International Journal of Precious Engineering Research and Applications (IJPERA)
www.ijpera.com ISSN: 2456-2734,
Volume 7, Issue 2 (Jul. – Aug.) 2022), PP 10-14

Efficient Note Taking App Using Web API and JWT Token

HEAD OF THE DEPARTMENT COMPUTER SCIENCE AND ENGINEERING
,SRIPERUMBUDUR,CHENNAI.

Dr.M.NAVANEETHA KRISHNAN, M.E., Ph.D.
FINALYEAR COMPUTER SCIENCE AND ENGINEERING
SRIPERUMBUDUR,CHENNAI.
Ms.A.Priyanshu Singh

ABSTRACT:
The scope of this project is to develop a real time, cross-platform, scalable and reliable note taking Application. Note Keep is a note-taking service included as part of the free, web-based Docs Editors suite Implemented by Web API's. Note Keep can be available as a web application as well as mobile app for Android. The app offers a variety of tools for taking notes, including: texts, lists and images. Text from images can be extracted using optical character recognition. The interface allows for a single-column view or a multi-column view. Notes can be color-coded and labels can be applied for organization. It has functionality to pin notes and to collaborate on notes with other Keep users in real-time.
These notes are stored in tables in the databases with the User interface and web Application programming interfaces(API). Web API's are used as a interface for the communication between two or more different applications.These API's are build with interpretation of action methods that reveals what to be returned after execution of the API.
Web API have their end points and routes to be specific that are used for API testing with Postman and Swagger.I have implemented concept of code first approach for the creation of a full Stack project.This project will help a person to login and add notes for their preferences.

I. Introduction

The aim of our project is to develop a real time, cross-platform, scalable and reliable note taking Application using latest web technologies. The note taking is done along with different components. The data sharing from UI components using parent to child components along with one way and two-way data binding. Moreover, it considers the information from the database, such as note id and its content with user authorization which improves the accuracy. The application provides sophisticated user interface with transparent functionalities.

II. Literature Survey

ZheCao,GinesHidalgo,TomasSimon[1] proposed An ECMAScript application programming interface (API)forcontent-to-runtime-servicescommunicationisdescribedinthisstandard. It is based on a current industry practice called "CMI-computermanagedinstruction."ThisAPIenablesthecommunicationofinformationbetween content and a runtime service (RTS) typically provided by alearning management system (LMS) via common API services using theECMAScript language. The purpose of this standard is to build consensusaround, resolve ambiguities, and correct defects in existing specificationsfor an ECMA Script API for exchanging data between learning-relatedcontent andanLMS.

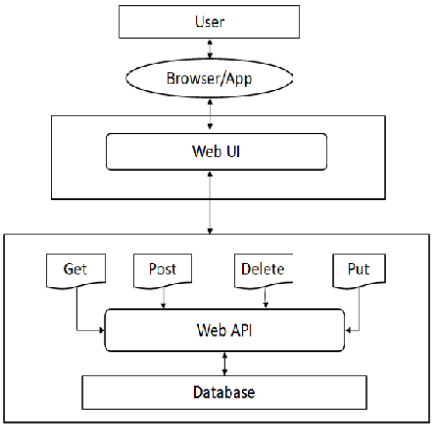
SandraCasas,DianaCruz,GracielaVidal,MarcelaConstanzo[2] proposedTheOpenAPI or Swagger specification hasbecomethereference choice for describing web APIs (REST, RESTful, etc.). That iswhy this specification has been and is currently the center of many researcharticles. Several activities, tasks, and problems associated with softwaredevelopment can be improved, simplified, solved and/or automated withthis specification. The purpose of this study is to synthesize the uses andapplicationsoftheOpenAPI/Swaggerspecification.Thedevelopmentactivitiesinwhichithasbeenuseful,thesoftwareartifacts(results)generated from it, and which domains have been covered, among others,are identified. The literature review that we analyzed had a total of 47articles published between 2011 and 2020. The search corresponded withthese parameters: a) mention of the development activities in which thespecificationhasbeenuseful,b)inclusionofthesoftwareartifacts(results)thatitgeneratesandtheircorrespondingvalidation,andc)referencetothe domains covered. Results show that the OpenAPI/Swagger

specificationhasbenefitedvarioussoftwaredevelopmentactivities,beingtheimprovementof thedocumentation theonethatpredominates(43%).Similarly, 77% of the articles provide solutions to the requirements and/orproblems of the web API consumers. The automation of tasks, based ondiversetools,constitutesthemostpredominantresultin68%ofthemanuscripts.Besides,40%oftheworksconsultedcoveraspecificdomain(IoT,Cloud,etc.).

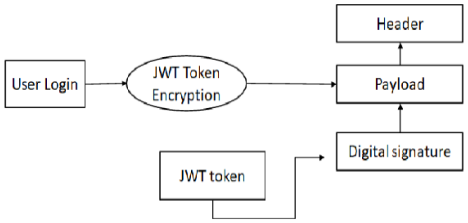
SalmanAhmed, QamarMahmood[3] proposed the first single-network approach for 2D whole-body (body, face, hand and foot) pose estimation, capable of detecting anarbitrary number of people from in-the-wild images. Unlike the originalOpenPose,ournewmethoddoesnotneedtorunanadditionalnetworkforeach hand and face candidate, making it substantially faster for multi-person scenarios. This provides Better tracking speed and accuracy thanoriginalopenpose.Itdoesnotusethedataforanypurpose,cando trackingbut nothingelse.

III. System Design

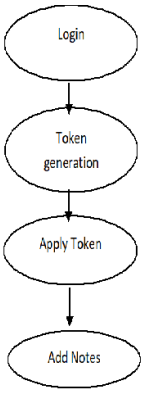
The proposed system solves the drawbacks faced by the existing system while using advanced technologies to ensure future sustainability of the software. The client-side application is built to be cross-platform. It is deployed in website, mobile app and desktop app for seamless interoperability of the system.



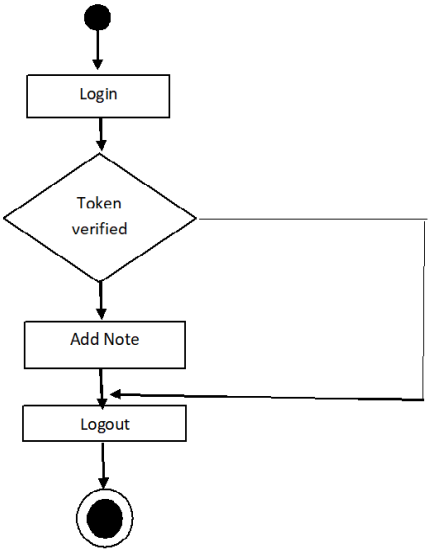
usage of the resources in the server. The note taking system is implemented using MS SQL and Web APIs, which provide low latency communication between client and server, while also implementing efficient caching policies.



The client-side application is developed for following entities, namely User, Notes and Collaborators. The User has complete control of the data flow between server and client.



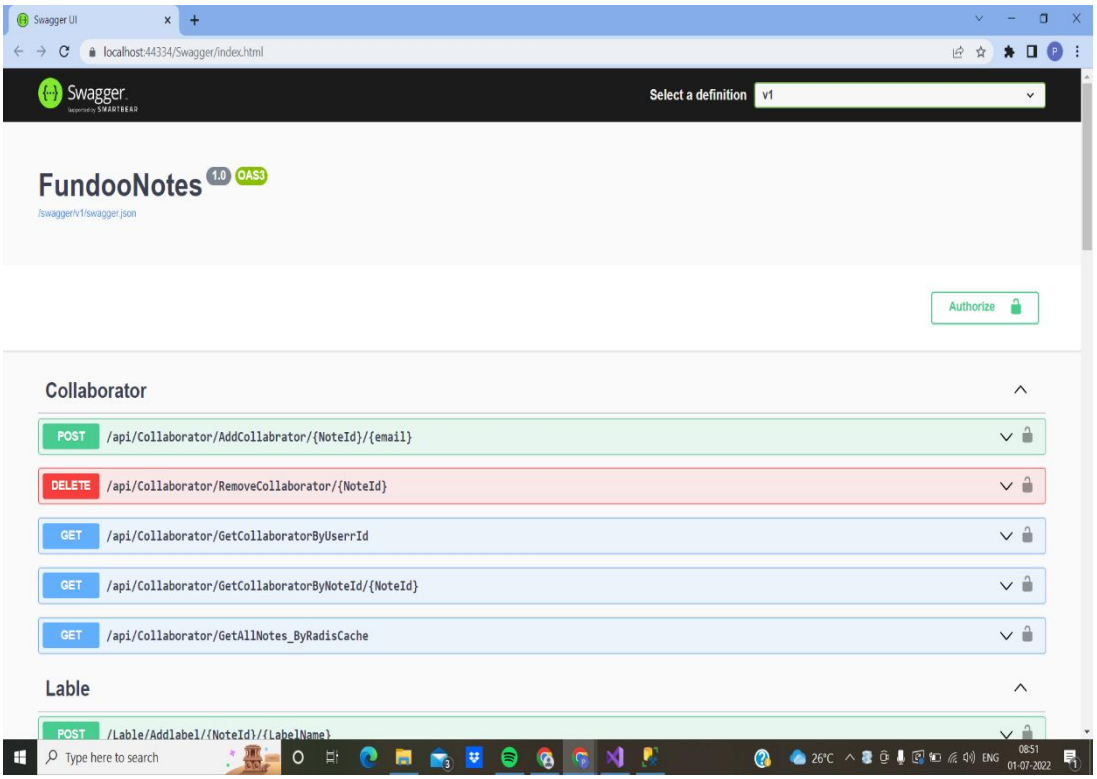
The JWT token is used by the system to authorize the User of the Note keep. The User can add the notes only after the authentication.



The caching system implemented using Redis provides low latency and high frequency updates to the system.The overall infrastructure can be scaled both vertically and horizontally without affecting the existing infrastructure.
The User can track the notes of other users of the organization and can read and update the notes as per requirements.
The entire application can be scaled vertically and horizontally without affecting the existing system.

IV. Implementations

The system is composed of two major components, namely the server and client. The server is hosted in swagger instance and virtualized using User interface. The MSSQL database is used as server for storing data. The server-side application is built using Angular and the API system is built using MS SQL. The caching system is implemented using Redis, which is also virtualized using container.



The authentication system is built using JSON web token. The JWT is developed using the following notation:

$token = $header + '.' + $payload + '.' + $signature; The header is created using the HMAC256 algorithm with following format:

{
"alg":"HS256",
"type": "jwt"
}

The payload comprises client ID, client name who gives the token, the expiry date of the token, and access right to the token. Following is the JSON format for payload:

{
"iss": "api.tracker.tech",
"exp": 1426420800,
"access":
[{
"read-order"
"read-outlet"
....
}],
"client_id": 1,
"client_name": "Jim"
}

The JWT is signed using the signature which is generated as follows: $signature= HMAC($header + $payload, $key); The client application is developed using React.js. The state management is handled by Redux.js. The client can be connected to the server using CORS policy. The client application will be deployed to localhost of angular. The User application can perform CRUD operations for Login, Notes, label and collaborators. The User will use the user application and be able to login to the system. The User will be able to share the notes to the other users. The user will use the MSMQ server and be able to resetpassword to the system. The user will be able to receive mail notifications after forgetting password.

V. Conclusions and Future Enhancement

In our paper, we focused to eliminate the need for dedicated note taking requirements to support multiple users. This led to being cost-effective and providing better services to the organization than the existing system. Since our project is cross-platform compatible, we can update the project to accommodate any future requirements. The project was built for the majority of users who use Android smartphones as their primary device. However, in future, we expect to cater the needs of iOS users as well. Moreover, there is scope for building a dedicated note taking device using Arduino Uno and a GSM Module.

References

[1]. ZheCao,GinesHidalgo,TomasSimon "ECMAScriptApplication - ProgrammingInterfaceforContenttoRuntimeServicesCommunication"
[2]. SandraCasas,DianaCruz,GracielaVidal,MarcelaConstanzo "Uses and applications of the OpenAPI/Swagger specification"
[3]. SalmanAhmed, QamarMahmood "AnauthenticationbasedschemeforapplicationsusingJSONwebtoken"