Institute of Technology of Cambodia

Department of Information and Communication Engineering

Delivery Service Management System

Student : Mr. ROTHA DAPRAVITH

Academic Supervisor : Mr. HOK TIN

Internship Advisor : Mr. HOK TIN

Duration : 07th August 2023 – 07th October 2023

Academic Year 2022-2023

Table of Contents

INTRODUCTION

1.1. Presentation of the Internship Company

The **Institute of Technology of Cambodia (ITC)** is a university that was founded in 1964. ITC aims to support and be a resource for all graduate students, as well as for faculty and staff, by establishing connections, enhancing communication and teamwork, and providing thorough research and statistics to educate about graduate education.

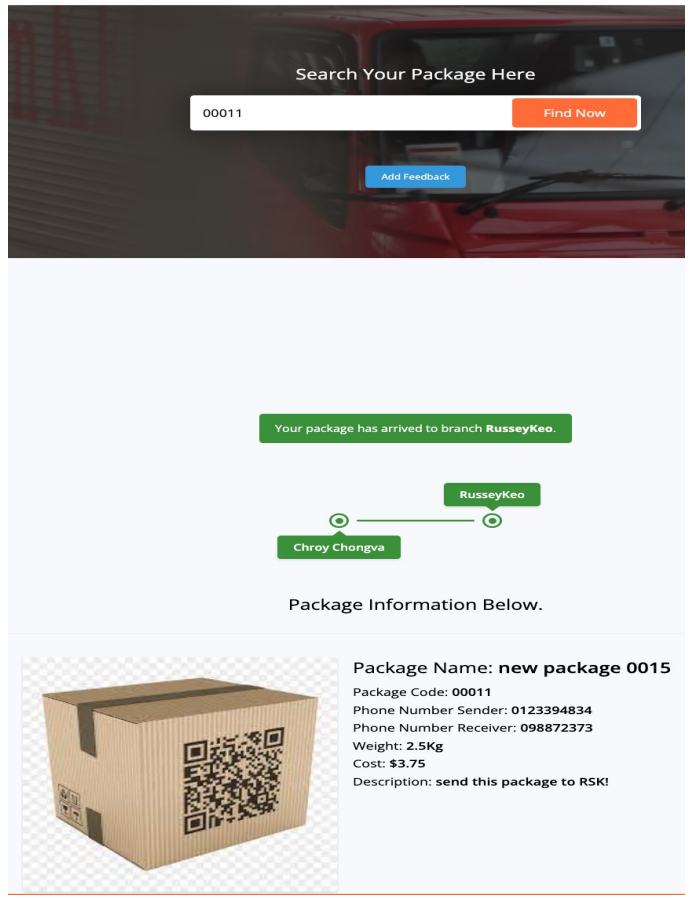
- Address: PO Box 86, Russian Conf. Blvd. Phnom Penh, Cambodia.
- Tel: (+855) 23 880 370 / 982 404
- Email: info@itc.edu.kh
- Website: www.itc.edu.kh/
- Duration: 7th August 2023 – 7th October 2023

1.2. Project Overview

What is Delivery Service Management System?

The **delivery Service Management System (DSMS)** is a web-based platform to developed manage delivery of packages set from original branch to destination branch. It offers the following functionalities include:

- User account management.
- package management.
- **QRCode** Scanner to verify package delivery.



1.3. Problems Overview

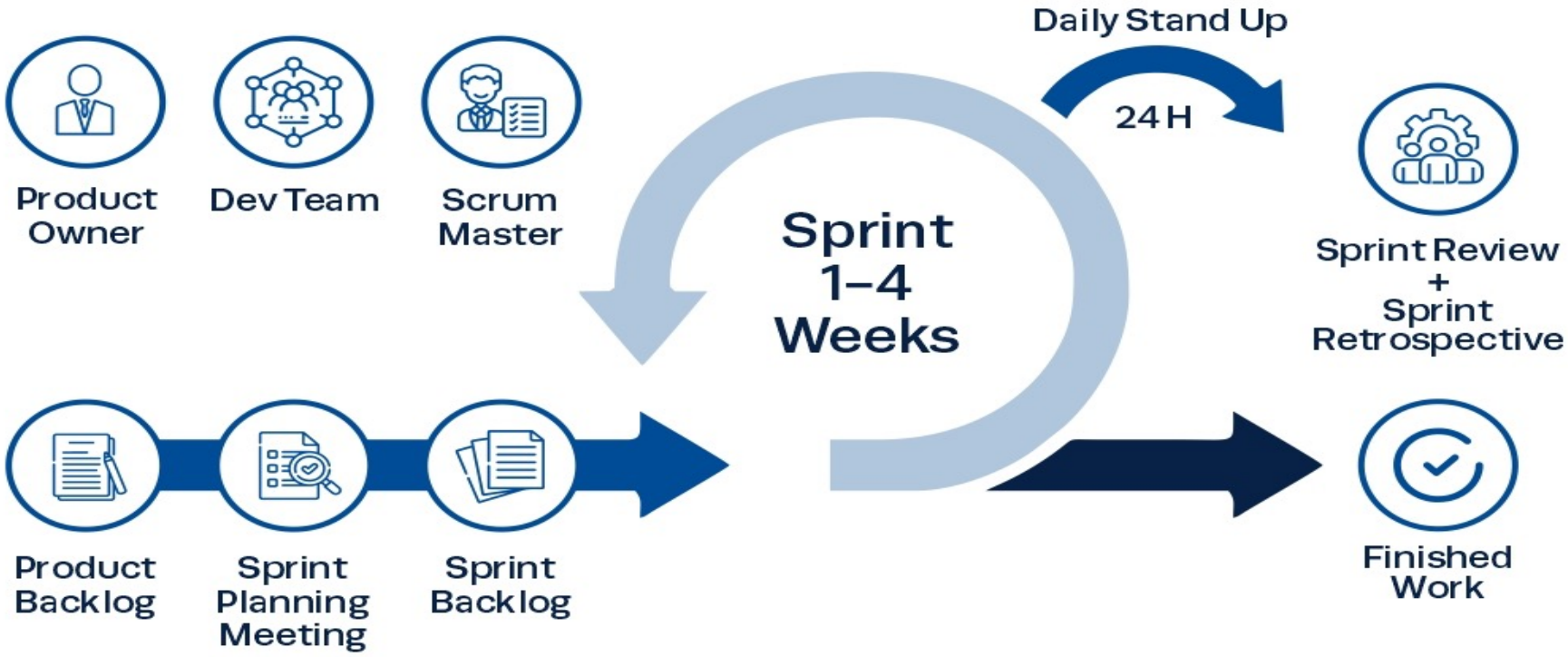
- Delivery process management is challenging to handle
- Manage delivery packages is quite difficult for schedule delivery
- Branch owners have trouble controlling packages and determining delivery destinations
- Customers have trouble finding the package code to track their packages

1.4. Objectives

- Enhance faster delivery process
- User Convience to use system for admin, branch owner and customers.
- Package and location tracking process
- QRCode scanning process to verify package update arrival status

1.5. Scrum Methodology

This project was developed with Scrum Methodology, which is a project management framework that emphasizes teamwork and iterative called **(sprint)** and strengthen collaboration progress toward a well-defined goal. It use a cross-functional team and involves progress tracking tasks.



1.6. Project Team Structure

Roles	Members
Academic Supervisor and Product Owner	Lecturer. **Hok** Tin
Scrum Master/ DevOps	**Toun** Sreynit
Developer Team	**Rotha Dapravith** **Sok Pagnavath**

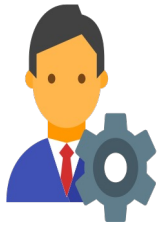
1.7. Planning Project

Tasks	WEEKS								
	1	2	3	4	5	6	7	8	9
Sprint 1									
Sprint 2									
Sprint 3									
Sprint 4									
Sprint 5									
Sprint 6									

- **Sprint 1 :** Learn new technology, UI design and system analysis design database.
- **Sprint 2 :** Initalize project, start working on API, Web and Mobile.
- **Sprint 3,4 and 5**: Implemetation development core functionalities.
- **Sprint 6:** Maintanance testing and document report delivered to department.

ANALYSIS AND DESIGN

2.1. Roles and Features



Administrator

- Authentication
- Dashboard
- Account Management for admin and branch owner
- Edit personal account

Branch Owners

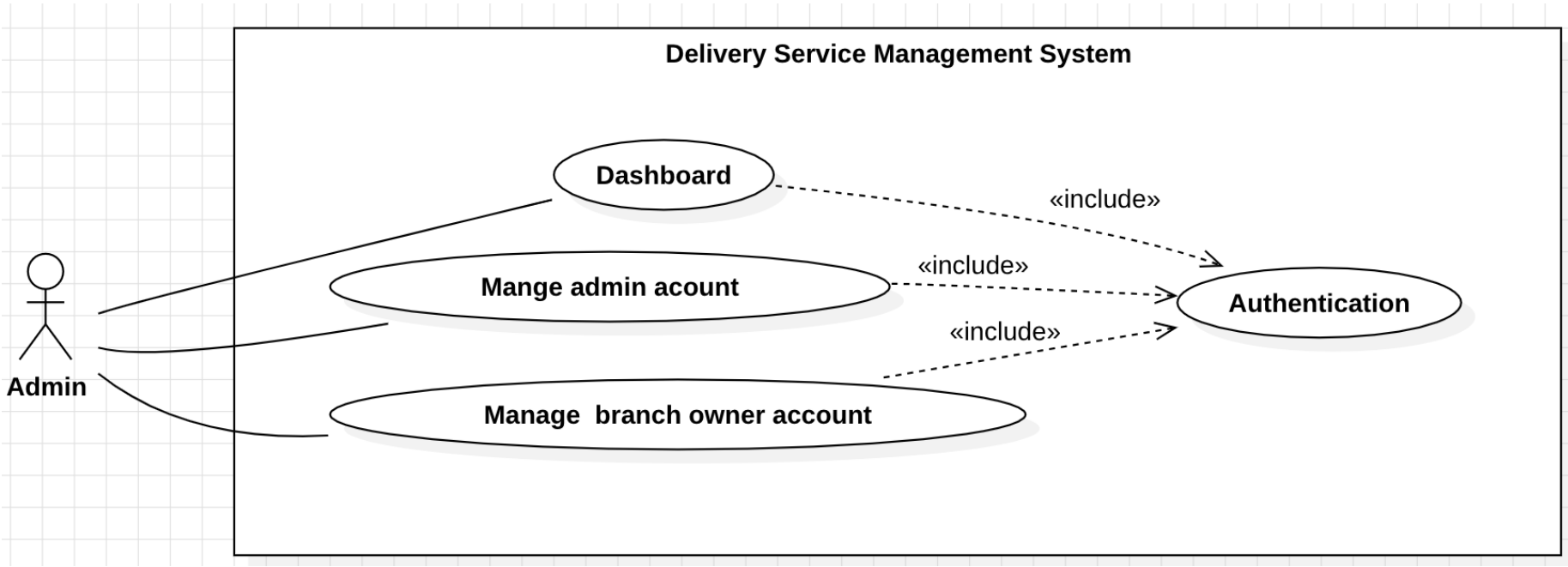
- Authentication
- Dashboard
- Package management
- QRCode management
- Destination management
- View package history
- Edit personal account

Customers

- Find package code
- View details Package
- Feedback rating service

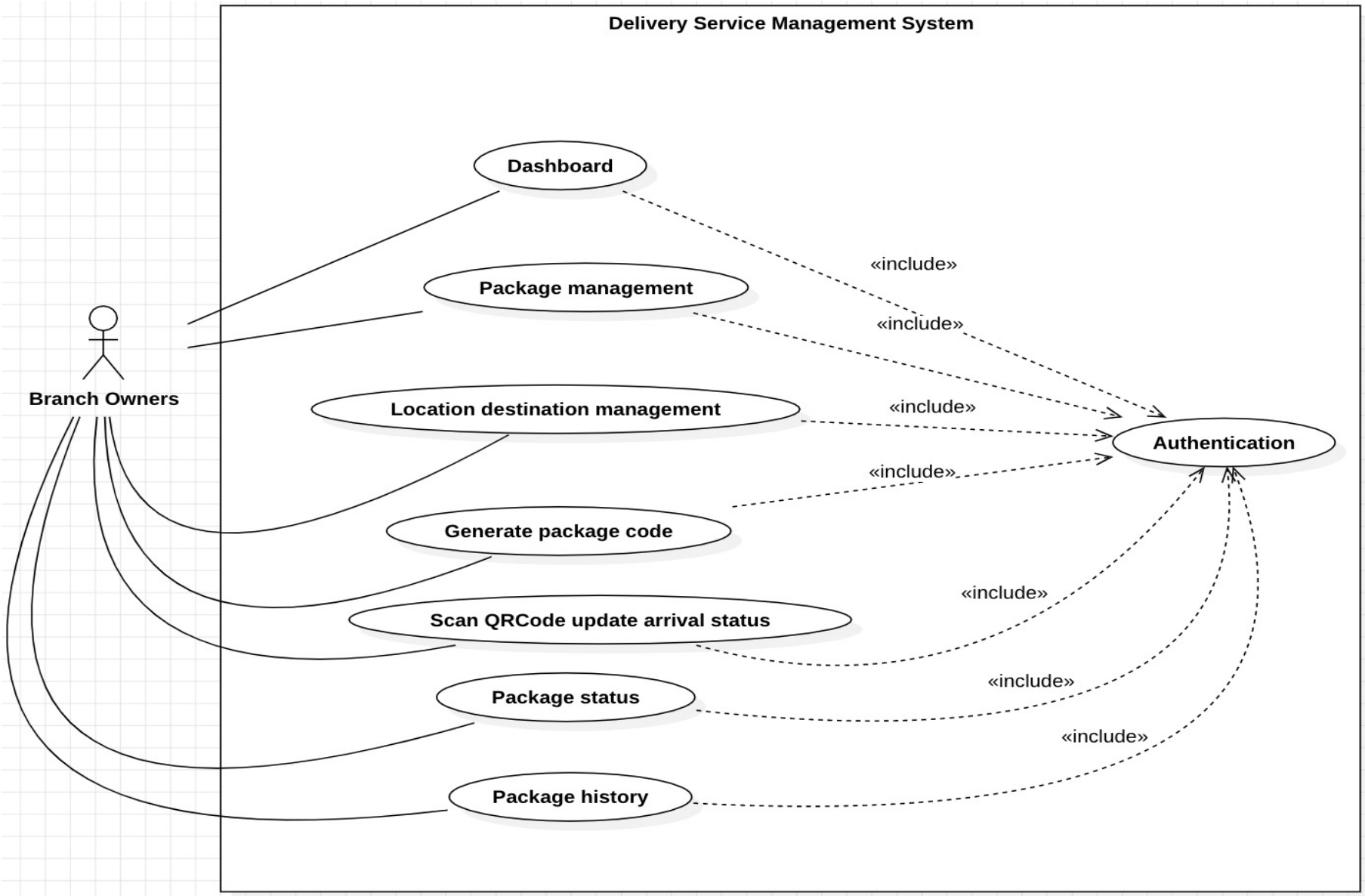
2.2. Use Case Diagram

- Admin Use Case Diagram



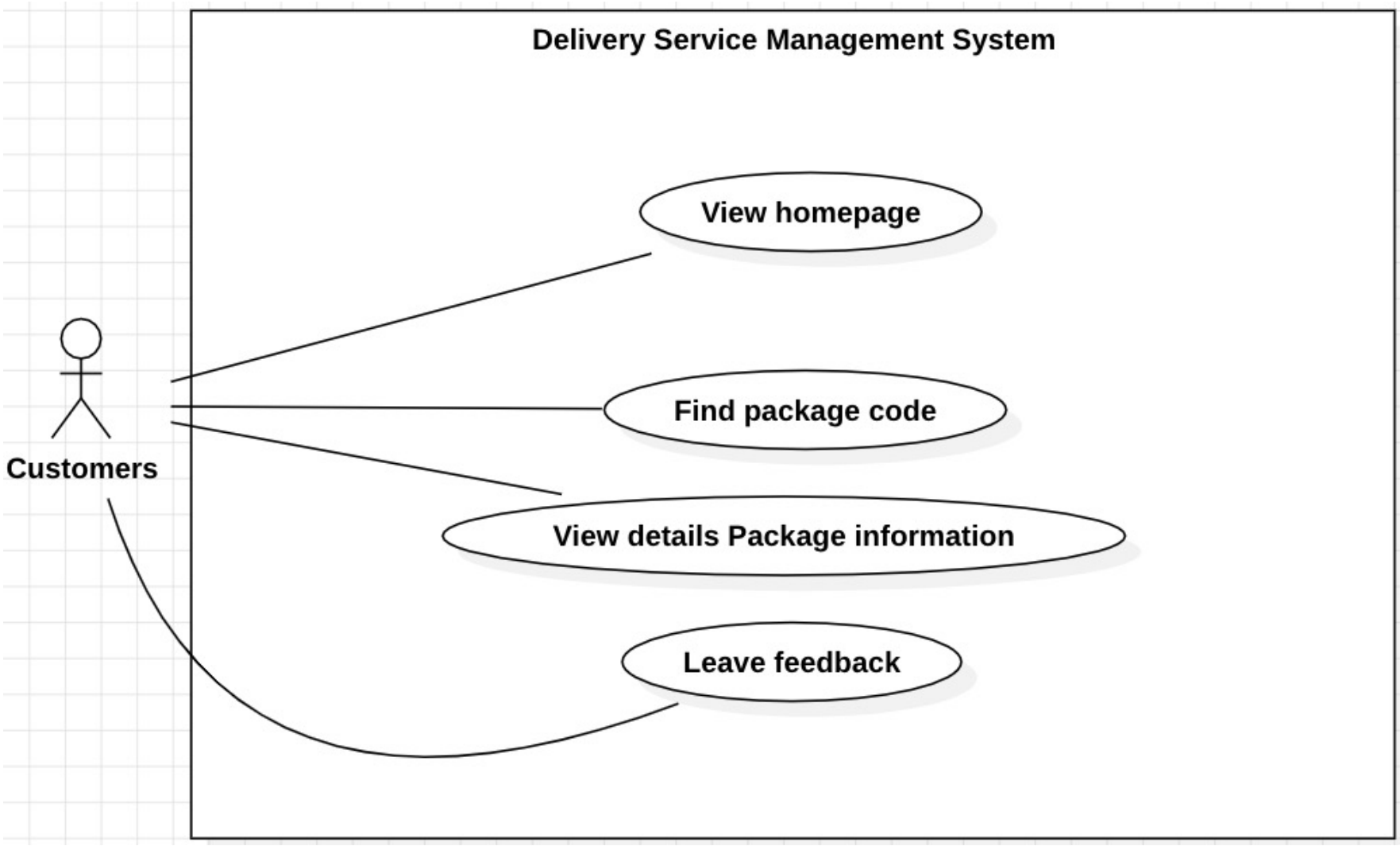
2.2. Use Case Diagram

- Branch Owner Use Case Diagram

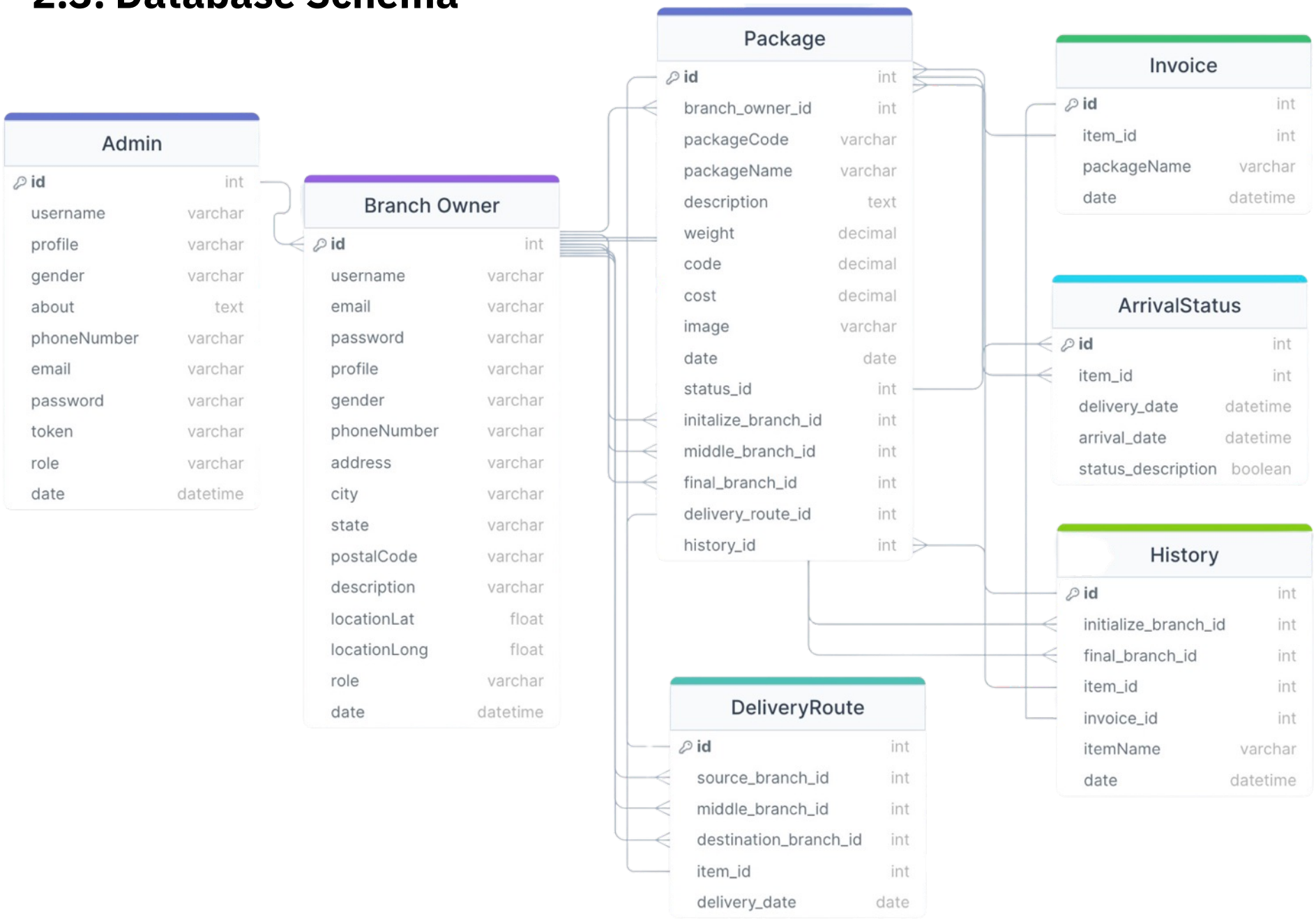


2.2. Use Case Diagram

- Customer Use Case Diagram



2.3. Database Schema

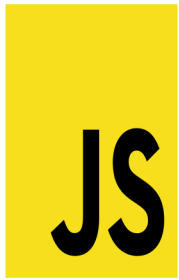


2.4. Choice of Technologies

- **Frontend**

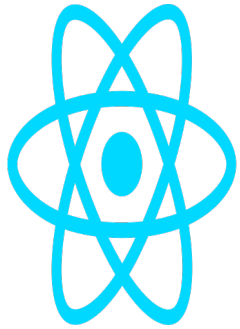

HTML

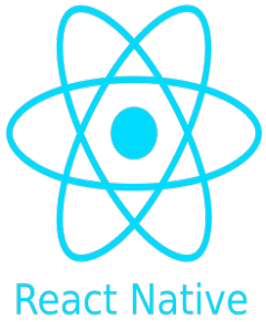

CSS



JavaScript

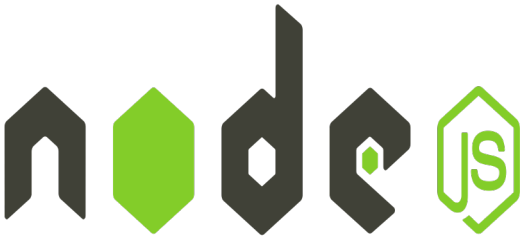
Sass


ReactJS



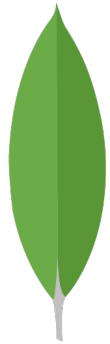
React Native

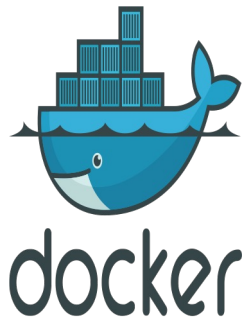
- **Backend**



Node JS

Express JS


MongoDB



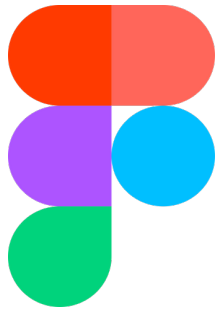
Docker

2.5. Choice of Tools

- Tools

Visual Studio Code

Postman



Figma

Star UML

Draw.io

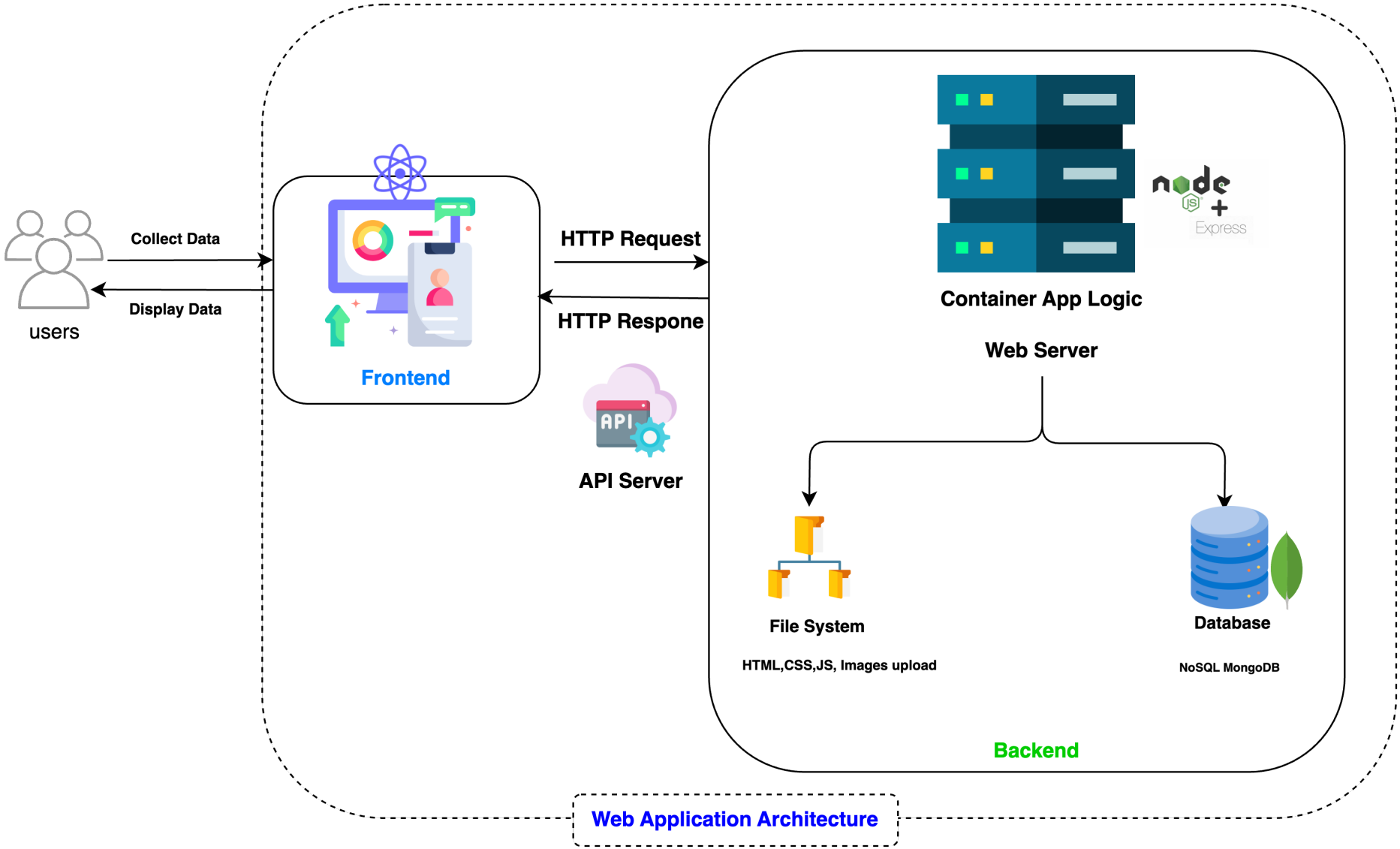
Trello

- Version Control System

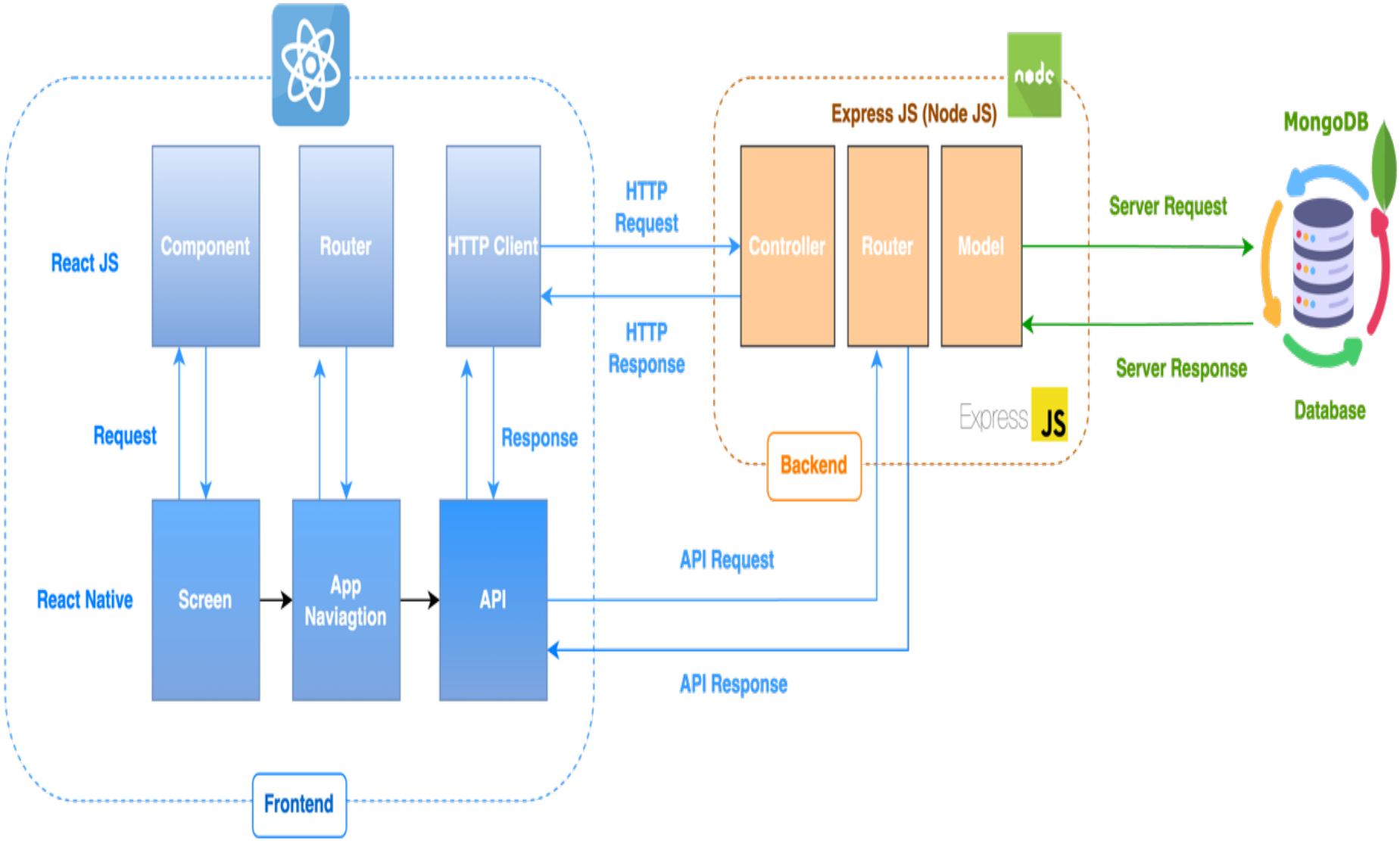
GitLab

SourceTree

2.6. Physical Architecture

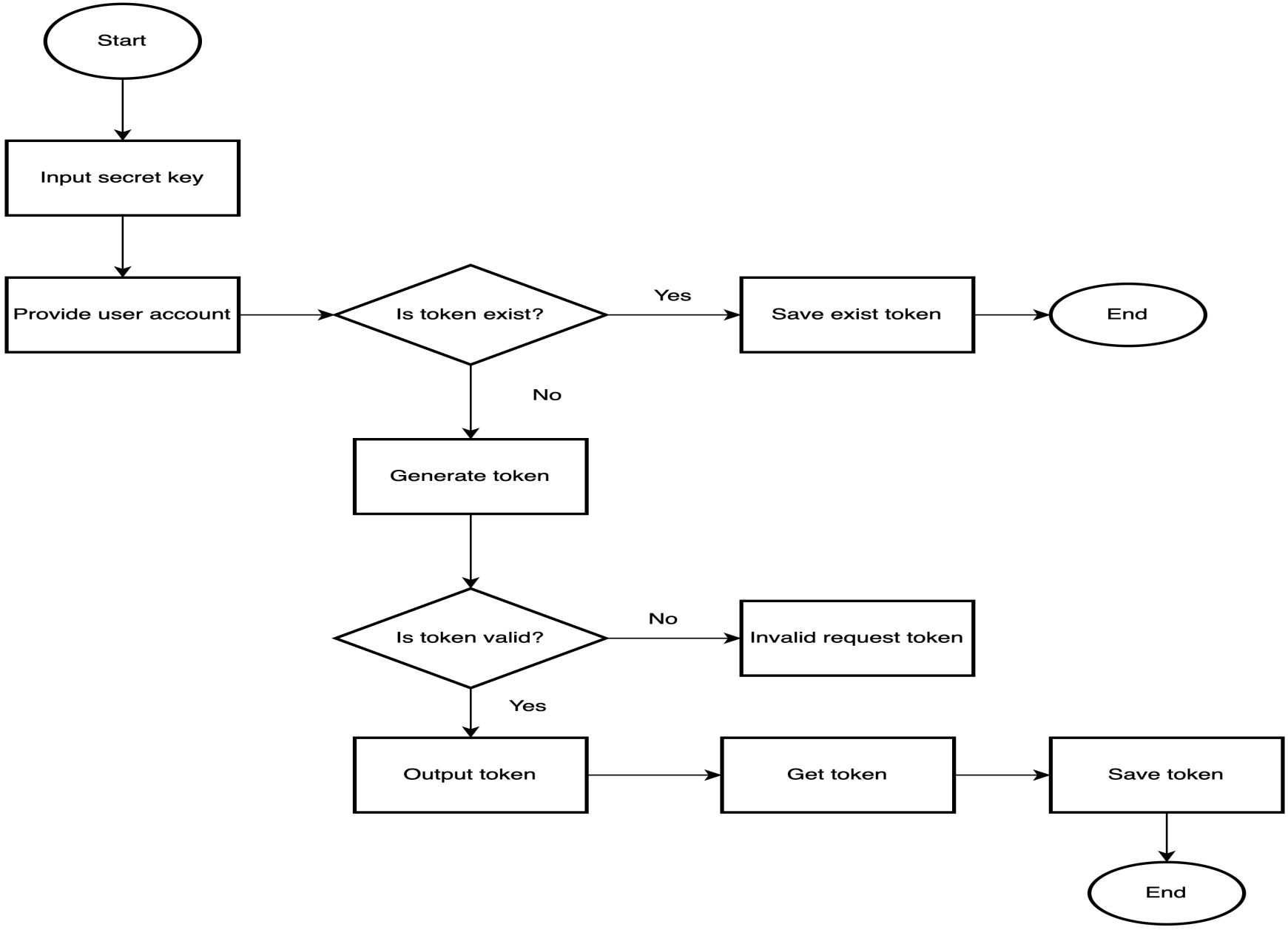


2.7. Logical Architecture



IMPLEMENTATION

3.1. Generate token flowchart



- Validate token for generate

```
const secretKey = getSecretKey(userType);
const existingToken = req.headers.authorization?.split(" ")[1];
const validatedToken = existingToken ? validateAndReturnExistingToken(existingToken, secretKey) : null;

if (validatedToken) {
return res.json({ token: validatedToken });
}
```

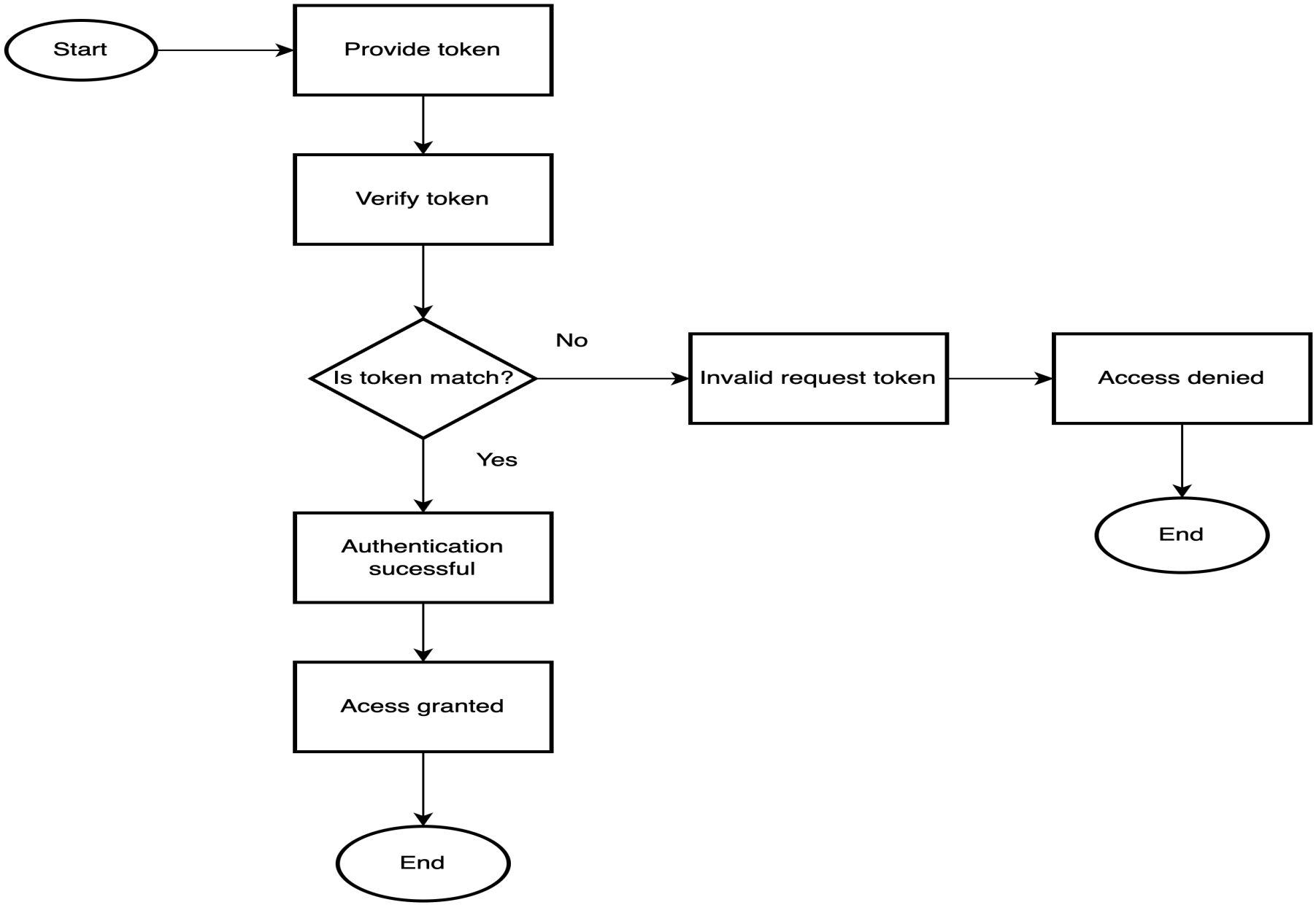
This condition use for input secret key and get valid token of user account.

- Generate new token for user account

```
// If existing token not valid or not provided, generate a new token
const newToken = await generateNewToken(user, secretKey, userType);
res.json({ token: newToken });
} catch (error) {
console.error('Error in generate token:', error);
res.status(500).json({ error: 'Internal server error.' });
}
```

If token not exist, it willgenerate new token for user. Otherwise it is an error during generated token for validate provide wrong user account.

3.2. Verify token flowchart



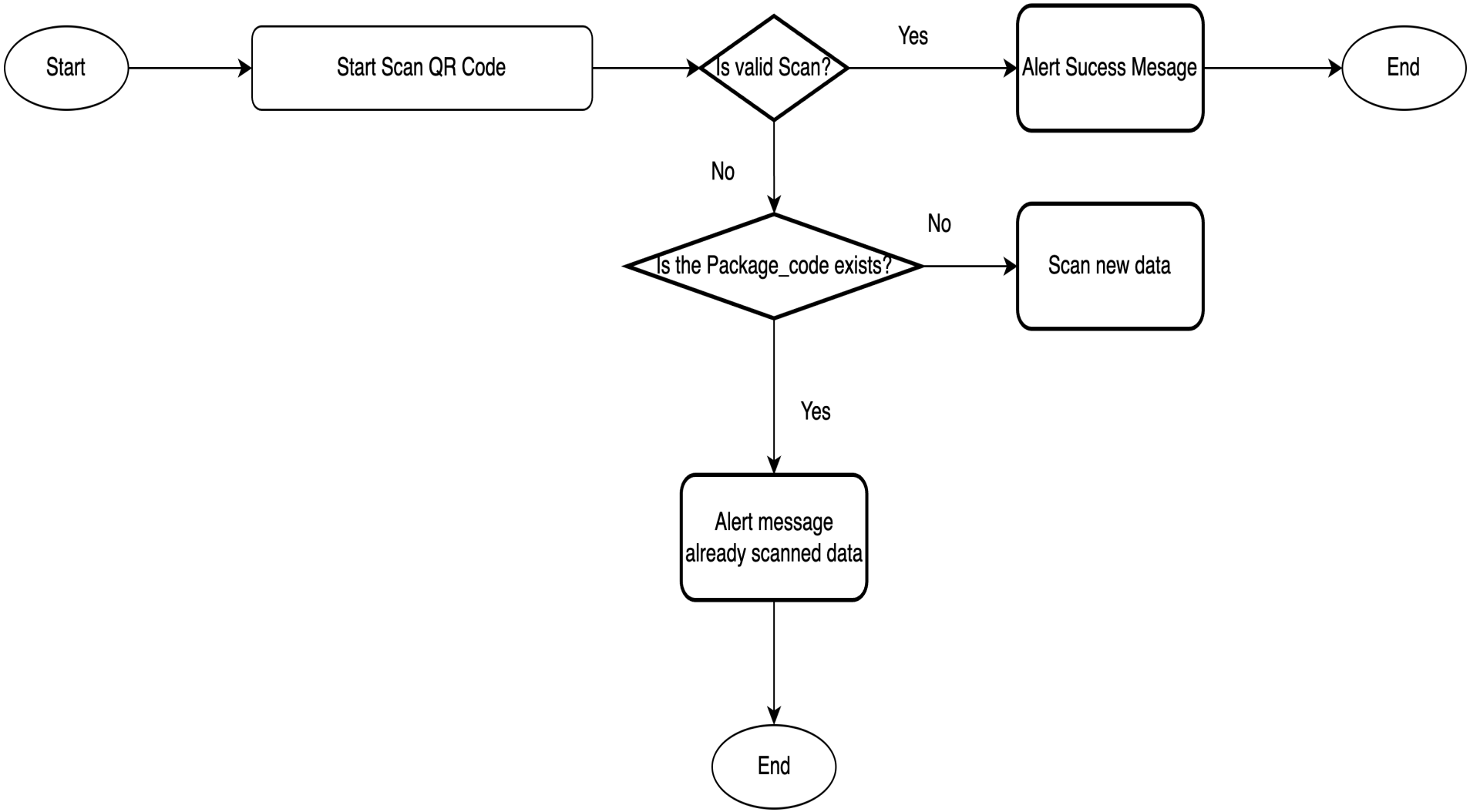
- Verify token of user account

```
const Admin = require("../models/Admin.js");
const jwt  = require('jsonwebtoken');

const adminAuth = async (req, res, next) => {
    try {
        const token = await req.headers.authorization.split(" ")[1];
        const decodedToken = await jwt.verify(token, process.env.ADMIN_SECRET_KEY);
        const user = await decodedToken;
        req.user = user;
        next();
    } catch (error) {
        return res.json({
            message: "Invalid Request!",
            error
        })
    }
}

module.exports = { adminAuth}
```

After token was generated use to verify token provide by user account.

When token are not verify, server display invalid request for wrong validate token.

3.3. Flowchart for QR Code Scanner



- QRCode scanner permission

```
const Scanner = ({ onScan }) => {
  const [hasPermission, setHasPermission] = useState(null);
  const [scanned, setScanned] = useState(false);
  const scannerRef = useRef(null);

  useEffect(() => {
    (async () => {
      const { status } = await BarCodeScanner.requestPermissionsAsync();
      setHasPermission(status === 'granted');
    })();
  }, []);
```

OnScan is the callback function use to scan QRCode with permission before start scanning.

- Handle to scan barcode

```
const handleBarCodeScanned = ({ type, data }) => {
  setScanned(true);
  onScan(data); // Pass the scanned data to the parent component
};

return (
  <BarCodeScanner
    ref={scannerRef}
    onBarCodeScanned={scanned ? undefined : handleBarCodeScanned}
    style={{ flex: 1 }}
    barCodeTypes={[BarCodeScanner.Constants.BarCodeType.qr]}
    focusDepth={0} // Change the focus depth (you can experiment with different values
    focusStyle={{
      borderColor: 'blue', // Change the border color when focused
      borderRadius: 16, // Change the border radius when focused
      borderWidth: 3,
    }}
  />
);
```

BarCodeScanner is configured to scan only QRCode with custom styles of scanner.

CONCLUSION

4. Overall Result

4.1. Admin Roles

Roles	Features	Status	Responsible by
Admin	Admin Dashboard	✓	**ROTHA Dapravith**
	Authentication	✓	**ROTHA Dapravith**
	Branch Owner Account Management	✓	**SOK Pagnavath**
	Change Password	✓	**ROTHA Dapravith**
	Change Personal Information	✓	**SOK Pagnavath**
	List of Admin	✓	**ROTHA Dapravith**
	Logout	✓	**ROTHA Dapravith**
	Monitoring all Branches	✓	**ROTHA Dapravith**
	Update Profile	✓	**SOK Pagnavath**

4.2. Branch owner Roles

Roles	Features	Status	Responsible by
Branch Owners	Login Autehntication	✓	**ROTHA Dapravith**
	Dashboard	✓	**SOK Pagnavath**
	Package management	✓	**SOK Pagnavath**
	Manage Delivered Product	✓	**SOK Pagnavath**
	View all Pending Items	✓	**SOK Pagnavath**
	View all Incoming Items	✓	**SOK Pagnavath**
	View Delivered Items	✓	**SOK Pagnavath**
	Set Source Destination	✓	**SOK Pagnavath**
	Generate Package Code into QR Code	✓	**ROTHA Dapravith**
	Scan QR Code	✓	**ROTHA Dapravith**
	Mark Delivered Product	✓	**SOK Pagnavath**
	Item History	✓	**SOK Pagnavath**

4.3.Customer Roles

Roles	Features	Status	Responsible by
Customers	Find Package Code	✓	ROTHA Dapravith
	View Details Package	✓	ROTHA Dapravith
	View Tracking Branch Location	✓	SOK Pagnavath
	Leave Feedback	✓	ROTHA Dapravith
	Advance Search Filter	×	SOK Pagnavath
	Display Location	×	SOK Pagnavath

4.4.Backend API Tasks

Modules	Features	Status	Responsible by
API Auth	Register Admin's Account	✓	ROTHA Dapravith
	Login Admin's Account	✓	ROTHA Dapravith
	Register Branch Owner's Account	✓	SOK Pagnavath
	Login Branch Owner's Account	✓	SOK Pagnavath
	Get all list of Admin	✓	ROTHA Dapravith
	Get all list of Branch Owner	✓	ROTHA Dapravith
	Update Profile User	✓	ROTHA Dapravith
	Change Password	✓	ROTHA Dapravith
	Encrypt and Decrypt Token	✓	ROTHA Dapravith
	CRUD user of Admin	✓	ROTHA Dapravith
	CRUD user of Branch Owner	✓	SOK Pagnavath
API Package	CRUD Package of Products	✓	SOK Pagnavath
	Get all list of Package	✓	SOK Pagnavath
	Get Package by Images	✓	SOK Pagnavath
	CRUD Package history	✓	SOK Pagnavath
	Get all pending Package	×	SOK Pagnavath
	Get all incoming Package	×	ROTHA Dapravith
	Get all delivered Package	×	SOK Pagnavath
	Mark not Delivered	✓	SOK Pagnavath
	Mark Delivered	✓	SOK Pagnavath
	Get all Package history	✓	SOK Pagnavath
API Feedback	Create customer's feedback	✓	ROTHA Dapravith
	Get all customer's feedback	✓	ROTHA Dapravith

4.5. Difficulties

- Challenges with **New technologies**.
- More debug **requirement in code**.
- User interface design.
- Quick time to **develop the system**.

4.6. Experiences

- Learn new technologies and gain knowledge coding skill.
- Understand about **software development life cycle**.
- Soft-skill communication with teammates.

4.7. Perspectives

- Improve system website **performance and high security**.
- Update Scanner mobile app more attractive.
- Completed with undone tasks.

References

- JavaScript Programming Language Reference Website, from https://www.javascript.com/
- Mern Stack: A Complete Guide, from https://www.mongodb.com/languages/mern-stack-tutorial
- Introduction to HTML, from https://www.w3schools.com/html/html_intro.asp
- CSS Introduction, from https://www.w3schools.com/css/css_intro.asp
- JavaScript.com, from https://www.w3schools.com/js/default.asp
- Sass, from https://sass-lang.com/
- React JS, from https://legacy.reactjs.org/docs/getting-started.html
- React Native, from https://reactnative.dev/docs/environment-setup
- NodeJS, from https://nodejs.org/en
- Express JS, from https://expressjs.com/
- MongoDB, from https://learn.mongodb.com/learning-paths/introduction-to-mongodb
- Docker, from https://www.docker.com/
- GitLab, from https://about.gitlab.com/
- SourceTree. from https://www.sourcetreeapp.com/
- Visual Studio Code, from https://code.visualstudio.com/
- StarUML, from https://staruml.io/
- Draw.io, from https://drawio-app.com/
- Postman, from https://www.postman.com/
- Trello, from https://trello.com
- Figma, from https://www.figma.com/

5

DEMONSTRATION

Thank You & QA!