TEKACTPUB

A PROJECT REPORT

Submitted by

SINDOORI B

REG NO: TKM20MCA-2038

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 **TekactPub**, 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 SINDOORI 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 "**TekactPub**" submitted by **SINDOORI B (TKM20MCA2038),**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 Private Limited,
Balaraj's Arcade, 4th Floor,
Whitefield Main Road,
Opp Brigade Cosmopolis.
Bengaluru – 560066

Tekact
www.tekact.com

TO WHOM IT MAY CONCERN

This is to certify that Ms. Sindoori 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

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.

SINDOORI B

ABSTRACT

In event-driven architecture, when a service performs some piece of work that other services might be interested in, that service creates a record-setting occasion of the executed a task. These events are used by other services , so that they can perform any of their own tasks needed as a result of the occasion. Services that create requests in contrast to REST do not need to know the details of the services consuming the demands Here is an easy illustration: When a command is placed on an ecommerce site, a single "order placed" event is generated and then consumed by several microservices. Events can deployed in different ways. For example, the events can be deployed to a queue that guarantees delivery of the event to the suitable consumers, or events should be provided to a "pub/sub" model stream and the interested parties are allowed to access the publishes the events . In either case, the event is published by the producer , the customer receives it, and they respond to the events . Note that these two parties may alternatively be referred to as the publisher and the subscriber in some 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. A system with many tenants, then it employs a software engineering methodology that emphasises breaking down an application into discrete components with clear interfaces. The system's necessity has been determined in an initial effort. A thorough research that is user-friendly and simple to use has been created to satisfy user needs

Contents

List of Figures

CHAPTER1
INTRODUCTION

Tekact Private Limited is creating a system called TekactPub to broadcast events to a specific audience. Many independent services is comprised in these solution , each of these solutions suites to a different requirement. Users will use these small services to create such applications. Users are able to create many applications that work in a variety of industries with the help of the APIs that created. The user base of the system TekactPub is sizable. It is a method of software engineering that places an emphasis on breaking down programmes into manageable chunks with simple interfaces.

The development of reusable services and application programming interfaces (APIs) shortens the time it takes for new products to reach the market today, when many firms are at the forefront of technological growth. This microservices-based solution illustrates how several services may function independently of one another by using the best microservices concepts for scalability, performance, and robustness. The project's goals included developing APIs for building apps, strategies for putting apps together using these APIs, straightforward REST API interaction, workflow management, and a working app. These APIs will enable users to create applications that operate in a range of different contexts. It is a method of software engineering that places an emphasis on breaking down programmes into manageable chunks with simple interfaces.

1.1 OBJECTIVE

The following objectives are set for the TekactPub product by the project:

1. Make APIs available for app creation

   Using Spring Boot for your REST APIs has a number of advantages, including no need for intricate XML setups. Tomcat server embedded for Spring Boot apps, a Spring Boot auto-configuration functionality that sets up your application's dependencies automatically.

   .

2. Allow developers to use these APIs to build apps.

   A REST API is a web app, just like all the other web apps we've made in the past. The primary difference between the two is that a GET request sends data rather than displaying a website. Furthermore, it allows POST requests from other programmes as opposed to generating them by using HTML forms on its own.

3. REST APIs can make it simple to connect things.

   For example, the classes and methods we used to make Processing code are part of the reference, or API, for Processing. The Java API is the same thing as the set of classes and methods we use to write Java code. A programming interface (API) is a list of things that programmers can do. So, when we say we're making a REST API, we mean that we're using REST ideas to make a tool that programmers can use to interact with our data.Workflow management

4. Keeping an eye on the progress of the project

   Finding, organising, and managing a series of activities that lead to a certain outcome is what workflow management is all about. Optimizing, optimising and automating processes is the goal of workflow management, which is all about increasing production, eliminating duplication and reducing errors.

5. Provide a working App

   This project conforms to the scrum process so as to provide functioning apps. With the assistance of a team of Scrum masters, an Agile project management approach, the objective is to eliminate all obstacles to the task's completion. Regular stand-up meetings are held to address the day's tasks and any possible impediments.

1.2 COMPANY PROFILE

Tekact Private Limited is a technology services and product development company formed by technology experts with the intention of achieving breakthroughs using open source technologies. Tekact produces high-quality online, mobile, and cloud-based applications.

This organisation was created and is supported by former employees of Oracle, IBM, HSBC, and Yahoo! Professionals from the organisation contribute to many open source technologies to bring new goods and services to corporations and startups. The Architects, Developers, Testers, and Operations team members are passionate experts in their respective fields.

Services

1. Software Development

Data drives company activities in the current industrial setting. Rarely do firms get business-critical data about customers, suppliers, products, and procedures. The overwhelming majority of organisations outsource Big Data to find and implement the solutions that might turn their data into valuable information. Big Data stores, enhances, and provides analytical capabilities for all types of data, regardless of its volume, velocity, variety, truthfulness, variability, or worth.

Data aggregation is the process of gathering raw data from a variety of sources and making it available for further analysis. Data may come from RDBMS, files, RSS, social media platforms, and other sources. Before import and processing, data validation is the process of verifying the accuracy and quality of your data. This is a key step for any Data lake, since the available data must be made consistent and trustworthy for effective analytics.

Governance of Big Data is the process and administration of ensuring firm data's accessibility, utilisation, integrity, and confidentiality. It encompasses the whole process, from storing the data to safeguarding it against any tragedy. Through the use of data analytics, people and organisations are able to analyse data. Diverse sources of unprocessed data are modified to provide economic value. We provide descriptive analytics for determining what has occurred in an organisation, predictive analytics for determining what may occur, and prescriptive analytics for establishing what ought to occur. Big data and cloud computing are two separate concepts, yet they have grown so intertwined as to be almost synonymous. It is essential to identify and appreciate the relationship between the two concepts.

2. Quality Assurance

Meet customer specifications with precise software. We verify that the business functionality conforms to software specifications.

3. Consulting

Construct your apps using the most successful industry patterns and best practises on the most efficient platforms. Big Data, Micro Services, Event Driven, Kappa/Lambda, and Client-Server Architectures are all acquainted to our staff. We apply the most efficient business patterns to design software that is both maintainable and effective.

4. Build Operate Transfer

Our expertise will assist you in hiring top talent for your organisation, managing your teams to ensure on-time, high-quality delivery, and reallocating your teams to the most crucial duties. In order to do this, an experienced team will be embedded inside the teams; after the teams have stabilised, the experienced team will move on to their subsequent tasks.

During the construction phase, a software R&D office must be formed abroad. We organise a team of developers and search for office space for a client's project. As a client, you oversee all actions and make monthly payments.

The operating phase includes operational management. We will handle scheduling, payroll, bookkeeping, and human resources, allowing you to concentrate on your product. Additionally, you get legal consultation and financial aid.

After the client accepts our operational management services, the transfer period begins. As the original owner of the project, no transfer is possible. Once customers can independently run the R&D office, our specialised employees will leave.

CHAPTER 2

LITERATURE SURVEY

A literature review is a thorough analysis and review of all the relevant information. When doing a literature review, you choose relevant research questions and try to answer them by finding and evaluating relevant literature. A benefit of literature reviews is that they can give you new ideas by letting you look at the study's data in a different way. A literature review is a summary and explanation of what is known about a certain topic right now, as shown in scholarly articles and books. Literature reviews can be given to college students in one of two ways: as an independent assignment in a class or as an introduction or preparation for a larger paper, like a thesis or research report. Depending on the type of review, the topic and point of view of the review, as well as the type of hypothesis or thesis argument presented, will be different. One way to understand the differences between these two types is to read published literature reviews or the first chapters of theses and dissertations in our own field, analyse the structure of their arguments, and pay attention to how they approach the problems.

2.1 PURPOSE OF THE LITERATURE REVIEW

1. It makes it easy for people to find research on a certain topic by putting together a summary of relevant, meaningful, important, and valid articles or studies.
2. It gives researchers who are starting to look into a new area a great place to start by making them summarise, evaluate, and compare the original research in that area.
3. It stops researchers from doing work that has already been done.
4. It can give hints about where future research will go or suggest areas to focus on.
5. It shows what was found and what was most important.
6. It shows where there are gaps, differences, or contradictions in the literature.
7. It shows how and why other researchers did their work, which is useful.

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 MICROSERVICES

A microservice architecture decomposes a business domain into small, consistently bounded contexts implemented by autonomous, self-contained, loosely coupled, and independently deployable services . One of the pioneers of microservices was Netflix which began moving away from its monolithic architecture in 2009 when the term microservice did not even exist. Microservice-based applications scale well horizontally, not only in the technical sense, but also concerning the organization's structuring of developer teams, which can be kept smaller and more agile. Efforts to seamlessly integrate such options into adaptive business process management have already found their way into business practice.Furthermore, splitting large applications into individual microservices provides the next degree of independence to agile teams, which supports scaling agile methods. Each team may work on different microservices and develops user stories that affect only their microservices [1].

2.2.2 SPRING BOOT

The popular Java-based Spring Boot framework for making web and enterprise apps, and how it gives service-oriented architecture the flexibility it needs (SOA). One problem with apps that are built on Spring is that it can be hard to figure out how they are set up. Spring Boot makes it easy to make stand-alone Spring apps that are ready for production and can be deployed with very little Spring configuration. Spring Boot makes it easier to manage dependencies by using a complete but flexible framework and all the libraries that go with it as a single dependency. This gives you all the Spring tools you need to start a project, which is more than you need for CRUD web apps. This framework has some extra features that are useful for many projects, like an embedded server, security, metrics, health checks, and configuration that can be done outside of the framework. Most of the time, a war file is used to put a web application on a web server. Spring Boot applications, on the other hand, can be packaged as either a war file or a jar file.

This means that they don't have to be installed or set up on the application server in order to run[2].

2.2.3 OCR

Before Optical Character Recognition (OCR) can work, a document's text lines, words, and symbols must be broken into the right pieces. There is a lot of text on the internet in English, Hindi, and many other languages spoken all over the world. But most of the tools and methods that are available are for English. This shows that there isn't a good tool for text mining that works well with languages other than English. In this paper, we suggest a software interface, a web application, that automatically translates image documents into any Googletrans language. This will help us figure out what to do. Python-tesseract is a tool in the programming language Python that uses optical character recognition to read text (OCR). In other words, it can find and read hidden text in pictures. Google trans, on the other hand, is a free library for Python that uses the Google Translate API. Using the ideas in explain, you can make a web app that takes an image file of a document as input and returns the translation. Several steps are taken with the scanned image during the pre-processing phase. It makes the picture look better when it's being made and when it's being shown. It's good for breaking something into smaller pieces, and after noise reduction, the image is segmented [3].

2.2.4 TESSERACT

Tesseract is made so that it can be used with any language. At first, Tesseract was built to find white on black. Which led to the design, which was based on how the parts fit together and how their shapes were used. You can turn a picture into text with Tesseract-OCR. Tesseract is made up of several algorithms that work well together. "tesseract-OCR" is a part of the "Python-tesseract" module that turns an image into text. It's easy to use because all we have to do is include the module and call the defined method, which is 'image to string(image,lang='lang code'), which turns an image into a string. We need to make it clear that the document was written in a certain language. Tesseract sends back an image with different text. Tesseract-OCR is a library that does a great job and is easy to work with. We use the library function image to string(image file, language), which can work with.jpg,.png,.gif,.bmp,.tiff, and.bmp files [4].

2.2.5 POSTGRESSQL

PostgreSQL is a strong, free, and open source database framework. At protests, it can be used. Over more than 15 years, it has grown and changed, and its design has been tried and tested. It has a strong reputation for being right, correct, and reliable because of this. PostgreSQL is a way to handle database management (DBMS). A group of people worked on it for free. "post-gre-SQL" is how you say it. PostgreSQL is not owned by a company or other private group, and the source code is free to use. PostgreSQL was first called Postgres, and it was made by Michael Stonebraker, a professor of software engineering at UCB. In 1986, Stonebraker made Postgres to replace a programme called Ingres, which is now owned by Computer Associates.

PostgreSQL keeps running on all of the major working frameworks, such as Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows. It can work with text, pictures, sounds, and videos, and it has programming interfaces for C/C++, Java, Perl, Python, Ruby, Tcl, and Open Database Connectivity (ODBC). Lists help you find what you're looking for more quickly. When you add a record to a section, it's usually easier to find the information you're looking for. In exchange, each record will take you more time to fill out. When you embed information in a list, the information has to be written in two places and the order of the file has to be kept as you embed information [5].

2.2.6 RESTFUL API

This paper was put together in a way that was meant to Implementing a RESTful API web service will definitely make it easier to build software applications outside of the system or using different programming languages or platforms. The web service architecture for the Takeaway app will be built with the RESTful API. URIs work better when they have parameters like filtering, sorting, selection, and pagination. The website is the back end of the Takeaway app, and the Android app is the front end. By using the Postman app to test the function method, it is clear that the server's REST API Server has been working well. When the Apache JMeter app is used to check the response time, it shows that the response time is good. When responses and requests to SOAP and REST architectures are compared, it's clear that REST takes less time [6].

2.2.7 THYMELEAF

Thymeleaf is a Java template engine that can be used in servlet environments as well as web applications. It works best with HTML5 web apps, but it can also work well with apps that aren't online. It can be used instead of JSP because its template files can be opened in the browser. These are the most important parts: It can be used to record anything. You can use it as a place to start if you need a template engine. It can figure out what type of document it is and make the necessary changes. A feature called "parsed template cache" keeps the amount of data coming in and going out to a minimum. This means that both the way it works and how well it works are very good. It works with and without an internet connection. The best thing about this engine compared to others is that templates work like prototypes.

This makes using templates easy. The templates can be opened in the browser even when there is no server running. This means that the templates are always shown the same way. It's like putting together a template engine . The goal of Thymeleaf is to make it easy to make templates that look good and work well. XML tags and attributes are used to make it work. Instead of putting that logic as code in the template, these XML tags tell the DOM (Document Object Model) to run the logic that has already been set up. It can be used instead of JSP [7].

2.2.8 OPENAPI OR SWAGGER

The standard way to talk about web APIs is now with the OpenAPI or Swagger specification (REST, RESTful, etc.). Because of this, a lot of research articles have been written and will continue to be written about this spec. You can improve, simplify, solve, and/or automate a lot of software development tasks, activities, and problems with this specification. The goal of this study is to find out everything that can be done with the OpenAPI/Swagger specification. It shows, among other things, what development activities it has been useful for, what software artefacts (results) it has made, and what domains it has covered. For the literature review we looked at, 47 articles were written between 2011 and 2020. The search matched these criteria: a) a mention of the development activities where the specification has been useful, b) a mention of the software artefacts (results) that it creates and how they are validated, and c) a mention of the domains that are covered. The results show that the OpenAPI/Swagger specification has helped software development in many ways, but mostly by improving documentation (43 percent). In the same way, 77% of the articles on the web API help people with their needs or problems. The most common result in 68 percent of the manuscripts is the use of different tools to automate tasks. Also, 40% of the works looked at are in a certain field (IoT, Cloud, etc.) [8].

2.2.9 MAVEN

Maven is a great project management tool. It's related to POM (project object model). It's used to plan projects, build them, and keep track of them. Maven makes it easier for Java programmers to do their jobs every day and makes any Java-based project easier to understand. Maven repositories are collections of information about JAR files and lists of those files. Metadata are POM files that describe the projects that each packaged JAR file is part of, including what external dependencies each packaged JAR file needs. With this information, Maven can download the dependencies of your dependencies until all of the dependencies are on your local machine. With Maven, a project is built in six steps, and each step depends on the one that came before it.

The stages of Maven's default lifecycle are made up of these steps. - Validate checks to see if the project has all of the settings it needs to be built. Compile takes the project's source code and turns it into the project's raw deliverables (e.g., .class files). Test builds and runs test suites to find bugs. Packaging prepares raw materials for use (e.g., .jar, .war, .ear). Install puts the deliverables on the local system where they belong. When you click "Deploy," the local installation is sent to the place where it will be used. Maven is a tool that helps developers manage and understand projects by giving them a full framework for the build lifecycle. Maven has a standard directory structure and a standard build lifecycle, so it doesn't take long for the development team to automate the project's build infrastructure.

In a setting with many development teams, Maven can set up the way to work so that it follows standards in a very short amount of time. Developers can easily set up automated reports, checks, builds, and tests with Maven [9].

2.2.10 POSTMAN

A test in Postman is just a piece of JavaScript code that is run after a request has been sent and a response has been received from the server. It's very easy to use POSTMAN. It gives a list of API calls, and to test the APIs of an app, you have to follow that list. You can write and run tests for each request with Postman. These tests are written in the programming language JavaScript. Download the app Postman to your phone. It can be downloaded as a native app for macOS, Windows, and Linux. There's also a Chrome app for it. If you use Chrome, click on the square that says "Apps." The most common ones are GET (to get data), POST (to change data in an existing file), PUT (to replace an existing file), and DELETE (to get rid of data) (Delete data from server). Send Request: a. In the toolbar at the top of the page, click "New." b. Type the request URL (URL-REST endpoint) as shown in the image below. URL: http://get.postman-echo.com. Users can save requests for later use by clicking the Save button next to the Send button. You can use Newman to connect to the tool for managing Build. It is a command line tool that works with Postman and lets you connect Postman to a build management tool to help automate testing. Authorization: The authorization process checks to see if you are allowed to get the information you want from the server. It makes sure that security is working. In the next part of this paper, we'll talk about the different kinds of permissions that the tool has [10].

CHAPTER 3

METHODOLOGY

In “TekactPub”, the CRUD (create, read, update and delete) and list operations is done using the Rest API and Springboot framework. When it runs, the controller gets the request parameters and path variables from the URL. After that, it calls the service class methods, which must be autowired in the controller. The interface then calls the service implementation class, which runs the code with the parameters from the Rest API calls. For the implementation, the right models and entities are needed. This means that when a client sends a request through Postman: It passes through the controller class

- Then it is put into action in the service class and the implementation class. After that, it does what it should do next.
- The data is then sent to a database called postgresql.
- A model can be sent as the body when calling an API.
- The model is sent to the implementation class, which runs the code written in the class and then adds the model to the entity, which stands for the Database.
- Either BaseRepository or JPARepository makes the interface of Repo better. These add some basic methods, such as findById(), findAll(), and so on.
- The Entity class is built on the BaseEntity class, which has some entities in common.
- Helping people with small jobs.

3.1 ARCHITECTURAL DESIGN

Organization is the parent, and everything else is its offspring. The group is where renters sign up for apps, and the group also makes apps. Users are both a part of the Organization and a part of the tenants. Each user has a role and works with a group of other users. Users can access the resources related to permissions. Software components, their properties that can be seen from the outside, and how they connect to each other make up the structure of a computer system's software architecture. It shows how the program's big building blocks fit together.

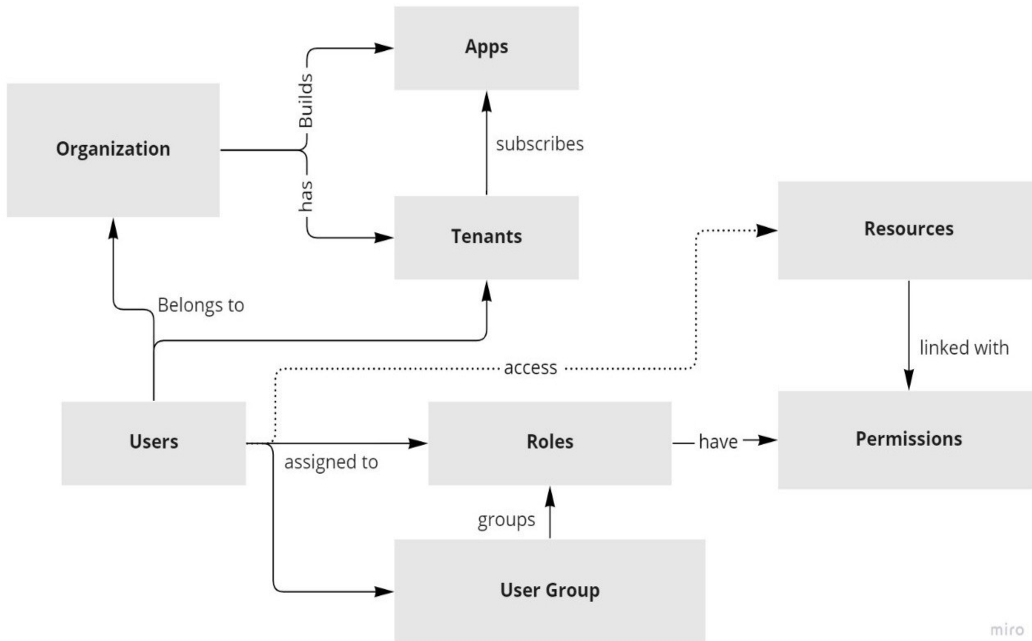


Figure 3.2: Architectural Design

You can figure out how a computer programme is put together by looking at the analysis models and how their subsystems work with each other. The main goal is to give the programme a modular structure and show how the different parts fit together. Architectural design is the process of figuring out how a system is made up of smaller parts, called subsystems, and how these subsystems work together and talk to each other. At the end of this process, a description of how the software is built is given.

3.2 MODULE DESCRIPTION

- Admin
    - Organization
    - Tenant
    - User

- Role
- User Group
- Feedback
- Apps

- Common
- MasterData

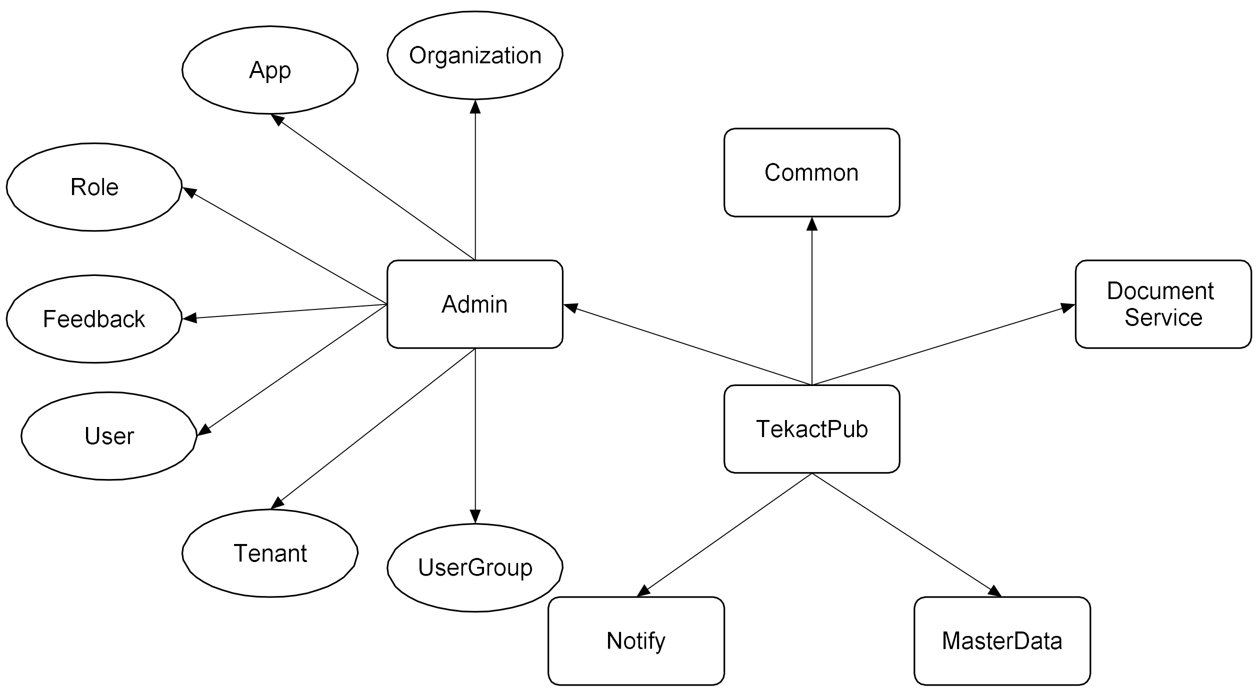


Fig : Module Description of Tekact Pub

3.2.1 ADMIN

The Admin module takes care of all the basic entities, such as the organisation, tenant, app, user, user groups, and roles. You can control how Business Process Server works, add and remove apps, and manage all users and groups with the Administration module. Here, admin tasks are done, like managing users, giving them permissions, and making sure they are who they say they are. Most of the time, the APIs can be used by the company, its tenants, and its users through this Platform service. Administrators have to do things like add, remove, approve, and update users. Also, adding, removing, approving, and updating organizations, end users, tenants, different user roles, and building apps.

3.2.1.a ORGANIZATION

All the other parts of this project are taken care of by the Organization. As an example, Tekact is a company that lets its customers use different APIs to build apps. All other entities, like tenant, user, role, and app, depend on this parent entity.

3.2.1.b TENANT

Users who need different APIs from the company can be tenants. A tenant can be an individual person or a whole business. The same business can rent to more than one person. The renter signs up for the different apps that the group.

3.2.1.c USERS

A user can be in more than one group at the same time. Either they are renting or they are a big group. It is the job of the administrator to let people in and out. Users can be superusers, admin, end-users, tenants etc. As Organizational Admin, I should be able to add users, see their information, change it, and "soft delete" them. You should be able to reach all of the users in the database through an API. I should also be able to give different apps to different people. One person can have more than one role. Someone can be added to a group.

3.2.1.d USER GROUPS

A user group is a group of users who share a set of permissions that are given to the group and then to each user in the group. When a user joins a user group, the permissions that group has are given to that user. As an Organizational Admin, I should be able to create, read, update, and (softly) delete UserGroups for a Tenant, as well as assign roles to UserGroups. There should be an API that lets you get from the database all of a tenant's user groups. I should also be able to give users roles. Both users and people from other groups can be in the user group. Eg: Employees, Developers etc

3.2.1.e APPS

APIs are the parts of apps that allow them to work. As an Organizational Admin, I should be able to create, read, update, and soft-delete documents (get rid of apps without actually deleting them). You should be able to get to all of the apps in the database through an API. I should also be able to sell apps to companies. Apps are made by the company. The renters are the ones who sign up for these apps. The organisation and administrator run and take care of the apps.

3.2.1.f ROLES

Admin gives users and groups of users different roles and takes care of them. Everyone is given at least one role. Roles can be like developers , managers etc.. The job of the administrator is to assign roles to people and keep track of who has what. Roles are given to users and groups of users. Depending on what you do, there are different ways to use resources. Roles can also be used to figure out who can do what and what they can do. Role module needs to be able to do CRUD operations, like create role, get role by id, get all roles in the database, delete role, and update role.

A role's body can be a model. The model is sent to the database, where it is saved. With the assignRole method, users or groups of users can be given roles. Users or user groups can have more than one role. Depending on what role they were given, they can use the resources. UserGroup includes both individual users and groups of users. On this page, you can also delete a user's roles. You can take away a role from a user or usergroup by using the user or usergroup controller. Find out what a user's roles are and what they are.

3.2.1.g FEEDBACK

Through the feedback management module, users can tell the client about their experiences on the platform and rate how happy they are with the service. This feedback tells the people in charge of the app how people use it and how to make it better, which helps them make better products. The Feedback module lets users send messages to the system administrator that aren't about a specific patient but are about the system as a whole. Users won't have to send all of their questions and ideas to the same Admin. Instead, they can send them straight to the person who can help them the most. Feedback can be positive or negative.

It should be able to write, read, and change feedback as a user with different roles. It can get all of the information in the database, whether it's about an organisation or a tenant. You can get rid of feedback, close it, or do both. Close can be changed from 0 to 1 to stop the feedback. Validating feedback is set to 1, invalidating feedback can be set to 0, and validating feedback can also be set to 0. A user can post feedback with the create method and userId. The feedback can also be changed by the user. The comments section is still there, even though it is closed. You can get all of the feedback from the database by using either the feedbackId or an organizationId. Feedback, whether it's good or bad, helps a group get better.

3.2.2 COMMON

In our project, we use a technique called "multimodule." This means that each module does the same thing. This unit has all of those things. All of the other modules use the classes in the common module. Here are the things that are the same for all models and entities. This module uses a number of methods and features from the base repository. This module also has a common exception class and handler that are used by other modules like admin, master data, document service, etc.

3.2.3 MASTERDATA

The admin module is the parent of the master module. We figure out how modules and bundles need each other. Things like the country, city, state, contact type entity, organisation type entity, and address type entity are taken care of by the master module. The base entity and base repository from the common module, as well as a number of its features and methods, are used by the entities in this module.

3.2.4 DOCUMENT SERVICE

From the document service, you can use APIs to create, upload, update, download, delete, list, and get documents. OCR: Use the image you upload to make metadata out of it. Image data can be made ready to use by making it grayscale, smoothing it, getting rid of skew, and filtering it. Try to find as many words, lines, and people as you can. Make a list of the ranks of the candidate characters by using the trained data set. In this case, the setDataPath() method is used to set the path of the trainer data. After the characters have been recognised, choose the best ones based on how sure you are about them from the previous step and language data. Language data includes dictionary, grammar rules, etc. It makes office work go faster and better.

3.3 RESTFUL WEBSERVICE

REST is a type of architecture that uses HTTP Protocol and is based on web standards. It is built on resources, and each part is a resource that can be accessed through a standard HTTP interface.

3.3.1 WORKING

To understand how RESTful APIs send data between clients and applications, it helps to break the process into two steps.

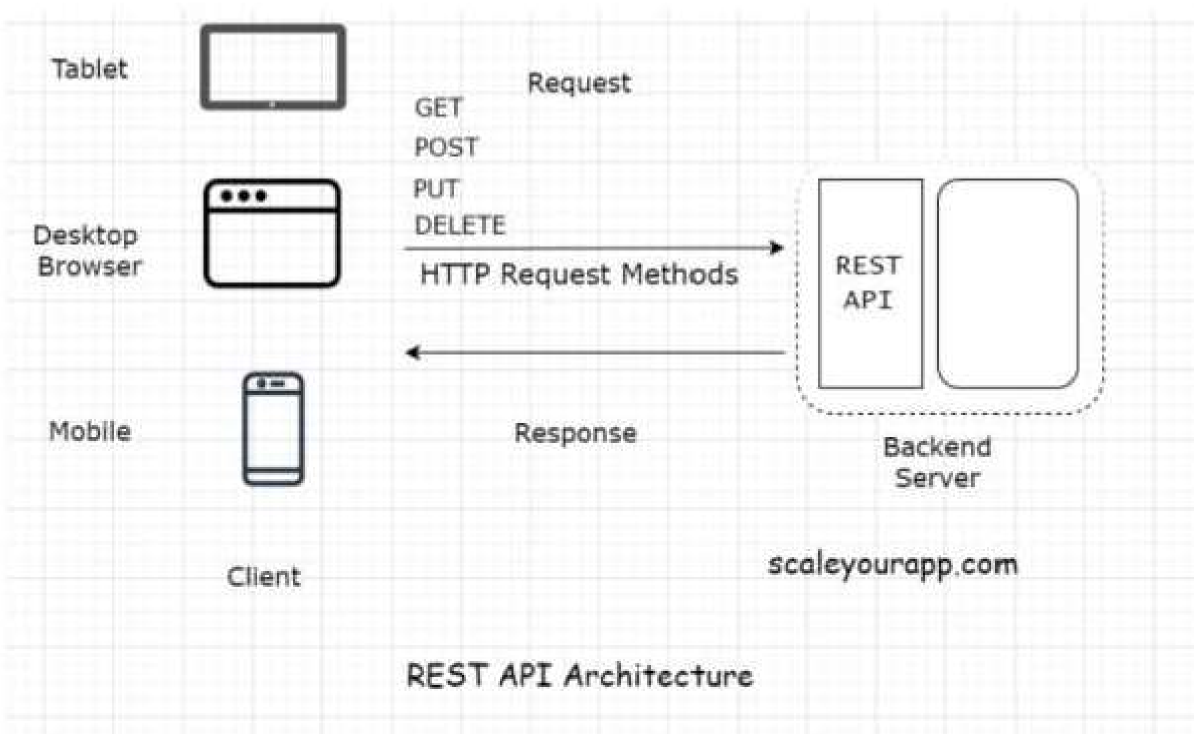
3.3.2 CLIENT REQUEST

A client is a person or programme that uses the API's services. For example, your programme would be the client if you wanted to connect it to a web app like Instagram or YouTube. If you asked for a URL, your browser would be the client. Most of the time, HTTP is used in these ways:
"GET" means "to get a resource."
To change something :

- PUT: To change or add to a resource that is already there.

• DELETE: To delete a resource.

For instance, an HTTP request to Instagram's API could be a GET request for a specific video or post or a POST request to upload a new photo. In the same way, a call centre platform that works with an auto-attendant application might use the PUT command to change a custom greeting or the DELETE command to get rid of it.



3.3.3 SERVER RESPONSE

A piece of information that an API gives to a client is called a resource. This could be anything, such as hashtags, profiles, comments, web addresses, tweets, and so on. A unique name for each resource is a resource identifier. A server stores all of the resource identifiers. When a client makes a request through a RESTful API, the server sends the client's system a standard representation of the resource's state. This means that the server doesn't send the client the real resource. Instead, it sends a picture of what the resource looked like at a certain time. The answers are given in a way that is easy to understand. The JavaScript Object Notation (JSON) format is popular because it can be read by both people and computers. It's also a great way to make it easier for people to use the Internet. It can also work with a lot of different programming languages. Data from an API can also be saved as XML, YAML, CSV, HTML, or plain text.

3.4 SYSTEM SPECIFICATIONS

In this section, the application development architecture chosen for this project is explained in terms of the requirements.

3.4.1 SOFTWARE SPECIFICATION

- Programming Language: JAVA

Java is an object-oriented programming language that can be used to make software for a wide range of platforms. When a programmer writes a Java app, the compiled code (called "bytecode") runs on most operating systems (OS), like Windows, Linux, and Mac OS. The languages C and C++ gave Java a lot of its syntax. James A. Gosling, a computer scientist who used to work for Sun Microsystems, made Java in the mid-1990s with Mike Sheridan and Patrick Naughton.

Java makes applets, which are programmes that run in browsers and make it easier for people to use graphical user interfaces (GUIs) and other objects on the internet. Before Java applets, most web pages were just pictures you couldn't do anything with. Since Adobe Flash and Microsoft Silverlight came out, Java applets aren't used as much as they used to be. With the help of the Java Virtual Machine, Java applets can run in a web browser (JVM). The JVM translates Java bytecode into instructions that the processor can understand. This lets programmes for the OS or platform run indirectly. The JVM comes with most of what you need to run bytecode, which is usually smaller than programmes written in other languages. A system can't run bytecode if it doesn't have the right JVM.

To make a Java programme, you need a Java software development kit, or SDK. An SDK usually includes a compiler, an interpreter, a documentation generator, and other tools that can be used to make a full application.

- IDE: IntelliJ

IntelliJ IDEA is a smart IDE that lets you build apps with Java and other JVM languages like Kotlin, Scala, and Groovy. With IntelliJ IDEA Ultimate's powerful integrated tools, support for JavaScript and related technologies, and advanced support for popular frameworks like Spring, Spring Boot, Jakarta EE, Micronaut, Quarkus, and Helidon, you can also build full-stack web applications. JetBrains also makes free plugins for IntelliJ IDEA that let you work with Go, Python, SQL, Ruby, and PHP, among other programming languages. IntelliJ IDEA is an Integrated Development Environment (IDE) for JVM languages that is designed to help developers work as quickly and efficiently as possible. It takes care of routine and repetitive tasks by giving you smart code completion, static code analysis, and refactorings. This lets you focus on making the good parts of the software, which is both more fun and more productive.

- Framework:Spring Boot

Java Spring Framework, also known as Spring Framework, is a popular open source enterprise-level framework for making apps that run on the Java Virtual Machine that can run on their own (JVM).
Java Spring Boot, which is also called Spring Boot, is a tool that makes it faster and easier to build web apps and microservices with Spring Framework. Most of the time, this is done in three ways:

1. Autoconfiguration
2. Being able to say what you think about how things are set up
3. The ability to create apps that can run on their own

Together, these features make up a tool that lets you set up a Spring-based app with as little configuration and setup as possible.
Dependency injection is a feature of the Spring Framework that lets objects define their own dependencies, which are then filled in by the Spring container. This lets programmers make

apps with parts that are only loosely connected to each other. These work well for distributed microservices and network applications. Spring Framework also has built-in support for tasks that most applications need to do, like data binding, type conversion, validation, handling exceptions, resource and event management, internationalisation, and more.

FEATURES OF SPRING

Autoconfiguration means that when you start an application, all of its dependencies are already set up, so you don't have to do it yourself. The Spring Framework and third-party packages are set up automatically based on the settings you choose because Java Spring Boot has autoconfiguration built in (and based on best practices, which helps avoid errors). Even though you can change these settings after initialization, Java Spring Boot's autoconfiguration makes it easy to make Spring-based apps quickly and reduces the chance of human error.

Approach with an opinion: Spring Boot uses a method called "opinionated" to add and configure starter dependencies based on your project's needs. Spring Boot chooses which packages to install and which default values to use based on what it thinks is best. You don't have to set up everything and make all of these choices by hand. During the initialization process, you can pick the dependencies your project needs from a list of Spring Starters that covers most use cases. To run Spring Boot Initializr, all you have to do is fill out a simple web form. You don't have to be able to write code.

Standalone apps: Spring Boot makes it easier for developers to make apps that can run on their own. It lets you make apps that can run even if you don't have a web server. You can do this by having your app start up with a web server like Tomcat or Netty. So, all you have to do to run your programme on any platform is press the Run command. You can turn this feature off if you want to make apps that don't have a Web server built in.

- REST API

A REST API is an API that sticks to the rules of the REST architecture. It's a way to connect things. HTTP is the way that the client and the server talk to each other. A REST API lets the client and server use HTTP methods to talk to each other. The response can also be saved in a cache, which speeds up the application. Client and server can talk to each other in any way they want. This means that each time the client and server talk, it's like starting a new conversation. When they last saw each other, they have no idea what they talked about. So, every time a client talks to the backend, it needs to send the authentication information. This tells the backend whether or not the client is allowed to access the data. The client talks to the endpoints on the back end when a REST API is set up. This keeps the code on the server completely separate from the code on the client. "CRUD" stands for "Create, Read/Get, Update, and Delete." This is the most important thing the storage is for. The CRUD operation is a set of rules that tell users how to use forms and reports to look at, search for, and change information. CRUD is based on data and uses standard HTTP action verbs. There are some important verbs in HTTP. • POST: Makes a brand-new tool GET lets you read a resource, and PUT lets you change it. • The DELETE command gets rid of something. Same Old, Same Old (CRUD) • The INSERT statement is used to create a new record when the CREATE operation is used. • READ: Based on the parameter you give it, it reads table records. • UPDATE: Running this statement changes the table. It depends on what was given as a parameter. • The DELETE command removes a row from the table. What is given as a parameter also makes a difference.

- Web Browser: Any web browser

A web browser, or just "browser," is a programme that lets you get to and look at websites. Internet Explorer, Google Chrome, Mozilla Firefox, and Apple Safari are some of the most well-known web browsers. A web browser's main job is to show HTML, which is the code that is used to make pages or "mark them up."

- Database: PostgreSQL

  PostgreSQL is a free and open source relational database for business use. It can handle both relational queries (using SQL) and non-relational queries (using JSON). Because its database management system is very stable, it is very correct, reliable, and honest. PostgreSQL is used by a large number of web, mobile, geospatial, and analytics apps as their main data store or warehouse. PostgreSQL is at version 12, which is the most recent major version. Here are some of the most common things that people do with PostgreSQL. For the LAPP stack to work well, it needs a strong database. Linux, Apache, PostgreSQL, and PHP are what LAPP stands for (or Python and Perl). PostgreSQL is mostly used as a strong back-end database that runs many dynamic websites and web apps. General purpose transaction database. PostgreSQL is the main database that both large companies and small startups use to back up their apps and products. PostgreSQL can be used with Python, Java, C#, C/C+, Ruby, JavaScript (Node.js), Perl, Go, and Tcl. There are parts of PostgreSQL that stand out. PostgreSQL has a lot of advanced features that are also found in other large-business database management systems. These features include User-defined types, Table inheritance, an advanced locking system, Foreign key referential integrity, Views, rules, subqueries, nested transactions (save points), Multi-version concurrency control (MVCC), and asynchronous replication.

3.4.2 HARDWARE SPECIFICATION

A computer with internet connectivity and with minimum 4GB Ram

3.4.3 SERVER SPECIFICATION

Using AWS t2.xlarge instances with 4 virtual CPUs, 16 GB of RAM, and backups for each module. Each instance will have 5 GB of EBS-based storage space. The database on AWS will be Postgres RDS. Amazon Web Services (AWS) is a cloud computing platform that is made up of remote computing services called web services. You can get these services in 11 places around the world. Amazon Elastic Compute Cloud and Amazon S3 may be the most important and well-known of these services. Amazon sells these products as a service because it's easier and cheaper for them to do so than for a client company to build a physical server farm. For this project, AWS is used as a SAAS (Software as a Service) Server Cloud Processing platform (SAAS). AWS has a lot of services, including all of the Apache services extensions that our system needs.

3.5 SYSTEM DESIGN

Design has been described as a multi-step process in which information requirements are used to create a representation of the data structure, programme structure, interface characteristics, and procedural details. Design is the first step in making software and keeping it up to date. Design is the starting point for everything else. It's a job where you have to make choices, usually about how things are put together. Design makes clear, well-thought-out pictures of programmes that show how the parts work together at the top level and how they work logically at the lower levels. Depending on the applications and project needs, a good design is one that lets the most efficient code be written and has the smallest possible implementation. Elements of design include functional hierarchy diagrams, screen layout diagrams, business rules tables, business process diagrams, pseudo code, and a full entity-relationship diagram with a full data dictionary. The goal of these design elements is to describe the software in enough detail that skilled programmers can make it with little extra work. The idea behind design is that a project is made up of these steps: Getting Out: Different parts - Software Architecture - Software Procedures - Data Structure - Structural Partitioning The plan for a system has two parts: – Logical design – Physical design In logical design, the person making the system makes a list of the most important parts that help it reach its goals. By looking at how the system is put together, you can see how it really works.

3.5.1 LOGICAL DESIGN

When we look at the model, we can see how all the parts of the system work together. The user can see how a system will work by looking at the analysis model. This is the first way to talk about a system in a technical way. The modelling of the analysis has to meet three main goals. To find out what needs to be done to make software. Find out what it is the user wants. To make a list of requirements that can be checked once the software has been made. The goal of a use-case diagram is to show how a system changes over time. But because this definition is too general, it doesn't say what the goal is. Since the other four types of diagrams (activity, sequence, collaboration, and state chart) do the same thing, let's look at what makes this one different. Use case diagrams look at both the inside and outside of a system to figure out what it needs. Most of these rules are about how something should look. So, when a system is looked at to see what it can do, use cases and actors are made. Use case diagrams show how things look from the outside after the first task is done. So, in short, use case diagrams can be used to do the following: Used to figure out a system's needs. Used to see how something works from the outside. Find the things that can change the system, both inside and outside of it. Show how each requirement fits in with the others.

3.6 INPUT DESIGN

Input design is the process of putting user inputs into a format that a computer can understand. During the design process, when decisions were made about how to handle input, the computer learned how to get data to process. Choosing what will happen is an important part of designing a system as a whole, which is called "input design." The part that costs the most is getting the data into the system. This is because it takes a lot of people and tools to do. The process of getting data into a computer so that it can be used is called input design. Through some kind of map support or link, data is put into the system using maps. Most errors in data processing happen when the wrong information is put in. When an input is made, the following things are taken into account.

- What data to input?

When making an input, the first step is to decide what data to put in. The "Tekact Platform" uses the JSON format to store data. This format has the same benefits as XML when it comes to exchanging data in a mixed environment. It says what it is in JSON. Applications that don't know what type of data to expect can sometimes figure out what JSON strings mean by looking at their syntax and hierarchical structure. JSON is just plain text. Because of this, it can be sent safely between platforms and operating systems that have trouble sharing other kinds of documents. Simple editors can also show JSON as text, and it can be changed quickly. JSON is not very big. A JSON string is about two-thirds as long as the same data in XML, on average. It's easy to learn, read, and understand how to use JSON.

The input design has two goals:

1) to keep track of how much,

2) to make sure that things don't take too long.

3. Avoiding errors.

4. Avoiding extra steps

In this "Tekact Platform," all of the text boxes are checked. If any required field is left blank, the message will appear.

The design of the input is made up of the following parts:

• The inputs are set up so that the user and the system can work together as well as possible.

• Steps have been taken to make it easier for the user to do their job.

• When there are less steps, the process is easier.

3.7 OUTPUT DESIGN

"Output" refers to the results and information that the system gives you. Here, you choose what information to show, how to lay it out, which output medium to use, how to show the information in a standard format, and how to get it to the people you want to see it. Headings for columns and page numbers are set up and placed in a certain way. Making the output is a big part of giving the user the format they want. The main purpose of the output is to share information, so how it is laid out and designed is important. Information needs to be carefully organised based on what the user wants. Standards for printed output say that each Output should have a name or title, that a layout example should be given, and that the process for making sure that the output data is correct should be explained.

When choosing output devices, think about how well they work with the system, how fast they need to be, and how well they need to print. Care is taken when making the output form to make sure it has the right name and words, is easy to read and use, has a good layout, and has clear instructions. It should be easy to understand if the labels are clear and the form is well made. A group needs to be able to control its structure from one place for it to work well. The most important and direct source of information for the user is what the computer puts out. Output design is the process of making reports about what users want and what outputs they should get. If the results are useful and easy to understand, the system should get along better with the user and help them make decisions. Since management uses the reports to make decisions and come to conclusions, they must be carefully written and the details must be clear and simple. So, here are some things to keep in mind when making output.•

Determine what information to present as follows.

- Make sure that the way information is shown makes sense.
- Figure out how to split the income to get the money you need.
- Depending on the type of output and how it will be used in the future, it can be shown on the monitor for immediate use or printed out.

Smart and effective output design should make it easier for the user to figure out how to use the system and what to do. That is, this makes the system easy to use so that it can be shown or printed in any way the user wants. A good output meets the needs of the end user and shows the information in a way that is clear, easy to read, and looks good. To choose the right way to present and the right format, you need to think about a number of things, like who will get the output, how it will be received, etc.

CHAPTER 4

RESULT AND DISCUSSION

A lot of businesses now use the latest technology. When services and APIs can be used over and over, it's easier to get new products on the market faster. This project based on microservices shows how different services can work on their own by using the best microservices patterns to improve scalability, performance, and reliability. The EventPub project's goals are to provide APIs for making apps, ways to put apps together using these APIs, easy integration through the use of REST APIs, workflow management, and a working app.

This product is built on a microservices architecture, which means that each microservice solves problems that are unique to a certain domain. Users will put together these "microservices" to make apps. The goal of this product is to give users APIs that they can use to make Apps that work with different domains. EventPub is a service that can be used by more than one person at once. It uses a software engineering method that focuses on breaking an app up into modules that each do one thing well and have clear interfaces. Adaptive business process management is already working to make these kinds of decisions work well together. Also, breaking up big applications into separate microservices gives agile teams the next level of independence, which helps agile methods grow. Each team could work on a different microservice and write user stories that only apply to that microservice.

4.1 TESTING METHODS

There are lots of ways to test something. Testing is used in this project to find errors in the requirements, design, or code.

4.1.1 UNIT TESTING WITH MOCKITO

Unit testing is a part of how software is made. During this step, "units," which are the smallest parts of an app that can be tested on their own, are checked to make sure they work correctly. Unit testing is an important part of the development process because, if done right, it can help find bugs in the

code early on, when they might be harder to find.
Test-driven development (TDD) is a way to build a product carefully by testing it over and over and making changes. Unit testing is part of TDD. This type of testing is also the first level of testing software. It is done before other testing methods, like integration testing. Unit tests are usually run by themselves to make sure that a unit doesn't depend on any code or functions from outside the unit. Testing can be done by hand, but most of the time it is done automatically. Mockito is a well-known open source framework for testing software by simulating objects. Using Mockito makes it much easier to make tests for classes that depend on things outside of the programme. A fake copy of a class or interface is called a "mock object." It lets you choose what will happen when a method is called. Most of the time, tests show how they track how people use the system.

The main parts of a unit test are the plan, the cases and scripts, and the actual unit test. The first step is to get ready for the unit test and review it. The next step is to create test cases and scripts, and then the code needs to be tested. For test-driven development to work, developers must first create unit tests that fail. Then they write code and change the app until the test passes. Most of the time, TDD results in code that is clear and easy to understand.

4.1.2 INTEGRATION TESTING

Integration testing is the next step in the testing process. During this testing, groups of software units or parts are checked at the same time. At the integration testing level, the goal is to find bugs that happen when two or more parts or units that have been tested separately work together.

Integration testing looks at how well all the modules work together. During unit testing, each module is tested on its own. The software is made up of different software modules that were coded by different coders or programmers. Integration testing makes sure that all of the modules can talk to each other in the right way.

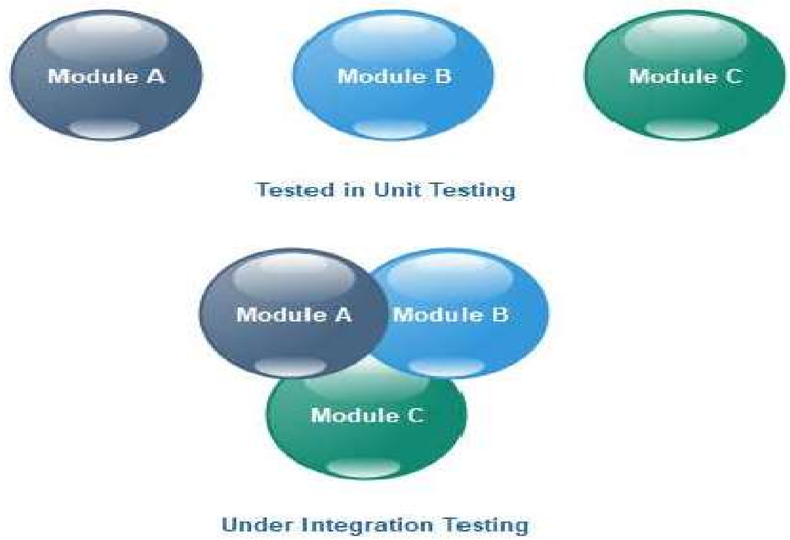


Figure 4.1.2: Testing Modules

4.2 OUTPUT SCREENS AND RESULTS

1. ORGANIZATION CONTROLLER

As an Admin, It should be able to create, read, update, and (soft delete) an organisation. You should be able to access all of the organisations in the database through an API.

1.1 Give the name of the group in charge

As a Super Admin, It should be able to add, change, soft-delete, and read addresses and contacts for an Organization. There should be an API that lets you get all the contacts and addresses in the database.

1.2 Talk to the person in charge of the group.

As a Super Admin, It should be able to add, change, soft-delete, and read addresses and contacts for an Organization. There should be an API that lets you get all the contacts and addresses in the database.

2. DOCUMENT TEXT CONTROLLER

CHAPTER 5

CONCLUSION

The project wanted to give APIs for making apps and ways to put apps together using these APIs. It also wanted to make integration easy by using REST APIs, manage workflow, and make an app that works. The goal of this product is to give users APIs they can use to make Apps that work with different domains. A multi-tenant system. It uses a software engineering method that focuses on breaking an app into modules that each do one thing well and have clear interfaces. This is the first time that the system's need has been tried to figure out. There has been a lot of research done to make sure it is easy for people to use and meets their needs.

This system was designed to look nice and be easy to use, even for people who don't know much about computers. This method works well. This product is built on a microservices architecture, which means that each microservice solves problems that are specific to a certain domain. These "microservices" will be put together by users to make apps. Many businesses now use the most advanced technology. Getting new products on the market faster is easier when services and APIs can be used over and over. This microservices-based project shows how different services can work on their own by using the best microservices patterns to improve scalability, performance, and reliability.

5.1 FUTURE ENHANCEMENT

The way the system is set up makes it easy to add new modules. If the system is rebuilt, there are more ways to use it. With this technology's advanced features in mind, the system was made to be as adaptable and easy to use as possible.

When REST APIs, workflow management, and a working app are used, integration is easy. This makes it easy for people to know when things take place. At this point, some of the work on the back end of the project is done. You can use different "microservices" with this app. User will put together these "microservices" to make apps. Also, not all of the modules and the user interface are done yet.

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 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 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 and Menaka Pushpa Arthu,“Survey on Template 40 Engines in Java”,School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus,2020 India 600 127

[8].Lei, Kai, Yining 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

SCREENSHOT

ROLE CONTROLLER

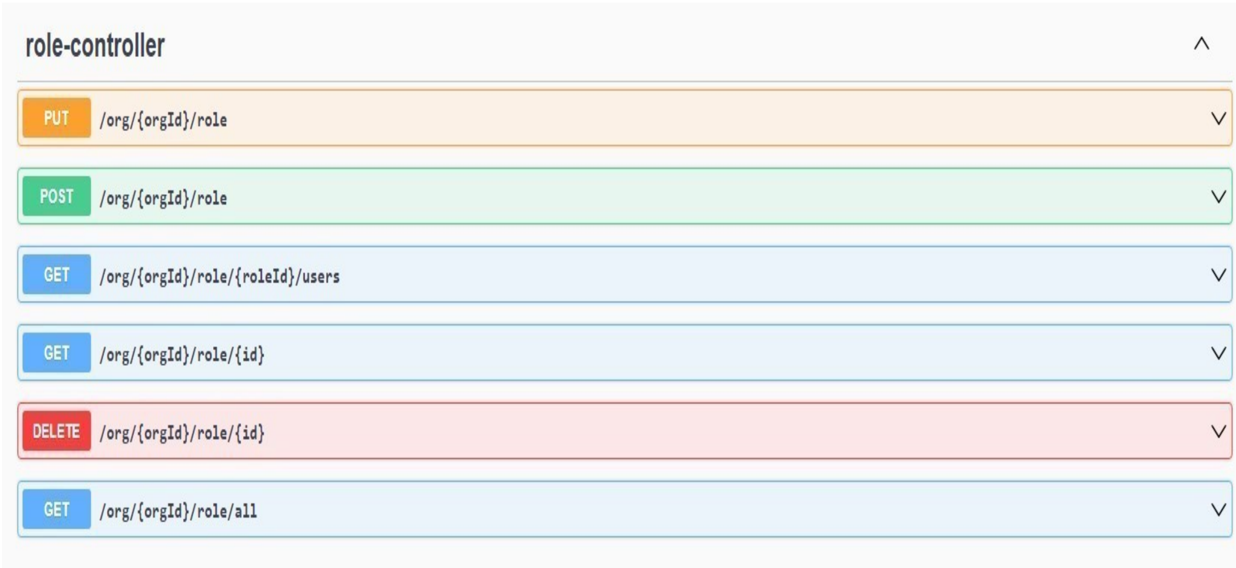


Figure A.1 Role controller

FEEDBACK CONTROLLER

Figure A.2 Feedback controller

TESTCASES

ROLES

Element	Class, %	Method, %	Line, %
com.tekact.platform.admin.service	84% (11/13)	48% (56/115)	41% (307/734)
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)
ConfigValueServiceImpl	100% (1/1)	0% (0/2)	12% (1/8)
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)
RoleServiceImpl	100% (1/1)	100% (7/7)	90% (29/32)
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 A.3 Role Test coverage

FEEDBACK

Coverage: FeedbackServiceTest

Element	Class, %	Method, %	Line, %
com.tekact.platform.admin.service	69% (9/13)	10% (12/113)	14% (92/649)
AppService	0% (0/1)	0% (0/1)	0% (0/6)
UserService	0% (0/1)	0% (0/1)	0% (0/6)
TenantService	0% (0/1)	0% (0/1)	0% (0/6)
OrganizationService	0% (0/1)	0% (0/3)	0% (0/18)
OrganizationServiceImpl	100% (1/1)	0% (0/23)	0% (1/124)
UserServiceImpl	100% (1/1)	0% (0/21)	0% (1/127)
TenantServiceImpl	100% (1/1)	0% (0/24)	0% (1/141)
AppServiceImpl	100% (1/1)	0% (0/8)	2% (1/43)
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)
FeedbackServiceImpl	100% (1/1)	100% (12/12)	96% (84/87)
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)

Figure A.4 Feedback Test coverage