DATA MIGRATION FRAMEWORK AND TOOL FOR INVOICE MANAGEMENT

An Internship Report for

CSA 652 Industry Internship/Software Project

Credits: 16 Credits

Submitted in partial fulfilment of Masters Degree

Master of Computer Application (MCA)

by

SHAIKH SAFWAN

Seat No: 2201

ABC No: 575905236320

PRN: 201811474

Under the Mentorship of

MR. ROHIT KOTHAWADE

Goa Business School
Computer Science Discipline

GOA UNIVERSITY

DATE: MAY 2024

Examined By:

Seal of the School/Dept

DECLARATION BY STUDENT

I hereby declare that the data presented in this Internship Report entitled, "DATA MIGRATION FRAMEWORK AND TOOL FOR INVOICE MANAGEMENT" is based on the results of the investigation carried out by me in the Masters of computer Application(MCA) at the Goa Business School, Goa University , under the mentorship of Mr. Rohit Kothawade and the same has not been submitted elsewhere for the award of degree or diploma by me. Further I understand that Goa University or its authorities/Goa University will be not be responsible for the correctness of the observations or other findings given in the internship report.

I hereby authorize the Goa University authorities to upload this dissertation on the dissertation repository or anywhere else as the UGC regulations demand and make it available to any one as needed.

Shaikh Safwan

Seat No: 2201

Date:

Place: Goa University

COMPLETION CERTIFICATE

This is to certify that the internship report "DATA MIGRATION FRAMEWORK AND TOOL FOR INVOICE MANAGEMENT" is a bonafied work carried out by Mr Shaikh Safwan under my mentorship in partial fulfilment of the requirements for the award of the degree of Masters in Computers Application in the Dicipline of Computer Science at the Goa Business School, Goa University.

Mr. Rohit Kothawade

Date:

Signature of Dean of Goa Business School

School/Department Stamp

Date:

Place: Goa University

CONTENT

OFFER LETTER

neighborly

September 22, 2023

Mr.Safwan Shaikh
155, ward no 5, nanus roda, valpoi, Sattari, Valpoi, North Goa - 403506.
xec.safwan@gmail.com
9145786248

Dear Mr. Safwan Shaikh,

Neighborly Global Capability Center LLP is pleased to have you on board as an Intern. The duration of this internship is **6 months** starting from January 08, 2024 to July 05, 2024.

You are eligible for a monthly stipend of **INR 30,000** subject to withholdings/deduction of tax at source under prevailing regulations.

Neighborly Global Capability Center LLP subjects you to be bound by all employment rules, regulations, policies, code of ethics issued by the Organization from time to time. Furthermore, upon the successful completion of your internship and in alignment with meeting the organization's expectations, an offer letter for a full-time position will be extended to you.

Congratulations on your internship!

Best wishes,

Shekhar Manjargi
VP Engineering, India Site Lead

Acknowledgement:
I have read and understood the provisions of this letter, and I accept the internship opportunity.

Intern Signature: _____

Date: 23/09/2023

INTERNSHIP COMPLETION CERTIFICATE

neighborly

INTERNSHIP CERTIFICATE

This letter is to certify that **Mr. Safwan Shaikh**, student at **Goa Business School,** undergoing **Master of Computer Application** has successfully completed internship between **January 08, 2024, and July 05, 2024**. at **Neighborly Global Capability Center LLP**. He actively participated in several activities during the period of internship and learned skills such as React, C#, various automated testing frameworks, databases, and the Agile/Scrum processes and practices.

Thank you.

Sincerely,

Shekhar Manjargi
VP Engineering, India Site Lead
Date: June 05, 2024
Bangalore

NEIGHBORLY GLOBAL CAPABILITY CENTER LLP
(Entity registered with Limited Liability)
Registered Office: GRA-108, WeWork Roshani Arcade, Marathahalli Main Road, Lakshminarayana pura, EPIP Zone, Chinnappan Halli Marathahalli Colony, Bangalore, Bangalore North, Karnataka, India, 560037
LLPIN: ABZ-4259 | Email: neighborlyGCC@nbly.com ,Phone no: 080-37012626

ACKNOWLEDGEMENT

I would like to express my sincere gratitude towards Neighborly for giving me the opportunity to do internship for the duration of 6 months to enhance my horizon of knowledge and learn the latest and greatest of tech stack available out in the industry. Also, I am thankful to Mr. Shekhar Manjargi for considering me for the internship and enlightened me with his valuable guidance throughout the course of the internship period.

I extend my heartfelt thanks to Mr. Pankaj Jain for his continuous guidance and insights about Neighborly which helped me to elevate my corporate and technical skills. A great part in smooth internship experience also goes to Pankaj Jain as he organized a weeklong bootcamp where every senior manager from each vertical explained us about the company and helped me to have a clear vision on what vertical of the company I want to work on for the internship. This bootcamp also helped me to understand how the company functions and makes profit.

My special gratitude goes to Mr. Rohit Kothawade, Senior manager of Brand Integration Team for granting me the opportunity to be a part of his esteemed team during my internship at Neighborly. His guidance, support, and encouragement were invaluable to me throughout the duration of my project.

Mr. Rohit Kothawade extensive knowledge and experience were instrumental in helping me navigate the challenges I faced. His willingness to share insights and provide constructive feedback greatly enhanced my learning experience and contributed significantly to the successful completion of this project.

I am deeply grateful towards Miss Alisha Lotlekar and Mr. Ritesh Singh for their patience and guidance. Their insightful suggestions, thorough explanations, and hands-on assistance significantly enhanced my understanding and ability to tackle the technical challenges I encountered.

EXECUTIVE SUMMARY

Neighborly®, the world's largest home services franchisor, began as the Dwyer Group over 40 years ago. Currently, it operates in six countries, serving over 10 million people through more than 30 brands and over 5500 franchises. Neighborly provides comprehensive business services, including coaching, marketing, and training, aimed at enhancing work-life balance and business efficiency. With a $4.1 billion valuation, Neighborly's mission is to repair, maintain, and enhance properties, making life easier for homeowners. The company operates four main verticals: Digital, Onverity, and Integration Apps and Data Analytics each focusing on different aspects of the business .

During the internship, significant contributions were made to the Invoice Audit System, an internal tool designed and developed to compare vendor invoices with internal data present in the company, highlighting discrepancies. The tasks included backend development to manage vendor entries and logic to find the discrepancy, frontend development for user input and data display, comprehensive testing, and a major database migration from PostgreSQL to MS SQL Server. These efforts collectively enhanced the system’s functionality, user experience, and overall efficiency.

The internship provided a deep dive into modern tech stacks and enterprise-scale project management. Key learnings included backend and frontend development, data management, and system integration. The exposure to real-world business problems and their solutions, especially in automating processes to minimize human error and enhance productivity, was invaluable. Additionally, the internship offered insights into corporate culture and project management within a large, dynamic organization.

Several challenges were encountered during the internship. The primary challenge was ensuring data integrity during the database migration, which required meticulous planning and execution to prevent data loss or corruption. Additionally, developing a user-friendly interface that accurately captured and displayed discrepancy data required iterative testing and refinement. Navigating the complex

infrastructure of a global company and integrating various systems also posed significant challenges, demanding a robust understanding of both technical and business processes.

CHAPTER 1: COMPANY

1.1 BIRDS-EYE-VIEW

Neighborly® is the biggest home services franchisor in the world, having started as the Dwyer Group over 40 years ago with just one brand. Currently operating in six countries, it serves over 10 million people through more than 30 brands and over 5,500 franchises. Neighborly®, a $4.1 billion firm, offers franchise owners a comprehensive variety of business services, including coaching, marketing, and training, in an effort to improve work-life balance and company efficiency. Neighborly brands repair, maintain, and enhance properties — to make life easier and more enjoyable for homeowners. To effectively serve their communities, franchisees maintain their freedom while benefiting from being a part of a well-known global network.

Neighborly®, the world's leading home services firm, have a new Global Capabilities Center (GCC) in Bangalore, India. The GCC will enhance the company's technology transition by developing capabilities that benefit its brands, franchise owners, and customers. When combined with our Code of Values of Respect, Integrity, Customer Focus, and Having Fun in the Process, Neighborly sets the tone for our culture and business, and empowers the pursuit of entrepreneurship and a meaningful career. Neighborly has over 2,000 corporate employees worldwide with the Global Headquarters located in Waco, Texas and Irving, Texas.

The company has four main verticals that are namely Digital, Onverity and Integration Apps and Data Analytics. The Digital is client facing where all the customer portals and websites are developed. Onverity is the Field service management and point of sales for neighborly. Integration deals with smoothly integrating a newly acquired brand into Neighborly ecosystem.

1.2 PRODUCTS

1.2.1 Onverity

Onverity stands as Neighborly's cornerstone Field Service Management (FSM) platform, meticulously crafted to orchestrate the intricate management of concepts, brands, services, and service professionals within the Neighborly network. Guided by its namesake principle of "Always on with true principle," Onverity serves as the backbone for ensuring seamless operations in the repair, maintenance, and enhancement of properties. Its comprehensive capabilities are tailored to cater to the diverse array of service-based franchise businesses that comprise the Neighborly ecosystem, offering a unified solution to streamline operations and elevate service standards.

The genesis of Onverity is rooted in the recognition that each Neighborly brand currently operates using its distinct FSM/POS system, resulting in a fragmented landscape. This realization underscores the critical need for Neighborly to consolidate all its brands onto a single, cohesive FSM platform. By centralizing operations through Onverity, Neighborly aims to eliminate silos and establish a unified experience across all brands. This strategic alignment is not merely about operational efficiency; it represents a transformative step towards enhancing customer satisfaction and fostering stronger brand cohesion across the Neighborly network.

As Neighborly embraces the vision of a unified FSM platform, the implementation of Onverity heralds a new era of efficiency and collaboration. By harmonizing processes and data across brands, Onverity empowers Neighborly to unlock synergies, drive cross-selling opportunities, and optimize resource allocation. Moreover, the consolidation onto a singular platform enables Neighborly to leverage insights and analytics more effectively, facilitating data-driven decision-making and positioning the company for sustained growth and innovation in the dynamic home services market.

1.2.2. Field App

The Field App is an essential tool used by service professionals on-site within the Neighborly network. This app is designed to facilitate a wide range of tasks critical to field operations. Service professionals use the Field App to create work orders, manage ongoing services, and track job progress in real-time. It provides a user-friendly interface that allows technicians to access customer information, update service details, and capture important data directly from the job site. By streamlining these processes, the Field App enhances efficiency, reduces paperwork, and ensures that service professionals can deliver high-quality service consistently. This tool is integral to maintaining the seamless operations and high standards expected across all Neighborly brands.

1.3 SECTIONS WITH ORGANISATION

1.3.1 Four Verticals

1.3.1.a. Digital

Digital is a B2C portal that focuses largely on lead generation and acts as a uniform platform for all brands onboarded. The dedicated digital team manages the website neighborly.com. The platform has two versions: OPUS 1.0 and OPUS 2.0, with OPUS 2.0 being the most recent edition designed for onboarding new businesses, while brands presently using OPUS 1.0 will shortly migrate to OPUS 2.0. In addition to the digital website, the digital team oversees two other significant products: the franchise portal, which is utilized by franchisees, and the customer-portal.

The digital team is organized into five specialized groups. Four of these teams focus on different aspects of the digital website: Chargers handle OPUS 1.0 and the customer portal; Striker is dedicated to OPUS 2.0; Quarterback is responsible for website maintenance; and Explorers

manage the franchise portal. The fifth team, Thunderbolt, is dedicated to managing the mobile app.

1.3.1.b. Integration Apps

The Integration team manages data for all Neighborly brands. Their tasks include acquiring, integrating, and delivering a brand's data throughout Neighborly's infrastructure. Amey Poyekar, the TPM of BIT, and Senior Manager Rohit Kothawade manage this dynamic team.

The Brand Integration team manages contracts through FranConnect and uses RAN (Refer a Neighbor) to add new brands and CAN (Call A Neighbor) for similar connections. They play an important role in acquiring current POS data for brands on ONverity, ensuring that data is seamlessly integrated with Neighborly's data warehouse. This team also oversees data mapping, which is critical for ensuring data accuracy and consistency across multiple platforms.

1.3.1.c. Onverity

Onverity is Neighborly's field service management (FSM) system, which provides a feature-rich and highly adaptable solution targeted to a wide range of service-based franchise enterprises. Its primary goal is to repair, maintain, and improve properties. Neighborly, as a corporation, purchases several brands, each with its own POS and FSM systems. To streamline operations, it is critical to integrate all brands into a single FSM platform, which Onverity provides. Currently, three brands have been successfully integrated into Onverity, which works with the majority of Neighborly's applications.

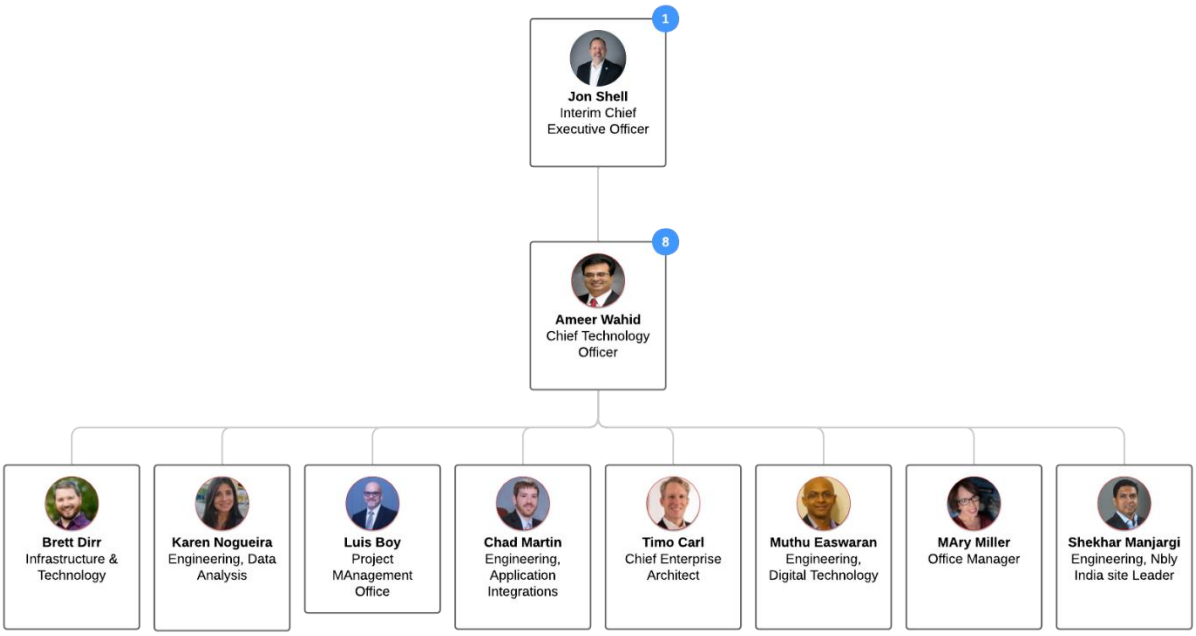
Manoj Panickar leads the Onverity team, which is divided into specialized scrum teams. Starfleet organizes work orders, Galaxy operates at the enterprise level, Artemis handles invoice-related operations, and Supernova oversees the Field App, which is utilized by service workers. Each of these teams consists of both frontend and backend devs, making Onverity a strong FSM capable of onboarding new brands. This organized

approach ensures that Onverity stays a powerful and scalable solution that can satisfy the changing needs of Neighborly's expanding brand portfolio.

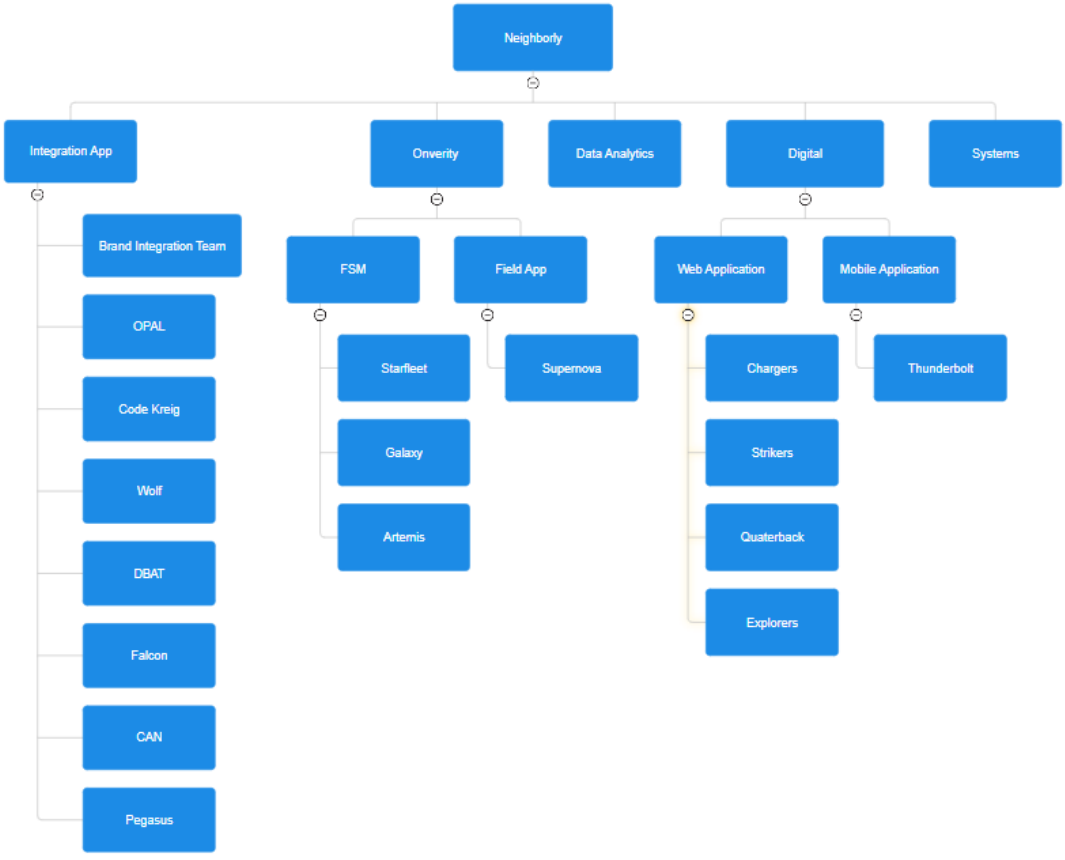
1.3.1.d Data Analytics

Neighborly's Data Analytics division is essential to the company's expansion since it carefully examines data to find insightful patterns, discover possible leads, and create task orders. This area uses cutting-edge analytical methods to sort through enormous volumes of data and turn it into insight that can be used to make strategic decisions. Neighborly may optimize marketing efforts, improve consumer engagement, and streamline operations by utilizing data analytics. Additionally, this vertical is in charge of producing thorough reports that give a clear and simple summary of business performance and trends based on the data that has been gathered. These reports are essential for ongoing development, which helps Neighborly keep its competitive advantage and provide clients with better service.

1.3.2 CTO Organization



1.3.4 Oninogram



CHAPTER 2: TASK(S) HANDELED

2.1 INVOICE AUDIT SYSTEM

2.1.1 About the Invoice Audit System

The internal tool for the company assists scrum master's by comparing vendor invoices with the company's internal data, highlighting any discrepancies between the two. This tool not only identifies inconsistencies but also pinpoints the exact reasons behind these discrepancies. Traditionally, scrum masters have had to manually review each project's individual vendor invoice, which then must be validated against the internal data. This manual process is not only time-consuming but also prone to errors due to the meticulous nature of checking each submission.

2.1.2 Why We Need This System

By automating this comparison, the internal tool significantly reduces the time and effort required to verify invoices, minimizing the risk of human error. Scrum masters can rely on this tool to ensure accuracy and efficiency in financial validation, allowing them to focus on more strategic aspects of their projects. This streamlined process enhances overall productivity and accuracy, providing a reliable solution for managing vendor invoices and internal data alignment.

2.1.3 My Contributions

During my time working on the Invoice Audit System, I handled various critical tasks across multiple sprints. In the initial sprint, I focused on backend development, implementing functionalities to create and delete vendor entries, ensuring robust data management and integrity. In the subsequent sprint, I transitioned to frontend development, working on capturing user input, importing data from Excel files, fetching discrepancy data, and displaying these

discrepancies clearly. In the third sprint, I conducted thorough testing of both frontend and backend systems, implementing suggested improvements to enhance functionality and user experience. Finally, in the fourth sprint, I managed the migration of our database from PostgreSQL to MS SQL Server, ensuring a smooth transition and maintaining system performance and reliability. These efforts collectively contributed to the development of a comprehensive and efficient Invoice Audit System.

2.2 ONVERITY BRAND MIGRATION (DATA MIGRATION FRAMEWORK)

2.2.1 Problem Statement

The goal was to develop a new Automated Data Migration Framework to migrate data from MS-SQL to Postgres that runs on demand and not on a local system.

2.2.2 Current Status

The current system migrates a subset of specific franchises for a specific POS. This migration is performed on-demand through a console application. The process involves pulling the data, transforming it into a format acceptable to FSM APIs, and then calling the API endpoints to create and/or update data in FSM.

2.2.3 Need for New Approach

The time to the first migration is too long, with migration projects taking 30-45 days for the first business unit migration. Migration is done via a custom-built console application, requiring manual code modifications. The BIT (Brand Integration Team) is directly involved in every migration, which requires a manual run on a local machine via debug and needs a lot of manual intervention.

2.2.4 My Contributions

During the Onverity Brand Migration project, I played a key role in enhancing the efficiency and automation of the data migration process. My work included:

- Clean Architecture: Implemented clean architecture principles to ensure the codebase was maintainable, scalable, and easily testable.
- Rx.NET: Utilized Rx.NET for managing asynchronous data streams, which improved the responsiveness and reliability of the migration process.
- Quartz: Integrated Quartz for scheduling and managing background tasks, enabling the migration framework to operate on demand without manual intervention.
- CQRS (Command Query Responsibility Segregation): Applied CQRS patterns to separate read and write operations, optimizing the performance and scalability of the data migration process.

In addition to these technical contributions, I developed the backend systems for several key functionalities:

- Sync Customer: Created backend services to automate the synchronization of customer data from Franforce to Onverity.
- Update Sync Customer: Implemented backend processes to update previously synced customer data, ensuring data consistency and accuracy.
- Sync Contractor: Developed backend functionalities to sync contractor data, facilitating seamless data migration for contractor-related information.
- Sync Workorder Media: Built backend systems to handle the migration of workorder media, ensuring all media files associated with work orders were accurately transferred and accessible in the new system.

These contributions collectively enhanced the speed, reliability, and efficiency of the data migration process, significantly reducing the manual effort required and enabling the BIT team to focus on strategic tasks.

CHAPTER 3: LEARNINGS

Through my involvement in the various tasks handled during the internship, I gained significant technical and practical insights, which are detailed below:

3.1 CLEAN ARCHITECTURE

Implementing clean architecture principles ensured the maintainability, scalability, and testability of the codebase. This approach helped in structuring the code in a way that separated concerns effectively, making the system more robust and easier to modify or extend.

3.2 OBSERVABLE PATTERN (Rx.NET)

Utilizing Rx.NET for managing asynchronous data streams improved the responsiveness and reliability of the migration process. This pattern was crucial in handling real-time data streams and event-driven programming, which enhanced the overall system performance.

3.3 COMMAND QUERY RESPONSIBILITY SEGRIGATION (CQRS)

Applying the CQRS pattern allowed for the separation of read and write operations, optimizing the performance and scalability of the data migration process. This pattern facilitated a clear distinction between commands (actions that change state) and queries (actions that retrieve data), resulting in a more efficient and maintainable system.

3.4 SCHEDULING CRON JOB

Integrating Quartz for scheduling and managing background tasks enabled the migration framework to operate on demand without manual intervention. Understanding how to schedule and

automate tasks using cron jobs was critical for ensuring timely and reliable execution of background processes.

3.5 LANGUAGE INTEGRATED QUERY (Linq)

Leveraging LINQ (Language Integrated Query) provided a powerful tool for querying data in a concise and readable manner. This skill was essential for manipulating and accessing data within the .NET framework, streamlining the process of data retrieval and transformation.

3.6 CODE FIRST APPROACH

Adopting the Code First approach in Entity Framework Core allowed for the creation of database schemas directly from the code. This methodology facilitated a more agile development process, where changes to the data model could be quickly reflected in the database structure.

3.7 ENTITY FRAMEWORK CORE

Utilizing Entity Framework Core for database interactions ensured efficient data management and access. This ORM (Object-Relational Mapping) tool simplified the process of working with relational data in applications, providing a seamless integration between the database and the code.

3.8 REACT WITH TYPESCIPT

Developing frontend applications using React with TypeScript enhanced my ability to build scalable and type-safe web applications. TypeScript’s static typing and Reacts component-based architecture together improved code quality, maintainability, and development speed.

3.9 XUNIT TESTING

Implementing xUnit testing for .NET applications helped in writing unit tests to ensure code reliability and correctness. This framework's features, such as data-driven tests and custom assertions, were instrumental in maintaining high code quality and preventing regressions.

3.10 CYPRESS TESTING

Using Cypress for end-to-end testing of web applications provided a robust framework for ensuring the application's functionality from a user perspective. Cypress's real-time reloads and debugging capabilities made it an invaluable tool for maintaining high-quality user interfaces.

3.11 KARATE TESTING

Leveraging Karate for API testing allowed for efficient testing of web services. Its intuitive DSL (Domain Specific Language) and seamless integration with CI/CD pipelines ensured comprehensive API test coverage and reliable service interactions.

These learnings collectively contributed to my professional growth, enhancing my technical proficiency, and enabling me to contribute effectively to the projects during my internship.

CHAPTER 4: CHALLENGES

During the initial phases of the internship, several challenges were encountered, particularly in understanding and implementing Clean Architecture, Observables, and Entity Framework Core. These challenges, while daunting initially, provided valuable learning opportunities and insights into best practices in software development.

4.1 DEALING WITH DATA:

Since data migration project is all related to data and tremendous amounts of data. The data was coming from 3 different places we had to do a lot of transformations of the data and get all the synergy from it and save that in a single database with all JsonB columns.

4.2 UNDERSTANDING CLEAN ARCHITECTURE:

Initially, grasping the principles of Clean Architecture posed a significant challenge. Decoupling components, defining boundaries, and structuring the application layers required a paradigm shift in my approach to software design.

Overcoming this challenge involved extensive research, seeking guidance from mentors, and studying real-world examples of Clean Architecture implementations. Through persistent effort and experimentation, I gradually gained a deeper understanding of architectural patterns and their implications on software design.

4.3 WORKING WITH OBSERVABLES:

The concept of Observables and reactive programming presented a steep learning curve. Understanding asynchronous data streams, managing subscriptions, and handling complex event flows proved to be challenging.

To address this challenge, I engaged in hands-on practice, experimenting with Rx.NET and exploring various reactive programming concepts through tutorials and documentation. Collaborating with team members and seeking assistance from experienced developers also played a crucial role in overcoming this hurdle.

4.4 NAVIGATING ENTITY FRAMEWORK CORE:

Initially, working with Entity Framework Core for database interactions posed challenges in terms of configuration, performance optimization, and understanding the intricacies of ORM (Object-Relational Mapping).

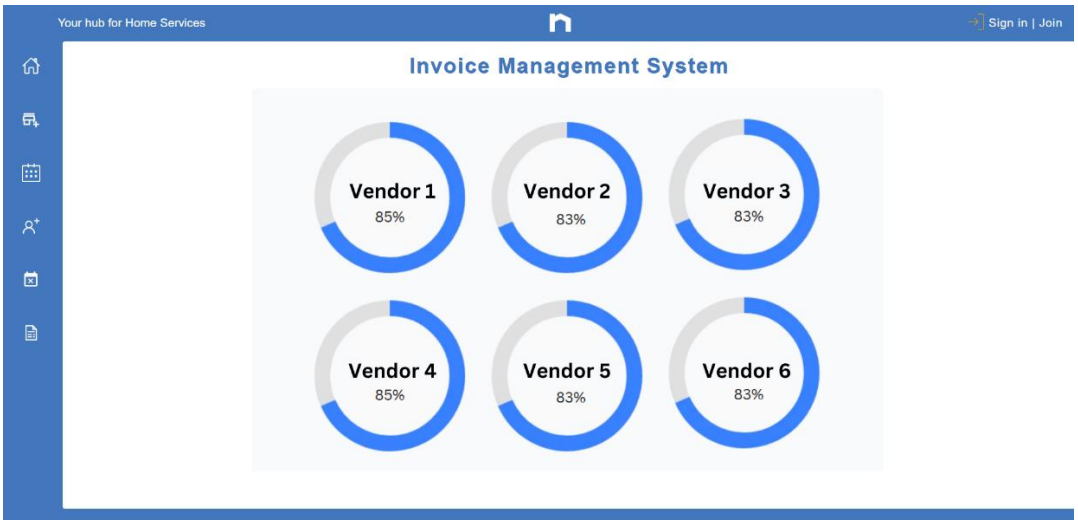
To overcome these challenges, I dedicated time to studying EF Core documentation, attending workshops, and seeking mentorship from senior developers. By gradually building proficiency through practical application and troubleshooting, I gained confidence in leveraging EF Core effectively for database operations.

While these challenges initially presented hurdles in the internship journey, they ultimately served as valuable learning experiences. Through perseverance, collaboration, and continuous learning, I was able to overcome these obstacles and make significant contributions to the projects.

APPENDIX I: SNAPSHOTS OF THE WORK DONE

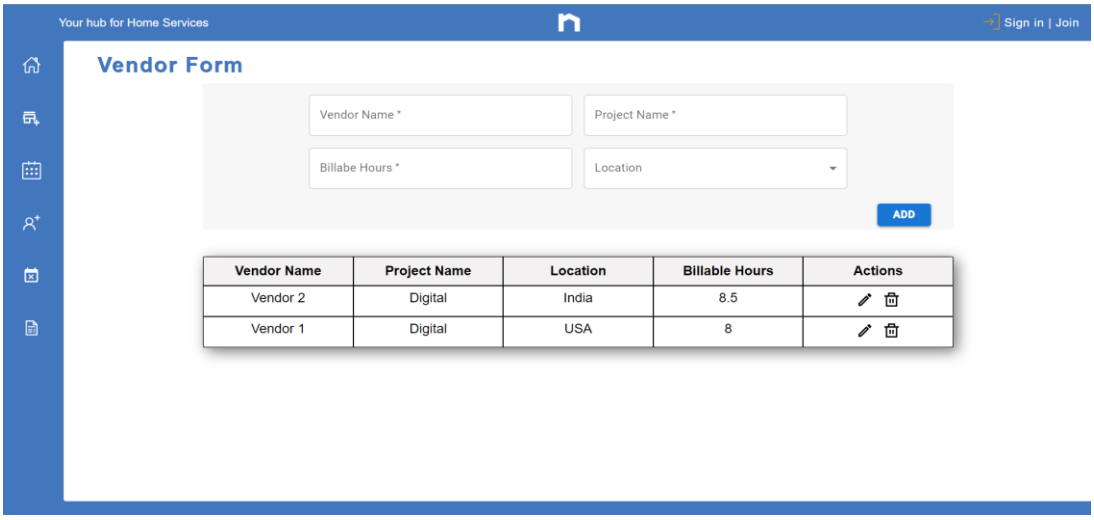
- INVOICE AUDIT SYSTEM

This is the landing page of the application.



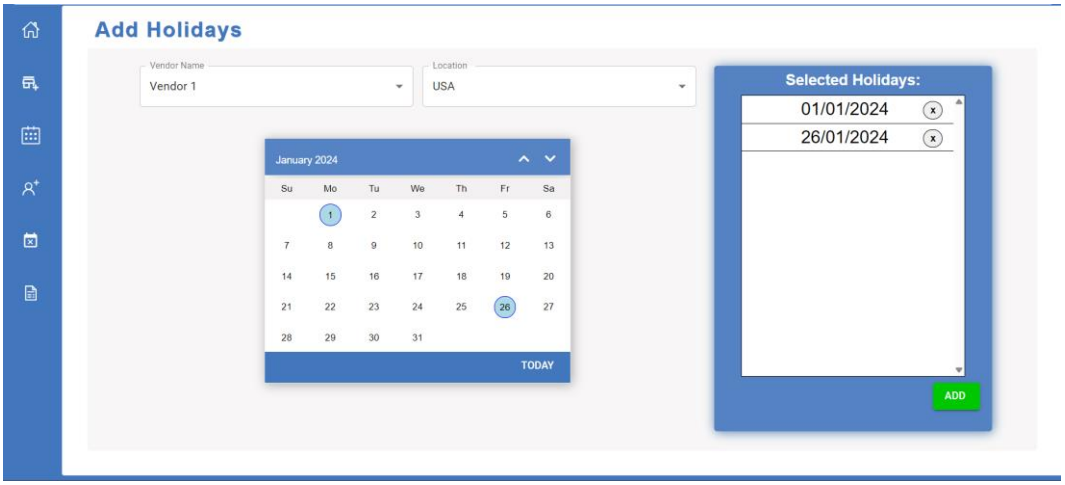
This page was not in scope of project and was an additional future scope added where it shows the discrepancies found in the data.

Vendor Form:



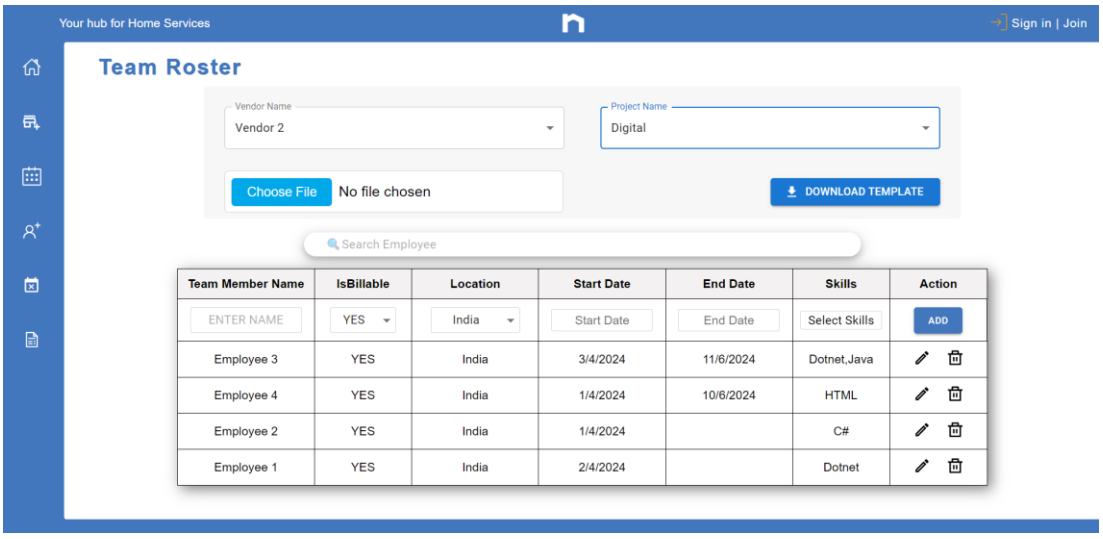
In this page you add all the vendor team's data and the number of billable hours per day that the team will work. Also, here you add the location of the vendor as based on the location total billable hours changes.

Add Holidays page:



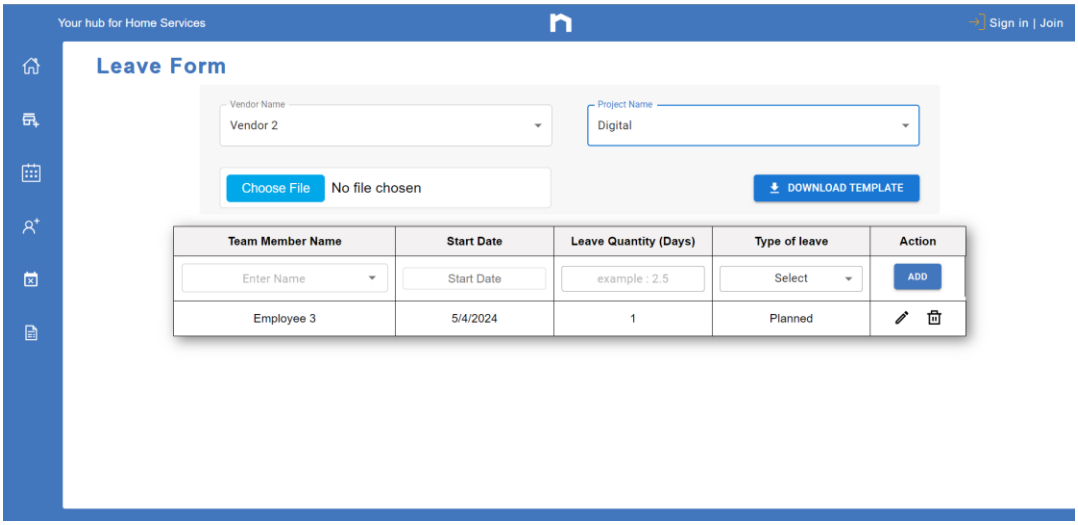
In section of the app, you add all the vendor specific holidays. As the vendor cannot bill on holidays. Here there is a location dropdown as the holiday list for the company differs based on the location.

Team Roster:



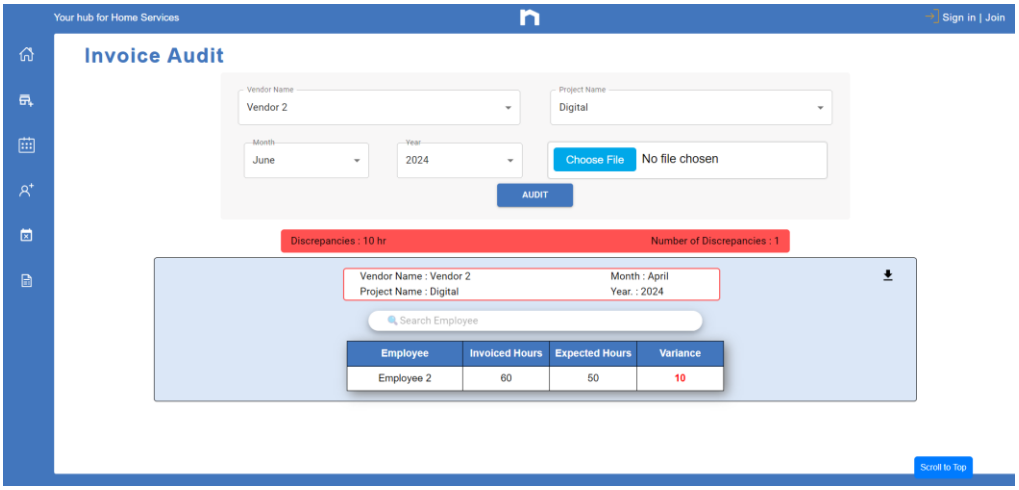
In team roster you add all the members of the team along with their necessary data such as if they are billable or not. Also, with some important data like their start date and end date (if any).

Leave form:



In this form we add all the leaves that the employee has taken. The leave can of type planned, unplanned or ad hoc. It has start date and leave quantity with an increment of 0.5 (half-day).

Discrepancy Page:



This is the final module of the application. Here you add the vendor invoice received along with all the necessary data required. And you get the discrepancy present in the vendor invoice against company internal data about the same.

- DATA MIGRATION FRAMEWORK

  Due to company confidentiality I can only share the snapshots that are reviewed and permissible by the company to be included in the report.

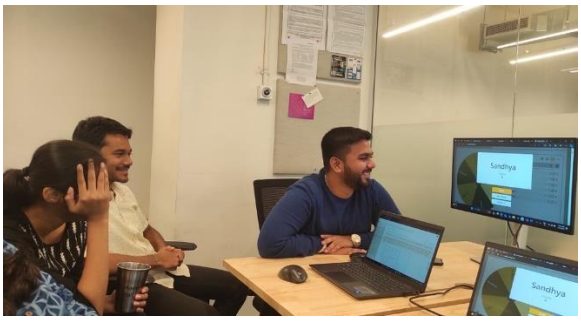
```
#region Customer
1 reference | Sanket Narvekar, 2 days ago | 2 authors, 3 changes
Task<CustomApiResponse> GetByIds(CustomersByIdsRequest customersByIdsRequest, FsmContext fsmContext);
1 reference | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> GetCustomerAsync(Guid correlationId, Guid id, FsmContext fsmContext);
1 reference | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<ServiceBulkInsertResponse> CreateBulkCustomerAsync(Guid correlationId, CreateBulkCustomerRequest customers, FsmContext fsmContext);
2 references | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> CreateCustomerAsync(Guid correlationId, FSMCreateCustomerDto customers, FsmContext fsmContext, long SyncCustomerId);
2 references | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> UpdateCustomerAsync(Guid correlationId, UpdateCustomerRequest customers, FsmContext fsmContext);
1 reference | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> GetCustomersAsync(Guid correlationId, FsmContext fsmContext, int pageSize, int pageNumber);
4 references | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> GetCustomerByExternalIdAsync(Guid correlationId, FsmContext fsmContext, long externalId);
2 references | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> UpdateCustomerContactAsync(Guid correlationId, UpdateContactDetails customers, FsmContext fsmContext);
1 reference | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<BlobSasUrlResponse> CreateCustomerMediaAsync(Guid correlationId, BlobSasUrlRequest customers, FsmContext fsmContext);
1 reference | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> CreateCustomerMediaAsync(List<AttachmentRequest> attachmentRequests, FsmContext fsmContext);

1 reference | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> GetFileMetaDataAsync(Domain.Enums.EntityType entityType, Guid entityId, FsmContext fsmContext);
#endregion
```

Product Catalog

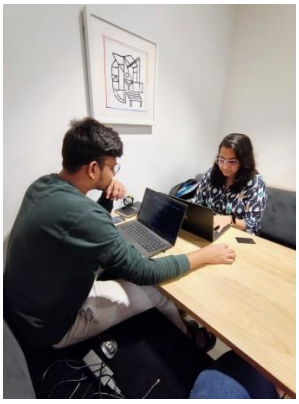
```
#region Workorders
2 references | Sanket Narvekar, 2 days ago | 2 authors, 3 changes
Task<CustomApiResponse> CreateWorkorder(Guid correlationId, CreateWorkorderRequest workorder, FsmContext fsmContext, string timezone);
1 reference | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> CreateWorkorderInvoiceDetails(Guid correlationId, InvoiceDetailsRequest detail, FsmContext fsmContext);
2 references | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> SearchWorkorderByExternalId(Guid correlationId, FsmContext fsmContext, WorkOrderGenericSearchRequest req);
5 references | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> UpdateWorkorderStatus(Guid correlatedId, FsmContext fsmContext, WorkOrderStatusUpdateRequest model);
1 reference | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> UpdateWorkorder(Guid correlationId, WorkOrderMultidayUpdateRequest model, FsmContext fsmContext, string timezone);
1 reference | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> CreateCheckList(Guid correlationId, CheckListCreateRequest checklist, FsmContext fsmContext, Guid Id);
4 references | Shaikh Safwan, 22 days ago | 2 authors, 3 changes
Task<CustomApiResponse> GetConfigDetailsById(Guid correlationId, ConfigName configName, OrgEntityType orgEntityType, string entityId);
2 references | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> WorkorderByExternalId(Guid correlationId, FsmContext fsmContext, List<long> ids);
2 references | Shaikh Safwan, 22 days ago | 2 authors, 3 changes
Task<CustomApiResponse> GetTimeZoneDetailsById(Guid correlationId, ConfigName configName, List<string> buid);
5 references | Shaikh Safwan, 48 days ago | 1 author, 1 change
Task<CustomApiResponse> GetWorkorderById(Guid correlationId, FsmContext fsmContext, Guid id);
#endregion
```

APPENDIX II: PHOTOS WHILE AT WORK

Presentation to Neighborly India Leadership Team:



Selecting Scrum Master and Product owner for the warmup project:



Having discussion on backend and meetings with team:

And most a crucial code of value of Neighborly (Having fun in the process):

At the t-shirt painting event

Women's Day celebration at Neighborly: