IndusTree - Factory Management Platform

Team Members: Nestor Lobo, Shreya Verma, Varsha Teratipally

Project Title: IndusTree (Factory Management Platform)

Project Summary:

A platform to aid in the management of daily workings in a factory. It enables a centralized management system of tasks such as tracking machine stock levels, request stock. It also provides a management system for the Human Resources department for creating employee and managing employee information and enabling employees to apply for leaves and claims, search for employees and edit personal information.

Project Requirements:

Business Requirements:			
ID	Requirement	Topic Area	User
BR – 001	Login email must be "*@riomhaire.com"	Authentication	All

User Requirements:			
ID	Requirement	Topic Area	User
UR – 001	Users logs into the application	Authentication	All
UR – 002	Creation of user profiles	Authentication	HR Manager
UR – 003	Users can view their profile	Profile	All
UR – 004	Users can edit their personal information	Profile	All
UR – 005(a)	User can search for and view the public profile of other employees	Search	All
UR – 005(b)	User can search for and view all employee information	Search	HR Manager

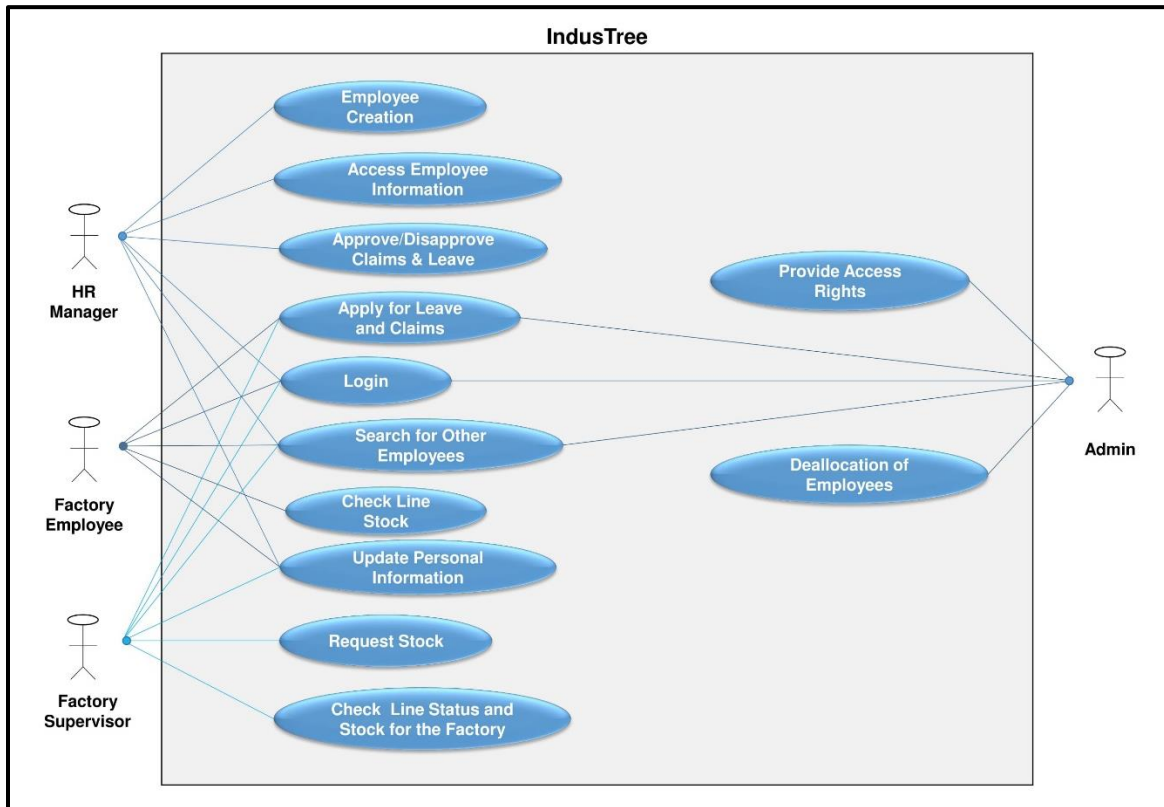| UR – 006(a)	Users can view their machine line status	Line Status	Factory Line Employee
UR – 006(b)	Users can view machine line statuses of all lines	Line Status	Factory Supervisor
UR – 007	Users can apply for leave	Leaves and Claims	All
UR – 008	Users can apply for claims	Leaves and Claims	All
UR – 009	Users can view and delete applied leave & claims	Leaves and Claims	All
UR – 010	Users can approve/disapprove the applied leaves and claims	Approvals Central	HR Manager
UR – 011	Users can request for stock	Line Status	Factory Supervisor
UR – 012	Users can sign out	Authentication	All
UR – 013	Users can reset their passwords	Authentication	All
UR -- 014	Users can deactivate the employee profile	Authorization	Admin
UR -- 015	Users can change employee access rights	Authorization	Admin

Functional Requirements:

ID	Requirement	Topic Area	User
FR – 001	System should generate a unique username and password for each employee	Authentication	All
FR – 002	System should authenticate the user based on login information and can differentiate the employee type	Authentication	All
FR – 003	System should show the live status of the machine line; refreshed every minute	Line Status	Factory Line Employee, Factory Supervisor

FR – 004	Users should be notified about the applied leaves and claims	Approvals Central	HR Manager
FR – 005	Users should be notified about the approval status of their leaves and claims	Leaves and Claims	All
FR – 006	System session is maintained for 30 minutes	User Session	All
FR – 007	System should be able to notify the user when stocks are running low	Line Status	Factory Line Employee, Factory Supervisor
FR – 008	Email the employee a link to reset their password	Authentication	All

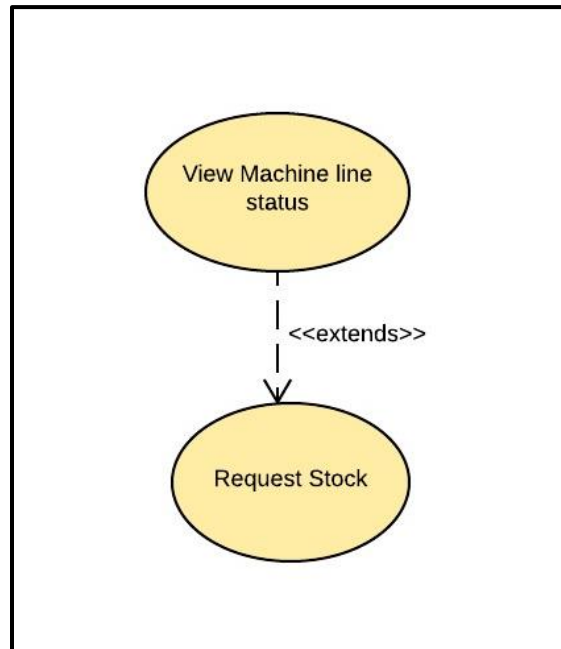| Non-Functional Requirements:			
ID	Requirement	Topic Area	User
NR – 001	Browser Independence	User Centered Design	All
NR – 002	User authentication should not take more than 5 seconds after valid credentials are entered	Latency	All
NR – 003	Search results should not take more than 10 seconds to compute after valid data is entered	Latency	All

Use Cases:

- **Actors:** Factory Employee, Factory Supervisor, HR Manager, Admin
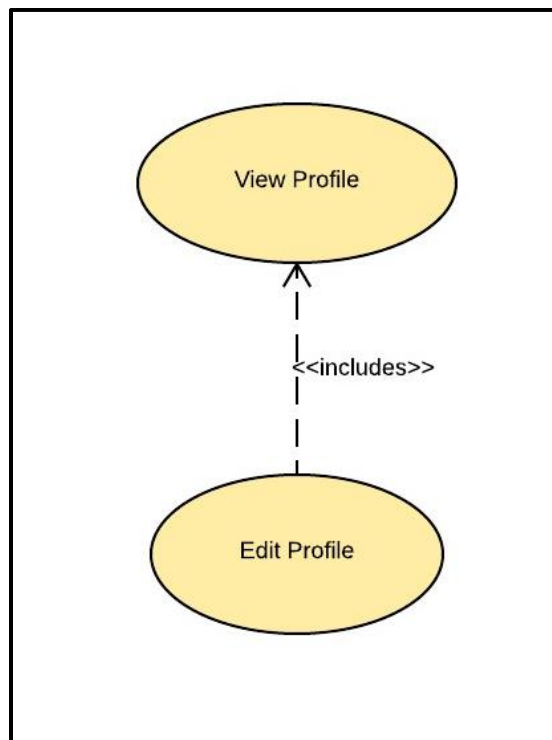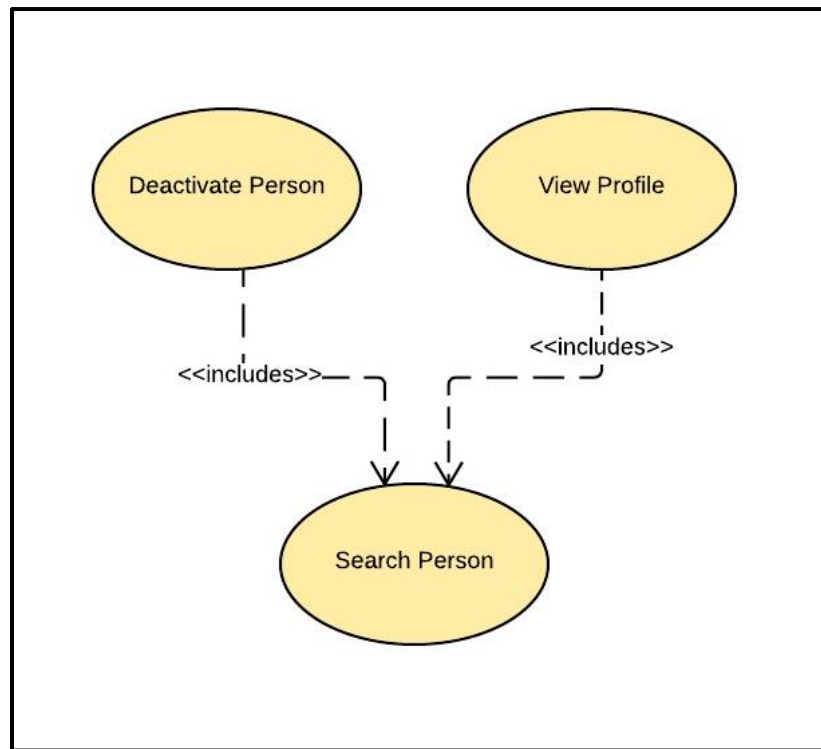
- **Use Case Overview:**

Sub Diagrams:

Sub Diagram 1

View Machine line status

<<extends>>

Request Stock

Sub Diagram 2

View Profile

<<includes>>

Edit Profile

Sub Diagram 3

Use Case Documents:

Use Case ID	UR-001
Use Case Name	User logs into the application
Description	User logs into the application by providing their username (email-ID) and password

Actors	All Affiliates		
Pre-conditions	User should have an existing profile with a riomhare.com email-ID as a username		
Post-conditions	Login Successful: The Employee homepage window is displayed. Users can now view their details and logs		
Frequency of Use	Daily		
Flow of Events	S.No	Actor Action	System Response
	1	Visit the login page.	The username and password fields get displayed for entry.
	2	Enter their credentials and click on the Log-in button.	The system authenticates the user based on information provided by them and lets the user access the Employee homepage on a successful login.
Exceptions	An error message is displayed if the user's credentials are not authenticated or if mandatory fields are left blank.		
Notes and Issues			
Developer Notes			

Use Case ID	UR-002
Use Case Name	Creation of User profiles
Description	The HR-manager is responsible for creating profiles of new employees by generating a unique username as per the details entered

Actors	HR-Manager	
Pre-conditions	User must be logged in as an HR Manager	
Post-conditions	A new user account gets created successfully	
Frequency of Use	Frequently	
Flow of Events	**S.No**	**Actor Action**	**System Response**

Flow of Events	S.No	Actor Action	System Response
	1	Click on the link to the Employee Central.	The Employee Details fields to create a new employee are displayed.
	2	Click on the Create Employee button.	The system authenticates the information provided by the actor and adds the new employee to the company's database.
Exceptions	When mandatory fields are left vacant, an error message is displayed.		
Notes and Issues			
Developer Notes			

Use Case ID	UR-003
Use Case Name	Users view their profile
Description	Users can view their profile by clicking on the My Profile section

Actors	Riomhaire Affiliates		
Pre-conditions	User must be logged in		
Post-conditions	User's Profile page is displayed		
Frequency of Use	Frequently		
Flow of Events	S.No	Actor Action	System Response
	1	Click on the My Account button.	The system redirects the user to a page where they can view their account details.
Exceptions			
Notes and Issues			
Developer Notes			

Use Case ID	UR-004
Use Case Name	Users can edit their personal information
Description	Users can modify their account and access information

Actors	Riomhaire Affiliates		
Pre-conditions	User must be logged in		
Post-conditions	User can view their updated profile		
Frequency of Use	Frequently		
Flow of Events	**S.No**	**Actor Action**	**System Response**
	1.	Click on the My Account button.	The system redirects the user to their account information page.
	2.	Click on the Update profile button.	The system redirects the user to a page where they can update their personal information.
	3.	Enter the new details and clicks on the Update button.	The system validates that all the required fields have realistic data entered in them and the user's password. It then saves all the changes requested and updates the employee database with the new information. The newly updated profile is now displayed to the user.
Exceptions	When mandatory fields are left vacant, an error message is displayed.		
Notes and Issues			
Developer Notes			

Use Case ID	UR-005(a)
Use Case Name	User can search for and view the public profile of other employees
Description	Users can search for contact information about other employees

Actors	Riomhaire Affiliates		
Pre-conditions	User must be logged in		
Post-conditions	User can view the public profile of the fellow employees		
Frequency of Use	Frequently		
Flow of Events	**S.No**	**Actor Action**	**System Response**
	1.	Click on the Employee Search link.	The system redirects the user to a page where they can search for other employees. The Name and Department fields are displayed.
	2.	Enter the details of the employee to be searched for and click on the Search button.	The system validates that all the required fields have data and checks the database for all employees whose details fit the required description. It then displays the relevant employees.
	3.	Click on the required search result	The system redirects the user a page with the corresponding employee's public profile information.
Exceptions	When all fields are left vacant, an error message is displayed.		
Notes and Issues			
Developer Notes			

Use Case ID	UR-005(b)
Use Case Name	User can search for and view all employee information
Description	Users can search for all available information about all employees

Actors	HR-Manager	
Pre-conditions	User must be logged in as an HR Manager	
Post-conditions	User can view proof all employees	
Frequency of Use	Frequently	
Flow of Events	**S.No** / **Actor Action** / **System Response**	

Flow of Events	S.No	Actor Action	System Response
	1.	Click on the Employee Search link.	The system redirects the user to a page where they can search for all employees. The Name and Department fields are displayed.
	2.	Enter the details of the employee to be searched for and click on the Search button.	The system validates that all the required fields have data and checks the database for all employees whose details fit the required description. It then displays the relevant employees.
	3.	Click on the required search result	The system redirects the user a page with the corresponding employee's public profile information.

Exceptions	When all fields are left vacant, an error message is displayed.
Notes and Issues	
Developer Notes	

Use Case ID	UR-006(a)
Use Case Name	Users can view their machine line status
Description	Users can view the status of all the lines administered/operated by their, the Item ID of the units currently processing and the stock quantity of a type

Actors	Factory Line Employee		
Pre-conditions	User must be logged in		
Post-conditions	User can view their detailed machine line status		
Frequency of Use	Frequently		
Flow of Events	S.No	Actor Action	System Response
	1.	User visits the Employee homepage and clicks on the My Lines link.	The system redirects the user to a page where they can view the functional status of the line, the Item ID of the units currently processing and the stock quantity of the machine type.
Exceptions			
Notes and Issues			
Developer Notes			

Use Case ID	UR-006(b)
Use Case Name	Users can view the machine line status of all the lines
Description	Users can view the status of all the lines, the Item ID of the units currently processing and the stock quantity of a type

Actors	Factory Supervisor		
Pre-conditions	User must be logged in as a Factory Supervisor		
Post-conditions	User can view the detailed machine line status of all lines		
Frequency of Use	Frequently		
Flow of Events	S.No	Actor Action	System Response
	1.	User visits the Employee homepage and clicks on the Line Statuses link.	The system redirects the user to a page where they can view, for all the lines, the functional status of the lines, the Item ID of the units currently processing and the stock quantity of the machine type.
Exceptions			
Notes and Issues			
Developer Notes			

Use Case ID	UR-007
Use Case Name	Users can apply for leave
Description	Users can apply for leaves by entering the start date, end date and type of leave

Actors	Riomhaire Affiliates	
Pre-conditions	User must be logged in	
Post-conditions	The user is intimated about their successful leave application with a corresponding leave ID number and their request is logged	
Frequency of Use	Frequently	
Flow of Events	**S.No**	**Actor Action**	**System Response**

Flow of Events	S.No	Actor Action	System Response
	1.	Click on the My Leave and Claims link.	The system redirects the user to a page where they can apply for leaves by entering the start date, end date and the notes
	2.	Enter fields with relevant information about the leave request and click the Submit button.	The system validates that realistic data has been entered and enters the leave request information into the Leave Database. It then reloads the My Leave and Claims page for the user with a notification at the top of the page with a generated leave id. This notification is also displayed on the employee's Dashboard.
Exceptions	When mandatory fields are left vacant or have invalid data, an error message is displayed.		
Notes and Issues			
Developer Notes			

Use Case ID	UR-008
Use Case Name	Users can apply for claims
Description	Users can apply for claims by entering the item ID, the purchase date and the amount to be reimbursed

Actors	Riomhaire Affiliates		
Pre-conditions	User must be logged in		
Post-conditions	The user is intimated about their successful claim application with a corresponding claim ID number and their request is logged		
Frequency of Use	Frequently		
Flow of Events	S.No	Actor Action	System Response
	1.	Click on the My leaves and Claims link.	The system redirects the user to a page where they can apply for claims for any item purchased lines by entering the Item type, Item ID, purchase date and the amount to be reimbursed.
	2.	Enter fields with relevant information about the claim request and click the Submit button.	The system validates that realistic data has been entered and enters the claim request information into the Claim Database. It then reloads the My Leave and Claims page for the user with a notification at the top of the page with a generated claim id. This notification is also displayed on the employee's Dashboard.
Exceptions	When mandatory fields are left vacant or have invalid data, an error message is displayed.		
Notes and Issues			
Developer Notes			

Use Case ID	UR-009
Use Case Name	Users can view and delete applied leave & claims
Description	Users can view the status of their previous requests and delete applied requests

Actors	Riomhaire Affiliates		
Pre-conditions	User must be logged in		
Post-conditions	The user can view their request statuses and their deleted requests are removed from the databases		
Frequency of Use	Frequently		
Flow of Events	S.No	Actor Action	System Response
	1.	Click on the Dashboard link.	The system redirects the user to the dashboard page where all notifications are shown. The system populates this page with the user's requests from the requests databases.
	2.	Click on the Delete link for the request to be deleted.	The system deletes the corresponding request from the request database and refreshes the dashboard page without the corresponding deleted request.
Exceptions			
Notes and Issues			
Developer Notes			

Use Case ID	UR-010
Use Case Name	Users can approve/disapprove the applied leaves and claims
Description	Users can approve the leaves and claims as requested by other employees

Actors	HR Manager		
Pre-conditions	User must be logged in as an HR Manager		
Post-conditions	User can approve the applied requests and the corresponding employee is notified		
Frequency of Use	Frequently		
Flow of Events	**S.No**	**Actor Action**	**System Response**
	1.	Click on the Approval Central link.	The system redirects the user to a page where they can view all the requests for leaves and claims appealed by the employees.
	2.	Click the link corresponding to a specific request.	The system redirects the user to a page where they can view more details about the specifics of the request of the employee.
	3.	Click the Approve/Disapprove button.	The corresponding request is marked for approval/disapproval in the database. The system then archives the request from the approval center and notifies the user that made the request of its approval/disapproval.
Exceptions			
Notes and Issues			
Developer Notes			

Use Case ID	UR-011
Use Case Name	Users can request for stock
Description	Users can request for any stock to be replenished in the selected machine

Actors	Factory Supervisor		
Pre-conditions	User must be logged in as a Factory Supervisor		
Post-conditions	The user's request for stock is logged		
Frequency of Use	Frequently		
Flow of Events	S.No	Actor Action	System Response
	1.	Click on the Line Statuses link.	The system displays the status of all the lines (whether working, malfunction, stock-off or low), their Line manager, the Item ID of the units currently processing and the stock quantity of a type.
	2.	Click the corresponding Request Stock link.	The system redirects the user to a page where they can enter the details of the unit to be re-stocked.
	3.	Click Submit Request button.	The system validates the details provided by the user and notifies the supply room about the requested stock on a successful submission of the appeal. The request is logged in the database.
Exceptions	When mandatory fields are left vacant, an error message is displayed.		
Notes and Issues			
Developer Notes			

Use Case ID	UR-012
Use Case Name	Users can sign out
Description	Users can sign out their session by clicking on the Sign Out link

Actors	Riomhaire Affiliates		
Pre-conditions	User must be logged in		
Post-conditions	User is redirected to the login page.		
Frequency of Use	Rarely		
Flow of Events	S.No	Actor Action	System Response
	1.	User clicks on the Sign Out link.	The system ends the user's active session and redirects the user to the Login page.
Exceptions			
Notes and Issues			
Developer Notes			

Use Case ID	UR-013
Use Case Name	Users can reset their password
Description	Users are emailed a link to reset a forgotten password

Actors	Riomhaire Affiliates		
Pre-conditions	User should have a valid riomhaire.com email ID for the reset password link to be sent		
Post-conditions	User is sent a link to reset their password		
Frequency of Use	Rarely		
Flow of Events	S.No	Actor Action	System Response
	1.	Click the Forgot Password link.	The system redirects the user to a page where they can enter their email-ID. The system validates the email and sends a link to the email provided for resetting the password.
	2.	Click the reset password link in the received email.	The system redirects the user to a window where they can update their new password.
	3.	Click the Submit button.	The system validates that whether the password has been entered twice and that they match. The system updates the users password in the database to the newly entered password. The system now redirects the user to the login page again.
Exceptions	When mandatory fields are left vacant, an error message is displayed.		
Notes and Issues			
Developer Notes			

Use Case ID	UR-014
Use Case Name	Users can deactivate the employee profile
Description	Users can deactivate an employee's profile if the employee is no longer working in the company

Actors	Admin		
Pre-conditions	User must be logged in as an Admin		
Post-conditions	The corresponding profile is deactivated		
Frequency of Use	Rarely		
Flow of Events	S.No	Actor Action	System Response
	1.	Click on the Access Central Link	The system redirects the user to a page where they can view all the employees in a department along with their role and access rights.
	2.	Click on a specific Employee	The system redirects to the employee profile where the user can deactivate the profile.
	3.	Click on the Deactivate button.	The system updates the active Flag in the user table effectively deactivating the profile. The system then redirects the user to the access central page.
Exceptions			
Notes and Issues			
Developer Notes			

Use Case ID	UR-015
Use Case Name	User can change access rights for employees
Description	User can change the role of the employee as per company requirements

Actors	Admin		
Pre-conditions	User must be logged in as an Admin		
Post-conditions	The corresponding user's access rights are modified		
Frequency of Use	Rarely		
Flow of Events	S.No	Actor Action	System Response
	1.	Click on the Access Central Link	The system redirects the user to a page where they can view all the employees in a department along with their role and access rights. The view has links above to switch to other departments.
	2.	Click on a specific Employee	The system redirects to the employee profile where the user can change the access rights.
	3.	Set the required options in the drop-down menus and Click the Update button.	The system updates the role in the user table. The system then refreshes the employee profile page with the new roles and displays a notification message of the update. The system then notifies the corresponding employee that their role has been changed.
Exceptions			
Notes and Issues			
Developer Notes			

Data Storage:

The Project uses a MYSQL database. JDBC is used to connect with the Java driver from MYSQL to update and retrieve data from the database. The interaction with the database is done with dbConnection() which automates queries and retrieves the data.
The dbConnection class has functions with various methods to get and save updates to the databases for Employee Profiles, Machine Line Statuses and Employee requests for leaves and claims. These methods perform insert, delete and update operations on various tables (some of which are tables for Employees, Machine Types and Requests).
The dbConnection class is constructed in a way to reduce the amount of changes required in the program to modify the database.

Classes:

User class to store the information regarding the user.
- userName (unique)
- password(stored in encrypted form)
- Role (FactoryEmployee / FactoryManager / HRManager)
- lastName
- firstName
- middleName

Employee class to store the information regarding the employee.
- employeeId (unique)
- designation (designation of the employee)
- department (department which the employee is in)
- emailAddress (employee contact address)
- dateOfJoining (date on which the employee joined)
- workContact
- managerId (employeeId of the manager)

EmployeeLeave class to store the information regarding the employee leave.
- employeeLeaveId (unique)
- leaveType (CASUAL/OPTIONAL)
- leaveStatusType(APPLIED/PENDING/APPROVED)
- leaveappliedDate(date on which the leave has been applied)
- leaveapprovedDate (date on which the leave has approved)
- leaveStartDate (startdate of the leave)
- leaveEndDate (enddate of the leave)
- notes (comments)

EmployeeClaim class to store the information regarding the employee claim.
- employeeClaimId (unique)

- claimItemType (ItemTypes that can be claimed)
- purchaseDate (Date on which the Item has been bought)
- responseDate (Date on which the Item has been approved)
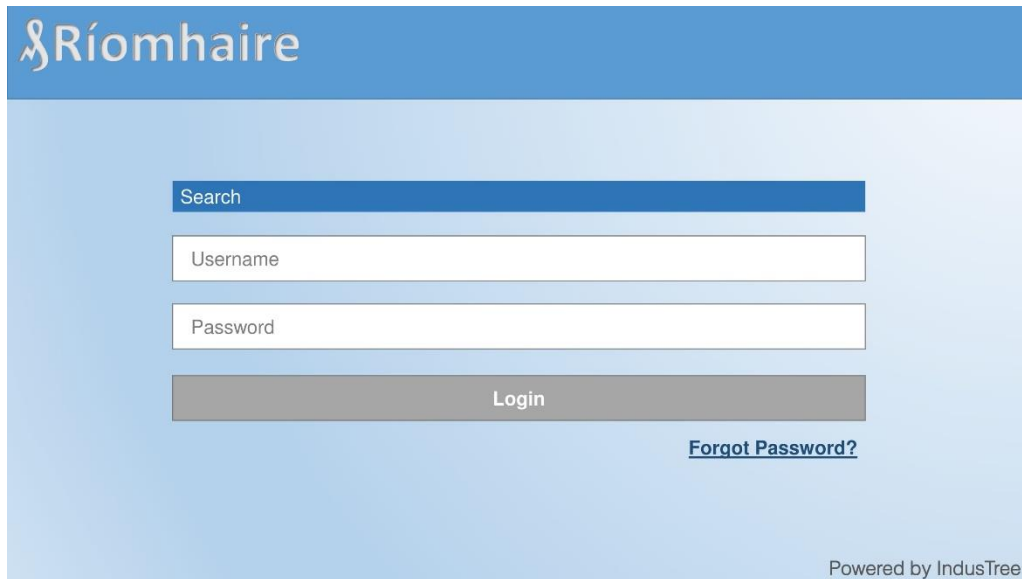- amount (amount of the item)
- claimStatus(APPLIED/APPROVED)

WorkingMachineLineStatus class to store the information regarding the machine line status
- workingMachineLineStatusId (unique)
- stock (stock number of the line)
- firstInLine (the product number of the first product in line)
- lastInLine(the product number of the last product in line)
- machineType (CASING/SCREEN/BATTERY/ELECTRONICS)

UI Mockups

All Users:

Login Page

Forgot Password Page

Dashboard

Ríomhaire

My Account   Sign Out

Dashboard

My Line

Employee Search

My Leave & Claims

Notifications

- ❖ You have applied for Claim 4156 (Screwdriver) Delete
- ❖ Your leave application 124 for 12/25/17 has been approved

Powered by IndusTree

Employee Search

Ríomhaire

Dashboard

My Line

Employee Search

My Leave & Claims

Employee Search

Employee Name	Role	Department	Contact Information
James Acaster	Intern	Manufacturing	+4-568-382-7742

Email
james.acaster@riomhaire.com

Powered by IndusTree

Application for Leave & Claims

Ríomhaire

My Account   Sign Out

Dashboard

My Line

Employee Search

My Leave & Claims

Claims

Item Name

Item ID

Purchase Date

Amount

Apply for Claim

Leave

Start Date

End Date

Note

Apply

Ríomhaire

My Account   Sign Out

Dashboard

My Line

Employee Search

My Leave & Claims

Successfully Applied for Screwdriver. Your Claim ID is 4256.

Claims

Item Name

Item ID

Purchase Date

Amount

Apply for Claim

Leave

Start Date

End Date

Note

My Account

Factory Employee:

Line Status

HR Manager:

Approval page for Leave and Claims

Ríomhaire

My Account   Sign Out

Dashboard	
Approval Central	
Employee Creation	
Employee Search	
My Leave & Claims	

Approval Central

Employee Name	Item Name	Item ID	Purchase Date	Amount	Approved
James, A	Screwdriver	4598	1/22/17	55.00 $	☐
Donna, P	Stapler	522	1/11/17	12.50 $	■

Powered by IndusTree

Ríomhaire

My Account   Sign Out

Dashboard	
Approval Central	
Employee Creation	
Employee Search	
My Leave & Claims	

Pending Approvals

Employee Name	Item Name	Item ID	Purchase Date	Amount
James, Acaster	Screwdriver	4598	1/22/17	55.00 $

Approve	Disapprove

Powered by IndusTree

Create a New Employee

ᕠRíomhaire

Dashboard

Approval Central

Employee Central

Employee Search

My Leave & Claims

Employee Central

First Name*

Last Name*

Middle Initial

Contact Number*

Date of Birth*

Department*

Create Employee

Factory Supervisor:

Factory Line Statuses

Admin:

Modify Access Rights

Class Diagram

HRManager
- + employeeList:List<Employee>
- + toApproveLeaves:List<EmployeeLeave>
- + toApproveClaims:List<EmployeeClaim>

EmployeeLeave
- employeeId: int
- leaveType : int
- leaveAppliedDate : Date
- leaveApprovedDate : Date
- leaveStartDate : Date
- leaveEndDate : Date
- notes :String
- leaveStatus :int

- + getLeaveStatus(): LeaveAndClaimStatus
- + setLeaveStatus(LeaveAndClaimStatus leaveStatus):int
- + getnumOfLeavesApplied():int
- + daysBetween(Date startDate, Date endDate):int
- + getEmployeeId():int
- + setEmployeeId(int employeeId):void
- + getLeaveType():String
- + setLeaveType(LeaveTypes leaveType):void
- + getLeaveAppliedDate():Date
- + setLeaveAppliedDate(Date leaveAppliedDate):void
- + getLeaveApprovedDate():Date
- + setLeaveApprovedDate(Date leaveApprovedDate):void
- + getLeaveStartDate():Date
- + setLeaveStartDate(Date leaveStartDate):void
- + getLeaveEndDate():Date
- + setLeaveEndDate(Date leaveEndDate):void
- + getNotes():String
- + setNotes(String notes):void

User
- - userId:int
- - firstName :String
- - lastName:String
- - middleName:String
- - dateOfBirth: Date
- - userName: String
- - password: String
- - emailAddress: String
- - userRoleId: String
- - address :String
- - activeFlag: boolean

- + getAddress() : String
- + setAddress(String address) :void
- + getUserRole() : String
- + setUserRole(UserRoles userRole):void
- + getUserRoleId() : int
- + setUserRole(UserRoles userRole):void
- + getUserRoleId(): String
- + setUserRoleId(int userRoleId):void
- + getUserId(): int
- + setUserId(int userId):void
- + getFirstName(): String
- + setFirstName(String firstName):void
- + getMiddleName(): String
- + setMiddleName(String middleName):void
- + getLastName() : String
- + setLastName(String lastName):void
- + getDateOfBirth() : Date
- + setDateOfBirth(Date dateOfBirth):void
- + getUserName() : String
- + setUserName(String userName):void
- + getPassword() : String
- + setPassword(String password):void
- + getEmailAddress(): String
- + setEmailAddress(String emailAddress):void
- + getActiveFlag(boolean activeFlag):void
- + getActiveFlag()

EmployeeClaim
- - employeeId: int
- - claimItemType: String
- - amount :int
- - purchaseDate: Date
- - appliedDate: Date
- - approvedDate : Date
- - claimStatus: String

- + getEmployeeId(): int
- + setEmployeeId(int employeeId):void
- + getClaimItemType(): String
- + setClaimItemType(String claimItemType):void
- + getAmount(): int
- + setAmount(int amount): void
- + getPurchaseDate(): Date
- + setPurchaseDate(Date purchaseDate): void
- + getAppliedDate(): Date
- + setAppliedDate(Date appliedDate): void
- + getApprovedDate(): Date
- + setApprovedDate(Date approvedDate): void
- + getClaimStatus(): String
- + setClaimStatus(String claimStatus): void

Employee
- - employeeId : String
- - userId : int
- - designation :String
- - position : String
- - emailAddress :String
- - dateOfJoining :Date
- - workContact: String
- - department :String
- - managerId :int

- + getEmployeeId() :String
- + setEmployeeId(String employeeId):void
- + getUserId() :int
- + setUserId(int userId):void
- + getDesignation():String
- + setDesignation(String designation):void
- + getEmailAddress():String
- + setEmailAddress(String emailAddress):void
- + getDateOfJoining():Date
- + setDateOfJoining(Date dateOfJoining):void
- + getWorkContact():String
- + setWorkContact(String workContact):void
- + getManagerId():int
- + setManagerId(int managerId):void

WorkingLineStatus
- - int lineId
- - int userId
- - int stock
- - String machineType
- - int firstInLine
- - int lineStatusId
- - String lineStatus

- + getLineStatus():String
- + setLineStatus(String lineStatus):void
- + getLineStatusId(): int
- + setLineStatusId(int lineStatusId): void
- + getLineId(): int
- + setLineId(int lineId): void
- + getUserId(): int
- + setUserId(int userId): void
- + getStock(): int
- + setStock(int stock): void
- + getFirstInLine(): int
- + setFirstInLine(int firstInLine): void

EmployeeClaims
- - employeeId:int
- - employeeClaims:List<EmployeeClaim>

- + getEmployeeId(): int
- + setEmployeeId(int employeeId): int
- + getEmployeeClaims():List<EmployeeClaim>
- + setEmployeeClaims(List<EmployeeClaim> employeeClaims):void
- + getAppliedClaims():List<EmployeeClaim>
- + getApprovedClaims():List<EmployeeClaim>

Admin
- + DeactivateUser(int userId) : void
- + ChangeUserRole(int userId, int userRoleId):void

EmployeeLeaves
- - employeeId:int
- - employeeLeaves:List<EmployeeLeave>

- + setEmployeeId(int employeeId): void
- + getEmployeeLeaves():List<EmployeeLeave>
- + setEmployeeLeaves(List<EmployeeLeave> employeeLeaves): void
- + getNumOfCasualLeavesApplied(): int
- + getNumOfCasualLeavesApproved(): int
- + getNumOfSickLeavesApplied(): int
- + getNumOfSickLeavesApproved(): int
- + getLeavesApplied():List<EmployeeLeave>
- + getLeavesApproved():List<EmployeeLeave>

FactoryManager
- + List<Employee> employeeList
- + List<WorkingLineStatus> workingLineStatuses

DBConnection
- + validateLoginUser(String userName, String password): User
- + getEmployeeProfile(int userId): Employee
- + getWorkingMachineLineStatus(int userId): WorkingLineStatus
- + getEmployeeLeaves(int employeeId):List<EmployeeLeaves>
- + getEmployeeClaims(int employeeId):List<EmployeeClaim>
- + getWorkingMachineLineStatuses(int employeeId):List<WorkingLineStatus>
- + WorkingLineStatus(int employeeId):EmployeeLeaves
- + getemployeeClaims(int employeeId):EmployeeClaims
- + saveEmployeeProfile(Employee employeeProfile):void
- + saveAppliedLeave(EmployeeLeave employeeLeave):void
- + saveAppliedClaim(EmployeeClaim employeeClaim):void
- + saveApprovedLeaves(List<EmployeeLeave> employeeLeaves): void
- + saveApprovedClaims(List<EmployeeClaim> employeeClaims):void
- + searchForEmployeeByName(String name):List<Employee>
- + searchForEmployeeByDepartmentAndName(String name, String Department):List<Employee>
- + DeactivateUser(int userId) :void
- + ChangeUserRole(int userId, int userRoleId):void
- +cancelEmployeeLeave(List<EmployeeLeave> employeeLeave):void
- +cancelEmployeeClaim(List<EmployeeClaim> employeeClaim)

<<Interface>>
IDBConnection

LoginController
- + validateLoginUser(String userName, String password)

EmployeeProfileController
- + ViewProfile(int employeeId):Employee
- + SaveProfile(Employee employee) : void
- + GetEmployeeLeaves(int employeeId):EmployeeLeaves
- + GetEmployeeLeaveRequest()
- + GetEmployeeClaimForm()
- + SaveEmployeeLeaves(EmployeeLeaves employeeLeaves): void
- + GetEmployeeClaims(int employeeId): List<EmployeeClaims>
- + SaveEmployeeClaims(EmployeeClaims employeeClaims) :void
- + GetMachineLineStatus(int employeeId):WorkingLineStatus
- + GetMachineLineStatusForManager(int employeeId):Line<WorkingLineStatus>
- + SaveApprovedLeaves(List<EmployeeLeaves> employeeLeaves) : void
- + SaveApprovedClaims(List<EmployeeClaims> employeeLeaves): void
- + SaveDeclinedLeaves(List<EmployeeLeaves employeeLeaves) : void
- + SaveDeclinedClaims(List<EmployeeLeaves employeeLeaves) : void
- + CancelAppliedLeaves(List<EmployeeLeaves> employeeLeaves) : void
- + CancelAppliedClaims(List<EmployeeLeaves employeeLeaves) : void
- + DeactivateUser(int userId) :void
- + ChangeUserRole(int userId, int userRoleId):void

SearchController
- SearchEmployeeProfile(string name, string department)