A project report on

ONELINK

(Restful API to fetch and display data)

Submitted in partial fulfilment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

LAKSHAY MALHOTRA (16BCE2273)

VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

May,2020

ONELINK
(Restful API to fetch and display data)

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

LAKSHAY MALHOTRA (16BCE2273)

School of Computer Science and Engineering

May,2020

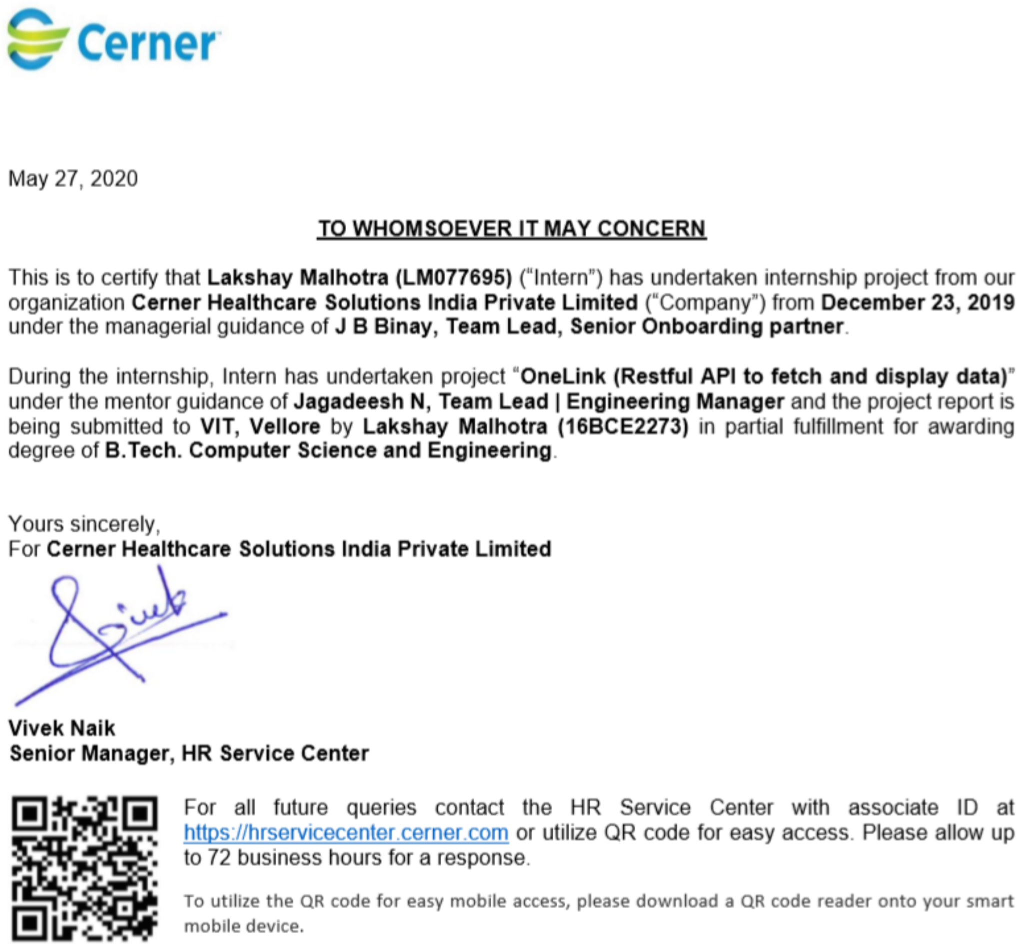
DECLARATION

I hereby declare that the thesis entitled "**OneLink**" submitted by me, for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a record of bonafide work carried out by me under the supervision of **Mr. N Jagadeesh and Mr. JB Binay**, Cerner Healthcare Solutions Pvt. Ltd.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore Lakshay Malhotra (16BCE2273)

Date: 20th May,2020 **Signature of the Candidate**



Cerner

May 27, 2020

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Lakshay Malhotra (LM077695)** ("Intern") has undertaken internship project from our organization **Cerner Healthcare Solutions India Private Limited** ("Company") from **December 23, 2019** under the managerial guidance of **J B Binay, Team Lead, Senior Onboarding partner**.

During the internship, Intern has undertaken project "**OneLink (Restful API to fetch and display data)**" under the mentor guidance of **Jagadeesh N, Team Lead | Engineering Manager** and the project report is being submitted to **VIT, Vellore** by **Lakshay Malhotra (16BCE2273)** in partial fulfillment for awarding degree of **B.Tech. Computer Science and Engineering**.

Yours sincerely,
For **Cerner Healthcare Solutions India Private Limited**

Vivek Naik
Senior Manager, HR Service Center

For all future queries contact the HR Service Center with associate ID at https://hrservicecenter.cerner.com or utilize QR code for easy access. Please allow up to 72 business hours for a response.

To utilize the QR code for easy mobile access, please download a QR code reader onto your smart mobile device.

ABSTRACT

Currently in many IT companies' information and data is split across multiple platforms that employees access on a daily basis. This data is in the form of websites and platforms that software engineers and team leaders need to access. The problem in status quo is that it becomes a very tedious and redundant task for employees to search to multiple websites and platforms to collect and corelate data from them for their use. These platforms include Jira, Sharepoint, RQM, Jenkins etc. All of them contain data and details of a particular project and or code that employees use on a daily basis. The idea is to use Restful API to access data from these platforms and accommodate it on a common platform the allows the user to access all data on one common platform and it helps the first time users and new employees at the firm to be familiar with hierarchy of the department with respect to the project and sub-groups. This concept therefore acts like beneficial tool for both daily users and new employees. Using Restful API makes the entire process much easier and simpler.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to **Mr. N Jagadeesh** Team Lead at Cerner Healthcare Solutions Pvt. Ltd, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Web Application Development.

I would like to express my gratitude to **Dr. G. Viswanathan, Dr. Anand A Samuel**, **Dr. S. Narayanan** and **Dr. Saravanan R,** School Of Computer Science and Engineering, for providing me with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to **Dr. Santhi V,** Head of the Department (HoD), all teaching staff and members working as limbs of our university for their not self centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore　　　　Lakshay Malhotra

Date: 20th May,2020　　　　**Name of the studemt**

CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

API	**A**pplication **P**rogramming Interface
CSS	**C**ascading **S**tyle **S**heets
JSON	**J**ava **S**cript **O**bject **N**otation
HTML	**H**yper **T**ext **M**arkup **L**anguage
HTTP	**H**ypertext **T**ransfer **P**rotocol
MVC	**M**odel **V**iew **C**ontroller
OC	**O**peration **C**enter
iNOC	**I**nfrastructure **N**etwork **O**peration **C**enter
IRC	**I**ncident **R**esponse **C**enter
MEF	**M**onitoring **E**vent **F**abric

Chapter 1

Introduction

Currently in many IT companies' information and data is split across multiple platforms that employees access on a daily basis. This data is in the form of websites and platforms that software engineers and team leaders need to access. The problem in status quo is that it becomes a very tedious and redundant task for employees to search to multiple websites and platforms to collect and corelate data from them for their use. These platforms include Jira, Sharepoint, RQM, Jenkins etc. All of them contain data and details of a particular project and or code that employees use on a daily basis. The idea is to use Restful API to access data from these platforms and accommodate it on a common platform the allows the user to access all data on one common platform and it helps the first time users and new employees at the firm to be familiar with hierarchy of the department with respect to the project and sub-groups.

Cerner prides itself on 99.99% uptime. As part of the billing team employees visit sites and platforms such Jenkins, Jira, RQM and sharepoint. So the idea was to use Restful API to remotely access all of these platforms to get data and display it on the OneLink platform for users to get a unified screen for multiple functions and sites with varied functionalities.

1.1 OVERVIEW

The primary objective of the proposed system is to overcome one or more problems of the existing system of a lack of unified system for various daily use important platforms and websites with varied functionalities.

1.1.1 LACK OF A COMMON PLATFORM

The lack of a common platform leads to a tedious task of new users and current employees to collect data from various platforms and use it daily.

1.1.2 OBSOLETE REMOTE ACCESS SYSTEM

The current system used utilizes the flat file system to store all the data. Along with Jira and Jenkins for jobs and tasks doesn't provide users with the ability to remotely access jiras or even trigger jobs for a particular department and/or System.

1.1.3 UNAPPEALING ALTERNATIVES

The alternative to the suggested solution is the status quo which is multiple access for these various tasks without and unification factor in the them, which cause a reduction in efficiency and performance.

1.2 PROBLEM FORMULATION

With regard to the above issues and concerns, the main focus is to formulate a system that has an appealing user interface with proper user authentication so as to reduce the time taken to access data and moreover provide a system for unification of daily needs and tasks.

Thereby providing a portal that is consistent, efficient, scalable and reliable for all the functions required to be performed by the associates not only saving time and resources but also increasing performance and reducing complexity.

1.3 OBJECTIVES

Looking into the various problems that are currently being faced with respect to the manual traversing of various websites and softwares to accumulate data, the intent is to create a system that simplifies the whole process by reducing complexity and time by building an integrated platform that will enable all the operation center lanes namely, storage, virtualization, networking and skybox to create, access and maintain the log records in the simplest way possible. The objectives are as follows:

- To provide easy workflow handling and remote access.
- To implement a simplified and updated data model design.
- To make it API enabled so that it is easily pluggable and can therefore extend functionalities and provide extensibility irrespective of the frontend technology.

1.4 LIMITATIONS

Every project comes with its own set of advantages and disadvantages that one should be aware of. Although every issue and problem is analysed and discussed and a resultant system is planned with all the solutions integrated into it, there are some limitations that need to be known and understood without which the proposed system will not be able to function and operate at the optimum level.

Since this is an industry project there are certain constraints to the technology that is used. The usage of some third-party libraries are disallowed due to compliance issue. The Rest API used for Remedy Integration is hardcoded so if the API endpoint changes corresponding changes need to be made in the application.

Chapter 2

Methodology

A RESTful API -- also referred to as a RESTful web service or REST API -- is based on representational state transfer, an architectural style and approach to communications often used in web services development.

A RESTful API breaks down a transaction to create a series of small modules. Each module addresses a particular underlying part of the transaction. This modularity provides developers with a lot of flexibility, but it can be challenging for developers to design their REST API from scratch. Currently, several companies provide models for developers to use; the models provided by Amazon S3, Cloud Data Management Interface and OpenStackSwift are the most popular.

A RESTful API uses existing HTTP methodologies defined by the RFC 2616 protocol. They use GET to retrieve a resource; PUT to change the state of or update a resource, which can be an object, file or block; POST to create that resource; and DELETE to remove it.

With REST, networked components are a resource the user requests access to – a black box whose implementation details are unclear. All calls are stateless; nothing can be retained by the RESTful service between executions. Because the calls are stateless, REST is useful in cloud applications. Stateless components can be freely redeployed if something fails, and they can scale to accommodate load changes.

This is because any request can be directed to any instance of a component; there can be nothing saved that has to be remembered by the next transaction. That makes REST preferable for web use, but the RESTful model is also helpful in cloud services because binding to a service through an API is a matter of controlling how the URL is decoded. Cloud computing and microservices are almost certain to make RESTful API design the rule in the future.

Chapter 3

Implementation

After formulating, identifying and doing thorough research on the problem, the actual work project is taken up. We identified the problem in the existing manual system access to various types and fields of data. The proposed system is an application that can be used with ease to log the various events that continuously flow into the system. This system will create a common platform in which data will be stored and maintained, this data will be taken from platforms like jira, Jenkins,sharepoint,RQM etc. The platform will have a search engine which searches for a particular keyword in all the above-mentioned platforms and give accumulated results from all of them. There will be an admin part and a user part of the system which uses LDAP credentials. Rest API is used to get the data from the platforms and display when required. Data can be pin pointed, stored, displayed and worked upon all from one platform – ONELINK.

The implementation is split into further parts.

- Methodology of the study
- Experimental and/or analytical work completed in the project
- Modeling, Analysis and Design
- Prototype Development and Product Implementation

4.1 METHODOLOGY FOR THE STUDY

As known by everyone planning is the most important and time taking step of any project in the software industry. Planning was given a high priority so that nothing goes wrong further down the road. A very realistic plan was made with decent deadlines to cover certain topics each day. This helped in understanding the concepts better and trying to find solutions on how to execute the task of building the solution. This planning phase can be further divided into the following phases.

4.1.1 REQUIREMENT GATHERING

4.1.1.1 FUNCTIONAL REQUIREMENTS

Several functional requirements were gathered from the users through multiple interactive sessions. After understanding the problem statement, a solution was identified which incorporated all the features requested by the users. The functional requirements identified are:

- User Authentication to be implemented using LDAP protocol.
- Sending alerts to the concerned associates who needs to take further actions on the incidents.
- Interactive user interface which is tailored to users from different lanes.
- Simplified and updated Data Model Design.
- Easy Workflow Handling and Remote access.
- Gathering APIs and learning their implementation in java code
- Gathering data from the platforms and getting auth from them.

Platform constraint: The associate handling the request could be using any operating system. The tool developed must work on all platforms the same. The backend should not be affected by the system on which it is run. The tool is developed on Spring Frame- work which executes the same on every operating system.

Security: This tool is hosted on a secure server. LDAP user authentication is enabled in this software, this minimizes the risk of unauthorized personnel accessing the data.

Availability: This tool should always be available, or else there will be hundreds of incidents that are not monitored. This will lead to widespread outages in the infrastructure. This standalone tool is available at all time.

Reliability: All the associates using this tool should be able to rely on it completely or else the tool would be of no use. This tool is developed with the organization rules of performing a task and it will always provide information adhering to the norms. Every data provided by the tool is well defined and verified.

Scalability: This system was built using microser vice architecture. This means that the different modules of the project was built as a Rest API. This increases the extensibility of the software. The was utmost care taken to decrease the interdependence between the different modules.

4.2 EXPERIMENTAL WORK COMPLETED IN THE PROJECT

There was a need to integrate a web-framework with a database and have a separation between the business logic and the frontend. The Spring framework is excellent for creating models for the database required to store the data. It also fits harmoniously with Bootstrap for the frontend. The data which comes in, needs to be extracted using the APIs continuously,APIs play a vital role in executing the task of fetching the required data with the help of HTTP methods. This project required the knowledge of frontend, backend and database technologies.

Before proceeding with the implementation, a thorough research was done regarding the technologies to adapt. Knowledge of Spring Boot and Spring MCV(Model-View-Controller) Model for the backend, hibernate for object relational mapping and PostgreSQL was chosen as the database. After the research and learning phase, two POCs were created. The following two sections contain a detailed overview of the POCs presented.

4.2.1 FIRST POC

This task was assigned as a way to learn about the various technologies that were going to be used in the project. This task was essential in understanding how the different technologies fit together. A spring boot project was created where in the user interface was created using HTML and CSS as the base with MongoDB as the database.

The business logic to perform the search engine operations were implemented using Java complete with the spring boot annotations. All the dependencies for the database, HTML, CSS and SpringBoot Web were added to the pom.xml file. This POC was also created by enabling API endpoints which can be called separately. The API was tested using Postman. Postman is an application that is used to check whether we are hitting the API end point. After successful creation and linking of frontend, backend and the database all the combinations of CRUD operations were performed from the front end. Hence concluding this experimental work it was made clear that I'm comfortable with all the technologies to be used further in the industry project.

4.2.2 SECOND POC - GENERIC TOKEN GENERATION API

A REST API built on Spring Boot has been developed to generate the access token. The Client Credentials grant type is one of the types which are used to obtain the access to- ken. The Client Credentials grant type is used by users to mainly access resources about themselves rather than to access another user's resources. This API sends a POST re- quest to the Cerebro endpoint with the client ID, client secret and the grant type as the payload to generate the token. When this API endpoint is hit the token is regenerated only if the previous token has expired, thus eliminating the need for unnecessary API calls.

OAuth (Open Authorization) is an open standard that is used for access granting and or delegation protocol. It used as a way for users to grant applications access to their information on other applications/websites without using passwords. Designed in particular to work with Hypertext Transfer Protocol (HTTP). The goal of this application is to generate an access token using OAuth2.0 authentication which hits the Cerner to- ken generating endpoint to give the latest access token. The token will be regenerated after the expiration time. All the applications which require the access token can hit this application endpoint to generate the access token instead of hitting the Cerner endpoint and starting a new session each time, they need the token. After the application has been deployed all the applications that are currently hitting the Cerner endpoint directly should make the API call to this REST API.

4.3 MODELING, ANALYSIS & DESIGN

4.3.1 OVERVIEW

Currently in many IT companies' information and data is split across multiple platforms that employees access on a daily basis. This data is in the form of websites and platforms that software engineers and team leaders need to access. The problem in status quo is that it becomes a very tedious and redundant task for employees to search to multiple websites and platforms to collect and corelate data from them for their use. These platforms include Jira, Sharepoint, RQM, Jenkins etc. All of them contain data and details of a particular project and or code that employees use on a daily basis. The idea is to use Restful API to access data from these platforms and accommodate it on a common platform the allows the user to access all data on one common platform and it helps the first time users and new employees at the firm to be familiar with hierarchy of the department with respect to the project and sub-groups.

Cerner prides itself on 99.99% uptime. As part of the billing team employees visit sites and platforms such Jenkins, Jira, RQM and sharepoint. So the idea was to use Restful API to remotely access all of these platforms to get data and display it on the OneLink platform for users to get a unified screen for multiple functions and sites with varied functionalities.

OneLink gives users the ability to use a search engine to type in keywords or details from their Jira, code project etc and get all the details about it from multiple platforms. The search engine comes with a hierarchical structure of the department and all the projects within it and uses that to help users navigate to pin point and narrow their search.

The platform is used for navigation, storage, unified collection of data and trigger jobs remotely which reduces the effort and saves valuable time of the employee.

4.3.2 ARCHITECTURE

Architecture significantly affects the project's success as it's the project's base. A number of factors affecting the project depend on the architecture which is chosen. This includes attributes like scalability, resilience, flexibility, ease of deployment, and the size of the development team. Microservices architecture focuses on designing software as a set of small, loosely-coupled services. This is different from the traditional monolithic solutions that are built as an entity — composing everything in one piece with interdependent components.

Rest API functionality combines with java code with MongoBD database and HTML CSS as the front end to create a single platform. The Structure is similar to a tree in which filters are applied as the user searches and adds keywords. These filters follow a tree hierarchy to scout and display data. LDAP auth on top followed by, Search Engine that uses Rest API and filters added in java code to sort out through json data.

The rapid, frequent and reliable delivery of large, complex applications is enabled by the microservice architecture. This also helps in the evolution of an organization's technology stack. In brief, the microservice architectural style is an approach to developing and implementing a single application as a collection of small services, each of them runs in its own process and communicates with lightweight mechanisms. This is mostly an HTTP resource API. These services are built around the business objectives. The independent deployment is also a feature of microservice architecture. The amount of centralized management required by these services are minimal. The microservices may be implemented via different programming languages and can utilize different data storage technologies.

Flow Diagram

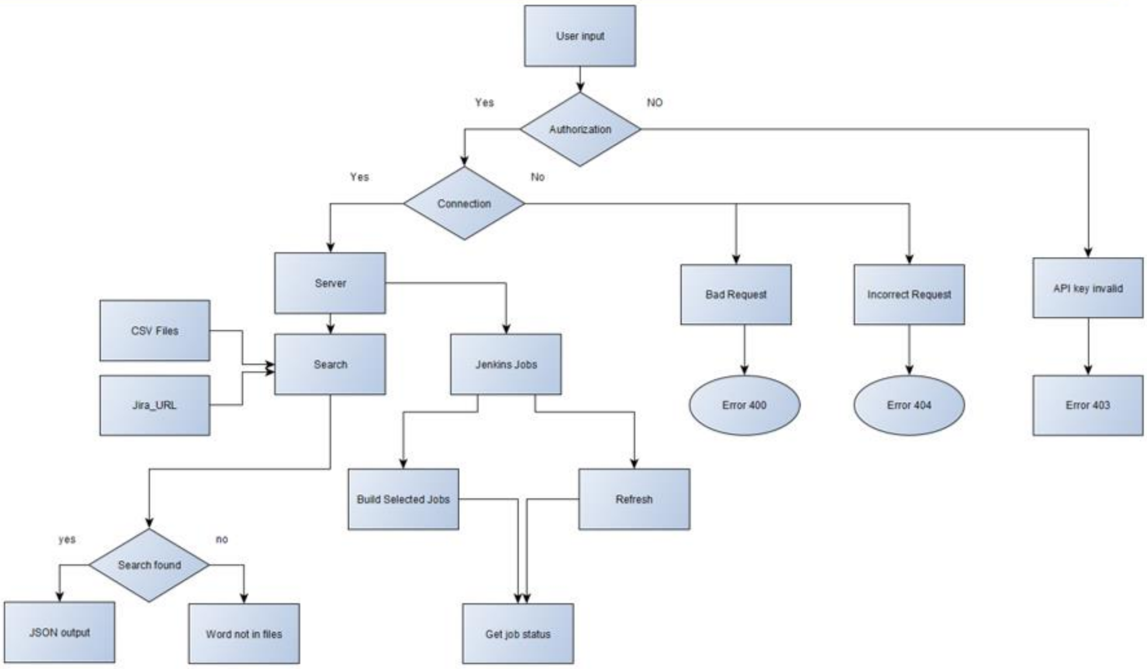


Figure: Flow diagram

4.3.3 USECASES

A use case diagram is a graphical representation in its simplest form of a user’s interaction with the system and depicting the specifications and details of a use case. A use case diagram portrays the various types of users of the target system and the different ways that they interact with the system.

There are various use cases that come the large potential and need of this project.

Users use OneLink to gain access to particular fields of data that otherwise take them excess amount of time to search, accumulate and then use. So users having an exact idea what to search use this service along with users who wish to navigate through the project.

The idea there for navigators is to provide a hierarchy for new users to explore the system and the projects done or ongoing with the company.

Another user system interaction is when users use this to trigger Jenkins jobs and view the jobs and their results and details in a table displayed upon search in oneLink.

4.4 PROTOTYPE & TESTING

Testing is a crucial element of the project this helps in ensuring the quality and effectiveness of the proposed system in meeting its objectives and goals. Testing should be done at various stages of the software development process. This should be kept in mind during the system designing and implementation process. Testing is done with the objective of developing a transparent, flexible and secured system that can withstand all manner of attacks from intruders, also to make sure that a quality product can be deliv- ered in time. This testing process ensures that the product developed works according to expectations and that this product is developed according to the standards. The testing process involves building of test cases using the same programming language and these test cases are the ones against which, the product has to be tested.

4.4.1 TEST OBJECTIVES

- Testing is a process of examining a project to see if it works as expected and to find errors in it.
- A good case is the one which has a relatively higher probability of finding a new/undiscovered error.
- A test is successful if it can discover an undiscovered error. If testing is conducted systematically and successfully it will be able to uncover undiscovered errors in the software.

  Testing cannot show defects or features that are missing, it can only find errors or defects that are already present.

4.4.2 TESTING PRINCIPLES

Before designing any effective test cases, it is important that the software engineer understands the product in its entirety and what each component is supposed to do. This is what should guide the software test case designing. All the tests should be traceable back to customer requirements that was gathered during the requirements gathering process. Each and every class of the project should be tested, and a code coverage report should be generated. This report shows the percentage of code that was covered in testing.

4.4.3 TESTING DESIGN

Software products can be tested in the following two ways:

- **White box Testing:**
  The basic concept of this type of testing is that the tester knows the specified function that a product has been designed to perform and so the test can be conducted that will demonstrate that each function is completely operational and at the same time searches for errors in each of the functions. This test case design method utilizes the control structure of the procedural design in order to design the test cases.
- **Black box Testing:**
  This type of testing is conducted by knowing the internal operation of the product. The test is conducted to make sure that all the components work together seamlessly. This will ensure that the internal operation is performed according to specification and that all internal components of the project have been adequately exercised. This type of*xx* testing focuses on the functional requirements of the software.

  The various types of testing done in this project using this approach includes:

- Equivalence partitioning: For the input values of date and time this approach was used. Since it is practically impossible to test all the possible values.
- Boundary value analysis: This method was used in testing all the input fields. Where the boundary values were chosen to ensure that the function works perfectly. As a part of this the null checks were placed in all of the inputs.
- Comparison testing: Since this is a in house tool which was customized entirely for the use of the associates there was no product in the market to which it could be compared. However, a comparison was made to an existing application

that was used for this purpose in the organization. It was found to be far superior in performance especially considering the additional features added to it.

4.4.4 TESTING STRATEGIES

A software testing strategy is one which provides the software developer a road map. Testing should be something that should be planned meticulously in advance so that the systematic implementation of this is possible during software development. Hence a template for the software testing, should be defined for software development process. That is a set of steps into which we can put specific test cases. The software testing strategies should have the characteristics mentioned below:

- Testing should always begin at the module level and should work outward to the integration of the entire project.
- The testing methods appropriate at each point of the development life cycle is different. Initially, unit testing is appropriate while towards the end integration testing is appropriate.
- Both the developer and the selected stakeholders should test the application.

4.4.5 LEVELS OF TESTING

Testing should be done at different levels of the development in the life cycle of software. In this project, testing has been done in these levels.

- **Unit Testing:** This is the first level of testing. Unit testing will verify the smallest unit of software - the module. White box orientation is the feature of unit testing. Different modules of the software are tested against the specifications which is provided during the design phase for the different modules. This is basically the verification of the code which is produced during the coding phase, and so this tests the internal logic of the code. This has been done using JUnit test cases designed for positive as well as negative test cases. The code coverage of these test cases was 80%.

- **Integration Testing:** Integration testing is the second level of testing. This is done to ensure that all the different modules of the code work well together when integrated. Since the UI, database and parts of the functionality were done separately integration testing is done after each integration to ensure that no bugs have occurred due to integration.

- **Functional Testing:** Functionality testing is a blackbox oriented testing. This is done to ensure that the functionalities of the system are working as expected. In this project this was done by the software developer as well as the stakeholder after each version of the project.

Chapter 5

Conclusions and Future Work

As every project starts with a goal to establish, a problem to solve and to make existing projects better, they all lead to a result. These results help us to determine whether the approach that is taken, the job done, analysis and research conducted is correct and up to the mark or not. These results then help us to conclude the aim of the project.

5.1 RESULTS & ANALYSIS

This project has addressed and solved a problem that was faced by the associates of the team in the organization in an effective easy to use manner thereby reducing the time taken to perform tasks. The results highlighted the use of the existing applications and tools to automate the task of manually creating the logs. Since the clients use software that directly impacts the health of a patient it is important to ensure the correctness and accuracy of the solution and the ease of use. Thus, the aim of the project has been accomplished although there is scope for more additions and improvements.

5.2 COMPARATIVE STUDY

This project was aimed to provide a unified platform with search engine capabilities that help users look for data, Jenkins jobs, codes etc within particular projects and departments. Data retrieval and data sharing was extremely time consuming and sometimes downright impossible.

The user interface of this project is a vast improvement over the existing product which was built using HTML, CSS and Java Script. In this project the UI has been built using Bootstrap in addition to HTML, CSS and Java Script. The other major issue was the sheer amount of information that had to be entered.

5.3 DISCUSSIONS

The project opens up a lot of topics for discussions related to DevOps which focuses on both development and operations which is the upcoming trend in the industry required by each team to be familiarized with both the development as well as the infrastructure end in an organization hereby creating increased transparency, communication, and collaboration across development, IT operations and the business plan.

This project also helped to underline the importance of automation in software development. This project also gave an opportunity to work with the operations side of software development while deploying the product in the cert environment for testing and then the production environment.

5.4 CONCLUSIONS

The problem in status quo is that it becomes a very tedious and redundant task for employees to search to multiple websites and platforms to collect and corelate data from them for their use. These platforms include Jira, Sharepoint, RQM, Jenkins etc. All of them contain data and details of a particular project and or code that employees use on a daily basis. The idea is to use Restful API to access data from these platforms and accommodate it on a common platform the allows the user to access all data on one common platform and it helps the first time users and new employees at the firm to be familiar with hierarchy of the department with respect to the project and sub-groups. This concept therefore acts like beneficial tool for both daily users and new employees. Using Restful API makes the entire process much easier and simpler.

The project successfully establishes the thought-out platform and system and the search engine feature within it to create a multi-purpose, unified system that saves time, effort and increases efficiency. The system not only helps the regular employees but also provides an opportunity to see the bigger picture for the newly joined.

5.5 SCOPE FOR FUTURE WORK

Future research should concentrate on best practices, new privacy and security standards, robust privacy preserving technologies, and privacy awareness and education. In this present technological world, needs, ideas, demands and wishes would keep on increasing and hence calling a project complete is just an abstract concept. This project is well capable of expanding itself, and if we ponder upon, it is having scope for future work. There are more inhouse tools which are developed which needs to be integrated with this solution. The integration of API and search engines can be really helpful for many important projects. Currently the way in which this tool executes, the first step towards enhancement will be refining the software to understand the alerts generated by the monitoring software and building the unified platform and increase the services and with a mountain of data.

Appendices

Development Technologies:

Technology is the total sum of techniques, methods, skills and procedures utilized in the development of products and services or in the attainment of objectives. Technology include the knowledge of techniques or processes, or it can be integrated into machines in order to allow their functioning without the actual knowledge of their workings. Systems which applies the technology by taking any input, then manipulating it according to the use of the system, and then presenting an expected outcome.

A.1 APPENDIX A FRONTEND

Front-end development is the process of converting the data from the backend into a graphical user interface, via HTML, CSS, and JavaScript. This enables the users to view and manipulate that data. Bootstrap is a free, open-source framework built using CSS that aims at building mobile-first front-end web development which is highly responsive. It also consists of CSS as well as JavaScript based design templates. This has been used for the various components such as forms, dropdowns, buttons, navigation, and other such interface components. The development of single-page or mobile applications base can be achieved through Bootstrap.

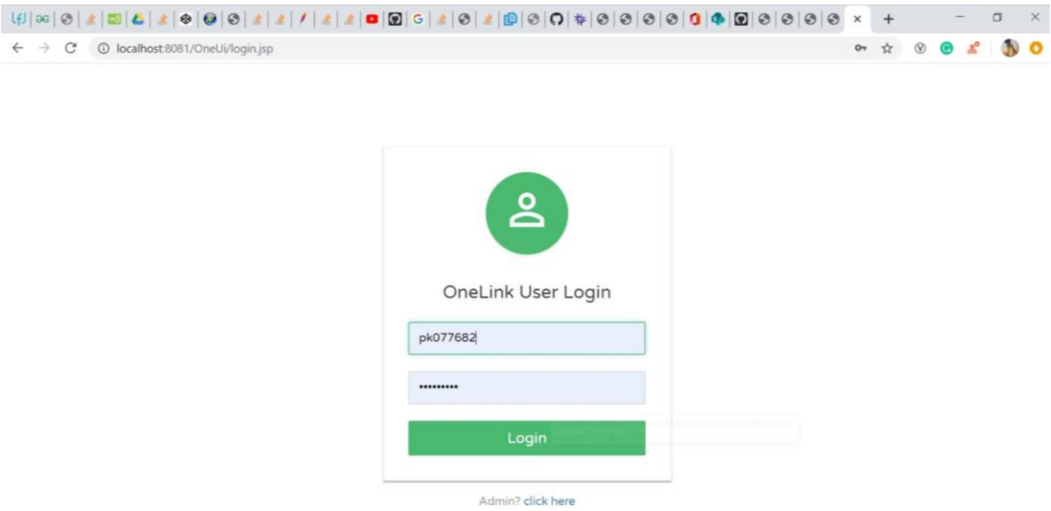
A.2 APPENDIX A BACKEND

The back end refers to the components of an application or a particular program's code that enables it to operate, this part is not accessible by a user. The majority of the data and the syntax are stored in the backend of a computer system. The code can be comprised of one or more than one programming languages. Eclipse IDE has been used for developing the application. Spring Framework has been used extensively. In particular Spring Boot has been used so that no external server is required to run the application. Spring Framework helps in creating micro service based project. This increases the adaptability and extensibility of the project. This has been developed by the Pivotal Team. It is used to develop stand-alone, production ready spring web applications. This enables your application handle concurrent access with a relatively small number of kernel threads.

A.3 APPENDIX A DATABASE

A database management system (DBMS) is built to manipulate, to retrieve, to define and to manage data in a particular database. A database management system will in general to manipulates the data, it can also alter the data format, the field names, the record structure as well as the file structure. Rules to validate and manipulate this data is also defined by it. The use of a database management system can help to opt out of developing programs for data maintenance. Query languages, such as SQL, is used along with the DBMS package to communicate with a database. PostgreSQL, also called as Postgres, is a free, open source relational database management system which emphasizes extensibility and a high technical standards compliance. It has been built to handle a wide range of workloads, from a solitary machine to data warehouses or a web service which many have concurrent users.

A.4 APPENDIX A SCREENSHOTS



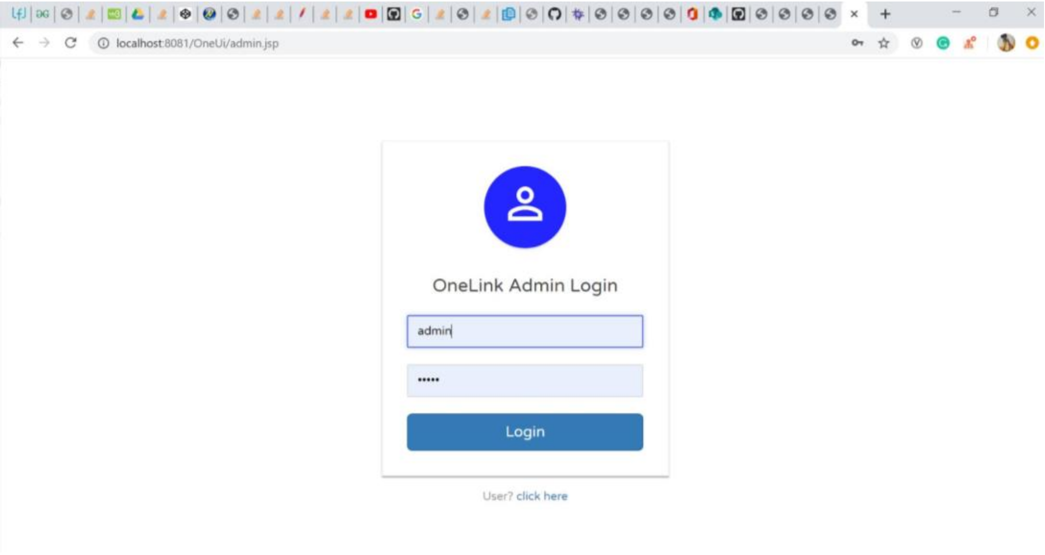
localhost:8081/OneUi/login.jsp

OneLink User Login

pk077682

•••••••••

FIGURE A.1: Login Page



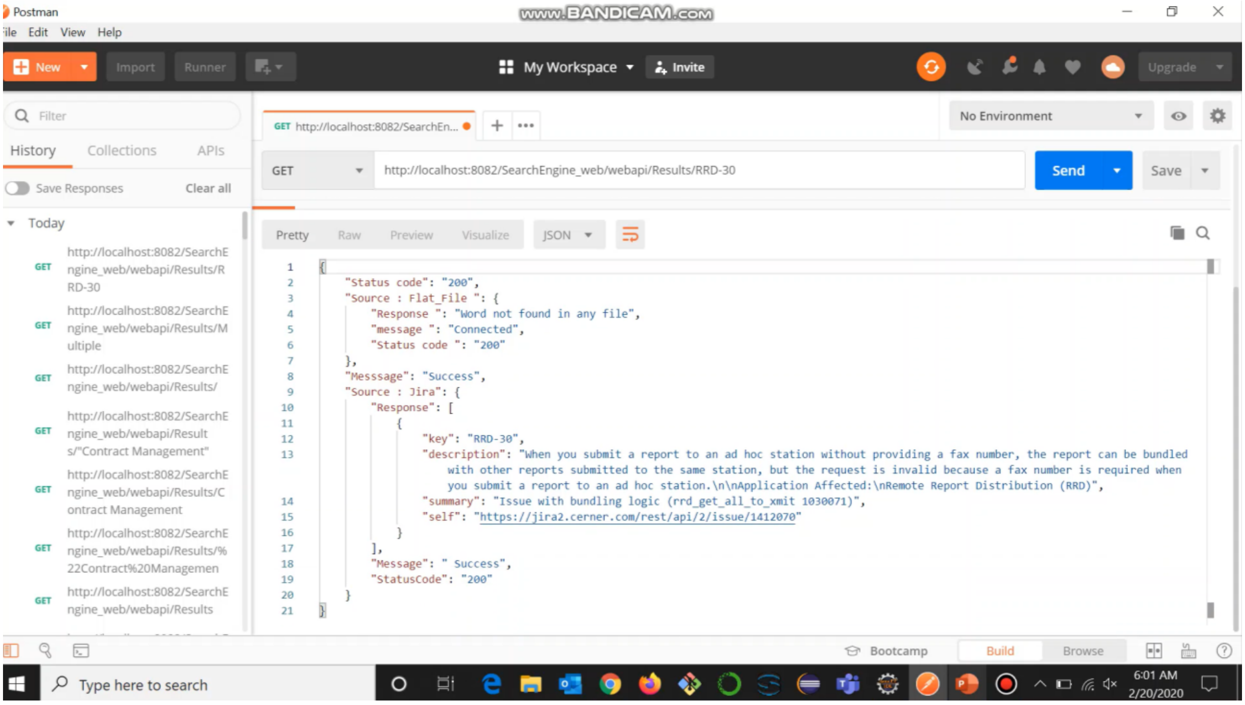
localhost:8081/OneUi/admin.jsp

OneLink Admin Login

admin

•••••

FIGURE A.2: Admin login Page



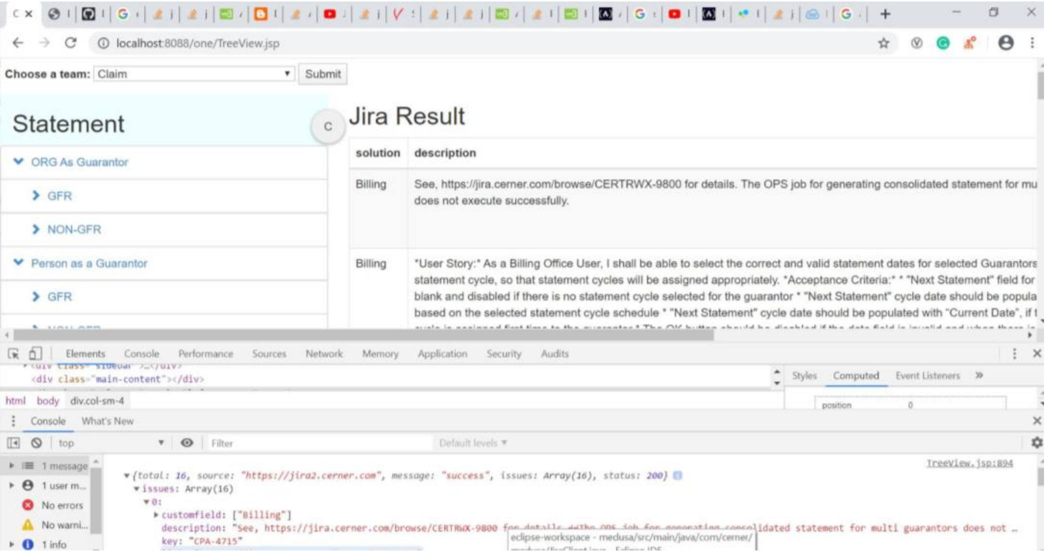
GET http://localhost:8082/SearchEngine_web/webapi/Results/RRD-30
GET http://localhost:8082/SearchEngine_web/webapi/Results/Multiple
GET http://localhost:8082/SearchEngine_web/webapi/Results/
GET http://localhost:8082/SearchEngine_web/webapi/Results/"Contract Management"
GET http://localhost:8082/SearchEngine_web/webapi/Results/Contract Management
GET http://localhost:8082/SearchEngine_web/webapi/Results/%22Contract%20Managemen
GET http://localhost:8082/SearchEngine_web/webapi/Results

GET http://localhost:8082/SearchEn...

GET http://localhost:8082/SearchEngine_web/webapi/Results/RRD-30

```
{
    "Status code": "200",
    "Source : Flat_File ": {
        "Response ": "Word not found in any file",
        "message ": "Connected",
        "Status code ": "200"
    },
    "Messsage": "Success",
    "Source : Jira": {
        "Response": [
            {
                "key": "RRD-30",
                "description": "When you submit a report to an ad hoc station without providing a fax number, the report can be bundled with other reports submitted to the same station, but the request is invalid because a fax number is required when you submit a report to an ad hoc station.\n\nApplication Affected:\nRemote Report Distribution (RRD)",
                "summary": "Issue with bundling logic (rrd_get_all_to_xmit 1030071)",
                "self": "https://jira2.cerner.com/rest/api/2/issue/1412070"
            }
        ],
        "Message": " Success",
        "StatusCode": "200"
    }
}
```

FIGURE A.3: Postman results



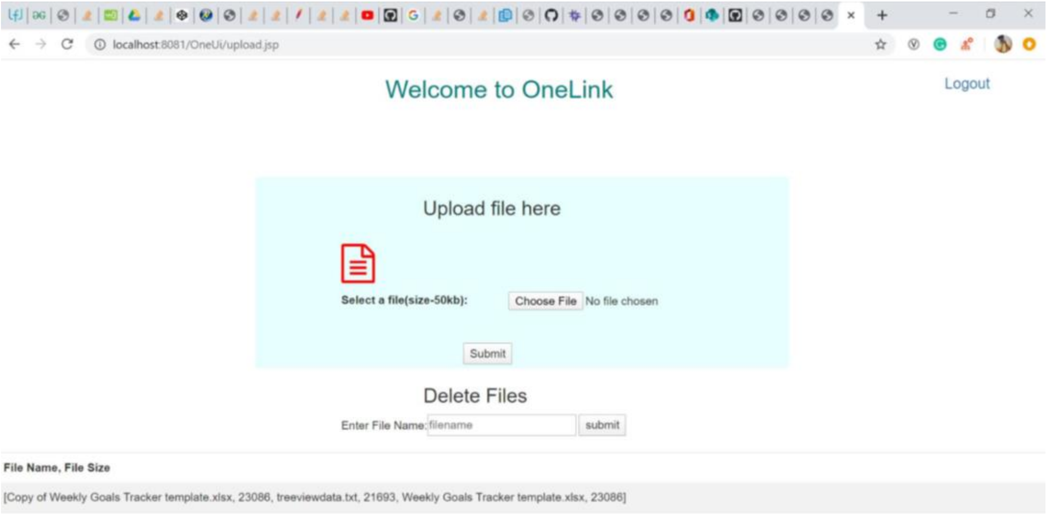
Jira Result

solution	description
Billing	See, https://jira.cerner.com/browse/CERTRWX-9800 for details. The OPS job for generating consolidated statement for mu does not execute successfully.
Billing	*User Story:* As a Billing Office User, I shall be able to select the correct and valid statement dates for selected Guarantors statement cycle, so that statement cycles will be assigned appropriately. *Acceptance Criteria:* * "Next Statement" field for blank and disabled if there is no statement cycle selected for the guarantor * "Next Statement" cycle date should be popula based on the selected statement cycle schedule * "Next Statement" cycle date should be populated with "Current Date", if t

```
<div class="main-content"></div>
```

```
▾{total: 16, source: "https://jira2.cerner.com", message: "success", issues: Array(16), status: 200}
  ▾issues: Array(16)
    ▾0:
      ▸customfield: ["Billing"]
       description: "See, https://jira.cerner.com/browse/CERTRWX-9800 for details ##The OPS job for generating consolidated statement for multi guarantors does not …
       key: "CPA-4715"
```

FIGURE A.4: Jira Results



localhost:8081/OneUi/upload.jsp

Welcome to OneLink

Upload file here

Select a file(size-50kb): No file chosen

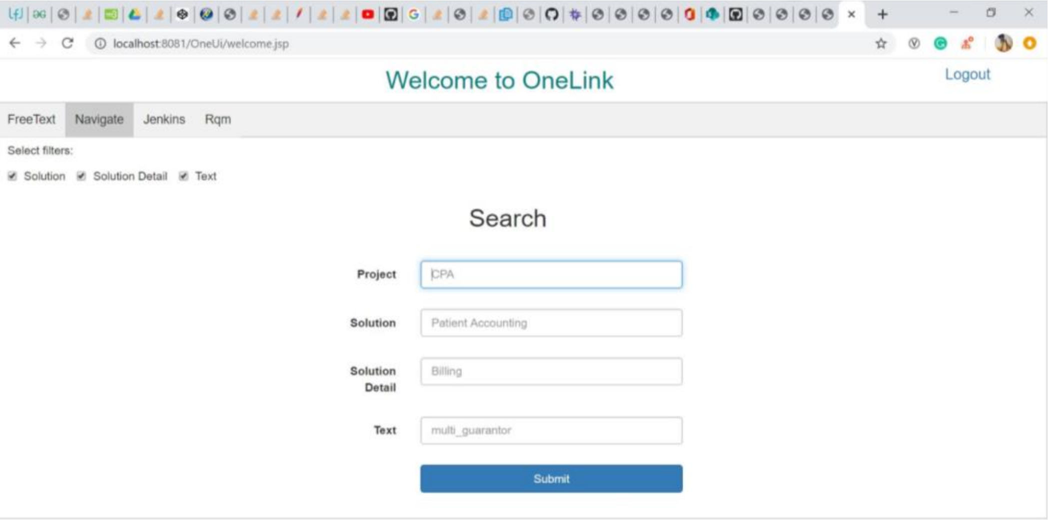
Delete Files

Enter File Name: filename

File Name, File Size

[Copy of Weekly Goals Tracker template.xlsx, 23086, treeviewdata.txt, 21693, Weekly Goals Tracker template.xlsx, 23086]

FIGURE A.5: Admin file upload page



Select filters:

☑ Solution ☑ Solution Detail ☑ Text

Search

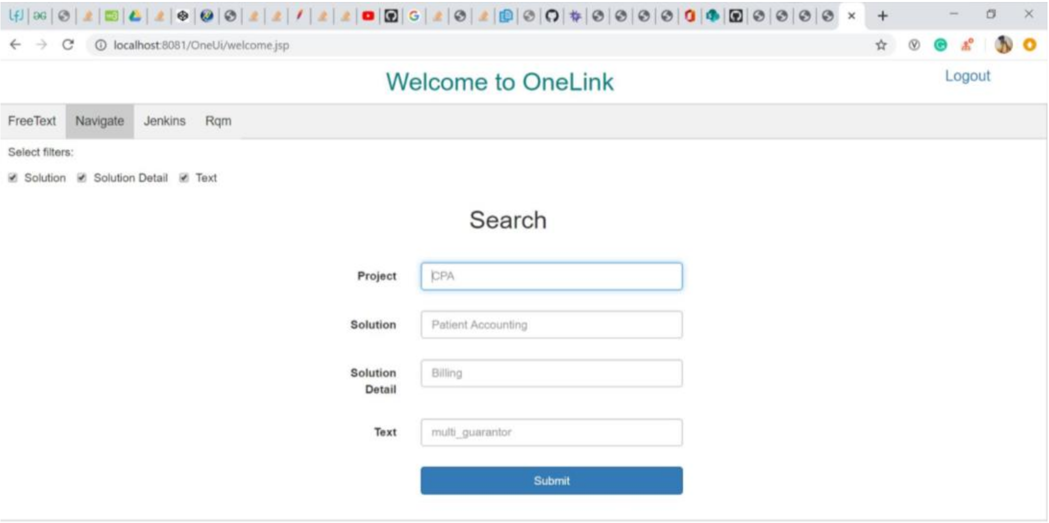
Project: CPA

Solution: Patient Accounting

Solution Detail: Billing

Text: multi_guarantor

FIGURE A.6: Landing page



localhost:8081/OneUi/welcome.jsp

Welcome to OneLink

Select filters:

☑ Solution ☑ Solution Detail ☑ Text

Search

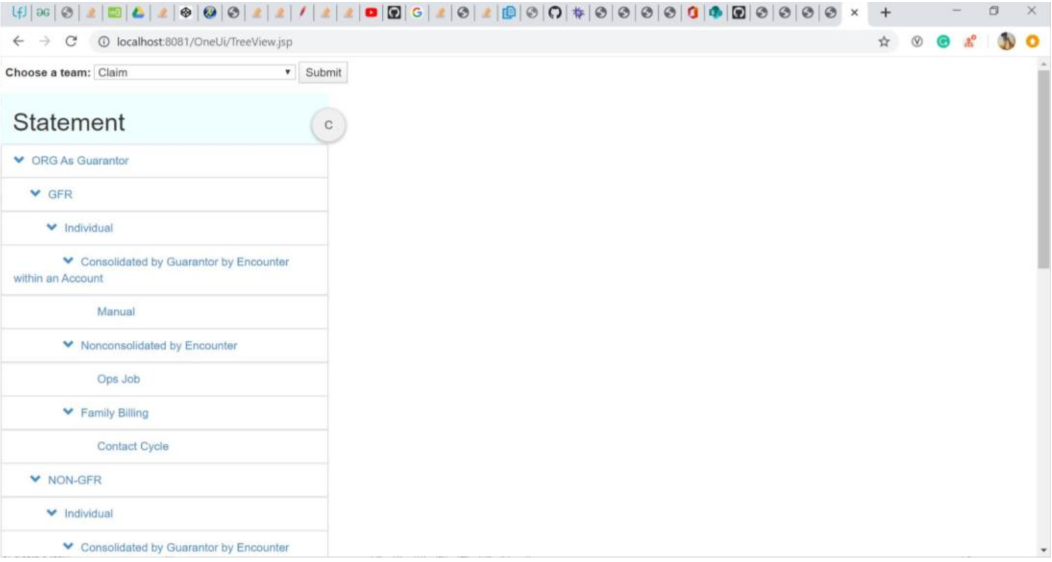
Project: CPA

Solution: Patient Accounting

Solution Detail: Billing

Text: multi_guarantor

FIGURE A.7: Landing page(Navigate)



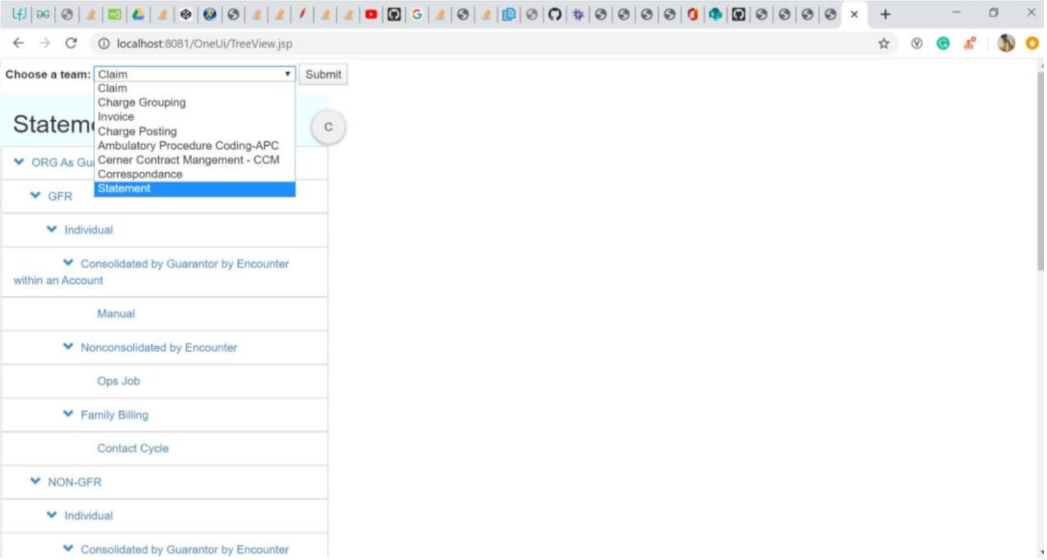
localhost:8081/OneUi/TreeView.jsp

Choose a team: Claim

Statement

- ORG As Guarantor
  - GFR
    - Individual
      - Consolidated by Guarantor by Encounter within an Account
        - Manual
      - Nonconsolidated by Encounter
        - Ops Job
      - Family Billing
        - Contact Cycle
  - NON-GFR
    - Individual
      - Consolidated by Guarantor by Encounter

FIGURE A.8: Hierarchy page

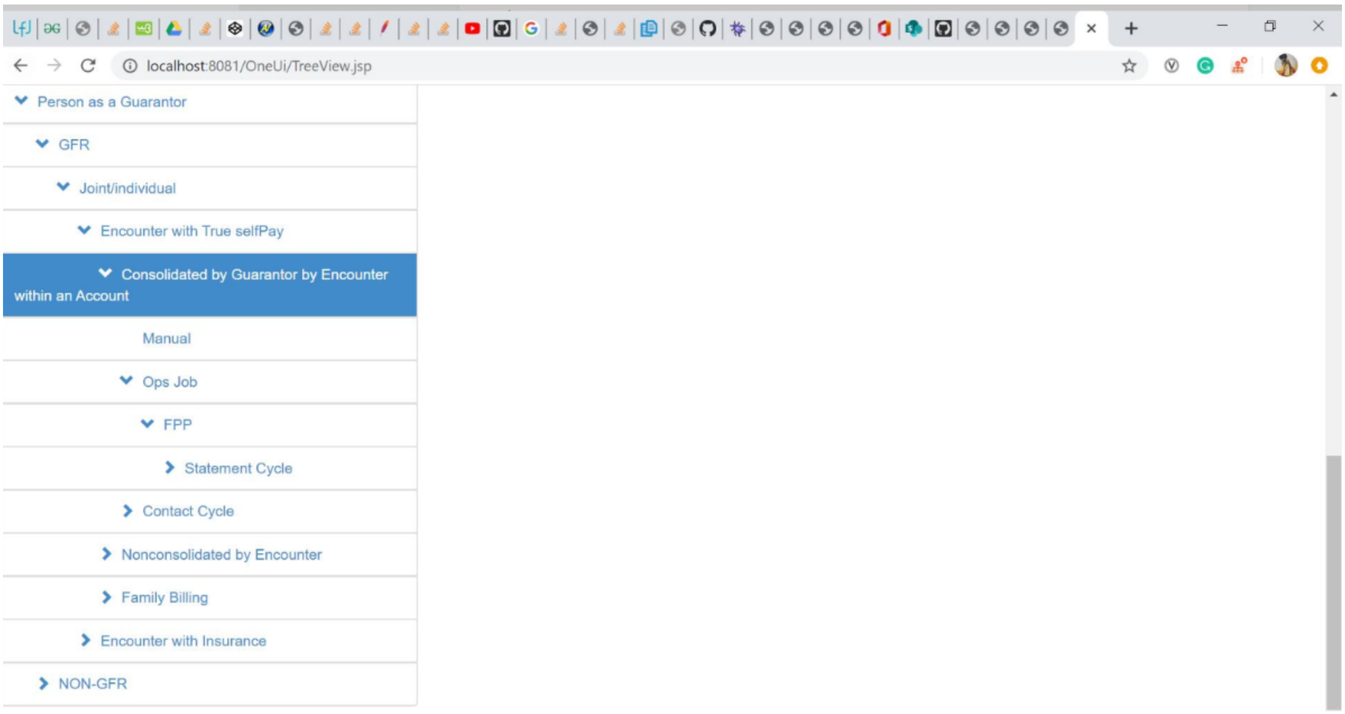


localhost:8081/OneUi/TreeView.jsp

Choose a team: Claim

Submit
Claim
Charge Grouping
Invoice
Charge Posting
Ambulatory Procedure Coding-APC
Cerner Contract Mangement - CCM
Correspondance
Statement
Statem
C
ORG As Gu
GFR
Individual
Consolidated by Guarantor by Encounter within an Account
Manual
Nonconsolidated by Encounter
Ops Job
Family Billing
Contact Cycle
NON-GFR
Individual
Consolidated by Guarantor by Encounter

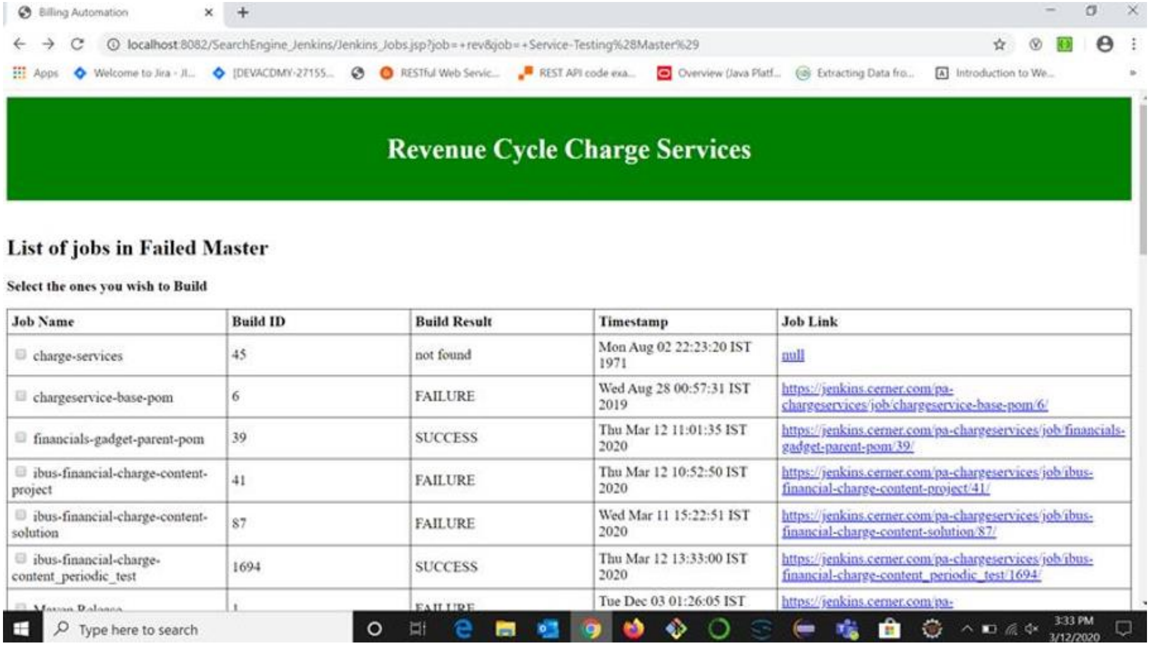
FIGURE A.9: Hierarchy page(search)



localhost:8081/OneUi/TreeView.jsp

- Person as a Guarantor
  - GFR
    - Joint/individual
      - Encounter with True selfPay
        - Consolidated by Guarantor by Encounter within an Account
          - Manual
          - Ops Job
            - FPP
              - Statement Cycle
          - Contact Cycle
        - Nonconsolidated by Encounter
        - Family Billing
      - Encounter with Insurance
  - NON-GFR

FIGURE A.10: Hierarchy page structure



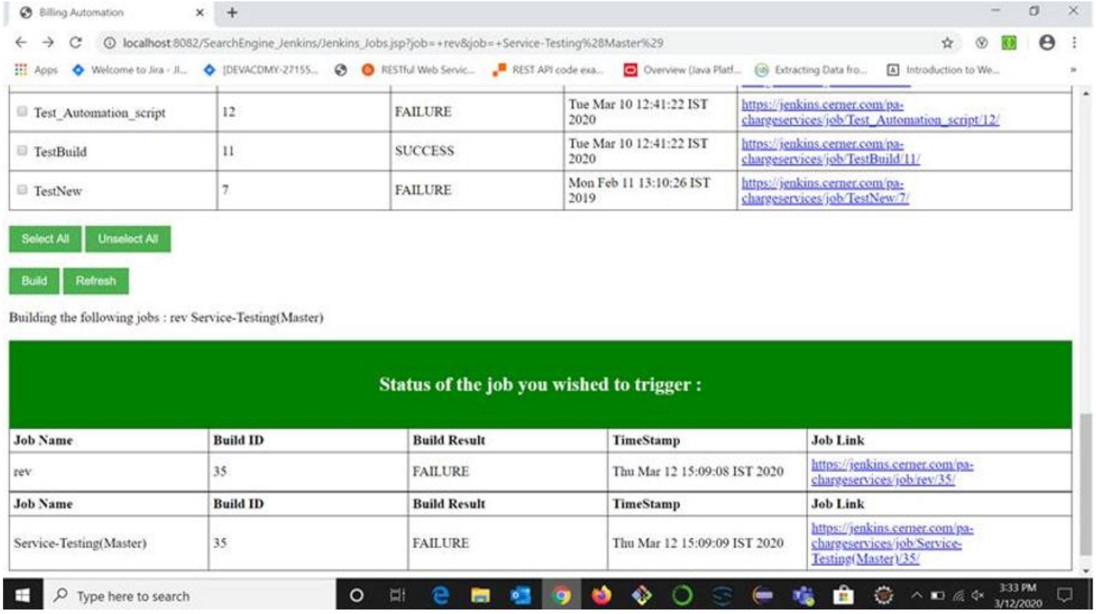
Revenue Cycle Charge Services

List of jobs in Failed Master

Select the ones you wish to Build

Job Name	Build ID	Build Result	Timestamp	Job Link
□ charge-services	45	not found	Mon Aug 02 22:23:20 IST 1971	null
□ chargeservice-base-pom	6	FAILURE	Wed Aug 28 00:57:31 IST 2019	https://jenkins.cerner.com/pa-chargeservices/job/chargeservice-base-pom/6/
□ financials-gadget-parent-pom	39	SUCCESS	Thu Mar 12 11:01:35 IST 2020	https://jenkins.cerner.com/pa-chargeservices/job/financials-gadget-parent-pom/39/
□ ibus-financial-charge-content-project	41	FAILURE	Thu Mar 12 10:52:50 IST 2020	https://jenkins.cerner.com/pa-chargeservices/job/ibus-financial-charge-content-project/41/
□ ibus-financial-charge-content-solution	87	FAILURE	Wed Mar 11 15:22:51 IST 2020	https://jenkins.cerner.com/pa-chargeservices/job/ibus-financial-charge-content-solution/87/
□ ibus-financial-charge-content_periodic_test	1694	SUCCESS	Thu Mar 12 13:33:00 IST 2020	https://jenkins.cerner.com/pa-chargeservices/job/ibus-financial-charge-content_periodic_test/1694/
□ Maven Release	1	FAILURE	Tue Dec 03 01:26:05 IST	https://jenkins.cerner.com/pa-

FIGURE A.11: Jenkins Results page



□ Test_Automation_script	12	FAILURE	Tue Mar 10 12:41:22 IST 2020	https://jenkins.cerner.com/pa-chargeservices/job/Test_Automation_script/12/
□ TestBuild	11	SUCCESS	Tue Mar 10 12:41:22 IST 2020	https://jenkins.cerner.com/pa-chargeservices/job/TestBuild/11/
□ TestNew	7	FAILURE	Mon Feb 11 13:10:26 IST 2019	https://jenkins.cerner.com/pa-chargeservices/job/TestNew/7/

Building the following jobs : rev Service-Testing(Master)

Status of the job you wished to trigger :

Job Name	Build ID	Build Result	TimeStamp	Job Link
rev	35	FAILURE	Thu Mar 12 15:09:08 IST 2020	https://jenkins.cerner.com/pa-chargeservices/job/rev/35/

Job Name	Build ID	Build Result	TimeStamp	Job Link
Service-Testing(Master)	35	FAILURE	Thu Mar 12 15:09:09 IST 2020	https://jenkins.cerner.com/pa-chargeservices/job/Service-Testing(Master)/35/

FIGURE A.12: Jenkins Results page

Flow Diagram

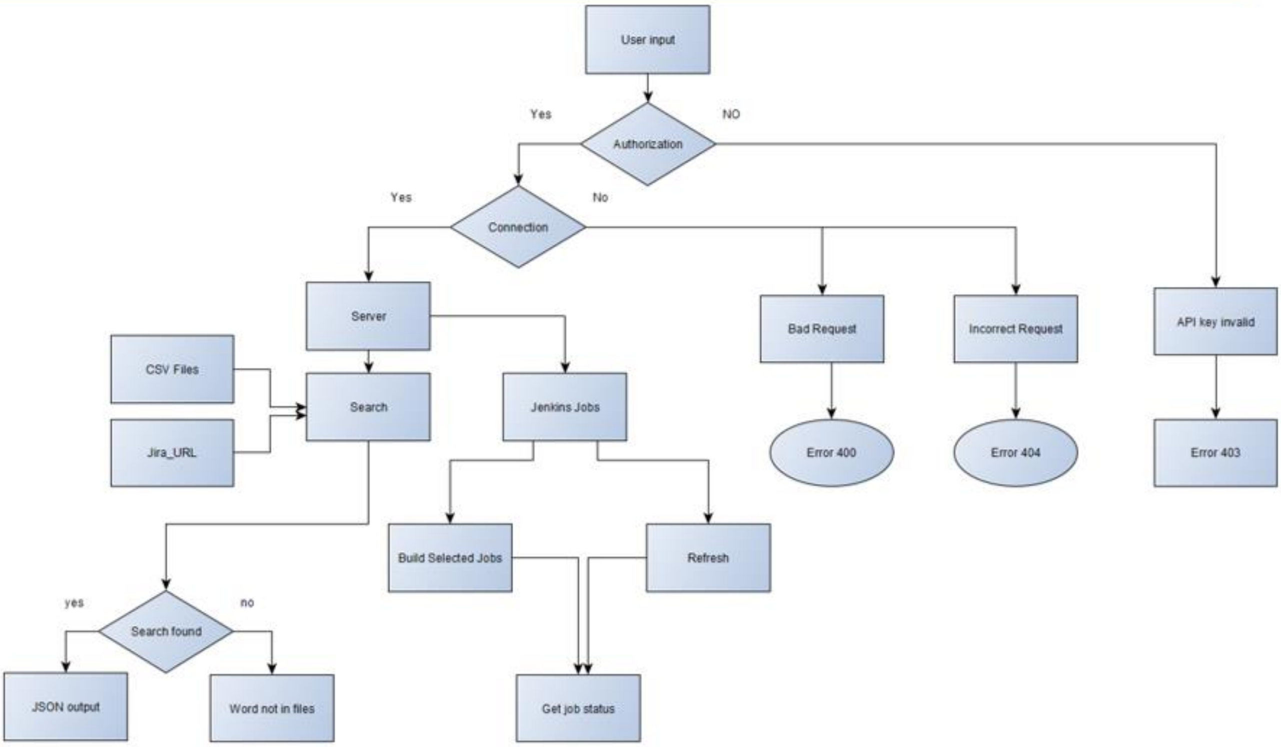


FIGURE A.13: Flow diagram

REFERENCES

[1] F. Pacquier, “The ELOG Home Page”, *https://elog.psi.ch/elog/*, Elog.psi.ch, 2002.

[2] Salah, Tasneem, M. Jamal Zemerly, Chan Yeob Yeun, Mahmoud AlQutayri, and Yousof Al-Hammadi. “The evolution of distributed systems towards microservices architecture.” *In 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 318-325. IEEE, 2016.

[3] Guntupally, Kavya, Ranjeet Devarakonda, and Kenneth Kehoe. “Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example.” *In 2018 IEEE International Conference on Big Data (Big Data)*, pp. 5328-5329. IEEE, 2018.

[4] Preeti Yadav and Bhupender Singh, “A comparative study of versions of JavaScript”, *International Journal of Computational Intelligence Research*, 2017.

[5] C. Walls, *Spring Boot in action*, 4th ed. Manning Publications, 2016.

[6] Mark Otto, “Bootstrap”, *https://getbootstrap.com*, 2020.

[7] “PostgreSQL 11.7 Documentation”,*https://www.postgresql.org/docs/manuals/* , The PostgreSQL Global Development Group, 2020.

[8] “Spring Initializr”, *Spring Initializr*, 2020. [Online]. Available: https://start.spring.io . [Accessed: 14- Mar- 2020].

[9] “W3.CSS Home”, *W3schools.com*, 2020. [Online]. Available: https://www.w3schools.com/w3css/ . [Accessed: 14- Mar- 2020].

[10] “Font Awesome”, *Fontawesome.com*, 2020. [Online]. Available: https://fontawesome.com . [Accessed: 14- Mar- 2020].

[11] “Thymeleaf”, *Thymeleaf.org, 2020*. [Online]. Available: https://www.thymeleaf.org/. [Accessed: 14- Mar- 2020].

[12] “PostgreSQL: The world’s *lxi*most advanced open source database”, *https://www.postgresql.org/*, 2020.

[13] “HTML Snippets for Twitter Boostrap framework : Bootsnipp.com”,

https://bootsnipp.com/snippets/dlZAN

[14] "Stack Overflow - Where Developers Learn, Share, & Build Careers", *https://stackoverflow.com* , 2020.