Software Requirements Specification

for

DB Version Tracking System

Version 1.0 approved

Prepared by Suyash Wagh

SE2 Digital Service LLP

26-08-2020

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this application is to help to track the executed DB queries during a releases. This version management application will help to rectify the query bugs in initial phase, which will reduce the later risk of bug or failure of continues release.

1.2 Product Scope

Our scope is to maintain the DB queries request made by the associate and the follow up of that query at every stage. The reminder system will help associates and approver to track the query requests.

Our goal is to reduce the overhead of the queries management and bring operations like approving, rejecting, commenting on a single web based application.This system also stores the logs of all queries for better management.

1.3 Definition, Acronym, And Abbreviations

DB : Database
SMTP : Simple Mail Transfer Protocol

2. Overall Description

2.1 Product Overview

This project is based on client and server based architecture. The Users will be able to access this application using a URL which will open a web page on web browser.

There are mainly three pages:

1. **Request page** :
   Request Page creates a request of DB query and send notification to the all approvers.

2. **View page**
   View page shows a list of all queries with their status and also provides the hyperlink on ticket number for approver page of respective ticket.

3. **Approver page**
   Approver page validate the approver and provide the privilege to approve or reject the query with proper commenting.

2.2 Product Functions

- TicketNo generator creates the ticket number with the help of timestamp.
- CRUD functions for .csv file add, update, retrieve the data in csv file on server.
- Add function for .sql file add, update, retrieve the data in csv file on server.
- Notification System sends the notification to the approvers and request creator.

2.3 User Characteristics

The DB Version Tracking system includes the request creator and approver as a user with their capabilities.

1. Request Generator : Creates a request for DB queries.
2. Request Approver : Accept or reject the requests with proper comment.

2.4 Design and Implementation Constraints

DB version tracker system includes following constraints

- Set status as “Pending” whenever request get created.
- Approve and reject buttons should be disabled after status gets updated from “Pending”.

- If .log file exceeds 1 Mb then, new log file gets created.

3. Product Perspective

3.1 User Interfaces

The application GUI provides navbar, buttons allowing for easy control by a keyboard and a mouse.

3.2 Hardware Interfaces

- *2 GB RAM*
- *100 GB Hard Disk*

3.3 Software Interfaces

Technology Stack	Component
Operating System	Windows 8 and onwards
Backend & Server	Java 8 , Spring Boot
Database	Oracle 12C
Front End (UI Design)	Angular 1.5

3.4 Communications Interfaces

- *Spring Boot SMTP server for mail communication*
- *Web browsers*

3.5 Functional Requirements

Requirement Id	Requirement Statement
01	System should have request page, view page and approver page.
02	Mail must be send to the approver after creation of request.
03	Status of request set to “Pending” at time of creation.
04	SQL query can be editable for the Approver.
05	Approve and reject buttons get disabled after status updation from “Pending”.
06	Mail must be send to the request creator after approval or rejection.

3.6 Non Functional Requirements

Performance :

Ticket id generated as dynamically at time of loading request page.
Each Rest Api should be executed in appropriate time.

Compatibility :

Application should be run on Chrome 74 and onwards and Mozilla Firefox 72 and onwards.

Availability :

Availability of system will be available through out the working day of SE2 organization.

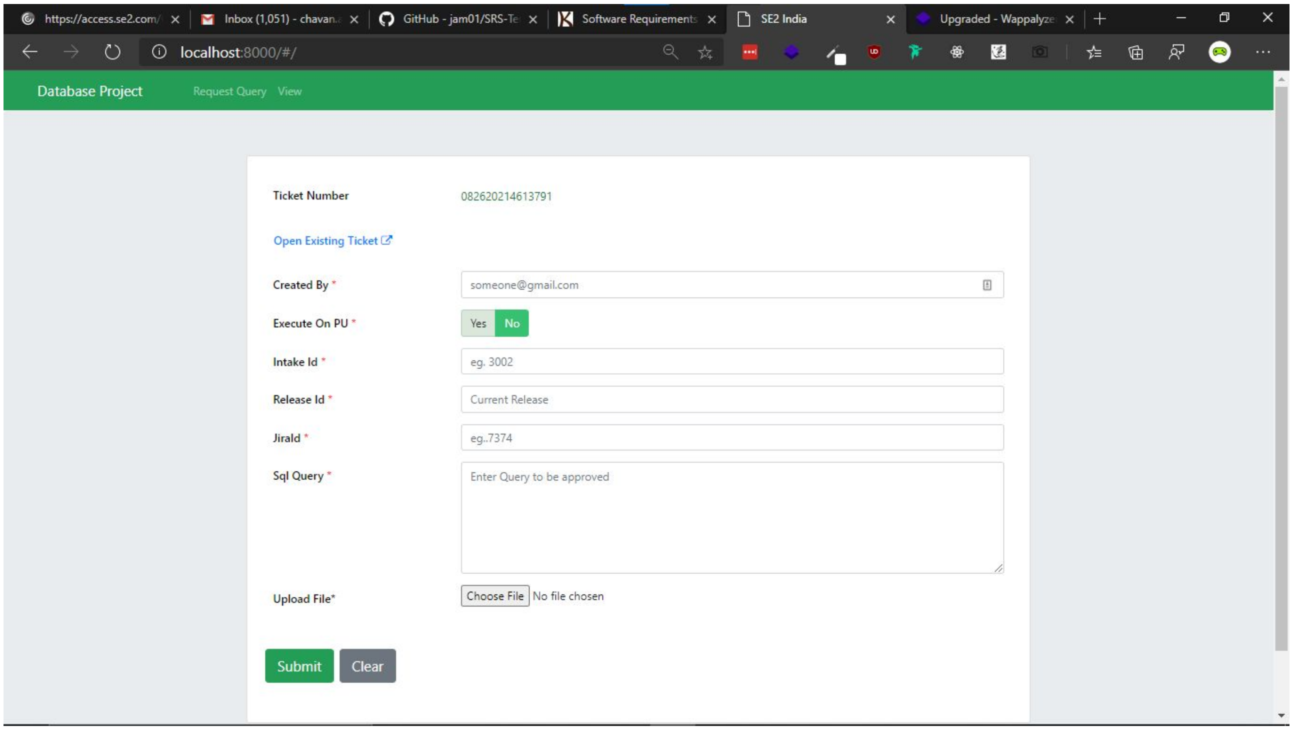
4. Workflow

The DB Version Tracking System includes various function to create query request and track the request through out the system.

4.1 Pages Overview

Request page is also homepage of our system which include link for view page.We can also render to view page through hyperlink on create request page.

4.1.1 Request Page



Ticket Number 082620214613791

Open Existing Ticket

Created By * someone@gmail.com

Execute On PU * Yes No

Intake Id * eg. 3002

Release Id * Current Release

JiraId * eg..7374

Sql Query * Enter Query to be approved

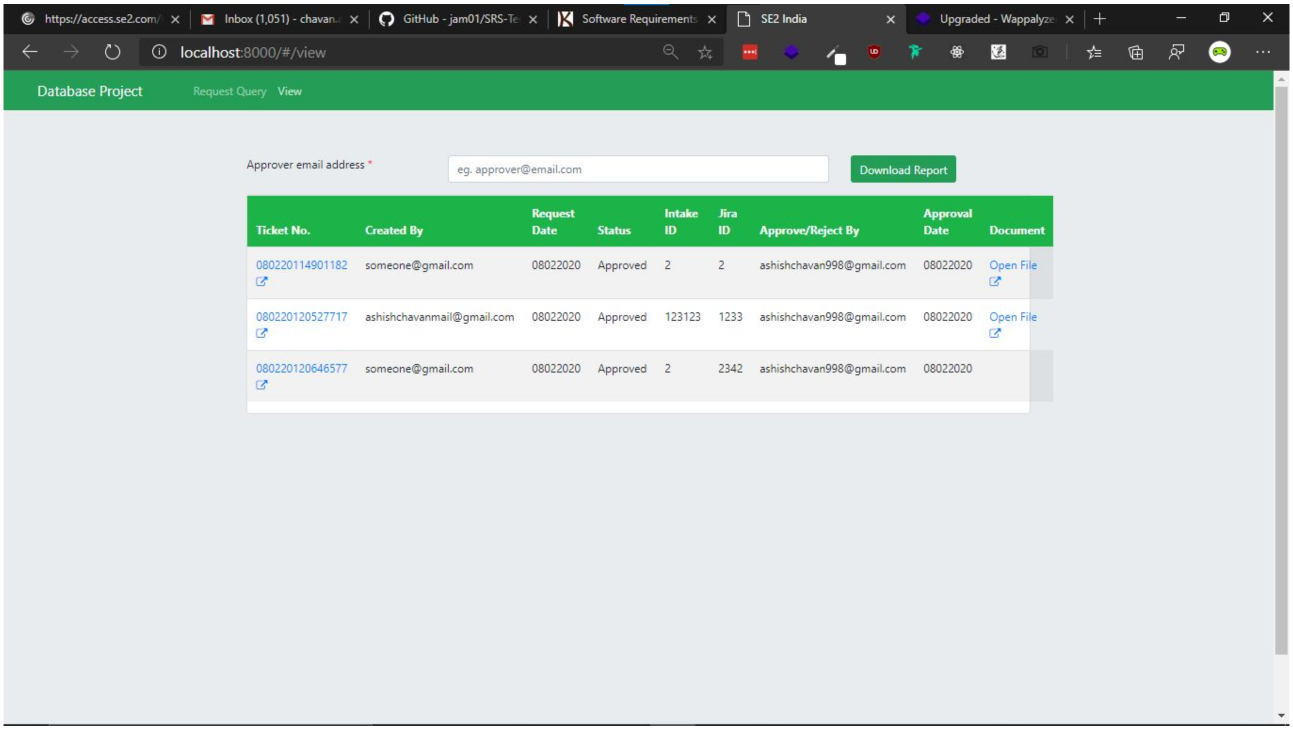
Upload File* Choose File No file chosen

Request Page

Field Name	Description
Ticket No	Auto Generated Field Format : MMDDYY + HHMMSSMS Read Only Field
Created By	Text Box Field is Mandatory Entry must be in email format
Intake Id	Text Box Field is Mandatory
Executed on PU?	Text Box Field is Mandatory Radio Button (Yes/No)
Jira Id	Text Box Field is Mandatory
Release Id	Text Box

	Field is Mandatory
SQL Query	Text Box Field is Mandatory
Open Existing Ticket	Hyperlink Redirect to view page
Upload Document	Field is Mandatory Browse the local PC for document
Submit	Button ● Validate all fields ● Save all data in .csv file ● Document get uploaded on server ● Send notification mail to the approvers ● "Request Created Successfully" message generated.
Clear	Button ● Clear all fields

4.1.2 View Page



Approver email address *

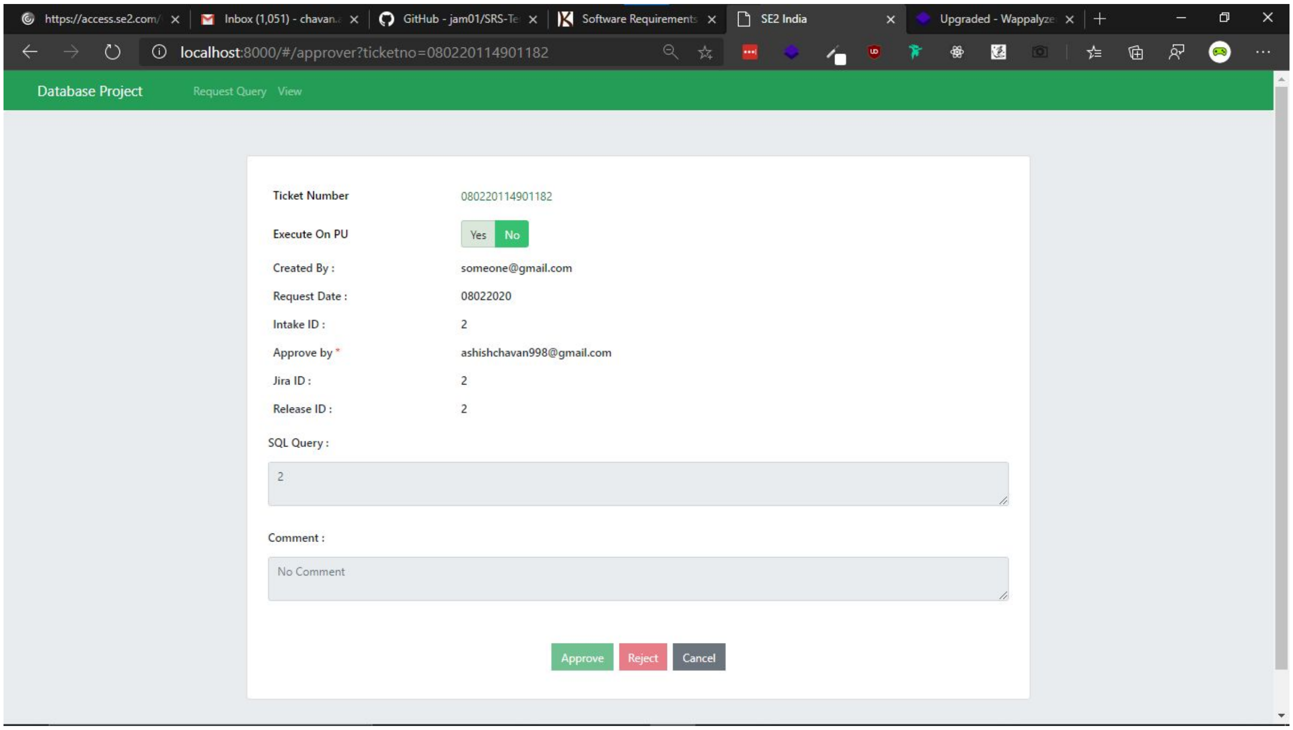
eg. approver@email.com

Ticket No.	Created By	Request Date	Status	Intake ID	Jira ID	Approve/Reject By	Approval Date	Document
080220114901182	someone@gmail.com	08022020	Approved	2	2	ashishchavan998@gmail.com	08022020	Open File
080220120527717	ashishchavanmail@gmail.com	08022020	Approved	123123	1233	ashishchavan998@gmail.com	08022020	Open File
080220120646577	someone@gmail.com	08022020	Approved	2	2342	ashishchavan998@gmail.com	08022020	

View Page

- Display all necessary fields in tabular data.
- Ticket No field is hyperlink which redirect you to approval page of respective ticket number.
- View Document is hyperlink which open the document uploaded for respective ticket number.
- Download button validate approval mail id and download the report about the requests generated.

4.1.3 Approver Page



Ticket Number 080220114901182

Execute On PU

Created By : someone@gmail.com

Request Date : 08022020

Intake ID : 2

Approve by * ashishchavan998@gmail.com

Jira ID : 2

Release ID : 2

SQL Query :

2

Comment :

No Comment

Approval Page

Field Name	**Description**
Ticket No	Read Only
Created By	Read Only
Intake Id	Read Only
Executed on PU?	Read Only
Jira Id	Read Only
Release Id	Read Only
Approved By	Text Box Mandatory
SQL Query	Editable Field Not Null

Comment	Text Box Mandatory
Approve	Button ● Validate approved by field with approval list. ● Update data in .csv with status as approved. ● Send mail to creator about status ● Add request in .sql file in proper format
Reject	Button ● Validate approved by field with approval list. ● Update data in .csv with status as rejected. ● Send mail to creator about status.
Cancel	Button ● Redirect to view page.

4.2 Script Overview

- After approval of request, request gets added into .sql file in specific format.
- Explicitly running script read .sql file and added data in “T_DB_VERSION” table having following fields.

Ticket No	Intake Id	Created By	Approved By	System Time

- Query executed on database and logs are generated and updated into .log file.

5. Constraints

- Users must be from SE2 organization.
- System must be access through SE2 systems or through citrix application.
- Script must be run on UNIX shell script.

6. Future Work

- Syntax of query can be checked at request page.
- Report generation can be done for particular time span.
- Rejected query can be raised with changes.