Programming Application Frameworks IT3030

ElectroGrid System

Group Members

IT20020958 Mayadunne J.N.A.

IT20062392 Withana W.D.T.S.J.

IT20003128 Jayathilaka G.W.C.D.

IT20111724 Gunasekara H.M.D.P.K.

Group Number

Y3S1.WE.IT.1.02. G1.105

Table of Contents

Contents

1. Introduction

ElectroGrid (EG) is the company in charge of the country's power system. They have a certain system which enables to monitor the customers' power consumption, generating monthly invoices and sending them to the users automatically, and accepting online payments from the users. Here, the functions of this system are executed by the RESTful architecture based on Java technologies such as JAX-RS and Jersey. Apparently, this support system is designed with the aim of increasing the scalability of EG's online platform in order to serve all the users.

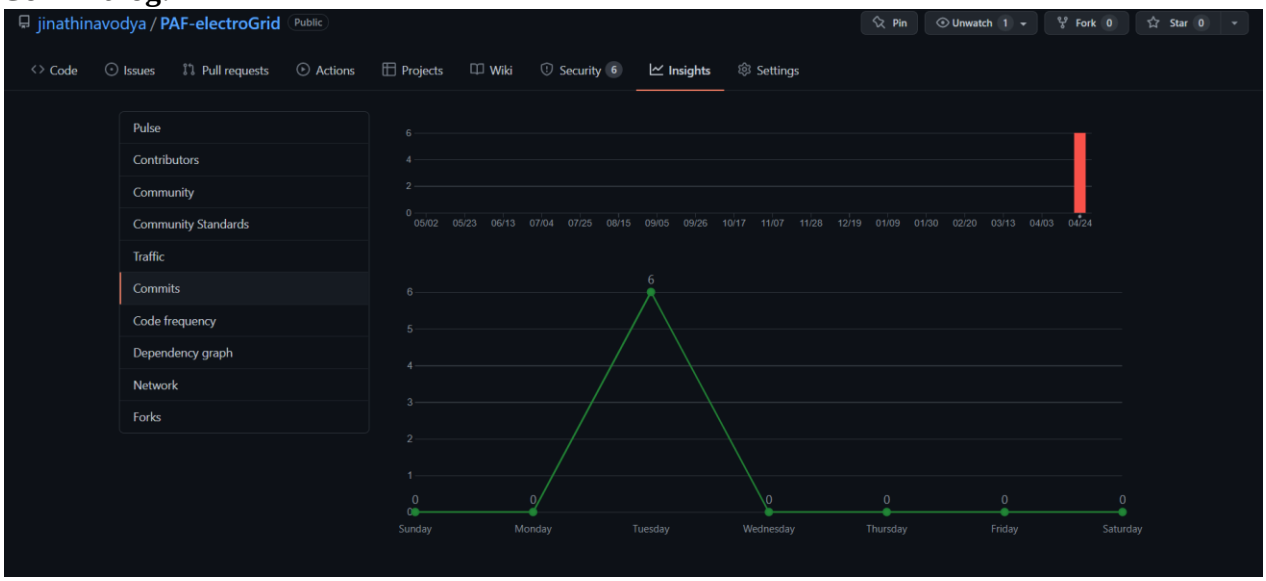
2. Workload Distribution

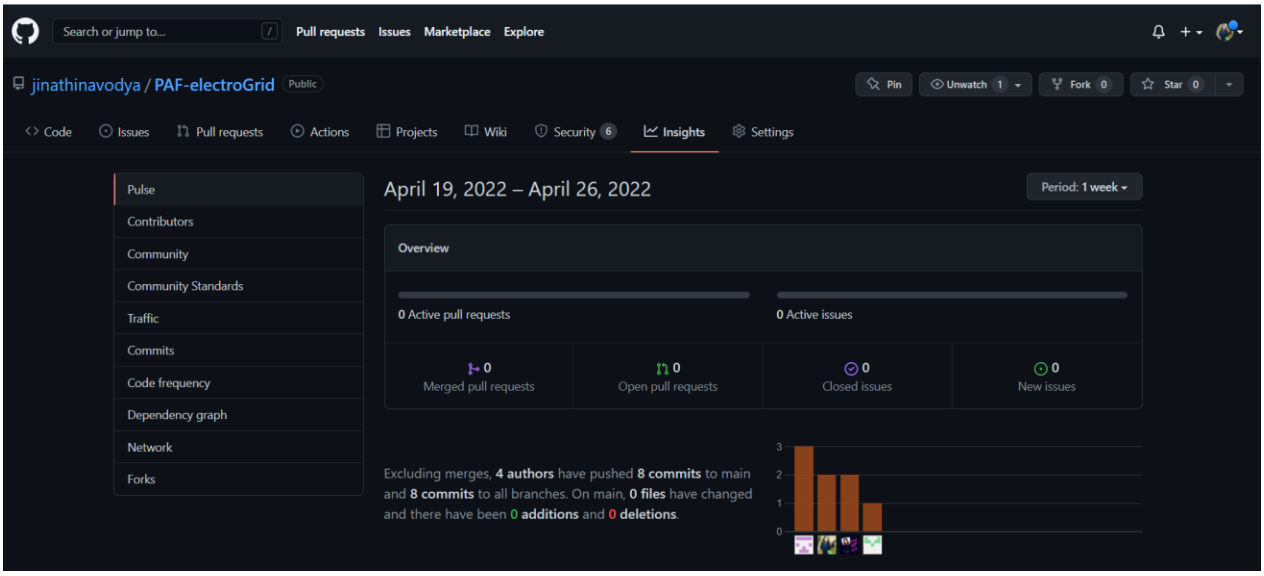
Student ID	Name	Service	Work allocated
IT20020958	Mayadunne J.N.A.	Auditing Service (Accounts)	Implementing all the services related for handling Accounts
IT20062392	Withana W.D.T.S.J.	Invoicing Service (Billing)	Implementing all the services related to billing procedure
IT20003128	Jayathilaka G.W.C.D.	Payment Service	Implementing all services related with paying process
IT20111724	Gunasekara H.M.D.P.K.	End-user Service	Implementing all the services related for user handling

3. Version control system

VCS Repo: https://github.com/jinathinavodya/PAF-electroGrid

Commit log:


jinathinavodya / PAF-electroGrid
Public
Pin
Unwatch 1
Fork 0
Star 0
Code
Issues
Pull requests
Actions
Projects
Wiki
Security 6
Insights
Settings
Pulse
Contributors
Community
Community Standards
Traffic
Commits
Code frequency
Dependency graph
Network
Forks
05/02 05/23 06/13 07/04 07/25 08/15 09/05 09/26 10/17 11/07 11/28 12/19 01/09 01/30 02/20 03/13 04/03 04/24
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday


Search or jump to...
Pull requests
Issues
Marketplace
Explore
jinathinavodya / PAF-electroGrid
Public
Pin
Unwatch 1
Fork 0
Star 0
Code
Issues
Pull requests
Actions
Projects
Wiki
Security 6
Insights
Settings
Pulse
Contributors
Community
Community Standards
Traffic
Commits
Code frequency
Dependency graph
Network
Forks
April 19, 2022 – April 26, 2022
Period: 1 week
Overview
0 Active pull requests
0 Active issues
0 Merged pull requests
0 Open pull requests
0 Closed issues
0 New issues
Excluding merges, 4 authors have pushed 8 commits to main and 8 commits to all branches. On main, 0 files have changed and there have been 0 additions and 0 deletions.

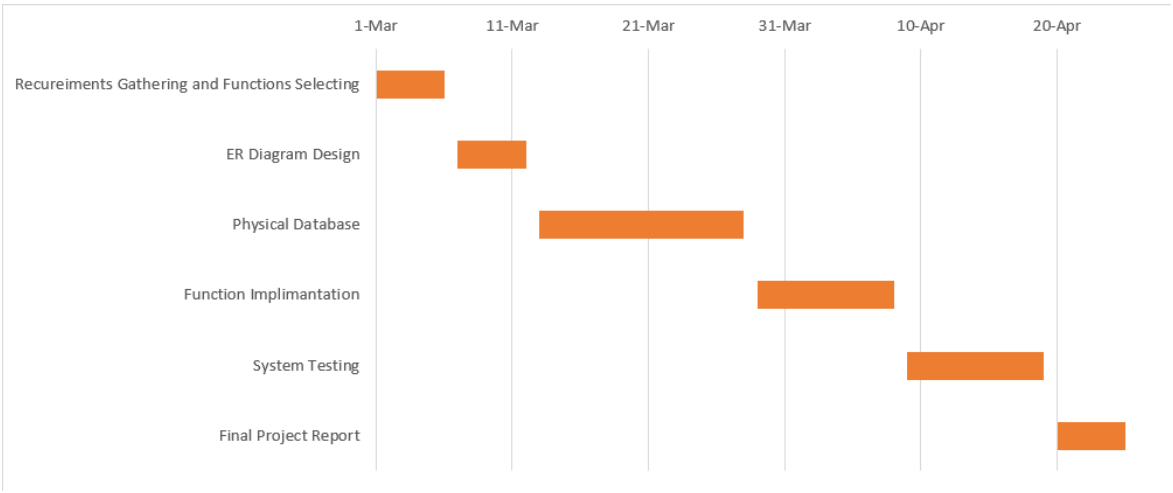
4. SE Methodology

Because there is no uncertainty in the behavior of the requirements, ElectroGrid's system architecture and implementation are closely tied to the waterfall development approach.

We collected stakeholder and customer requirements as the first phase in the waterfall technique. As a consequence of this process, we were able to have a better grasp of the project scope, and each team member was able to comprehend their role in the project and what it included. JAX-RS and Jersey, as well as Java, were used to create the system architecture. Based on the knowledge from the previous stage, each member of the group began developing the Electro grid system, and a superior product resulted. The implementation procedure is being followed.

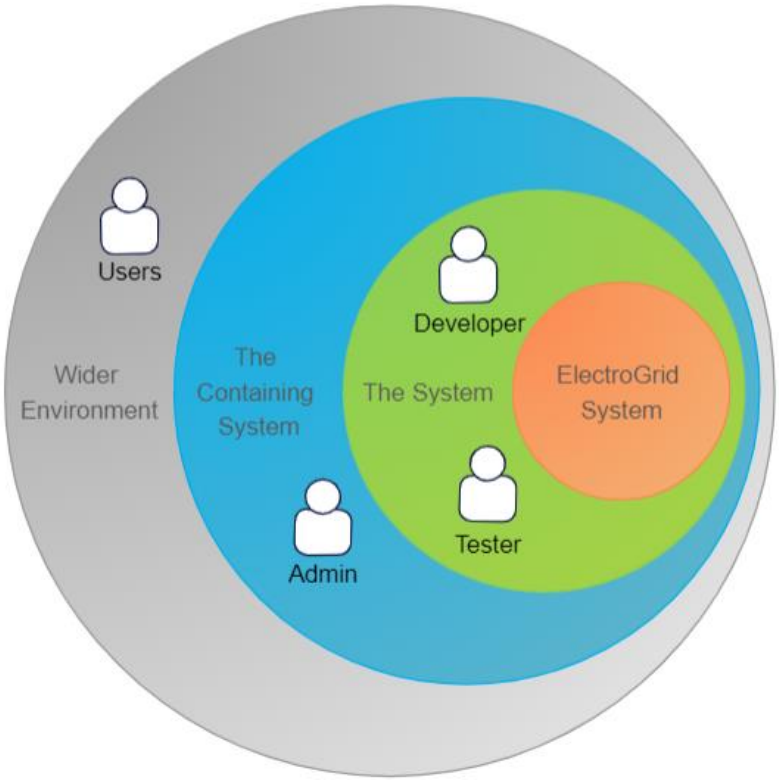
We were able to simplify the difficult process of project development using the waterfall paradigm. We were able to construct the Electro Grid system without overlapping each phase since we used the waterfall methodology to design it.

5. Time Schedule (Gantt chart)



6. Requirements

6.1 Stakeholder analysis (onion diagram)



6.2 Requirements analysis

Individual Functionality	Functional Requirements	Non-Functional Requirements	Technical Requirements

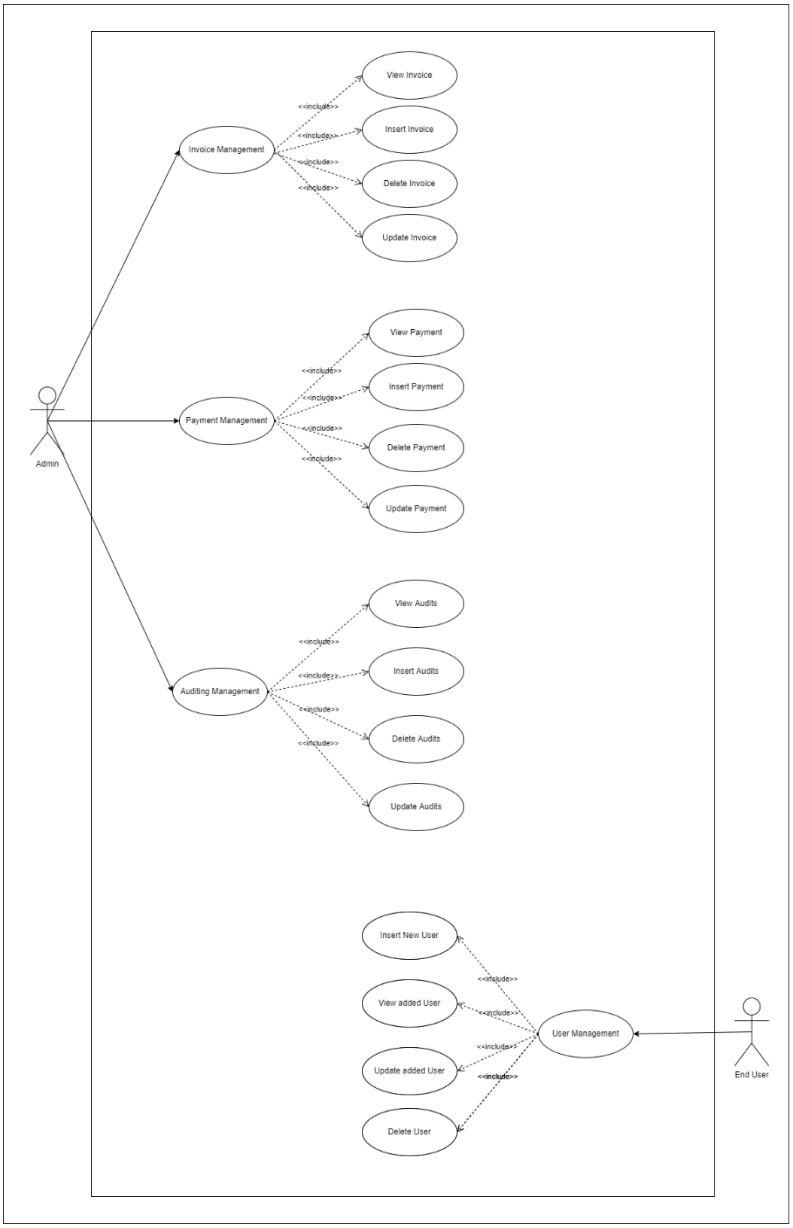
Auditing Service Management (Accounting)	• Insert Auditing details • Can get all Auditing Details • Can get one Auditing Details • Update Auditing details • Delete Auditing Details	• Quality • Reliability • Efficiency • Availability	keeping records of financial affairs of the system and make changers when they required to do insert update, delete, changes As well as can view their details.

Invoice Service Management (Billing)	• Insert Invoice details • Can get all Invoice Details • Can get one Invoice Details • Update Invoice details • Delete Invoice Details	• Quality • Reliability • Efficiency • Availability	Creating the list of services provided by the system invoicing and make changers when they required to do insert update, delete, changes As well as can view their details.
Payment Service Management	• Insert Payment details • Can get all Payment Details • Can get one Payment Details	• Quality • Reliability • Efficiency • Availability	Creating records of Payment details and make changers when they required to do insert update, delete, changes As well as can view their details.

	• Update Payment details • Delete Payment Details		
End User Service Management	• Insert User details • Can get all User Details • Can get one User Details • Update User details • Delete User Details	• Quality • Reliability • Efficiency • Availability	Keeping records of the user's information and make changers when they required to do insert update, delete, changes As well as can view their details.

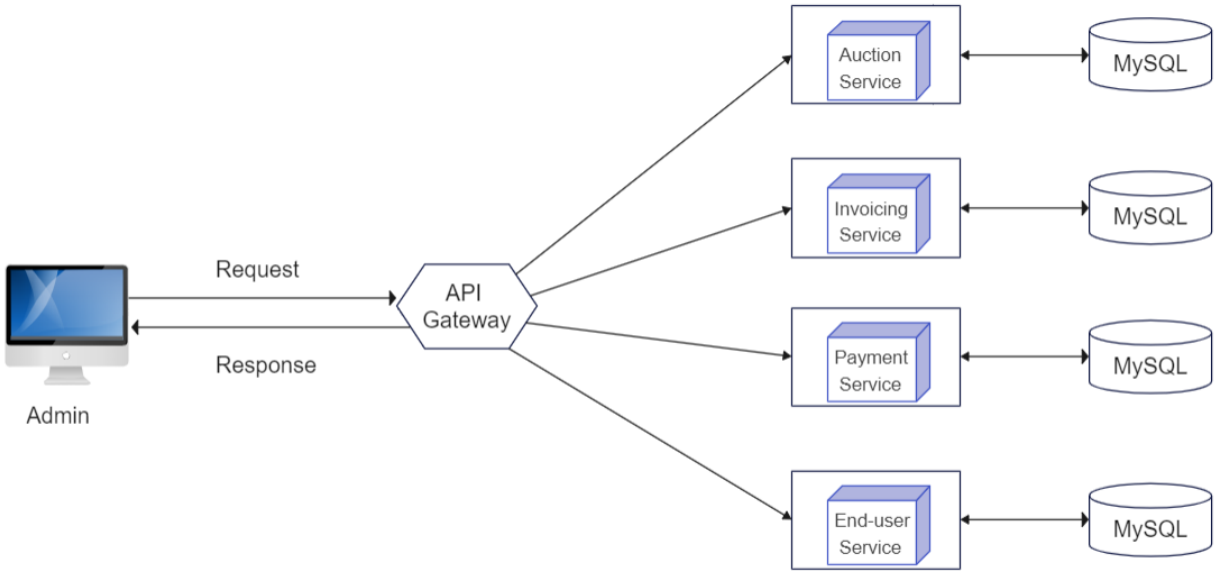
6.3 Requirements modelling

Use Case Diagram

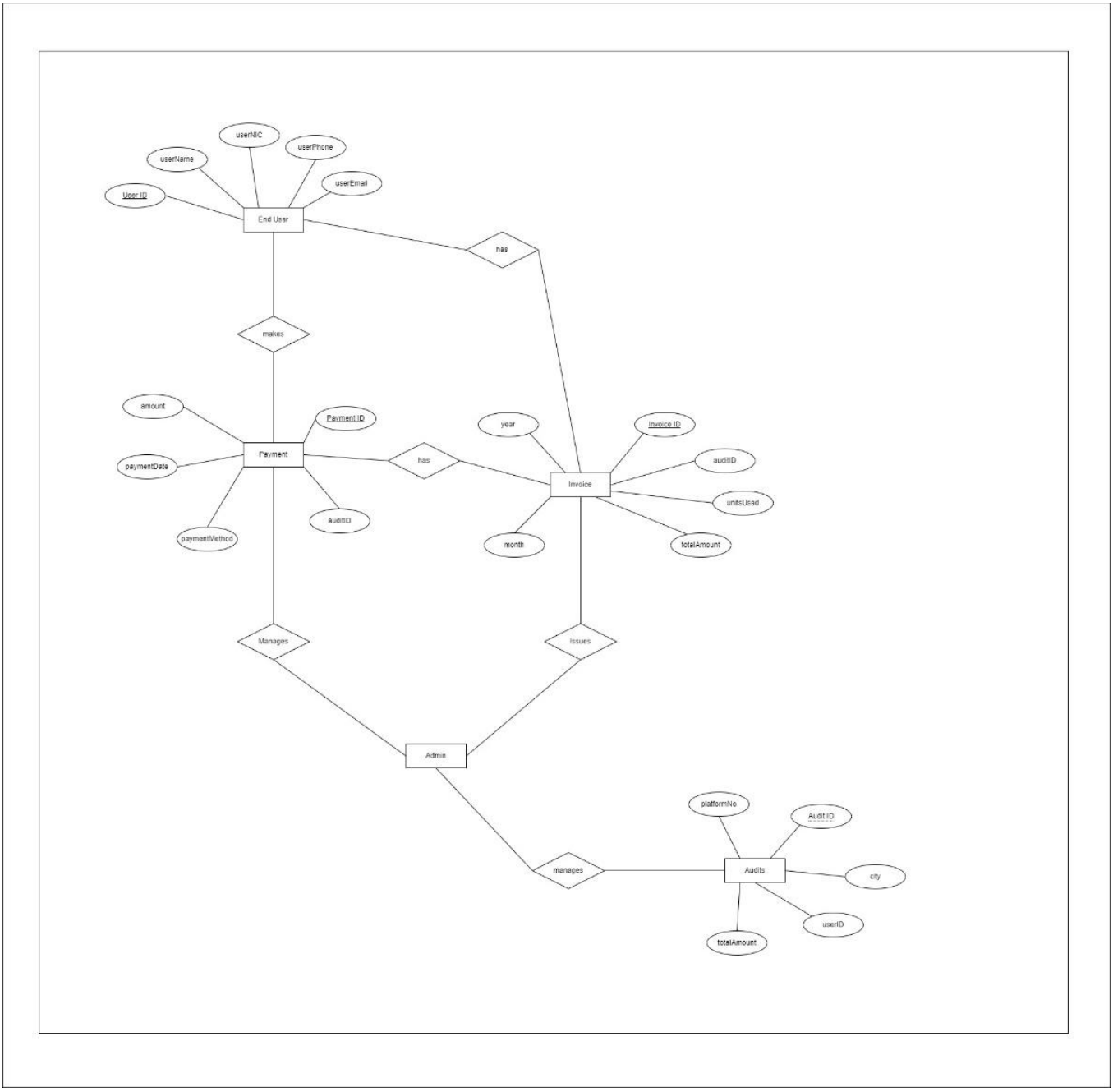


7. System overall design

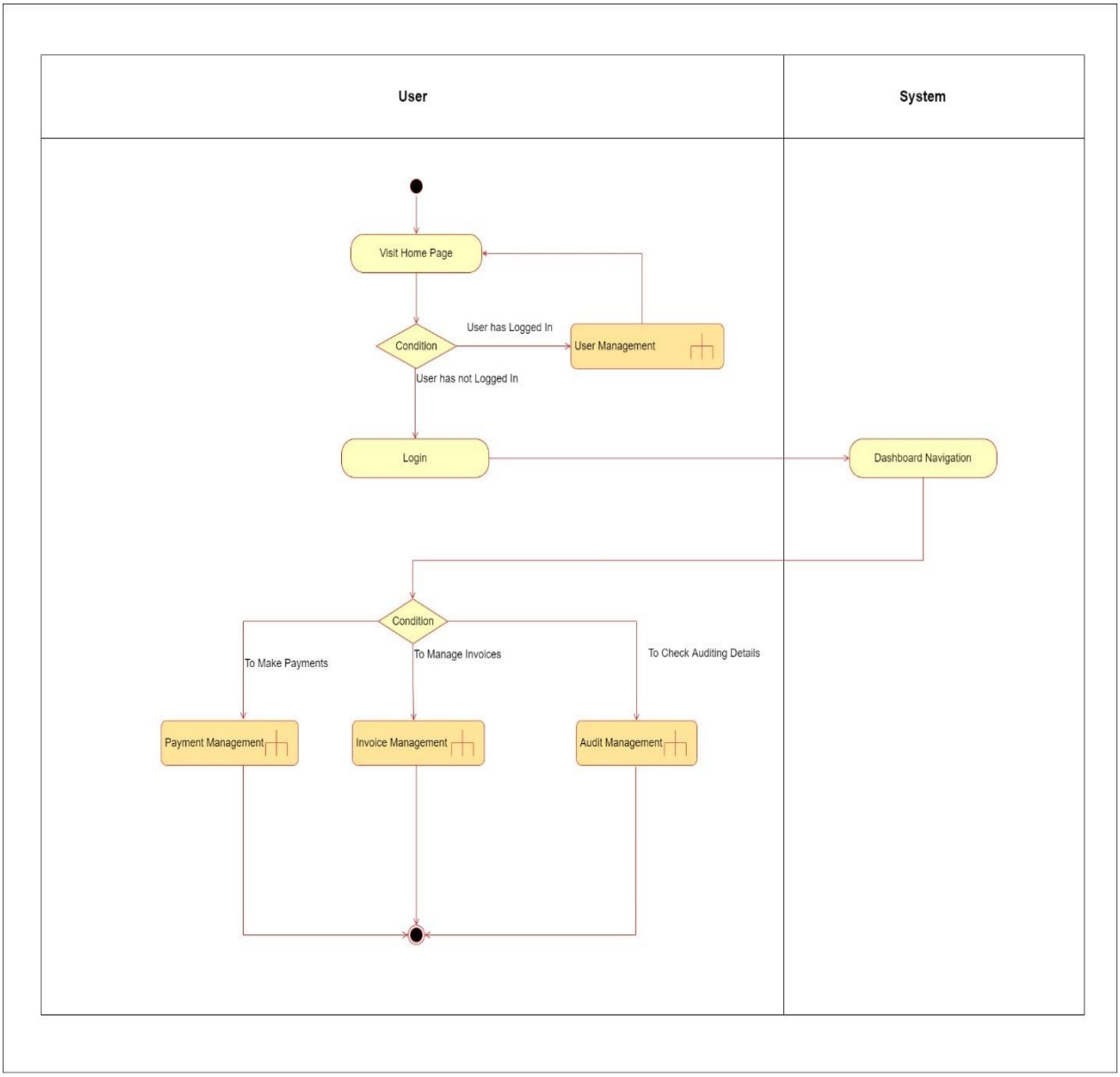
7.1 Overall architecture



7.2 ER Diagram



7.3 Activity Diagram



8. Individual Section

Note: Testing methodology and results, Testing Results Screenshots(postman) are attached in Appendix.

8.1 Auditing Management (Accounting)

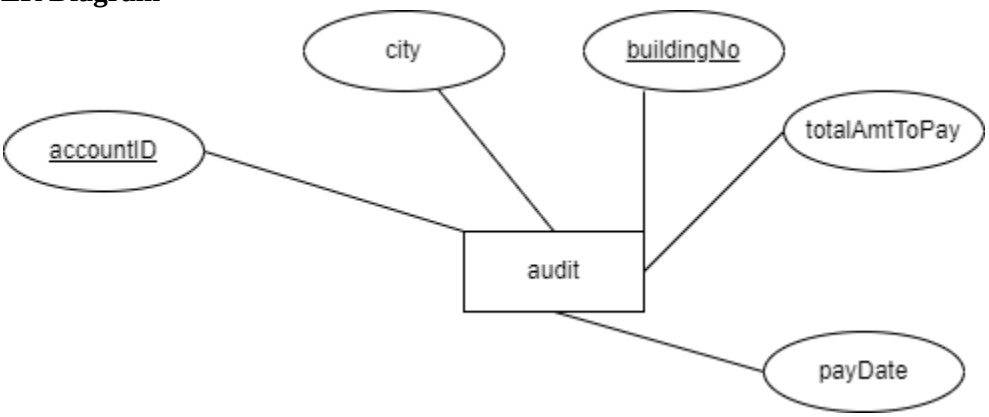
IT20020958 Mayadunne J.N.A.

GitHub – Auditing Service

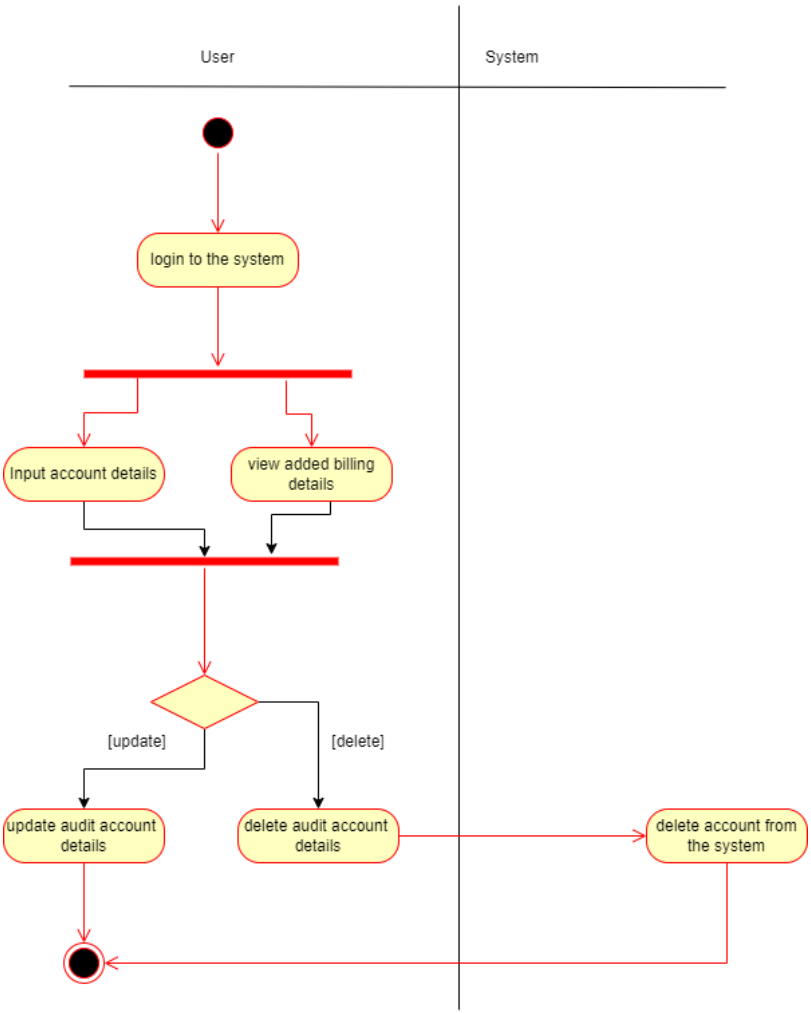
API Design Relational

Auditing management service is used to and new accounts in to the ElectroGrid (EG) System. In addition to that it facilitates update and delete the details of added accounts through the system. admin can view their profile and they can change any detail when they need. User has all the authentication for add, update, delete and view user details.

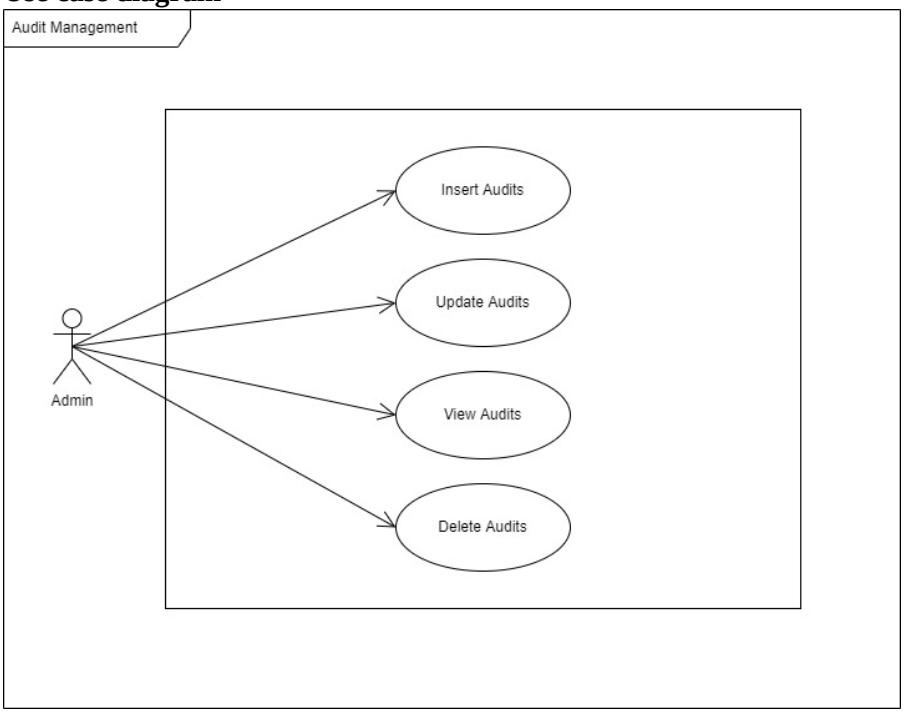
ER Diagram



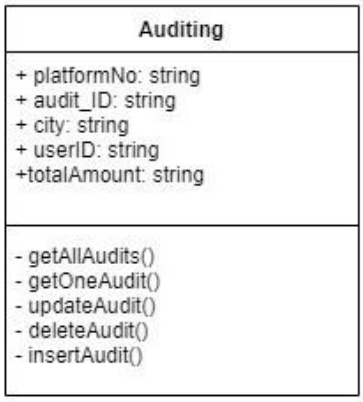
Activity Diagram



Use case diagram



Class Diagram



Tools Used

	Tool	**Reason**
Back End	JAX-RS, Jersey, Java	Easy configuration and popularity in the industry.
Server	Apache Tomcat 9.0	Comparatively highly secured
Database	MySQL+ workbench	Easily able to handle the Data Base
Build Tool	Maven	Simplifies the process of project building
IDE	Eclipse IDE (2020-12)	Great for handling larger projects and highly stabilized
Test Client	Postman	Easy creation of test suits

8.2 Invoice Management (billing)

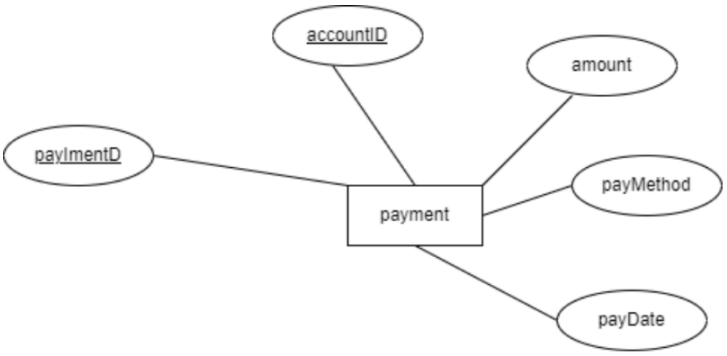
IT20062392 - Withana W.D.T.S.J.

GitHub – Invoice Service

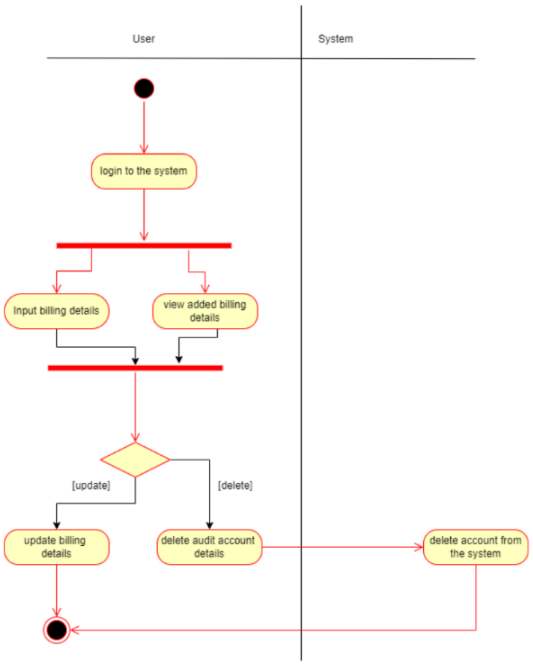
API Design Relational

Invoice management service is used to and new accounts in to the ElectroGrid (EG) System. In addition to that it facilitates update and delete the details of added invoices through the system. admin can view their profile and they can change any detail when they need.

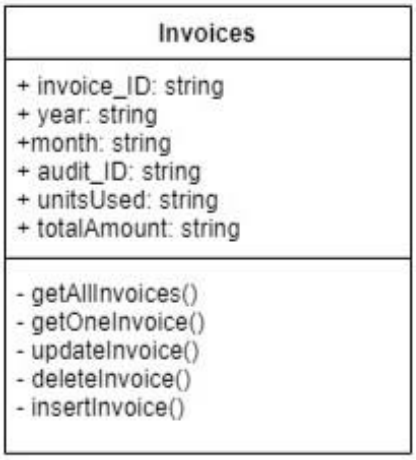
ER Diagram



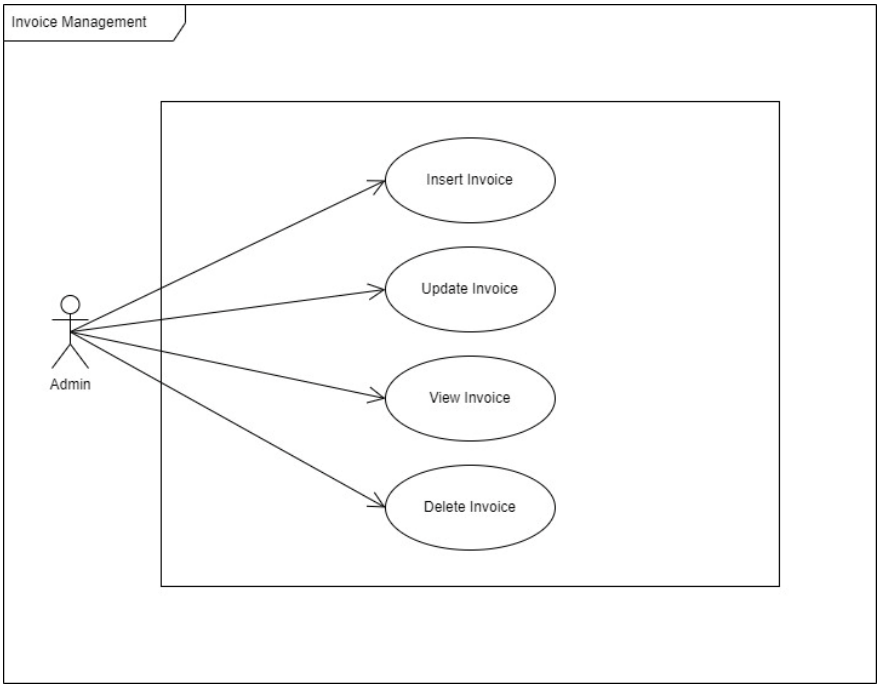
Activity Diagram



Class Diagram



Use case diagram



Tools Used

	Tool	**Reason**
Back End	JAX-RS, Jersey, Java	Easy configuration and popularity in the industry.
Server	Apache Tomcat 9.0	Comparatively highly secured
Database	MySQL+ workbench	Easily able to handle the Data Base
Build Tool	Maven	Simplifies the process of project building

IDE	Eclipse IDE	Great for handling larger projects and highly stabilized
Test Client	Postman	Easy creation of test suits

8.3 Payment Management

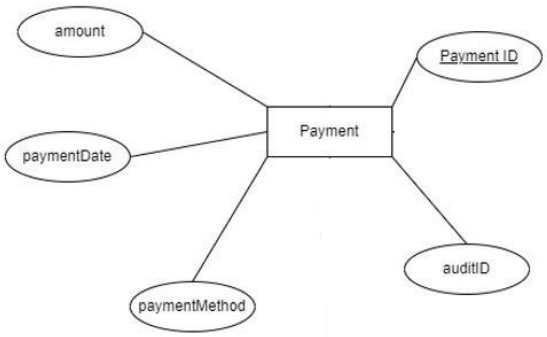
IT20003128 - Jayathilaka G.W.C.D.

GitHub – payment Service

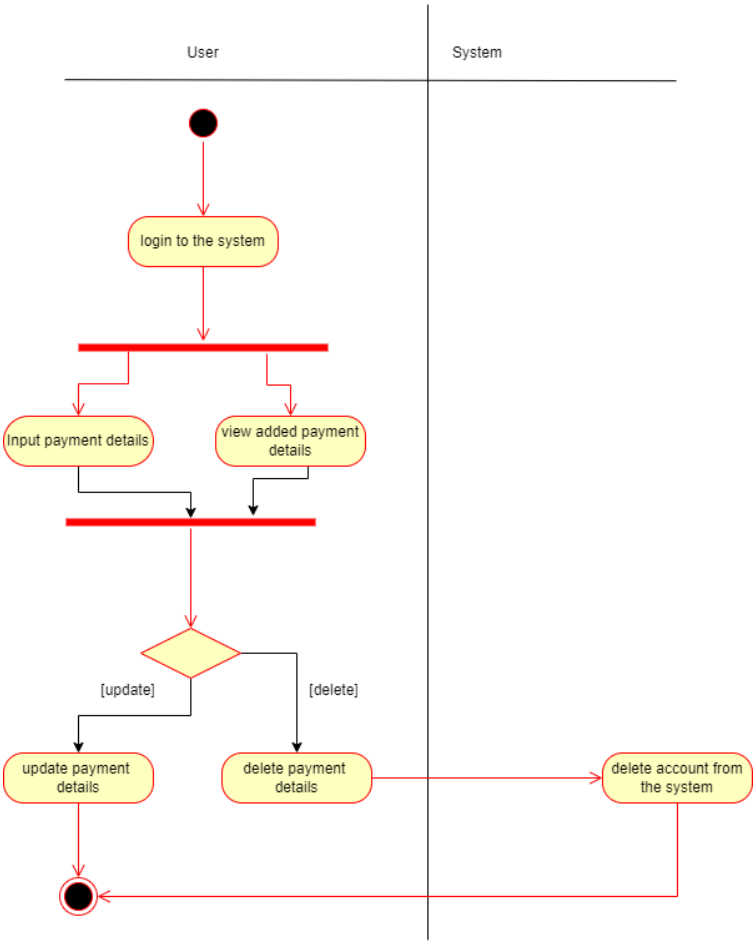
API Design Relational

Payment management service is used to and new accounts in to the ElectroGrid (EG) System. In addition to that it facilitates update and delete added payment details through the system. admin can view their profile and they can change any detail when they need.

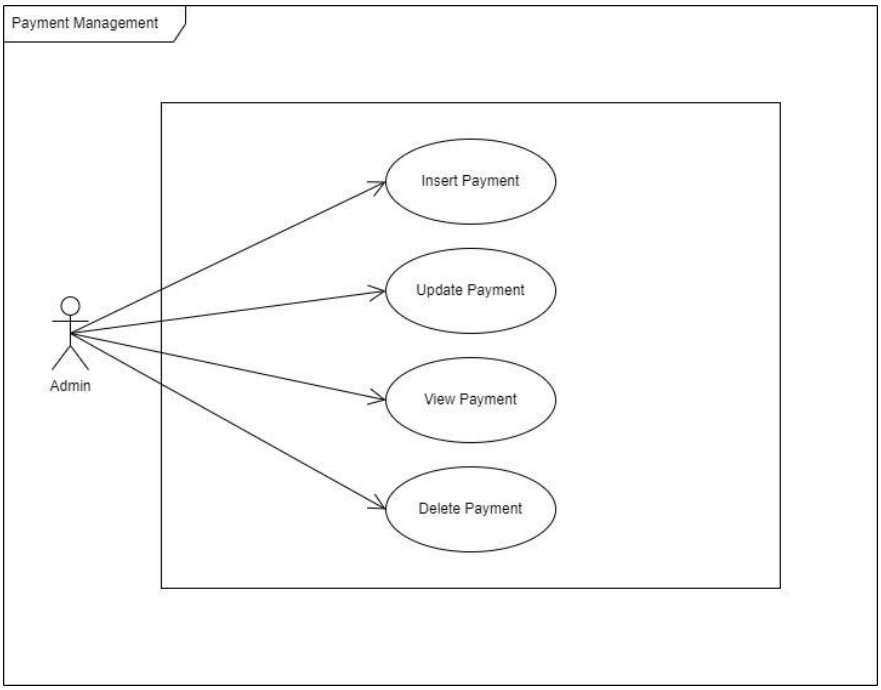
ER Diagram



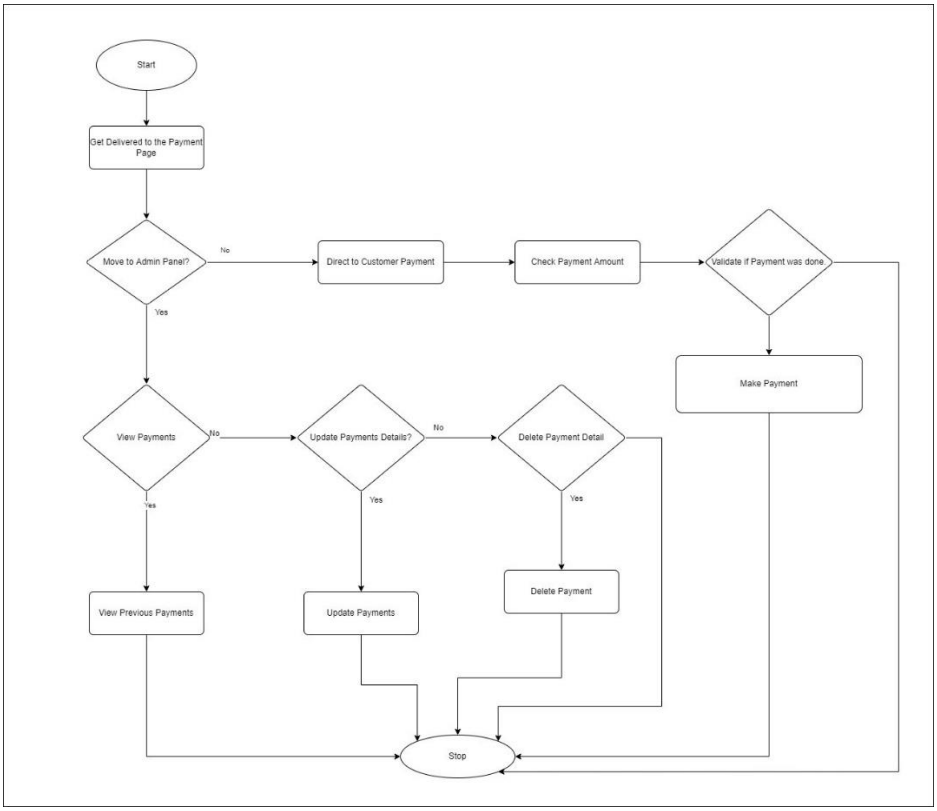
Activity Diagram



Use case Diagram



Flow chart



Tools Used

	Tool	**Reason**
Back End	JAX-RS, Jersey, Java	Easy configuration and popularity in the industry.
Server	Apache Tomcat 10.0	Comparatively highly secured
Database	MySQL	Easily able to handle the Data Base
Build Tool	Maven	Simplifies the process of project building
IDE	Eclipse IDE for Enterprise Java and Web Developers- 2020-12	Great for handling larger projects and highly stabilized
Test Client	Postman	Easy creation of test suits

8.4 User Management

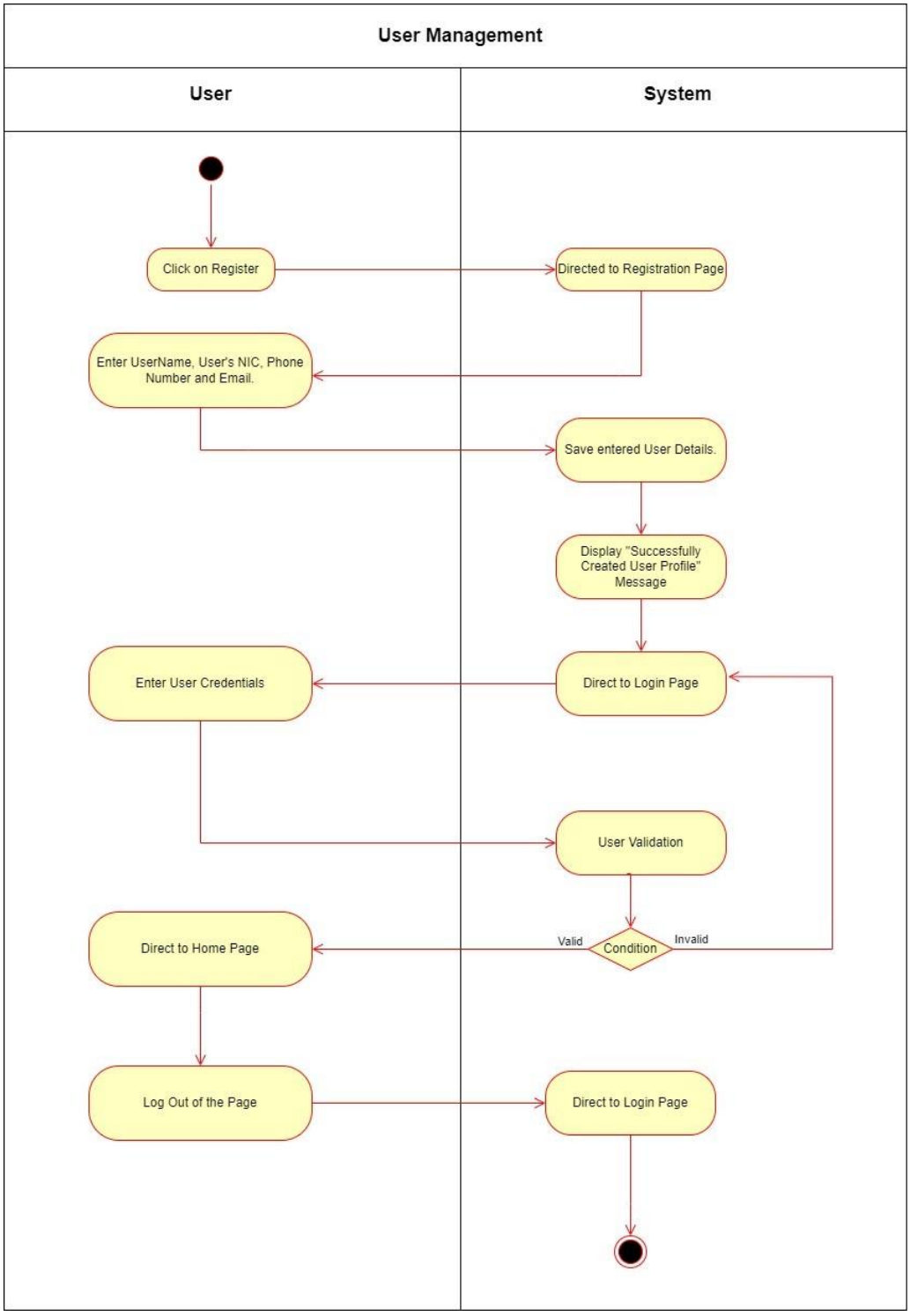
IT20111724 - Gunasekara H.M.D.P.K.

GitHub – User Service

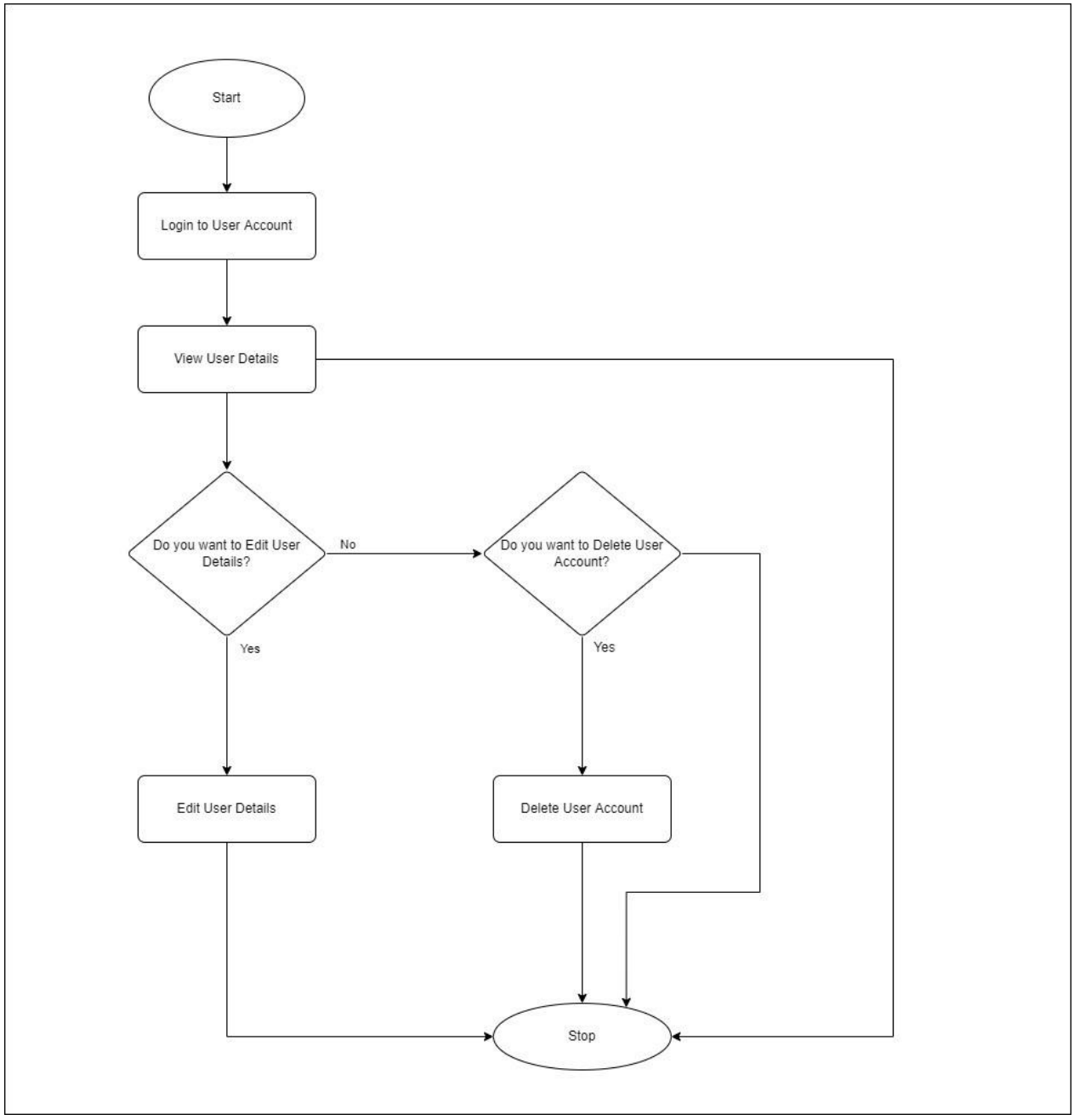
API Design Relational

In User Management Function If users want to update profile details they can update easily and if they want to delete accounts, they can also delete accounts from the system.an add user accounts to the system. User Management function is implemented as an API system that helps in the adding, updating, viewing and deleting the details regarding the users' details.

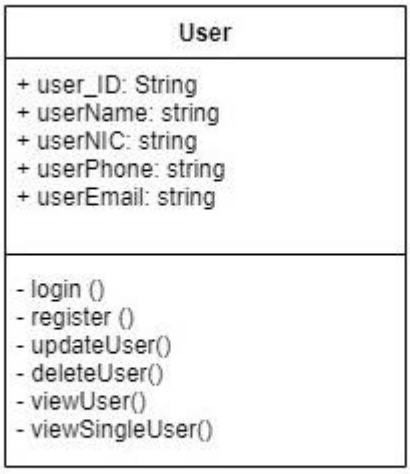
Activity Diagram



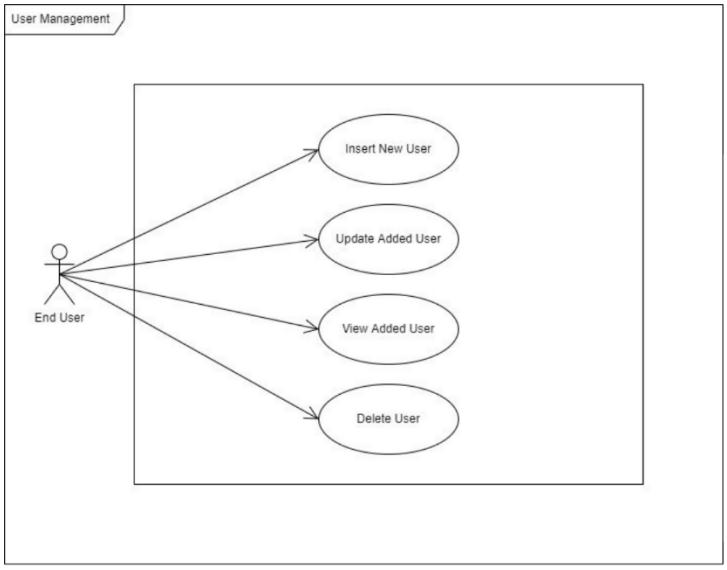
Flow Chart


Start
Login to User Account
View User Details
Do you want to Edit User Details?
No
Do you want to Delete User Account?
Yes
Yes
Edit User Details
Delete User Account
Stop

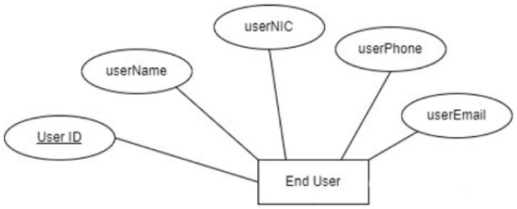
Class Diagram



Use Case Diagram



ER Diagram



Tools Used

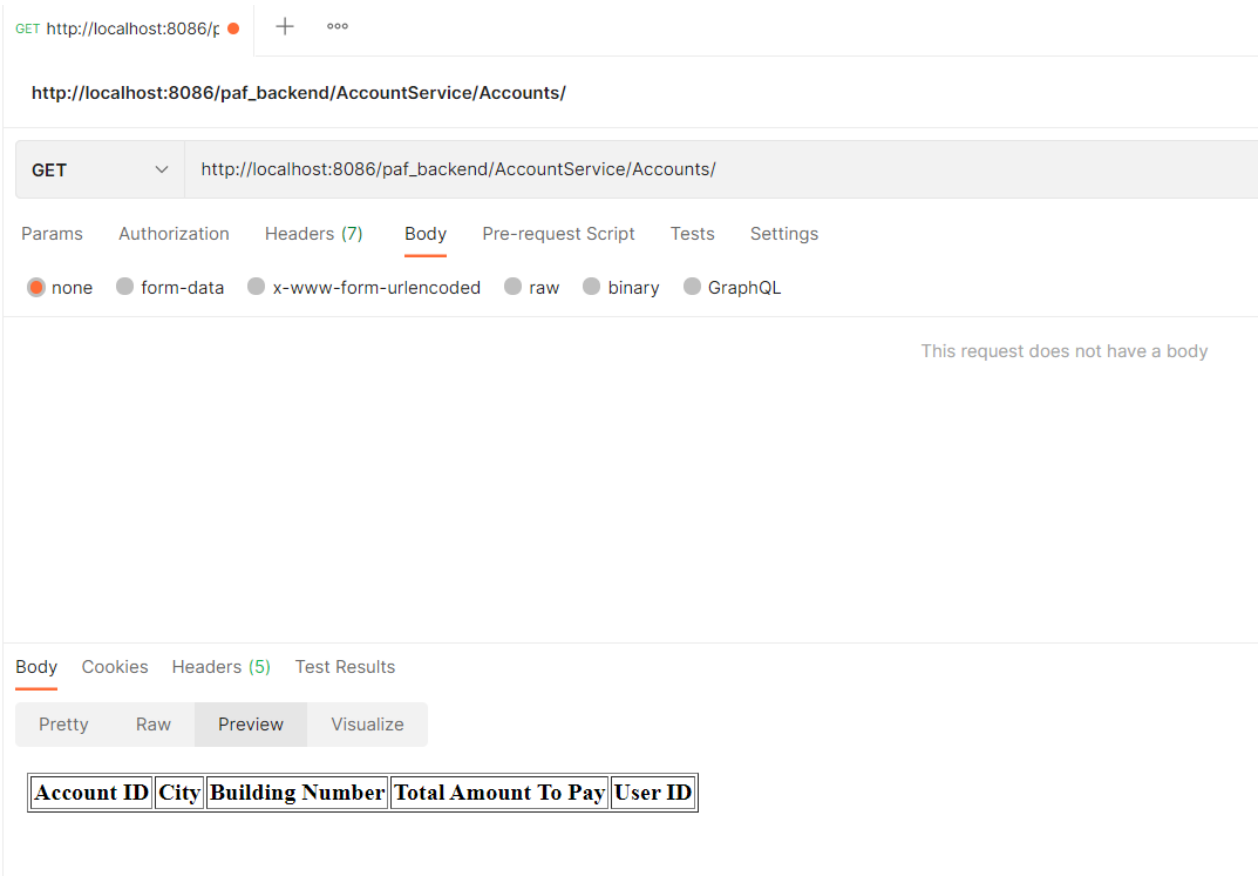
	Tool	**Reason**
Back End	JAX-RS, Jersey, Java	Easy configuration and popularity in the industry.
Server	Apache Tomcat 10.0	Comparatively highly secured
Database	MySQL+ phpMyAdmin	Easily able to handle the Data Base
Build Tool	Maven	Simplifies the process of project building
IDE	Eclipse IDE for Enterprise Java and Web Developers- 2020-12	Great for handling larger projects and highly stabilized
Test Client	Postman	Easy creation of test suits

9. Appendix

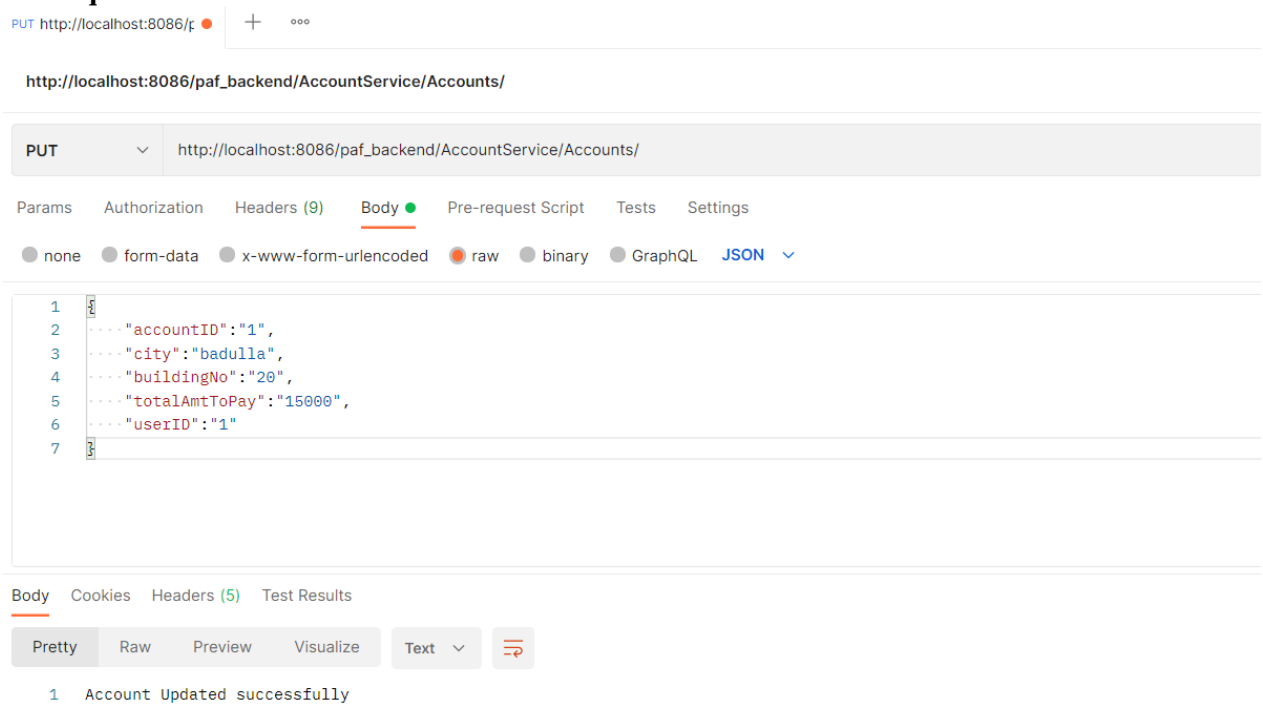
10.1 IT20020958 Mayadunne J.N.A.

10.1.2Test Results Screenshots -Postman

Get Request



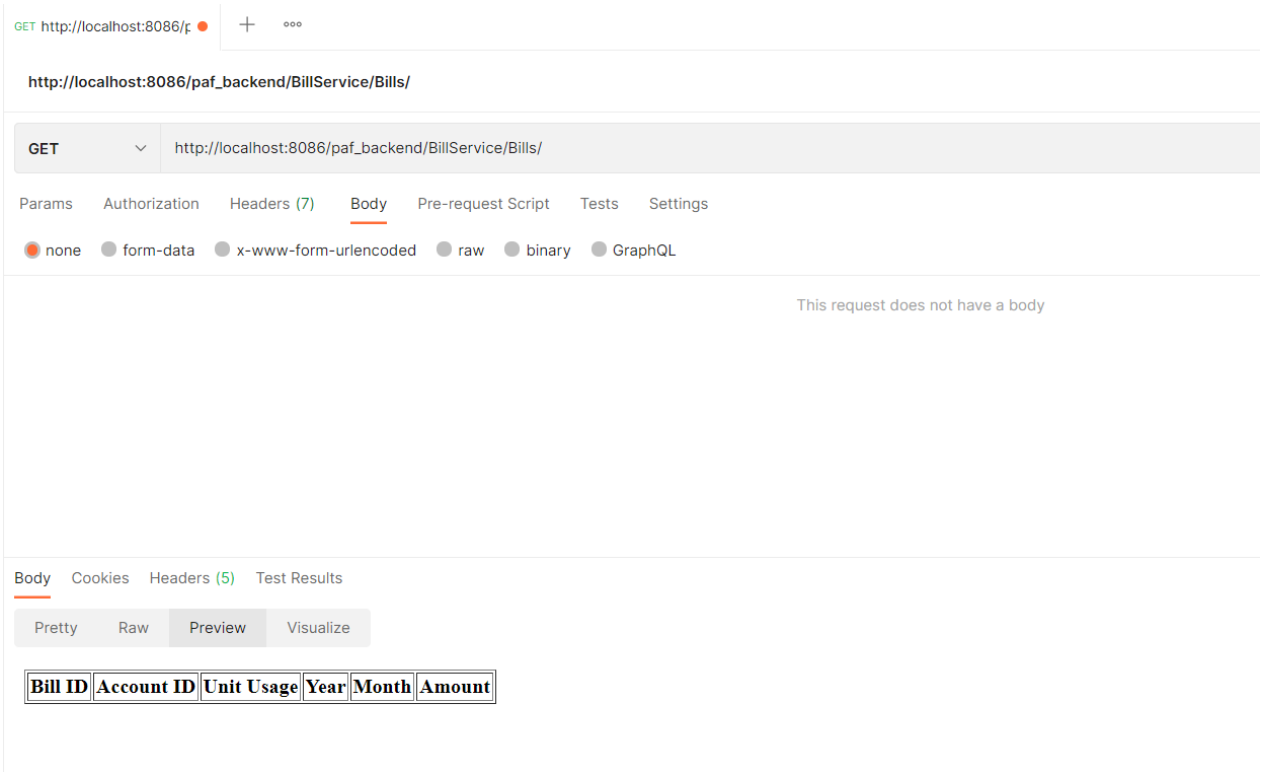
Put request



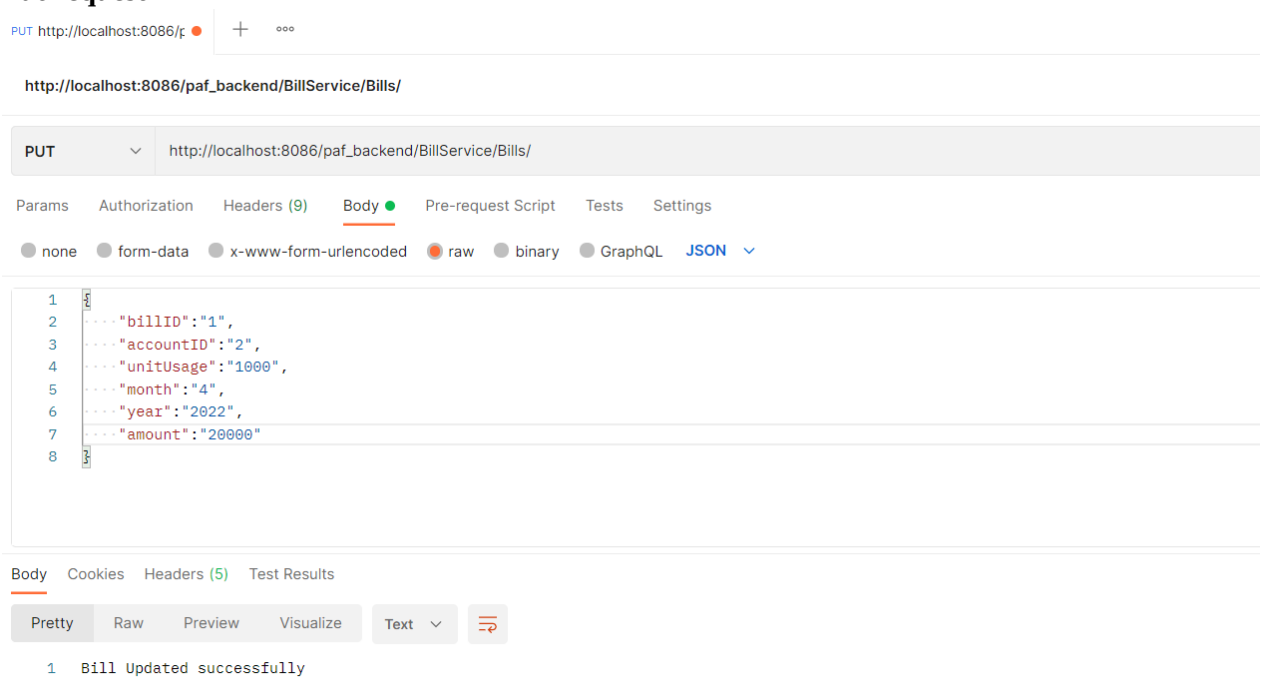
10.2 IT20062392 Withana W.D.T.S.J.

10.2.2Test Results Screenshots –Postman

Get Request


GET http://localhost:8086/
http://localhost:8086/paf_backend/BillService/Bills/
GET
http://localhost:8086/paf_backend/BillService/Bills/
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
This request does not have a body
Body
Cookies
Headers (5)
Test Results
Pretty
Raw
Preview
Visualize
Bill ID
Account ID
Unit Usage
Year
Month
Amount

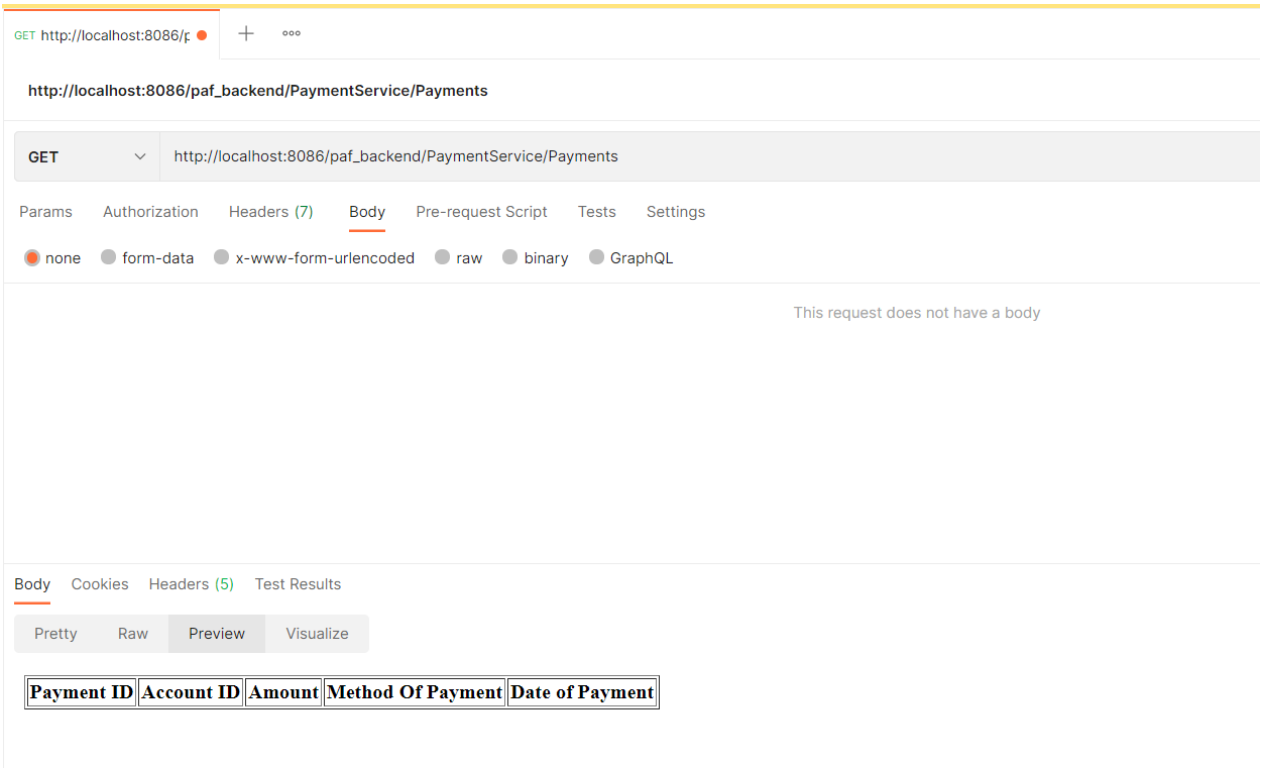
Put request


PUT http://localhost:8086/
http://localhost:8086/paf_backend/BillService/Bills/
PUT
http://localhost:8086/paf_backend/BillService/Bills/
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"billID":"1",
"accountID":"2",
"unitUsage":"1000",
"month":"4",
"year":"2022",
"amount":"20000"
}
Body
Cookies
Headers (5)
Test Results
Pretty
Raw
Preview
Visualize
Text
Bill Updated successfully

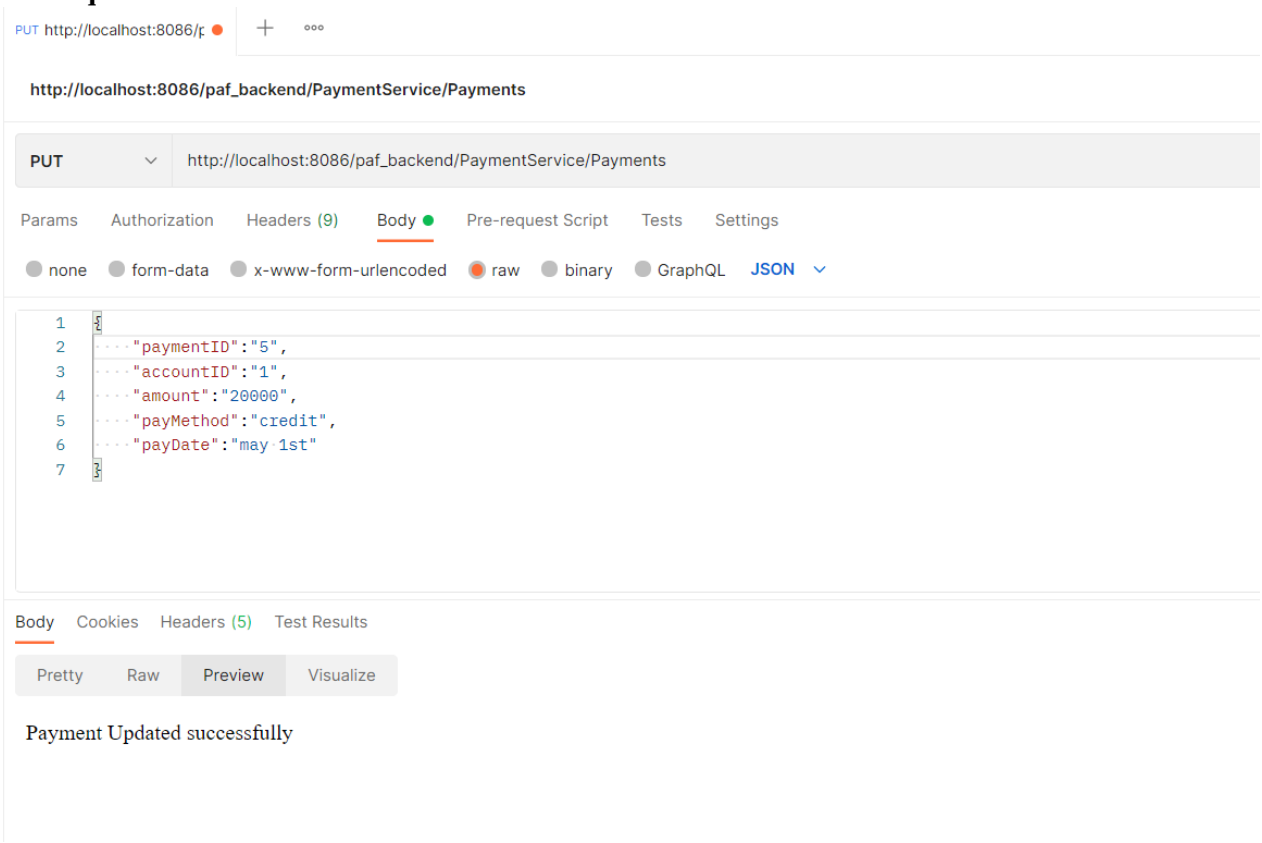
10.3 IT20003128 Jayathilaka G.W.C.D.

10.3.2Test Results Screenshots -Postman

Get Request



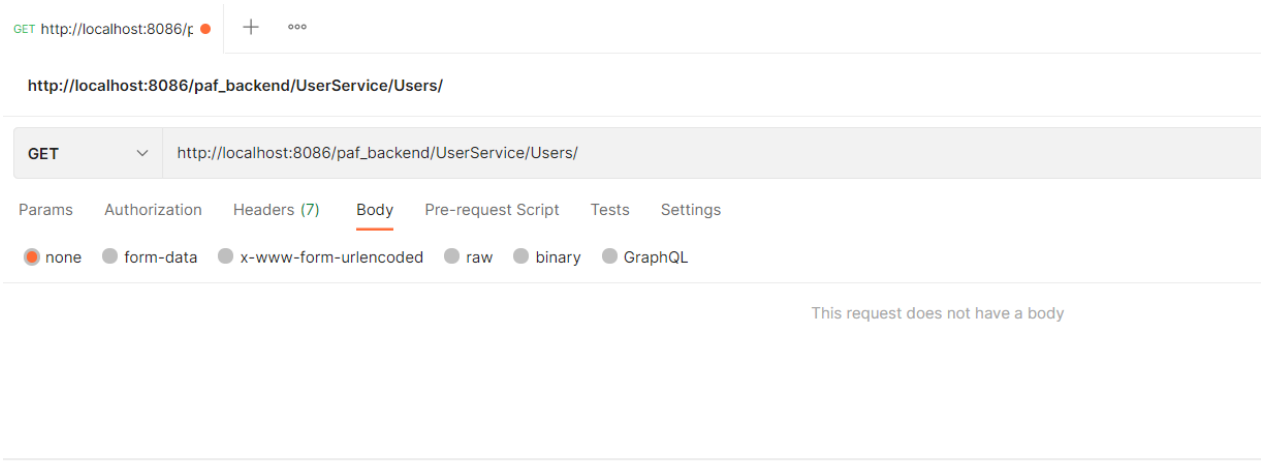
Put request

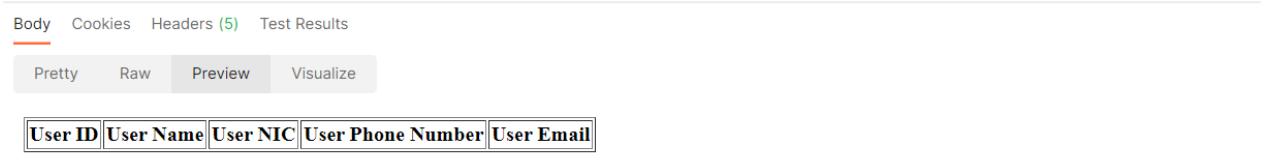


10.4 IT20111724 Gunasekara H.M.D.P.K.

10.4.2Test Results Screenshots -Postman

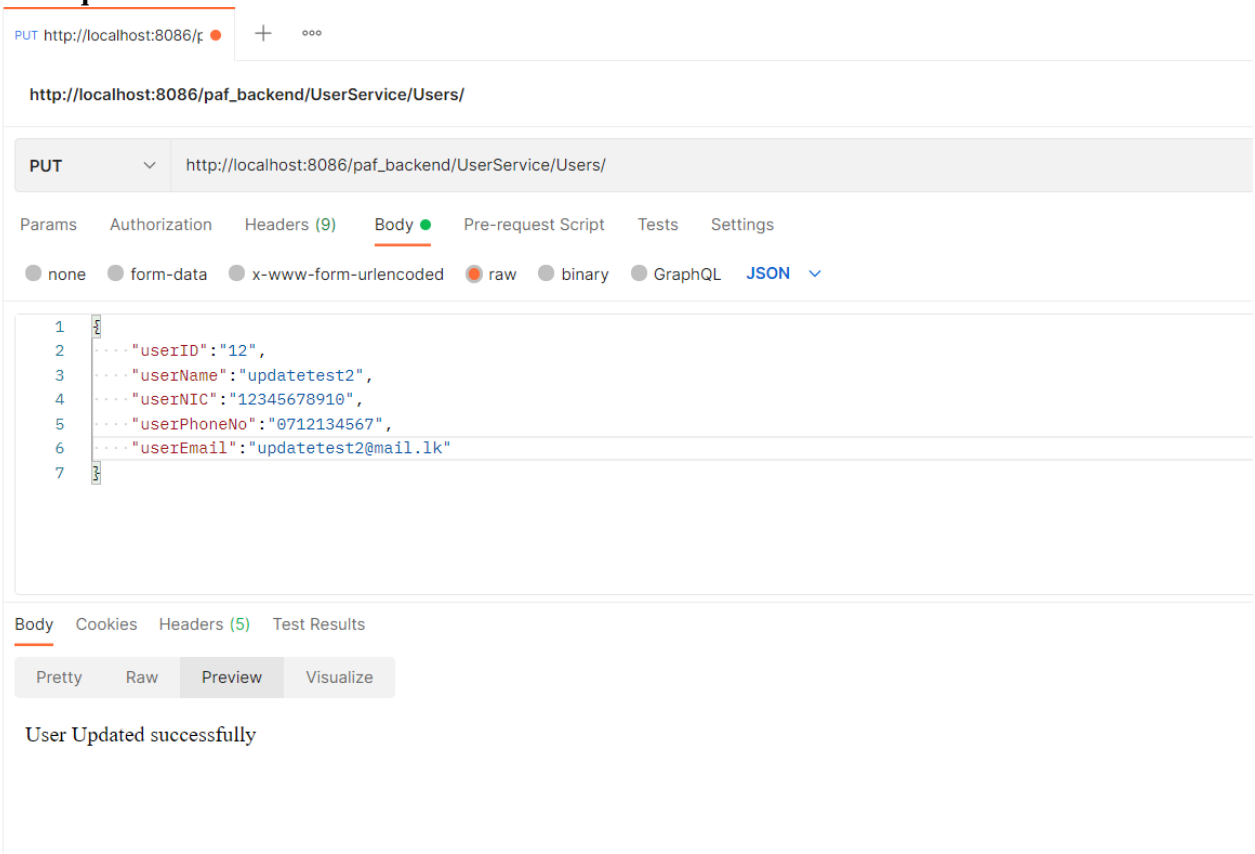
Get Request





User ID	User Name	User NIC	User Phone Number	User Email

Put request



9. References

1.Youtube

2.Stack Overflow

3.Eclipse [Online] https://www.tutorialspoint.com/eclipse/index.htm

4.Postman [Online] https://www.postman.com/