Contact Visibility | Admin | To manage the visibility of contacts by the user  Example – user can able to view his own contacts (created by the user) or user can view all contacts irrespective of the user who has created the contact |
| Swapping Contact | Admin | To manage swapping of contact(s) from one user to another user  Example – to disassociate some or all contacts from user1 to user2 |
| Edit Approval | Admin | To approve the edit requests sent by normal user  Example – any modification done in the profile of a contact requires approval from admin user. |
| Manage Remainder | Admin | To manage remainders set by users  Example – to edit and / or cancel remainder(s) of a user |
| Contacts | User | To manage contact(s) |
| New | User | To create a new contact |
| View | User | To view contact(s) |
| Search | User | To search contact(s) |
| Edit Status | User | To view the status of contact(s) for which profile has been changed |
| Calendar | User | To manage travel plans and to view calendar entries |
| Calendar | User | To view and edit activities, remainders, travel plans in calendar.  The edit facility shall be prohibited to the owner i.e. the user shall be able to edit own entries only, not others. However admin shall be able to edit all entries. |
| Travel Plan | User | To manage travel plans |
| Remainder List | User | To manage remainders |
| New | User | To create new remainder – for self and for a contact |
| View | User | To manage remainders |
| Reports | User / Admin | To view and download reports in MS-XLS format |
| Details by Office | User | View office and their respective contact details |
| Activities by Company | User | View activity details with respect to a company |
| Activities by Contact | User | View activities of contact(s) with respect to a company |
| Activities by Member | User | View activity details of a user |
| Scheduled Items | User | View upcoming remainders with respect to company or contact |
| Probable Visits | User | View relationship status i.e. status of company visits |
| Contact Report | User | View details of contacts with respect to a company |
| Employee Performance | Admin | View user engagement with the application i.e. time spent by the user in different activities for corporate relationship |
| Travel Report | Admin | View travel report of a user |
| **Events / Notices** | **Admin** | **To manage event(s) notification to the users of the application** |
| View | Admin | View events and notices |
| New | Admin | To create new event and notice |
| Delete | Admin | To delete an event |

## User Logon

The system shall provide an interface to manage the users. The system shall have a logon screen (Appendix 1), where the user shall enter his/her logon credentials to logon to the system. Only authorized users (active users – see section 4.1.1) shall be allowed to access the application.

**Actions:**

1. The system shall display a welcome page (**Appendix 2**).
2. The system shall show **Welcome <user>** with a provision to logout from the application.
3. The user shall be able to edit his/her logon profile.
4. The logged in user shall be able to exit from the application via **Logout**.
5. The system shall display the top level menu items as per the access privilege.
6. The system shall display any event and (or) notice in a scrolling bar.
7. The system shall show the remainder list for the logged in user.

## Page Structure

The individual pages in the application shall have mainly two sections – header and body section. The header section and its contents shall be fixed.

|  |  |
| --- | --- |
| **Section** | **Description** |
| **Header Section - shall be horizontally split into three sections** | |
| Top Header | shall display the logged in user following a provision to edit the logon profile.  shall have a provision to log out from the application. |
| Menu Items | shall display all the menu items as per the access privilege. |
| Event Notification | shall display the events and (or) notices in a scrolling fashion. |
| **Body Section - shall be vertically split into two sections** | |
| Page Content | shall display the sub menu items and their associated functionality as per the actions  shall display the items affected subjected to actions in a tabular format (list), rows representing the records (items) and columns representing attributes of the item. |
| Functional Constraint | 1. The list shall be accompanied with navigation facility both at the top and bottom, if the number of rows exceeds a limit of ten (10). 2. The item list shall be sorted in ascending order (alphabetic) for text based column and descending order (most recent) for date specific column. 3. The last column of the item list shall contain links (or images) to facilitate actions e.g. edit, delete etc. for the particular row item. 4. The respective actions shall be activated on single mouse click event. 5. All the actions shall have the effect of refreshing the item list without explicitly refreshing the page. 6. All the date fields (editable) used in the application shall open a date picker on single mouse click event. |
| My Remainder List | shall display all types of remainder for the logged in user. |

# Functional Requirements

## Administrator

### Manage User

|  |  |
| --- | --- |
| **Navigation** | Administrator → Manage User |
| **Access Privilege** | Admin |
| **Actions** | Add, Edit, Delete, and Search |
| **Page Content** | The page shall display the add/edit form (Appendix – User) in the left side and the User Listin the right side.  The form in the left side shall have a submit button and a reset button at the end when opened in add mode.  The form in the left side shall have only one submit button at the end when opened in edit mode.  The page shall have a search text field above the User List for searching particular user**(s)** in the **User List** |
| **Functional Constraint** | 1. The form shall be opened in add mode by default. 2. The **User List** shall display all the existing **user(s)** in the system by default. 3. The search text field above the User List shall be enabled with auto-complete feature, i.e. as the user types in characters; the effect shall be propagated and visible in the User List. 4. The column headers of the **User List** shall be the following.    1. Sl. No. (#) – number starting from 1    2. User Name – name of the user    3. User ID – logon user id of the user    4. Password – logon password of the user    5. User Type – type of the user (user or admin)    6. DOB – birth date (dd-mm-yyyy) of the user    7. Email – email of the user    8. Mobile – mobile number of he user    9. Address – address of the user    10. Status – status of the user (Active or Inactive)    11. Actions - shall display the following images, which shall be activated by single mouse click event.        1. Edit – to edit a user        2. Delete – to delete a user 5. The User List shall be sorted in ascending order of user name column. 6. The add/edit form shall be populated with the information when Edit action is selected for the particular row item in the User List. 7. The system shall ask for confirmation for deleting a particular row item from the User List. |
| **Actions** | |
| **Add** | To add a user, the admin shall manually enter the information in the form shown in the left side of the page, which opens in add mode by default.  The reset button shall clear the manually entered information except the default ones.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The newly created user shall be displayed in the User List. |
| **Edit** | To edit a user, the admin shall mouse click on the edit image in the row item of the User List.  The user information from the row item shall be populated in the add/edit form.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The User List shall be refreshed with the updated user information. |
| **Delete** | To delete a user, the admin shall click on the delete image in the row item of the list, and the system shall ask for confirmation.  The action shall have no effect on cancellation.  The **user** shall be deleted from the database on confirmation.  The success / failure message during database operation shall be displayed at the top of the page.  The User List shall be refreshed. |
| **Search** | To search user**(s)**, the admin shall type in the search text field, and the effect shall be propagated to the User List. The typed in characters (pattern) shall be matched against the user name column of the User List and the list shall be refreshed to show only those user**(s)** that satisfy the matching criteria. |

### Manage Company Type

|  |  |
| --- | --- |
| **Navigation** | Administrator → Company Type |
| **Access Privilege** | Admin |
| **Actions** | Add, Edit, Delete, and Search |
| **Page Content** | The page shall display the add/edit form (**Appendix – Company Type**) in the left side and the **Company Type List**in the right side.  The form in the left side shall have a submit button and a reset button at the end when opened in add mode.  The form in the left side shall have only one submit button at the end when opened in edit mode.  The page shall have a search text field above the **Company Type List** for searching particular **Company Type(s)** in the **Company Type List**. |
| **Functional Constraint** | 1. The form shall be opened in add mode by default. 2. The **Company Type List** shall display all the existing **Company Type(s)** in the system by default. 3. The search text field above the **Company Type List** shall be enabled with auto-complete feature, i.e. as the user types in characters; the effect shall be propagated and visible in the **Company Type List**. 4. The column headers of the **Company Type List** shall be the following.    1. Sl. No. (#) – number starting from 1    2. Company Type – type of the company    3. Status – status of the company (Active or Inactive)    4. Edit Panel - shall display the following images, which shall be activated by single mouse click event.       1. Edit – to edit a company type       2. Delete – to delete a company type 5. The **Company Type List** shall be sorted in ascending order of **Company Type** column. 6. The add/edit form shall be populated with the information when Edit action is selected for the particular row item in the **Company Type List**. 7. The system shall ask for confirmation for deleting a particular row item from the **Company Type List**. |
| **Actions** | |
| **Add** | To add a **Company Type**, the admin shall manually enter the information in the form shown in the left side of the page, which opens in add mode by default.  The reset button shall clear the manually entered information except the default ones.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The newly created **Company Type** shall be displayed in the **Company Type** List. |
| **Edit** | To edit a **Company Type**, the admin shall mouse click on the edit image in the row item of the **Company Type** List.  The user information from the row item shall be populated in the add/edit form.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The **Company Type** List shall be refreshed with the updated **Company Type** information. |
| **Delete** | To delete a **Company Type**, the admin shall click on the delete image in the row item of the list, and the system shall ask for confirmation.  The action shall have no effect on cancellation.  The **Company Type** shall be deleted from the database on confirmation.  The success / failure message during database operation shall be displayed at the top of the page.  The **Company Type** List shall be refreshed. |
| **Search** | To search **Company Type (s)**, the admin shall type in the search text field, and the effect shall be propagated to the **Company Type** List. The typed in characters (pattern) shall be matched against the **Company Type** column of the **Company Type** List and the list shall be refreshed to show only those **Company Type (s)** that satisfy the matching criteria. |

### Manage Company

|  |  |
| --- | --- |
| **Navigation** | Administrator → Company |
| **Access Privilege** | Admin |
| **Actions** | Add, Edit, Delete, and Search |
| **Page Content** | The page shall display the add/edit form (**Appendix – Company**) in the left side and the **Company List**in the right side.  The form in the left side shall have a submit button and a reset button at the end when opened in add mode.  The form in the left side shall have only one submit button at the end when opened in edit mode.  The page shall have a search text field above the **Company List** for searching particular C***ompany*(s)** in the **Company List**. |
| **Functional Constraint** | 1. The form shall be opened in add mode by default. 2. The **Company List** shall display all the existing **Company(s)** in the system by default. 3. The search text field above the **Company List** shall be enabled with auto-complete feature, i.e. as the user types in characters; the effect shall be propagated and visible in the **Company List**. 4. The column headers of the **Company List** shall be the following.    1. Sl. No. (#) – number starting from 1    2. Company Name – name of the company    3. Company Type – type of the company    4. Company Website – website address of the company    5. Edit Panel - shall display the following images, which shall be activated by single mouse click event.       1. Edit – to edit a company       2. Delete – to delete a company 5. The **Company List** shall be sorted in ascending order of **Company Name** column. 6. The add/edit form shall be populated with the information when Edit action is selected for the particular row item in the **Company List**. 7. The system shall ask for confirmation for deleting a particular row item from the **Company List**. |
| **Actions** | |
| **Add** | To add a **Company**, the admin shall manually enter the information in the form shown in the left side of the page, which opens in add mode by default.  The reset button shall clear the manually entered information except the default ones.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The newly created **Company** shall be displayed in the **Company** List. |
| **Edit** | To edit a **Company**, the admin shall mouse click on the edit image in the row item of the **Company** List.  The user information from the row item shall be populated in the add/edit form.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The **Company** List shall be refreshed with the updated **Company** information. |
| **Delete** | To delete a **Company**, the admin shall click on the delete image in the row item of the list, and the system shall ask for confirmation.  The action shall have no effect on cancellation.  The **Company** shall be deleted from the database on confirmation.  The success / failure message during database operation shall be displayed at the top of the page.  The **Company** List shall be refreshed. |
| **Search** | To search **Company(s)**, the admin shall type in the search text field, and the effect shall be propagated to the **Company** List. The typed in characters (pattern) shall be matched against the **Company Name** column of the **Company** List and the list shall be refreshed to show only those **Company (s)** that satisfy the matching criteria. |

### Manage Office Type

|  |  |
| --- | --- |
| **Navigation** | Administrator → Office Type |
| **Access Privilege** | Admin |
| **Actions** | Add, Edit, Delete, and Search |
| **Page Content** | The page shall display the add/edit form (**Appendix – Office Type**) in the left side and the **Office Type List**in the right side.  The form in the left side shall have a submit button and a reset button at the end when opened in add mode.  The form in the left side shall have only one submit button at the end when opened in edit mode.  The page shall have a search text field above the **Office Type List** for searching particular **Office Type(s)** in the **Office Type List**. |
| **Functional Constraint** | 1. The form shall be opened in add mode by default. 2. The **Office Type List** shall display all the existing **Office Type(s)** in the system by default. 3. The search text field above the **Office Type List** shall be enabled with auto-complete feature, i.e. as the user types in characters; the effect shall be propagated and visible in the **Office Type List**. 4. The column headers of the **Office Type List** shall be the following.    1. Sl. No. (#) – number starting from 1    2. Office Type – type of the office    3. Status – status of the office type(Active or Inactive)    4. Edit Panel - shall display the following images, which shall be activated by single mouse click event.       1. Edit – to edit a office type       2. Delete – to delete a office type 5. The **Office Type List** shall be sorted in ascending order of **Office Type** column. 6. The add/edit form shall be populated with the information when Edit action is selected for the particular row item in the **Office Type List**. 7. The system shall ask for confirmation for deleting a particular row item from the **Office Type List**. |
| **Actions** | |
| **Add** | To add a **Office Type**, the admin shall manually enter the information in the form shown in the left side of the page, which opens in add mode by default.  The reset button shall clear the manually entered information except the default ones.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The newly created **Office Type** shall be displayed in the **Office Type** List. |
| **Edit** | To edit a **Office Type**, the admin shall mouse click on the edit image in the row item of the **Office Type** List.  The user information from the row item shall be populated in the add/edit form.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The **Office Type** List shall be refreshed with the updated **Office Type** information. |
| **Delete** | To delete a **Office Type**, the admin shall click on the delete image in the row item of the list, and the system shall ask for confirmation.  The action shall have no effect on cancellation.  The **Office Type** shall be deleted from the database on confirmation.  The success / failure message during database operation shall be displayed at the top of the page.  The **Office Type** List shall be refreshed. |
| **Search** | To search **Office Type (s)**, the admin shall type in the search text field, and the effect shall be propagated to the **Office Type** List. The typed in characters (pattern) shall be matched against the **Office Type** column of the **Office Type** List and the list shall be refreshed to show only those **Office Type (s)** that satisfy the matching criteria. |

### Manage Department

|  |  |
| --- | --- |
| **Navigation** | Administrator → Department |
| **Access Privilege** | Admin |
| **Actions** | Add, Edit, Delete, and Search |
| **Page Content** | The page shall display the add/edit form (**Appendix – Department**) in the left side and the **Department List**in the right side.  The form in the left side shall have a submit button and a reset button at the end when opened in add mode.  The form in the left side shall have only one submit button at the end when opened in edit mode.  The page shall have a search text field above the **Department List** for searching particular **Department(s)** in the **Department List**. |
| **Functional Constraint** | 1. The form shall be opened in add mode by default. 2. The **Department List** shall display all the existing **Department(s)** in the system by default. 3. The search text field above the **Department List** shall be enabled with auto-complete feature, i.e. as the user types in characters; the effect shall be propagated and visible in the **Department List**. 4. The column headers of the **Department List** shall be the following.    1. Sl. No. (#) – number starting from 1    2. Department Name – name of the department    3. Status – website address of the department    4. Edit Panel - shall display the following images, which shall be activated by single mouse click event.       1. Edit – to edit a department       2. Delete – to delete a department 5. The **Department List** shall be sorted in ascending order of **Department Name** column. 6. The add/edit form shall be populated with the information when Edit action is selected for the particular row item in the **Department List**. 7. The system shall ask for confirmation for deleting a particular row item from the **Department List**. |
| **Actions** | |
| **Add** | To add a **Department**, the admin shall manually enter the information in the form shown in the left side of the page, which opens in add mode by default.  The reset button shall clear the manually entered information except the default ones.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The newly created **Department** shall be displayed in the **Department** List. |
| **Edit** | To edit a **Department**, the admin shall mouse click on the edit image in the row item of the **Department** List.  The user information from the row item shall be populated in the add/edit form.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The **Department** List shall be refreshed with the updated **Department** information. |
| **Delete** | To delete a **Department**, the admin shall click on the delete image in the row item of the list, and the system shall ask for confirmation.  The action shall have no effect on cancellation.  The **Company** shall be deleted from the database on confirmation.  The success / failure message during database operation shall be displayed at the top of the page.  The **Department** List shall be refreshed. |
| **Search** | To search **Department (s)**, the admin shall type in the search text field, and the effect shall be propagated to the **Department** List. The typed in characters (pattern) shall be matched against the **Department Name** column of the **Department** List and the list shall be refreshed to show only those **Department (s)** that satisfy the matching criteria. |

### Manage Stream

|  |  |
| --- | --- |
| **Navigation** | Administrator → Stream |
| **Access Privilege** | Admin |
| **Actions** | Add, Edit, Delete, and Search |
| **Page Content** | The page shall display the add/edit form (**Appendix – Stream**) in the left side and the **Stream List**in the right side.  The form in the left side shall have a submit button and a reset button at the end when opened in add mode.  The form in the left side shall have only one submit button at the end when opened in edit mode.  The page shall have a search text field above the **Stream List** for searching particular **Stream(s)** in the **Stream List**. |
| **Functional Constraint** | 1. The form shall be opened in add mode by default. 2. The **Stream List** shall display all the existing **Stream(s)** in the system by default. 3. The search text field above the **Stream List** shall be enabled with auto-complete feature, i.e. as the user types in characters; the effect shall be propagated and visible in the **Stream List**. 4. The column headers of the **Stream List** shall be the following.    1. Sl. No. (#) – number starting from 1    2. Stream Name – name of the stream    3. Status – website address of the stream    4. Edit Panel - shall display the following images, which shall be activated by single mouse click event.       1. Edit – to edit a stream       2. Delete – to delete a stream 5. The **Stream List** shall be sorted in ascending order of **Stream Name** column. 6. The add/edit form shall be populated with the information when Edit action is selected for the particular row item in the **Stream List**. 7. The system shall ask for confirmation for deleting a particular row item from the **Stream List**. |
| **Actions** | |
| **Add** | To add a **Stream**, the admin shall manually enter the information in the form shown in the left side of the page, which opens in add mode by default.  The reset button shall clear the manually entered information except the default ones.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The newly created **Stream** shall be displayed in the **Stream** List. |
| **Edit** | To edit a **Stream**, the admin shall mouse click on the edit image in the row item of the **Stream** List.  The user information from the row item shall be populated in the add/edit form.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The **Stream** List shall be refreshed with the updated **Stream** information. |
| **Delete** | To delete a **Stream**, the admin shall click on the delete image in the row item of the list, and the system shall ask for confirmation.  The action shall have no effect on cancellation.  The **Stream** shall be deleted from the database on confirmation.  The success / failure message during database operation shall be displayed at the top of the page.  The **Stream** List shall be refreshed. |
| **Search** | To search **Stream (s)**, the admin shall type in the search text field, and the effect shall be propagated to the **Stream** List. The typed in characters (pattern) shall be matched against the **Stream Name** column of the **Stream** List and the list shall be refreshed to show only those **Stream(s)** that satisfy the matching criteria. |

### Manage District

|  |  |
| --- | --- |
| **Navigation** | Administrator → District |
| **Access Privilege** | Admin |
| **Actions** | Add, Edit, Delete, and Search |
| **Page Content** | The page shall display the add/edit form (**Appendix – District**) in the left side and the **District List**in the right side.  The form in the left side shall have a submit button and a reset button at the end when opened in add mode.  The form in the left side shall have only one submit button at the end when opened in edit mode.  The page shall have a search text field above the **District List** for searching particular **District(s)** in the **District List**. |
| **Functional Constraint** | 1. The form shall be opened in add mode by default. 2. The **District List** shall display all the existing **District(s)** in the system by default. 3. The search text field above the **District List** shall be enabled with auto-complete feature, i.e. as the user types in characters; the effect shall be propagated and visible in the **District List**. 4. The column headers of the **District List** shall be the following.    1. Sl. No. (#) – number starting from 1    2. City – city of the district    3. State – state of the district    4. Country – country of the district    5. Edit Panel - shall display the following images, which shall be activated by single mouse click event.       1. Edit – to edit a district       2. Delete – to delete a district 5. The **Stream List** shall be sorted in ascending order of **City** column. 6. The add/edit form shall be populated with the information when Edit action is selected for the particular row item in the **District List**. 7. The system shall ask for confirmation for deleting a particular row item from the **District List**. |
| **Actions** | |
| **Add** | To add a **District**, the admin shall manually enter the information in the form shown in the left side of the page, which opens in add mode by default.  The reset button shall clear the manually entered information except the default ones.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The newly created **District** shall be displayed in the **District** List. |
| **Edit** | To edit a **District**, the admin shall mouse click on the edit image in the row item of the **District** List.  The user information from the row item shall be populated in the add/edit form.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The **District** List shall be refreshed with the updated **District** information. |
| **Delete** | To delete a **District**, the admin shall click on the delete image in the row item of the list, and the system shall ask for confirmation.  The action shall have no effect on cancellation.  The **District** shall be deleted from the database on confirmation.  The success / failure message during database operation shall be displayed at the top of the page.  The **District** List shall be refreshed. |
| **Search** | To search **District(s)**, the admin shall type in the search text field, and the effect shall be propagated to the **District** List. The typed in characters (pattern) shall be matched against the **City** column of the **District** List and the list shall be refreshed to show only those **District(s)** that satisfy the matching criteria. |

### Manage Travel Destination

|  |  |
| --- | --- |
| **Navigation** | Administrator → Travel Destination |
| **Access Privilege** | Admin |
| **Actions** | Add, Edit, Delete, and Search |
| **Page Content** | The page shall display the add/edit form (**Appendix – Travel Destination**) in the left side and the **Travel Destination List**in the right side.  The form in the left side shall have a submit button and a reset button at the end when opened in add mode.  The form in the left side shall have only one submit button at the end when opened in edit mode.  The page shall have a search text field above the **Travel Destination List** for searching particular **Travel Destination(s)** in the **Travel Destination List**. |
| **Functional Constraint** | 1. The form shall be opened in add mode by default. 2. The **Travel Destination List** shall display all the existing **District(s)** in the system by default. 3. The search text field above the **Travel Destination List** shall be enabled with auto-complete feature, i.e. as the user types in characters; the effect shall be propagated and visible in the **Travel Destination List**. 4. The column headers of the **Travel Destination List** shall be the following.    1. Sl. No. (#) – number starting from 1    2. Travel Destination – name of the Travel Destination    3. Status – status of the Travel Destination (Active or Inactive)    4. Edit Panel - shall display the following images, which shall be activated by single mouse click event.       1. Edit – to edit a Travel Destination       2. Delete – to delete a Travel Destination 5. The **Travel Destination List** shall be sorted in ascending order of **Travel Destination** column. 6. The add/edit form shall be populated with the information when Edit action is selected for the particular row item in the **Travel Destination List**. 7. The system shall ask for confirmation for deleting a particular row item from the **Travel Destination List**. |
| **Actions** | |
| **Add** | To add a **Travel Destination**, the admin shall manually enter the information in the form shown in the left side of the page, which opens in add mode by default.  The reset button shall clear the manually entered information except the default ones.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The newly created **Travel Destination**shall be displayed in the **Travel Destination**List. |
| **Edit** | To edit a **Travel Destination**, the admin shall mouse click on the edit image in the row item of the **Travel Destination**List.  The user information from the row item shall be populated in the add/edit form.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The **Travel Destination**List shall be refreshed with the updated **Travel Destination**information. |
| **Delete** | To delete a **Travel Destination**, the admin shall click on the delete image in the row item of the list, and the system shall ask for confirmation.  The action shall have no effect on cancellation.  The **Travel Destination**shall be deleted from the database on confirmation.  The success / failure message during database operation shall be displayed at the top of the page.  The **Travel Destination**List shall be refreshed. |
| **Search** | To search **Travel Destination(s)**, the admin shall type in the search text field, and the effect shall be propagated to the **Travel Destination**List. The typed in characters (pattern) shall be matched against the **Travel Destination**column of the **Travel Destination**List and the list shall be refreshed to show only those **Travel Destination(s)** that satisfy the matching criteria. |

### Managing Salutation

|  |  |
| --- | --- |
| **Navigation** | Administrator → Salutation |
| **Access Privilege** | Admin |
| **Actions** | Add, Edit, Delete, and Search |
| **Page Content** | The page shall display the add/edit form (**Appendix – Salutation**) in the left side and the **Salutation List**in the right side.  The form in the left side shall have a submit button and a reset button at the end when opened in add mode.  The form in the left side shall have only one submit button at the end when opened in edit mode.  The page shall have a search text field above the **Salutation List** for searching particular **Salutation (s)** in the **Salutation List**. |
| **Functional Constraint** | 1. The form shall be opened in add mode by default. 2. The **Salutation List** shall display all the existing **Salutation(s)** in the system by default. 3. The search text field above the **Salutation List** shall be enabled with auto-complete feature, i.e. as the user types in characters; the effect shall be propagated and visible in the **Salutation List**. 4. The column headers of the **Salutation List** shall be the following.    1. Sl. No. (#) – number starting from 1    2. Salutation– name of the Salutation    3. Status – status of the Salutation(Active or Inactive)    4. Edit Panel - shall display the following images, which shall be activated by single mouse click event.       1. Edit – to edit a Salutation       2. Delete – to delete a Salutation 5. The **Salutation List** shall be sorted in ascending order of **Salutation** column. 6. The add/edit form shall be populated with the information when Edit action is selected for the particular row item in the **Salutation List**. 7. The system shall ask for confirmation for deleting a particular row item from the **Salutation List**. |
| **Actions** | |
| **Add** | To add a **Salutation**, the admin shall manually enter the information in the form shown in the left side of the page, which opens in add mode by default.  The reset button shall clear the manually entered information except the default ones.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The newly created **Salutation** shall be displayed in the **Salutation** List. |
| **Edit** | To edit a **Salutation**, the admin shall mouse click on the edit image in the row item of the **Salutation** List.  The user information from the row item shall be populated in the add/edit form.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The **Salutation** List shall be refreshed with the updated **Salutation** information. |
| **Delete** | To delete a **Salutation**, the admin shall click on the delete image in the row item of the list, and the system shall ask for confirmation.  The action shall have no effect on cancellation.  The **Salutation** shall be deleted from the database on confirmation.  The success / failure message during database operation shall be displayed at the top of the page.  The **Salutation** List shall be refreshed. |
| **Search** | To search **Salutation(s)**, the admin shall type in the search text field, and the effect shall be propagated to the **Salutation** List. The typed in characters (pattern) shall be matched against the **Salutation** column of the **Salutation** List and the list shall be refreshed to show only those **Salutation(s)** that satisfy the matching criteria. |

### Manage Contact Status

|  |  |
| --- | --- |
| **Navigation** | Administrator → Contact Status |
| **Access Privilege** | Admin |
| **Actions** | Add, Edit, Delete, and Search |
| **Page Content** | The page shall display the add/edit form (**Appendix – Contact Status**) in the left side and the **Contact Status List**in the right side.  The form in the left side shall have a submit button and a reset button at the end when opened in add mode.  The form in the left side shall have only one submit button at the end when opened in edit mode.  The page shall have a search text field above the **Contact Status List** for searching particular **Contact Status(s)** in the **Contact Status List**. |
| **Functional Constraint** | 1. The form shall be opened in add mode by default. 2. The **Contact Status List** shall display all the existing **Contact Status(s)** in the system by default. 3. The search text field above the **Contact Status List** shall be enabled with auto-complete feature, i.e. as the user types in characters; the effect shall be propagated and visible in the **Contact Status List**. 4. The column headers of the **Contact Status List** shall be the following.    1. Sl. No. (#) – number starting from 1    2. Contact Status Type– type of the Contact Status    3. Status – status of the Contact Status (Active or Inactive)    4. Edit Panel - shall display the following images, which shall be activated by single mouse click event.       1. Edit – to edit a Contact Status       2. Delete – to delete a Contact Status 5. The **Contact Status List** shall be sorted in ascending order of **Contact Status Type** column. 6. The add/edit form shall be populated with the information when Edit action is selected for the particular row item in the **Contact Status List**. 7. The system shall ask for confirmation for deleting a particular row item from the **Contact Status List**. |
| **Actions** | |
| **Add** | To add a **Contact Status**, the admin shall manually enter the information in the form shown in the left side of the page, which opens in add mode by default.  The reset button shall clear the manually entered information except the default ones.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The newly created **Contact Status** shall be displayed in the **Contact Status** List. |
| **Edit** | To edit a **Contact Status**, the admin shall mouse click on the edit image in the row item of the **Contact Status** List.  The user information from the row item shall be populated in the add/edit form.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The **Contact Status** List shall be refreshed with the updated **Contact Status** information. |
| **Delete** | To delete a **Contact Status**, the admin shall click on the delete image in the row item of the list, and the system shall ask for confirmation.  The action shall have no effect on cancellation.  The **Contact Status** shall be deleted from the database on confirmation.  The success / failure message during database operation shall be displayed at the top of the page.  The **Contact Status** List shall be refreshed. |
| **Search** | To search **Contact Status (s)**, the admin shall type in the search text field, and the effect shall be propagated to the **Contact Status** List. The typed in characters (pattern) shall be matched against the **Contact Status Type**column of the **Contact Status** List and the list shall be refreshed to show only those **Contact Status(s)** that satisfy the matching criteria. |

### Manage Contact Relationship

|  |  |
| --- | --- |
| **Navigation** | Administrator → Contact Relationship |
| **Access Privilege** | Admin |
| **Actions** | Add, Edit, Delete, and Search |
| **Page Content** | The page shall display the add/edit form (**Appendix – Contact Relationship**) in the left side and the **Contact Relationship List**in the right side.  The form in the left side shall have a submit button and a reset button at the end when opened in add mode.  The form in the left side shall have only one submit button at the end when opened in edit mode.  The page shall have a search text field above the **Contact Relationship List** for searching particular **Contact Relationship(s)** in the **Contact Relationship List**. |
| **Functional Constraint** | 1. The form shall be opened in add mode by default. 2. The **Contact Status List** shall display all the existing **Contact Relationship(s)** in the system by default. 3. The search text field above the **Contact Relationship List** shall be enabled with auto-complete feature, i.e. as the user types in characters; the effect shall be propagated and visible in the **Contact Relationship List**. 4. The column headers of the **Contact Relationship List** shall be the following.    1. Sl. No. (#) – number starting from 1    2. Contact RelationshipType– type of the Contact Relationship    3. Status – status of the Contact Relationship(Active or Inactive)    4. Edit Panel - shall display the following images, which shall be activated by single mouse click event.       1. Edit – to edit a Contact Relationship       2. Delete – to delete a Contact Relationship 5. The **Contact Relationship List** shall be sorted in ascending order of **Contact Relationship Type** column. 6. The add/edit form shall be populated with the information when Edit action is selected for the particular row item in the **Contact Relationship List**. 7. The system shall ask for confirmation for deleting a particular row item from the **Contact Relationship List**. |
| **Actions** | |
| **Add** | To add a **Contact Relationship**, the admin shall manually enter the information in the form shown in the left side of the page, which opens in add mode by default.  The reset button shall clear the manually entered information except the default ones.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The newly created **Contact Relationship** shall be displayed in the **Contact Relationship** List. |
| **Edit** | To edit a **Contact Relationship**, the admin shall mouse click on the edit image in the row item of the **Contact Relationship** List.  The user information from the row item shall be populated in the add/edit form.  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The **Contact Relationship** List shall be refreshed with the updated **Contact Relationship** information. |
| **Delete** | To delete a **Contact Relationship**, the admin shall click on the delete image in the row item of the list, and the system shall ask for confirmation.  The action shall have no effect on cancellation.  The **Contact Relationship** shall be deleted from the database on confirmation.  The success / failure message during database operation shall be displayed at the top of the page.  The **Contact Relationship** List shall be refreshed. |
| **Search** | To search **Contact Relationship (s)**, the admin shall type in the search text field, and the effect shall be propagated to the **Contact Relationship** List. The typed in characters (pattern) shall be matched against the **Contact Relationship Type**column of the **Contact Relationship** List and the list shall be refreshed to show only those **Contact Relationship(s)** that satisfy the matching criteria. |

### Manage Contact Visibility

|  |  |
| --- | --- |
| **Navigation** | Administrator → Contact Visibility |
| **Access Privilege** | Admin |
| **Description** | To set the contact visibility by users of the application.  Example – user shall able to view his / her own contacts (created by the user) or user shall be able to view all the contacts irrespective of the user who has created the contact. |
| **Page Content** | The page shall display the single selection option so that the admin shall able to set the contact visibility.  The selection options shall be:   1. View only own contacts 2. View all contacts |
| **Functional Constraint** | The admin shall be able to select only one option of contact visibility.  The setting condition shall be valid throughout the application irrespective of the user.  The default selection option shall be - View only own contacts. |
| **Action** | The admin user shall set the option which shall affect the user-contact view throughout the application.  If option (1) is selected, then the logged in user shall be able to see only his / her own contacts and no other contacts created by any other user of the system.  If option (2) is selected, all users shall be able to see all the contacts irrespective of the respective owner of the contacts. |

### Manage Contact Swapping

|  |  |
| --- | --- |
| **Navigation** | Administrator → Swapping Contact |
| **Access Privilege** | Admin |
| **Description** | Swapping of the contact owner from one user to another user.  Example – the contacts (some or all) associated with user1 shall be transferred to user2 |
| **Page Content** | The page shall display a form (**Appendix – Contact Swapping**) in the left side and the **Contact** List in the right side, which is the action of the left side form submission.  The page shall have a search text field above the **Contact List** for searching particular **Contact(s)** in the **Contact List**. |
| **Functional Constraint** | 1. The form in the left side shall have two drop down lists – one for **From User** and the second one for **To User**. 2. The items in both the drop down list shall be in sorted in ascending order. 3. Both the drop down lists shall be enabled with single selection option. 4. The form shall have a submit button and a reset button at the end. 5. The **From User drop down list** shall be pre-populated with the existing users in the system, so that the logged in user shall be able to select only one item. By default no item shall be selected. 6. The **To User drop down list** shall be populated with the existing users in the system only when the admin selects an item (a user) from the **From User** **drop down list**. By default no item shall be selected. 7. The **To User drop down list** shall be in default state if the selection changes in the **From User drop down list**. 8. Both the drop down lists shall be in default state on form reset. 9. The **Contact List** shall be visible (and refreshed) only when the admin user shall select a valid item from the **From User drop down list**. 10. The **Contact List** shall display all the **contact(s)** available in the system which are added by the selected user from the **From User drop down list**. 11. The column headers of the **Contact List** shall be     1. Contact Name – name of the **contact**     2. Company Name – name of the **company** with which the contact is associated 12. The **Contact List** shall be sorted in ascending order of **Contact Name** column. 13. The **Contact List** shall have multiple selection option to select multiple row items. 14. The **Contact List** shall be accompanied with a search text field above of it. 15. The search text field above the **Contact List** shall be enabled with auto-complete feature, i.e. as the user types in characters; the effect shall be propagated and visible in the **Contact List**. |
| **Action** | To swap contacts from one user to another, the admin user shall select one valid user from the **From User drop down list** and one valid item from the **To User drop down list**.  The admin user shall also select the **contact(s)** from the **Contact List.** The admin user shall take the help of the search text field to minimize the list items or searching particular **contact** from the **Contact List**.  The form submission shall ask for confirming the swapping operation, the cancel action shall have no effect.  The selected **contact(s)** from the **Contact List** shall be de-associated from the user selected from the **From User drop down list** and shall be associated with the user selected from the **To User drop down list**  The data shall be updated in the database on successful form submission.  The success / failure message on form submission and database operation shall be displayed at the top of the page.  The **Contact List** shall be refreshed. |

## Contacts

### Add Contact

|  |  |
| --- | --- |
| **Navigation** | Contacts → New |
| **Access Privilege** | User |
| **Data Constraints** | 1. A ***contact*** shall be added to an existing company, which needs a company must have been added to the system prior to the contact. 2. A contact shall be associated with only one company. 3. A contact shall be associated with only one office type of the company. The office type shall have the option of selecting whether recruitment calls can be taken from this office or not 4. A contact shall be associated with only one department of the company. 5. The status of the new contact shall be active (by default). 6. A contact shall be associated with at least one stream. |
| **Page Content** | The page shall have a search text field to facilitate searching for an existing company to which the new contact shall be associated.  The page shall display a form where the user shall enter the detail information for a new contact  The input form has three sections of information (Appendix –Contact) – personal details, office details, and other details. The form shall have a submit button. |
| **Functional Constraint** | 1. The search text field (company) shall be enabled with auto-complete feature, i.e. as the user types in characters, a pop up list shall appear listing the existing companies in the system which name matches with the entered pattern. The items in the list shall be in the alphabetic order. From the list, the user shall be able to select a particular company. 2. The search text field (company) shall also have a facility to be converted into a drop down list (via the option – Display Company List). The list shall show all the existing companies in the system with alphabetic sorting on the company name. From the list, the user shall be able to select a particular company. 3. On select of the stream checkbox, a text field shall appear wherein the user shall enter the offered salary for recruitment. |
| **Operational Steps** | 1. On click the Save Details button, the form entries shall be validated and the failure / success message shall be displayed at the top of the page. 2. On success of field validation, the data shall be inserted in the database. 3. The failure/success message (if any) during the operation (database) shall be displayed at the top of the page. 4. On success of database insertion, the new contact shall be added to the system. The system shall also display alert information to add a remainder as well as an activity for the contact. |

### View Contact

|  |  |
| --- | --- |
| **Navigation** | Contacts → View |
| **Access Privilege** | User |
| **Page Content** | The page shall have a search text field to facilitate searching for an existing company.  The page shall display a form (***Contact List***) in the left side where the user shall able to select particular (or all) contacts.  The page shall display the ***Contact*** Details List (Appendix – View Contact) in the right side. |
| **Functional Constraint** | 1. The search text field (company) shall be enabled with auto-complete feature, i.e. as the user types in characters, a pop up list shall appear listing the existing companies in the system which name matches with the entered pattern. . The items in the list shall be in the alphabetic order. From the list, the user shall be able to select a particular company. 2. The search text field (company) shall also have a facility to be converted into a drop down list (via the option – Display Company List). The list shall show all the existing companies in the system with alphabetic sorting on the company name.   There shall be an option to limit the number of items in the drop down list via ***Display only my Company List***. The list shall only those companies for which the logged in user is the owner of the contact(s). From drop down list, the user shall be able to select a particular company.   1. On select of the company, the Contact List is populated with contact names with checkbox selection. By default, all the contacts are selected; however the user shall have the flexibility to select and deselect any one or all. 2. On select of the company, the ***Contact*** Details List shall display the details of all the contacts (by default). 3. The column headers of the ***Contact*** Details List are 4. Sl. No. (#) – number starting from 1 5. Name – name of the contact 6. Primary Contacts – shall display the Email, Mobile, Direct Number, and Extension. This information shall be visible to the owner of the contact (not even for admin). The column shall display “Not Authorised” for other users of the application. 7. Other Contacts – shall display the board line number of the contact 8. Job Position – shall display the designation of the contact 9. Contact(s) in Touch – shall display the name of the user who had most recently contacted the contact. 10. Office Details – shall display the office address of the contact 11. Actions - shall display the following images, which are activated by the mouse click event. On the mouse click event on the image, a dialog box shall open for the respective actions.     * 1. Edit contact details- A modal dialog page shall open       2. Delete the contact – a confirmation alert message shall appear       3. Add a remainder for the contact - A modal dialog page shall open       4. Add an activity for the contact - A modal dialog page shall open       5. View all remainders of the contact - A modal dialog page shall open       6. View all activities of the contact - A modal dialog page shall open 12. The images for edit and delete shall only be visible for the owner of the contact and admin, no other user of the system. 13. The images in the action column of the list shall be operated by the owner of the contact. However, the delete action shall only be authorised to the admin who has the privilege of doing all such actions. |
| **Operational Steps** | 1. Select the company name from the company search text field. 2. The ***Contact List*** shall be populated with all the contacts of the company. By default all the contacts are selected. And the ***Contact*** Details List shall display the details of all the contacts. 3. On selection/de-selection of the particular contacts from the ***Contact List***, the ***Contact*** Details List shall be refreshed by clicking the button ***Fetch Contact Details***. |

### Edit Contact

|  |  |
| --- | --- |
| **Navigation** | Contacts → View → Edit |
| **Access Privilege** | User |
| **Page Content** | To edit a contact, the user shall click the edit image in the Actions column of a particular row in the Contact Details List.  The page shall display an input form.  The form shall have a submit button Send for Approval at the end. |
| **Functional Constraint** | 1. The input form shall be pre-populated with the information of the selected contact. 2. Only personal information section shall be visible. 3. The office details and other details information shall be hidden. 4. The office details information shall be displayed or hidden by selecting / de-selecting the checkbox – Edit Office Details respectively. 5. The other details information shall be displayed / hidden on selecting / de-selecting the checkbox – Edit Other Details respectively. 6. Once the form has been submitted for admin approval, the same contact shall not be allowed to edit further. |
| **Operational Steps** | 1. On click the Send for Approval button, the form entries shall be validated and the failure / success message shall be displayed at the top of the page. 2. On success of field validation, the data shall be updated in the database. 3. The failure/success message (if any) during the operation (database) shall be displayed at the top of the page. 4. On success of database operation, a message shall be displayed at the top of the page. |

### Delete Contact

|  |  |
| --- | --- |
| **Navigation** | Contacts → View → Delete |
| **Access Privilege** | Admin |
| **Functional Constraint** | To delete a contact, the admin shall click the delete image in the Actions column of a particular row in the ***Contact Details List***. The system shall ask for confirmation. |
| **Operational Steps** | 1. On cancel, the action shall have no effect. 2. On confirmation, the particular contact shall be deleted from the database. 3. The success / failure message (if any) during the database operation shall be displayed at the top of the page. 4. On success of database operation, the Contact Details List shall be refreshed. |

### Add Remainder

|  |  |
| --- | --- |
| **Navigation** | Contacts → View → Add Remainder |
| **Access Privilege** | User |
| **Page Content** | To add a remainder for the contact, the user shall click the ***Add Remainder*** image in the ***Actions*** column of a particular row in the Contact Details List.  The page shall display an input form (Appendix –Remainder).  The form shall have a submit button and a reset button at the end. |
| **Functional Constraint** | 1. The ***Today’s Date*** shall be pre-populated with system date and shall not be editable. 2. The ***Expected Closing Date*** shall be picked up from the date picker which shall be activated on mouse click in the field. 3. The ***Expected Closing Date*** shall not precede the ***Today’s Date***. 4. The default status of the remainder shall be O***pen***. |
| **Operational Steps** | 1. The reset button shall clear the form entries except the pre-populated information. 2. The form entries shall be validated and the success / failure message shall be displayed at the top of the page on submission. 3. The data shall be updated in the database on successful form submission. 4. The success / failure message (if any) during the operation (database) shall be displayed at the top of the page. 5. A message shall be displayed at the top of the page on successful database operation. |

### View all Remainders

|  |  |
| --- | --- |
| **Navigation** | Contacts → View → View all Remainders |
| **Access Privilege** | User |
| **Page Content** | To view all remainders for the contact, the user shall click the ***View All Remainders*** image in the ***Actions*** column of a particular row in the Contact Details List.  The page shall display the list of all remainders for the contact.  The list shall have a provision to edit the remainders. |
| **Functional Constraint** | 1. The column headers of the list shall be    1. Sl. No (#) – number starting from 1.    2. Schedule Details – the description of the remainder    3. Schedule Type – the type of remainder (Telephone / Conversation, Email, Meeting)    4. CR Team Member – name of the user who contacted the contact last    5. Closing Date – closing date of the remainder    6. Schedule Status – current status of the remainder    7. Action – shall include an edit image to facilitate editing of the particular row item or remainder 2. The Remainder Edit form shall open on the click of edit image in the Action column of the row item. |

### Edit Remainder

|  |  |
| --- | --- |
| **Navigation** | Contacts → View → View all Remainders → Edit Remainder |
| **Access Privilege** | User |
| **Page Content** | To edit a remainder for the contact, the user shall click on the Edit image in the Action column of the Remainder List.  The page shall display a form (Appendix –Remainder) in edit mode.  The form shall have a provision to change the status of the contact.  The form shall have a submit button at the end. |
| **Functional Constraint** | 1. The Remainder Publish Date, Expected Closing Date, Remainder Type, and Description shall be pre-populated in the form. 2. The status change options shall be displayed in a drop down list if the user wishes to change the current status of the contact. |
| **Operational Steps** | 1. The form entries shall be validated and the success / failure message shall be displayed at the top of the page on submission. 2. The data shall be updated in the database on successful form submission. 3. The success / failure message (if any) during the operation (database) shall be displayed at the top of the page. 4. A message shall be displayed at the top of the page on successful database operation. |

### Add an Activity

|  |  |
| --- | --- |
| **Navigation** | Contacts → View → Add Activity |
| **Access Privilege** | User |
| **Page Content** | To add an activity for the contact, the user shall click the ***Add Activity*** image in the Actions column of a particular row in the Contact Details List.  The page shall display an input form (Appendix –Activity).  The form shall have a provision to change the status of the contact.  The form shall have a submit button and a reset button at the end. |
| **Functional Constraint** | 1. The ***Activity Date*** shall be picked up from the date picker which shall be activated on mouse click in the field. 2. The status change of the contact shall be facilitated via a radio button – yes, no (default). 3. The status change options shall be displayed in a drop down list if the user wishes to change the current status of the contact. |
| **Operational Steps** | 1. The reset button shall clear the form entries except the pre-populated information. 2. The form entries shall be validated and the success / failure message shall be displayed at the top of the page on submission. 3. The data shall be updated in the database on successful form submission. 4. The success / failure message (if any) during the operation (database) shall be displayed at the top of the page. 5. A message shall be displayed at the top of the page on successful database operation. |

### View all Activities

|  |  |
| --- | --- |
| **Navigation** | Contacts → View → View all Activities |
| **Access Privilege** | User |
| **Page Content** | To view all activities for the contact, the user shall click the ***View All*** Activities image in the ***Actions*** column of a particular row in the Contact Details List.  The page shall display the list of all activities for the contact.  The list shall have a provision to edit the activity. |
| **Functional Constraint** | 1. The column headers of the list shall be    1. Sl. No (#) – number starting from 1.    2. Activity Details – the description of the activity    3. CR Team Member – name of the user who contacted the contact last    4. Activity Type – the type of activity (Telephone / Conversation, Email, Meeting)    5. Activity Date – date of the activity    6. Action – shall include an edit image to facilitate editing of the particular row item or activity 2. The Activity Edit form shall open on the click of edit image in the Action column of the row item. |

### Edit Activity

|  |  |
| --- | --- |
| **Navigation** | Contacts → View → View all Activities → Edit Activity |
| **Access Privilege** | User |
| **Page Content** | To edit an activity for the contact, the user shall click on the Edit image in the Actions column of the Activity List.  The page shall display a form (Appendix –Activity) in edit mode.  The form shall have a provision to change the status of the contact.  The form shall have a submit button at the end. |
| **Functional Constraint** | 1. The Activity Date, Activity Type, and Description shall be pre-populated in the form. 2. The status change options shall be displayed in a drop down list if the user wishes to change the current status of the contact. |
| **Operational Steps** | 1. The form entries shall be validated and the success / failure message shall be displayed at the top of the page on submission. 2. The data shall be updated in the database on successful form submission. 3. The success / failure message (if any) during the operation (database) shall be displayed at the top of the page. 4. A message shall be displayed at the top of the page on successful database operation. |

### Contact Search

|  |  |
| --- | --- |
| **Navigation** | Contacts → Contact Search |
| **Access Privilege** | User |
| **Page Content** | The page shall display a search text field in which the user shall enter some minimal information so as to search a contact. |
| **Functional Constraint** | 1. The search keywords shall be of any one of the following keywords 2. Contact Name. 3. Email Address. 4. Mobile Number. 5. Direct Number. 6. Extension Number. 7. Contact Designation. 8. Company Name. 9. Company Type. 10. Company Website. 11. Office Board Line Number. 12. Office Address. 13. Office City. 14. The search text field shall be enabled with auto-complete feature, i.e. as the user types in characters, a pop up list shall appear listing the existing information in the system which matches with the entered pattern. The items in the list shall be in the alphabetic order. From the list, the user shall be able to select particular information. 15. On select of the information, the ***Contact*** Details List shall display the details of all the contacts satisfying the search criteria. 16. The column headers of the ***Contact*** ***Details*** ***List*** shall be 17. Sl. No. (#) – number starting from 1 18. Name – name of the contact 19. Company – name of the company 20. Primary Contacts – shall display the Email, Mobile, Direct Number, and Extension. This information shall be visible to the owner of the contact (not even for admin). The column shall display “Not Authorised” for other users of the application. 21. Other Contacts – shall display the board line number of the contact 22. Job Position – shall display the designation of the contact 23. Contact(s) in Touch – shall display the name of the user who had most recently contacted the contact. 24. Office Details – shall display the office address of the contact 25. Actions - shall display the following images, which are activated by the mouse click event. On the mouse click event on the image, a dialog box shall open for the respective actions.     * 1. Edit contact details- A modal dialog page shall open       2. View all remainders of the contact - A modal dialog page shall open       3. View all activities of the contact - A modal dialog page shall open 26. The images in the action column of the list shall be operated by the owner of the contact. However, the admin who has the privilege of doing all such actions. |

### Contact Edit Status

|  |  |
| --- | --- |
| **Navigation** | Contacts → Contact Edit Status |
| **Access Privilege** | User |
| **Page Content** | The page shall display an Approval Status List where all the contact edit requests shall be displayed. The user shall able to view the list edit requests. |
| **Functional Constraint** | 1. The column headers of the Approval Status List shall be 2. Sl. No. (#) – number starting from 1 3. Company Name – name of the company 4. Contact Name – name of the contact 5. Category – the category of information (personal details, office details, other details) subjected to edit 6. Time – timestamp of edit request 7. Status – Pending 8. The status of all such requests shall be pending, once approved by the admin; the item shall be cleared from the list. |

## Remainder List

### Add Remainder

|  |  |
| --- | --- |
| **Navigation** | Remainder List → New |
| **Access Privilege** | User |
| **Page Content** | A remainder shall be created either for self or for a corporate contact person.  The page shall display an input form (Appendix –Remainder).  The form shall have a submit button and a reset button at the end. |
| **Functional Constraint** | 1. The ***Today’s Date*** shall be pre-populated with system date and shall not be editable. 2. The ***Expected Closing Date*** shall be picked up from the date picker which shall be activated on mouse click in the field. 3. The ***Expected Closing Date*** shall not precede the ***Today’s Date***. 4. There shall be a text box where the user shall provide a brief description of the remainder. 5. The default status of the remainder shall be O***pen***. 6. The form shall have a provision to set the remainder for a corporate contact person via a checkbox selection, if not selected; the remainder shall be set for the user. 7. The form shall display a search text field to search for the company the contact belongs. The search text field shall be enabled with auto-complete feature, i.e. as the user types in characters, a pop up list shall appear listing the existing companies in the system which matches with the entered pattern. The items in the list shall be in the alphabetic order. From the list, the user shall be able to select particular company. 8. The search text field (company) shall also have a facility to be converted into a drop down list (via the option – Display Company List). The list shall show all the existing companies in the system with alphabetic sorting on the company name. 9. As the user selects a particular company, the contact drop-down list shall be pre-populated with the existing contacts belonging to the selected company, from which the user shall select a particular contact for which the remainder shall be set. |
| **Operational Steps** | 1. The reset button shall clear the form entries except the pre-populated information. 2. The form entries shall be validated and the success / failure message shall be displayed at the top of the page on submission. 3. The data shall be updated in the database on successful form submission. 4. The success / failure message (if any) during the operation (database) shall be displayed at the top of the page. 5. A message shall be displayed at the top of the page on successful database operation. |

### View Remainder

|  |  |
| --- | --- |
| **Navigation** | Remainder List → View |
| **Access Privilege** | User |
| **Page Content** | The page shall have different provisions to view the remainders.  The page shall display a form (***Appendix – View Remainder***) in the left side where the user shall able to select the criteria for viewing the remainders and the Remainder List in the right side.  The form shall have a submit button and a reset button at the end. |
| **Functional Constraint** | 1. The user shall be able select to view the remainders either created by self or created by other users. 2. The user shall be able to view the remainder having different status. 3. The user shall be able to view the remainders set for a particular company. 4. The user shall be able to view the remainders for a particular contact belonging to a particular company. 5. The search text field (company) shall be enabled with auto-complete feature, i.e. as the user types in characters, a pop up list shall appear listing the existing companies in the system which name matches with the entered pattern. . The items in the list shall be in the alphabetic order. From the list, the user shall be able to select a particular company. 6. The search text field (company) shall also have a facility to be converted into a drop down list (via the option – Display Company List). The list shall show all the existing companies in the system with alphabetic sorting on the company name. 7. On select of the company, the Contact drop down list is populated with contact names. The user shall be able to select only one contact name in order to view the remainders. 8. The reset button shall clear the form entries except the default options. 9. The column headers of the ***Remainder*** List shall be    1. Sl. No. (#) – number starting from 1    2. Published Date – remainder create date    3. Closing Date – scheduled closing date    4. Remainder Type – type of remainder    5. Details – brief description of the remainder    6. Status – status of the remainder    7. Contact – Company Name along with the contact name for which the remainder has been set. This field shall be empty in case of self remainder.    8. Action - shall display the following images, which are activated by the mouse click event. On the mouse click event on the image, modal dialog box shall open for the respective actions.       1. Edit Remainder- to edit the remainder       2. Close Remainder – to close the remainder |
| **Operational Steps** | 1. Select the member (self or others) for which the remainders shall be displayed. 2. Select the status of the remainders to view. 3. Select the company name from the company search text field / drop down list. 4. Select the contact name for which remainders shall be displayed. 5. The ***Remainders List*** shall display all the remainders matching the selection criteria. |

### Close Remainder

|  |  |
| --- | --- |
| **Navigation** | Remainder List → View → Close Remainder |
| **Access Privilege** | User |
| **Page Content** | A remainder shall be closed against an activity. To close a remainder the user shall add an activity and then close the remainder. The user shall click on the Close Remainder image in the Action column of the Remainder List.  The page shall display a form (Appendix –Close Remainder).  The form shall have a provision to change the status of the contact.  The form shall have a submit button and a reset button at the end.  The form shall have an option to add a new remainder for the contact (see section 4.14.5). |
| **Functional Constraint** | 1. The status change options shall be displayed in a drop down list if the user wishes to change the current status of the contact. |
| **Operational Steps** | 1. The form entries shall be validated and the success / failure message shall be displayed at the top of the page on submission. 2. The data shall be updated in the database on successful form submission. 3. The success / failure message (if any) during the operation (database) shall be displayed at the top of the page. 4. A message shall be displayed at the top of the page on successful database operation. |

## Calendar

### Calendar

|  |  |
| --- | --- |
| **Navigation** | Calendar → Calendar |
| **Access Privilege** | User |
| **Page Content** | The page shall display the calendar for the current month. The header of the calendar shall display the weekdays starting from Sunday to Saturday. |
| **Functional Constraint** | 1. The calendar page shall have navigation facility for backward and forward. 2. The page shall also have the facility to select a particular month and year as well. 3. The calendar shall show all the activities, remainders, travel plans for the logged in user only, however shall also include a facility to view the calendar entries for all individual users. 4. To highlight different entries in the calendar proper colour combination shall be used. The colour code shall also be displayed as legend at the bottom of the page. 5. All the entries in the calendar shall be hyperlinked and modal dialog boxes shall open to view the respective entries. 6. Travel plans for the logged in user shall also be displayed at the bottom of the page. |

### Travel Plan

|  |  |
| --- | --- |
| **Navigation** | Calendar → Travel Plan |
| **Access Privilege** | User |
| **Actions** | Add, Edit, and Delete |
| **Page Content** | The page shall display the add/edit form (***Appendix – Travel Plan***) in the left side and the ***Travel Plan*** List in the right side.  The form shall have a submit button and a reset button at the end. |
| **Functional Constraint** | 1. A travel plan shall not be added or updated with overlapping date and time with any other existing travel plan belonging to the same user. However the system shall display a warning message if the new travel plan conflicts with any other existing travel plan created by other user. 2. The Travel Plan List shall display all the existing travel plans created by the logged in user. 3. The list shall display the most recently created travel plan in the first place. 4. The column headers of the ***Travel Plan*** List shall be    1. Sl. No. (#) – number starting from 1    2. Start Date – start date of travel    3. End Date – end date of travel    4. Destination – travel destination    5. Description – brief description of the travel plan    6. Edit - shall display the edit and delete images, which are activated by the mouse click event. 5. The page shall display any success / failure message on form submission and database operation |
| **Actions** | |
| **Add** | To add a ***travel plan***, the user shall manually enter the information in the form shown in the left side of the page.  The reset button shall clear the form entries except the default options.  The newly created travel plan shall be displayed in the Travel Plan List as the first entry. |
| **Edit** | To edit a ***travel plan***, the user shall click on the edit image in the Edit column of a particular row in the ***Travel Plan*** List.  The system shall populate the ***travel plan*** information in the edit form (on the left side of the page).  The form has one submit button.  The updated travel plan shall be refreshed in the Travel Plan List. |
| **Delete** | To delete a ***travel plan***, the user shall click the delete image in the Edit column of a particular row in the ***Travel Plan*** List.  The system shall ask for confirmation. The cancel shall have no effect on the system  The travel plan shall be deleted from the system on confirmation and the ***Travel Plan*** List shall be refreshed. |

## Reports

### Details by Office

|  |  |
| --- | --- |
| **Navigation** | Reports → Details By Office |
| **Access Privilege** | User |
| **Description** | The page shall generate a report about the existing offices associated with a particular company (search item) and the respective contact details. |
| **Page Content** | The page shall have a search text field to facilitate searching for an existing company for which the different offices and their respective contact details shall be viewed.  The page shall display a form (***Office List***) in the left side where the user shall able to select particular (or all) offices associated with the company.  The page shall display the ***Office and Respective Contact*** Details List in the right side. |
| **Functional Constraint** | 1. The search text field (company) shall be enabled with auto-complete feature, i.e. as the user types in characters, a pop up list shall appear listing the existing companies in the system which name matches with the entered pattern. . The items in the list shall be in the alphabetic order. From the list, the user shall be able to select a particular company. 2. The search text field (company) shall also have a facility to be converted into a drop down list (via the option – Display Company List). The list shall show all the existing companies in the system with alphabetic sorting on the company name. 3. The Office List shall populate all the offices associated with the selected company. The list shall have multiple selection option so as to select individual item as well as all the items. By default, all the offices shall be selected. 4. The ***Office and Respective Contact*** Details List shall display the details of all the offices and the respective contact details (by default). 5. The column headers of the list shall be    1. Sl. No. (#) – number starting from 1    2. Office Type – type of the office    3. Address – address of the office.    4. City – City / District of the office    5. Name – name of the contact person    6. Position – designation of the contact person    7. Contact Details – shall display the email, mobile, direct, and extension information of the contact    8. Actions - shall display images for following actions.  * Add a Remainder * Add an Activity  1. The images in the action column of the list shall open modal dialog box on single mouse click event for respective functionality. 2. The list shall have an option to download the contents into a MS-XLS spreadsheet. |

### Activities by Company

|  |  |
| --- | --- |
| **Navigation** | Reports → Activities By Company |
| **Access Privilege** | User |
| **Description** | The page shall generate a report about the activities recorded by different users for a particular company (search item). |
| **Page Content** | The page shall have a search text field to facilitate searching for an existing company for which the different activities recorded by the users shall be viewed.  The page shall display a form in the left side where the user shall able to select particular date range within which the details of activities shall be viewed.  The page shall display the ***Activity*** Details List in the right side. |
| **Functional Constraint** | 1. The search text field (company) shall be enabled with auto-complete feature, i.e. as the user types in characters, a pop up list shall appear listing the existing companies in the system which name matches with the entered pattern. . The items in the list shall be in the alphabetic order. From the list, the user shall be able to select a particular company. 2. The search text field (company) shall also have a facility to be converted into a drop down list (via the option – Display Company List). The list shall show all the existing companies in the system with alphabetic sorting on the company name. 3. The form shall have a search facility to view the activities with a date range i.e. the form shall have two date entry fields and a submit button. 4. The Activity Details List shall display all the activities recorded by the users for the company within the selected date range. 5. The column headers of the list shall be    1. Sl. No. (#) – number starting from 1    2. Name – name of the contact    3. Primary Contacts – shall display the email, mobile, direct, and extension information of the contact    4. Job Position – designation of the contact person    5. Office Details - address of the office.    6. Contacted By – name of the user who recorded an activity with the contact    7. Activity Type – the type of activity    8. Activity Date – the date of activity    9. Activity Detail – brief description of the activity    10. Actions - shall display images for following actions.        1. Add a Remainder        2. Add an Activity        3. Edit the Activity 6. The images in the action column of the list shall open modal dialog box on single mouse click event for respective functionality. 7. The list shall have an option to download the contents into a MS-XLS spreadsheet. |

### Activities by Contact

|  |  |
| --- | --- |
| **Navigation** | Reports → Activities By Contact |
| **Access Privilege** | User |
| **Description** | The page shall generate a report about the activities recorded against each of the contacts associated with a particular company (search item). |
| **Page Content** | The page shall have a search text field to facilitate searching for an existing company for which the different activities recorded by the users with various contacts shall be viewed.  The page shall display a form (***Contact List***) in the left side where the user shall able to select particular contact associated with the company.  The page shall display the ***Activity*** Details List in the right side. |
| **Functional Constraint** | 1. The search text field (company) shall be enabled with auto-complete feature, i.e. as the user types in characters, a pop up list shall appear listing the existing companies in the system which name matches with the entered pattern. . The items in the list shall be in the alphabetic order. From the list, the user shall be able to select a particular company. 2. The search text field (company) shall also have a facility to be converted into a drop down list (via the option – Display Company List). The list shall show all the existing companies in the system with alphabetic sorting on the company name. 3. The Contact List shall populate all the contacts associated with the selected company. The list shall have single selection option so as to select individual item. By default, the selection option shall highlight the last contact person on behalf of the company. 4. The ***Activity*** Details List shall display the details of all the activities with respective to the contact. 5. The column headers of the list shall be    1. Sl. No. (#) – number starting from 1    2. Name – name of the contact person    3. Activity Details – the brief description of the activity    4. Contacted By – name of the user who recorded an activity with the contact    5. Activity Type – the type of activity    6. Activity Date – the date of activity    7. Actions - shall display images for following actions.       1. Add a Remainder       2. Add an Activity       3. Edit the Activity 6. The images in the action column of the list shall open modal dialog box on single mouse click event for respective functionality. 7. The list shall have an option to download the contents into a MS-XLS spreadsheet. |

### Activities by Member

|  |  |
| --- | --- |
| **Navigation** | Reports → Activities By Member |
| **Access Privilege** | User |
| **Description** | The page shall generate a report about the activities recorded by each of the existing users of the system. |
| **Page Content** | The page shall display a form (***Member List***) in the left side where all the existing users of the application shall be listed so that the logged in user shall be able to select only one member at a time and view the activity details of the member.  The page shall display the ***Activity*** Details List in the right side. |
| **Functional Constraint** | 1. The ***Member*** List shall populate all the existing users (name of users) of the system. The list shall have single selection option so as to select individual item. 2. The Member List shall have a submit button which shall submit the selected user, so that the activity details of the selected user shall be displayed in the Activity Details List. 3. The ***Activity*** Details List shall display the details of all the activities with respective to the selected user. 4. The column headers of the list shall be    1. Sl. No. (#) – number starting from 1    2. Contact Name – name of the contact person    3. Activity Details – the brief description of the activity    4. Company Name – name of the company the contact associated with    5. Activity Type – the type of activity    6. Activity Date – the date of activity    7. Actions - shall display images for following actions.       1. Add a Remainder       2. Add an Activity       3. Edit the Activity 5. The list shall be sorted in descending order of Activity Date column. 6. The images in the action column of the list shall open modal dialog box on single mouse click event for respective functionality. 7. The list shall have an option to download the contents into a MS-XLS spreadsheet. |

### Scheduled Items

|  |  |
| --- | --- |
| **Navigation** | Reports → Scheduled Items |
| **Access Privilege** | User |
| **Description** | The page shall generate a report about the remainders scheduled for the respective contacts associated with a particular company (search item) by the existing users of the system. |
| **Page Content** | The page shall have a search text field to facilitate searching for an existing company for which the different remainders recorded by the users shall be viewed.  The page shall display a form in the left side where the user shall able to select particular date range within which the details of remainders shall be viewed.  The page shall display the ***Schedule*** Details List in the right side. |
| **Functional Constraint** | 1. The search text field (company) shall be enabled with auto-complete feature, i.e. as the user types in characters, a pop up list shall appear listing the existing companies in the system which name matches with the entered pattern. . The items in the list shall be in the alphabetic order. From the list, the user shall be able to select a particular company. 2. The search text field (company) shall also have a facility to be converted into a drop down list (via the option – Display Company List). The list shall show all the existing companies in the system with alphabetic sorting on the company name. 3. The form shall have a search facility to view the remainders with a date range i.e. the form shall have two date entry fields and a submit button. 4. The Schedule Details List shall display all the remainders recorded by the users for the company within the selected date range. 5. The column headers of the list shall be    1. Sl. No. (#) – number starting from 1    2. Name – name of the contact    3. Primary Contacts – shall display the email, mobile, direct, and extension information of the contact    4. Job Position – designation of the contact person    5. Office Details - address of the office.    6. Scheduled By – name of the user who recorded a remainder with the contact    7. Schedule Date – the date of remainder    8. Activity Detail – brief description of the remainder    9. Closure Date – closing date of the remainder    10. Actions - shall display images for following actions.        1. Add a Remainder        2. Add an Activity        3. Edit the Remainder 6. The images in the action column of the list shall open modal dialog box on single mouse click event for respective functionality. 7. The list shall have an option to download the contents into a MS-XLS spreadsheet. |

### Probable Visits

|  |  |
| --- | --- |
| **Navigation** | Reports → Probable Visits |
| **Access Privilege** | Admin |
| **Description** | The page shall generate a report about the final contact status (search item) of the activities recorded by users for companies in order to have campus recruitment drive. |
| **Page Content** | The page shall have a drop down list to select the final contact status with respect to company and accordingly shall display a list of companies satisfying the search criteria.  The page shall display a form in the left side where the user shall able to select particular contact status for which the details of companies shall be viewed.  The page shall display the ***Company Response*** List in the right side. |
| **Functional Constraint** | 1. The form shall have a drop down list pre-populated with the validated entries from Contact Status (see Manage Contact Status) so that the user shall be able to select only one item. By default no item shall be selected. 2. The form shall have the following additional search criteria when the user selects “positive” contact status from the drop down list.    1. Probable month of visit – user shall be able to select a calendar month (January through December)    2. Probable year of visit – user shall be able to select a calendar year (last year through five advance years)    3. Stream – user shall able to select (multiple) any of the available Streams (see Manage Stream) 3. The form shall have a submit button and a reset button as well. 4. The reset button shall bring the form back to default state. 5. The Company Response List shall be displayed on form submission. 6. The Company Response List shall display all the companies satisfying the selected status from the drop down list. 7. The column headers of the list shall be    1. Sl. No. (#) – number starting from 1    2. Company Name – name of the company    3. Company Type – type of company    4. Expected Date / Comment – expected date of visiting the campus drive    5. Employee – name of the user who contacted last with the company 8. The list shall be sorted in ascending alphabetic order of Company Name. 9. The list shall have an option to download the contents into a MS-XLS spreadsheet. |

### Employee Performance

|  |  |
| --- | --- |
| **Navigation** | Reports → Employee Performance |
| **Access Privilege** | Admin |
| **Description** | The page shall generate a report about the use of this application for different purposes by a particular user (search item) for a valid time period. |
| **Page Content** | The page shall display a form in the left side where the user shall able to select particular user and the date range within which the details of use of the application by the user shall be viewed.  The page shall display the ***User Performance Report*** in the right side. |
| **Functional Constraint** | 1. The form shall have a drop down list pre-populated with the existing users in the system, so that the user shall be able to select only one item. By default no item shall be selected. 2. The form shall have a date range entry i.e. from date and to date, within which the performance report of the selected user shall be viewed. 3. The search criteria shall be inclusive of the two dates selected. 4. The form shall have a submit button and a reset button. 5. The reset button shall bring the form back to default state. 6. The ***User Performance Report*** shall be displayed on form submission. 7. The ***User Performance Report*** shall display all the activities by the selected user. 8. The column headers of the ***User Performance Report*** shall be    1. Sl. No. (#) – number starting from 1    2. Date – date of logon into the system    3. Login Time – duration of logon (hh:mm:ss)    4. Meeting – time duration for Meeting type activity    5. Telephone - time duration for Telephone type activity    6. Email – time duration for Email type activity 9. The report shall be sorted in descending order of Date column. 10. The list shall have an option to download the contents into a MS-XLS spreadsheet. |

### Travel Report

|  |  |
| --- | --- |
| **Navigation** | Reports → Travel Report |
| **Access Privilege** | Admin |
| **Description** | The page shall generate a report about the travel by different users (search item) for valid time period. |
| **Page Content** | The page shall display a form in the left side where the user shall able to select particular user and the date range within which the details of the travel by the selected user shall be viewed.  The page shall display the ***Travel Report*** in the right side. |
| **Functional Constraint** | 1. The form shall have a drop down list pre-populated with the existing users in the system, so that the logged in user shall be able to select only one item. 2. The form shall have a date range entry i.e. from date and to date, within which the travel report of the selected user shall be viewed. 3. By default neither any item in the drop down list nor the date range shall be selected in the form. 4. The search criteria shall be inclusive of the two dates selected. 5. The form shall have a submit button and a reset button. 6. The reset button shall bring the form back to default state. 7. The ***Travel Report*** shall be displayed on form submission. 8. The ***Travel Report*** shall display all the travels by the selected user. For the specific time period. 9. The column headers of the ***Travel Report*** shall be    1. Sl. No. (#) – number starting from 1    2. Destination – travel destination    3. Start Date – start date (dd-mm-yyyy) and time of travel (hh:mm:ss)    4. End Date – end date (dd-mm-yyyy) and time of travel (hh:mm:ss)    5. Travel Time - time (hh:mm) duration of travel (difference value of start time and end time) 10. The report shall be sorted in descending order of Start Date column. 11. The list shall have an option to download the contents into a MS-XLS spreadsheet. |

### Contact Report

|  |  |
| --- | --- |
| **Navigation** | Reports → Travel Report |
| **Access Privilege** | Admin |
| **Description** | The page shall generate status report about the contacts of the respective companies based on various search criteria. The following are the search criteria.   1. Employee Wise - report based on particular user 2. Company Wise – report based on particular company 3. Stream Wise – report based on particular stream 4. Company + Stream Wise – report based on both company and stream |
| **Employee Wise Contact Report** | |
| **Page Content** | The page shall display a form in the left side where the logged in user shall able to select particular user and a particular company (or all companies) for which the contact details shall be viewed.  The page shall display the ***Contact Details*** in the right side. |
| **Functional Constraint** | 1. The form shall have a drop down list pre-populated with the existing users in the system, so that the logged in user shall be able to select only one item. By default no item shall be selected. 2. The user names in the user drop down list shall be in sorted in ascending order of user name. 3. The form shall have a drop down list pre-populated with the existing companies in the system and have a All option (for all companies), so that the logged in user shall be able to select only one item. By default All option shall be the selected item. 4. The company names in the company drop down list shall be sorted in ascending order of company name. 5. The company drop down list shall be visible and the default option shall be the selected item only when the logged in user selects an item (a user) from the user drop down list. 6. The company drop down list shall remain visible as long as the user drop down list shows a valid user as the selected item. 7. The search criteria shall be inclusive of both the selected items from the two drop down lists. 8. The ***Contact Details*** shall be displayed (and refreshed) on change of the selected items from the two drop down lists. 9. The ***Contact Details*** shall display all the records available in the system satisfying the search criteria. 10. The column headers of the ***Contact Details*** shall be     1. Sl. No. (#) – number starting from 1     2. Name – name of the contact     3. Company – name of the company     4. Primary Contacts - the email, mobile, direct, and extension information of the contact     5. Other Contacts - the board line number of the contact     6. Job Position – designation of the contact along with department information     7. Office Details – the office address     8. Status – final contact status 11. The report shall be sorted in ascending order of Name column. 12. The list shall have an option to download the contents into a MS-XLS spreadsheet. |
| **Company Wise Contact Report** | |
| **Page Content** | The page shall display a form in the left side where the logged in user shall able to select particular company for which the contact details shall be viewed.  The page shall display the ***Contact Details*** in the right side. |
| **Functional Constraint** | 1. The form shall have a drop down list pre-populated with the existing companies in the system, so that the logged in user shall be able to select only one item. By default no item shall be selected. 2. The company names in the company drop down list shall be sorted in ascending order of company name. 3. The ***Contact Details*** shall be displayed (and refreshed) on change of the selected items from the company drop down list. 4. The ***Contact Details*** shall display all the records available in the system satisfying the search criteria. 5. The column headers of the ***Contact Details*** shall be    1. Sl. No. (#) – number starting from 1    2. Name – name of the contact    3. Company – name of the company    4. Primary Contacts - the email, mobile, direct, and extension information of the contact    5. Other Contacts - the board line number of the contact    6. Job Position – designation of the contact along with department information    7. Office Details – the office address    8. Status – final contact status 6. The report shall be sorted in ascending order of Name column. 7. The list shall have an option to download the contents into a MS-XLS spreadsheet. |
| **Stream Wise Contact Report** | |
| **Page Content** | The page shall display a form in the left side where the logged in user shall able to select particular stream for which the contact details shall be viewed.  The page shall display the ***Contact Details*** in the right side. |
| **Functional Constraint** | 1. The form shall have a drop down list pre-populated with the existing streams in the system, so that the logged in user shall be able to select only one item. By default no item shall be selected. 2. The items in the stream drop down list shall be sorted in ascending order of stream name. 3. The ***Contact Details*** shall be displayed (and refreshed) on change of the selected items from the stream drop down list. 4. The ***Contact Details*** shall display all the records available in the system satisfying the search criteria. 5. The column headers of the ***Contact Details*** shall be    1. Sl. No. (#) – number starting from 1    2. Name – name of the contact    3. Company – name of the company    4. Primary Contacts - the email, mobile, direct, and extension information of the contact    5. Other Contacts - the board line number of the contact    6. Job Position – designation of the contact along with department information    7. Office Details – the office address    8. Status – final contact status 6. The report shall be sorted in ascending order of Name column. 7. The list shall have an option to download the contents into a MS-XLS spreadsheet. |
| **Company + Stream Wise Contact Report** | |
| **Page Content** | The page shall display a form in the left side where the logged in user shall able to select particular company and a particular stream for which the contact details shall be viewed.  The page shall display the ***Contact Details*** in the right side. |
| **Functional Constraint** | 1. The form shall have a drop down list pre-populated with the existing companies in the system, so that the logged in user shall be able to select only one item. By default no item shall be selected. 2. The items in the company drop down list shall be in sorted in ascending order of company name. 3. The form shall have a drop down list pre-populated with the existing streams associated with the selected company from the company drop down list, so that the logged in user shall be able to select only one item. By default no item shall be selected. 4. The items in the stream drop down list shall be sorted in ascending order. 5. The stream drop down list shall be visible and the default option shall be the selected item only when the logged in user selects an item (a company) from the company drop down list. 6. The stream drop down list shall remain visible as long as the company drop down list shows a valid company as the selected item. 7. The search criteria shall be inclusive of both the selected items from the two drop down lists. 8. The ***Contact Details*** shall be displayed (and refreshed) on change of the selected items from the two drop down lists. 9. The ***Contact Details*** shall display all the records available in the system satisfying the search criteria. 10. The column headers of the ***Contact Details*** shall be     1. Sl. No. (#) – number starting from 1     2. Name – name of the contact     3. Company – name of the company     4. Primary Contacts - the email, mobile, direct, and extension information of the contact     5. Other Contacts - the board line number of the contact     6. Job Position – designation of the contact along with department information     7. Office Details – the office address     8. Status – final contact status 11. The report shall be sorted in ascending order of Name column. 12. The list shall have an option to download the contents into a MS-XLS spreadsheet. |

# Documentation

(Innovadors Lab Pvt. Ltd. shall furnish user guide as supportive document describing how to use the system.

# Application Access

The application shall be accessible globally via internet. The following internet browser (latest version as of delivery date) compatibility shall be enforced.

* Internet Explorer
* Mozilla Firefox
* Google Chrome

# System Requirements

### Hardware

### Software

### User Capacity

The maximum number of simultaneous concurrent users (during 2014) is estimated not to exceed 10.

### Location Requirement

# Operational / Maintenance Support

The system shall be supported and maintained by Innovadors Lab Pvt. Ltd. as per the signed Service Level Agreement.

# Exceptions

# Acceptance Criteria

|  |  |
| --- | --- |
| **ID** | **Description** |
| AC-01 |  |
| AC-02 |  |
| AC-03 |  |

# Constraints

*Constraints that inhibit the project’s ability to deliver the requirements.*

# Assumptions

*Any assumptions made, in conjunction with the user(s), which may become apparent at the later stages of the specification process. For example, the Functional Specification may show certain requirements not to be feasible.*

# Dependencies

*Any dependency to which the project’s completion is contingent upon.*

# Risks

*Any quantifiable losses that might arise.*

# References

# Appendix

## Appendix 1: User Logon

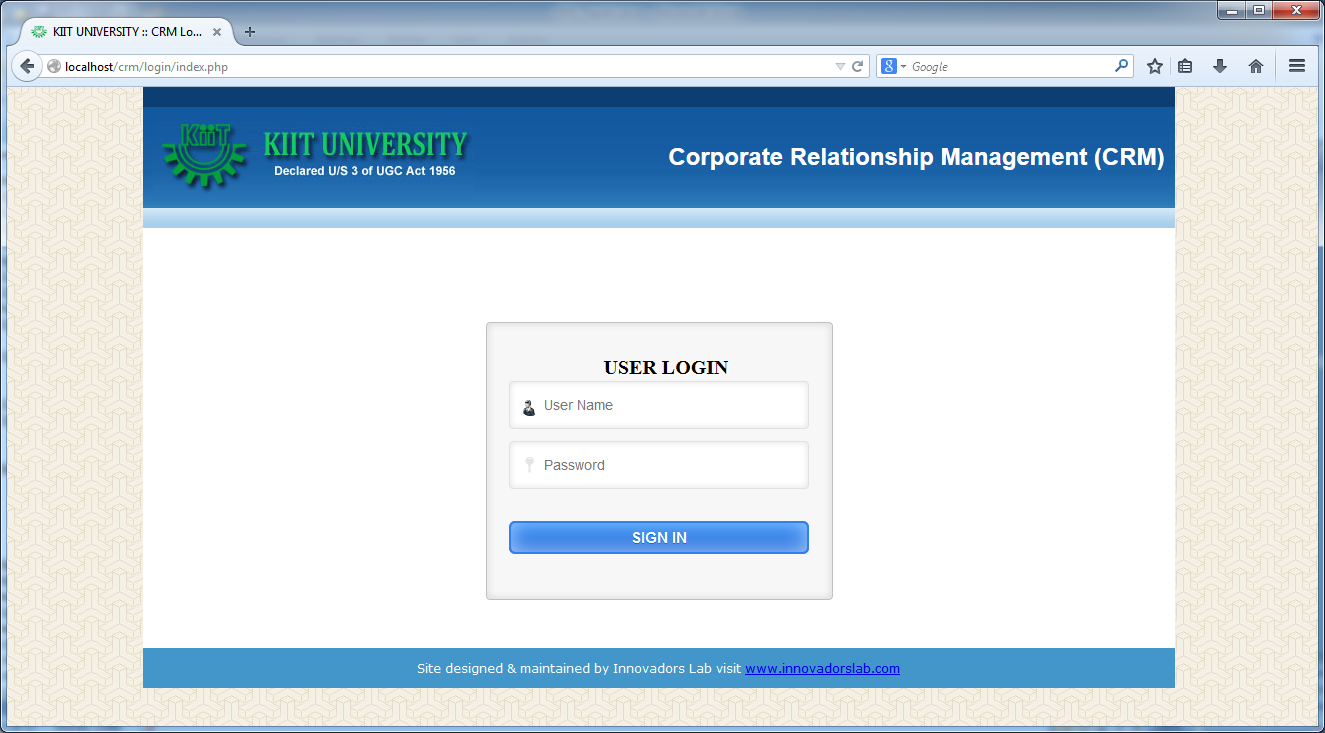


Figure 1: User Logon

## Appendix 2: Welcome Screen and Page Structure

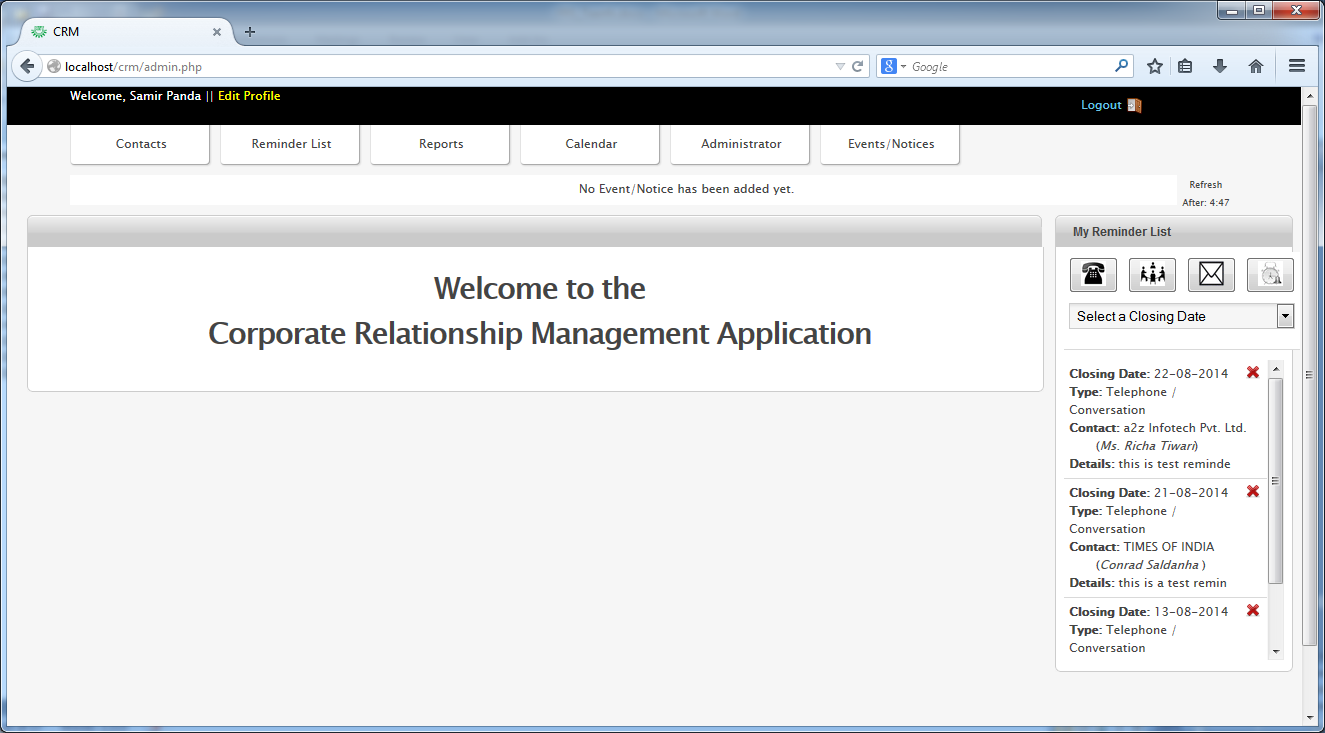


Figure 2: Welcome User and Page Structure

## Appendix - User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Employee Name | Text | M | Add/Edit | Name of the employee |
| User ID | Text | M | Add/Edit | Unique user id, required for log on |
| Password | Text | M | Add/Edit | Password, required for log on |
| User Type | Drop-Down | M | Add/Edit | User Type – Admin (for administrator), User (for normal user) |
| Date of Birth | Date | O | Add/Edit | Date of Birth of user |
| Email | Text | O | Add/Edit | Email of user |
| Mobile | Text | O | Add/Edit | Mobile number of user |
| Address | Text | O | Add/Edit | Address of user |
| Status | Drop-Down | O | Edit | Status – Active, Inactive |

## Appendix - Company Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Company Type | Text | M | Add/Edit | Name of the company type |
| Status | Drop-Down | O | Edit | Status – Active, Inactive |

## Appendix - Company

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Company Name | Text | M | Add/Edit | Name of the company |
| Company Type | Drop-Down | M | Add/Edit | Company Type – validated entries from Company Type |
| Website | Text | O | Add/Edit | http website address of the company |

## Appendix - Office Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Office Type | Text | M | Add/Edit | Name of the office type |
| Status | Drop-Down | O | Edit | Status – Active, Inactive |

## Appendix - Department

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Functional Department | Text | M | Add/Edit | Name of the Functional Department |
| Status | Drop-Down | O | Edit | Status – Active, Inactive |

## Appendix - Stream

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Stream Name | Text | M | Add/Edit | Name of the Stream |
| Status | Drop-Down | O | Edit | Status – Active, Inactive |

## Appendix - District

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| City | Text | M | Add/Edit | City |
| State | Drop-Down | M | Add/Edit | State – all the states of India, and other.  When the user selects a state of India, the country field shall be populated with India, and the country field is disabled.  When the user selects other, a free-text field appears below, where the user has to enter the district, and country field is enabled and becomes mandatory for selection. |
| Country | Drop-Down | M | Add/Edit | Country – all countries of the world |

## Appendix - Travel Destination

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Travel Destination | Text | M | Add/Edit | Travel Destination |
| Status | Drop-Down | O | Edit | Status – Active, Inactive |

## Appendix - Salutation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Salutation | Text | M | Add/Edit | Salutation |
| Status | Drop-Down | O | Edit | Status – Active, Inactive |

## Appendix - Contact Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Contact Status | Text | M | Add/Edit | Contact Status |
| Status | Drop-Down | O | Edit | Status – Active, Inactive |

## Appendix - Contact Relationship

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Contact Relationship | Text | M | Add/Edit | Contact Relationship |
| Status | Drop-Down | O | Edit | Status – Active, Inactive |

## Appendix - Swapping Contact

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| From User | Drop-Down | M | Swap | From user name |
| To User | Drop-Down | M | Swap | To user name |

## Appendix - Contact

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| **Personal Details** | | | | |
| Salutation | Drop-Down | M | Add | Salutation – validated entries from Salutation |
| Name | Text | M | Add | Name of the contact |
| Email | Text |  | Add | Email of the contact |
| Mobile | Digit |  | Add | Mobile number of the contact |
| Direct Number | Digit |  | Add | Direct phone number of the contact |
| Extension Number | Digit |  | Add | Extension Number of the contact |
| Designation | Text |  | Add | Designation of the contact |
| Department | Drop-Down | M | Add | Department of the contact – validated entries from Department |
| Current Status | Drop-Down | M | Add | Status of the contact – Active (default), Inactive |
| **Office Details** | | | | |
| Way to enter the office details | Drop-Down | M | Add | Whether to select an office type (already attached to the selected company) or a new office type |
| Office Type | Drop-Down | M | Add | Office Type – validated entries from Office Type |
| Board Line Number | Digits | M | Add | Board Line Number of the office |
| Address Line 1 | Text | M | Add | Address Line 1 of the office |
| Address Line 2 | Text | O | Add | Address Line 2 of the office |
| Country | Drop-Down | M | Add | Country of the office – India or Others |
| City | Drop-Down | M | Add | City of the office – validated entries from District |
| Pin | Digits | O | Add | Pin-code number of the city |
| Recruitment Calls are taken from this office | Checkbox | O | Add | Yes (default) |
| **Other Details** | | | | |
| Stream | Checkbox | M | Add | Stream of the contact – validated entries from Stream |
| Offered Salary | Text | O | Add | Salary offered during recruitment |
| Contact Status | Drop-Down | M | Add | Status of the contact – validated entries from Contact Status |
|  |  |  | Add |  |

## Appendix - Remainder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Remainder Date | Date | M | Add / Edit | Auto populated with today’s date and not editable |
| Expected Closing Date | Date | M | Add / Edit | Date selected from date picker for expected closing date |
| Remainder Type | Drop-Down | M | Add / Edit | Validated entries are  Telephone / conversation  Email  Meeting |
| Description | Text | O | Add / Edit | Brief description |
| Current Status | Drop-Down | M | Add | Open (default) |
| Corporate Contact | Drop-Down | O | Add | The corporate contact person for which the remainder shall be set. If not selected the remainder shall be set for the self. |

## Appendix – View Remainder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Select the Member | Radio Button | M | View | Options are - sellf (default) and others |
| Status of Remainder | Check Box | M | View | Options are – open, close, and cancelled |
| Company | Drop-Down | O | View | All existing companies in the system |
| Contact | Drop-Down | O | View | All existing contacts belonging to the selected company |

## Appendix – Close Remainder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Activity Date | Date | M | Close | Date selected from date picker |
| Activity Type | Drop-Down | M | Close | Validated entries are  Telephone / conversation  Email  Meeting |
| Description | Text | O | Close | Brief description |
| Status Change of Contact | Radio button | M | Close | Yes, No (default) |
| Change Status | Drop-Down | O | Close | Validated entries from Contact Status |

## Appendix - Activity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Activity Date | Date | M | Add / Edit | Date selected from date picker |
| Activity Type | Drop-Down | M | Add / Edit | Validated entries are  Telephone / conversation  Email  Meeting |
| Description | Text | O | Add / Edit | Brief description |
| Status Change of Contact | Radio button | M | Add | Yes, No (default) |
| Change Status | Drop-Down | O | Add / Edit | Validated entries from Contact Status |

## Appendix – Travel Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **M / O** | **Operation** | **Description** |
| Start Date / Time | Date and Time | M | Add / Edit | Date selected from date picker and time selected from drop down list |
| End Date / Time | Date and Time | M | Add / Edit | Date selected from date picker and time selected from drop down list |
| Destination | Drop-down | M | Add / Edit | Validate entries from Travel Destination |
| Description | Text | O | Add / Edit | Brief description |