**Slack Custom Integration**

**Technologies**

1. Asp.Net Core (MVC + WebAPI)
2. Angular 2
3. PostgreSQL
4. XUnit (server side testing framework)
5. Jasmine (client side testing framework)

**Login/Registration:**

Administrator will be created by the system. Administrator can add users and send out mail with email and password to the users. Users can login with those credentials and later can change password if they wish.

**Projects-Employee Management:**

Administrator can add ongoing projects to the system. Also he can assign team member and team leader to project the project. Team Leader of the project can view leave, scrum, task mails of his/her team members. Team Member can only view his own details.

**Automate Leave Application via Slack:**

Requirement is to make leave application request using slack. We need to design custom slack command for leave application and parameters will be passed to that command for applying leave. Parsing that command we will identify which users should get notification of leave application. And those users will get slack notification message with leave details and facility to Approve or Reject it. Leave command and its parameters are mentioned below.

**/leaves –** This command will be used to apply for leave with the following options and parameters. Different options specified with /leaves command will have different parameters

Format: /leaves {Command} [Parameters]

1. To Apply for leave: /leaves apply [Leave Type] [Reason] [From Date] [To Date]
2. To List Leaves: /leaves list [@user (for TL only)] TL can list leave of particular user

It will list all the leaves of current year.

1. To Cancel Leaves: /leaves cancel [Leave Id]
2. To Check status of leaves: /leaves status [@user (for TL only)] TL can check leave status of particular user, team members can check status of their own leave
3. To check balance leaves: /leaves balance will show user’s balance leaves from CL and SL
4. To get all available command’s help and format: /leaves help

Web interface will also be provided to view leave records.

**Automate Task Mail via Slack:**

Requirement is to send task mail using slack also, to perform all daily activities at one place. To perform this, we need to design custom command to start task mail and custom slack BOT to ask questions for the status of their work every day. BOT will be initiated by user who wants to write task mail.

**/taskmail –** This command will initiate Taskmail BOT who will ask following questions and repeat the same if user has multiple tasks to mention.

1. On which task you worked on today? (Task Name)
2. How many hours you spent on this task? (whole numbers or decimal)
3. What is the Status of this task?

(In progress, Completed, Roadblock)

1. Mention if any Comment/Roadblock.

Any comment for the task or roadblock

1. Do you have any more task to mention? (-yes or -no)

-yes will repeat point 1-5 again. And -no will consider that you are done with taskmail

**Automate Scrum/Stand up Meeting via Slack:**

Requirement is to conduct scrum meeting via slack. To perform this, we need to design command which will start the Scrum BOT. BOT will be programmed to ask set of questions to each member of the particular project team (in which its connected). Flow will be as follows:

1. Team leave will start BOT with **/startscrum** command
2. BOT will pick up name of one member and show him his yesterday’s TODO list and will ask him following questions
   1. What have you done yesterday?
   2. What you will be doing today?
   3. Any Roadblocks?

And the member to whom BOT is conversing with, will reply to respective questions. Once current member responds to last question, BOT will repeat point 2 until all members of the team are taken.

1. If someone is absent from the team, then any other member of that team can write “-leave” to inform that currently addressed member is not available.

Once BOT has asked to all team members, Scrum will end.

**Common Functions:**

For each of the above slack custom integrations, data will be stores in database and well as mail will be send to all responsible person as well as management of that team.

**Web Interface:**

For each of the above slack custom integrations, Web interface will be provided where by users can review their data of leave, scrum and taskmail.

**Deployment:**

Need to Ask for Hosting Details

**Future Enhancements:**

1. Ask employee joining date and after that check for which all type of leaves he is eligible for.
2. Rectify wrong data from web part
3. List task mail of user for specific date (either by TL or by employee)

**Database**

**Tables Name**

1. **Employee**
2. **Project**
3. **EmployeeProject**
4. **Leave**
5. **LeaveType**
6. **Task**
7. **TaskDetails**
8. **Scrum**
9. **ScrumQuestion**
10. **ScrumAnswer**

**Table fields name and table short description**

**Employee**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Key / characteristics** |
| Id | Integer | Primary Key |
| Name | Varchar(15) | not null |
| SlackName | Varchar(25) | not null |
| Email | Varchar(25) | not null |
| Role | Varchar(15) | not null |

**Project**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Key / characteristics** |
| Id | Integer | Primary Key |
| Name | Varchar(15) | not null |
| GroupName | Varchar(25) | not null |
| TeamleaderId | Integer | Foreign key (Employee table) |

**EmployeeProject**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Key / characteristics** |
| Id | Integer | Primary Key |
| ProjectId | Integer | Foreign Key(Project table) |
| EmployeeId | Integer | Foreign Key(Employee table) |

**Leave**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Key / Characteristics** |
| Id | Integer | Primary Key |
| EmployeeId | Integer | Foreign Key(Employee table) |
| LeaveType | Integer | Foreign Key(LeaveType table) |
| ApplyDate | DateTime | not null |
| FromDate | DateTime | not null |
| ToDate | DateTime | not null |
| JoiningDate | DateTime | not null |
| Reason | Varchar(Max) |  |
| Status | Varchar(10) | not null |
| ApproveRejectEmployeeId | Integer | Foreign Key(Employee table) |
| ApproveRejectDate | DateTime | not null |

**LeaveType**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Key / Characteristics** |
| Id | Integer | Primary Key |
| Type | Varchar(10) | not null |
| LeaveDays | Decimal | not null |

**Task**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Key / Characteristics** |
| Id | Integer | Primary Key |
| Date | DateTime | not null |
| EmployeeId | Integer | Foreign Key(Employee table) |
| TeamleaderId | Integer | Foreign Key(Employee table) |
| Projectid | Integer | Foreign Key(Project table) |

**TaskDetails**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Key / Characteristics** |
| Id | Integer | Primary Key |
| Description | Varchar(Max) | not null |
| Hours | Decimal | not null |
| Comment | Varchar(Max) |  |
| TaskId | Integer | Foreign Key(Task table) |

**Scrum**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Key / Characteristics** |
| Id | Integer | Primary Key |
| GroupName | Varchar(25) | not null |
| TeamLeaderId | Integer | Foreign Key(Employee Table) |
| ProjectId | Integer | Foreign Key(Project Table) |
| ScrumDate | Datetime | not null |

**ScrumQuestion**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Key / Characteristics** |
| Id | Integer | Primary Key |
| Question | Varchar(Max) | not null |
| CreatedAt | Datetime | not null |

**ScrumAnswer**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Key / Characteristics** |
| Id | Integer | Primary Key |
| Answer | Varchar(MAX) | not null (Answer or – leave) |
| ScrumId | Integer | Foreign Key(Scrum Table) |
| ScrumQueId | Integer | Foreign Key(ScrumQuestion Table) |
| EmployeeId | Integer | Foreign Key(Employee Table) |
| AnswerDate | Datetime | not null |