Synopsis Abstract

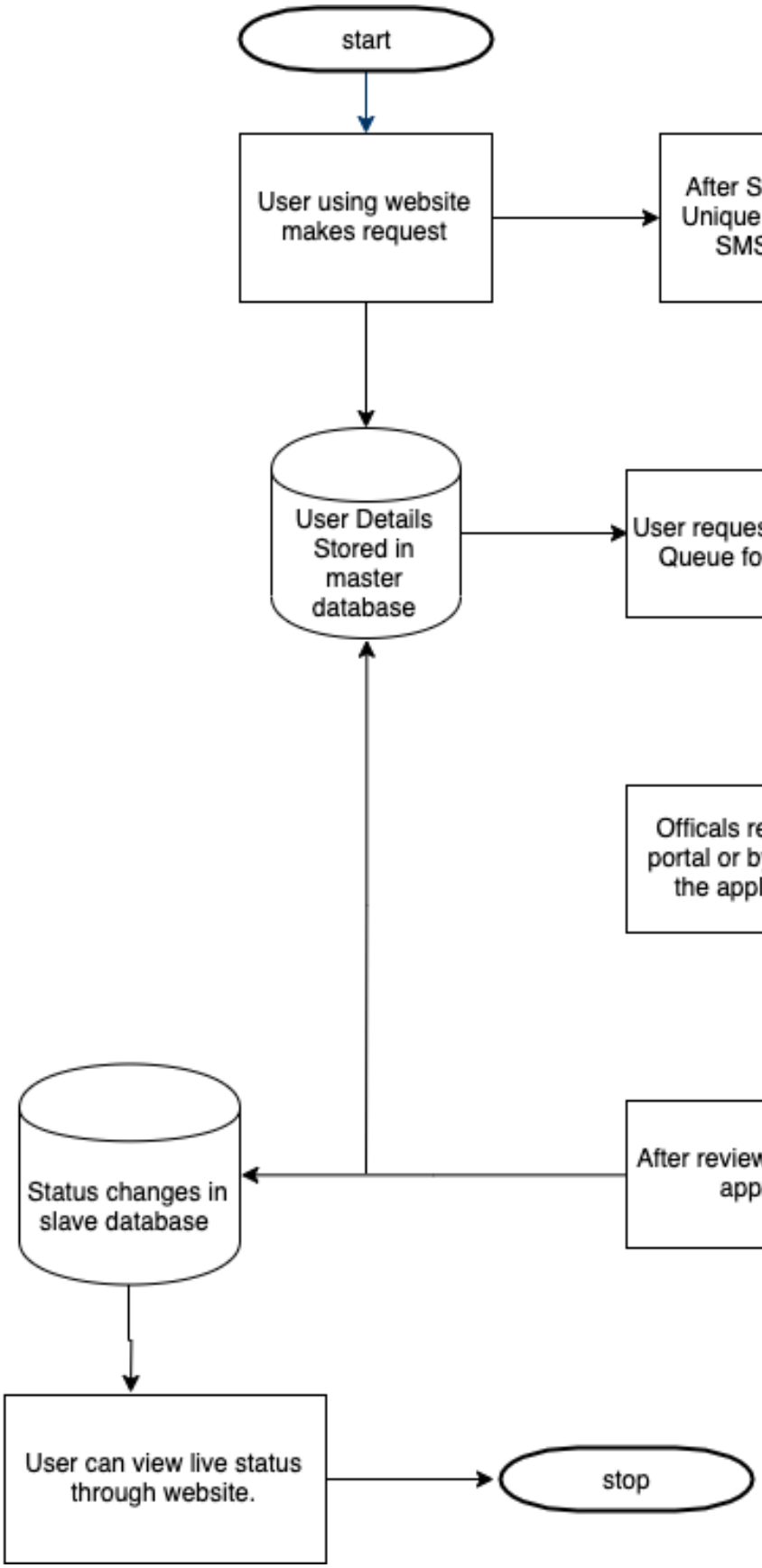
Based on the requirement of the better handling and acknowledgment of file, Thereby, constructing an environment in which the file handling and tracking on the progress can be monitored in an organised method. Using the advanced IDE tools, Database Connectivity, Encryption, and Primary Key Formation, we can have a secure, reliable and efficient workflow with complete acknowledgment and updates. Roles of Client, Officers, and Admin are mentioned to understand the requirement and the ease of use of the system, including the level of security to allow an individual to access and manipulate data. Ease of use to operate, with the help of devices and dynamic interface.

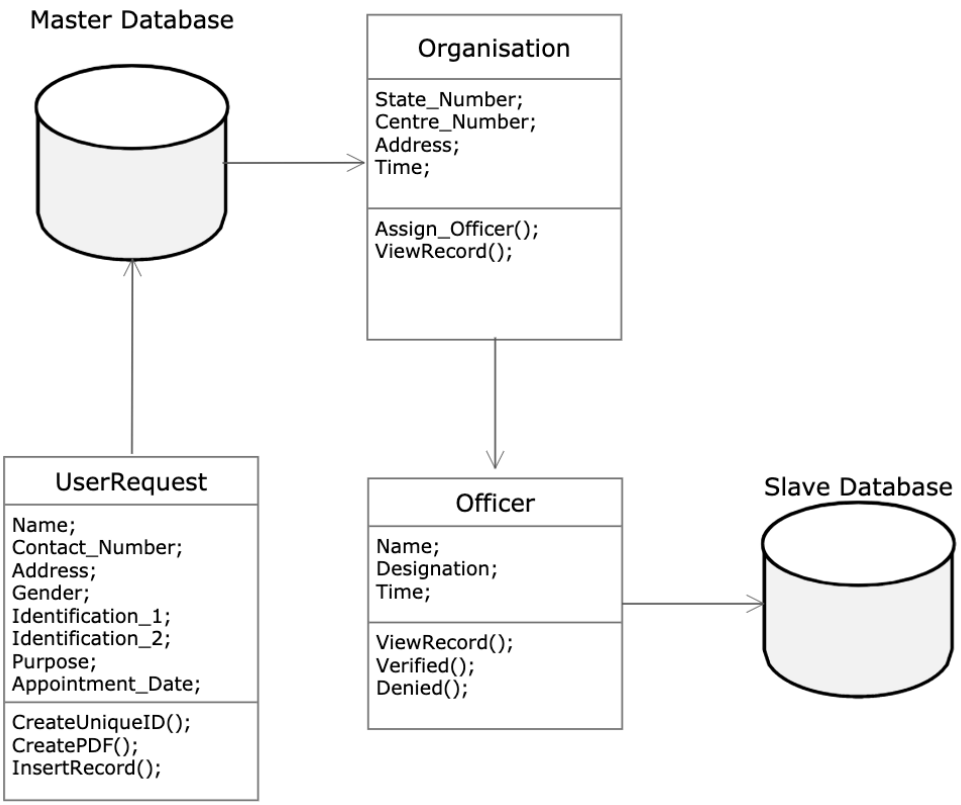
Existing Innovation-technology

There been many implementations of code generation and tracking systems, some of which is been used in many professional fields. Taking an example of code generation like; Barcode, PDF417, QR Code, etc. These are commonly used codes to keep records or some data. This method can help to store a large amount of data and are easily readable by the system. Implementation of this method along with database connectivity is the primary solution towards the problem. In the current scenario, when an individual send's some request to the government organisation the individual has no clue about the progress of his/her document, which is not admissible in work ethics. This practice puts the individual on the blind side of the whole infrastructure where there is no reliability or updates in real-time. Unlike the giant firms or e-commerce website where they have well-trained, implemented tracking system, which an individual can use it to have a status check on his/her bought product.

Approach to solve the problem

For Online, a request is made through a web-application, which is forwarded to the organisation. When this request is formed, a unique ID with PDF417 Barcode is generated, readable by the system, which consists of a pattern: SS/OOOOO/AAAAA/MM/YYYY
SS: State Initials
OOOOO: Organisation Code
AAAAA : Application Id
MM/YYYY: Month/Year
This ID is available to the user and is stored in organisation's database with user details.
So when the organisation officials examine the request of the user, upon completion of the review, they simply scan the PDF417 Barcode (on paper) or complete online via portal, which will directly update the status of the document in a separate database along with official's name and timestamp.
Admin keeps track of request in a database and any action necessary to allow access to a database with the authentic request.


start
User using website makes request
After Submission, Unique ID sent via SMS or Mail
User Details Stored in master database
User request allocated in Queue for reviewing
Officals review using portal or by examining the application ID
After reviewing, Officials approves
Status changes in slave database
User can view live status through website.
stop



Tools and Technology to be used

Considering all possible scenarios, in some organisation paperwork is needed and in other everything is done online. When applying for some government ID, we need to have a hard-copy, Others like creating some certificates, documentation and reviewing is done online.
Using NetBeans IDE and MySQL Server, we can construct a tracking system. With the help of IDE and additional Java Library, we can create a dynamic website with database, to store user details, along with that a portal is created for the organisation official's to review the details online and update the status simply by using a scanner to read PDF417 Barcode (on hard-copy). This method can drastically reduce the amount of paperwork required as all the scanned documents are stored online, only a single application paper is required with PDF417 Code, which will display all the detail on the portal. If any doubt, official can send back request with comment.
Java also supports access specifiers, which will limit the user or any unauthorised person to gain access to the database.

Challenges/Risk

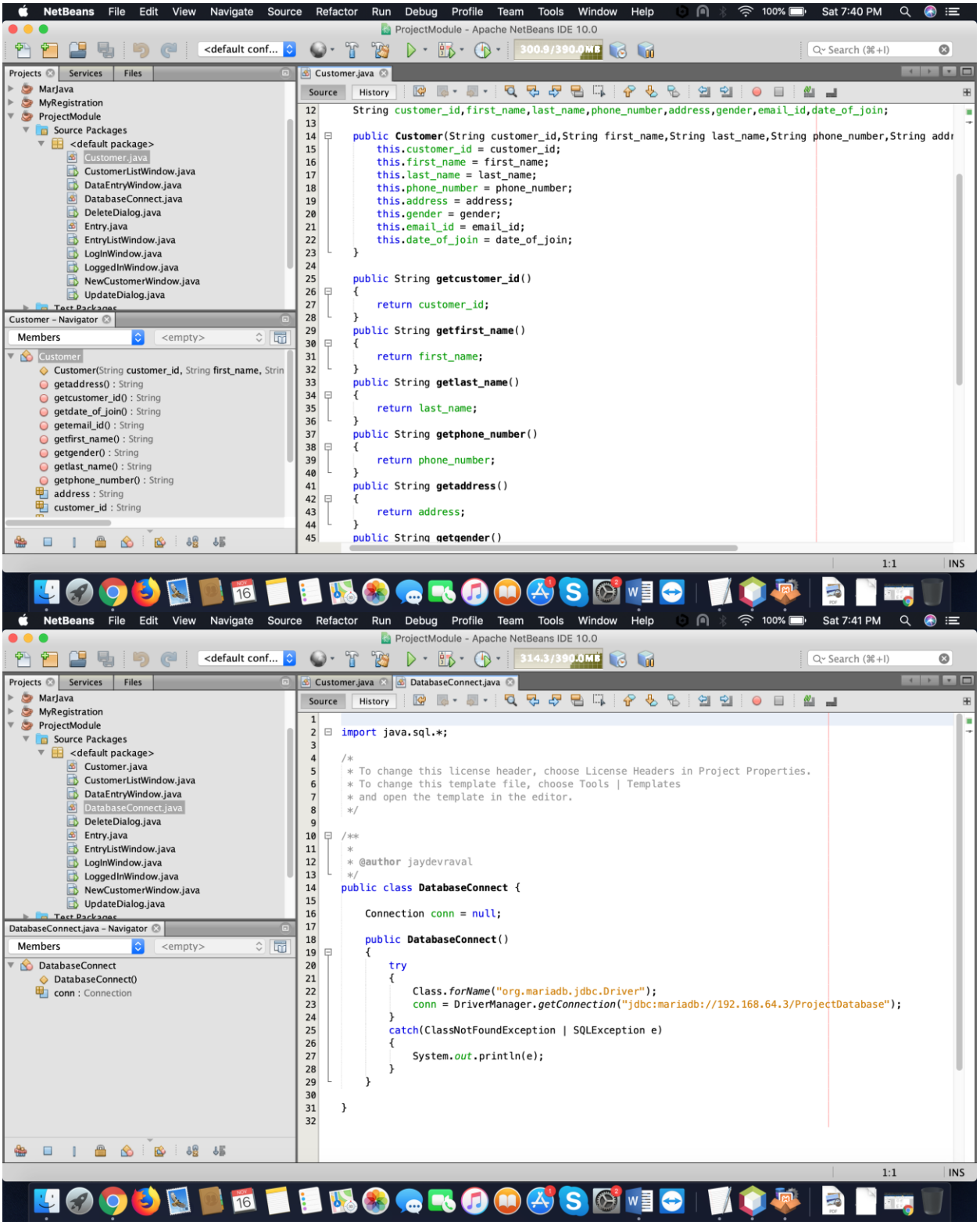
The risk involved when creating a new system is always to be taken into consideration. As we are trying to reduce the paperwork and have a better workflow with acknowledgment, chances for new unidentified challenges and risks may arise before or after implementation.
Security is one of the most important factors as to keep the privacy of user detail it is important to have a strict protocol for database access.
Officials in organisation will work on the client PC which will be connected to the server, so proper topology needs to be planned so that the chances of failure of the system reduces.
SQL injection and other DDoS attacks could harm the whole system, possibly leak the details of the user, so a strong firewall needs to be implemented for the requests.
Power Failure is also the parameter to be taken under consideration, a backup should be readily available.

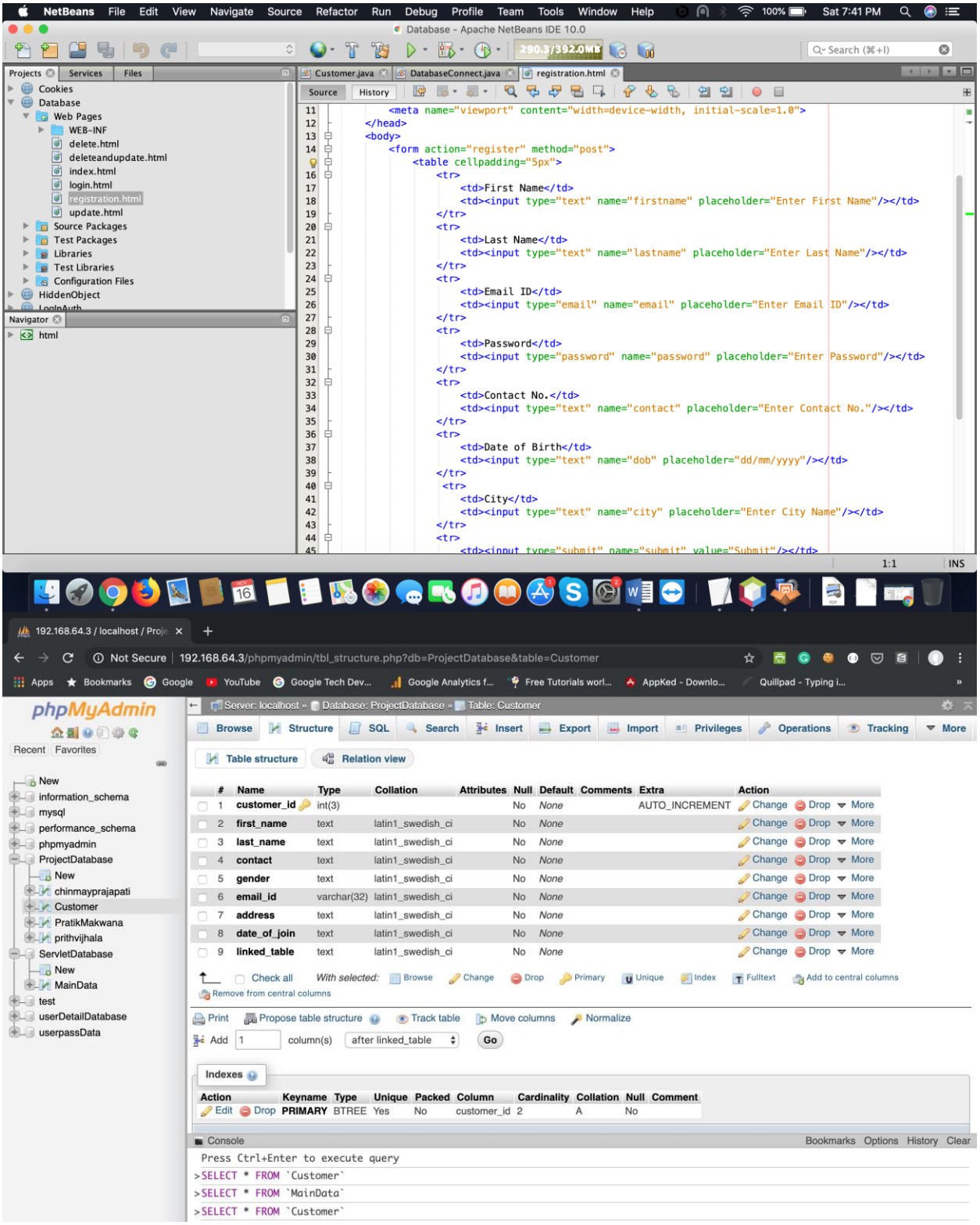
Possible outcome

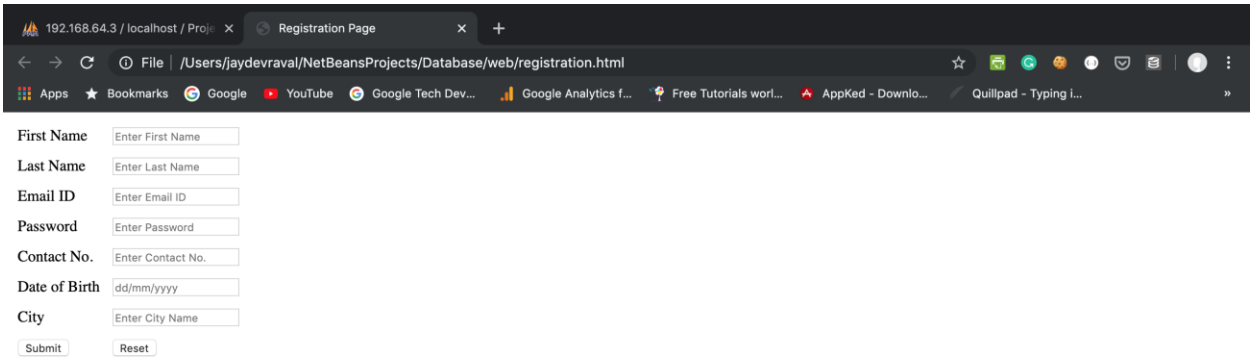
After the implementation of the file tracking system, overall productivity will increase, as a user will have track of the application and won't have to worry about the time. Using a website to upload scanned documents will also reduce the extra hard-copy for review purposes. Scanning of barcode also reduces the hassle to write down the whole unique ID. While maintaining a separate database, Only the admin will have rights to view and edit the master database, officials can only view the database, and the user should not be given rights to either read or write in the database. For updating the status for tracking, a separate database is maintained for a user to view, where official who reviews along with timestamp is displayed and shows further remaining steps. Such setup is very flexible as it can be easily implemented for any organisation. Further, it can easily be a part of a central warehouse, where past data can be achieved.

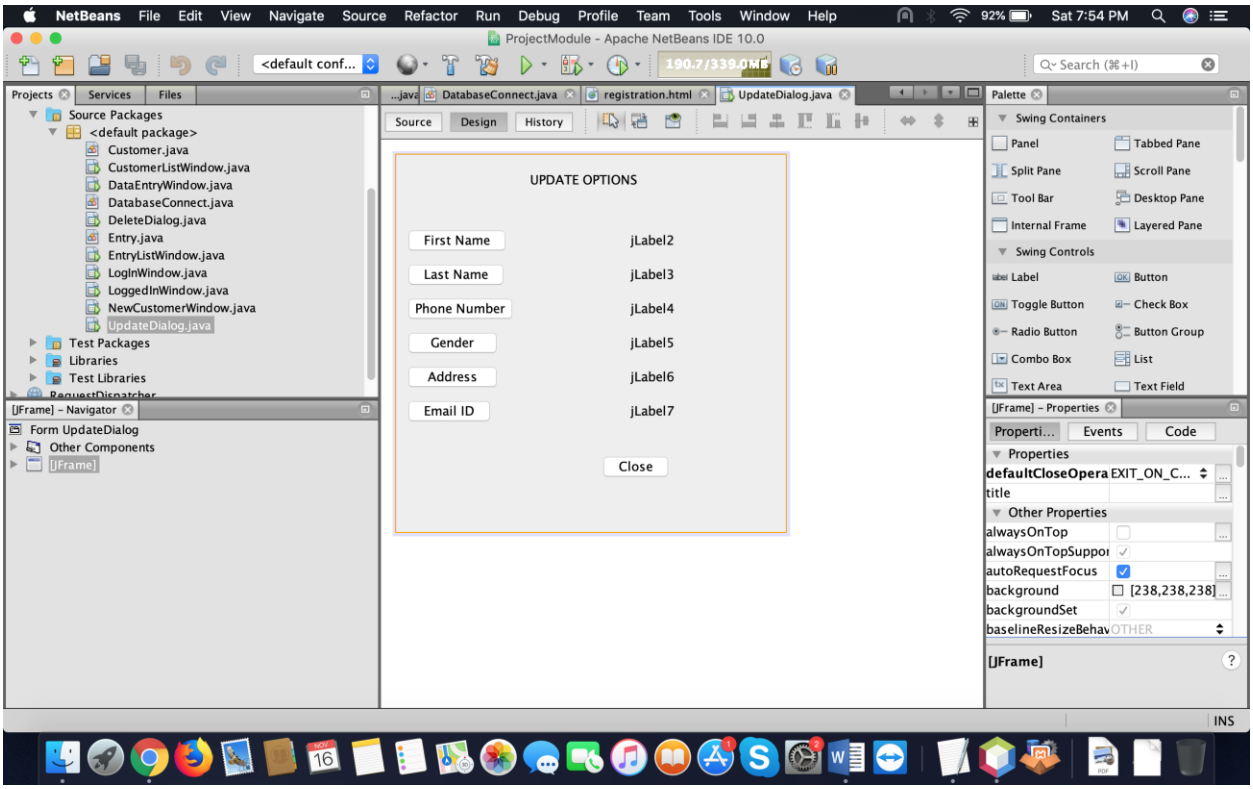
Work done till date

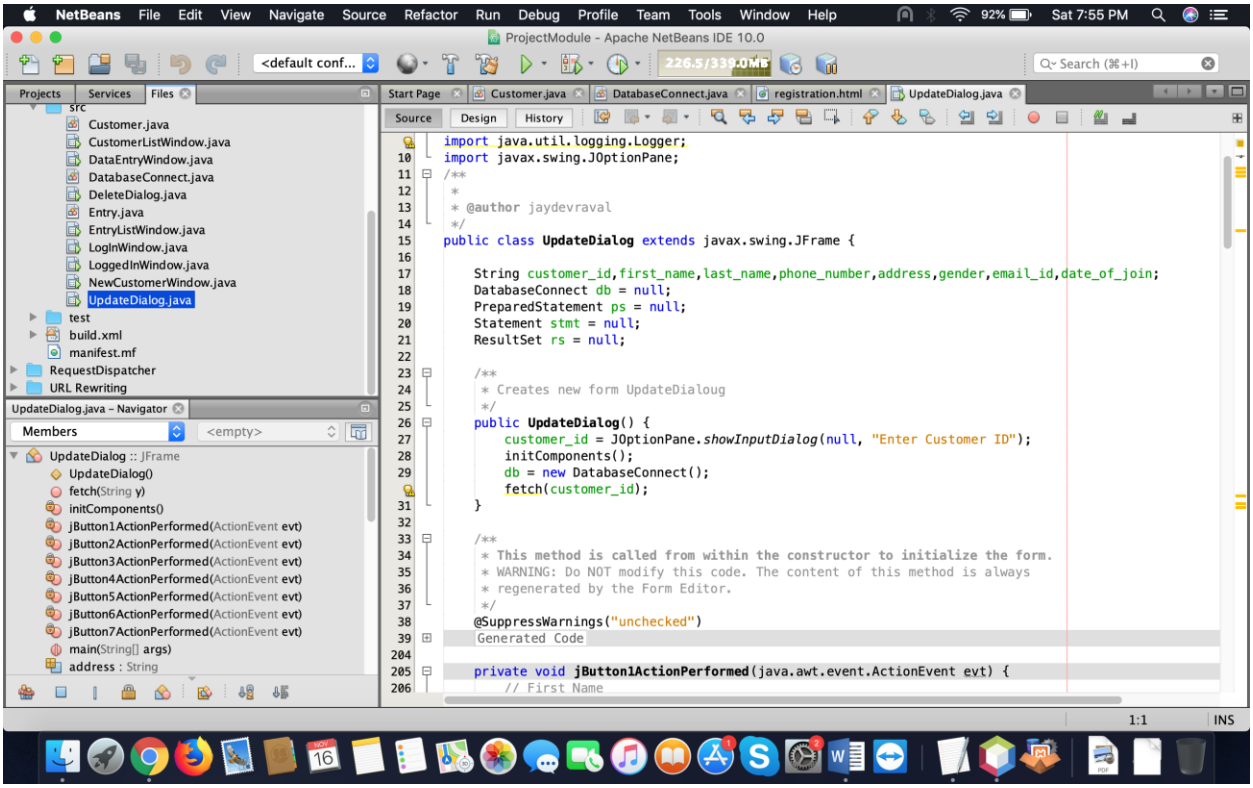
So far we managed to complete the database settings for all user request fields and to make a web-application for a registration form which will take user input and create an insert query to feed data in a table. Then after insertion of record, there will be an option to create a pdf file with PDF417 Code.
The database structure is set for user and functioning, now we have created a portal for the admin where he will have permission to update or delete a field, this is based on JFrame and the data is real-time show including edit option, so it very flexible to make changes to the database. Portal for officer is under progress, with backend code ready, only need to create another JFrame which will have functionality either to choose unique ID from drop-down, manually entering or scanning from the paper itself, and will have button to execute the operation on database with all the necessary parameters, which will be later on visible by user in website.


NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProjectModule - Apache NetBeans IDE 10.0
Projects Services Files
MarJava
MyRegistration
ProjectModule
Source Packages
<default package>
Customer.java
CustomerListWindow.java
DataEntryWindow.java
DatabaseConnect.java
DeleteDialog.java
Entry.java
EntryListWindow.java
LogInWindow.java
LoggedInWindow.java
NewCustomerWindow.java
UpdateDialog.java
Customer – Navigator
Members
Customer
Customer(String customer_id, String first_name, Strin
getaddress() : String
getcustomer_id() : String
getdate_of_join() : String
getemail_id() : String
getfirst_name() : String
getgender() : String
getlast_name() : String
getphone_number() : String
address : String
customer_id : String
Source History
String customer_id,first_name,last_name,phone_number,address,gender,email_id,date_of_join;
public Customer(String customer_id,String first_name,String last_name,String phone_number,String addr
this.customer_id = customer_id;
this.first_name = first_name;
this.last_name = last_name;
this.phone_number = phone_number;
this.address = address;
this.gender = gender;
this.email_id = email_id;
this.date_of_join = date_of_join;
public String getcustomer_id()
return customer_id;
public String getfirst_name()
return first_name;
public String getlast_name()
return last_name;
public String getphone_number()
return phone_number;
public String getaddress()
return address;
public String getgender()
1:1 INS
NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProjectModule - Apache NetBeans IDE 10.0
Customer.java DatabaseConnect.java
DatabaseConnect.java – Navigator
DatabaseConnect
DatabaseConnect()
conn : Connection
import java.sql.*;
/*
* To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.
*/
/**
*
* @author jaydevraval
*/
public class DatabaseConnect {
Connection conn = null;
public DatabaseConnect()
try
Class.forName("org.mariadb.jdbc.Driver");
conn = DriverManager.getConnection("jdbc:mariadb://192.168.64.3/ProjectDatabase");
catch(ClassNotFoundException | SQLException e)
System.out.println(e);
1:1 INS


NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Database - Apache NetBeans IDE 10.0
Projects Services Files
Cookies
Database
Web Pages
WEB-INF
delete.html
deleteandupdate.html
index.html
login.html
registration.html
update.html
Source Packages
Test Packages
Libraries
Test Libraries
Configuration Files
HiddenObject
Navigator
html
Customer.java DatabaseConnect.java registration.html
Source History
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<form action="register" method="post">
<table cellpadding="5px">
<tr>
<td>First Name</td>
<td><input type="text" name="firstname" placeholder="Enter First Name"/></td>
</tr>
<tr>
<td>Last Name</td>
<td><input type="text" name="lastname" placeholder="Enter Last Name"/></td>
</tr>
<tr>
<td>Email ID</td>
<td><input type="email" name="email" placeholder="Enter Email ID"/></td>
</tr>
<tr>
<td>Password</td>
<td><input type="password" name="password" placeholder="Enter Password"/></td>
</tr>
<tr>
<td>Contact No.</td>
<td><input type="text" name="contact" placeholder="Enter Contact No."/></td>
</tr>
<tr>
<td>Date of Birth</td>
<td><input type="text" name="dob" placeholder="dd/mm/yyyy"/></td>
</tr>
<tr>
<td>City</td>
<td><input type="text" name="city" placeholder="Enter City Name"/></td>
</tr>
<tr>
1:1 INS
192.168.64.3 / localhost / Proje
Not Secure 192.168.64.3/phpmyadmin/tbl_structure.php?db=ProjectDatabase&table=Customer
Apps Bookmarks Google YouTube Google Tech Dev... Google Analytics f... Free Tutorials worl... AppKed - Downlo... Quillpad - Typing i...
phpMyAdmin
Recent Favorites
New
information_schema
mysql
performance_schema
phpmyadmin
ProjectDatabase
New
chinmayprajapati
Customer
PratikMakwana
prithvijhala
ServletDatabase
New
MainData
test
userDetailDatabase
userpassData
Server: localhost » Database: ProjectDatabase » Table: Customer
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking More
Table structure Relation view
Name Type Collation Attributes Null Default Comments Extra Action
1 customer_id int(3) No None AUTO_INCREMENT Change Drop More
2 first_name text latin1_swedish_ci No None Change Drop More
3 last_name text latin1_swedish_ci No None Change Drop More
4 contact text latin1_swedish_ci No None Change Drop More
5 gender text latin1_swedish_ci No None Change Drop More
6 email_id varchar(32) latin1_swedish_ci No None Change Drop More
7 address text latin1_swedish_ci No None Change Drop More
8 date_of_join text latin1_swedish_ci No None Change Drop More
9 linked_table text latin1_swedish_ci No None Change Drop More
Check all With selected: Browse Change Drop Primary Unique Index Fulltext Add to central columns
Remove from central columns
Print Propose table structure Track table Move columns Normalize
Add 1 column(s) after linked_table Go
Indexes
Action Keyname Type Unique Packed Column Cardinality Collation Null Comment
Edit Drop PRIMARY BTREE Yes No customer_id 2 A No
Console Bookmarks Options History Clear
Press Ctrl+Enter to execute query
>SELECT * FROM `Customer`
>SELECT * FROM `MainData`
>SELECT * FROM `Customer`


192.168.64.3 / localhost / Proje
Registration Page
File | /Users/jaydevraval/NetBeansProjects/Database/web/registration.html
Apps
Bookmarks
Google
YouTube
Google Tech Dev...
Google Analytics f...
Free Tutorials worl...
AppKed - Downlo...
Quillpad - Typing i...
First Name
Enter First Name
Last Name
Enter Last Name
Email ID
Enter Email ID
Password
Enter Password
Contact No.
Enter Contact No.
Date of Birth
dd/mm/yyyy
City
Enter City Name
Submit
Reset


NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Sat 7:54 PM
ProjectModule - Apache NetBeans IDE 10.0
Projects
Services
Files
Source Packages
<default package>
Customer.java
CustomerListWindow.java
DataEntryWindow.java
DatabaseConnect.java
DeleteDialog.java
Entry.java
EntryListWindow.java
LogInWindow.java
LoggedInWindow.java
NewCustomerWindow.java
UpdateDialog.java
Test Packages
Libraries
Test Libraries
DatabaseConnect.java
registration.html
UpdateDialog.java
Source
Design
History
UPDATE OPTIONS
First Name
Last Name
Phone Number
Gender
Address
Email ID
jLabel2
jLabel3
jLabel4
jLabel5
jLabel6
jLabel7
Close
Palette
Swing Containers
Panel
Tabbed Pane
Split Pane
Scroll Pane
Tool Bar
Desktop Pane
Internal Frame
Layered Pane
Swing Controls
Label
Button
Toggle Button
Check Box
Radio Button
Button Group
Combo Box
List
Text Area
Text Field
[JFrame] - Navigator
Form UpdateDialog
Other Components
[JFrame]
[JFrame] - Properties
Properti...
Events
Code
Properties
defaultCloseOpera
EXIT_ON_C...
title
Other Properties
alwaysOnTop
alwaysOnTopSuppor
autoRequestFocus
background
[238,238,238]
backgroundSet
baselineResizeBehav
OTHER
[JFrame]
INS


NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProjectModule - Apache NetBeans IDE 10.0
Projects Services Files
src
Customer.java
CustomerListWindow.java
DataEntryWindow.java
DatabaseConnect.java
DeleteDialog.java
Entry.java
EntryListWindow.java
LogInWindow.java
LoggedInWindow.java
NewCustomerWindow.java
UpdateDialog.java
test
build.xml
manifest.mf
RequestDispatcher
URL Rewriting
UpdateDialog.java – Navigator
Members
UpdateDialog :: JFrame
UpdateDialog()
fetch(String y)
initComponents()
jButton1ActionPerformed(ActionEvent evt)
jButton2ActionPerformed(ActionEvent evt)
jButton3ActionPerformed(ActionEvent evt)
jButton4ActionPerformed(ActionEvent evt)
jButton5ActionPerformed(ActionEvent evt)
jButton6ActionPerformed(ActionEvent evt)
jButton7ActionPerformed(ActionEvent evt)
main(String[] args)
address : String
Start Page
Customer.java
DatabaseConnect.java
registration.html
UpdateDialog.java
Source
Design
History
import java.util.logging.Logger;
import javax.swing.JOptionPane;
/**
*
* @author jaydevraval
*/
public class UpdateDialog extends javax.swing.JFrame {
String customer_id,first_name,last_name,phone_number,address,gender,email_id,date_of_join;
DatabaseConnect db = null;
PreparedStatement ps = null;
Statement stmt = null;
ResultSet rs = null;
/**
* Creates new form UpdateDialoug
*/
public UpdateDialog() {
customer_id = JOptionPane.showInputDialog(null, "Enter Customer ID");
initComponents();
db = new DatabaseConnect();
fetch(customer_id);
}
/**
* This method is called from within the constructor to initialize the form.
* WARNING: Do NOT modify this code. The content of this method is always
* regenerated by the Form Editor.
*/
@SuppressWarnings("unchecked")
Generated Code
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// First Name
1:1
INS