* **PRINTLOK**

Printlok is an Employee tracking application, where we can track the employee’s location. When the employee is on leave we do not track them.

I worked as backend developer in a team of 2 members. I was responsible to create REST API’s.

I worked in ,

* Organization module
* User module
* Tracking module
* Attendance module

An employee with Role SuperAdmin and HR can register based on particular organization. Once the user is registered, An email or sms will be sent to that user based on Authentication type.

The user can login to the application and create multiple users and assign Roles, create Location groups, deactivate users, etc.

All the employee’s location will be tracked except SuperAdmin.

We store the employee Workplace, and capture the employees location details on daily basis based on the login and logout. We do not track the employee after the logout time. When the employee’s leave is Approved, we do not track them.

We record the employee is Present or Absent based on the location details.

Tracking your employees is the best way to ensure they are working when they should be, this is one of the advantages

**Environment:** Spring boot Rest API, Hibernate, Maven, MySQL, Postman, STS

**Role:** Analysis, Design, Implementation, Development and Team Lead

In future, we are going to implement parent tracking their children’s functionality.

* **PROJECT MANAGEMENT**

Project Management is an application which involves the planning and organization of a company's resources to move a specific task towards completion.

I worked in,

* Employee module
* Organization module
* Project module
* Task module
* Task Group module
* Approvers Policy
* Common tasks
* Project stage
* Task approvers
* Dashboard module

This application has multiple roles,

Here we can manage employees, projects, approvers, task, task group, timesheets. We can assign employees to project and tasks, assign tasks to project, etc.

**Environment:** Spring boot Rest API, Hibernate, Maven, MySQL, Postman, STS

**Role:** Analysis, Design, Implementation and Development, Team Lead

* **EMAIL APPLICATION**

Email Application is a web-based application and in cloud that maintains overall functionality in the application

I worked in,

* Compose mail
* Reply, Forward mail
* Inbox module
* Sent module
* Draft module
* Trash module
* Unread mails module
* Important mails module
* Spam module
* Signature module
* Folder module

We have used java mail API to send, receive, reply, relply all with attachment, forward, delete, etc.

We can create folders, move mails from one folder to another.

We can create signatures, etc.

**Environment:** Spring boot Rest API, Hibernate, Maven, MongoDB, Postman, STS

**Role:** Analysis, Design, Implementation and Development, Team Lead

* **LEAVE MANAGEMENT**

Leave Management allows us to track the leaves taken by the employees in an organization.

I have worked on,

* Employee module
* Organization module
* Leave balance module
* Leave request module
* LeaveType module
* Leave Withdraw module
* WorkFromHome module
* Business Travel module
* Dashboard module

With this application, we can apply leave, work from home, business travel.

We can withdraw leave, this can be approved or denied. We can view the Leave Request, Leave balance, Workfrom home, business travel.

We can apply for permission, cancel permission.

We can edit the profile and upload image, etc.

**Environment:** Spring boot Rest API, Hibernate, Maven, MySQL, Postman, STS

**Role:** Analysis, Design, Implementation and Development, Team Lead

* **EYETRAC - VEHICLE TRACKING WEB APPLICATION**

Eyetrac is the GPS tracking software developed for Vehicle tracking and supports

more protocols. With Help of Eyetrac you can monitor your devices in real-time, It

Provides instant web notifications in cases of geo-fencing, over speed, Text Message,

Motion, Ignition. Eyetrac supports simple location history, trip and summary reports.

You can view data directly in the web and also export and download an Excel file. History can also be projected on the map providing visual representation.

**Environment:** Java, Spring MVC Framework and SQL, HTML, CSS, Javascript.

**Role:** Analysis, Design and Implementation

* **NURTURE FARM**

Nurture farm is an application where farmers can buy and sell their crops online, make payment, chat etc.

I have worked on

Listing module.

Payment using stripe

We can sell an item based on category, buy an item

**Environment:** Java GRPC, GRAPHQL, AWS Code Commit

**Role:** Backend Development

* **LOCATO**

Locato is a mobile application that enables local vendors to register and customers can track their location details so that any customer can easily reach them easily.

I have worked on

* User
* Business
* Category
* Items

The user can register as a Normal user and Vendor,

Vendors can add the Business details, category details, items details.

Normal users can directly visit the location based on the location provided in the app.

So the you doesn’t need to search for the location of what he needed.

**Environment:** Spring boot Rest API, Hibernate, Maven, MySQL, Postman, STS

**Role:** Analysis, Design, Implementation and Development, Team Lead

* **Integration Designer and Developer**

**Domain: Retail Domain**

**Description**

This project is intended to provide secured end to end integration solutions between Oracle Fusion, Legacy File systems and other Custom applications

1. Customer Creation(REST API based)
2. Sales order (FBDI - File Based Data Import)(Bulk import)
3. Account Payable(FBDI - File Based Data Import)(Bulk import)
4. Account Receivable(FBDI - File Based Data Import)(Bulk import)
5. General Ledges(FBDI - File Based Data Import)(Bulk import)

We worked on the Sales Order,

Sales Orders are handled in Order Management, which falls under SCM (Supply Chain Management).

In order to create a Sales Order, we checked if there is a customer, and if there is no customer, we created a customer.

Next, we checked if the item is in the inventory or not.We create a Sales Order if the item exists.

Whenever we create a customer, there is something called a Party and Account. A Party can have multiple Accounts. Party can be a Individual Person(consumer/end-user) or an Organization(customer)

Every Party will have a partyId, partyNumber, partySiteId, partySiteNumber

**ERP Party and Account creation via REST and SOAP**

Using REST and SOAP services, we have created an ERP party and account.

Using the ERP Adapter, we make a REST call, we create the Party and Address records using a payload. After the request has been created, we receive a response payload that contains the partyId and addressId.

In the CustomerAccount creation SOAP payload, we must pass the partyId and addressId to the partySiteId field. When executed, a CustomerAccount record will be created with the CustomerAccountSite.

W

CustomerAccountProfile: ?

**Integration Flow**

The entire integration is a scheduled based.

1. We will fetch a list of files from the FTP server provided by the customer and we are iterating it.
2. If the file exists, then download the file to an internal local storage.

Then we insert the fileMetaData like fileName, filePath, instanceId, sourceSystem, errorReason if error exists into the custom table called File\_Header(DBAAS). Once inserted, fileId gets created.

This file also have a status, once we read the file and once we create the records successfully in the Saas, then we update the status as **Success**. If few records are successful and few records are not then we update the status as **Partial**. If all the records are failed, then we update the status as **Error.**

1. Now weare going to read the file in segments, In the 1st Round, the stage file by default will read 200 records.
2. Then we are inserting these records into Staging\_Table(DBAAS).
3. Whatever customer numbers we got from the file, we are checking against a MasterTable and this MasterTable is populated by another scheduled integration which will get from the SAAS to here.

[Now we have to check if the customers that we got in the file are completely new or already exists.

For that, we have written a separate scheduled integration, for every 15 minutes this integration will go and query the BIP Report

BIP Report is a place holder where we can write SQLs,

1. Write a SQL to get Party and Account details(Join Party and Account table)
2. Expose this SQL as SOAP webservice

We will create another scheduled integration and call this SOAP webservice using SOAP Adapter.

Write to another customer MasterTable(ETL\_CUST\_DATA\_TABLE)

Get customer MasterData(all the customer party account, numbers ids) ---> Staging Master Table

Now we will compare the Staging\_Table(DBAAS) and ETL\_CUST\_DATA\_TABLE, if the customer already exists, it will mark the record as N.]

1. Now we are moving the file to Archive Directory, if the records are not present already.

Now we are fetching the data from StagingTable which are new, Now we fetch each record and I am making call to CreatePartyService(REST call), CreateAccountService(SOAP call), CreateProfileService.(**ERP Party and Account creation via REST and SOAP**)

**BIP Report:**

Tools -> Reports and Analytics -> Browse Catalog -> Create -> DataModel ->

Create a SQL Query

Give a Name, Select DataSource(ApplicationDB\_CRM), Type of SQL(Standard SQL), SQL Query.

**Customer:**

Receivables -> Billing -> Customer

**Receivables:** Receiving money, after you sell product to a customer. We are billing the customer here.

**Payables:** List of items or invoice you give, Invoices, etc