EVENT PLATFORM

A PROJECT REPORT

Submitted by

AMAL S

REG NO: TKM20MCA2005

to

The APJ Abdul Kalam Technological University

In partial fulfillment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Thangal Kunju Musaliar College of Engineering
Kerala

DEPARTMENT OF COMPUTER APPLICATIONS

JULY 2022

DECLARATION

I undersigned hereby declare that the project report on **Event Platform**, submitted for partial fulfillment of the requirements for the award of the degree of M.C.A of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Vaheetha Salam. This submission represents my ideas in my own words and where ideas or words of others have been included; I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam AMAL S

18/07/2022

DEPARTMENT OF COMPUTER APPLICATION

THANGAL KUNJU MUSALIAR COLLEGE OF ENGINEERING

C E R T I F I C A T E

This is to certify that, the report entitled "**Event Platform**" submitted by **AMAL S (TKM20MCA2005),** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications, is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose

Internal Supervisor Head of the Department External Examiner

Tekact
www.tekact.com

Tekact Private Limited,
Balaraj's Arcade, 4th Floor,
Whitefield Main Road,
Opp Brigade Cosmopolis.
Bengaluru – 560066

TO WHOM IT MAY CONCERN

This is to certify that Mr. Amal S from TKM College of Engineering, Kollam is currently undergoing the internship from **April 6th, 2022** to **August 3rd, 2022** with **Tekact Pvt Ltd**. During the tenure with us he was punctual, hardworking, and inquisitive.

We wish him all the success in future endeavors.

Sincerely,

Place: Bengaluru

Date: 15-July-2022

SVK Pillai

Director

ACKNOWLEDGEMENT

First and foremost, I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely grateful to **Prof. Dr. Fousia M Shamsudeen**, Head of the Department, Department of Computer Applications, for providing me with best facilities.
I would like to thank our coordinator and project guide **Prof. Vaheetha Salam**, Department of Computer Applications, who motivated me throughout the project.

I would like to thank my external coordinator **Mr. Rajesh S R**, Tekact Private Limited, who guided me throughout my work.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped
me in the successful completion of this project.

AMAL S

ABSTRACT

In an event-driven architecture, when a service performs a task that may be of interest to other services, that service generates an event, a record of the action performed. Other services use these events so that they can perform their own necessary tasks as a result of the event. Unlike REST, the services that make the request do not need to know the details of the services that use the requests. Here's a simple example: When an order is placed on an e-commerce site, an "order placed" event is generated and then consumed by some small service. Events can be published in different ways. For example, they can be published to a queue that ensures delivery of the event to the appropriate consumers, or they can be published to the "pub/sub" template stream that publishes the event and allows access to all interested parties. In both cases, the producer announces the event and the consumer receives the event, reacting accordingly. Note that in some cases these two actors may also be referred to as publishers and subscribers. This project proposes a microservice architecture multi-tenant product for publishing targeted events. And this is to provide APIs for users to compose applications that can work with different domains.

Contents

1. INTRODUCTION 1

1.1 Objective …….....2

1.2 Company Profile ……...…...2

2. LITERATURE SURVEY 3

2.1. Purpose Of The Literature Review ……....5

2.2. Related Works ……...6

2.2.1 Microservice……...6

2.2.2 Spring Boot……6

2.2.3 OCR……....7

2.2.4 Tesseract……7

2.2.5 Postresql……8

2.2.6 Restful Api……9

2.2.7 Thymeleaf……..9

2.2.8 Swagger……10

2.2.9 Maven……10

2.2.10 Postman……11

3. METHODOLOGY 12

3.1 Architectural Design ……13

3.2 Module Description …….....14

3.2.1 Admin ……..15

3.2.1.a Organization …….....15

3.2.1.b Tenant …….……..15

3.2.1.c Apps ……..16

3.2.1.d Users…….....16

3.2.1.e User Groups …….….16

3.2.1.f Roles …….……17

3.2.2 Common …….....17

3.2.3 Document Service ……17

3.2.4 Master Data……...18

3.2.4.a Master Data Location……..19

3.2.4.b Master Data Type……19

3.2.5 Notify……20

3.3 RESTFul Webservices ……...20

3.3.1 Working ……..20

3.3.2 Client Request ……20

3.3.3 Server Response …….….22

3.4 System Specifications . . . . . …….....22

3.4.1 Software Specification ……....22

3.4.2 Hardware Specification ……...22

3.4.3 Server Specification ……23

3.4.4 Software Description ……...23

3.5 System Design …….27

3.5.1 Logical Design . ……...28

3.6 Input Design ……...29

3.7 Output Design ……30

4. RESULT AND DISCUSSION 32

4.1 Testing methods ……32

4.1.1 Unit Testing with Mockito……...32

4.1.2 Integration Testing……34

4.2 Output Screens and Results …….....35

5. CONCLUSION 36

5.1 Future Enhancement ……..……..36

6. REFERENCES 37

APPENDIX 39

List of Figures

3.1 Architectural Design……..…13
3.2 Module Design……..…14
3.2 Master Data Design…….……18
3.3.2 REST API Architecture…….…..…...21
4.1.2 Testing Modules……....31
4.2.1 UserGroup Service Test……35
4.2.2 Master Data Service Test……..35
A.1 UserGroup Controller…….39
A.2 Master Data Controller……...40
A.6 UserGroup Service Test Coverage……..40
A.7 Document Service Test Coverage.....41

CHAPTER 1
INTRODUCTION

Event Platform is a system being developed for Tekact Private Limited to publish target-based events. This is a product based on microservices architecture where each microservice handles problems specific to a particular domain. The users will be creating apps by stitching those various microservices. This product aims to provide APIs for users to compose Apps that can work with various domains. Event Platform is a multi-tenant system. It employs a software engineering methodology that emphasizes breaking down an application into discrete modules with clear interfaces.

Creating reusable services and application programming interfaces (APIs) enables faster time to market for new products, which is being driven by an increasing number of organisations. This microservices-based project demonstrates how multiple services can operate independently by leveraging the most effective microservices patterns for scalability, performance, and resilience. Focused on the customer, its vision, its products, and its people, the software organisation. Beginning with consultation, provide comprehensive services. And work in every phase of the development life cycle, in addition to assisting with the organization's setup in India with all legal compliances. The company's primary services consist of Software Development, Consulting, Quality Assurance, and Build Operate Transfer.

1.1 Objective

The project aims to achieve the following for the Event Platform product:

1. Provide APIs to create Apps

   Using Spring Boot for your REST APIs has a number of advantages, such as: No intricate XML configuration is necessary. Tomcat server embedded for Spring Boot applications.

2. Provide ways to compose apps using such APIs

   The REST API is merely a web application, much like all the others we have ever made. The only distinction is that it sends data in response to a GET request rather than displaying a web page.

3. Easy integration via usage of REST APIs

   The list of classes and functions we use to create Java code is known as the Java API. The idea is that an API is a set of operations that can be performed when writing code. Therefore, when we say that we are developing a REST API, we simply mean that we are utilizing REST concepts to develop a tool that programmers can use to interact with our data.

4. Workflow management

   Workflow management refers to the identification, organization, and coordination of a particular set of tasks that produce a specific outcome.

5. Provide a working App

   The scrum model is used in this project to deliver functional apps. A small team is led by a Scrum master in this Agile project management methodology, whose primary responsibility is to eliminate any barriers to getting work done. The team meets every day to discuss upcoming tasks and any obstacles that need to be removed. Work is completed in brief cycles known as sprints.

1.2 Company Profile

Tekact Private Limited technology services and product development organization founded by technology experts with a vision to provide Innovations with open-source technologies. Tekact builds innovative products catering to web, mobile and cloud with superior quality measures. The organization is founded and backed by industry leaders having tenures in the past with Oracle, IBM, HSBC & Yahoo!. The experts in the organization contributes to various open-source technologies provide innovative solutions and products for enterprises and startups. The team of Architects, Developers, Testers and Operations are experts in their area of work and are so obsessive and passionate about it. The organization mainly focuses on software development, Quality Assurance, Consulting, Build operate transfer.

Services:

1. **Software development**
   - **Big Data**

     We provide Big Data development services that enable organizations to take advantage of many opportunities, such as better information management, better pricing, better operational efficiency, increased sales and loyalty, greater responsiveness, data-driven planning and more.
   - **Aggregation of data**

     Data aggregation is the process by which unprocessed data from various sources is collected and made accessible for additional analytics.
   - **Validation of Data**

     Before importing and processing your data, you should usually perform data validation to ensure its quality and accuracy.
   - **Data Management**

     The process and management of a company's use of data's availability, usability, integrity, and security are known as big data governance.
   - Data Visualization & Analytics

     Individuals and organizations can make sense of data with the aid of data analytics. To provide business value, raw data is transformed from a variety of sources.

2. **Quality Assurance**

   - **Functional Test**

     Meet your customer expectations with accurate software. We ensure that the business feature works as per the software requirements
   - **Performance Test**

     With our experts ensure that the ability, speed, scalability, and responsiveness of an application holds up, even as your customer size grow
   - **Pen Test**

     Evaluate the security of the system - including applications, network & infrastructure, so that you don’t fall victim to the vulnerabilities.

- **Automation**

  Automate your quality checks with our Automation experts. This will help you with regressions and assure that the application/services are not broken with rapid changes.
- **API Test**

  Ensure that your APIs are running with accurate functionality, reliability, performance, and security
- **Big Data Test**

  Accurate data helps businesses to better decision making and market advantages. Our experts help the verify and validate the date before its used for business ends

3. Consulting

- **Architecture & Design**

  Build your applications on the best of the platforms employing the best patterns and practices in the industry. We apply the best of the enterprise patterns to achieve maintainable and performant software.
- **Best Practices**

  We help your team adopt best of the technology and process practices in the software industry
- **Cloud Computing**

  We help you to identify and gain the best possible business values from cloud computing. We help you with identification of the right cloud vendor, migration, design and implementation of applications in the cloud

4. Build Operate Transfer

- Our experts will help you to build your organization with top notch talent, operate the teams with quality timely deliveries with well-equipped process, and transfer the teams to the needy.

CHAPTER 2

LITERATURE SURVEY

Literary review is the in-depth study and interpretation of literature relevant to a particular topic. Using literature reviews, relevant research questions are identified, and then one seeks to answer those research questions by searching and analyzing relevant literature. A certain importance of literature reviews is that new ideas can be developed by re-analyzing the results of the research. The whole and current state of knowledge on a certain topic as found in scholarly books and journal articles is summarized and explained in a literature review. There are two different kinds of literary critique that one might write in college: one that is written as part of an introduction or as preparation for a lengthier course, and one that is written as a stand-alone exercise in a course. a thesis or research paper, respectively. The type of review that was produced will affect the review's focus, point of view, and type of hypothesis or thesis. Reading the introductory chapters of theses and dissertations in our subject or published literary reviews can help you comprehend the differences between the two by allowing you to analyze the argument structure and how the authors approach the problem.

2.1. PURPOSE OF THE LITERATURE REVIEW

1. It allows readers to easily access research on a particular topic by selecting relevant, important, important and valid articles or research and summarizing them in one journal. comprehensive report.

2. It provides an excellent starting point for researchers beginning to study a new field by forcing them to summarize, evaluate, and compare original research in that particular field.

3. It ensures that researchers do not duplicate work already done.

4. It can provide clues about future research directions or suggest areas to focus on

5. It highlights the main results.

6. It points out the document's gaps, contradictions, and inconsistencies.

7. It provides a constructive analysis of the methodology and approaches of other researchers.

2.2. RELATED WORKS

The section mainly describes related study modes in the area of efficient fire detection. And some of them are listed below.

2.2.1 MICROSERVICE

A microservices architecture divides the business domain into discrete, logically distinct contexts that are carried out by independent, decoupled, self-contained services. Netflix, one of the early adopters of microservices, began moving away from monolithic architecture in 2009, before the term "microservices" was even coined. Applications built using microservices scale effectively horizontally, both technically and in terms of the structure of the organization's developer teams, which may be more compact and responsive. It has become common practice to make an effort to seamlessly incorporate such options into adaptive business process management. Additionally, dividing up large applications into separate, little services gives agile teams more independence, supporting the scaling of agile techniques. Each team can focus on various subservices and create user stories that only have an impact on those subservices. [1]

2.2.2 SPRING BOOT

Popular for developing web and enterprise applications, the Java-based Spring Boot framework offers flexibility for service-oriented architecture (SOA). All Spring-based applications face some difficulties due to the configuration's complexity. With very little Spring configuration, Spring Boot makes it simple to create and deploy production-quality, standalone Spring applications. Spring Boot makes dependency management easy by using a comprehensive yet flexible framework and linking libraries into a single dependency. provides all the Spring related technology you need for startup projects compared to CRUD web applications. This framework provides a wide range of additional features that are common to many projects,

such as server integration, security, metrics, health checks, and more. outsourced configuration. Web applications are usually packaged as a war and deployed to a web server, but a Spring Boot application can be packaged as a war or jar file, allowing the application to run without installation. and/or configuration on the application server [2].

2.2.3 OCR

Before being recognized, lines of text, words, and symbols in a document must be segmented into the proper sections for optical character recognition (OCR). There is a tone of textual information online in both Hindi and many other languages, including English, which is the most widely used language in the world. However, English-specific tools and methods make up the majority of the market. This implies that an effective text extraction tool for languages other than English is required. To circumvent this limitation, we have suggested in this article a software interface or web application that performs automatic translation of the image document into any language supported by the Googletrans library. Python-tesseract is an optical character recognition (OCR) engine for Python software. In other words, it will recognize and read deeply inserted text in a portion of the image. while The Google Trans Python library implements the Google Translate API at no cost and with no limitations. Create web applications that accept document image files as input and output translation using explanatory concepts. The digitized input image is subjected to a number of operations during the preprocessing phase. The image is improved by segmenting it after noise reduction and creating and rendering it to divide something into smaller parts. [3]

2.2.4 TESSERACT

Tesseract is intended to be a language-neutral system. The Tesseract's initial objective was to identify white against a black background. This results in design in terms of mining on part sketches and related parts analysis. To extract images from a text, Tesseract-OCR is used. Tesseract is a collection of exceptionally optimised algorithms. Tesseract-OCR is implemented by the Python "Python-tesseract" module to convert images to text. Its implementation is very simple because all we have to do to turn the provided image into a string is include the module and call the defined method image to string(image, lang = 'lang code'). The language in which the document's text is written must be made clear. Tesseract transforms existing text into

an image and then returns it. Tesseract-OCR has an efficient implementation and is a very efficient and maintainable library.The library function image to string (image file, language) supports the following file types: ".jpg," ".png," ".gif," ".bmp," ".tiff," and ".bmp" [4].

2.2.5 POSTGRESQL

PostgreSQL is an effective open-source antisocial database framework. It has more than fifteen years of active development, a tried-and-true design, and a proven track record of dependability, information accuracy, and accuracy. PostgreSQL, also known as PostgreSQL, was created by a global team of volunteers as an open-source social database management system (DBMS). PostgreSQL's source code is freely accessible and not under the control of any other business or government entity. PostgreSQL, formerly known as Postgres, was created at UC Berkeley by professor of software engineering Michael Stonebraker. Postgres was introduced by Stonebraker in 1986 as a continuation of Ingres, which has since been acquired by Computer Associates. PostgreSQL remains compatible with all of the major desktop frameworks, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows. It integrates programming interfaces for C/C, Java, Perl, Python, Ruby, Tcl, and Open Database Connectivity to improve content, images, audio, and video (ODBC). Lists facilitate faster access to information. In general, adding records to a segment expedites the retrieval of information. However, the trade-off is that you will integrate information more slowly for each record you have. Essentially, when you embed information in a list, the list must organize the information in two locations and preserve the arrangement in the file [5].

2.2.6 RESTFUL API

The proposed system has attempted to implement a RESTful API web service in this article, which will undoubtedly make it easier to develop software applications that don't use the system or run-on other languages or platforms. various programming. In this study, a takeaway application's web service architecture will be developed using RESTful APIs. The backend of the Takeaway app is a website, and the front end is an Android-based platform. The functional method test results using the Postman application demonstrate that REST The API server that

was constructed over the server is functional. The Apache JMeter application can be used to test the application's response time, and the results show a good response time. While this is going on, a comparison of the SOAP and REST architectures' requests and responses reveals that REST takes longer [6].

2.2.7 THYMELEAF

Thymeleaf is a Java template engine that can be used in non-web and servlet-based applications and environments. It works well with offline applications, but is best suited for HTML5 web applications. It essentially replaces JSP due to the fact that its template files can be accessed directly from the browser. The following are the primary characteristics: - Suitable for comprehensive documentation. If necessary, it can serve as a model engine frame. It can identify the document type and convert accordingly. With the Parsing Template Cache feature, which offers high performance and efficiency, it is possible to reduce input output. It can function in both online and offline settings. The templates act as prototypes, enabling the development of a natural templating system, which is the primary advantage of this tool over others. Without any running servers or "static rendering," browsers display models. In essence, it resembles a template engine framework [7].

2.2.8 SWAGGER

The OpenAPI or Swagger specification has become the first choice to describe web APIs (REST, RESTful, etc.). This is why this specification has been and is the focus of many research papers. Many software development related activities, tasks and problems can be improved, simplified, solved and/or automated with this specification. The purpose of this study is to summarize the use and application of the OpenAPI/Swagger specification. The development activities in which it is useful, the software artifacts (results) generated from it, and the areas in question, among others, are identified. The literature review we analyzed included a total of 7 articles published between 2011 and 2020. Searches that fit these parameters: a) refer to development activities where the specification is useful. benefits, b) includes the software artifacts (results) it generates and their respective assertions, and c) references to the domains in question. The results show that the OpenAPI/Swagger

specification has benefited many software development activities, with improved documentation being the most dominant (43%). Similarly, 77% of articles provide solutions to web API consumer needs and/or problems. The automation of tasks, based on different tools, generated the most significant results in 68% of the manuscripts. In addition, 40% of the works consulted deals with a specific domain (IoT, Cloud, etc.) [8].

2.2.9 MAVEN

Maven is a potent POM-based project management tool (project object model). Projects, dependencies, and documentation are built using it. Maven makes it easier for Java developers to complete their daily tasks and frequently aids in organizing any Java-based project. A JAR file directory with some metadata is what makes up the Maven repository. POM files called metadata are linked to the projects to which each packaged JAR file and its external dependencies belong. This metadata enables Maven to download all of your dependencies repeatedly until they are all downloaded and installed on your local machine. Maven builds projects in a series of six build phases, each of which depends on the one before it. The following is a description of the phases that make up the Maven1 lifecycle by default: The project's validation ensures that all of the parameters needed to build it are accurately specified. The project's source code should be compiled into raw distribution files (e.g. .class files). To check for errors, the test suite is compiled and run. packaging that transforms received raw goods into the desired deliverable form (e.g.. Jar,. war,. ear). The delivered goods are installed on the local system in the intended location or locations. Local installations are pushed into the production environment by deployment [9].

2.2.10 POSTMAN

A test in Postman is essentially a piece of JavaScript code that executes after a request is sent and a response is received from the server. POSTMAN is very easy to use. To test the APIs of the application, one must adhere to the set of API calls that it provides. For each request, Postman allows you to create and run JavaScript-based tests. first Get the Postman application. For the operating systems macOS, Windows, and Linux, it is a native application. The Chrome app version is also available. If you want to, click on the square in the Applications section.

Are using Chrome to access it. The most popular techniques are GET (Get Data), POST (Update Data in Existing File), PUT (Replace Existing File). Send request: a. Click the new button in the title toolbar. b. Type imageURL-postman-echo.com/get as the request URL (URL - rest endpoint) is specified below. By selecting the Save button located next to the Submit button, the user can save the request for later use. Utilize Newman to integrate with construction management tools. With its command line companion, Postman, you can automate testing by integrating Postman with a build management tool. Authorization: The authorization procedure determines if you have permission to access the server's desired data. Testing for security includes doing this. [10]. Send request: a. Click the new button in the title toolbar. b. Type imageURL-postman-echo.com/get as the request URL (URL - rest endpoint) is specified below. By selecting the Save button located next to the Submit button, the user can save the request for later use. Utilize Newman to integrate with construction management tools. With its command line companion, Postman, you can automate testing by integrating Postman with a build management tool. Authorization: The authorization procedure determines if you have permission to access the server's desired data[10].

CHAPTER 3
METHODOLOGY

Event Platform is a multi-tenant system. It employs a software engineering methodology that emphasizes breaking down an application into discrete modules with clear interfaces. The users will be creating apps by stitching those various microservices. This product aims to provide APIs for users to compose Apps that can work with various domains. Event Platform is a multi-tenant system. In an event-driven architecture, when a service performs a task that may be of interest to other services, that service generates an event, a record of the action performed. This project proposes a microservice architecture multi-tenant product for publishing targeted events. And this is to provide APIs for users to compose applications that can work with different domains.

In “Event Platform”, the CRUD (create, read, update and delete) and list operations is done using the Rest API and Spring boot framework. While running it goes through the controller along with the path variables and request parameters that specified in the URL. After that it invokes the methods that are declared in service class and that needs to be autowired in the controller. The interface then calls the service implementation class and execute the code using the parameters that are specified in the Rest API calls. The implementation invokes respective models and entities respectively. That means when client sends a request using postman:

- It passes through the controller class
- Then it goes through service class and it gets implemented from the implementation class
- After that it takes respective dto
- Then data goes to PostgreSQL database
- While calling an API, a model can be send as body.
- The model reaches the implementation class and run the code written in the class then it add the model to entity which represent the Database.

- The Repo(interface) is extended by either BaseRepository or JPARepository which include some basic methods like findById (), findAll () etc.
- The Entity class extends BaseEntity which includes some common entities.
- Rendering micro-services.

3.1 Architectural Design

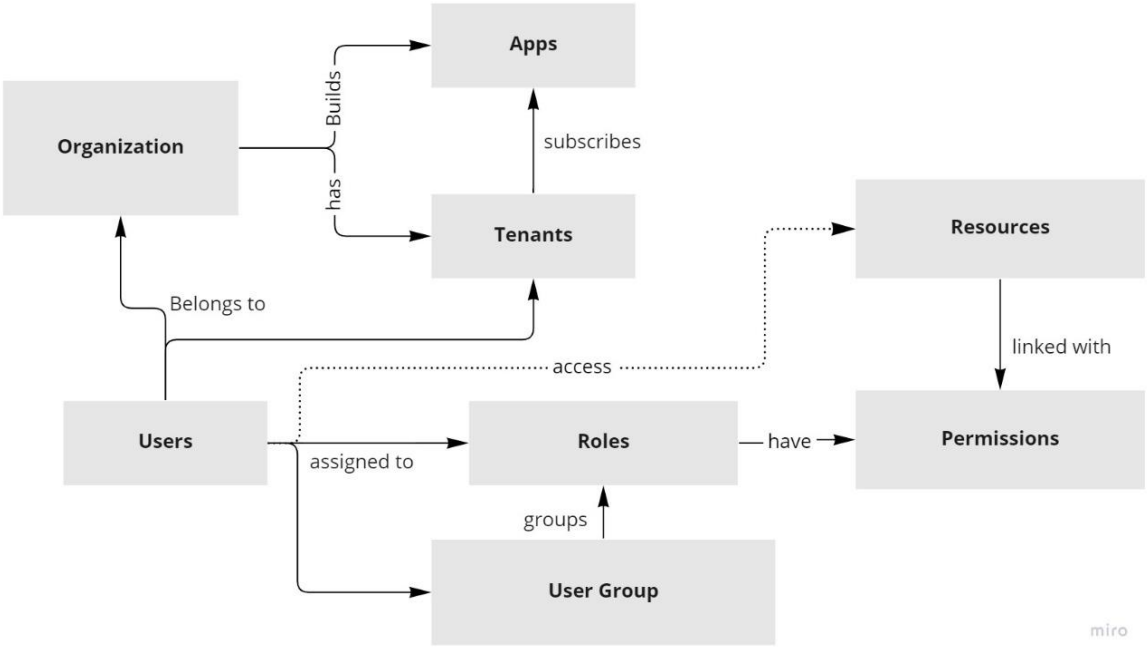


Figure 3.1: Architectural Design

Organization is the parent entity; all other entities are the child entities of organization. Organization has tenants which subscribes for Apps and organization builds Apps. Users belongs to Organization and tenants. Users are assigned to roles and user groups. Users can access the Resources which are linked with Permissions. The software architecture of a computing system is the structure of the system, which consists of software components, the externally observable properties of those components, and their relationships; it defines the relationship between the program's major structural

elements. This computer program's modular structure can be derived from the analysis models and subsystem interactions within the analysis model. The primary objective is to create a modular program structure and represent the modules' relationship. The process of identifying the subsystems that comprise a system and developing the framework for subsystem control and communication is architectural design. A software architecture description is the result of the design process.

3.2 Module Description

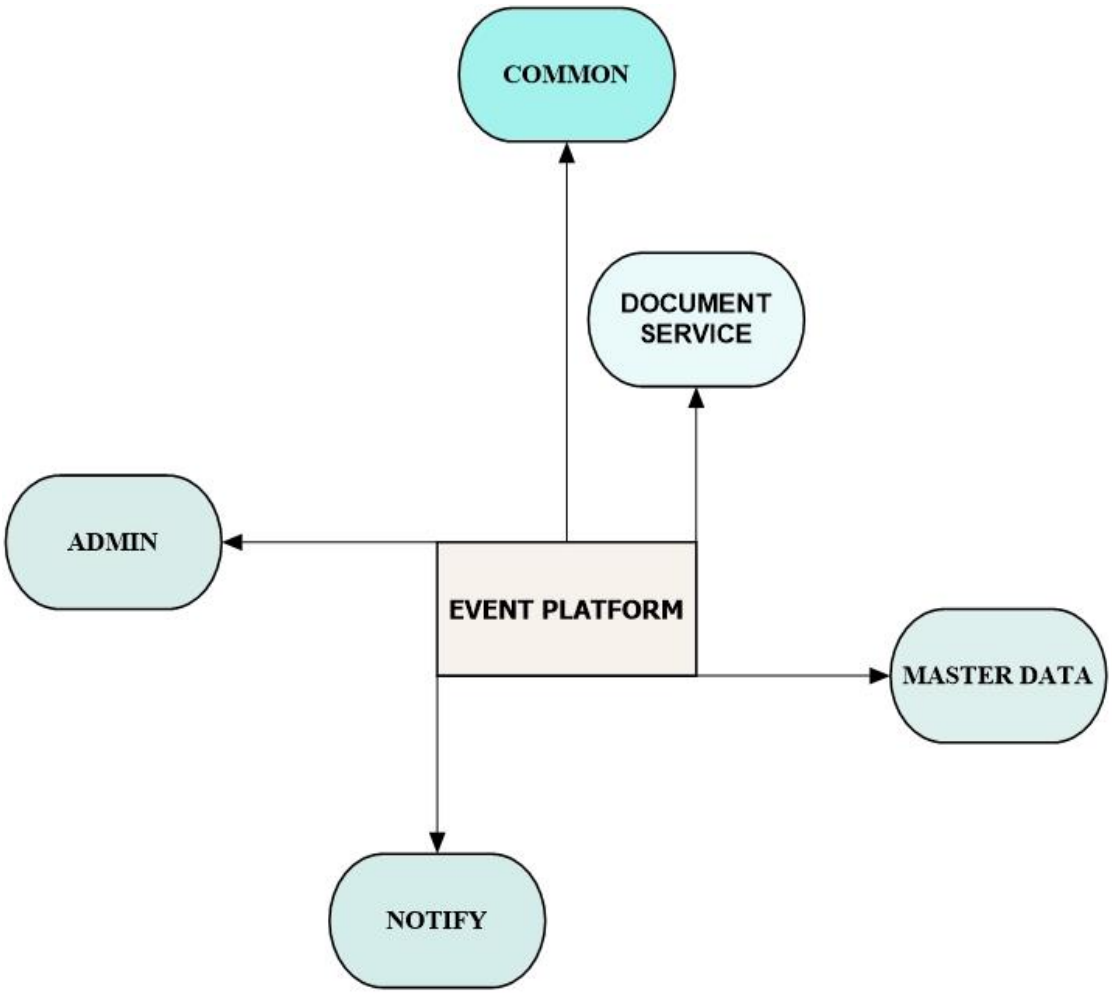


Figure 3.2: Module Design

3.2.1 Admin

Admin module handles all the basic entities like organization, tenant, app, user, user groups, roles. The Administration module allows you to control the operation of Business Process Server, install and uninstall applications, and manage all users and groups. The admin functionalities like managing, authorization, authentication of users is all performing under this module. Mainly this Platform service provides the Apis for the organization, tenants and users. The administrative functions like adding, removing, approving, updating users. Also, the adding, removing, approving and updating of organizations, end users, tenants, different user roles and build apps

3.2.1.a Organization

Organization is the parent entity of all entities in this project. For example, Tekact as an organization it provides various APIs which can be used by tenants to build apps. All other entities like tenant, user, role, app depends on this parent entity. CRUD and List APIs for Organization: As an Admin, I should be able to create, read, update and delete (soft delete) an organization. API should be available to retrieve all the organizations in the database. Authorization as admin firstly demonstrate creation of the entity then get the entity from the database, update the entity. Soft delete the entity in database And List all the organizations which are active. Deleted organizations should not be listed. The above mentioned apis are created. After that code coverage of at least 80% and no static violations is attained in this section.

3.2.1.b Tenant

Tenants are the Consumer that subscribes various Apps built by the organization. Tenants can be a single user or a complete organization. For an organization we can have multiple tenants. Tenant subscribes to various apps build by the organization to achieve their needs. CRUD and List APIs for Tenants: As a Tenant, I should be able to create, read, update and delete (soft delete). API should be available to retrieve all the tenants in the database. Tenant have to register by providing the details to database, After the creation of the entity, get the entity from

the database, update the entity. Soft delete the entity in database And List all the tenants which are active. Deleted tenants should not be listed. The above mentioned apis are created. After that code coverage of at least 80% and no static violations is attained in this section.

3.2.1.c Apps

Apps are the products that are created by composing various APIs. As an Organizational Admin, I should be able to create, read, update and delete (soft delete) apps. API should be available to retrieve all the apps in the database. I also should be able to assign apps to organizations Apps are build by the organization. Such apps are subscribed by the tenants. Apps are controlled and managed by the organization and administrator.

3.2.1.d Users

One organization can have multiple users. They belong to either organization or tenants. Giving the access to the users are controlled by the administrator. Users can be superusers, admin, end-users, tenants etc. As an Organizational Admin, I should be able to create, read, update and delete (soft delete) users. API should be available to retrieve all the users in the database. I also should be able to assign apps to organizations. Multiple roles can be assigned to a single user. User can be assigned to user group.

3.2.1.e User Groups

User group is a collection of users with a given set of permissions assigned to the groups and transitively to the users. When the users are associated with a user group, they inherit the permission of the user groups. As an Organizational Admin, I should be able to create, read, update and delete (soft delete) User Groups for a Tenant and associate roles to UserGroups. API should be available to retrieve all the user groups for the given tenant in the database. I also should be able to assign roles to users. Members of the user group can be a user and other groups eg: Employees, Developers etc.

CRUD operation of UserGroup include create, read, update and delete. POST method is used for create usergroup by passing orgid. For updating the values in the UserGroup use PUT method. There two GET method are used here first one is to get all the details of usegroup by passing orgid as pathvariable and the second one is to get the particular details of uergroup by

passing usergroup id. Soft delete the entity in database And List all the user group which are active. Deleted user group should not be listed. The above mentioned apis are created. After that code coverage of at least 80% and no static violations is attained in this section.

3.2.1.f Roles

Each user has at least one assigned role. Role management and assignment is controlled by administrator. Roles are assigned to users and user groups, as part of admin I should be able to create, read, update and delete (soft delete) roles. API should be available to retrieve all the roles in the database. I also should be able to assign roles to users Each role has particular permissions to access resources. Roles are also used for role-based access control.

3.2.2 Common

In our project we are following multimodule approach. Thus, for all modules we have common functionalities. All those functionalities used are available in this module. All other module inherits the classes in the common module. Common attributes in all entities, models are mentioned here. This module uses base repo and several features and methods. Also, there is common exception class and its own handler are contained in this module, which is used in another modules called admin, master data, document service etc

3.2.3 Document Service

Document service provides APIs for creating/uploading, updating, downloading, deleting, listing, and retrieving documents. OCR - Generate Metadata from uploaded images. To extract useful information from an image file, Convert image data to grayscale, smooth, de-skew, and filter, for example. Lines, words, and characters are detected. Create a ranked list of potential character candidates based on a trained data set. (here, the setDataPath() method is employed to set the trainer data path). Choose the best recognized characters based on confidence from the previous step and language data. Included in language data are dictionaries, grammar rules, etc. It improves the effectiveness and productivity of office work. The ability to instantly search through content is incredibly useful, particularly in an office environment that must deal with high volumes of document scanning or document influx. OCR is quick, maintaining the integrity of the document's content while saving time. Effort is increased since Employees are

no longer required to perform manual labor, allowing them to work more quickly and efficiently.

3.2.4 Master Data

Master data is the essential information used as the foundation for all transactions. Whether you are producing, transferring inventory, selling, purchasing, or conducting physical inventory, you must maintain certain master data. Data created centrally and applicable to all applications. It remains constant over time, but we must regularly update it. For instance: A vendor is a type of master data used to generate purchase orders and contracts.

Master module is one of parent module admin's offspring. We are free to create any type of module and bundle dependency we desire. The master module manages entities such as country, city, state, contact type, organization type, and address type. This module's entities utilize the base entity from the common module, the base repository from the common repository, and a variety of features and methods from the common module.

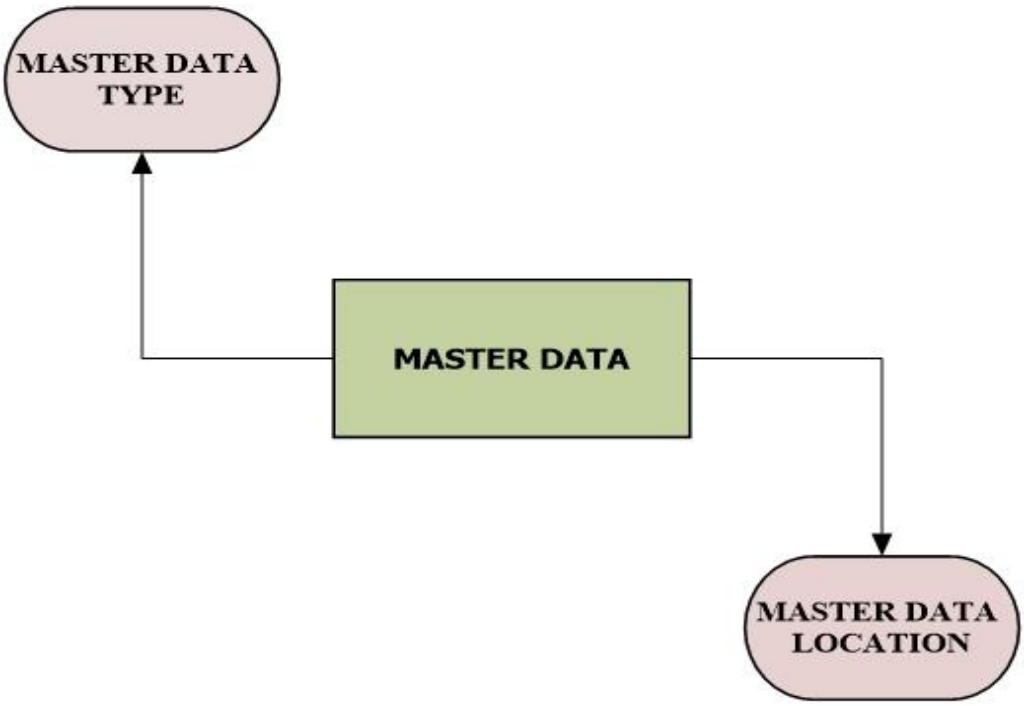


Figure 3.3: Master Data Design

3.2.4.a Master Data Location:

For many APIs, we need to specify details of a location. In a production application, it is important as a user to check which computers, phones, and other devices have recently used your account, to ensure that no one other than you has signed into your account. To improve security, the user must be able to know the geographical position of the equipment which has used his account in order to be able to delete the connections on them in the event that the connections do not come from him. For eg: Address. This entity deals with the creation of Location APIs. Also, the data needed to create the APIs should be developed as Liquibase scripts. In MasterData location we use mainly 3 different entities like city entity, state entity, country entity inorder to achieve the requirements.

3.2.4.b Master Data Type:

In Master data type we need to specify details of a master data types- organization Type, address Type, contact Type, user Type. This story deals with the creation of Master Data APIs. Also, the data needed to create the APIs should be developed as Liquibase scripts. Liquibase is an open-source solution for managing revisions of our database schema scripts. It works across various types of databases, and supports various file formats for defining the DB structure. Also, the data needed to create the APIs should be developed as Liquibase scripts.

Acceptance Criteria includes Liquibase scripts for master data corresponding to organization Type, address Type, contact Type, user Type, get APIs in MasterDataTypeController like get all type for organization Type, address Type, contact Type, user Type ,use cache to store the info at the server side (last priority), the mentioned entities like organization Type, address Type, contact Type, user Type are related to each other such that CRUD operation are performed according with that, test case with code coverage of at least 80% and no static violations.

3.2.5 Notify

This module deals with creating Apis for generating notifications over email. It provides a mechanism for users to get updates related to user onboarding, user registration and the status. This allows users to get timely updates. Acceptance criteria includes get notify medium API as list, get notify events - CRUD + list by get notify events by entity type, API to send email notifications. The data (key, value) to be replaced in template is to be passed as parameters and List of email ids should provide from the client. Separate template is generated for Feedback, User registration, Tenant registration, App and Org registration. Thymeleaf is used for rendering the data and display it in the template.

3.3 RESTful Webservice

REST is an architecture that utilizes the HTTP protocol and is based on web standards. It is built around a resource where each component is a resource and a resource is accessed through a common interface using standard HTTP methods.

3.3.1 Working

Understanding how RESTful APIs communicate data between clients and applications is made easier by splitting the process into two steps.

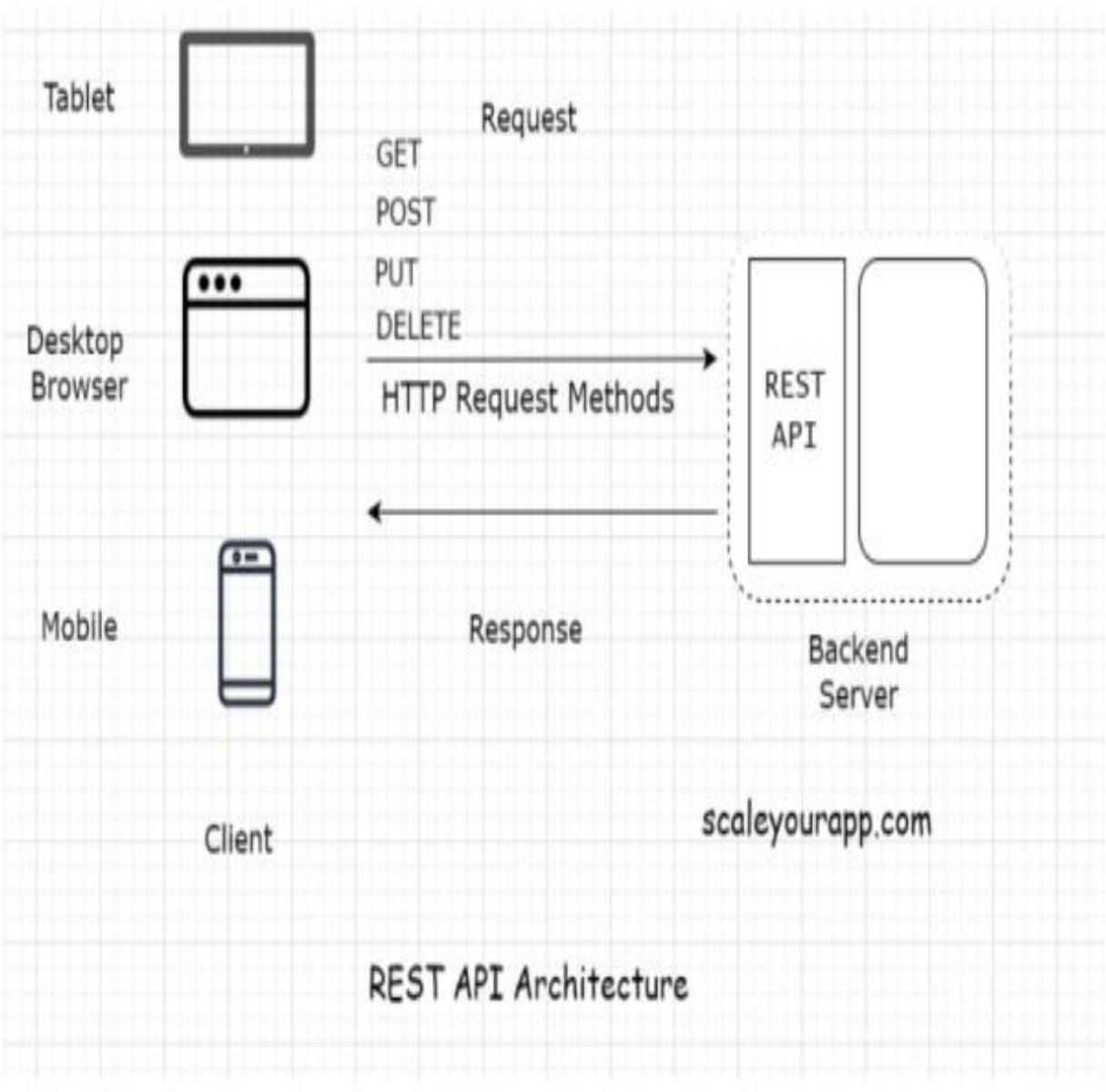
3.3.2 Client Request

Client refers to a program or human utilizing API services. Your software will be a client, for example, if you incorporate a web application like Instagram or YouTube. If you use your browser to request the URL, you will be the client.

The most common HTTP methods are:

- GET: To get a resource.
- POST: To create a new resource.
- PUT: To edit or update an existing resource.
- DELETE: To delete a resource.

For example, a POST request to submit a new picture or a GET request to get a specific video or post would both be HTTP requests to the Instagram API. In a manner similar to this, a contact center platform that includes an auto-answer application may update or remove a personalized greeting using the PUT command.



3.3.3 Server Response

Resources are specific pieces of data that clients of APIs receive. Hashtags, profiles, comments, web addresses, tweets, and more are all examples of possible content. Every resource is hosted on a single server and has a special name called a resource id. When a client uses the RESTful API to submit a request, the server sends a normalized representation of the resource's state to the client's computer. This indicates that instead of sending the client the actual resource, the server is sending a representation of the resource's state, or the state of the resource as of a specific timestamp. Responses are provided in a concise format for simple comprehension. Because it promotes web accessibility and can be read by both humans and machines, the JSON (JavaScript Object Notation) format is well-liked. Additionally, it works with a wide variety of other programming languages. Plain text, XML, YAML, CSV, HTML, and other data formats are alternatives for APIs.

3.4 System Specifications

The application development architecture recognized for this project is specified in this section on the basis of requirements.

3.4.1 Software Specification

- Programming Language: JAVA
- IDE: IntelliJ
- Framework: REST API
- Web Browser: Any web browser
- Database: PostgreSQL

3.4.2 Hardware Specification

A computer with internet connectivity and with minimum 4GB Ram

3.4.3 Server Specification

Using aws t2.xlarge instance per module (with redundancy) vCPU , 16 GB RAM. EBS-based storage of 5 GB per instance. Postgres RDS on AWS will be used as the database. Amazon.com offers a cloud computing platform called Amazon Web Services (AWS), which is a group of online computing services also referred to as web services. There are 11 geographical locations where these services are offered. One could argue that Amazon Elastic Compute Cloud and Amazon S3 are the most important and well-known of these. In contrast to an enterprise customer building a real physical server farm, Amazon markets these products as services to deliver mainframe capacity more quickly and affordably. Here in this project, AWS is used as the Server Cloud Processing platform using Software as a Service (SAAS). AWS is loaded with many services, including all the apache service extensions used in our system.

3.4.4 Software Description

- **JAVA**

  Java, an object-oriented programming language, can be used to create software for a variety of platforms. When a developer creates a Java application, the compiled code (also known as bytecode) is compatible with the majority of operating systems (OS), including Windows, Linux, and Mac OS. The C and C programming languages have a substantial influence on the syntax of Java. Midway through the 1990s, Java was created by Mike Sheridan, Patrick Naughton, and James A. Gosling, a former Sun Microsystems computer scientist. Java's ability to develop applets, which are browser-executable programs, facilitates the graphical user interface (GUI) and the interaction of Internet users with objects. Prior to Java applications, the majority of web pages were non-interactive and static. As competitors such as Adobe Flash and Microsoft Silverlight were introduced, the popularity of Java applications declined. By translating Java bytecode into native processor instructions, the Java Virtual Machine (JVM) enables indirect program execution from the operating system or platform. The Java Virtual Machine provides the majority of the elements required to execute bytecode,

which is typically smaller than program written in other programming languages (JVM). On a system lacking the necessary JVM, it is not possible to execute bytcode. The development of Java program requires a Java software development kit (SDK), which typically consists of a compiler, interpreter, documentation generator, and other tools used to build a complete application.

- **INTELLIJ IDEA**

Use IntelliJ IDEA, a smart, context-aware IDE, to work with Java and other JVM languages like Kotlin, Scala, and Groovy across all sorts of applications. You may also create whole web applications with the help of IntelliJ IDEA Ultimate's powerful built-in tools, support for JavaScript and associated technologies, and enhanced support for well-known frameworks like Spring, Spring Boot, Jakarta EE, Micronaut, Quarkus, and Helidon. You may also add more programming languages to IntelliJ IDEA by using free JetBrains plugins, including Go, Python, SQL, Ruby, and PHP.

- **SPRING BOOT**

The Java Spring Framework (Spring Framework) is a popular open-source enterprise framework for constructing production-ready standalone applications that run on the Java Virtual Machine (JVM). Java Spring Boot (Spring Boot) is a tool that accelerates and simplifies the development of web applications and microservices using Spring Framework through its three primary features:

1. Auto-config
2. Configuration-savvy approach
3. ability to create standalone applications

Together, these features provide a tool that enables you to configure and set up a Spring-based application with minimal configuration and effort. Dependency injection is a feature of the Spring Framework that allows objects to specify their own dependencies, which the Spring container will subsequently include. This makes it possible for developers to create modular, loosely coupled applications, which are perfect for

microservices and distributed network applications. Common application tasks like data binding, type conversion, validation, exception handling, resource and event management, internationalization, etc. are also supported by the Spring Framework's built-in support.

- **REST API**

  A REST API is an API implementation that follows REST architectural constraints. It serves as a conduit. Through HTTP, the client and server communicate with one another. The HTTP methodologies are used by the REST API to set up client-server communication. In order to enhance the performance of applications, REST also permits servers to cache responses. Client-server communication is a stateless process. Any communication between a client and a server, in other words, is news. No data or memories from previous communications are transferred. Therefore, the client must send credentials to the backend whenever it communicates with it. In doing so, the backend is able to ascertain whether the client has permission to access the data. The client interacts with the main endpoints when a REST API is implemented. This totally separates the client code from the backend code. The components of CRUD are Create, Read/Retrieve, Update, and Delete. User interface conventions that enable viewing, searching, and editing of data through forms and reports on a computer are known as CRUD operations. Data-oriented CRUD makes use of standardized HTTP action verbs.

  - POST: Creates a new resource
  - GET: Reads a resource
  - PUT: Updates an existing resource
  - DELETE: Deletes a resource
  - Standard CRUD Operation
  - CREATE Operation: It performs the INSERT statement to create a new record.
  - READ Operation: It reads table records based on the input parameter.
  - UPDATE Operation: It executes an update statement on the table. It is based on the input parameter.
  - DELETE Operation: It deletes a specified row in the table. It is also based on the input parameter

- **JUnit+Mockito**

  Developers can perform unit testing in Java using the JUnit framework, which improves code quality and speeds up programming. Any of the following software can easily be integrated with the JUnit framework:

  • Eclipse

  • Ant

  • Maven

  The JUnit testing framework provides the following important features.

  Fixtures: Fixtures are a set of objects in a fixed state that are used as a benchmark when running tests. The goal of the test fixture is to guarantee that the tests are conducted in a well-known and consistent environment so that the results are repeatable. It contains the setUp() method, which is executed prior to every test call. Following each test method is the drawDown() method. A test suite is a collection of unit test cases that is run all at once. To run a suite of tests in JUnit, use the @RunWith and @Suite annotations. The test classes TestJunit1 and TestJunit2 are used in the following example.

  Test Runners: Test cases are run using test runners. This serves as an illustration presuming the test class TestJunit is already in existence. a JUnit class - JUnit classes, which are used to create and test JUnits, are crucial classes. Several crucial courses include − Contains a collection of assert methods. TestCase: Contains a test case that identifies the machine that is being used to run several tests. TestResult - Consists of methods for gathering test case execution results. Mockito is a mocking framework for Java applications that is used for unit testing. Making applications testable requires the use of mockito. Under license from MIT, Mockito is made available as an open source testing framework (Massachusetts Institute of Technology). It internally creates mock objects for a specific interface using the Java reflection API. The mock object, also known as the proxy used for the actual implementation, is referred to as such. The Mockito framework's main goal is to make the development of tests simpler by simulating external dependencies and utilizing them in the test code.

- **PostgresSQL**

  Both SQL (relational) and JSON (non-relational) queries are supported by PostgreSQL, an advanced, enterprise-grade, open source relational database. Because of its extremely stable database management system, it has high levels of resilience, integrity, and accuracy. Many web, mobile, geospatial, and analytics applications use PostgreSQL as their main datastore or datastore. The most recent significant update is PostgreSQL 12. Here are some examples of PostgreSQL's typical uses. Strong database in the LAPP stack Linux, Apache, PostgreSQL, and PHP is referred to as LAPP (or Python and Perl).Most dynamic websites and web applications are powered by PostgreSQL, a potent back-end database. Both established businesses and start-ups rely on PostgreSQL as their back-end database to support their products and applications. The most widely used programming languages are supported by PostgreSQL, including Python, Java, C#, C/C, Ruby, JavaScript (Node.js), Perl, Go, and Tcl. Outstanding PostgreSQL Features. Other enterprise database management systems typically offer advanced features like user-defined types, table inheritance, complex locking mechanisms, referential integrity foreign keys, views, rules, and subqueries. PostgreSQL also has advanced features like nested transactions (save points), multiple version concurrency control (MVCC), and asynchronous replication.

3.5 System Design

Design has been defined as a multi-step process that synthesizes information requirements to represent data structures, programmed structures, interface characteristics, and procedural detail. All subsequent software engineering and maintenance steps are built on the design phase. It is a decision-making activity, frequently of a structural nature. Design creates coherent, well-thought-out representations of program that focus on related logic operations at a lower level and the Interrelations parts at a higher level. A good design is one that enables the production of efficient code and whose implementation is as compact as possible, depending on the applications and project requirements. Design elements typically include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram

with a full data dictionary. These elements describe the desired software features in detail. These design elements are meant to sufficiently describe the software so that skilled programmers can create it with little additional input design. The following methods to develop a project are specified by the fundamental design concept: - Abstract - Flexibility Data structure, structural partitioning, software architecture, and procedure Two levels of system design exist: - Logical planning - Formal design in logical design, the designer creates a specification of the main system components that achieve the goals. The system's actual design is revealed by the physical layout.

3.5.1 Logical Design

Understanding the link between the different system components is made easier via model analysis. The analysis model clearly shows the user how a system will function. This is the basic technical illustration of a system. Three main objectives must be reached by the analytical modelling. to provide the groundwork for the production of software designs. to outline the needs of the user. Make a list of requirements that can be verified when the programme is developed. The purpose of a use-case diagram is to demonstrate how dynamic a system is. However, this definition falls short of encapsulating the objective completely. because the objectives of the other four diagrams—the State chart, cooperation, sequence, and activity—are all the same. We will thus look at the function that sets it apart from the other four diagrams. Use case diagrams are used to gather both a system's internal and external needs. Most of them are design requirements. As a result, use cases are developed and actors are identified when a system is studied to establish its functions. Use case diagrams are made to show the external viewpoint once the primary work is finished. In conclusion, use case diagrams perform the following tasks: used to gather system specifications. used to see how a system seems from the outside. Identify the system's external and internal impacts. Actors should illustrate how the requirements interact.

3.6 Input Design

Input design is the process of transforming user-generated data into a computer-based format. The input-handling design decisions specify how data are accepted for computer processing. Input design is a component of system design that requires careful consideration and entails defining the means by which actions are to be performed. Input design involves the acceptance of data for computer processing and the input of data into the system via mapping using map support or links. Inaccurate input data is the leading cause of data processing errors. The design of input is concerned with the following points:

The first step in input design is determining which data will be input. JSON offers the same advantages as XML for exchanging data in a heterogeneous environment, such as the following, when used for data input on the 'Tekact Platform'. JSON is self-explanatory. In some instances, the syntax and hierarchical structure of JSON strings can be interpreted by applications that are unfamiliar with the expected data.

JSON is a textual format. This makes it appropriate and secure for transfer across platforms and operating systems that do not readily share more complex document formats. As text, JSON can be displayed and edited with ease in simple editors. JSON is condensed. A typical JSON string is approximately two-thirds the size of the equivalent XML string. JSON is simple to learn, simple to read, and simple to comprehend.

The objectives of the input design are:

1. Controlling amount

2. Avoiding delay.

3. Avoiding errors.

4. Avoiding extra steps

The features of the input design are:

• The input designing is done so as to have the most efficient way for interaction between the user and the system.

• Measures have been taken to minimize user inputs.

• Extra steps are eliminated and the process is made simple**.**

3.7 Output Design

Data and results generated by a system are referred to as output. Make decisions regarding the information to present, the layout and output medium, the arrangement of the information presentation in a recognized format, and the distribution of the output to the target audience. specifies the style, location, and pagination of column headings. The output design is crucial in supplying the user with the appropriate format. The layout and design are carefully considered, as the main purpose of the output is to convey information. The information should be carefully adjusted to suit the needs of the user. The printout standard suggests that each output be named or titled, an example output layout is given, and steps are defined to ensure the accuracy of the output data.

Consider your output device based on system compatibility, response time requirements, and print quality requirements. When designing the output form, attention is paid to proper identification and representation, readability and ease of use, composition and layout, order of data elements, and clarity of instructions.

A well-formatted form with well-identified captions should be self-explanatory. Organizational forms need to be centralized for efficient processing. The most important and direct source of information for users is computer output. Output design is a process that involves designing the required output in the form of reports that need to be provided to the user based on the user's needs. An efficient and easy-to-understand output design needs to strengthen the relationship between the system and the user and facilitate decision making. Because management refers to reports to make decisions and conclusions, reports must be designed with great care, and details must be simple, descriptive, and easy for users to understand.

Therefore, the following factors should be considered when designing the output: Specifies the information to display.

• Arrange the presentation of information in an acceptable format.

• Specifies how to map the output to the expected receipt.

• Appears on the monitor for immediate use or hard copy, depending on the type of problem and future intended use.

Effective and intelligent output design should improve the system-user relationship and help decision making. Therefore, the system can be viewed or printed according to the user's preference. High-quality output meets the needs of end users and presents information in a clear, readable, and aesthetically pleasing way. Before deciding on the proper presentation and format, there are several factors to consider, such as who will receive the results and under what circumstances.

CHAPTER 4

RESULT AND DISCUSSION

Creating reusable services/APIs enables a quicker time to market for new products, which is being driven by an increasing number of organizations today. This microservices-based project illustrates how multiple services can operate independently by leveraging the most effective microservices patterns for scalability, performance, and resilience. The project aims to accomplish the following for Event Platform: provide APIs to create Apps, provide ways to compose apps using such APIs, facilitate integration through the use of REST APIs, provide workflow management, and provide a functioning App. This product is based on a microservices architecture in which each microservice handles domain-specific problems. Users will create applications by combining these various microservices. This product aims to provide APIs for users to compose domain-agnostic applications. Event Platform is a system with multiple tenants. It adheres to a software engineering methodology that emphasizes the separation of an application into single-function modules with well-defined interfaces. The incorporation of efforts to integrate such options seamlessly into adaptive business process management has become standard business practice. In addition, separating large applications into individual microservices gives agile teams the next level of independence, which facilitates the scaling of agile methods. Each team is able to work on a variety of microservices and create user stories that only affect their microservices.

4.1 Testing methods

There are different types of test methods available. For this project, tests are performed to uncover requirements, design or coding errors in the program.

4.1.1 Unit Testing with Mockito

In the software development process known as unit testing, the smallest testable components of an application are individually examined. unusual and independent for common use. Unit testing is a crucial step in the development process because, when performed correctly, it can help find bugs in the code before they become more difficult to detect later on. A component of test-driven development (TDD), a systematic methodology. Unit testing is the process of building a product through ongoing testing and modification. This testing method is also the

first level of software testing before employing other techniques such as integration testing. Typically, unit tests are isolated to ensure that no external code or functionality is required. Although tests can be conducted manually, they are typically automated. Mockito is a popular open source framework for simulating software testing objects. Mockito makes the creation of tests for classes with external dependencies extraordinarily simpler. A mock object is an instance of a class or interface that is fictitious. It permits the output of particular method calls to be specified. Testing can confirm that user interactions with the system are typically recorded. Plan, cases and scenarios, and the unit test are the three typical steps of unit testing. First, unit tests are created and examined. In the first phase, the unit test is assessed and prepared. Test-driven development requires developers to create unit tests that fail completely. They continue to code and refactor the application once the test passes. TDD frequently results in a clean and dependable codebase.

Module testing is a type of software testing that examines specific classes, methods, subprograms, or subroutines within an application. Module testing suggests testing the program's smaller building components rather than the entire program. The majority of module testing focuses on white-box scenarios. The objective of module testing is to identify module errors, not to demonstrate the module's functionality. Utilize one or more white box methods to analyze the module's logic, and then supplement these test cases with black box methods to examine the module's specification. After the test case has been created, the module for testing must be assembled. For this, either an incremental or non-incremental method is used. Technique that is not incremental; each module is evaluated separately. It first combines all of the program's modules, and then tests the entire application. Using an incremental strategy, each module is tested before being added to the tested collection incrementally. It follows a sequential procedure. There are two methods for incremental testing retesting: top-down and bottom-up examination to run the module with the specified data, it requires a driver to provide the test data, monitor the execution, and record the results.

4.1.2 Integration Testing

After unit testing, integration testing is the next step in the software testing process. Individual software units or components are tested in groups during this testing. The purpose of the integration test level is to reveal flaws when the integrated parts or units interact. Modules are used in unit testing for testing purposes, and integration testing combines and tests these modules. The software is created using a number of software modules that were each programmed by a different programmer. Integration testing's goal is to confirm that all modules are communicating properly.

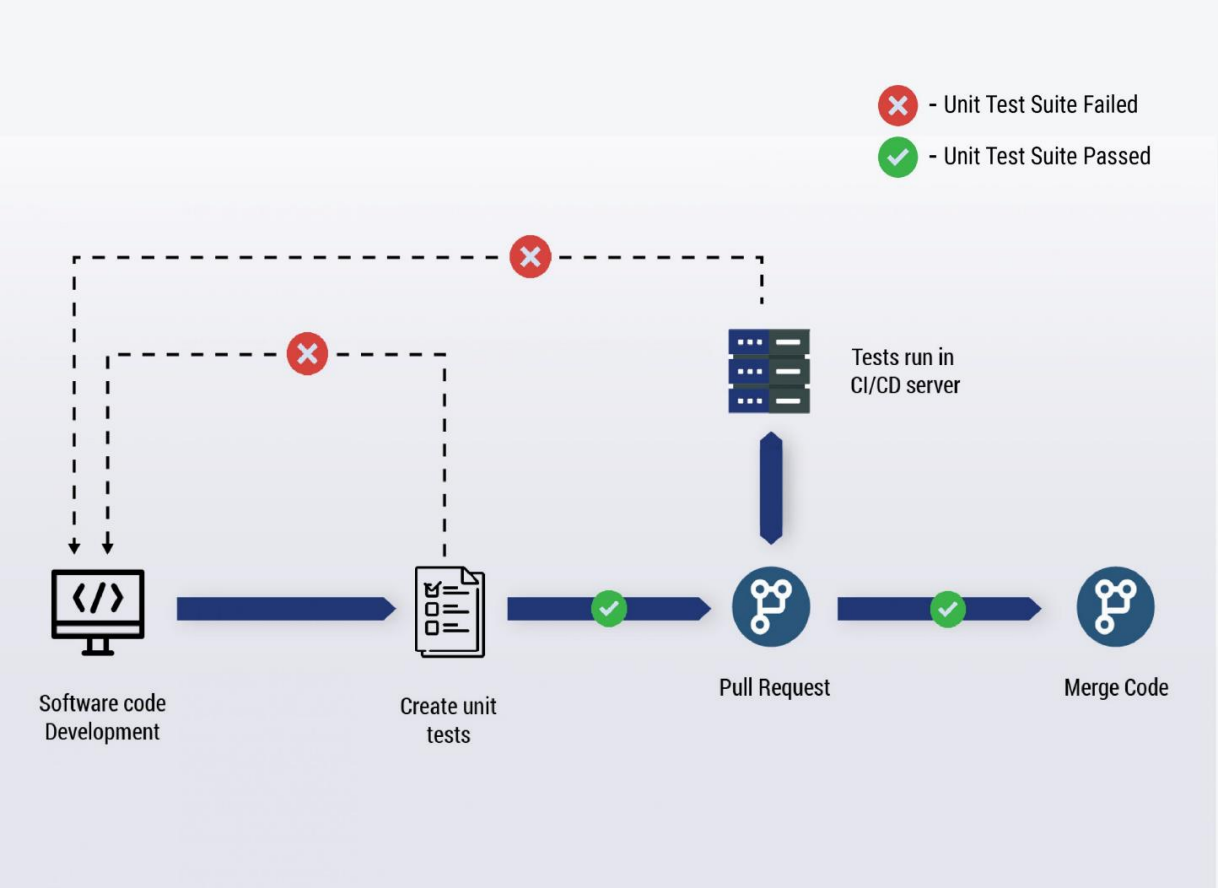
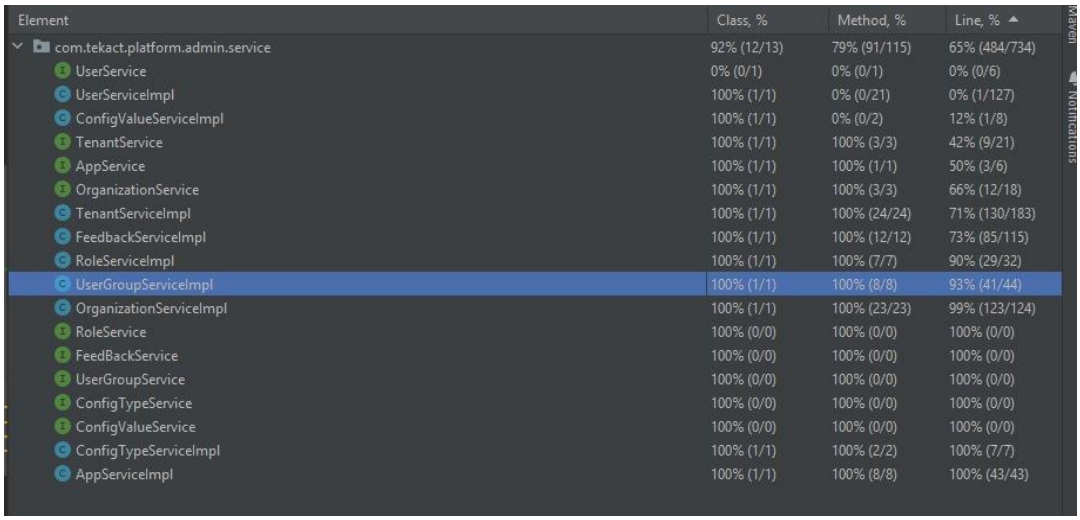


Figure 4.1.2: Modules Testing

4.2 Output Screens and Results

1. **UserGroup Controller**

   As an Admin, I should be able to create, read, update and delete (soft delete) an on UserGroup. API should be available to retrieve all the organizations in the database.

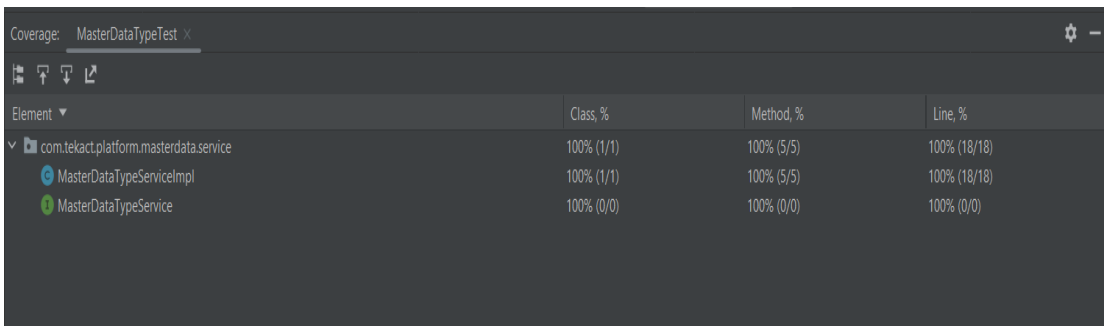


Element	Class, %	Method, %	Line, %
com.tekact.platform.admin.service	92% (12/13)	79% (91/115)	65% (484/734)
UserService	0% (0/1)	0% (0/1)	0% (0/6)
UserServiceImpl	100% (1/1)	0% (0/21)	0% (1/127)
ConfigValueServiceImpl	100% (1/1)	0% (0/2)	12% (1/8)
TenantService	100% (1/1)	100% (3/3)	42% (9/21)
AppService	100% (1/1)	100% (1/1)	50% (3/6)
OrganizationService	100% (1/1)	100% (3/3)	66% (12/18)
TenantServiceImpl	100% (1/1)	100% (24/24)	71% (130/183)
FeedbackServiceImpl	100% (1/1)	100% (12/12)	73% (85/115)
RoleServiceImpl	100% (1/1)	100% (7/7)	90% (29/32)
UserGroupServiceImpl	100% (1/1)	100% (8/8)	93% (41/44)
OrganizationServiceImpl	100% (1/1)	100% (23/23)	99% (123/124)
RoleService	100% (0/0)	100% (0/0)	100% (0/0)
FeedBackService	100% (0/0)	100% (0/0)	100% (0/0)
UserGroupService	100% (0/0)	100% (0/0)	100% (0/0)
ConfigTypeService	100% (0/0)	100% (0/0)	100% (0/0)
ConfigValueService	100% (0/0)	100% (0/0)	100% (0/0)
ConfigTypeServiceImpl	100% (1/1)	100% (2/2)	100% (7/7)
AppServiceImpl	100% (1/1)	100% (8/8)	100% (43/43)

Figure 4.2.1 UserGroup Service Test

2. **Master Data Type Controller**

   In Master data type we need to specify details of a master data types- organization Type, address Type, contact Type, user Type. Get all type for organization Type, address Type, contact Type, user Type, use cache to store the info at the server side mentioned entities like organization Type, address Type, contact Type, user Type.



Coverage: MasterDataTypeTest

Element	Class, %	Method, %	Line, %
com.tekact.platform.masterdata.service	100% (1/1)	100% (5/5)	100% (18/18)
MasterDataTypeServiceImpl	100% (1/1)	100% (5/5)	100% (18/18)
MasterDataTypeService	100% (0/0)	100% (0/0)	100% (0/0)

Figure 4.2.2 Master Data Service Test

CHAPTER 5
CONCLUSION

The project's goals included providing App-creation APIs, ways to compose apps using these APIs, simple integration through the use of REST APIs, workflow management, and a functioning app. This product aims to provide APIs for users to compose domain-agnostic applications. A multi-tenant system. It adheres to a software engineering methodology that emphasises the separation of an application into single-function modules with well-defined interfaces. Initial efforts have been made to determine the necessity of the system. To meet the needs of the users, a detailed study has been designed to be user-friendly and straightforward. This particular system has been attractively designed so that even a user with minimal knowledge could operate it without difficulty. This is an extremely helpful system This product is based on microservices architecture, with each microservice handling domain-specific problems. Users will create applications by combining these various microservices. Creating reusable services/APIs enables a quicker time to market for new products, which is being driven by an increasing number of organisations today. This microservices-based project illustrates how multiple services can operate independently by leveraging the most effective microservices patterns for scalability, performance, and resilience.

5.1 Future Enhancement

The system is designed in a way that makes the addition of new modules relatively simple. The reconstruction of the system will increase the system's adaptability. The system has been developed to be as flexible and user-friendly as possible, noting the advanced features of this technology.

Easy integration via usage of REST APIs, workflow management and provide a working App. This helps users in easy way to notify based on certain events. Now a part of backend section is completed for the above project. This application provides various microservices, the users will be creating apps by stitching those various microservices. Also, UI Development (Reactive native for mobile, ReactJS for Web), Cloud hosting (AWS), Containerization (Docker & Kubernetes) will be completed in future.

CHAPTER 6

REFERENCES

[1]. Grzegorz Blinowsk, Anna Ojdowska, and Adam Przybyłek, “Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation”, Institute of Computer Science, Warsaw University of Technology IEEE 2022, 00-665 Warsaw, Poland

[2].Kavya Guntupaly ,”Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example”, 2021 IEEE International Conference on Big Data (Big Data)

[3]. Sahil Thakare, Ajay Kamble, "Document Segmentation and Language Translation in Tesseract" -OCR, 13th IEEE International Conference on Industrial and Information Systems (ICIIS)

[4]. Sahil Thakare, Ajay Kamble, “Document Segmentation and Language Translation Using Tesseract”-OCR, 2020 IEEE 13th International Conference on Industrial and Information Systems (ICIIS)

[5]. Seema Sultana, Sunanda Dixit; “Indexes in PostgreSQL; International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)”,2020

[6]. Andy Neumann, Nuno Laranjeiro, Jorge Bernardino; An Analysis of Public REST Web Service APIs; July 2021 IEEE Transactions on Services Computing

[7] Jishnu Saurav Mittapalli et Menaka Pushpa Arthu, "Survey on Pattern Engines in Java", School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, 2020

[8]. Lei, Kai, Yining Ma et Zhi Tan. "Comparaison et évaluation des performance des tools de développement Web en php, python et node. Js. Computational Science and Engineering (CSE), 2021 IEEE International Conference on. IEEE.

[9] .Casimir Désarmeaux, Andrea Pecatikov et Shane McIntosh "Distribution of Building Maintenance Through Maven Lifecycle Stages".2021 IEEE/ACM 13th Working Conference on Mining Software Repositories

[10] .Dheeraj, Kalpana Sharma, “API Security Testing Using Postman and Swagger Tools and Its Use in the Internet of Things (IOT)”, 2020

APPENDIX

Screenshot

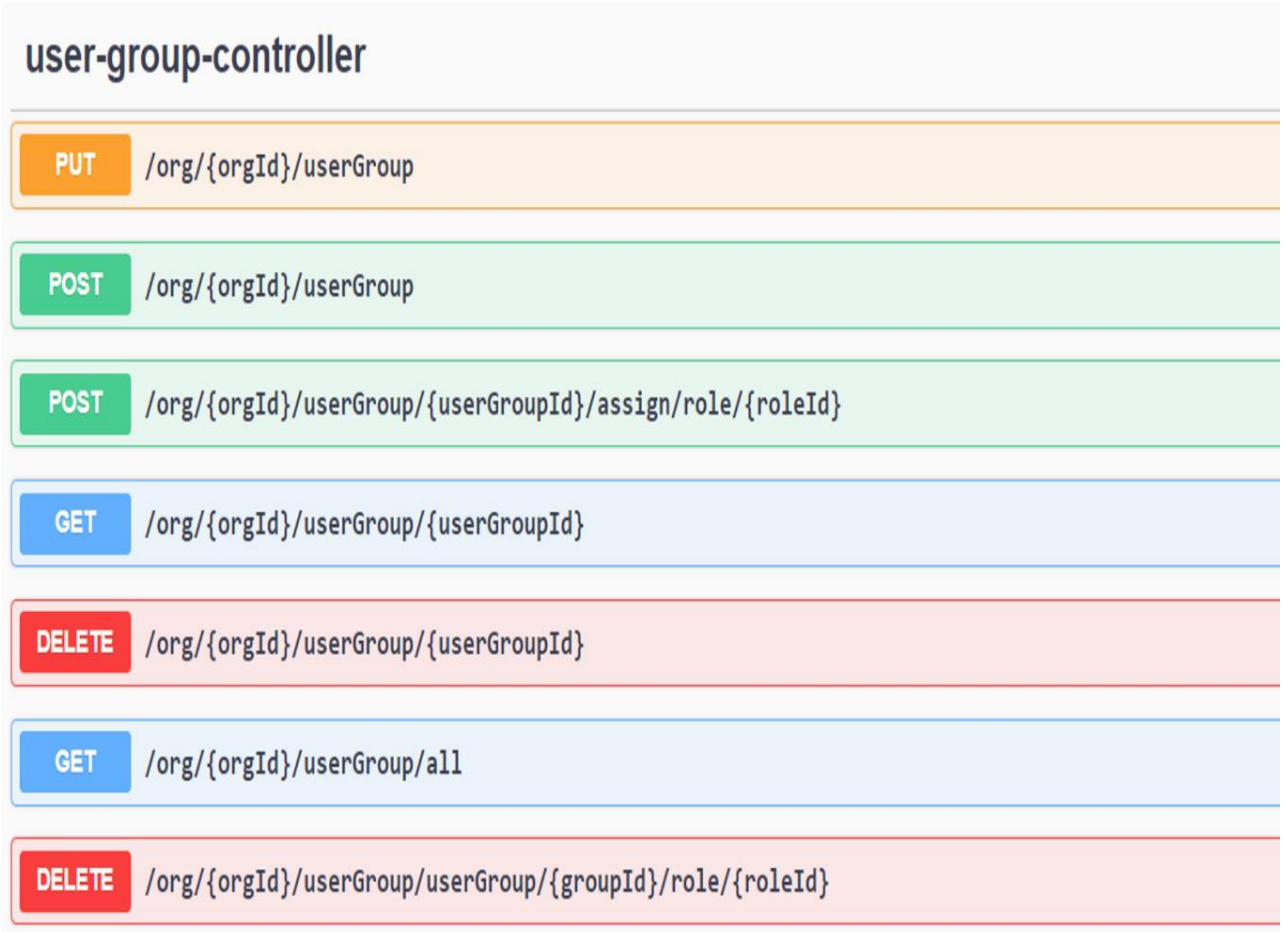


Figure A.1 UserGroup Controller

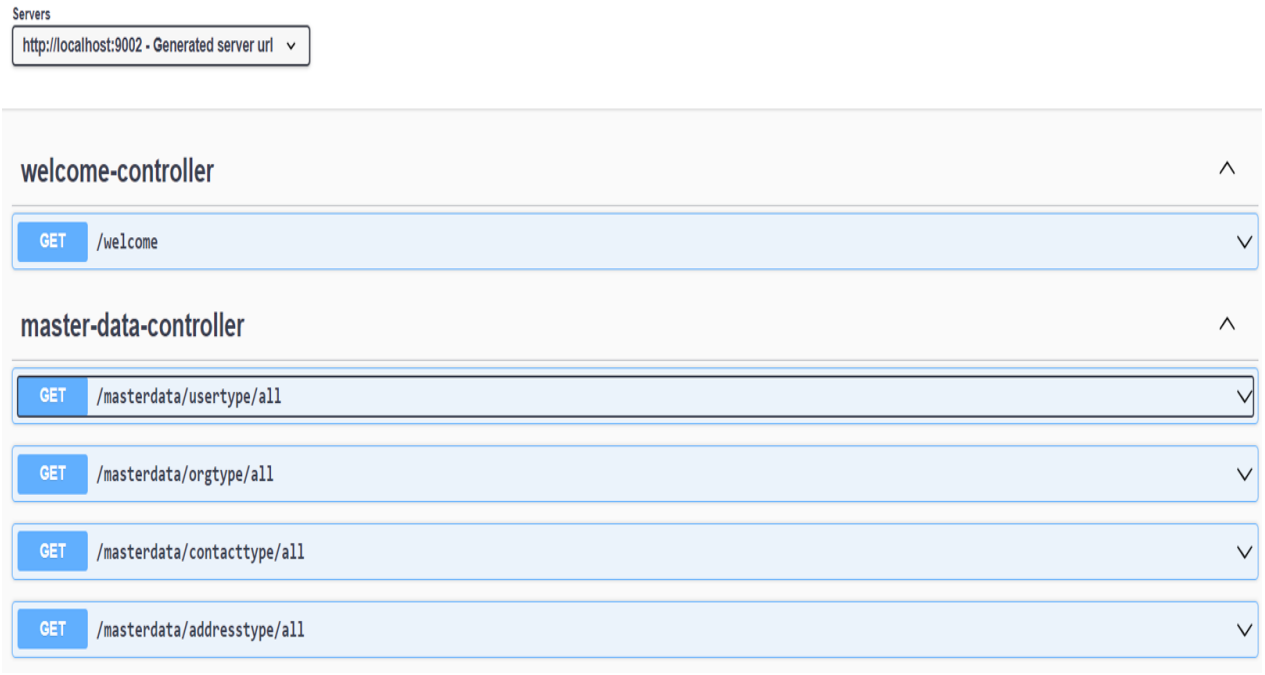


Figure A.2 Master Data Controller

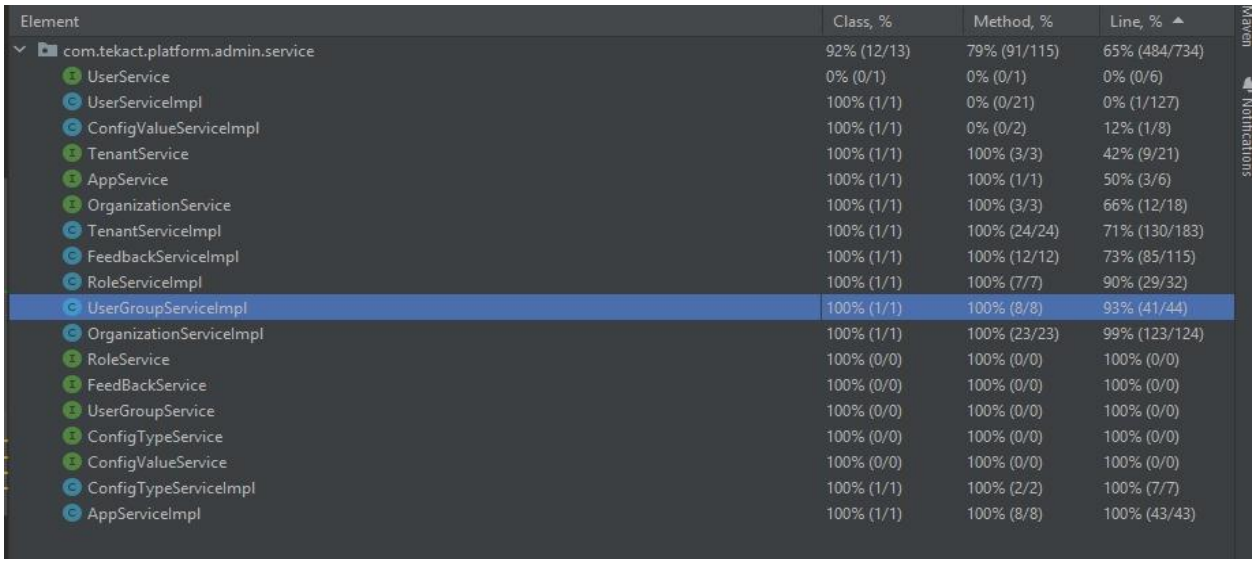


Figure A.3 UserGroup Service Test

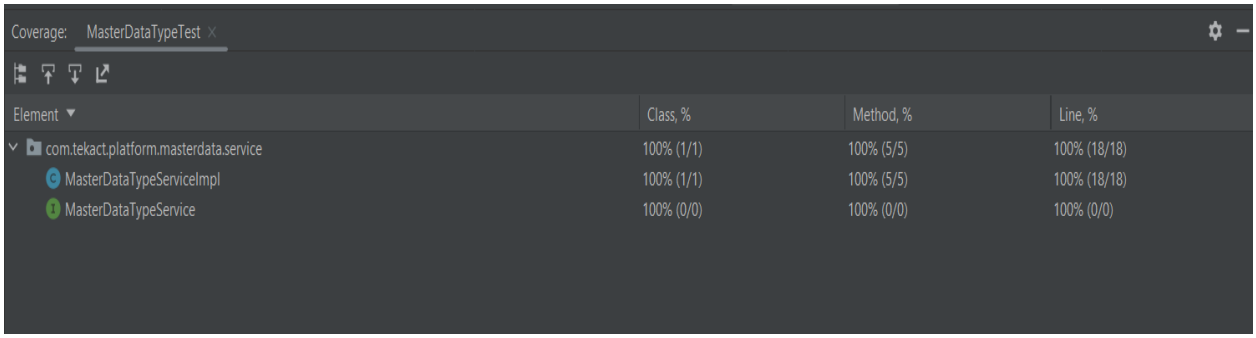


Figure A.4 Master Data Service Test