J.C. Bose University of Science & Technology, YMCA, Faridabad

DEPARTMENT OF COMPUTER ENGINEERING

FRANCHISEE ANALYSIS APPLICATION

Project Report

Submitted in the partial fulfilment of the requirement for the award of the Degree of

Bachelors of Technology

In

COMPUTER ENGINEERING

BY:	**Mentor:**	**Manager:**
Sarthak Nandwani	Sawani Mhaske	Sawani Mhaske
19001003118	Software Architect	Manager

TRAINING COMPLETION CERTIFICATE

ALLIED
Softech Pvt. Ltd.

CIN NO: U72200PN2006PTC022306
Regd. Office : Austin Belvedere S. No. 85/B/9/1, Lalit Estate, Near Ganraj Mangal Karyalaya, Baner, Pune - 411045
Ph. : 020 - 46919999. **E-mail** : sales@alliedsoftech.com, customercare@alliedsoftech.com **Visit us at** : www.alliedsoftech.com

Ref. No. :

Date :

Internship Certificate

This is to certify that Mr. Sarthak Nandwani S/o Dinesh Nandwani of B.Tech - (Computer Engineering) of J.C. Bose University of Science and Technology, YMCA, Faridabad, has undergone training with Allied Softech Pvt. Ltd. from 16/01/2023 to 07/07/2023 under Franchisee Analysis Application.

During the above-mentioned period, his conduct and behavior remains good. He has demonstrated willingness to learn new skills on his own and he was able to complete the assigned work on time.

We expect he continues good work and achieves further milestones.

For ALLIED SOFTECH PVT LTD

AUTHORISED SIGNATORY
(Manager – HR)

CANDIDATE DECLARATION

This is to certify that the work presented in the project report entitled **Franchisee Analysis App** in partial fulfilment of the requirement for the award of degree of "Bachelor of Technology in Computer Engineering of J.C. Bose University of Science and Technology, YMCA, Faridabad" is an authentic work carried out under supervision and guidance of the Mentor.

To the best of my knowledge, the content of this thesis does not form a basis for award of any previous Degree to anyone else.

ACKNOWLEDGEMENT

I would like to thank all the people who helped and supported me in this project.

I would like to express my gratitude to my manager, Sawani Mhaske for constant motivation for working on this project. I express sincere gratitude to my mentor Sawani Mhaske for her valuable guidance.

I would also like to thank all the other team members- Shyam Agarwal, Amruta Yewale who have been working with me on the project.

I extend my gratitude to the CE Department of J.C. Bose University of Science & Technology, YMCA, Faridabad and the respected teachers for giving me an opportunity toward a project like this. I perceived this opportunity as a big milestone in my career development.

Sarthak Nandwani

19001003118

EXAMINER CERTIFICATE

The following examiners have accessed the project report and conducted the viva-voce Examination:

Internal Examiners

Name _____

Affiliation _____

_____

Contact no. _____

Email Id. _____

Signature_____

External Examiners

Name _____

Affiliation _____

_____

Contact no. _____

Email Id. _____

Signature_____

TABLE OF CONTENTS

Chapter 1

COMPANY PROFILE

ALLIED SOFTECH PVT. LTD.

Fig 1: Allied Softech Pvt. Ltd. Logo

Allied Softech is a software development company that focuses on the pharmaceutical, hospitality, healthcare, and retail industries.

Founded in 1997, Allied Softech is a leading commercial software development company based in Pune, India. We provide softwares in Pharmaceutical Retail and Pharmaceutical Wholesale industries. More than 32000 medical retailers and wholesalers are using MediVision Pharmacy software satisfactorily. The ever-vibrant team at Allied works hard to ensure the high quality and excellence towards products. Dedicated Customer Relationships and After Sales Support where skilled and courteous customer care team is there to provide you professional and technical support. From Pharmaceutical supply chain to Pharma Retail and healthcare, Allied Softech provides tangible benefits to the customer by providing software that excels in services and performance.

Allied Softech Private Limited is a Private incorporated on 07 April 2006. It is classified as a Non-government company and is registered at Registrar of Companies, Pune. Its authorized share capital is Rs.1,000,000 and its paid up capital is Rs.900,000. It is involved in Software publishing, consultancy and supply [Software publishing includes production, supply and documentation of ready-made (non-customized) software, operating systems software, business & other applications software, computer games software for all platforms. Consultancy includes providing the best solution in the form of custom software after analyzing the user's needs and problems. Custom software also includes made-to-order software based on orders from specific users. Also, included are writing of software of any kind following directives of the users; software maintenance, web-page design].

Allied Softech Private Limited's Annual General Meeting (AGM) was last held on 30 November 2021 and as per records from Ministry of Corporate Affairs (MCA), its balance sheet was last filed on 31 March 2021.

Directors of Allied Softech Private Limited are Vidula Prasad Sahasrabuddhe, Jyoti Sameer Renuse, Sameer Madhukar Renuse and Prasad Chintamani Sahasrabuddhe.

Chapter 2

INTRODUCTION TO PROJECT

INTRODUCTION TO PROJECT

Franchisee Analysis Application (FAA) is a separate app for users who have their own or contracted franchisee locations.

The aim of this project is to facilitate purchase of goods in bulk for better deals, to pass on part of the benefit to franchisees and to collect data from franchisees for analysis. Unlike other applications of Allied Softech, order generation is out of scope (at least as of now) (user is expected to use MVWP (MediVision Wholesale Platinum)). Also, other business transactions (purchase, sale, receipts, payments, etc.) are out of scope for this app (user is expected to use MVWP).

Page Import data processing (**'pg_import_data_processing'**)

Export application sends the following requests from export application to FAA (Franchisee Analysis Application):

- Compare client version with server version: Validate that the client version is equal to or greater than the minimum client version supported by FAA server for franchisee export app. This version changes if there are breaking interface changes at client/server side.
- Begin export: FAA creates zip file in "Exported zips" folder. This folder is present at executable path. Data is written in this file during subsequent calls of 'export data in chunk'.
  1. Export data in chunk: In this request, 15 Mb of data is sent in each chunk.
  2. End export: FAA sends a job_id in 'end export' response upon successful upload of zip file.
  3. Check pending job status - export app checks pending job status periodically until it gets status as success or failure.

Chapter 3

REQUIREMENT ANALYSIS

REQUIREMENT ANALYSIS

3.1. INTRODUCTION

3.1.1 Purpose

Franchisee Analysis Application is a separate application for users who have their own or contracted franchisee locations.

3.1.2 Scope

- purchase goods in bulk for better deals
- pass on part of the benefit to franchisees
- collect data from franchisees for analysis
- order generation is out of scope (at least as of now) (user is expected to use MVWP)
- other business transactions (purchase, sale, receipts, payments, etc.) are out of scope for this app (user is expected to use MVWP)

3.2. OVERALL DESCRIPTION

3.2.1 Hardware Requirements

3.2.1.1 Standalone App:

- PC with Dual or higher core processor, more cores or processors may be required for more than average workload
- RAM (Memory) available for the app: 1 or more GB for 32-bit, 2 GB or higher for 64-bit systems
- Free Disk Space for program + temp storage: 100 MB.
- Free Disk Space for user data: 5 GB per year. This is just a guideline. Actual requirement may be substantially more or less depending upon user's actual data size.
- Screen resolution: 1024 x 768 or higher
- Operating System(s): Windows XP SP3 onward, Recommended: Windows 10 onward

3.2.1.2 Desktop Client App (LAN or WAN clients):

- PC with Dual or higher core processor

- RAM (Memory) available for the app: 1 GB or more for 32-bit, 2 GB or higher for 64-bit systems
- Free Disk Space for program + temp storage: 100 MB.
- Screen resolution: 1024 x 768 or higher
- Network bandwidth: 2 Mbps or higher. This is a guideline. Actual requirement may differ depending upon data size and usage.
- Operating System(s): Windows XP SP3 onward, Recommended: Windows 10 onward

3.2.1.3 Server App (LAN or WAN server):

- Server system with Dual or higher core processor, more cores or processors may be required for more than average workloads
- RAM (Memory) available for the app: 2 GB or more for 32-bit, 4 GB or higher for 64-bit systems
- Free Disk Space for program + temp storage: 1 GB.
- Free Disk Space for user data: 5 GB per year. This is just a guideline. Actual requirement may be substantially more or less depending upon user's actual data size.
- Screen resolution: 1024 x 768 or higher
- Network bandwidth: 8 Mbps or higher. This is a guideline. Actual requirement may differ depending upon data, usage and no. of concurrent users.
- Operating System(s): Windows XP SP3 onward, Recommended: Windows Server 2008 R2 SP1 onward

3.2.2 Software Requirements

- Visual Studio Code: Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.
- C++: C++ is a high-level, general-purpose programming language created by Danish computer scientist Bjarne Stroustrup. First released in 1985 as an extension of the C programming language, it has since expanded significantly over time; modern C++ currently has object-oriented, generic, and functional features, in addition to facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers, including the Free Software

Foundation, LLVM, Microsoft, Intel, Embarcadero, Oracle, and IBM. C++ was designed with systems programming and embedded, resource-constrained software and large systems in mind, with performance, efficiency, and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, video games, servers (e.g. e-commerce, web search, or databases), and performance-critical applications (e.g. telephone switches or space probes).

3.2.2 Non-Functional Requirements

Non-functional requirements define system properties and constraints it arises through user needs, because of budget constraints or organizational policies, or due to the external factors such as safety regulations, privacy registration and so on. Nonfunctional requirements are:

- Security
- Reliability
- Maintainability
- Portability
- Extensibility
- Reusability
- Compatibility
- Resource Utilization

Chapter 4

DESIGN OF PROJECT

DESIGN OF PROJECT

Design Steps which are followed in the solution:

- authenticate with franchisee's username/password - send login request.
- receive data (zip of csvs)
- support chunked upload of the zip file
- handle concurrent uploads from different franchisees
- process and import uploaded data
- log errors to a db table for reporting (one row in upload masters table, zero or more errors in tran table)
- support re-upload for the same period, overwrite previous data
- masters should be processed like fresh export processing.
- Transactions data
- Transactions for non included products for the period should be deleted.
- expectation is that if export fails, after successful retry, data should be correct as per exported data.
- usage notes (even though technically this doesn't matter for the implementation):
- mvrp to send master + transaction data
- mvwp to send only product master data - out of scope
- mvwp's product master will allow to identify products which are not found in the imported data - for promoting those products - report is in current scope

`

Chapter 5

MODULE INFORMATION

MODULE INFORMATION

Reports

1. Uploads Report
    - list of all/selected franchisee uploads, date, upload status (success, with errors)
    - on click, show errors (or "no errors" message)
    - upload status master table : id(PK) franchisee_ie_code, date, upload status, error count
    - upload status transaction table - fields = type, ie_code, name, extra info, error, master_id (FK)
2. Non Moving Products Report
    - inputs: all/selected franchisee, company - multi selection, period (days)
    - columns: product name, unit, company short name, last sale date, balance quantity (calculate using op + in tran - out tran)
    - processing:
        - find last sale date using transactional data (from batch-wise stock table)
        - if last sale date < sys date - period, it's a non-moving product
3. Expiry Report
    - inputs: all/selected franchisee, company - multi selection, from date, to date
    - columns: product name, unit, company short name, balance quantity (from batch-wise stock table)
    - processing:
        - product-wise sum of quantity of batches having expiry date non-null and between the specified period
        - on click - selected product (expiring) stock report (similar to mvwp/mvrp)
    - columns: batch code, expiry date, MRP, balance quantity

.

`

Chapter 6

CODE REFACTORING AND DESIGN PATTERN

CODE REFACTORING

As our task was to refactor the existing codebase of the app using best practices, design pattern and agile methodology and to remediate the cyber vulnerability. The look and feel of the front end should remain the same.

Code refactoring is the process of restructuring existing computer code—changing the factoring—without changing its external behaviour. Refactoring improves non-functional attributes of the software. Advantages include improved code readability and reduced complexity; these can improve source-code maintainability and create a more expressive internal architecture or object model to improve extensibility.

Typically, refactoring applies a series of standardized basic micro-refactoring, each of which is (usually) a tiny change in a computer program's source code that either preserves the behavior of the software, or at least does not modify its conformance to functional requirements. Many development environments provide automated support for performing the mechanical aspects of these basic refactoring. If done extremely well, code refactoring may help software developers discover and fix hidden or dormant bugs or vulnerabilities in the system by simplifying the underlying logic and eliminating unnecessary levels of complexity. If done poorly it may fail the requirement that external functionality not be changed, introduce new bugs, or both.

By continuously improving the design of code, we make it easier and easier to work with. This is in sharp contrast to what typically happens: little refactoring and a great deal of attention paid to expediently adding new features. If you get into the hygienic habit of refactoring continuously, you'll find that it is easier to extend and maintain code.

— Joshua Kerievsky, Refactoring to Patterns

Refactoring is usually motivated by noticing inefficiencies in software code. For example, a method at hand may be very long, or it may be a near duplicate of another nearby method. Once recognized,

such problems can be addressed by refactoring the source code, or transforming it into a new form that behaves the same as before but is

more efficient. For a long routine, one or more smaller subroutines can be extracted; or for duplicate routines, the duplication can be removed and replaced with one shared function. Failure to perform refactoring can result in accumulating technical debt; on the other hand, refactoring is one of the primary means of repaying technical debt.

There are two general categories of benefits to the activity of refactoring.

Maintainability. It is easier to fix bugs because the source code is easy to read and the intent of its author is easy to grasp. This might be achieved by reducing large monolithic routines into a set of individually concise, well-named, single-purpose methods. It might be achieved by moving a method to a more appropriate class, or by removing misleading comments.

Extensibility. It is easier to extend the capabilities of the application if it uses recognizable design patterns, and it provides some flexibility where none before may have existed.

Before refactoring a section of code, a solid set of automatic unit tests is needed. The tests are used to demonstrate that the behavior of the module is correct before the refactoring. If a test fails, it is generally best to fix the test first; if this is not done, it is hard to distinguish between failures introduced by refactoring and failures that were already present. After refactoring, the tests are run again to verify the refactoring did not break the tests. Unit tests cannot prove that there are no bugs, but the important point is that this process can be cost-effective: good unit tests can catch enough errors to make them worthwhile and to make refactoring safe enough.

The process is then an iterative cycle of making a small program transformation, testing it to ensure correctness, and making another small transformation. If at any point a test fails, the last small change is undone and repeated in a different way. Through many small steps the program moves from where it was to where you want it to be. For this very iterative process to be practical, the tests must run very quickly, or the programmer would have to spend a large fraction of his or her time waiting for the tests to finish. Proponents of extreme programming and other agile software development describe this activity as an integral part of the software development cycle.

`

Chapter 7

INFORMATION ABOUT TESTING STRATEGY

INFORMATION ABOUT TESTING STRATEGY

Users interact with your app on a variety of levels. Accordingly, you should test a variety of use cases and interactions as you iteratively develop your app. After introducing any new code or modification in existing code base, the changes are first tested in the local workspace. After local testing works well, we push it to the feature or develop branch on Git.

Since most products are interlinked and dependent on other components, different team members often need to collaborate to fix bugs or develop a solution for an issue. Microsoft Azure is very useful in such cases to track the progress and see which team members are involved in the fix. It also helps to find out if a similar issue has been encountered earlier so that it can be solved quicker. All tasks are given a priority ranking so that high priority tasks can be resolved first.

Chapter 8

CONCLUSION AND FUTURE WORK

CONCLUSION AND FUTURE WORK

1. CONCLUSION

The code for the initial version is ready for release. All the team members have collaborated successfully to ensure smooth development and testing process. The application successfully imports data remotely and displays reports accordingly.

2. FUTURE SCOPE

In the future, the team will work on developing new features to be released in further versions of the application.

Chapter 9

REFERENCES

REFERENCES

1. https://alliedsoftech.com
2. https://www.zaubacorp.com/company/ALLIED-SOFTECH-PRIVATE-LIMITED/U72200PN2006PTC022306
3. https://www.tofler.in/allied-softech-private-limited/company/U72200PN2006PTC022306
4. https://www.thecompanycheck.com/company/allied-softech-private-limited/U72200PN2006PTC022306

Chapter 10

BRIEF PROFILE OF STUDENT

BRIEF PROFILE OF STUDENT

Name : Sarthak Nandwani

Roll No : 19001003118

Branch : Computer Engineering

Email Id : nandwanisarthak@gmail.com

Mentor's Name: Sawani Mhaske

Mentor's Email Id: sawani.mhaske@alliedsoftech.com