TRIGGER PUBLISHER

A PROJECT REPORT

Submitted by

ANANYA B

REG NO: TKM20MCA2008

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 **TRIGGER PUBLISHER**, 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 ANANYA B

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 "**TRIGGER PUBLISHER**" submitted by **ANANYA B (TKM20MCA2008),**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 her 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 Ms. Ananya B 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 she was punctual, hardworking, and inquisitive.

We wish her all the success in future endeavors.

Sincerely,

Place: Bengaluru

Date: 15-July-2022

SVK Pillai

Director

ACKNOWLEDGEMENT

A successful project is a fruitful culmination of efforts by many people, some directly involved and some others indirectly, by providing support and encouragement. First and foremost, I would like to thank the almighty for giving me the wisdom and grace for making my project a memorable one. I thank him for steering me to the shore of fulfilment under his protective wings.

I am extremely grateful to ***Prof. Dr. Fousia M Shamsudeen***, Assistant Professor and Head of the Department, MCA, TKMCE, for her constant support and encouragement throughout the project work. With a profound sense of gratitude, I would like to express my heartfelt thanks to my guide ***Prof. Vaheetha Salam***, Department of Computer Applications, for her expert guidance, cooperation, and immense encouragement.

I would like to thank my external coordinator ***Mr. Rajesh S R***, Tekact Private Limited, who guided me throughout my work. I also extend my thanks to the entire faculty and staff of the Department of Computer Applications, TKMCE, who have encouraged me throughout my course of study.

I also express my thanks to my family and friends, for their support and encouragement in the successful completion of this project work.

ANANYA

ABSTRACT

When a service in an event-driven architecture completes a task that other services might find interesting, that it creates an event, which serves as a record of the action that was completed. Other services use such events as a resource to carry out any duties that become necessary as a result of the event. Services that make requests, unlike REST, do not have to be aware of the specifics of the services consuming the requests. Here's an example: A single "order placed" event is generated and then consumed by numerous microservices when an order is placed on an e-commerce website. Various methods can be used to publish events. For instance, they can be broadcast to a queue that ensures delivery of the event to the right consumers or to a stream using the "pub/sub" architecture, which publishes the event and makes it accessible to everyone who is interested. In either scenario, the event is published by the producer, which the consumer then gets and responds to. These two parties might alternatively be referred to as the publisher and the subscriber in various situations. This project suggests a multi-tenant solution that publishes target-based events using a micro-service architecture. Additionally, it will offer APIs so that users may create apps that can function in other domains.

Contents

List of Figures

CHAPTER 1

INTRODUCTION

Trigger Publisher is a system developed for Tekact Private Limited to publish targeted events. Events are one of the most overlooked features of the framework, but also one of the most useful. And like many other things in Spring, event publishing is one of the features provided by Application Context. The publisher constructs the event object and publishes it for anyone to listen to. It is a product based on microservices architecture where each microservice handles specific problems for a specific domain. Users will create applications by assembling these various small services. This product is intended to provide APIs that allow users to compose applications that can work with different domains. Publisher enabled is a multi-tenant system. It follows a software engineering approach that focuses on breaking down an application into single functional modules with well-defined interfaces.

The sender component of a Spring application that uses the event-based communication model simply publishes an event without knowing who the recipient will be. In fact, there may be more than one active receiver component here. In addition, the recipient does not need to know who is publishing the event. Listeners have the opportunity to simultaneously attend several events from different shippers. This way the transmitter and receiver components are only loosely coupled to each other. Microservices architecture, also known as MSA, is a method of developing a single application by breaking it down into a set of smaller services that are only loosely coupled together. Each service is designed and built around a specific function or process, and it communicates with other services using simple mechanisms (usually HTTP). Building reusable services and application programming interfaces (APIs) helps bring new products to market faster, which is encouraged by many organizations in today's world. This microservices-based project demonstrates how multiple services can work independently using best practices for microservices to enable scalability, performance, and resiliency.

1.1 Objective

The project aims to achieve the following for the Trigger Publisher product:

1. Provide APIs to build apps

Some of the benefits of using Spring Boot for your REST APIs include: No complex XML configuration required. There is an embedded Tomcat-server to run Spring Boot applications. Spring Boot autoconfiguration automatically configures your application for certain dependencies.

2. Provide facilities for composing applications using such APIs

The REST API is only a web application, much like all the others we have ever made. The only distinction is that it provides data in response to a GET request rather than producing a web page. Additionally, it will ask other programmes to POST instead of using HTML forms.

3. Easy integration through the use of REST APIs

An example of an API is the Processing reference, which contains the classes and procedures that were utilised to create the Processing code. An analogous list of classes and functions is the Java API, which we utilise to create Java programmes. The idea is that an API is a list of things we can perform in code. Therefore, when we say we are building a REST API, what we really mean is that we are employing REST concepts to build a tool that programmers can use to interact with data.

4. Workflow management

Workflow management refers to defining, organizing, and coordinating a specific set of tasks to produce a specific result. Workflow management is about optimizing, improving, and

automating workflows wherever possible to increase throughput, eliminate duplication, and reduce errors.

5. Provide an application that works

This project follows the Scrum model to deliver functional applications. Scrum is an Agile project management method that involves a small team led by a Scrum expert whose main job is to remove all obstacles to getting the job done. Work is done in short cycles called sprints, and the group meets daily to discuss ongoing tasks and obstacles that need to be cleared.

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.

1.3 Services

Software Development

Aimed at product management, engineering, operations, and evolving various software types.

Quality Assurance

With us be assured that your applications and services are running accurately, functionally and

non-functionally

Consulting

Be a front runner in your business domain, as we help you out race the technology and customer trends

Build Operate Transfer

We are your trusted partner for setting up offshore development center. Share your risks with us and reduce costs of setting up a development center.

CHAPTER 2

LITERATURE SURVEY

A literature review is a thorough examination and analysis of all published works on a certain subject. When using a literature review, it is possible to identify linked research questions. After that, it is possible to look for and analyse pertinent material in an effort to find the solution to the research question. The ability to re-examine study results can lead to the development of fresh insights, which is one benefit of literature reviews. A literature review summarises and discusses all of the most recent data on a given topic that can be found in academic books and journal articles. At a university, there are two different kinds of literature reviews that one may write: one that students must write as part of a course's stand-alone assignments, and the other that is prepared as part of the introduction to or preparation for a larger work, usually a thesis or research report. The focus, point of view, and style of hypothesis or thesis argumentation will all vary depending on the sort of review that was generated. By examining the organisation of their arguments and observing how they approach the topics, reading published literature reviews or the introductory chapters of theses and dissertations in our own field may help us better understand the distinctions between these two types.

2.2 Goal of the Literature Review

1. It makes it simple for readers to conduct research on a topic by choosing excellent articles or studies that are pertinent, significant, important, and valid, and compiling them into a single comprehensive report.

2. By requiring them to summarize, assess, and compare original research in that particular field,

it gives researchers starting out in a new field a great place to start.

3. It makes sure that previous research is not repeated by researchers.

4. It can suggest areas to focus on or give hints as to where future research is going.

5. It also emphasizes the main conclusions.

6. It points out gaps, contradictions, and inconsistencies in the literature.

7. It also offers a constructive critique of the methods and strategies used by 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

In a microservice architecture, a business domain is separated into small, routinely limited contexts that are implemented by independent, self-contained, loosely connected, and independently deployable services. When the phrase "microservice" did not even exist in 2009, Netflix was one of the forerunners of the technology, starting to move away from its monolithic architecture. Applications built using microservices grow horizontally well, both from a technological perspective and in terms of the organizational structure of development teams, which may be maintained more compact and nimble. There have already been attempts to smoothly include such possibilities into adaptive business process management. Additionally, dividing huge apps into separate microservices gives agile teams the next level of independence, which helps expanding agile methodologies. Each group could design various microservices. [1].

2.2.2. Spring Boot

Spring Boot is a popular java-based framework for creating online and business apps, and it gives SOA flexibility (SOA). Because of the intricacy of the settings, any Spring-based application may meet problems. Spring Boot simplifies the creation and deployment of independent, professional-grade Spring programmes with little Spring setup. When compared to CRUD web projects, Spring Boot offers simplified dependency management by utilising a comprehensive, but flexible framework and the necessary libraries in a single dependency. You may now access all of the Spring-related technologies you need to get started on projects. This framework includes several features that are common to many projects, such as embedded servers, security, metrics, health checks, and externalised settings. War files are commonly used to bundle web applications. [2]

2.2.3. OCR

Lines of text, words, and symbols in a document must be divided into the appropriate pieces for optical character recognition before they can be recognised (OCR). There is a wealth of textual material available online in both Hindi and many other languages, including English, the world's most frequently spoken language. The majority of the market, however, is dominated by English-specific tools and processes. This means that an effective text extraction approach for languages other than English is required. To get over this limitation, this article proposes a software interface, or web application, that may automatically translate an image document into any language supported by the Googletrans library. Python-tesseract is an optical character recognition (OCR) engine for Python apps.That is, it will detect and interpret text hidden deeply inside a section of the picture. Explanatory ideas may be used to build web apps that accept document image files as input and produce translations, while Google trans is a free and limitless Python package that implements the Google Translate API. The digitised input image is subjected to a number of procedures during the preprocessing phase. The image is improved by segmenting it after noise reduction and producing and rendering it to divide things into smaller segments [3].

2.2.4. Tesseract

Tesseract is intended to be a language-neutral system. The Tesseract's first objective was to identify white against a dark background. This results in the design in terms of the investigation of the outlines of the parts and analysis of the associated parts. To extract images from a text, Tesseract-OCR is employed. Tesseract is a set of extremely efficient algorithms. The Python "Python-tesseract" library implements Tesseract-OCR to convert pictures to text. Its implementation is straightforward since all we need to do to convert the supplied picture to a string is include the module and call the stated function image to string(image, lang = 'lang code'). The language in which the text of the document is written must be made clear. Tesseract converts existing text to a picture and returns it. Tesseract-OCR is an extremely efficient and maintained library with an efficient implementation. The image to string (image file, language) library function accepts the following file types: ".jpg," ".png," ".gif," ".bmp," ".tiff," and ".bmp" [4].

2.2.5. PostgreSQL

PostgreSQL is a powerful open source antisocial database framework. It has been actively developed for over 15 years, has a tested design, and has a proven track record of reliability, information accuracy, and correctness. PostgreSQL, sometimes known as post-gre-SQL, is an open source social database management system built by a global team of volunteers (DBMS). PostgreSQL's source code is freely accessible and not controlled by any other business or government organisation. PostgreSQL, formerly known as Postgres, was created at UC Berkeley by Michael Stonebraker, a professor of software engineering. Stonebraker released Postgres in 1986 as a continuation of Ingres, which Computer Associates later purchased. All of the major desktop operating systems, including Windows, Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, and Tru6), and BSD, continue to support PostgreSQL. It enhances the text, graphics, audio, and video quality and includes C/C, Java, Perl, Python, Ruby, Tcl, and Open Database Connectivity programming interfaces (ODBC). Lists are good for providing faster access to information. Adding records to a section typically makes finding information simpler. The trade-off is that information will be integrated more slowly for each record. Essentially, if you embed your information in a list, the list should retain the structure of the information in the file as you embed it and organise it in two places. [5].

2.2.6. RESTful API

The suggested system has attempted to construct 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. Several variables, including as filtering, sorting, selection, and pagination, among others, are employed to optimise the URI. The backend of the Takeaway app is a website, while the front end is an Android-based platform. According to test findings based on the function method using the Postman application, the REST API server established on the server was up and running without a hitch. The Apache JMeter tool may be used to evaluate the response time of the application, and the results show that it has a decent response time. While this is going on, a comparison of the requests and replies of the SOAP and REST architectures demonstrates that REST takes longer [6].

2.2.7. Thymeleaf

Thymeleaf is essentially a Java template engine that may be used in contexts and applications that are not web or servlet-based. It works well with offline apps but is best suited to work with HTML5 for web applications. It effectively replaces JSP because its template files may be accessed directly from the browser. The key characteristics are as follows: - Appropriate for extensive documentation. If necessary, it can be used as a model engine frame. It can identify the kind of document and transform it as needed. With a feature called Parsing Template Cache, which offers excellent performance and efficiency, input output can be minimised. The key benefit of this over other tools is that the templates act like prototypes, enabling the development of a natural templating system. Models that are statically rendered open in the browser without the need for any servers to be operating. It resembles a model engine frame in essence [7].

2.2.8. Swagger

To describe web APIs, the OpenAPI or Swagger specification has taken over (REST, RESTful, etc.). This is the reason that numerous research articles have focused on this specification and continue to do so. With the help of this specification, numerous processes, procedures, and issues linked to software development can be streamlined, resolved, or automated. This study aims to

provide an overview of how the OpenAPI/Swagger specification is used and applied. They include the domains in question, the development activities for which it is valuable, and the software artefacts (results) produced from them. A total of 7 articles that were published between 2011 and 2020 were included in the literature review that we looked at. These searches: a) point to development tasks where the specification is helpful. advantages, b) contains the software artefacts (results) it produces and the assertions associated with them, and c) makes references to the relevant domains. The findings indicate that various software development activities have benefited from the OpenAPI/Swagger specification, with better documentation being the most significant advantage (3 percent). In a similar vein, 77% of articles offer answers to web API users' problems or demands. In 68 percent of the manuscripts, work automation based on various tools led to the most notable outcomes. Additionally, none of the texts consulted specifically address any topic (IoT, Cloud, etc.) [8].

2.2.9. Maven

Maven might be a potent project management tool with POM support (project object model). Projects, dependencies, and documentation are built using it. Maven makes it easier for Java engineers to complete their everyday tasks and frequently aids in organising any Java-based project. A JAR file directory with some information is what makes up the Maven repository. POM files called metadata are linked to the projects to which each bundled 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 workstation. Maven builds projects in a succession of six build phases, each of which is dependent upon 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 stated. The project's source code should be compiled into raw distribution files (e.g. .class files). To check for faults, 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. Push local configurations into the production environment [9].

2.2.10. Postman

A test in Postman is just a piece of JavaScript code that runs when a request is submitted and a response from the server is returned. POSTMAN is really easy to use. To test the APIs of the programme, one must adhere to the set of API calls that it provides. You may use Postman to create and execute tests in JavaScript for each request. first. Run the Postman application. For the operating systems macOS, Windows, and Linux, it is a native application. Additionally, there is a Chrome app for it. If you're using Chrome to reach the Applications area, click the square there. GET (Get Data), POST (Update Data in Existing File), PUT (Replace Existing File), and DELETE (Delete Existing File) are the most often utilised strategies (Delete data from the server). Request for a message: a. In the title toolbar, click the New button. b. Type imageURL-postman-echo.com/get as the request URL (URL - rest endpoint) shown below. The user can store the request for later use by choosing the Save option positioned next to the Submit button. Utilize Newman to integrate with construction management tools. With its command line partner, Postman, you can automate testing by integrating Postman with a build management platform. Authorisation: The authorization procedure determines if you have permission to access the server's desired data. Testing for security includes doing this. The tool offers a variety of rights, which we shall discuss in more detail in the following section. [10].

CHAPTER 3
METHODOLOGY

This Trigger Publisher project builds the event object and publishes it for anyone to listen to. It is a product based on microservices architecture where each microservice handles specific problems for a specific domain. Users will create applications by assembling these various small services. This product is intended to provide APIs that allow users to compose applications that can work with different domains. Publisher enabled is a multi-tenant system. It follows a software engineering approach that focuses on breaking down an application into single functional modules with well-defined interfaces.

In "Trigger Publisher", CRUD (create, read, update and delete) and list operations are done using Rest API and Springboot framework. When executed, it passes through the controller with the path variables and query parameters specified in the URL. It then calls the methods declared in the service layer and will be connected to the controller automatically. The interface then calls the class that implements the service and executes the code using the parameters specified in the Rest API calls. The implementation calls the respective models and entities in turn. This means that when a customer sends a request by postal:

- 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 Base Repository or JPA Repository which include some basic methods like findById(), findAll() etc.
- The Entity class extends Base Entity which includes some common entities.
- Rendering micro-services.

3.1 Architecture

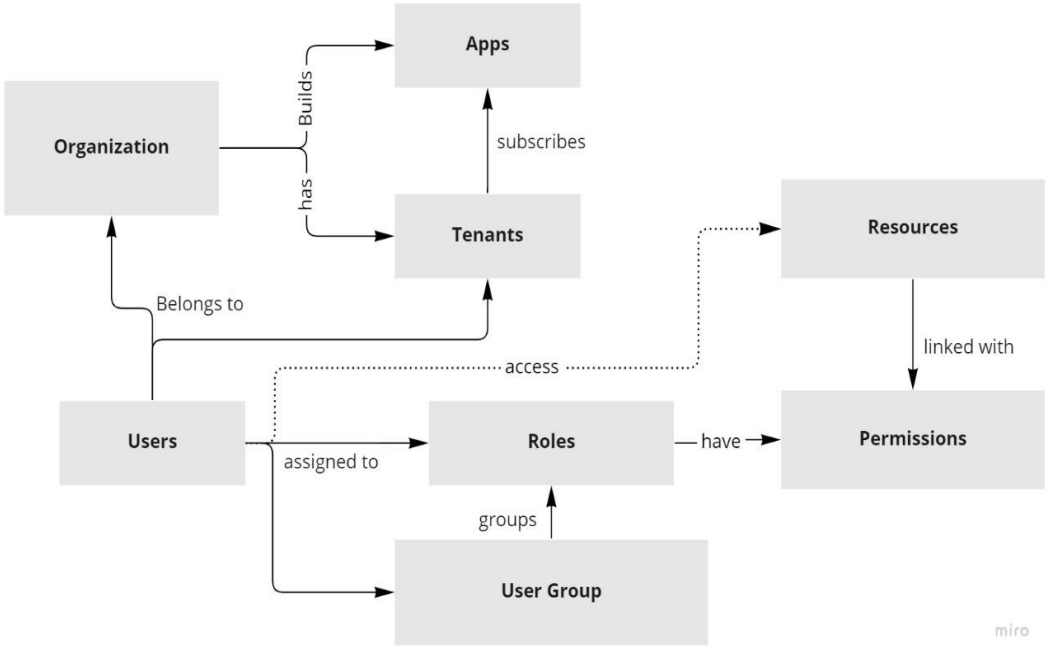


Figure 3.2: Architectural Design

All other entities are children of the organisation, which is the parent entity. Both the company that develops the programme and the tenants that subscribe to it are part of the organisation. The renter and the organisation both own the user. User roles and groups are given to users. Users have access to the resources linked to their permissions. The link between the primary structural

components of the programme is defined by the software architecture of a computer system, which is the structure of the system made up of software components, their externally observable attributes, and their relationship to one another. Analytical models and the interplay of its subsystems can be used to deduce the modular structure of a computer programme. The primary objective is to create a modular programme structure and illustrate the connections between modules. Architectural design is the process of establishing the subsystems that make up a system and the framework for regulating and interacting with those subsystems. A description of the software architecture is the end result of this design process.

3.2 Module Description

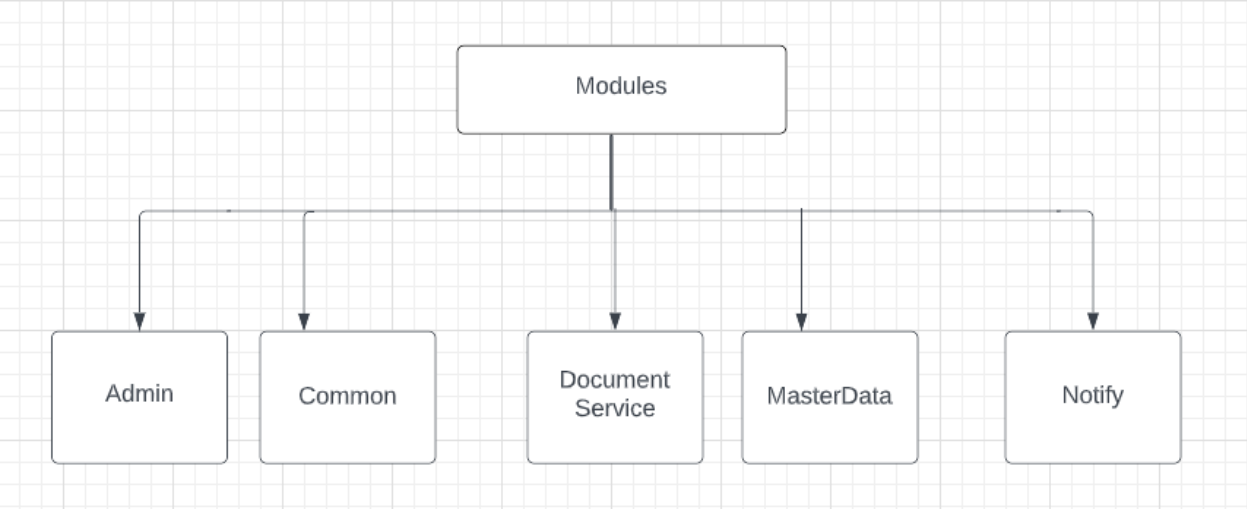


Figure 3.2 Modules

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 operations of Business Process Server, installation and uninstallation of 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

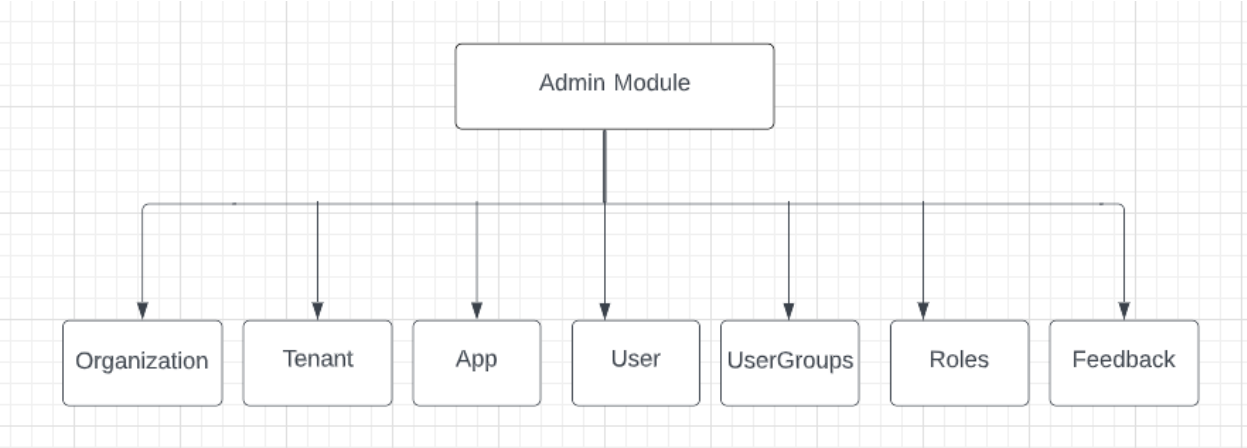


Figure 3.2.1 Admin Module

3.2.1.a Organization

The organization is the parent entity of all entities in this project. For example, Tekact as an organization provides various APIs that can be used by tenants to build applications. All other entities like tenants, users, roles, applications depend on this parent entity. CRUD and Listing API for Organizations: As an administrator, I can create, read, update, and delete (soft delete) an organization. The API must be available to retrieve all entities in the database. Delegating as administrator first represents creating the entity, then getting the entity from the database, updating the entity. Softly delete the entity in the database and list all active organizations. Deleted organizations will not be listed. The APIs mentioned above have been created. Then the code coverage is at least 80% and no static violation is achieved in this part.

CRUD - Organization Address and Contact: As a senior administrator, I can create, read, update, and delete (soft delete) an organization's addresses/contacts. The API must be

available to retrieve all the addresses/contacts in the database. Permissions as super admin, authentication including app name must be unique and cannot be null. Acceptance criteria include representing the creation of entities (address/contact). And get the entity from the database. Then update the entity, then list all the addresses/contacts of the organization. Then the code coverage is at least 80% and no static violation occurs.

CRUD and Lists API for apps, linking apps to organizations: As an organization administrator, I can create, read, update, and delete (soft delete) apps. The API must be available to retrieve all applications from the database. I can also assign apps to organizations. Permissions as an organization administrator, authentication including application names must be unique and cannot be null. Acceptance criteria include demonstrating the creation of the entity and then retrieving the entity from the database. Update objects and soft delete objects in the database. List all active tenants. Additionally, tenants that have been deleted will not be listed. Also assign the application to the organization, when this coverage is at least 80% and no static violations are reached.

3.2.1.b Tenant

As an Organizational Admin, will have provision to create, read, update and delete (soft delete) a tenant. API should be available to retrieve all the tenants in the database. Tenant can be a consumer that needs various APIs from the organization. Tenant can be a single user or a complete organization. One organization can have multiple tenants. Tenant subscribes various apps build by the organization.

3.2.1.c Apps

Apps are the products that are created by composing various APIs. As an Organizational Admin, should have provision to create, read, update and delete (soft delete) apps. API should be available to retrieve all the apps in the database.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, should have provision 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. One user can be assigned to multiple roles. 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, should have provision to create, read, update and delete (soft delete) UserGroups 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

3.2.1.f Roles

Every user has at least one role. Role management and assignment is controlled by administrator. Roles are assigned to users and user groups, as part of admin I should have provision 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.

3.2.1.g Feedback

The feedback management module can provide users with a way to exchange feedback with the client about their personal experiences using the platform as well as a way to rate the client's satisfaction. The administration may improve their product by using this input to better understand user experiences and how to make the programme better in the future. In essence, the Feedback module gives users a way to contact the system administrator with general (non-patient-specific) feedback. Instead of going via the same Admin, users will be able to send their issues or feedback directly to the most appropriate person. The comments may be favourable or unfavourable. Either the feedbackId or the organizationId can be used to access all of the feedback from the database. Positive or negative feedback is helpful to an organisation in its efforts to grow.

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. Data that is created centrally, and is valid for all applications. This module use 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

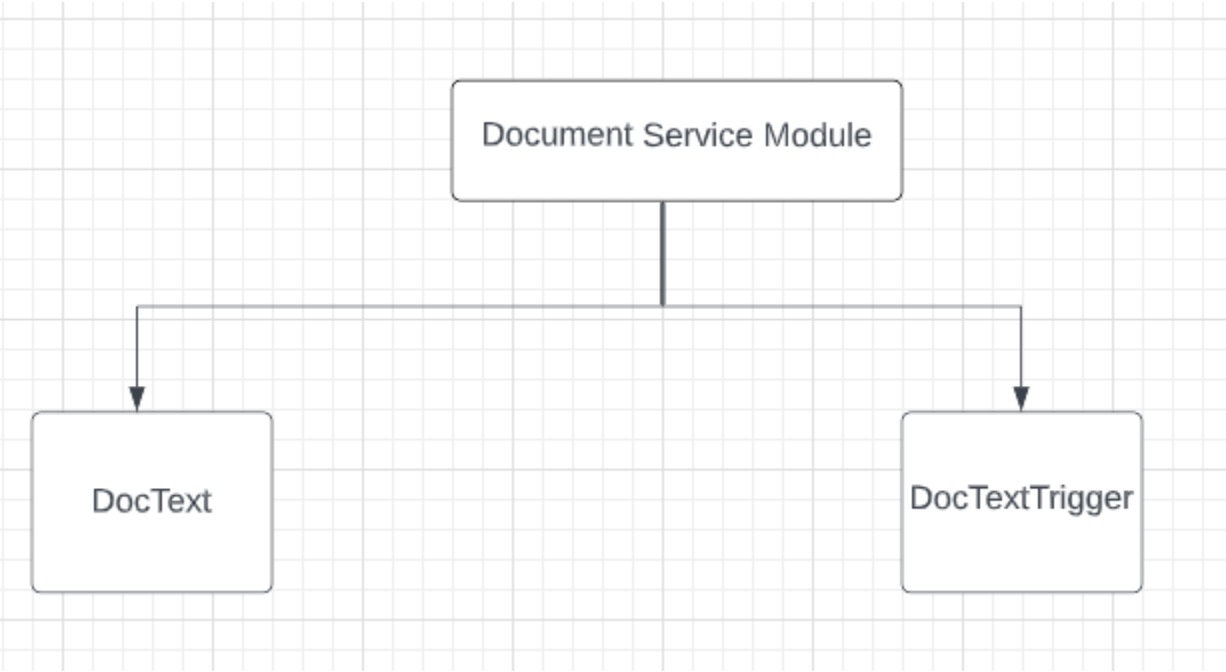


Figure 3.2.3 Document Service

To create/upload, update, upload, delete, list, and retrieve documents, use the Document Service's API. Create metadata from uploaded photos using OCR. Preprocess the image data, e.g., convert to grayscale, smooth, straighten, or filter, in order to retrieve useful information from a given image file. Lines, words, and characters are detected. Create a ranking of potential character candidates using the training dataset. (In this case, the training data path is set using the setDataPath() method.) After the characters have been recognised, choose the best ones based on linguistic information and the confidence from the previous stage. Language-related information, such as dictionaries and grammar rules. This makes office work more productive and efficient. Instant content search is quite helpful, especially in work settings that manage large amounts of scanning or dense document volumes. Fast OCR guarantees that the content of the document is preserved while conserving time. Employee productivity increases as a result of not having to perform manual labour and their ability to work more quickly and effectively. The system will

process the papers it receives as input in this module and produce useful data. This information is used by the system to produce events. identifies and discusses storing the data required to produce events. For additional processing, uploaded files (containing organisation, tenant, application, and user details) are used. An implementation of the DocTextController CRUD API for document text is one of the requirements for acceptance. Implement the listTextByIsApproved, delete, and approve APIs next. Use CRUD APIs to implement text trigger documentation. Also use the listTriggerByIsApproved approved and deprecated APIs for the document text trigger.

3.2.4 MasterData

The parent module is one of the child elements of the parent module administrator. We are free to create any kind of dependencies between modules and packages. The main module manages entities like country, city, state, contact type entity, organization type entity, address type entity. The entities in this module use the Common Module Base Entity, the Generic Repository Base Entity, and some of the Common Module features and methods.

3.3 RESTful Webservice

REST is a Web-based design that makes use of the HTTP protocol. Each component is a resource, and each resource is accessed through a common interface using conventional HTTP methods. It is based on resources.

3.3.1 Working

It helps to break the procedure up into two phases to better understand how RESTful APIs convey data between clients and applications.

3.3.2 Client Request

A programme or human using API services is referred to as a client. For instance, if you embed a web application like Instagram or YouTube, your programme will be a client. You will be the

customer if you request the URL through your browser.An HTTP call to the Instagram API can, for instance, be a POST request to submit a new photo or a GET request to get a particular video or post. Similar to this, a call centre platform that has an auto-answer application can utilise the PUT command to update or delete a customised greeting.

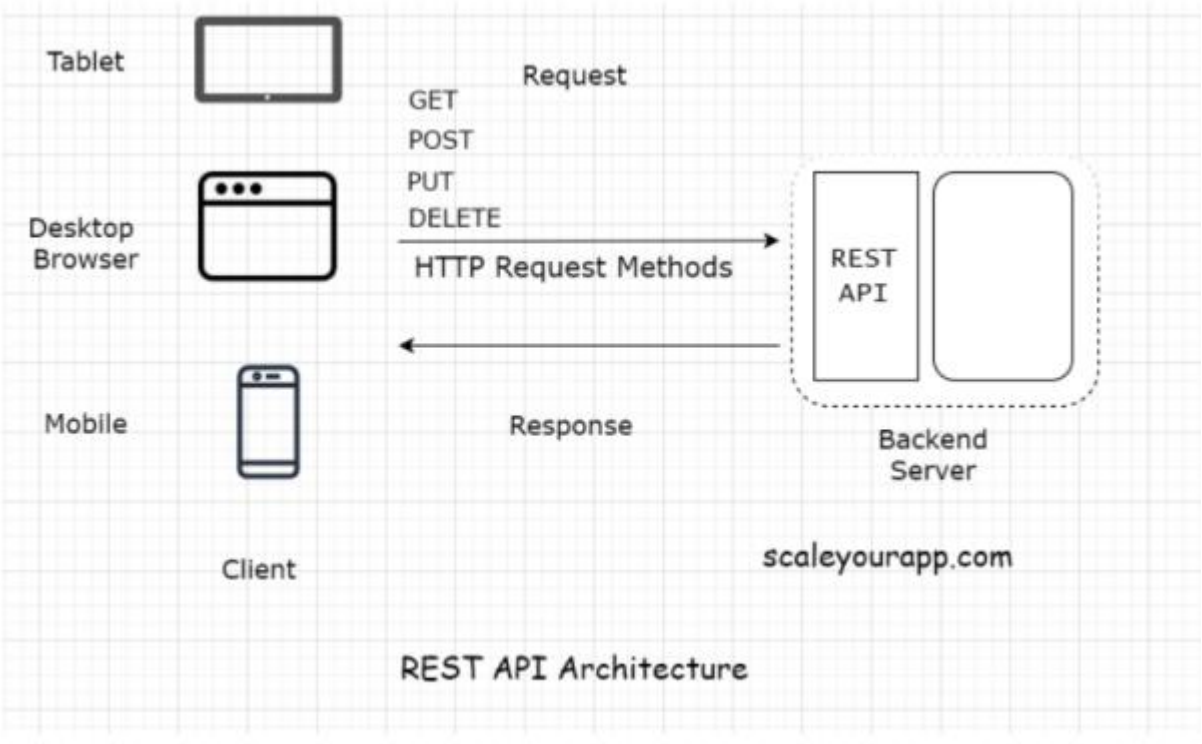


Figure 3.3.2: REST API Architecture

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 housed 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 normalised representation of the resource's state to the client's computer. This indicates that instead of providing the client the real resource, the server is delivering a representation of the resource's state, or the status of the resource at a specific

timestamp. Responses are provided in a concise way 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 different programming languages. Alternative API data formats include plain text, XML, YAML, CSV, and HTML.

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 Software Description

JAVA

Software written in Java, an object-oriented programming language, can run on various platforms. The compiled code, or "bytecode," created when a programmer creates a Java application runs on the majority of operating systems (OS), including Windows, Linux, and Mac

OS. Java's syntax is heavily influenced by the C and C programming languages. James A. Gosling, a former computer scientist with Sun Microsystems, along with Mike Sheridan and Patrick Naughton created Java in the middle of the 1990s. Java develops applets, which are browser-based programmes that allow graphical user interfaces (GUIs) and object interaction. Web pages were often static and non-interactive before Java programmes. With the introduction of rival products like Adobe Flash and Microsoft Silverlight, Java applications experienced a decline in popularity. The Java Virtual Machine (JVM), which converts Java bytecode into native processor instructions and permits indirect programme execution from the operating system or platform, allows Java programmes to run on a web browser. The majority of the parts required to run bytecode, which is typically smaller than executable programmes created in other programming languages, are provided by the Java Virtual Machine (JVM). If the required JVM is not present on the system, Bytecode cannot run.

A Java Software Development Kit (SDK), which normally consists of a compiler, interpreter, documentation generator, and other tools used to construct an application, is necessary for creating Java programmes. complete.

INTELLIJ IDEA

For working with Java and other JVM languages like Kotlin, Scala, and Groovy across all types of applications, use IntelliJ IDEA, an intelligent, context-aware IDE. Additionally, IntelliJ IDEA Ultimate's robust built-in tools, support for JavaScript and associated technologies, and improved support for well-known frameworks like Spring, Spring Boot, Jakarta EE, Micronaut, Quarkus, and Helidon can assist you in creating full web applications. You may also expand IntelliJ IDEA with free JetBrains plugins, enabling you to work with additional programming languages like Go, Python, SQL, Ruby, and PHP.

SPRING BOOT

A well-known open source enterprise framework for producing high-quality standalone applications that run on the Java Virtual Machine is the Java Spring Framework (Spring Framework) (JVM). Using three key characteristics, Java Spring Boot (Spring Boot) speeds up

and simplifies the process of creating web apps and microservices with the Spring Framework. With the help of these features, you may create a Spring-based application with less setup and configuration. Dependency injection is a feature of the Spring Framework that enables objects to specify their own dependencies rather than waiting for the Spring container to inject them. As a result, programmers can construct modular programmes with loosely linked components that are perfect for distributed network applications and microservices. Additionally, the Spring Framework comes with built-in support for many of the common activities that an application must complete, including data binding, type conversion, validation, exception handling, resource and event management, internationalisation, and more.

Applications that have autoconfiguration are started with pre-configured dependencies so you don't have to explicitly setup them. Due to built-in autoconfiguration features, Java Spring Boot configures the underlying Spring Framework and third-party packages according to your settings (and based on best practices, which helps avoid errors). Despite the fact that you can change these defaults after initialization is finished, Java Spring Boot's autoconfiguration feature makes it easier for you to get started creating Spring-based apps quickly and lowers the likelihood of human error. Opinionated method: Depending on the requirements of your project, Spring Boot takes an opinionated approach to adding and setting starting dependencies. Instead of having you to make all those decisions and configure everything manually, Spring Boot decides which packages to install and which default values to use. During the startup process, when you select from a variety of starter dependencies known as Spring Starters that address common use cases, you can specify the requirements of your project. You can use Spring Boot Initializr without knowing any code by completing a short web form. Standalone Applications: Spring Boot enables programmers to build operating systems. By integrating a web server like Tomcat or Netty into your application during setup, it specifically enables you to construct standalone apps that function independently without relying on an external web server. Therefore, by simply selecting the Run option, you may start your programme on any platform. (This can be turned off to allow you to develop programmes without an embedded web server.)

REST API

An API implementation that adheres to REST architectural principles is referred to as a REST API. It serves as a conduit. Through HTTP, the client and server communicate with one another. The HTTP methods are used by the REST API to set up client-server communication. In order to enhance the efficiency of applications, REST also permits servers to cache answers. 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 earlier communications are shared. Therefore, the client must submit credentials to the backend everytime it communicates with it. In doing so, the backend is able to ascertain whether the client has permission to view the data. The client interacts with the main endpoints when a REST API is implemented. The client code and backend code are entirely distinct in this way. Create, Read/Retrieve, Update, and Delete are the acronyms for CRUD. These are the four fundamental roles that persistent storage plays. User interface patterns that enable viewing, searching, and editing of data through forms and reports on a computer are known as CRUD operations. Data-oriented CRUD makes advantage of standardised HTTP action verbs. There are a few crucial verbs in HTTP.

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 applications can simply be connected with JUnit Framework ie;Maven, Eclipse, and Ant

The following crucial characteristics are offered by the JUnit testing framework. 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 before 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 execute a suite of tests in JUnit, use the @RunWith and @Suite annotations. The test classes TestJunit1 and TestJunit2 are used in the following illustration. Test Runners: Test cases are conducted using test runners. The test class

TestJunit is assumed to already exist in this example. a JUnit class - JUnit classes, which are used to create and test JUnits, are crucial classes. Several crucial courses include 'Assert' - Consists of a collection of assert methods. TestCase: Contains a test case that identifies the machine that is being used to execute 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 apps testable requires the use of mockito. Under licence 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, often known as the proxy used for the actual implementation, is referred to as such. The Mockito framework's main goal is to make the construction of tests simpler by simulating external dependencies and utilising them in the test code.

PostgresSQL

Both SQL (relational) and JSON (non-relational) queries are supported by PostgreSQL, a powerful, enterprise-grade, open source relational database. Because of its extremely robust database management system, it has excellent levels of resilience, integrity, and correctness. Many web, mobile, GIS, and analytics apps 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 apps are powered by PostgreSQL, a potent back-end database. database for general-purpose transactions. Both established businesses and start-ups rely on PostgreSQL as their back-end database to support their goods and apps. 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.

3.4.3 Hardware Specification

A computer with internet connectivity and with minimum 4GB Ram

3.4.4 Server Specification

Implementing an AWS t2.xlarge instance per module (with redundancy), a vCPU, and 16 GB of RAM. 5 GB of EBS-based storage 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 commonly 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 a business customer installing a real physical server farm, Amazon offers these products as services to supply mainframe capacity more quickly and affordably. AWS is used in this project as the server platform for cloud processing using software as a service (SAAS). All of the Apache service extensions utilised in our system are available on AWS, along with many more services.

3.5 System Design

The representation of data structures, software structures, interface elements, and procedural details are constructed from information needs in a multi-step process known as design. All following software engineering and maintenance procedures are built on top of design. It involves making decisions, frequently of a structural character. The focus of design is on the links between higher-level pieces and associated logical operations at lower levels, creating coherent, well-planned representations of programmes. A good design is one that generates efficient code and has the smallest implementation possible, depending on the applications and project needs.

Design elements, which often include functional hierarchy diagrams, screen layouts, business rule tables, business process diagrams, pseudocode, and flowcharts, explain the desired

programme functionality in detail. extensive data dictionary and complete entity relationship diagram. These design components are meant to sufficiently define the software so that skilled programmers may create it with little extra design input. The fundamental design ideas outline the following procedures for creating a project: - Abstract - Flexibility The architecture of software - Structured dividing Structures for data - Software techniques Two tiers are present. Level of system design: - Logical planning - Formal design In logical design, the designer creates a specification of the primary system attributes that achieve the objectives. The system's actual design is revealed via the physical layout.

3.5.1 Logical Design

Understanding the connections between various system components is aided by model analysis. The user can see exactly how a system will operate thanks to the analytical model. It is a system's first technical representation. Three primary objectives must be met by the analytical model. Lay the groundwork for designing software. to specify the user's requirements. Create a list of specifications that may be checked once the software has been developed. A use case diagram's goal is to depict a system's dynamic nature. However, this description is too generic to capture the objective. since the objectives of the other four diagrams—activity, sequence, collaboration, and state diagram—are similar.

Therefore, focusing on a goal in particular will set it apart from the other four diagrams. Use case diagrams are used to collect a system's needs, taking into account both internal and external factors. The majority of these requirements pertain to the design. Thus, use cases are created and actors are identified when a system is evaluated to gather its functionality. The use case diagrams are now modelled to depict the external perspective once the initial task is finished. In conclusion, a use case diagram's function can be: used to compile a system's needs. used to view a system from the outside. Determine the system's external and internal influencing forces. demonstrates how requirements interact with one another as actors.

3.6 Input Design

Input design is the process of converting the user defined inputs to a computer-based format. The design decisions handling input specify how data are accepted for computer processing. Input design is a part of overall system design that needs careful attention and it includes specifying the means by which actions are to be taken. The collection of input data is the most expensive part of the system in terms of the equipment used and the number of people involved. In input design, data is accepted for computer processing and input to the system is done through mapping via some map support or links. Inaccurate input data is most common cause of errors in data processing. The designing of input is concerned with the following points.

The initial steps input design is to determine what data to input. In data input 'Trigger Publisher' uses JSON format offers the same kind of benefits that XML used for exchanging the data in a heterogeneous environment, such as the following. The syntax and hierarchical structure of the JSON strings can be in some cases to be interpreted by applications that do not already know that what data to expect. JSON is of simple text. This fact makes it suitable and safe for transferring across platforms and operating systems that do not readily share more complex document types. As text, JSON can also be readily displayed and edited in simple editors. JSON is compact. An average JSON string is about two-third of the size of same data in XML. JSON is easy to learn, easy to read, and easy to understand also.

The objectives of the input design are:

1. Controlling amount

2. Avoiding delay.

3. Avoiding errors.

4. Avoiding extra steps

In this 'Trigger Publisher', all the text boxes are validated. If any mandatory field is not filled then it will display the message.

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

Output refers to the results and information produced by the system. Here defines the information to be presented, decides the layout and selection of output media, arranges the presentation of the information in an accepted format, and decides how to deliver the output to the intended recipient. The position and format characteristics of the column headers and pagination are specified. Output design plays an important role in providing the user with the required format. The main function of the output is to convey information and therefore its layout and design are carefully considered. The information must be carefully researched according to the needs of the user. Printed output standards suggest a name or title for each output, provide an example of output layout, and specify a procedure that provides output data accuracy.

The output devices to consider depend on the device's compatibility with the system, response time requirements, and required print quality. In the design output form, special attention is paid to proper identification and wording, readability and usability, composition and layout, order of data elements, and clarity of direction. guide. A well-designed form with clear captions should be self-informative. An organizational form must be centrally controlled for effective processing. The output of the computer is the most direct and important source of information for the user. Output design is a process that involves designing required outputs in the form of reports that should be delivered to users based on requirements. An efficient and easy-to-understand output design improves the system's relationship with users and aids decision-making. Since reports are directed by management to make decisions and draw conclusions, they must be carefully designed and the details of the report should be simple, descriptive, and clear to users. use. Therefore, when designing sockets, the following should be considered.

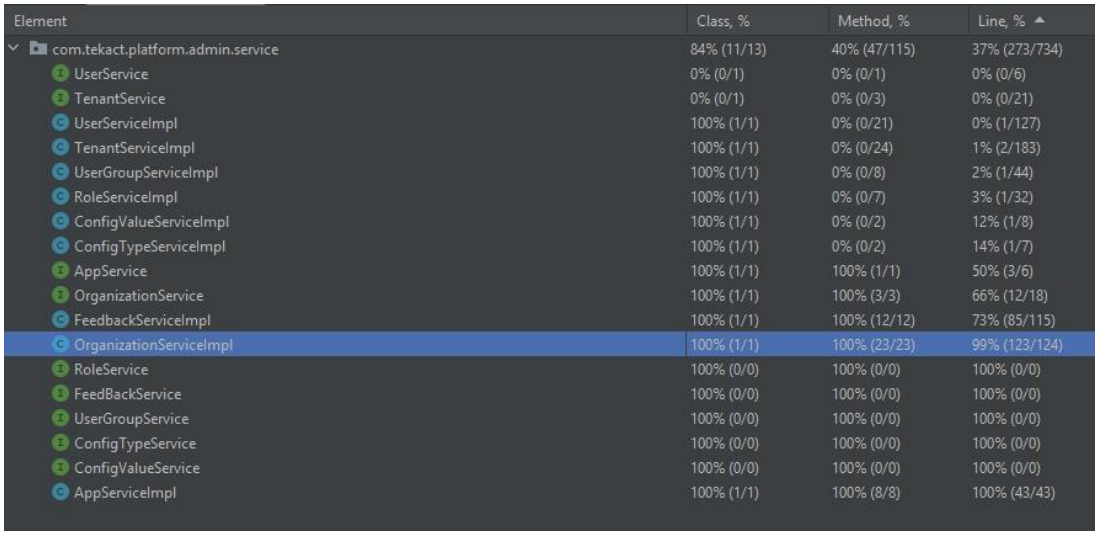
• Determine what information to present.

• Organize the presentation of information in an acceptable format.

• Decide how to deliver output to schedule reception

• Depending on the nature and future use of the required outputs, they will be displayed on screen for immediate need and for printing

An efficient and intelligent output design improves the system's relationship with the user and supports decision making. In other words, it makes the system user-friendly to be displayed or printed according to the user's choice. Quality output is output that meets end-user requirements and presents information in a way that is clear, legible, and visually appealing. To decide on an appropriate presentation method and an appropriate format, several issues need to be considered such as who receives the output, under what circumstances the output is received, and so on.



Element	Class, %	Method, %	Line, %
com.tekact.platform.admin.service	84% (11/13)	40% (47/115)	37% (273/734)
UserService	0% (0/1)	0% (0/1)	0% (0/6)
TenantService	0% (0/1)	0% (0/3)	0% (0/21)
UserServiceImpl	100% (1/1)	0% (0/21)	0% (1/127)
TenantServiceImpl	100% (1/1)	0% (0/24)	1% (2/183)
UserGroupServiceImpl	100% (1/1)	0% (0/8)	2% (1/44)
RoleServiceImpl	100% (1/1)	0% (0/7)	3% (1/32)
ConfigValueServiceImpl	100% (1/1)	0% (0/2)	12% (1/8)
ConfigTypeServiceImpl	100% (1/1)	0% (0/2)	14% (1/7)
AppService	100% (1/1)	100% (1/1)	50% (3/6)
OrganizationService	100% (1/1)	100% (3/3)	66% (12/18)
FeedbackServiceImpl	100% (1/1)	100% (12/12)	73% (85/115)
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)
AppServiceImpl	100% (1/1)	100% (8/8)	100% (43/43)

Figure 3.7.a Organization Service Test Coverage

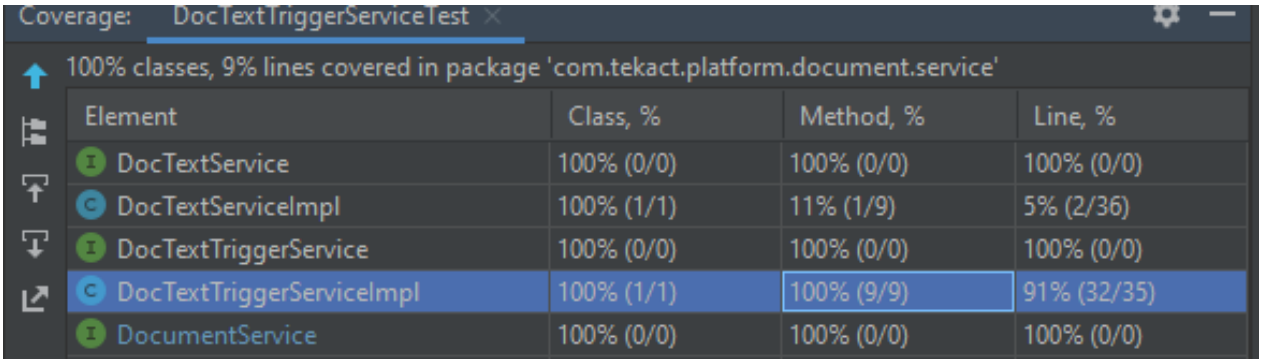


Figure 3.7.b Document Service Test Coverage

CHAPTER 4

RESULT AND DISCUSSION

Many businesses are currently at the forefront of technical development and the development of reusable services and APIs that hasten the time it takes to launch new goods. Utilizing best-in-class microservices patterns to provide scalability, performance, and resiliency, this microservices-based project shows how several services can operate independently. The project seeks to accomplish the following objectives for the Trigger editor: offering APIs for developing applications; offering facilities for composing applications using these APIs; facilitating simple integration through the use of REST APIs; managing workflows; and offering a functional application.

It is a product built on a microservices architecture, in which each microservice deals with certain issues unique to a given domain. Users will assemble these numerous tiny services to construct applications. The goal of this product is to offer APIs that let customers create apps that can function in various domains. A multi-tenant system is publisher enabled. It employs a software engineering methodology that emphasises segmenting an application into independent functional modules with clear interfaces. It has been common practise to make an effort to smoothly include such alternatives into adaptive business process management. Additionally, dividing up massive applications into separate, little services gives agile teams more independence, allowing the scaling of agile techniques. Each team is able to work on various microservices and create user stories that solely have an impact on those microservices.

4.1 Testing methods

There are different types of testing methods available. For this project, testing is implemented to uncover requirements, design or coding errors in the program.

4.1.1 Unit Testing with Mockito

There are various testing techniques available. Testing is used in this project to find requirements, design, or code flaws in the programme. The smallest testable components of an application, known as units, are separately and independently tested to perform properly as part of the software development process known as unit testing. If done correctly, unit testing can help uncover issues in the code early on that may be more difficult to find during later phases of testing, making it a crucial step in the development process. Unit testing is a part of test-driven development (TDD), a pragmatic approach to product development that takes a careful approach through continuous testing and adjustment. Prior to using other testing techniques like integration testing, this testing method is also the first level of software testing. To make sure a unit doesn't depend on any external code or functionality, unit tests are often isolated.

Though it can be done manually, testing is frequently automated. A well-liked open source framework for mocking objects during software testing is called Mockito. The creation of tests for classes with external dependencies is made significantly easier by using Mockito. An interface or class' mock implementation is referred to as a mock object. It enables the output of certain method calls to be defined. The system interaction is typically recorded, and tests can verify it. The plan, the cases and scenarios, and the actual unit test are typically the three parts of a unit test. The unit test is written and reviewed in the first stage. The creation of test cases and scripts is the following phase, after which the code is tested. Write failed unit tests first when using test-driven development. Once the test passes, they continue to code and refactor the programme. TDD frequently produces a tidy and reliable codebase.

4.1.2 Integration Testing

After unit testing, integration testing is the next step in the software testing process. Individual software pieces 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. Components 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. Integrity checks are performed to ensure that all of the modules are communicating properly.

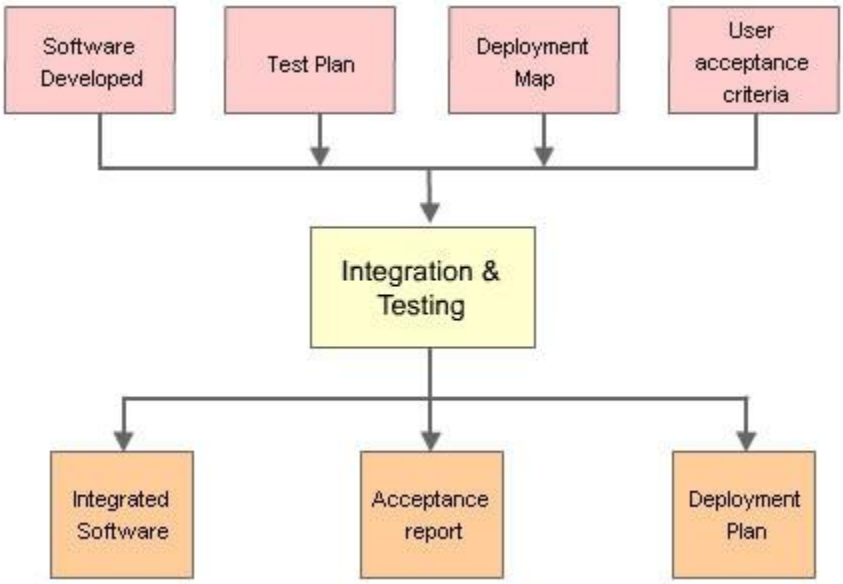


Figure 4.1.2: Testing Block Diagram

4.2 Output Screens and Results

1.Organization Controller

The ability to create, read, update, and delete (soft delete) an organisation should be available to administrators. All of the databases' organisations should be accessible via API.

1.1 Organization Address Controller

The duties of a Super Admin include adding, reading, updating, and deleting (soft deleting) addresses and contacts for an organisation. To access all of the addresses and contacts in the database, an API should be accessible.

1.2 Organization Contact Controller

I should be able to add addresses and contacts to an organisation, read them, change them, and even delete them (softly, of course). To access all of the addresses and contacts in the database, an API should be accessible.

2.Document Text Controller

The amount of lines, branches, or methods that are run by unit tests is measured as code coverage. Unit tests offer a way to verify refactoring efforts and help to ensure functioning.

3.Test Coverage Results

The amount of lines, branches, or methods that are run by unit tests is measured as code coverage. Unit tests offer a way to verify refactoring efforts and help to ensure functioning.

3.1 Organization Test Coverage

The durability and quality of the codebase are increased when developers unit test as they work. The time a development team spends on unit tests will pay

dividends because less effort will be needed in the future to debug errors and analyse issues.

3.2 Document Service Coverage

When writing unit tests for new application code, strive for higher coverage. The longevity and quality of the codebase are increased when developers unit test as they work. The development team's time investment in unit testing will pay off because it will save time on bug fixes and post-mortem analysis.

CHAPTER 5
CONCLUSION

The project's goals are to provide APIs for creating apps, facilities for using these APIs to compose applications, simple integration using REST APIs, workflow management, and provisioning. appoint a useful application. The goal of this product is to offer APIs that let customers create apps that can function in various domains. a system with many tenants. It employs a software engineering methodology that emphasises segmenting an application into independent functional modules with clear interfaces. A preliminary effort was undertaken to identify a demand for the system. A thorough research that is user-friendly and simple to use has been created to fulfil the needs of users. Even people with little technical expertise can easily utilise this system because to its beautiful appearance. It's a fantastic system. It is a product built on a microservices architecture, in which each microservice deals with certain issues unique to a given domain. Users will assemble these numerous tiny services to construct applications. Many businesses are currently at the forefront of technical development and the development of reusable services and APIs that hasten the time it takes to launch new goods. Utilizing best-in-class microservices patterns to provide scalability, performance, and resiliency, this microservices-based project shows how several services can operate independently.

5.1 Future Enhancement

The system is constructed in a way that makes adding additional modules very simple. The system will be more adaptable after being rebuilt. In order to take use of the sophisticated capabilities of this technology, the system has been designed to be as adaptable and user-friendly as feasible.

Deliver a functional application, manage workflows, and integrate quickly using REST APIs. Users can quickly and easily receive notifications based on specific occurrences. Future work will also include containerization (Docker and Kubernetes), cloud storage (AWS), and front end development (responsive native for mobile, ReactJS for web).

REFERENCES

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

[2] Kavya Guntupally ,”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 Using Tesseract”-OCR, 2020 IEEE 13th 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 Informations 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 and Menaka Pushpa Arthu,“Survey on Template Engines in Java”,School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus,2020 India 600 127

[8] Lei, Kai, Yenning Ma, and Zhi Tan.” Performance comparison and evaluation of web development technologies in php, python, and node.js.” Computational Science and Engineering (CSE), 2021 IEEE 17th International Conference on. IEEE.

[9] Casimir Désarmeaux, Andrea Pecatikov, and Shane McIntosh.” The Dispersion of Build Maintenance Activity Across Maven Lifecycle Phases”.2021 IEEE/ACM 13th Working Conference on Mining Software Repositories

[10] Dheeraj , Kalpana Sharma, “Security Testing of API using Postman and Swagger tools and its use in Internet of Things (IOT) ” ,2020

APPENDIX

Screenshots

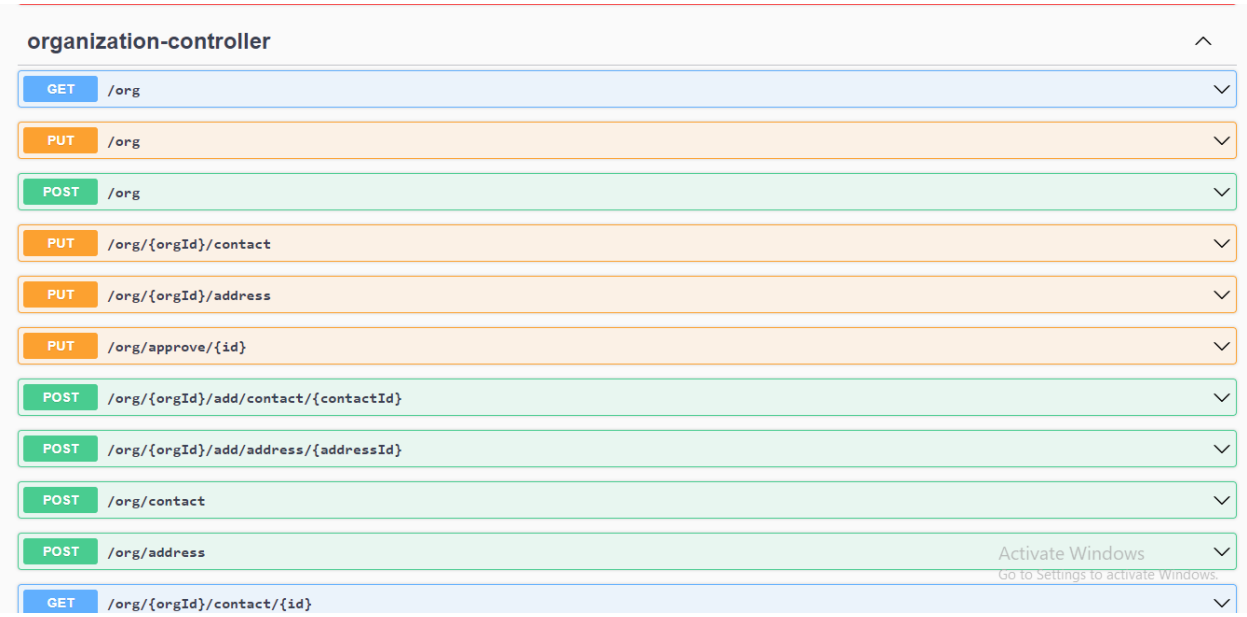


Figure A.1 Organization Controller

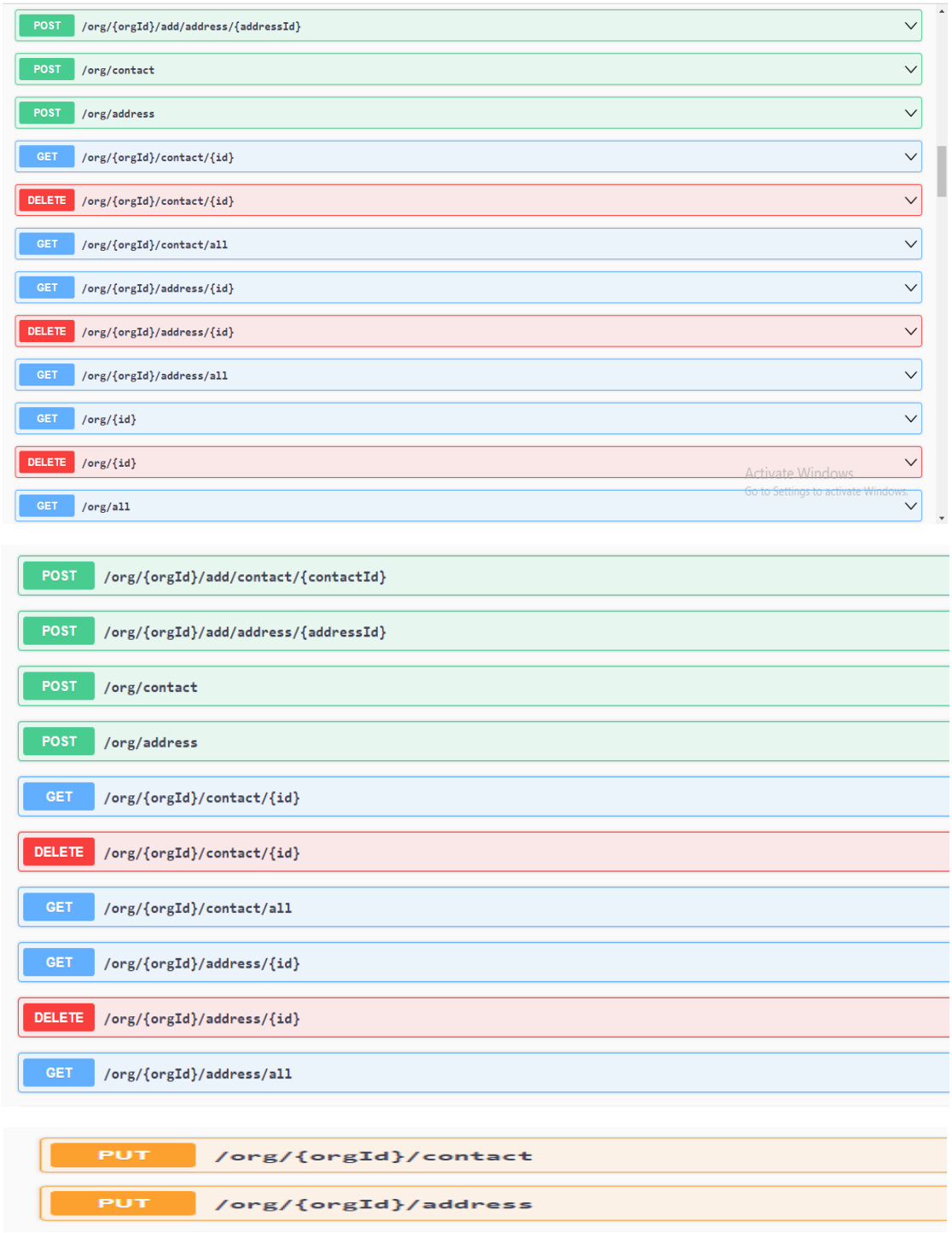


Figure A.2 Organization Address and Contact Controller

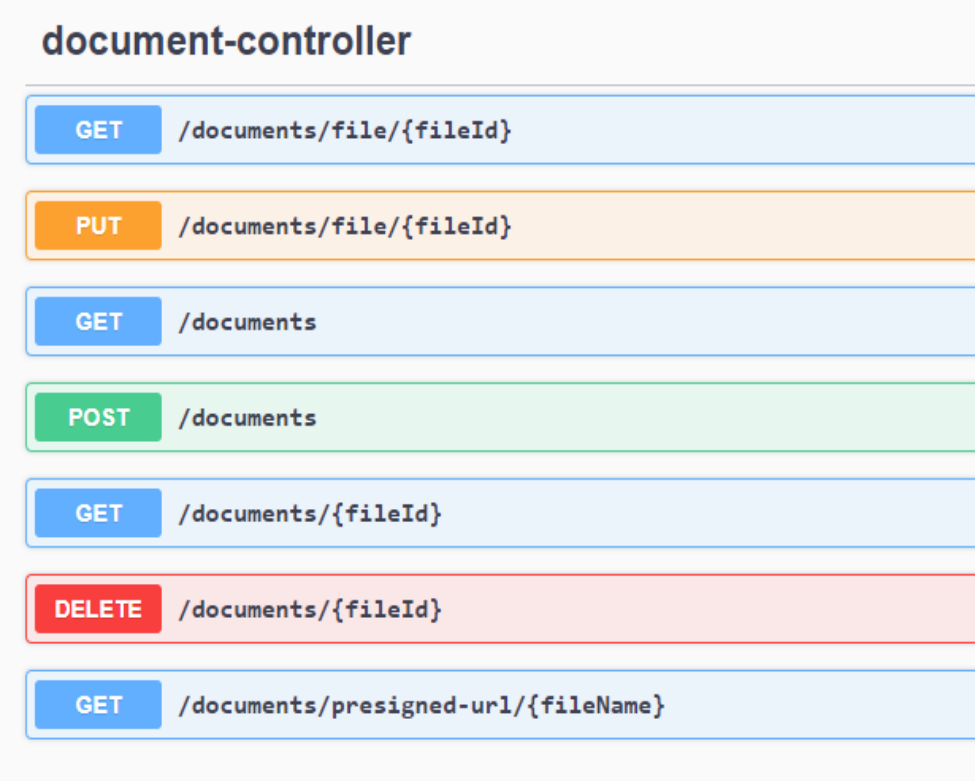


Figure A.3 Document Controller

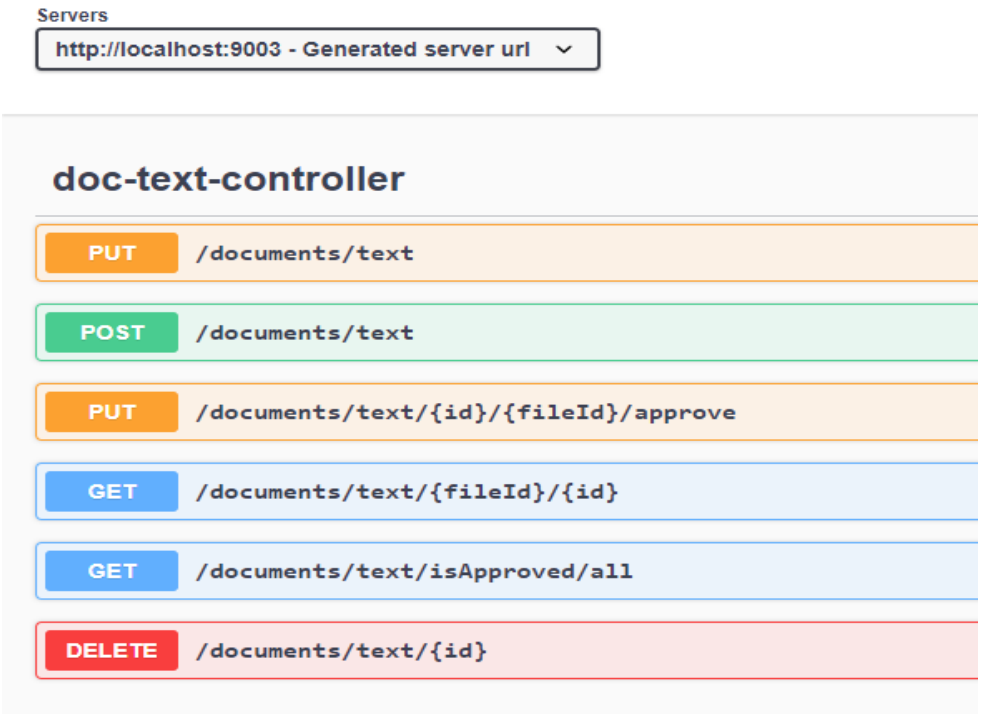


Figure A.4 Document Text Controller

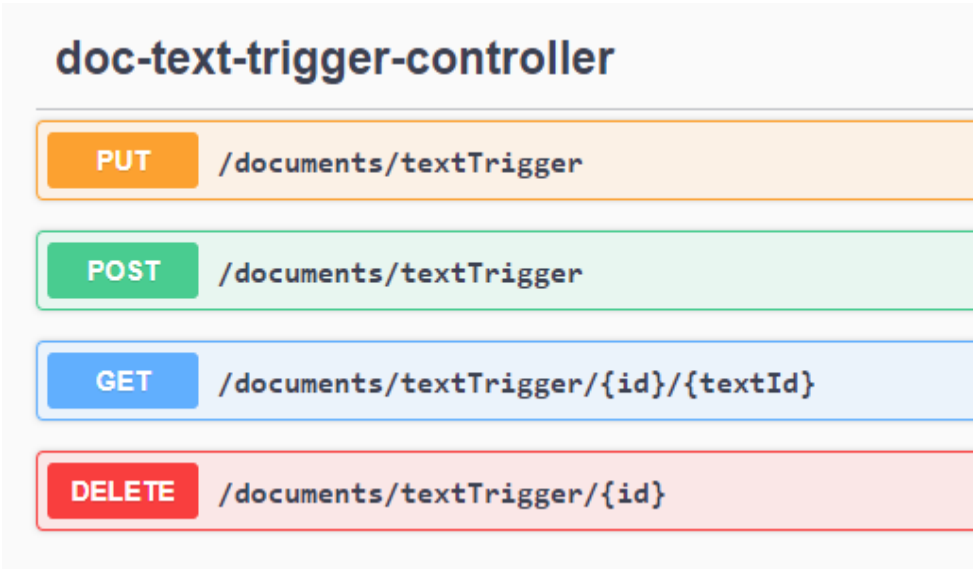
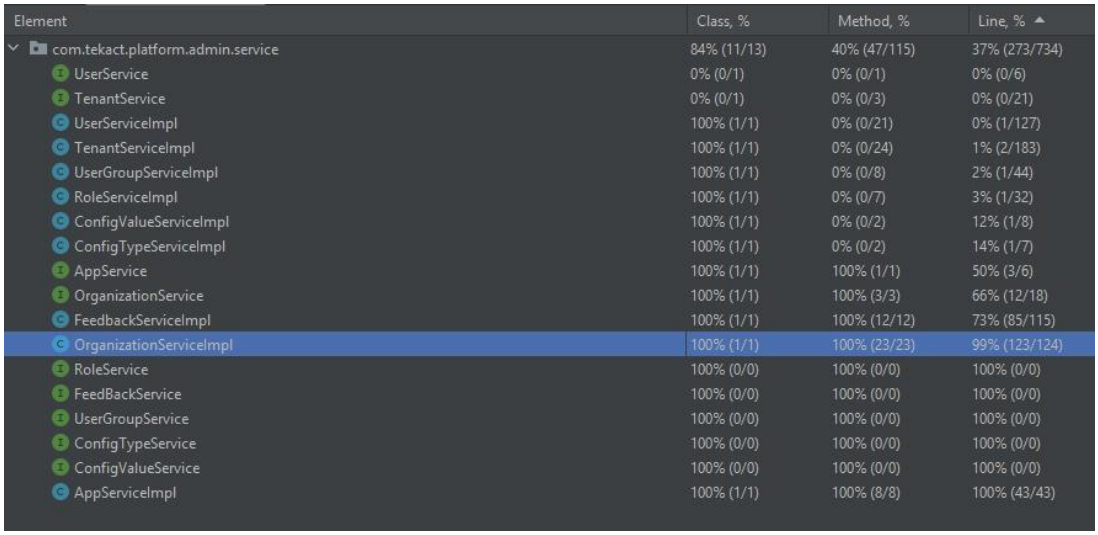


Figure A.5 Document Text Trigger Controller



Element	Class, %	Method, %	Line, %
com.tekact.platform.admin.service	84% (11/13)	40% (47/115)	37% (273/734)
UserService	0% (0/1)	0% (0/1)	0% (0/6)
TenantService	0% (0/1)	0% (0/3)	0% (0/21)
UserServiceImpl	100% (1/1)	0% (0/21)	0% (1/127)
TenantServiceImpl	100% (1/1)	0% (0/24)	1% (2/183)
UserGroupServiceImpl	100% (1/1)	0% (0/8)	2% (1/44)
RoleServiceImpl	100% (1/1)	0% (0/7)	3% (1/32)
ConfigValueServiceImpl	100% (1/1)	0% (0/2)	12% (1/8)
ConfigTypeServiceImpl	100% (1/1)	0% (0/2)	14% (1/7)
AppService	100% (1/1)	100% (1/1)	50% (3/6)
OrganizationService	100% (1/1)	100% (3/3)	66% (12/18)
FeedbackServiceImpl	100% (1/1)	100% (12/12)	73% (85/115)
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)
AppServiceImpl	100% (1/1)	100% (8/8)	100% (43/43)

Figure A.6 Organization Service Test Coverage

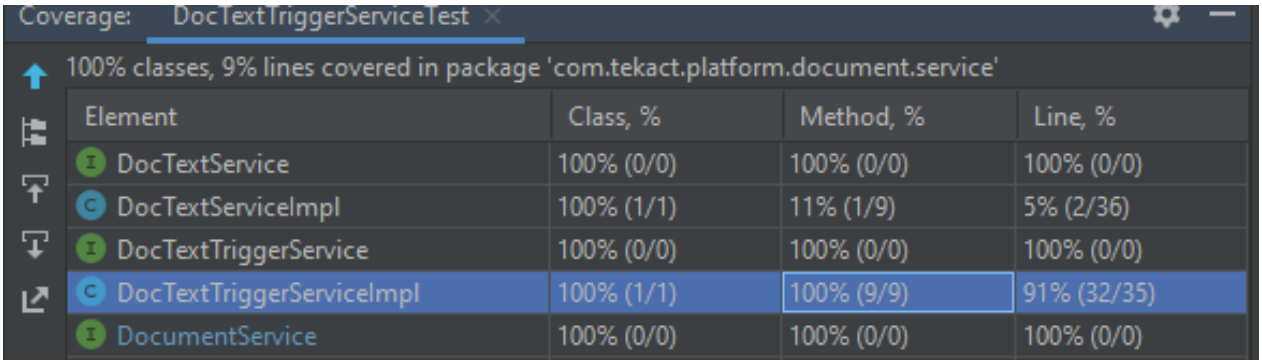


Figure A.7 Document Service Test Coverage