MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Stock Management System

Submitted By:

Shirish Saxena

0901CA201056

Mentor:

Vikas Garg, CTO, KaptureCRM

Dr. Parul Saxena, Assistant Professor, MITS

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Stock Management System

A project report submitted in partial fulfilment of the requirement for the degree of

MASTER OF COMPUTER APPLICATION

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Shirish Saxena

(0901CA201056)

Mentor:

Vikas Garg, CTO, KaptureCRM

Dr. Parul Saxena, Assistant Professor, MITS

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

kapture

TO WHOM IT MAY CONCERN

Date: 23.05.2022

This is to certify that Mr. Shirish Saxena, Has undertaken a Software Developer Trainee Internship period from 03.01.2022 to 03.05.2022 at Kapture CRM- Adjetter Media Network PVT LTD, Bangalore.

Enrollment No: 0901CA201056
Project Title: Stock Management

You have been an integral part of the company and we appreciate your contributions during the period of your Internship at Kapture CRM.

We wish you success in your future endeavors.

For Adjetter Media Network Pvt. Ltd.

Vikas Garg
(CTO)

ADJETTER MEDIA NETWORK PVT. LTD.
Bangalore

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Shirish Saxena**(0901CA201056) has submitted the project report titled **Stock Management System** under the mentorship of **Vikas Garg**, in partial fulfilment of the requirement for the award of degree of Master of Computer Application of Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Saxena 30/5/22

Dr. Parul Saxena
Faculty Coordinator
Assistant Professor
Computer Science and Engineering

30/05/2022

Dr. Manish Dixit
Professor and Head
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Master of Computer Application in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Vikas Garg (CTO)**, The Adjetter Media Network Pvt. Ltd. (KaptureCRM)

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Shirish Saxena
0901CA201056
2020 Batch,
Master of Computer Application,
Computer Science and
Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary project. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I would like to express my sincere gratitude to **Vikas Garg** at The Adjetter Media Network Pvt. Ltd. (KaptureCRM). His guidance has been a significant motivation for me to learn new skills and excel at work. The regular meetings and interactions have improved my technical skills to more than a level of self-satisfaction. I am glad that I have received such an opportunity to collaborate and enhance my inter and intrapersonal skills.

I am sincerely thankful to my faculty coordinator. I am grateful to the guidance of **Dr. Parul Saxena**, Assistant Professor, Computer Science and Engineering, for her continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Shirish Saxena

0901CA201056

2020 Batch,

Master of Computer Application,

Computer Science and

Engineering

ABSTRACT

This document specifies the details of the project I am working on. The name of the project is “Stock Management”, and we are making it for the client known as Prestige.

The project is completely related to how a company manages its products. The products must reach the customers via several routes. One of them is through sales executives. The chain involves moving the stock from the company to godowns at different places and allocating them to the executives. There are several internal processes that take place to maintain the whole system efficiently.

Some of the operations include creating godowns, adding and deducting stock from the godown, allocating or deallocating stock to and from the sales executives, and creating purchase orders that are made by authorized godowns to the company to receive stock from the factory or the company (known as internal orders), generating invoices, accepting stocks, returning damaged or any other stock, etc. Before all these tasks were handled manually or with the help of Google docs which was a very hectic and lengthy process, to make this repetitive task simple we developed multiple microservices. This document breaks down every functionality we have developed till the date of submission of this report and clearly explains it.

TABLE OF CONTENT

CHAPTER 1. INTRODUCTION

This project is a complete web-based application, and the main aim of the project is to develop a model in which all the information regarding the stock and employers in the organization will be presented.

1.1 Problem Identification

We have followed the agile methodology of software development to make sure every requirement is met. We had continuous development, meetings, and discussions. This project is divided into smaller chunks of functionalities, and they are periodically allocated to us. We are guided to complete them in a continuous flow, test them, and deploy them eventually. The non-functional objectives are considered in every phase of the cycle as they are never described but expected.

1.1.1 Identification

Some companies deliver different kinds of products and provide services to their customers. They must keep track of all the internal working of the stock movements. This chain includes other processes like storing stock at godowns, allocating stock to the sales representatives, importing stock from the main godowns or directly from the company, generating invoices, returning orders for different reasons, etc. Manual systems are quite tedious, time consuming, less efficient, and accurate in comparison to the modern computerized system. So, following are some disadvantages of the old system:

i. Slow data processing
ii. Lots of paperwork.
iii. Time consuming
iv. Less accurate & efficient

This system will provide a good service to the company like better transaction process, easy maintainability that brings bigger profit.

1.1.2 Objective

1.1.2.1 General Objective

The main objective of this research is to improve the manual system of **Prestige** in checking and keeping track of every single stock. The new proposed system can fix the problem in managing records for easier monitoring in how one stock relates to another as well as keeping track of every single transaction done by every single employer in the whole organization.

1.1.2.2 Specific Objective

i. To develop a microservice that tracks every single transaction.

ii. To accurately record, computer and produce reports & challan of every ticket.

iii. To develop a system that enables moving stocks from one godown to multiple as well as assigning stocks to employees.

iv. To develop a system that deals with the day-to-day requirement of any production organization.

1.1.2.3 Functional Objective

1.1.2.3.1 High Priority

Note: Any reference to deductions or additions to the godown points to the use of the database.

i. The system should be robust and able to handle any type of exceptions and send specific error/ response messages to the client.
ii. The microservice shall process the addition or update of the godown with the details in the payload. It should be able to fetch the godown details based on the customer id, employee, zone, or name.
iii. The microservice shall add or update the product stock to the database with the details in the payload. It should be able to fetch the product stock details based on different variables such as status, batch number, godown id, or product id.
iv. The client should be able to move stock from Good to Damaged (or any type possible) and allocate or deallocate stock to and from the sales partner. The microservice shall deduct the stock from the godown based on the orders received based on the client's request.
v. There is a feature where the client can add the stock to the godown from an excel file. The format of the file to be uploaded is specified in the UI and can only be added if the received data is in the given format.

vi. The system shall allow the client to add or update the product batch details with the details provided in the request payload. It should also be able to get the batch details based on the godown id.

vii. The microservice shall fetch all the transactions based on multiple parameters such as transaction date, product id, godown id, and stock, and filter the response to a certain limit, start date, or end date.

viii. The client should see a feature to create internal orders with or without entities according to the payload specified. The system shall have an option to retrieve all the internal orders based on the employee to whom it was assigned, the employee who created it, and the type of order, and filter the response to a certain limit, start date, or end date.

ix. The system should be able to approve and accept internal orders after passing through certain validations. There should be a feature to return an internal order in case of damage or any other valid reasons.

x. The client should be able to create invoice details for the required transactions.

xi. The system shall reflect new or updated changes within a second to the database. This will reduce the number of incorrect records displayed to other employees if both have concurrent requests.

xii. The system shall return orders based on the user's request.

xiii. The system shall validate all the data concerning the design and sends a response to the client in case of any error.

xiv. The whole system should be synchronized for each request on specific functionalities that might hinder the ACID properties.

1.1.2.3.2 Medium Priority

i. The system shall send the saved data in the response body.

ii. The system shall always send a specific error message to the user.

1.1.2.3.3 Low Priority

i. Validate every request for the customer id and employee details in one go rather than validating in each method or steps, thus increasing the code readability**.**

1.1.2.4 Non-Functional Objective

1.1.2.4.1 Reliability

i. The system should maintain minimum of 99% uptime.
ii. And download should be kept to a minimum even if system load exceeds the average.

1.1.2.4.2 Usability

i. The user should be able to operate the system with minimal of 2 hours of training.
ii. User should be able to find the feature they are in need in less than 2 navigations. And most used features should be on home page rather than on different pages that need page reloading.

1.1.2.4.3 Performance

i. The system should be able to support current client requirement of having upwards of 50,000 simultaneous users using the system at once.
ii. The system should be deployed in a load balanced environment, so user accessing the system from another country have minimum latency to ensure faster loading & response time.

1.1.2.4.4 Security

i. System should be password protected so pages and confidential information is restricted to the organization and their employees.
ii. Non-disclosable information such as error stack trace from the backend should be hidden from the user and instead an error code should be displayed.

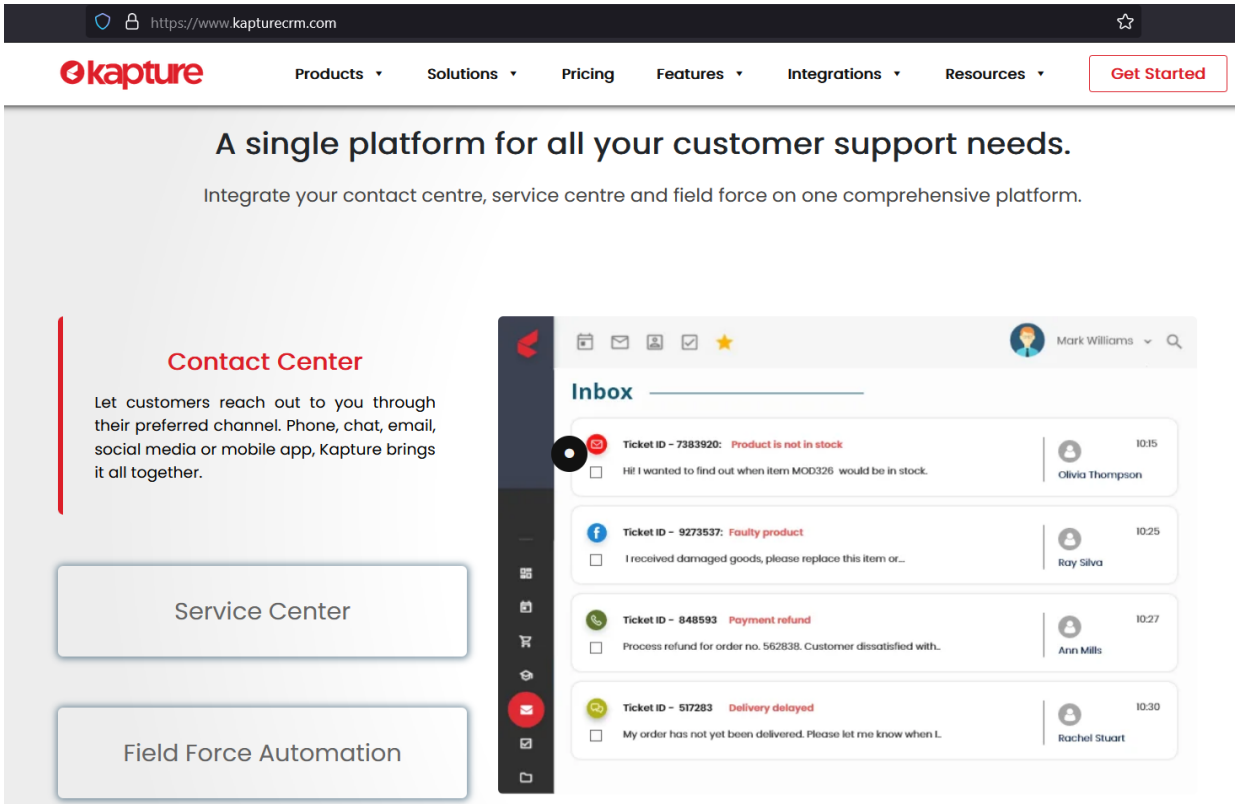
1.2 Parent Organization

For any company to sustain itself, the relationship and engagement with its customers play a vital role. One of the significant tasks in accomplishing so is satisfying the customer's demand within a possible short time and with greater efficiency. **KaptureCRM**, officially registered as **The Adjetter Media Network Private Limited**, is a SaaS-based Customer Support Automation company founded in 2011. Kapture was selected for the prestigious Google Launchpad program, wherein the founders will be trained under Google's mentorship and receive an amount of $50000 as equity-free support. Kapture provides businesses across the globe with all-in-one customer service software. Kapture develops customized tools and powerful integrations for smooth operations. Kapture is the first Mobile-first CRM platform that enables businesses to automate their processes.

One of the best features of Kapture includes the internal ticketing system, wherein the queries of the customers are handled in the form of tickets. The tickets can be from multiple manually created channels, cloud telephony, Kapture's integrated chatbot, etc. According to the priority, the tickets are allocated to the executives in a round-robin manner or one or many other configurations available at Kapture. In addition to this ticketing feature, several additional features like contact centre, service centre, live chat, LMS, Customer Data Management and many more.

Kapture has marked its existence in various industries such as travel, e-commerce, healthcare, cruise management, retail, etc. The company has hit higher growth during the pandemic. When asked how by the CiOL, the CTO of the company, Mr. Vikas Garg, replied, “As with most companies worldwide, Kapture CRM was also impacted by the COVID-19 pandemic. However, we quickly realized that customer enterprises increasingly focused on remote sales and customer service as the world resorted to online purchasing and interactions.”

He also mentioned that 2020 was Kapture’s best year, and it was full of challenges, opportunities, and ventures. There are over 500 businesses that trust Kapture with its excellent customer service operations. The company's accomplishment best depicts its team efforts and vision toward revolutionizing the CRM industry. It justifies its tagline, “Get closer to your customers,” by playing a vital role in giving its clients a fantastic experience.



1.3 Project Scope

This system that we are developing isn't limited to a single client, it can be used by every client who deals with stock management operations. The scope of this system is to provide a user efficient working environment and more output can be generated through this. This system provides a user-friendly interface resulting in knowing each and every usability features of the system. It helps in tracking records so that past records can be verified through them, and one can make decisions based on the past records, and thus this completes the work in a very less time resulting in less time consumption and high level of efficiency.

This system is developed in such a way that even a naïve user can also operate it easily. Also, this system provides a high level of security for data leaking as only limited people can access the database and no change can be made in it until it verifies the user by login-id and password.

1.4 Hardware & Software Specification

1.4.1 Software requirements

a. Java 8 (Language)
b. Spring boot (Framework)
c. Kafka (Data pipeline)
d. IntelliJ (IDE)
e. Linux (Deployment server)
f. MySQL (Database)
g. Apache Subversion (SVN) (Version Control System)
h. Redis (Cache)
i. Postman (Testing)
j. Microsoft Excel (For keeping record of test cases & working)
k. Microsoft Word (Reports)
l. Notepad++ (Scripting)

1.4.2 Hardware requirements

a. Developed on an intel CORE i7 computer with 16 GB RAM x64 architecture
b. Server details are non-disclosable.
c. Users only need a modern web browser.

CHAPTER 2. SYSTEM ANALYSIS

2.1 Problem Analysis

We started research by identifying the need for an online stock management system in the organization. Initially we bounded our research to find the general reasons that emerged the needs of Stock Management System. We used various techniques to collect the data that can clearly give us the overall image of the internal working. However, it wasn't just us as a team, our mentor who interacted with the organization had an opportunity to visit their organization and see how they initially handled the management. After visiting their online system, use of excel and so forth. Basically, the following factors forced them to have an online web system developed for their needs.

i. Cost and affordability
ii. Lack of stock Management
iii. Effective flow of stock transfer and management
iv. Difficulty in monitoring the stocks.
v. No effective way to handle customer tickets.
vi. Manual Internal godown to stock movement using email and google forms.

2.2 Feasibility Study

This software has been tested for various feasibility criterions from various points of view.

I. Economic Feasibility

This system is estimated to be economically affordable. As this system is a high scale web application, there is no need for maintenance by Prestige, all of that is also handled by KaptureCRM. As a bonus of this being a web application, our client Prestige, does not need to spend on high performance desktop and specifically allocate money and time to train employees to do the said work, all of it is handled in cloud-based servers. ‘

II. Technical Feasibility

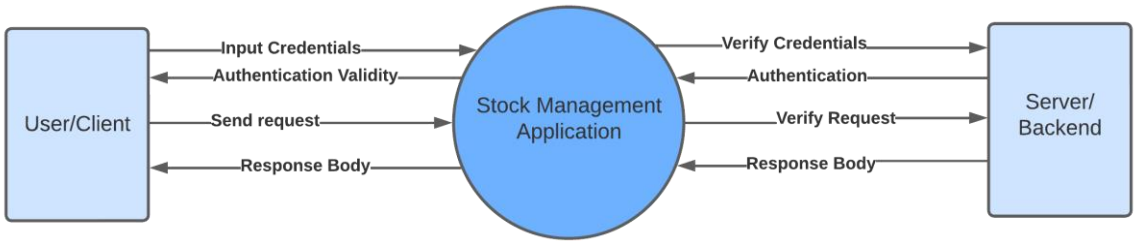
The system that will develop needs users to have knowledge in using a web browser, fortunately, most employees of Prestige are familiar with the internet and have used modern browser, and thus it meets our requirement of having “A modern browser.”

III. Operation Feasibility

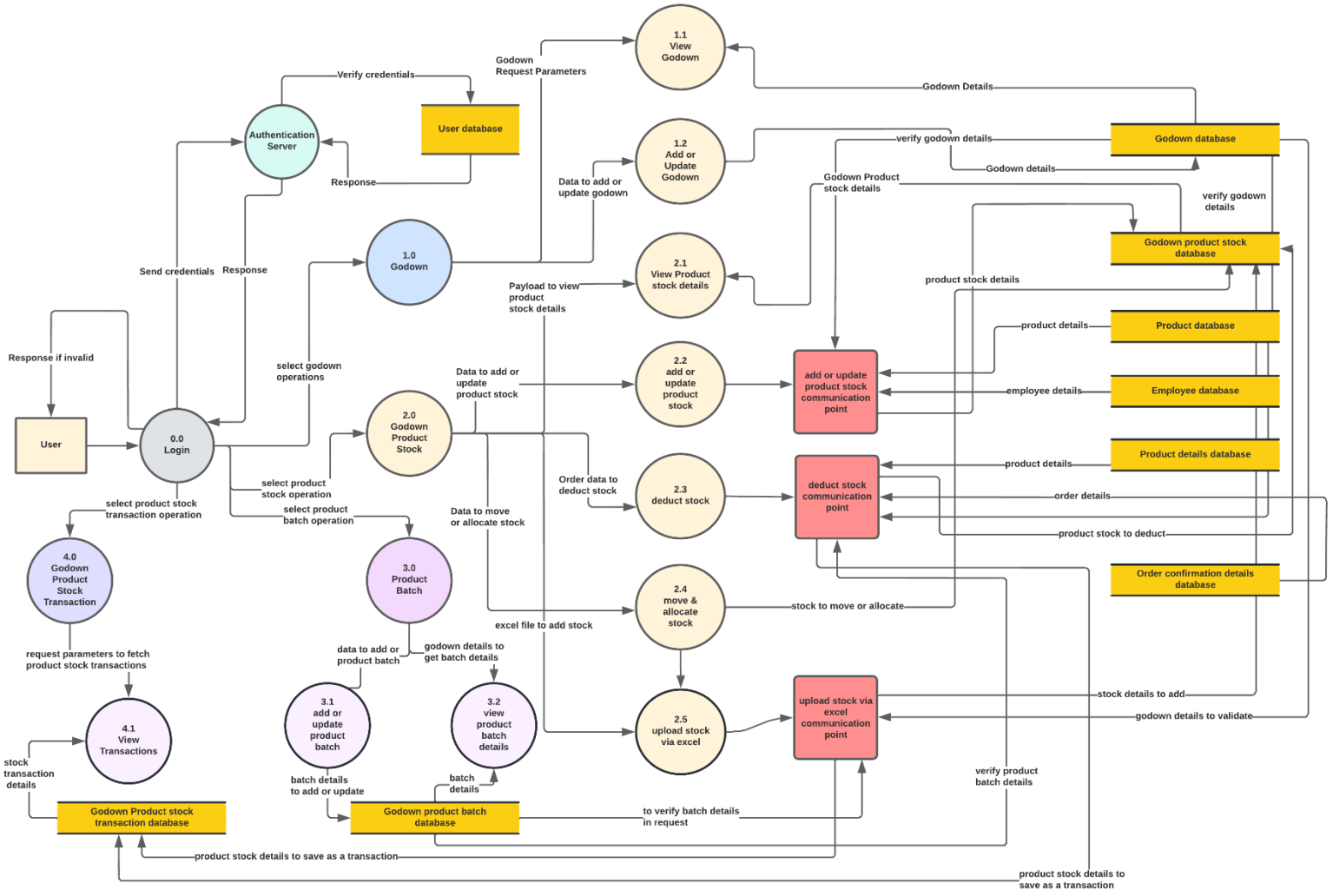
The system that we are developing will solve the problem that our client Prestige encounters, having to maintain each and every transaction manually is a tedious and labour-intensive task. Plus generating digital challans for Prestige clients all over the world is no joke. This proposed system shall automate these tasks so less manual labour is needed to maintain the said tasks and security measures are applied so other than Prestige’s employee, no other party has access to it.

2.3 Data Flow Diagram

2.3.1 Level 0



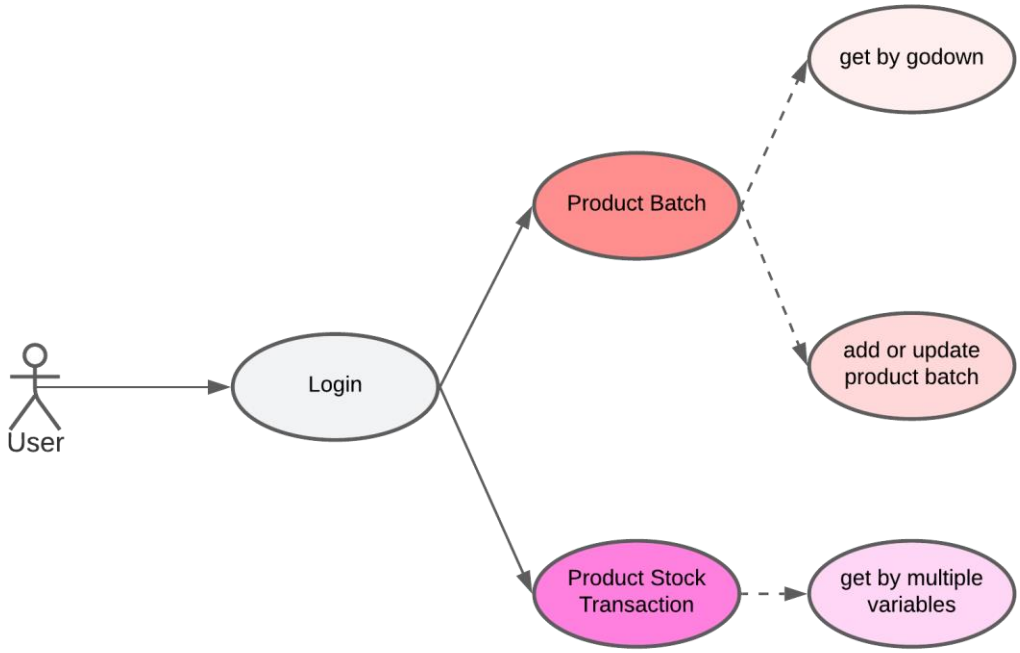
2.3.2 Level 1



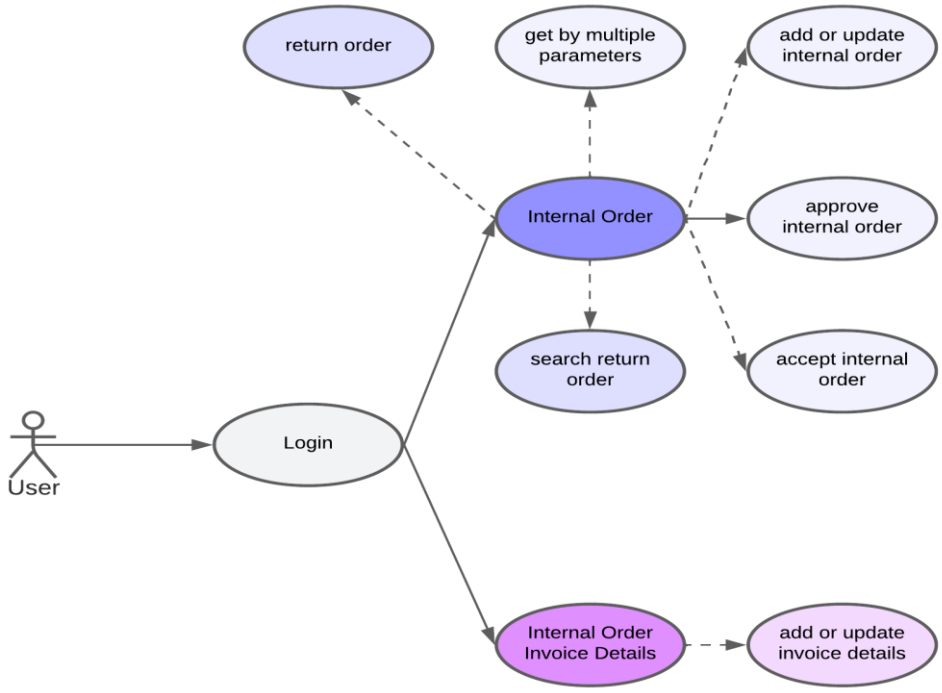
2.4 Use Case Model

a) Godown and GodownProductStock

b) Product Batch and Godown Product Stock Transaction

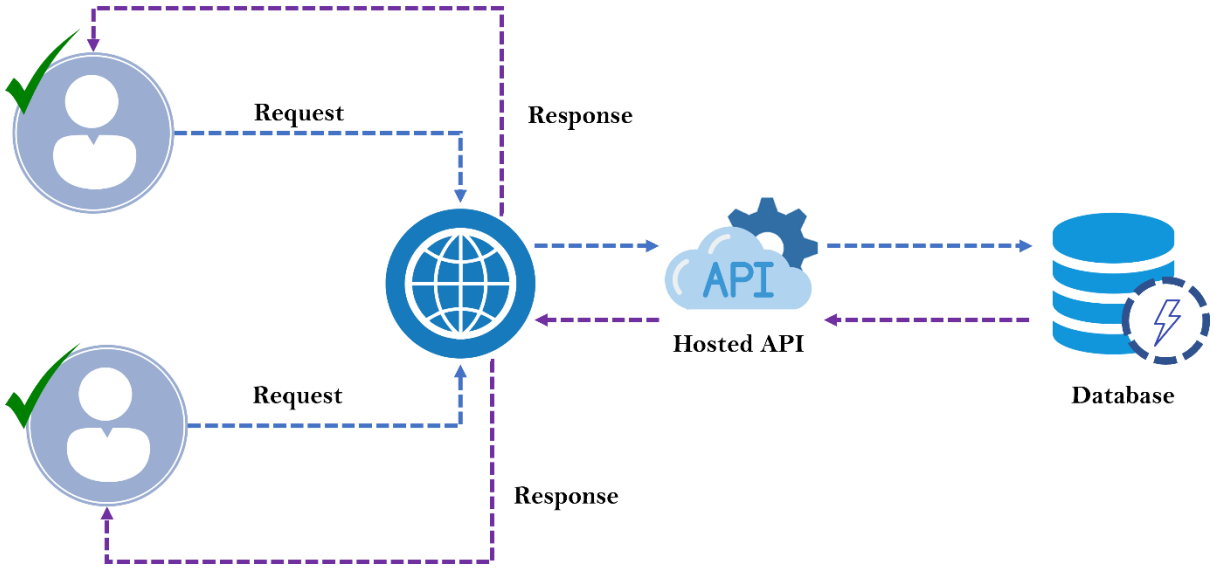


c) Internal Order and Internal Order Invoice Details



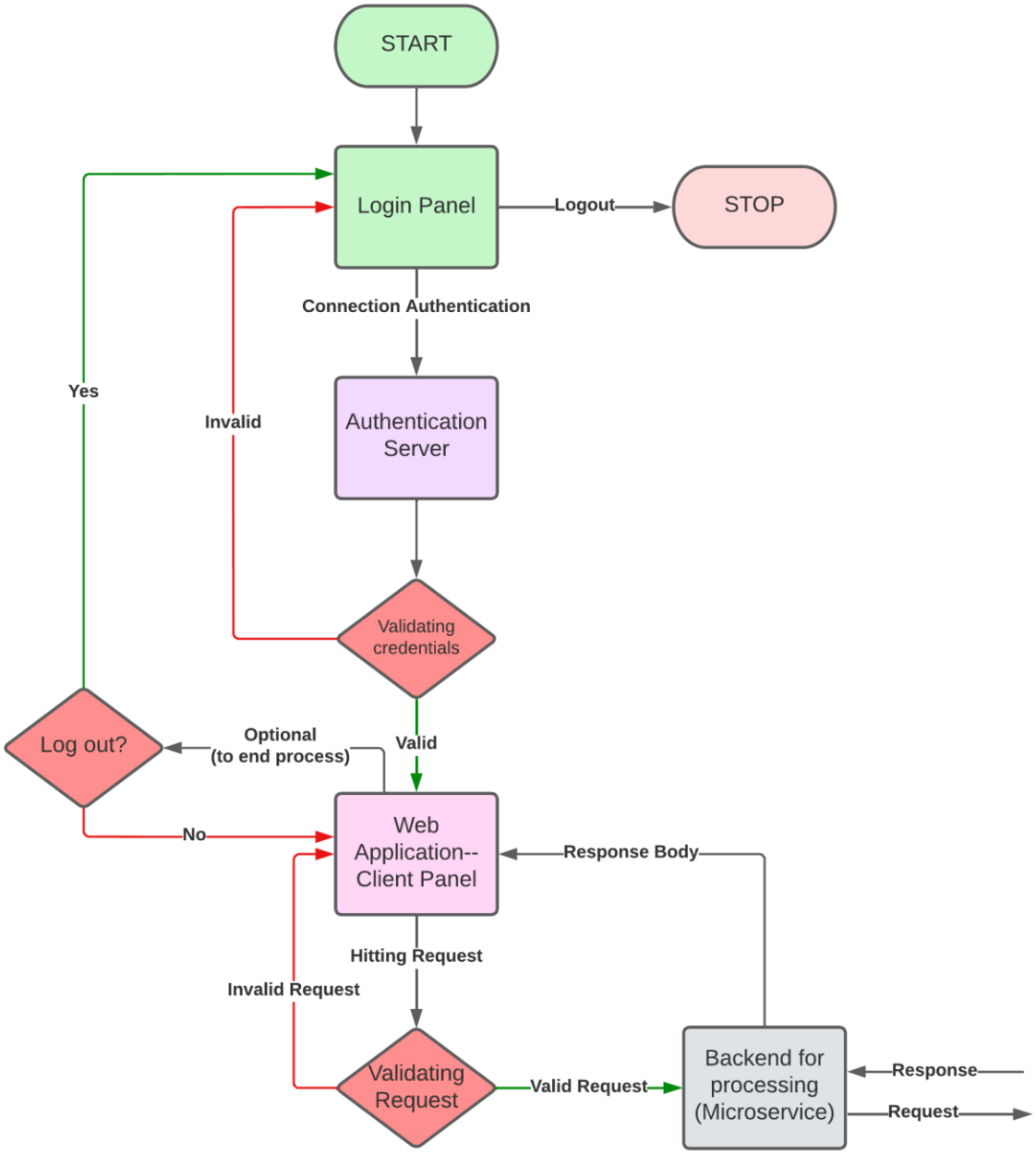
CHAPTER 3. SYSTEM DESIGN

3.1 System Architecture

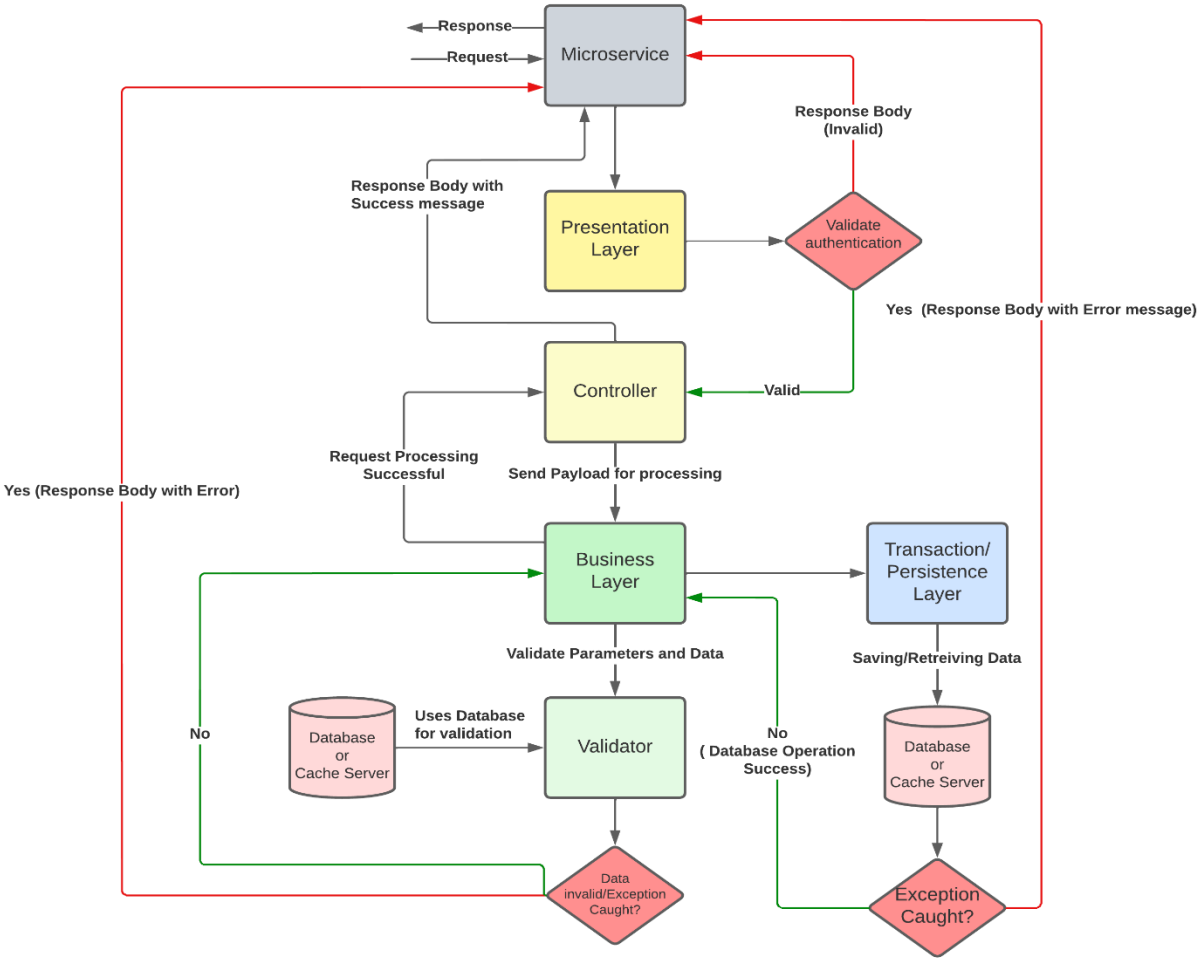


3.2 System Flow Chart

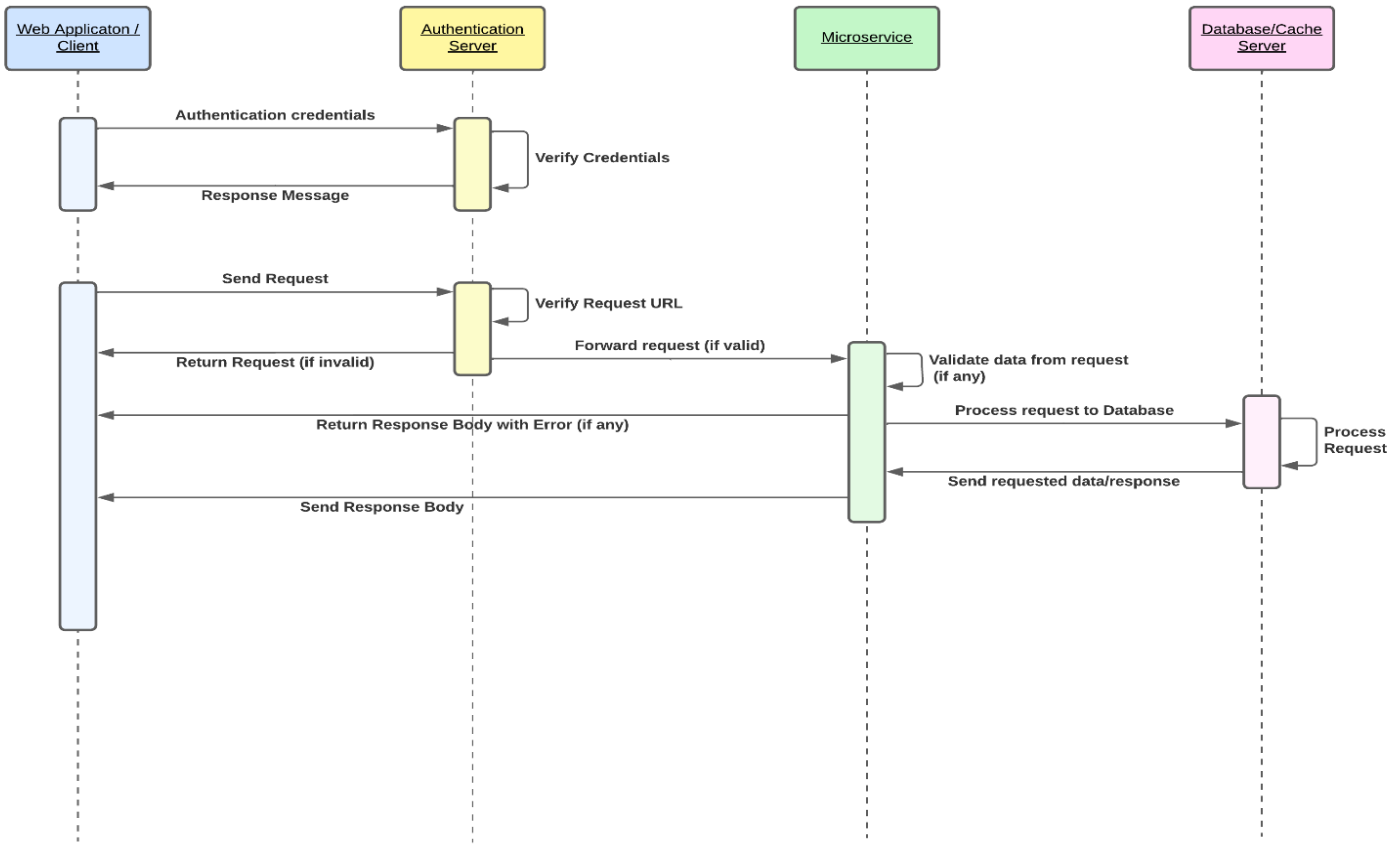
a) User Interface



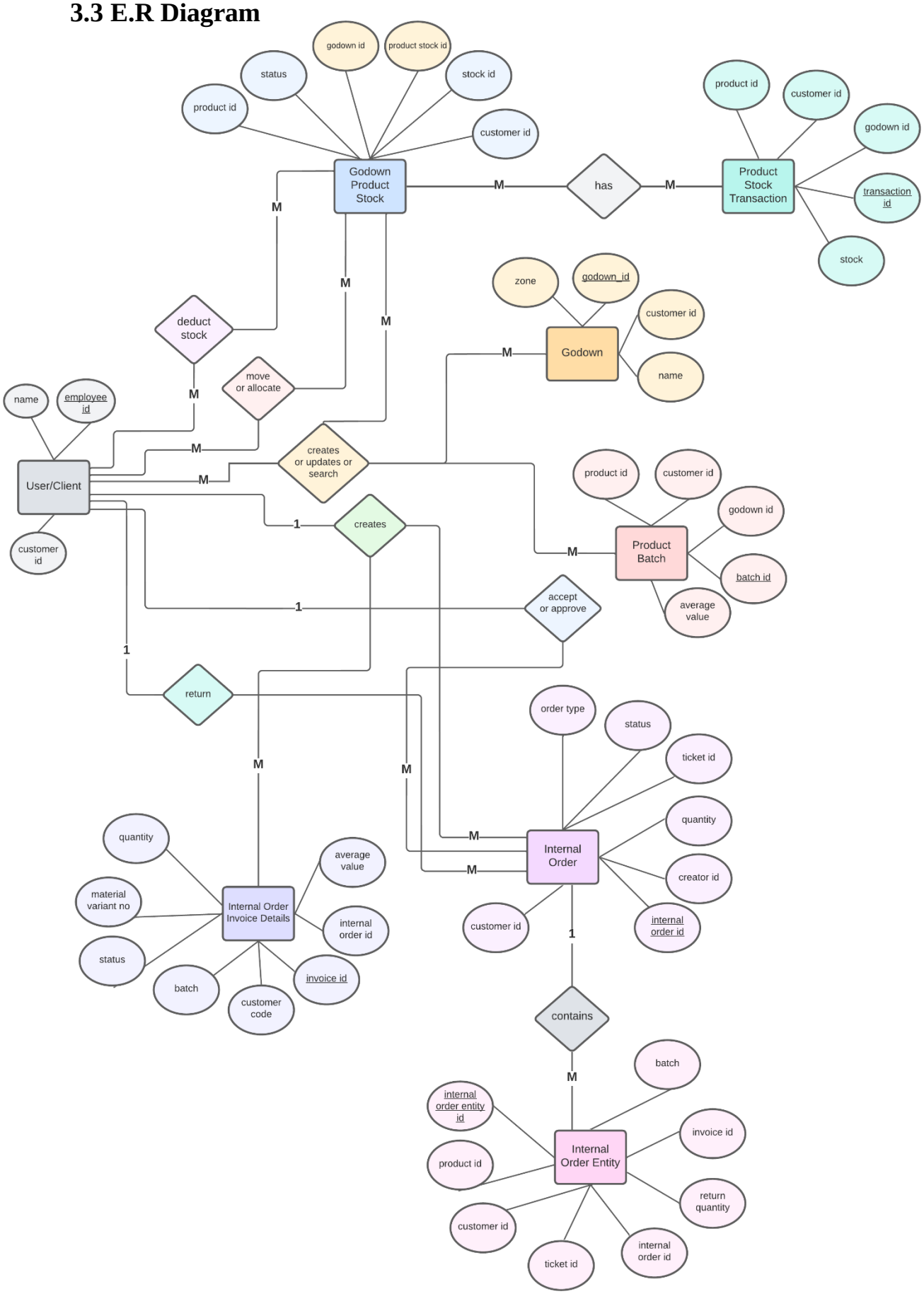
b) Backend or Microservice



c) Sequence Diagram



3.3 E.R Diagram



CHAPTER.4 CODING & TESTING

4.1 Coding

a) Filtering

Ensures each request is validated, if not return Unauthorized message to the request sender.

```
@WebFilter(urlPatterns = {GlobalConstants.GODOWN_PATH + "*",
GlobalConstants.INTERNAL_ORDER_PATH + "*"})
public class GlobalFilter implements Filter {
    private static final Logger LOGGER =
LoggerFactory.getLogger(GlobalFilter.class);

    @Autowired
    private GlobalStore globalStore;
    @Autowired
    private EmployeeCache employeeCache;

    @Override
    public void init(FilterConfig filterConfig) throws
ServletException {
        Filter.super.init(filterConfig);
    }

    @Override
    public void doFilter(ServletRequest servletRequest,
ServletResponse servletResponse, FilterChain filterChain) throws
IOException, ServletException {
        try {
            HttpServletRequest request = (HttpServletRequest)
servletRequest;

            if (SessionDataPreProcessor.setSessionData(request,
globalStore, employeeCache)) {
                filterChain.doFilter(request, servletResponse);
            } else {

servletResponse.setContentType(MediaType.APPLICATION_JSON_VALUE);
                servletResponse.getWriter().write(
                        JsonResponse.setResponse("failed",
ErrorCodes.UNAUTHORIZED_ACCESS.getMessage()).toString());
            }
        } catch (Exception e) {
            LOGGER.error("Exception in doFilter() : ", e);
        }
    }

    @Override
    public void destroy() {
        Filter.super.destroy();
    }
}
```

b) Synchronization

Enable sync for each ClientID so no two clients can make changes to the data at same time. This ensures data flow is consistent and reliable.

```
import java.util.concurrent.ConcurrentHashMap;

public class SyncLockObject {
    public static final ConcurrentHashMap<Integer,
SyncLockObject> clientLockObject = new ConcurrentHashMap<>();
    private int cmId;

    public static SyncLockObject getSyncLockObject(int clientId)
{
        SyncLockObject lockObject;

        if (clientLockObject.get(clientId) == null) {
            lockObject = getInstance(clientId);
            clientLockObject.put(clientId, lockObject);
        } else {
            lockObject = clientLockObject.get(clientId);
        }

        return lockObject;
    }

    private static SyncLockObject getInstance(int clientId) {
        SyncLockObject stockLockObject = new SyncLockObject();
        stockLockObject.setClientId(clientId);
        return stockLockObject;
    }

    public int getClientId() {
        return clientId;
    }

    public void setClientId(int clientId) {
        this.clientId = clientId;
    }
}
```

c) Login form (Employee Key updation and frontend validation) ~ JS Code

```
function updateEmpRegKey(regIdArry, islogout) {

    setCookie("chrome_browser_chk", "0", 1);
    setCookie("chrome_browser_notifi", "0", 1);

    updateFCMEmpRegKey(regIdArry, islogout);
}
function updateFCMEmpRegKey(regIdArry, islogout) {
    $.ajax({
        url: "/employee/update-fcm-registration-key",
        type: "post",
        data: "key=" + regIdArry,
        success: function(msg) {
            console.log(msg);
            if (islogout) {
                if (typeof beforeLogOut == 'function') {
                    beforeLogOut();
                }
                window.location = '/wllogin/employee-
logout.html';
            }
        }
    });
}
$('#logout').on('click', function() {
    updateEmpRegKey('', true);
});
```

4.2 Testing

Testing is vital for the success of any software; no system design is ever perfect. We carried testing in two phases. First phase is during the development of modules, and the second is after completion of the software.

1. White Box Testing

This approach allows testers to inspect and verify the inner workings of a software system — its code, infrastructure, and integration with external systems. We created multiple tests for POSTMAN to test our APIs.

a. Check Godown for add or update

```
var godownReq = pm.environment.get("req-bodies-addUpdate-
godown");
var godownRes = pm.environment.get("res-body-addUpdate-
godown");

if (godownReq != null && godownReq.length > 0){
    postman.setNextRequest("Add or update godown");
} else {
    postman.setNextRequest(null);
}

const postRequest = {
  "url": `localhost:8070/APIURL/ADDUPDATE`,
  "method": "POST",
  header: {
    'Content-Type': 'application/json'
  },
  "body": {
    mode: 'raw',
    raw: pm.environment.get("req-body-addUpdate-godown"),
        },
    };

pm.test("Validate Godown response", function () {
     var body = JSON.parse(responseBody);
     console.log(body);
//     pm.sendRequest(postRequest, (error, response) => {
//   console.log(error ? error : response.json());
  pm.response.to.have.status(200);
  pm.expect(body).to.deep.include(godownRes);
});
```

b. Check ProductBatch for godownId

```
pm.test("Verify get batch by godown-id", function () {
    //response body from the test json file
    var responseTest =
JSON.parse(JSON.stringify(pm.iterationData.get("batch-
response")));

    //response body from the API
    var response = JSON.parse(responseBody);

    if (response != null && response.length > 0) {
        for (var i = 0; i < response.length; i++) {
            console.log(i);

pm.expect(response[i].batch.godownId).to.eql(responseTest.go
downId);
        }
    } else {
        console.log("Empty body found");
       // pm.expect(response).deep.include(responseTest);
    }
})
```

c. Find godown stock by multiple parameters

```
pm.test("Get godown Product stock", function () {
    var response = JSON.parse(responseBody);
    var productId = pm.iterationData.get("productId");
    var stockId = pm.iterationData.get("stockId");
    var status = pm.iterationData.get("status");
    var godownId = pm.iterationData.get("godownId");
    //var list = [productId, stockId, status, godownId];
    for (var i = 0; i < response.length; i++) {
        if (productId > 0) {
            console.log("1");

pm.expect(response[i].productId).to.eql(productId);
        }
        if (godownId > 0) {
            console.log("2");

pm.expect(response[i].godownId).to.eql(godownId);
        }
        if (stockId > 0) {
            console.log("3");
            pm.expect(response[i].stockId).to.eql(stockId);
        }
        if (status != null) {
            console.log("4");

pm.expect(response[i].status).to.include(status);
        }
    }
})
```

2. Black Box Testing

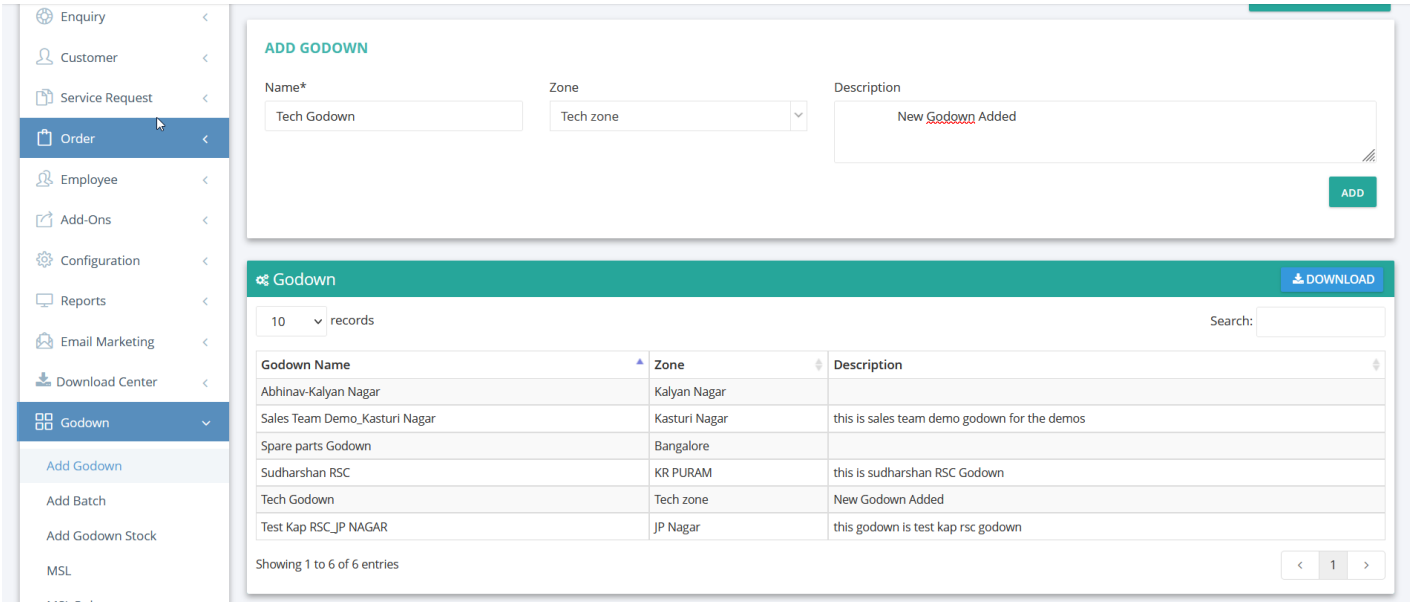
For Blackbox testing which involves testing an application from the user's perspective without any knowledge of its implementation: We ended up assigning an employee from the organization to input sample data and check if the result is as expected.

Steps followed for Blackbox testing: -

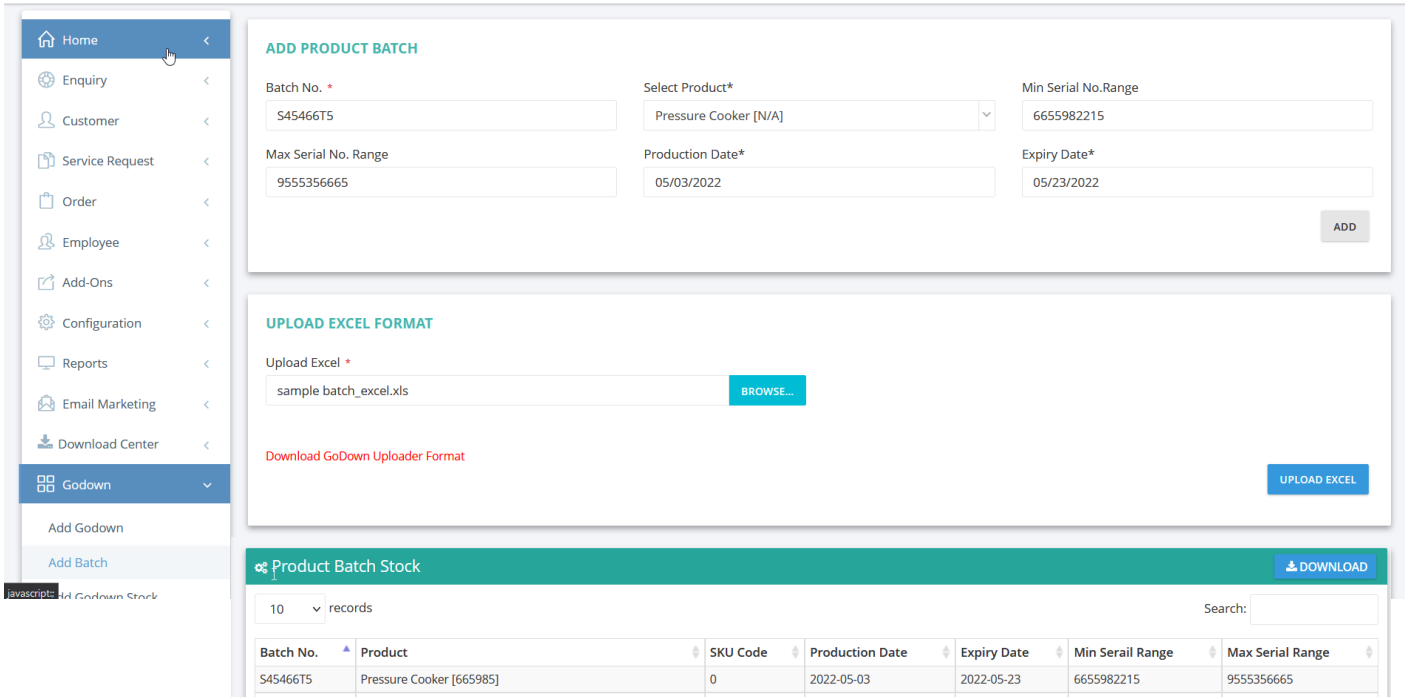
a. Initially, the requirements and specifications of the system were examined.
b. The we choose valid inputs (positive test scenario) to check whether our project processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that it is able to detect them.
c. Tester then constructs and determines expected outputs for all those inputs.
d. The test cases are executed.
e. Software tester compares the actual outputs with the expected outputs.
f. Defects if any are fixed and re-tested.

CHAPTER 5. SAMPLE FORMS AND REPORTS

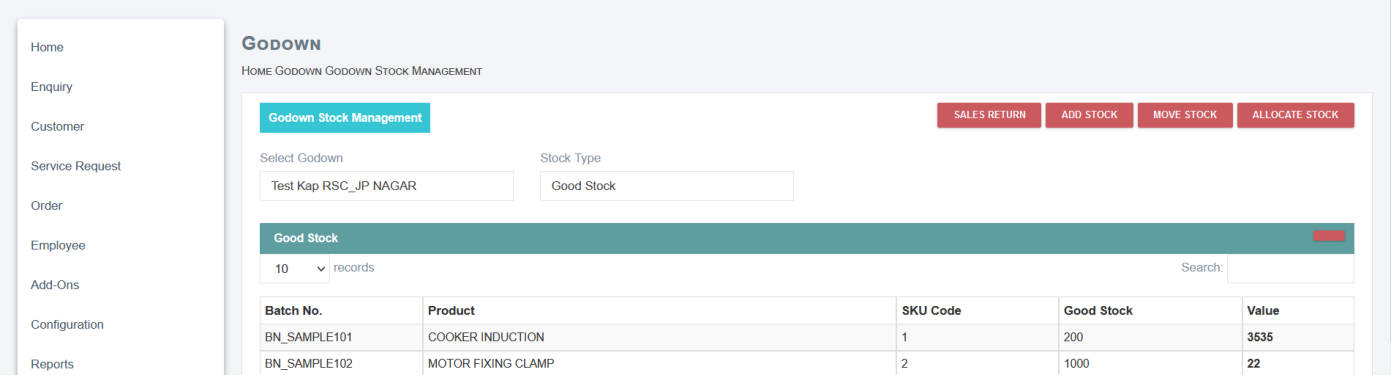
a) Add or Update Godowns



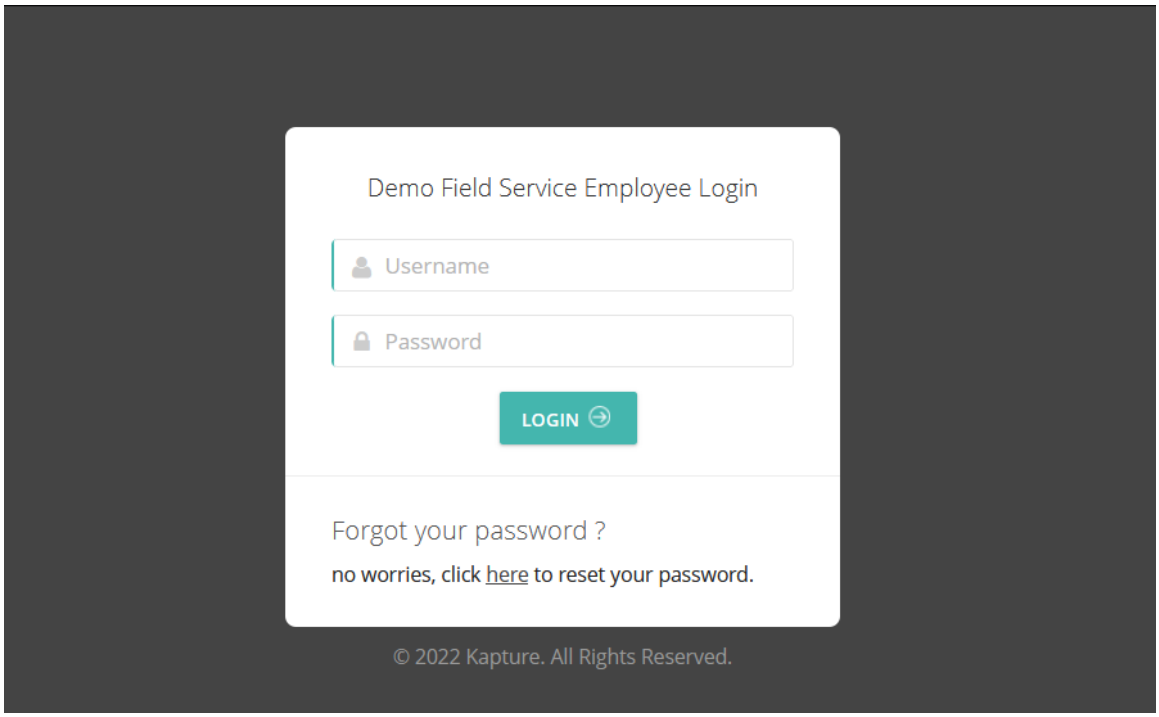
b) Add ProductBatch



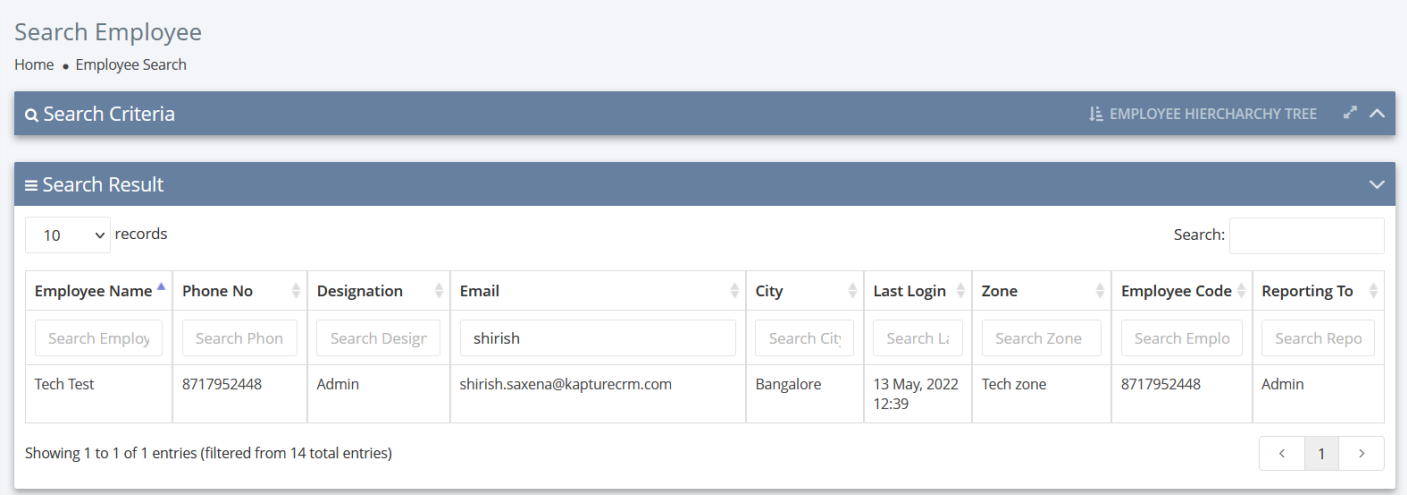
c) Add Update Godown Product Stock



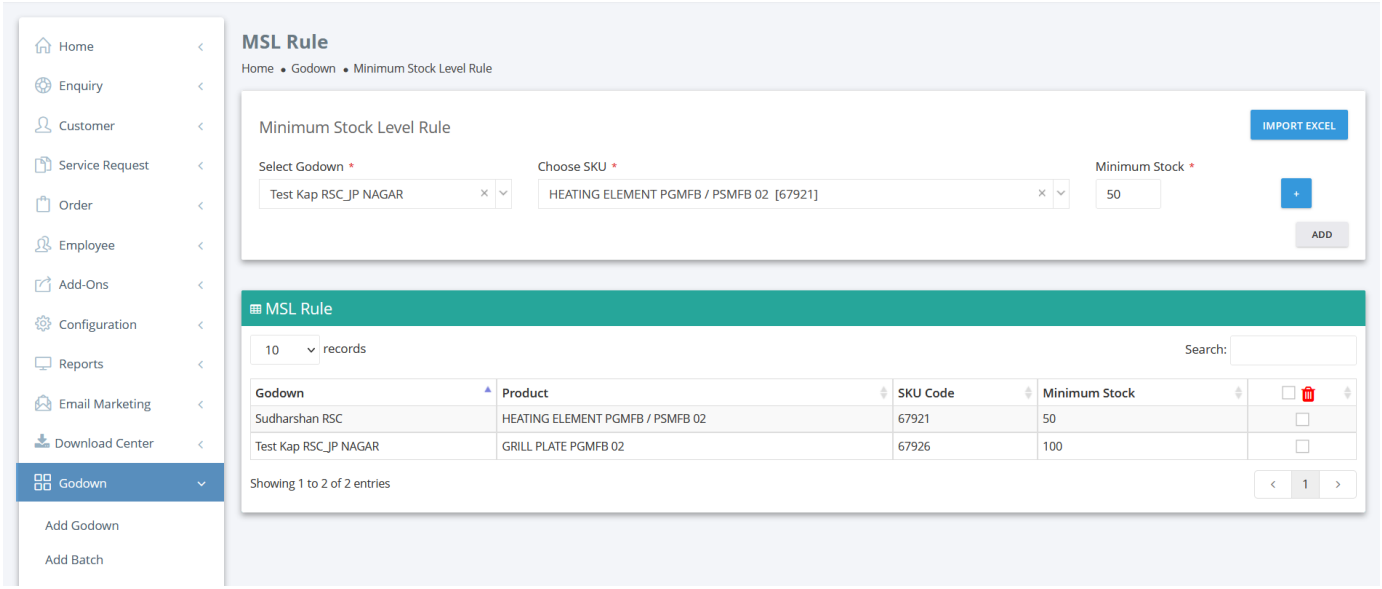
d) Employee Login



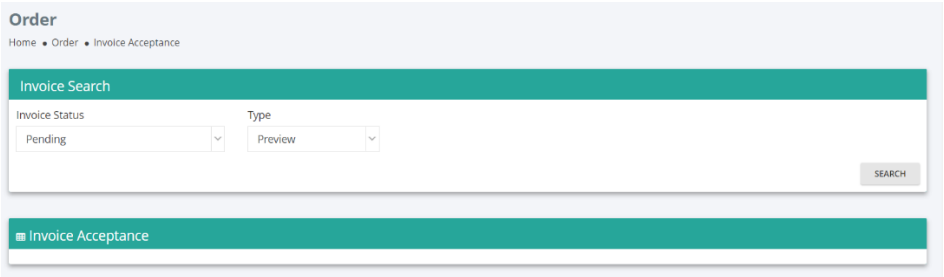
e) Employee search inside an organization



f) Set Minimum stock level to maintain for an organization



g) Accept/Return Order



CHAPTER 6. CONCLUSION

Our project is only a humble venture to satisfy the needs of our client Prestige. Several user-friendly coding has also been adopted. This project shall prove to be a powerful package in satisfying all the requirements of the organization. The objective of software planning is to provide a framework that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have tried on the following points.

i. A description of the background and context of the project and its relation to work already done in the area.
ii. Made statement of the aims and objectives of the project
iii. The description of Purpose, Scope, and applicability
iv. We define the problem on which we are working in the project.
v. We describe the requirement specifications of the system and can be done on these things.
vi. We understand the problem domain and produce a model of the system which describes operations that can be performed on the system.
vii. We included features and operations in details, including screen layouts.
viii. We designed user interface and security issues related to the system- Finally the system is implemented and tested according to the test cases.

Future Scope of the project.

i. We will host the platform on multiple online servers to make it accessible worldwide with low latency and high performance.
ii. Implement the backup mechanism for taking backup of codebase and database on a regular basis on an offload server.
iii. Integrate multiple load balancers to distribute the load of the system. Adding an option to generate pdf challan and invoices.

The above-mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of Stocks and tickets.

CHAPTER 7. LEARNING OUTCOMES

This internship semester and the opportunity has been an amazing experience to progress both technically and personally. Though there are times of difficulties in adaption, I was able to cope with the barriers and accomplish my tasks. Below are my learning outcomes through the internship period:

7.1 Personality Development Outcomes

a. I have learned the office etiquette and attitude required to sustain in the industry.
b. I was able to enhance my interpersonal skills, and other significant skills during the period.
c. I have communicated and collaborated effectively with professionals in the workplace through virtual and physical means.
d. I was able to recommend ideas to improve efficiency by considering viable alternatives.
e. I gained the confidence and skillset to try for opportunities in the domain I was working.
f. I have been appreciated for the work and was directly led to a full-time opportunity at the company following graduation from college.

7.2 Technical Outcomes

a. I was able to integrate theory into practice and differentiate the applicational knowledge from just theory.
b. I was able to break down complex problems into smaller pieces to implement the function more effectively.
c. I can demonstrate the ability to utilize resources by considering different opportunities.
d. I was able to adapt user-friendly and readable coding practices.
e. I was able to grasp the knowledge of the latest technologies in place to achieve the project goal.
f. I have learned new skills like Spring boot, Redis, and Kafka, and was able to apply MySQL and other technologies learned at the college level.
g. I was able to take down requirements and other necessary background information that is further put together to develop the project.
h. I have learned how to analyze the system for bugs and fix them.
i. I was able to write test cases to check the system for any vulnerabilities and errors in functionalities.
j. I have learned how to deploy the microservice on servers and test them.
k. I have learned how the industry works and was able to adapt to their coding practices with reasonable modifications from my end.
l. I was able to create mock servers to mimic the backend system and test the service.

m. I was able to handle communication of the microservice with a remote database server running on Virtual Private Servers.
n. I was able to create VPS and use it learn to Linux and as well deploy any service that requires synchronization of data on two clients.
o. I learned how to make use of Version Control Systems to collaborate and raise issues to create tasks.
p. I have learned to make use of the File Transfer Protocol to transfer files from one server to the other server or local storage.
q. I have learned how documentation is done and was able to prepare one for the system/project we are working on.

BIBLIOGRAPHY

[1] Chen, S., Tang, X., Wang, H., Zhao, H. and Guo, M., 2016, August. Towards scalable and reliable in-memory storage system: A case study with Redis. In 2016 IEEE Trustcom/BigDataSE/ISPA (pp. 1660-1667). IEEE.

[2] Suryotrisongko, H., Jayanto, D.P. and Tjahyanto, A., 2017. Design and development of backend application for public complaint systems using microservice spring boot. Procedia Computer Science, 124, pp.736-743

[3] Bull, C., 2003. Strategic issues in customer relationship management (CRM) implementation. Business process management Journal.

[4] Sandoe, K., 2001. Enterprise integration. John Wiley & Sons.

[5] Newell, F., 2000. Loyalty. com: Customer relationship management in the new era of Internet marketing (Vol. 577). New York: McGraw-Hill.

[6] Aziz Butt, S., Piñeres-Espitia, G., Ariza-Colpas, P. and Tariq, M.I., 2022, January. Project Management Issues While Using Agile Methodology. In International Conference on Lean and Agile Software Development (pp. 201-214). Springer, Cham.

[*] References for Agile Model:
→ https://www.red-gate.com/simple-talk/devops/software-delivery-devops-2/consider-agile-large-projects/
→ https://www.bcs.org/articles-opinion-and-research/agile-on-a-large-scale/