UNIVERSITY OF THE PHILIPPINES
OPEN UNIVERSITY

MASTER OF INFORMATION AND COMMUNICATION STUDIES

Rey Lawrence L. Torrecampo

Asset Management System for Database Management Systems

Thesis Adviser:

Ria Borromeo
Faculty of Information and Communication Studies

30 May 2022

Permission of the classification of this academic work access is subject to the provisions of applicable laws, the provisions of the UP IPR policy and any contractual obligations:

Invention (I)	☐	Yes	or	☒ No
Publication (P)	☐	Yes	or	☒ No
Confidential (C)	☐	Yes	or	☒ No
Free (F)	☒	Yes	or	☐ No

Student's signature:

Thesis adviser signature:

University Permission Page

Asset Management System for Database Management Systems

"I hereby grant the University of the Philippines a non-exclusive, worldwide, royalty-free license to reproduce, publish and publicly distribute copies of this Academic Work in whatever form subject to the provisions of applicable laws, the provisions of the UP IPR policy and any contractual obligations, as well as more specific permission marking on the Title Page."

"I specifically allow the University to:

Specifically, I grant the following rights to the University:

a. *Upload a copy of the work in the theses database of the college/school/institute/department and in any other databases available on the public internet*

b. *Publish the work in the college/school/institute/department journal, both in print and electronic or digital format and online; and*

c. *Give open access to the work, thus allowing "fair use" of the work in accordance with the provision of the Intellectual Property Code of the Philippines (Republic Act No. 8293), especially for teaching, scholarly and research purposes.*

Rey Lawrence L. Torrecampo May 30, 2022

Signature over Student Name and Date

Acceptance Page:

This paper prepared by **REY LAWRENCE L. TORRECAMPO** with the title: **"Asset Management System for Database Management Systems"** is hereby accepted by the Faculty of Information and Communication Studies, U.P. Open University, in partial fulfillment of the requirements for the degree Course.

14 July 2023

Ria Mae H. Borromeo, Ph.D.
Adviser

(Date)

14 July 2023

Roberto B. Figueroa Jr., Ph D.
Program Chair

(Date)

.

Diego S. Maranan, Ph.D.
Dean
Faculty of Information and Communication Studies

July 20, 2023

(Date)

Biographical Sketch

Born and raised in Cainta, Rizal, Philippines, Rey Lawrence Torrecampo is a licensed electronics engineer who pursued Data Analytics. After graduating in Electronics and Communication Engineering from the Ateneo De Manila University, he spent most of his early career as an IT professional delving into multiple industries from healthcare to logistics.

His core competencies range from software and system development to data analytics, and database administration. His knowledge of multiple database platforms (MS SQL Server, MySQL, Postgres, and Prestosql) has contributed to adapting to any task that involves data migration, ETL data pipelines, and dashboard development for companies he worked with. With his passion for learning, he spent most of his leisure time learning new skills. In October 2021, he earned his pathway certification from DAP's SPARTA program. In the following year, he completed his master's degree from the University of the Philippines Open University.

Rey Lawrence is not just an office worker and individual contributor to his team; he also shares his expertise in data analytics to fifth-year students in his alma mater.

Acknowledgement

I am deeply grateful to all those who supported me with this journey. First and foremost, I would like to thank the faculty and staff of the University of the Philippines Open University for giving me the opportunity to learn from your institution. Second, I would like to thank my former employer, dbWatch for providing me the resources to make this project a success. Finally, I would like to thank my family for the never-ending support they provided.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Databases hold critical data that is valuable for both consumers and businesses. As abusiness expands, the reliance on a healthy and working database becomes imperative for its business operations. Hence, it is essential to maintain and monitor relational databases as they are the backbone for business longevity and growth.

This project aims to retrofit the concept of an inventory management system to database systems. The scope and result of this project are to (1) host the system in awebserver, (2) capability to connect to any of the popular relational database systems(SQL Server, Postgres, Oracle, or MySQL), (3) monitor a database's availability and capacity, and (4) aggregate and display historical data into graphs and tables.

The resulting system satisfied the requirements mentioned above. The system is hosted on AWS EC2 Windows 2019. Through the web interface, you can monitor andanalyze historical data. With the pyodbc library, it can Extract-Transform-Load data from the target database and then load it to the local database. This method is possiblewith the help of SQLs native for each target database.

Even though numerous database monitoring tools are on the market, this system is alow-cost alternative. Also, the system is still in its infancy. It suffered from project ailments that hindered the progress of this project, such as underestimating tasks, shortage of time, and delayed testing. In essence, the system still satisfies the benchmarks for testing, and it is subjected to future improvements.

Keywords: Databases, Database Monitoring Tools, ETL, Database Systems

Chapter I

THE PROBLEM DOMAIN

Statement of the Problem

Daily business operations and business intelligence rely on the availability of database management systems. Since databases hold critical data that is valuable to both the consumer and the business, they are the backbone of the modern business.

However, as modern businesses continue to expand and meet their customer's needs, relational database management systems also grow in scale. Companies are struggling to track and monitor the databases properly. Plus, businesses also need the tools to cope in maintaining and auditing their database systems.

In that regard, the problem for this research centers on tracking and auditing database management systems. As they are the most critical assets, we want to find alternative monitoring and tracing them.

Background and Objectives of the Project

The purpose of this research is to retrofit and repurpose inventory management systems to monitor and track RDBMS precisely and the servers where they are hosted. There are four questions we want to address:

1. How do we transform the existing Inventory Management System to monitor and audit RDBMS primarily?
2. Is there any related literature on inventory management that needs to be considered?
3. What best practices should we consider transforming an existing inventory system?

4. What essential RDBMS properties are relevant in tracking and auditing?

Significance and Scope of the Project

This document will only include the software specifications for an Inventory Management System specifically designed for database and server monitoring.

As stated previously, a management inventory system explicitly designed for databases is the goal of this project. It can connect to any major brands (MySQL, Oracle. MS SQL Server, and Postgres).

The proposed product is akin to a database monitoring tool that monitors and tracks database availability and capacity. The product will be named RDB IMS. However, this is where the similarity ends. The product will be hosted on a web server as a centralize The development will include features such as:

- capacity auditing
- network traffic analysis
- server availability
- real-time analytics
- Machine Learning Algorithm for automated and consistent uptime monitoring

Documentation of Existence and Seriousness of the Problem

There are two relevant studies that are relevant for this project.

1. *Web based intelligent inventory management system.*

   The web-based intelligent inventory management system follows perfectly the proposed client-server architecture of the system. In the architecture seen below (Figure 1), it has an input and Output Module, Web Services, Database and Intelligent System.

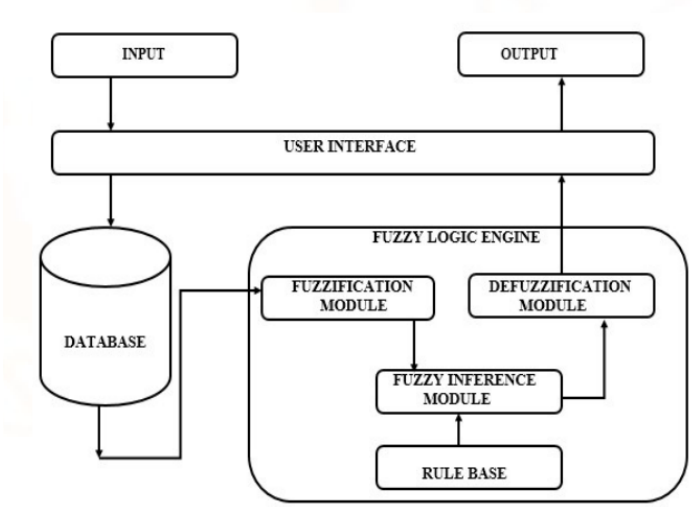


Figure 1: Web-based Intelligent Inventory Management System: Client Server Architecture

The input module provides a user login for different privileges and access restrictions while the Output Module enables users to view and monitor activity. The database, on the other hand, is a repository to store transactions and status of remaining stocks. Similarly, it has the intelligent system using fuzzy logic.All of the above are hosted in a web service for centralization and faster communication with other users. [1]

2. *An inquiry into machine learning-based automatic configuration tuning services on real-world database management systems*

   In a paper written by Aken and company, database performance tuning and optimization was done using machine learning. In their study, it resulted in a 45% increase in db time compared to the default settings.

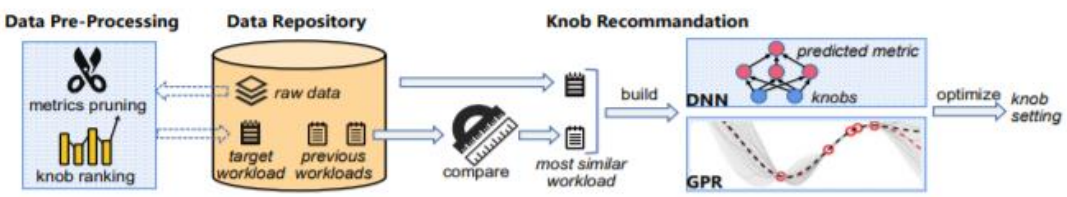


Figure 2: DNN Tuning Pipeline

Chapter II

REVIEW OF EXISTING ALTERNATIVES

All the existing systems are monitoring tools. For a detailed list, you can find it in the link provided. Below are the brief overview of the related systems, the similarity and differences to the proposed system and the rationale why the proposed system is a better alternative.

A. Site 24x7

Site 24x7 is a web hosted server that measures web traffic and server resources. It is marketed as an all-in-one monitoring solution that monitors web server, web site, network, and application. However, this monitoring solution does not include database monitoring which is an essential and main component of the proposed system. The goal of the proposed system is to have portability and easy connection to a centralized server. In that regard, it burrows the concept of a web server from Site 24 x7 with the added features of network and server resource monitoring.

B. PRTG Network Monitoring

PRTG Network Monitoring is a premium monitoring solution that tracks performance, load and services for Cloud application, Databases, Hardware and Network. For databases, it includes all four major platforms (MS SQL Server, MySQL, Postgres and Oracle). It also has monitoring features for Operating System cloud monitoring, service, and performance.

The monitoring tool is an application where it will be installed to your machine. Similar to the proposed system, it will need the network and server monitoring features but only the basics. However, the differences between the proposed system and this alternative

system is the cost and scope. It has comprehensive features that are outside the main features proposed earlier. Subsequently, the idea to use open-source software to minimize costs is another consideration for this project. The goal is to democratize database monitoring hence costing is a priority for this project without affecting the main the objectives.

C. Icinga

Icinga is a open-source monitoring tool that is only available for desktop. It can monitor servers and databases with minimum hardware requirement. It features Infrastructure Monitoring, Monitoring Automation, Cloud Monitoring, Metrics and Logs, Analytics and Notifications.

Similarly, the proposed system is in line with the philosophies and features of this product. As it is an open-source product then cost is the main selling point for it. Next, it monitors cloud servers and databases which is another key component of the proposed system. Finally, it has the added option for configurability since most the technologies used will have an open-source software.

The difference with Icinga is that it is community driven. Being heavily reliant to a community of developers has its pros and cons. But most likely, there is less control over the final product and a blurry vision from the initial specifications. This is a risk not worth taking.

D. dbWatch Control Center

dbWatch is a database-centric monitoring and management solution that covers multiple database platforms. It has the following features:

1. Database Monitoring
2. Database Management
3. Advanced Reporting
4. Database Maintenance Automation for SQL Server
5. Cluster support
6. Security package
7. Autodiscovery
8. Customization support package
9. Performance Management
10. Integrations
11. CLI scripting package
12. Multi-Site/Multi-Server support
13. Database Farm Management
14. Web dashboards
15. Auto-update

For similarities with the proposed system, it will only include the basics of database monitoring and management. The other modules will not be included such as reports, multi-server support and database cluster support. Furthermore, it will only limit it to database availability such as uptime and downtime statistics and database capacity such as growth rate and table space.

Another similarity is the seamlessness and non-intrusive nature to the operations of the targeted database. However, the software is installed in a machine. For the proposed system, it will be hosted on a server.

Chapter III

APPROACH TO BE TAKEN IN THIS PROJECT

Overview

A database Inventory Management System or DB IMS responds to the growingdemand to track and monitor relational databases. As a business today relies heavily on database management systems and operations, DB IMS aids in helping them keeptrack of all their database environments.

The system will use open-source technology and be hosted on a web server while monitoring relational databases. This is ideal for professionals who prefer an on-the-go type of monitoring. You log in, put in your database credentials, and the software will monitor your database availability and capacity.

Theoretical Framework

In this project, we will use the following theoretical framework for a functioning system:

- System Architecture - see Appendix A
- Entity Relationship Diagram - see Appendix B
- Use Case Model - see Appendix E
- Sequence Diagrams - see Appendix F

The default methodology for this project is a **Waterfall Model Methodology**.

Rational of Framework

A. System Architecture

The system architecture serves as the backbone for the proposed information system.

Appendix A shows the interaction with all the components inside the host server, user, and targeted databases.

Firstly, Input and Output are directly linked to the web interface. The web interface acts as the User Interface, and it displays the graphs, information, and connection details to the end-user. This serves as the output of the web interface. While the commands and interactions done by the user are the inputs. (See AppendixC for the list of web pages)

Similarly, two buffer scripts (Server Side Scripts and Database Connection) are connected between the local database to the web interface and the targeted database.The server-side script displays information from the local database to the webinterface. While the database connection script connects and interacts with thetargeted database and loads data to the local database. The database connection script will perform ETL on the database instance and load it to the local database.

It uses a stored procedure that will be installed inside the targeted database environment. This design was chosen to have little impact on the targeted database server.

B. Database Design

The Entity-Relationship Diagram (ERD, see Appendix B) intends to preserve data integrity and structure in the relational database management system (RDMS). This, in turn, prevents orphan records from being stored and normalizes data. Similarly,this acts as a filter when displaying data to the web interface.

First, a unique identifier is a primary key for the first two tables (users and

db_connection_details). A unique identifier is a more straightforward way to enhance security and join tables more accurately. Plus, adding a unique identifier will restrict access to other information in the system.

Next, each platform has separate tables allocated for them. Due to the nature of different database information, it is more convenient to separate data into different tables rather than consolidate them in one table and separate the information afterward.

Then, you have configuration tables and generic tables. Configuration tables are reference tables to know when to run a db ims job or track their status automatically. Generic tables contain all general values that can be used for the entire system.

Similarly, you also have the uptime table, which holds the uptime statistics. Youalso have the db_connection_logs, which holds status if the connection is ONLINE or OFFLINE.

Aside from those, the database has functions and stored procedures (AppendixM.1) to verify the integrity of the input data. SPs and functions are also used to store data in the local database. As for extracting data from the database, view tables are used. (See appendix M.2)

C. Use Case Models

The use case model shows the interactions between the user and the system. In Appendix E, it shows the multiple actions of the user and the corresponding resulting action of the system.

D. Sequence Diagrams

The sequence diagrams map out the from and to for each entity (user, web server, local database, and target database server). It defines how the data will be inputted and stored into the database. Not only that, but it also shows how data will travel between two entities.

For each scenario, the arrows points from the source entity (one doing the action) and target entity (one receiving the action). For each column, there is a duration represented by line bar for the action.

E. Waterfall Methodology

In choosing a methodology, the following questions should be answered: "who," "what," "when," "why," and "how." "Who" focuses on responsible teams and individualson activities that need to be done. "What" answers project elements that need to be achieved for the system to complete and work. "When" answers the timeline of deliverables so that the system will reach the target date based on the client and team's date of release. "Why" refers to activity dependencies, objectives of the project,and reasons for continuing the project. Finally, "How" describes the process ofimplementing the theoretical frameworks above.

When discussing methodologies, two models come into mind – waterfall or agile. The waterfall is more traditional, wherein it follows a linear software developmentapproach in sequential order and moves on to the next phase when development is complete. On the other hand, agile is a continuous and iterative development processwherein concurrent development and testing. The client has nearly negligible participation in the project in the

former, while the other has more exposure to the client.

Upon considering the project timeline (Appendix D), a more traditional approach like a waterfall should be implemented for this project due to the following points:

- minor changes to be done during the product life cycle
- smaller uncertainties and unknown variables are expected
- simplified system requirement means a more linear approach is morecompatible

Technologies you plan to consider or use

Listed below are the technologies used:

Components	Technologies/ Languages
Host Server	AWS EC2: Windows 10 (2019), 64bit, 30Gigabyte of ROM, 2.4 GHz
Web Services	Apache running in Xampp
Client-Server Interface	HTML, CSS, and Javascript
Server-side Scripting	PHP 8.0
Local Database	MySQL
Database Connection and Data Extraction	Python 3.8
Database data extraction	Native SQL (Oracle, MySQL, Postgres, and MS SQL Server)
Machine Learning	Python

Table 1: List of Corresponding technologies to be used

** Machine Learning was not implemented in this project due to the lack of data, time,and resources.

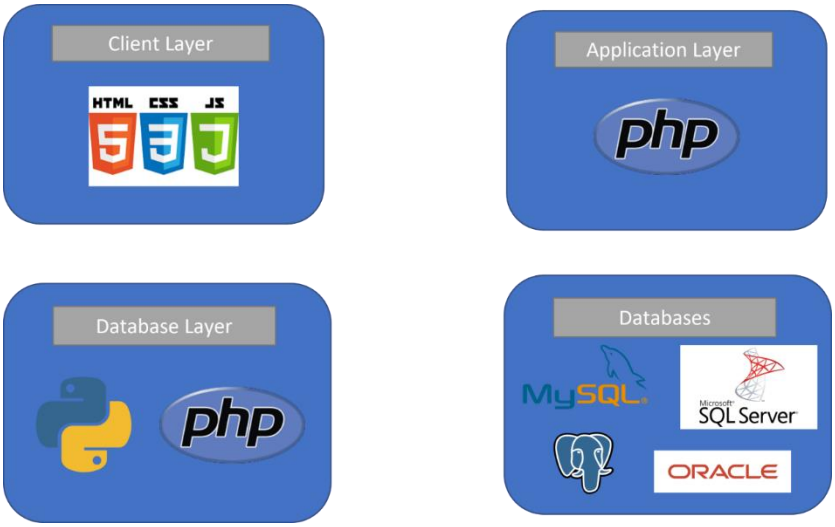


Figure 3: All Technologies to be used

1. *Host Server*

The web server is hosted in AWS EC2. AWS was chosen to accommodate theproposed testing plan, VPN connection to the dbWatch's network, and seamless transition between the integration environment (local pc) and the staging environment(AWS hosted). We are using a free tier to utilize the full capacity of AWS EC2, especially its high availability. Since we are working with a windows server, it's more natural to lean to AWS EC2.In addition, EC2 has an elastic IP and security group to whitelist all network connections. We can set the security group to limit root access only to my IP Address.

2. *Web Services*

Web services are still running on an Apache engine. The difference is that we are using Xampp to moderate the web server and database services. Both integrationand staging environment will be running on Xampp. When accessing the website, it'sseamless. For deployment purposes, xampp is much faster to deploy, and overwritingcodes is more manageable by opening an FTP connection between the server and the local pc. Also, Xampp's UI is much easier than killing and restarting services.

3. *Client-Server Interface*

The Client-Server interface uses Javascript, CSS, and HTML. By default, HTMLis the display format for the web interface. All HTML codes will be displayed on the client-side of the server. CSS is for formatting and stylizing HTML contents. Finally, Javascript is one of the most utilized programming languages used in the project. Fromanimations, adding HTML contents, graphs, and java queries, javascript has becomethe backbone to smoothening user experience.

4. *Server-side Scripting*

PHP is the go-to server-side scripting to access the local database and store them as variables for HTML. This style was used to reduce the strain in Javascript anddisplay pages statically. At most, PHP is used in sessions and local database queries.

Although it can query in the local database, the credentials set are only for the db_ims database.

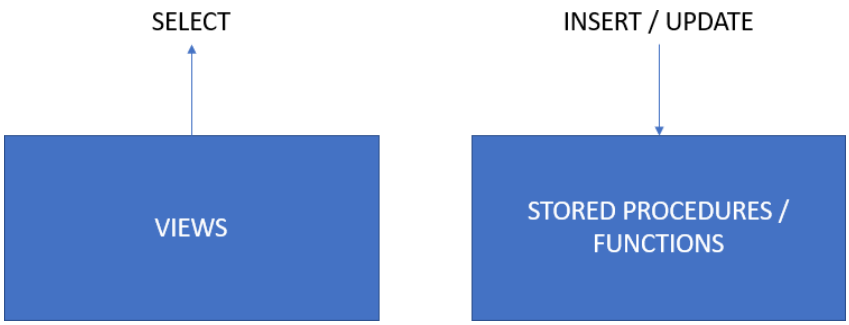


Figure 4: Data Display and Data Input

In addition, select statements are limited to views, while insert, delete and update statements are executed via a stored procedure.

5. Local Database

A MySQL database is hosted in the AWS EC2 instance. The database is running through XAMPP. MySQL was chosen as the database of due to the following reasons:

1) MySQL is an open-source platform that has been relatively long in the market. This project aims not to create a complex database system but to act as a storage unit for the local database. MySQL suffices in this aspect.
2) MySQL has all the functions, views, and stored procedures available even for a standard license. It is easier to restrict all transactions using View Tables, functions, and stored procedures for security purposes. Similarly, itis easier to do this on the command line.
3) Configuring MySQL databases is available on the internet. Since this projectaims to use open-source resources, it will be easier to implement it with a non-enterprise database such as MS SQL Server or a community-driven database system such as Postgres. It is also straightforward compared to the likes of Oracle.

6. Database Connection

Python is the most suitable program language to establish, maintain, and closeconnections to a database. Pyodbc uses the drivers available in windows. Later, we will discuss the implications of this structure. Python ODBC is better than JDBC because it is easy to implement, and it only requires a single library, making it platform-independent.

7. Native SQL

It uses an agentless monitoring procedure to capture database capacity and the remaining availability components. Stored procedures stores database data and information locally. A

python script from the server triggers this. Once data needs to be retrieved for analysis, it extracts the aggregated data back to the server.

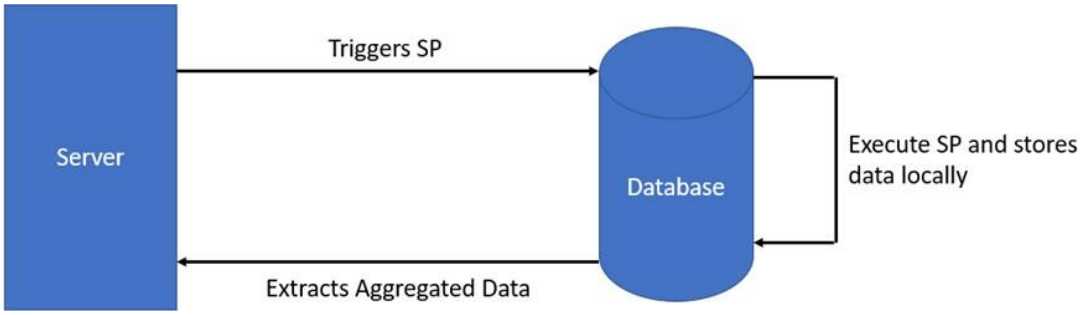


Figure 5: Data extraction, transformation and load to server and displaying of data to server

Several problems with using this process during implementation will be discussed later in the discussion section.

Chapter IV

CHAPTER PLAN

CONCEPT

The system architecture design will follow that in Appendix A. As discussed in the **Technologies you plan to use** section, the system architecture will be discussed with the following components:

A. Input and Output

The input describes the command interactions done by the user while output are the graphs, information and connection details displayed in the web interface for the end user.

B. Web Interface

The web interface acts as the user interface between the system local database and scripts running behind the host server. To see the initial web pages for the project, check appendix C for the Mockup of the web pages. This will be discussed later.

C. Server Side Script

This is the connecting script between the local database and the web interface. It provides information for database availability, error logs and aggregated data extracted from the targeted database.

D. Database Connection

The database connection captures availability and errors, maintains connection and loads information to the local database. It employs the ODBC library in Python. Similarly, when the database establishes its first connection, it will install several stored procedure.

E. Native SQL

In the same diagram, the targeted database will capture capacity information such as memory, table space check, log file statistics and growth rate. Using a stored procedure that will be installed inside the targeted database environment along with creating a database and schema, it will store capacity data locally. The reason for this setup is to lessen burden and storage for the web server.

F. Machine Learning

Machine learning algorithm will act as the database administrator for the database management system. Using one of the performance tuning algorithms defined by Aken and company, the implementation of machine learning will help in doing performance tuning for a single database system. The machine learning algorithm is also in charge of executing the stored procedures found in the target database. Due to time constraints, this was not implemented in the final design.

G. System Features

The sequence diagrams map out the connection for each entity (user, web server, local database, and target database server). It defines how the data will be inputted and stored into the database. Not only that, but it also shows how data will travel between two entities.

In Appendix C, the web pages are shown. These are:

- **Home Page** - displays the home screen when accessing the website. It provides information on the project.
- **Project Page** – illustrates how the system works.
- **Sign Up / Login Page** – access the system by creating an account oraccessing your account through Email or Username and your Password
- **Change Password** – password recovery page; accessed by clicking “forgot password” and opening the recovery email. Afterward, the userwill be redirected to the password

recovery page.

- **Home Page** – Once logged in, the user will be redirected to the homepage. This shows the notification widget, inventory graph, and quick access information tables.
- **CONFIGURE > ADD** – step-by-step process in connecting to a database. First, fill out the connection form; then, the summary will be prompted. After confirming, the system will connect to the database. After establishing a connection, it will ask for a "connection name." Once all are done, it will save the connection details in the database.
- **CONFIGURE > EDIT** – displays the list of connections. You can change the connection name or database details on this page. After refreshing the page, the changes will be re-displayed.
- **CONFIGURE > REMOVE** – removes a database connection from the list of database connections.
- **MONITOR > [CONNECTION]** – displays DB IMS jobs and installation status of DB IMS jobs. You can run jobs manually on this page. Db IMSjobs will differ with the platform.
- **INVENTORY –** shows the list of active databases currently being monitored.
- **Analyze > Environment** – displays an inventory graph and version details of the database you are currently monitoring.
- **Analyze > [CONNECTION]** – shows data extracted from the database you are monitoring. Data displayed will differ per platform.
- **LOGS** – shows python script logs thru iframe
- **Notification** – shows user notifications
- **Search Bar** – queries for connection names and list them
- **My Profile Page** – shows an editable form for user name and Password
- **Log Out** – destroys current session and redirects to index page

METHODS

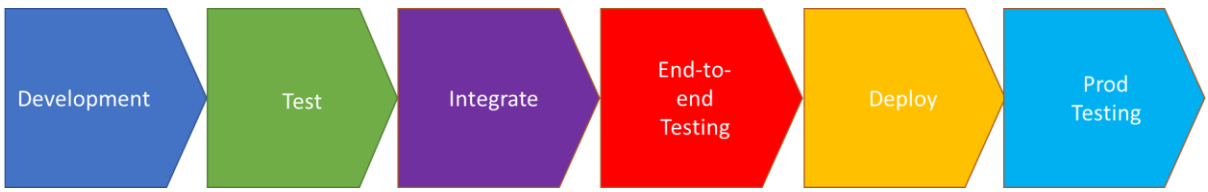


Figure 6: Waterfall Methodology of Project

This project will observe the above straightforward methodology. For the Development Phase, it will include each component: ODBC connector, Data Collector Script, Stored Procedure Scripting, PHP Scripting, Machine Learning Algorithm and Web Page Development. For every component developed, then component testing will be done. In addition, for every component completed then it will be uploaded in github with a corresponding version. This is to have reliable backup during the development process and with a version control.

Once all components are complete, it will be integrated and an end-to-end testing will ensue. If the system is stable and no potential errors can be found then it will be deployed to the web server. Otherwise, it will remain in the local environment until all bugs and fixes are done.

When the system is deployed in the web server, another testing phase will commence - Prod Testing. The prod testing will be a high-level testing where potential bugs should be intercepted and negate the probability of failure. In essence, only two environments are available: local environment and server environment.

Plan for User Testing and Project Assessment

A. Scheduling and Implementation

Since this project will only involve two environments, then the test and deployment will be limited to only the local environment and web server environment. For the full breakdown, It follows the Gantt chart shown in Appendix D.Source code for the system can be found in Appendix K. The breakdown of files can be found in Appendix M.

For the implementation of the system, the first three months were dedicated to developing the system. Afterward, two weeks were allotted for testing. Then, the last week was devoted to documentation.

B. Project Assessment and User Testing

1. *Test Plan*

As discussed in the test plan, testing is done in two environments composed of Ui Testing, White-box testing, and functional testing. The first environment involves the local environment with databases within my machine (local database 127.0.0.1) and other approved database instances. All instances for testing can be found in Appendix G.1

The local environment is running in Xampp, and it is accessed using Google Chrome and Microsoft Edge. I connected to the company's network through a VPN toconnect to each approved database. *Approved databases* are the databases green-litby the owner for testing purposes, while those with *Local databases* are databases running on my local machine.

Testing includes all the following scenarios, as seen in Appendix H.1. For a complete list of test scenarios, click on the link.

Next, the staging server is used for staging environment testing. A staging environment is provisioned and hosted in AWS. Similarly, it uses Xampp, but it’s hosted in a webserver running on SSL. Twenty-four databases is used from dbWatch's testing environment, and all of them are in the company’s network. It is ideal to have this many databases to see if there will be a different response to different database versions. The list of databases for

the staging environment can be found in Appendix G.2.

Similarly, the same test scenario is applied to the second testing phase. (See Appendix H.1 for details) In addition, all databases used are approved databases for this testing phase.

2. *Finding Bugs*

The expected results list of test scenarios is used when tracking bugs. In appendix H.2, the tabulated format of the test case shows the action, input data, and expected result. Any deviation from the “expected result” is marked as a bug. The test scenarios are influenced by Sequence Diagram (Appendix F) and the Use case diagram(Appendix E). Bug Severity is categorized into low, medium, and high. Categorization of bugseverity can be found in Appendix H.3

3. *Test Specifications*

Test results can be found in Appendix J. There were two attempts when testing the integration environment. The first one (Appendix J.1 and J.2) failed since there were a lot of bugs uncovered during the testing phase. (See Appendix I) The second attempt was made after fixing all the bugs. A re-test was done and produced more accurate results. (Appendix J.3 and J.4)

Then, testing was done in the staging environment. All bug fixes were migrated to the EC2 machine before testing commenced. Like the integration environment’s second attempt, most of the features were functioning. (See Appendix J.5 and J.6)

4. Bug Triage

There are 5 Major defects found. (See Appendix I).

Special characters are omitted for the first bug (Appendix I.1) when registeringfor a new account. This was more on the problem with Mail gun rather than Python. Iremoved all special characters in the Stored Procedure when generating a new password to fix this.

For the second bug (Appendix I.2), “Input Valid Email” returns inputting a duplicate email. Bug fixes include adjusting the functions related to email validation and adding an indicator that if the output code is 2, it produces a “duplicate email” message.

For the third bug (see Appendix I.3), below are the bugs found:

- Redirect to Login Page After submission
- Subject Line not found in the recovery email
- No Email Sent or Confirmation.
- Password Recovery should work; after submitting a newPassword, it does not save in DB level

Upon checking, the problem was that the PHP file was invoking the wrong javascript and python files. All the issues listed above were resolved by changing file pointers (of both JS and Py files).

The fourth bug (Appendix I.4) requires adjusting the python script for initial installation. Several functions were revised, and after that, it worked.

Finally, the javascript for inventory charts was adjusted in the last bug (Appendix I.5). So as the select statements of the home page and analyze_inventory page. Also,two views were

added to accommodate the missing MySQL and Postgres tables.

5. *Regression Testing*

At the bare minimum, regression testing was done to verify if the system was notaffected by any code changes. All parts are moving, and confirming if the system is working should be a high priority during any code change before setting the bug fix asresolved.

6. *Security Testing*

Using the ZAP application, we were able to get the following alerts:

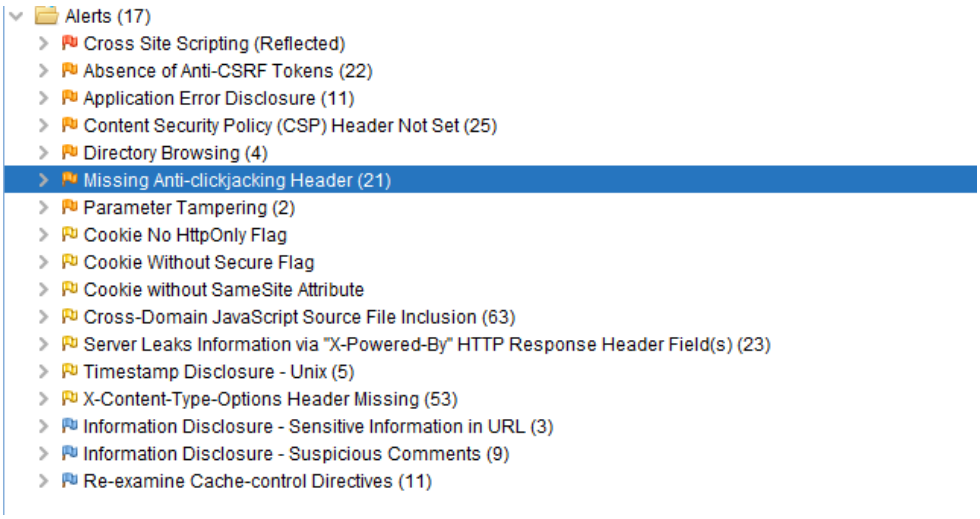


Figure 7: Network Testing Results

The complete list of Security Vulnerabilities can be found in Appendix J. But for discussion, the table below summarizes the security vulnerabilities:

	Security Vulnerability
1	Cross Site Scripting
2	Absence of Anti-CSRF Tokens
3	Application Error Disclosure
4	Content Security Policy (CSP) Header Not Set
5	Directory Browsing
6	Missing Anti-clickjacking Header
7	Parameter Tampering
8	Cookie No HttpOnly Flag

9	Cookie Without Secure Flag
10	Cookie without SameSite Attribute
11	Cross-Domain JavaScript Source File Inclusion
12	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
13	Timestamp Disclosure - Unix
14	X-Content-Type-Options Header Missing
15	Information Disclosure - Sensitive Information in URL
16	Information Disclosure - Suspicious Comments
17	Re-examine Cache-control Directives

Table 2: List of Security Vulnerabilities

Below are the future tasks to address the security vulnerabilities:

1) Cross-Site Script involves adding a script at the exposed part of the system's URL. You can do this by starting with a script identifier (<script>) and appending a JavaScript in between. Based on the check result, the exposed part is the exposed query selector (qres). Mitigating the exposed part will create another section and remove the query selector from the URL. All back-end processes will be done using Javascript and removing the header location in the PHP query file.
2) From what the evidence suggests and from research, implementing a synchronizer token pattern prevents unwanted sessions from getting critical information. [3] A redesign of the PHP files will be needed to accommodate the token submission.
3) The system does not have any error handling for Javascript or PHP executions. Adding that crucial try-catch statement will be necessary, so it does not disclose any sensitive information.
4) This was missed during the development of the system. A content security policy will be added in both the bg_background.php and bg_dashboard.php. [4] Both PHP files display the contents as HTML.
5) Since the server is running in Apache in Xampp, the same process of disabling the list of directories will be followed. This means changing httpd-vhosts.conf to

add the following Line [6]:

```
<Directory /{YOUR DIRECTORY}>
Options FollowSymLinks
</Directory>
```

6) Like number 5, the content security policy will be added. In addition, x-frame options will be modified in the logs.php file.

7) Same as number 3, an error-handling statement will be implemented to redirect pages to page 404.

8) ZAP suggests ensuring that the HTTP Only flag is set for all cookies. [7] This is done by setting the cookie to accept HTTP requests, thus blocking javascript from accessing the cookie. Cookies will be set in PHP files such as home.php, etc.

```
setcookie($name, $value, [
    'expires' => time() + 86400,
    'path' => '/',
    'domain' => 'domain.com',
    'secure' => true,
    'httponly' => true,
    'samesite' => 'None’,
]);
```

9) Same as number 8.

10) Same as number 8.

11) This is caused by accessing untrusted javascript files. A solution to this is to use sub source integrity [9]:

```
<script src="https://example.com/example-framework.js"
integrity="sha384oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQlGYl
1kPzQho1wx4JwY8wC"
crossorigin="anonymous">
</script>
```

12) Remove php exposure in php.ini file in xampp:

```
expose_php = off
```

13) Remove time-zone disclosure in files css/bootstrap.min.css and lib/tempusdominus/js/moment-timezone.min.js

```
<meta content="text/html; charset=UTF-8; X-Content-Type-Options=nosniff" http-equiv="Content-Type" />
```

14) Add the following to the HTML file [10] or add this in the Apache Web Server

```
<IfModule mod_headers.c>
   Header set X-XSS-Protection "1; mode=block"
   Header always append X-Frame-Options SAMEORIGIN
   Header set X-Content-Type-Options nosniff
</IfModule>
```

15) Alter Javascript and PHP so it does not return sensitive information over theURL.

16) Alter Javascript so that it does not return sensitive or user information over theconsole.

17) Adjust cache-control over the header [11]

```
<meta http-equiv="Cache-control" content="public">
```

Chapter V

RESULTS AND DISCUSSION

A. What I learned from this project

In this project, I learned several key lessons. First, overestimation is determinantal to a project's success. Before starting this project, I submitted a strict timeline and the system's initial design. I failed to consider the actual man-hours, information, and resources necessary for the system. I should have realized that building this system would need an entire team to check and balance the software andmethodology. Doing it alone, I was clouded in my judgment and had difficulty adjustingsome features.

Second, the timeline is too rigid. It did not consider the number of hours of research and trial and error; this further pushed the development period much later. For instance, the initial structure of the code is to install a stored procedure, trigger thesaid stored procedure, extract data, and transfer it to the local database. There are several problems with this design that will be expounded on later.

Another example, I forgot to factor in research, testing the script, and how the scripts will function per version. Upon further investigation, there are different ways todo monitoring and extract data. It takes a longer time to test and verify each suggestion. Not all SQL native languages are universally similar. Getting the agent status in SQL Server 2019 may differ from getting it in SQL Server 2005. Some databases are structured differently for each version. This was not considered during development. Thus, legacy versions will need a different approach.

Third, although it's logical to integrate Python, PHP, Native SQL, and Javascript for the system's backbone, it was harder to develop. The approach is different for eachprogramming language, and cross-applying one concept to the other is overwhelming.There are moments that

I’m stuck with Javascript or PHP, and it had to be remedied with overreliance on Python and Native SQL. But since most are moving parts, I cannot pinpoint where the problem occurs. Plus, the skillset for PHP and Javascript are different from my expertise.

There were times when working scripts and codes were also altered multiple times. Regression testing was done in conjunction with adding different platforms to verify that the script does not change the intended output of the program. In addition,many optimizations in the python scripts were done to negate the slowness and confusion in reading the code. Changes done were branching If-then statements wereconverted to one-line formats. Then, Try-catch statements were also re-structured to avoid redundancy. Finally, database timeouts were also added to the script, but later it was found out that these timeouts were not implemented correctly. Since it will affectthe system, it was postponed and not fixed.

Finally, the ETL structure might have been a significant factor in system development. Below are the major problems I found when using both the pyodbc library and developing the database python and SQL scripts:

1) Pyodbc library has an inherent design flaw; it uses odbc drivers installed in your windows machine. This is problematic since you must have a pre- installed version of the drivers to function correctly; otherwise, you need to add new drivers.
2) Development was too focused on one specific version and configuration per platform. This resulted in the following problems:
    a. SQL Server
        i. SQL Server drivers can only accommodate any SQL Server instance 2012 and after. Pyodbc cannot connect to earlier versions.
        ii. Some parts of the SQL Server stored procedure are designed to run for version 2014 and above. The SP will not run if the version extracts

information from an earlier version. Affected states are Agent Status.

iii. If the given credentials cannot access the master database or other database information table, the sp_installation job will fail.The system can still run the python file, but it will not produce any result. For example, if the credentials have no access to information schema, it becomes a significant blocker as the system will need those checks.

b. Postgres

i. Stored Procedures are only added after version 11. The earlierversion only used functions. This is problematic for earlierversions of Postgres as the system explicitly defines it as a “stored procedure” and not as a “function.”

ii. If the given credentials cannot access the Postgres database or other database information table, the sp_installation job will fail. The system can still run the python file, but it will not produce any result. For example, if the credentials have no access to information schema, it becomes a significant blockeras the system will need those checks.

c. MySQL

i. If the given credentials cannot access the MySQL database or other database information table, the sp_installation job will fail.The system can still run the python file, but it will not produce any result. Most approved databases have limited credentials, so accessing MySQL database is quite impossible even when creating a new user. For example, if the credentials have no access to information schema, it becomes a significant blockeras the system will need those checks.

d. Oracle

i. A major problem with Oracle is that it connects to the local oracle

database before accessing the target database. This produced errors in extracting information from Oracle.

ii. Oracle is also driver-sensitive. Connecting an Oracle 19 driver to an Oracle 12 version fails. But, connecting the same driver oracle 19 driver to Oracle version 19 also fails. An XE driver was considered, but it produced the same error.

iii. Also, failing to verify this later forced the developer not to investigate why these bugs are prevalent in the system.

B. Maintenance Plan

The system adopts the proposed maintenance plan found in the SRS. All codes will still be backed up in GitHub, and there will be a constant backup of data on the serverevery week.

Chapter VI

CONCLUSION

With the growing need for database monitor, this project aims to develop and implement a database inventory management system. As discussed earlier, the system should be hosted in a webserver and hold RDBMS connection details. Once connected, they should automatically monitor a database instance's availability and capacity. The system should also be readily available for the end-user to analyze. It should also neatly display historical data and graphs in the web interface.

The developed system, named DB-IMS, fulfilled those requirements. In addition,this project pushed the boundaries of developing a system that utilizes open-source technology. First, HTML is the language for the web interface embedded in PHP files.Tables and historical data are displayed in HTML format for the user to check and analyze their database state quickly.

Next, PHP and JavaScript are server-side scripts that extract and manipulate data and then display them in the web interface. PHP is used to extract, transform andload data in the database and the web interface. On the other hand, Javascript contributed to validating input data, web page animations, query data directly, and displaying graphs. Javascript and PHP work in parallel to produce a color interaction between web pages and databases.

Then, Python accesses and performs ETL on the target databases. Python scripts mostly do back-end processes. It installs stored procedures in the target databases, extracts database data, transform it, and loads it into the local database. The python script is integral in the ETL process as it's the direct link between the system and the target database.

Finally, SQL is the database language to extract and manipulate data from thedatabase instance. SQL is the native language of databases. Both the local databaseand target database utilize SQL to extract data. SQL is also the transactional languageon the local database to manipulate and extract data.

These technologies harmoniously work together to automate the extraction anddisplay them for the user's convenience. However, it’s not a perfect system. It is still under thorough reevaluation and realignment. It encountered multiple security threats,UI enhancement, error-handling improvements, database connection fixes, redesign of approach to legacy databases, and expansion of features.

Although other systems out in the market can do more than the current system's limitations, the system is still in its infancy. It’s also apparent that underestimating tasks,lack of supporting cast, shortage of time, and late testing played a massive factor in completing this project.

Even though the system has reached a good benchmark as the system has satisfied all testing scenarios, the improvements mentioned above should also be considered for further development.

Chapter VII

RECOMMENDATIONS

For future development, the priority is to fix all bugs, extend research ondatabase functionality and address the security vulnerabilities. Improving the system'sfunctionality and UI will be at the forefront of any future work. With a well-functioning system and smoother UI, then it will entice users to use the system.

Next, when data is sufficient, adding the missing feature – Artificial Intelligence,will be the next order of business. Artificial Intelligence requires a lot of data and timeto develop, verify and test. The system is already designed for Artificial Intelligence expansion as response time is recorded for databases. With a month’s data and moretime, this feature will be possible to implement.

Finally, the next logical step is to adjust the webpage to accommodate mobile users. The system is currently designed for desktop applications. Expanding it to mobile will benefit the philosophy of being on the go.

REFERENCES

[1] A. O. Madamidola, O. A. Daramola, and K. G. Akintola, "Web-based intelligent inventory management system," *International Journal of Trend in Scientific Research and Development*, vol.1,no. 4, May 2017 [Online]. Available:https://www.researchgate.net/publication/317276986_WEB_-_BASED_INTELLIGENT_INVENTORY_MANAGEMENT_SYSTEM. [Accessed: 12-Jan- 2022]

[2] D. V. Aken, D. Yang, S. Brillard, A. Fiorino, B. Zhang, C. Bilien, and A. Pavlo, "An inquiry into machine learning-based automatic configuration tuning services on real-world database management systems," *Carnegie Mellon Database Research Group*, 01-Mar-2021. [Online]. Available: https://db.cs.cmu.edu/papers/2021/p1241-aken.pdf. [Accessed: 12-Jan-2022].

[3] "Cross-site request forgery prevention cheat sheet," *Cross-Site Request Forgery Prevention -OWASP Cheat Sheet Series*. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html. [Accessed: 25-May-2022].

[4] "Content security policy (CSP) - http: MDN," *HTTP | MDN*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP. [Accessed: 25-May-2022].

[5] "X-Frame-Options - http: MDN," *HTTP | MDN*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options. [Accessed: 25-May-2022].

[6] Z. Banach, "How you can disable directory listing on Your web server – and why you should," *How to disable directory listing on your web server | Netsparker*, 14-Mar-2022. [Online]. Available: https://www.invicti.com/blog/web-security/disable-directory-listing-web-servers/#apacheserver. [Accessed: 25-May-2022].

[7] T. A. Nidecki, "The httponly flag – protecting cookies against XSS," *Acunetix*, 24-Aug-2020.[Online]. Available: https://www.acunetix.com/blog/web-security-zone/httponly-flag- protecting-cookies/. [Accessed: 25-May-2022].

[8] "SameSite cookies - http: MDN," *HTTP | MDN*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie/SameSite. [Accessed: 25-May-2022].

[9] "Subresource integrity - web security: MDN," *Web security | MDN*. [Online]. Available:

https://developer.mozilla.org/en- US/docs/Web/Security/Subresource_Integrity#browser_compatibility. [Accessed: 25-May- 2022].

[10] “X-content-type-options - http: MDN,” *HTTP | MDN*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options. [Accessed: 25-May-2022].

[11] “Cache-Control - http: MDN,” *HTTP | MDN*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control. [Accessed: 25- May-2022].

APPENDICES

Appendix A: System Architecture

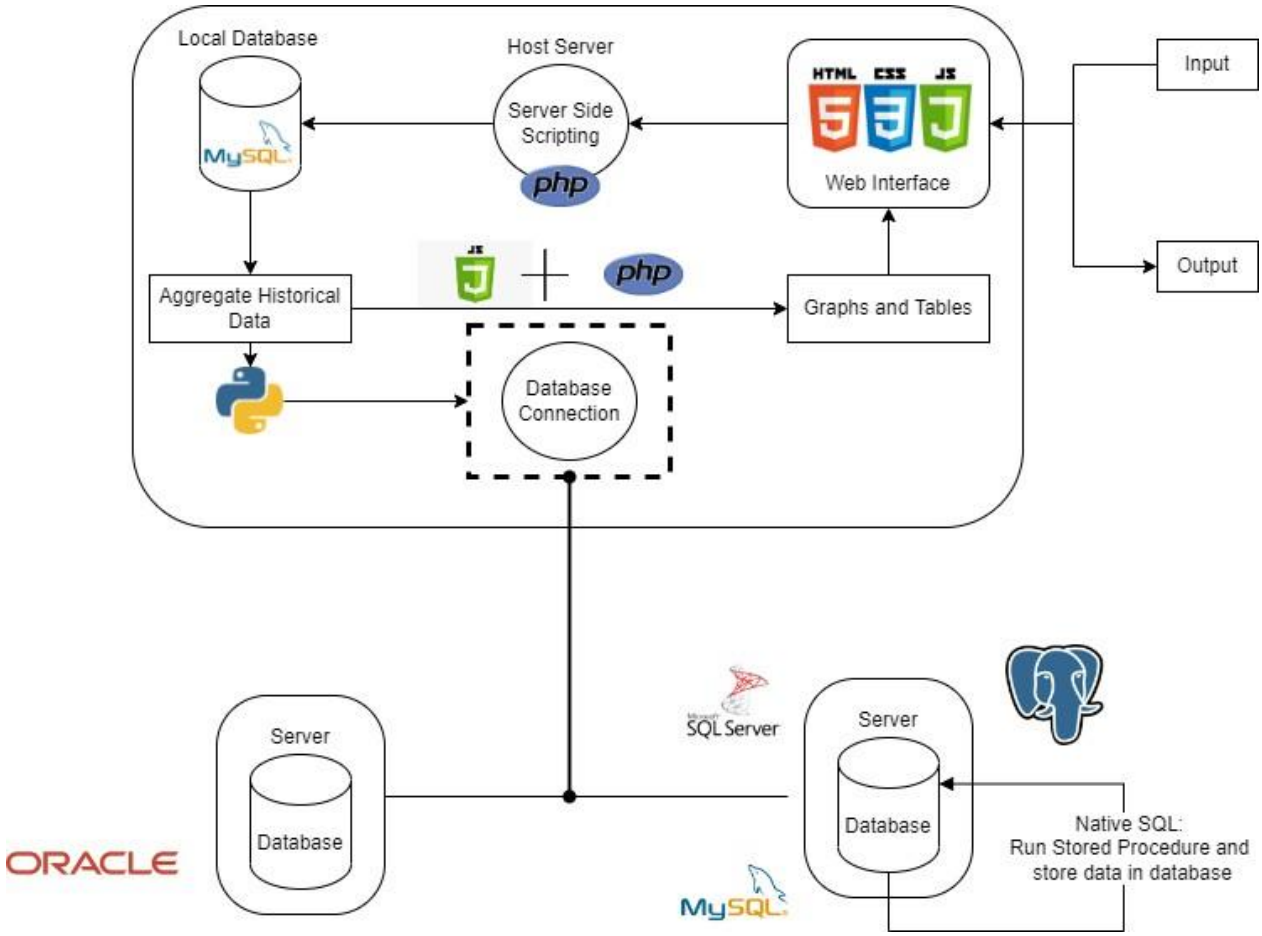


Figure 8: System Architecture

Appendix B: Entity Relationship Diagram

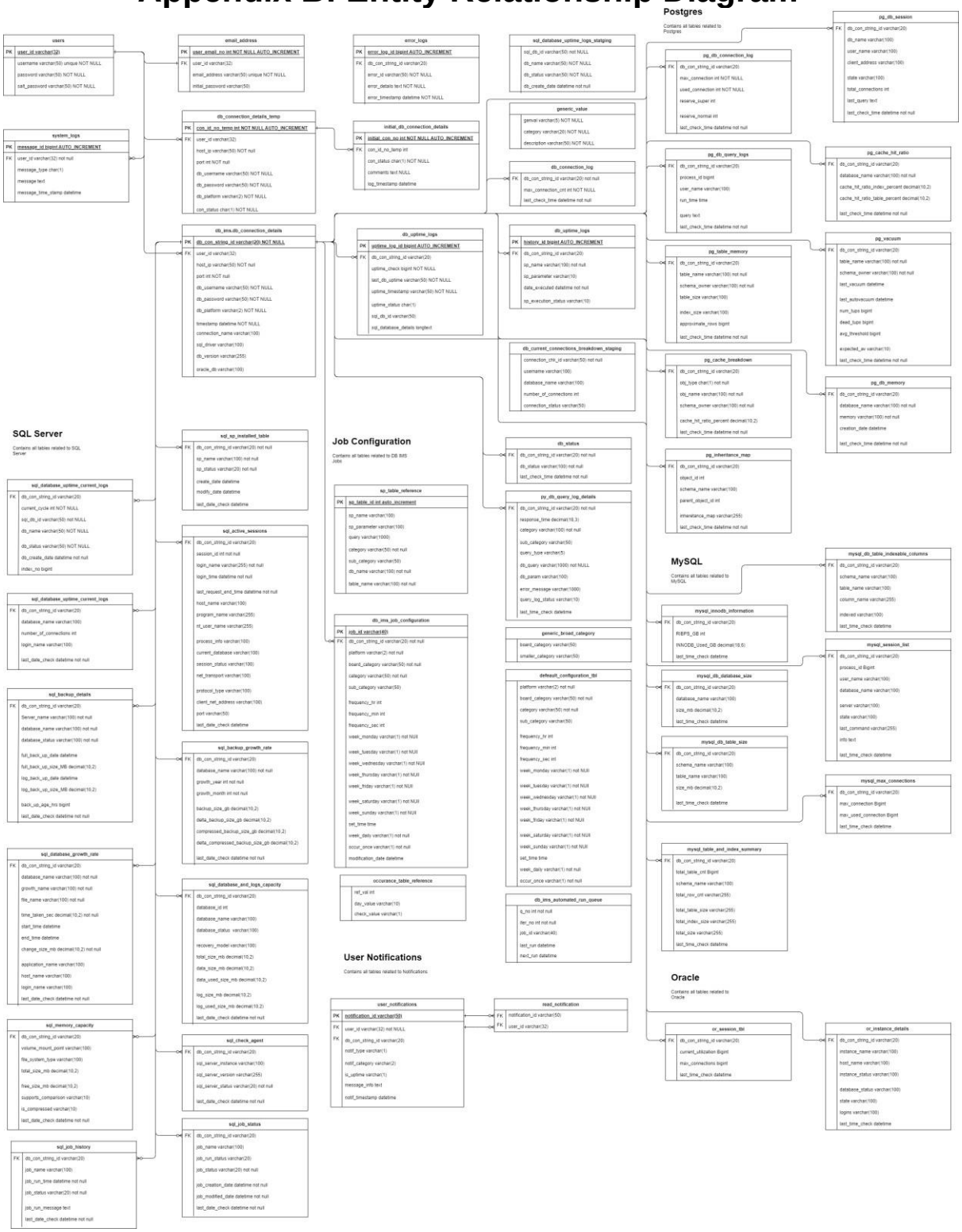


Figure 9: Entity Relationship Diagram

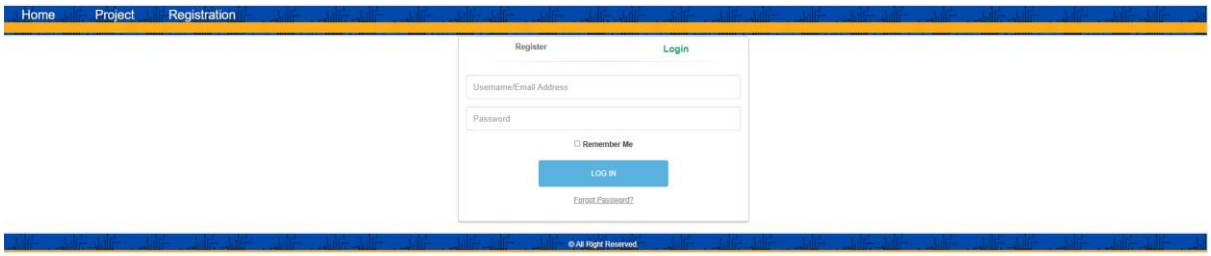
I. PNG File
II. HTML File
III. DrawIO

Appendix C: Webpages

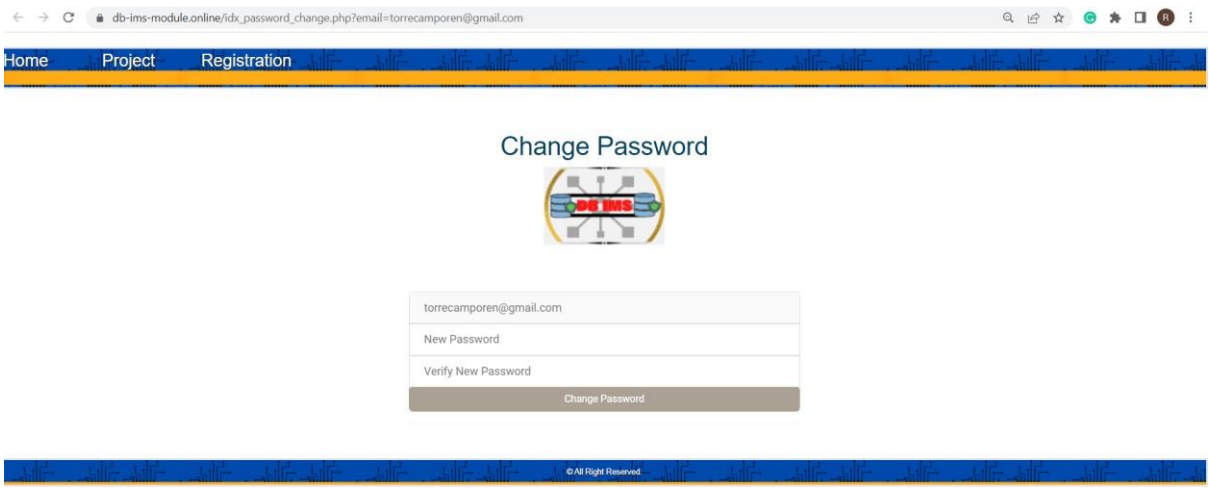
C.1: Home Page

C.2: Project Page

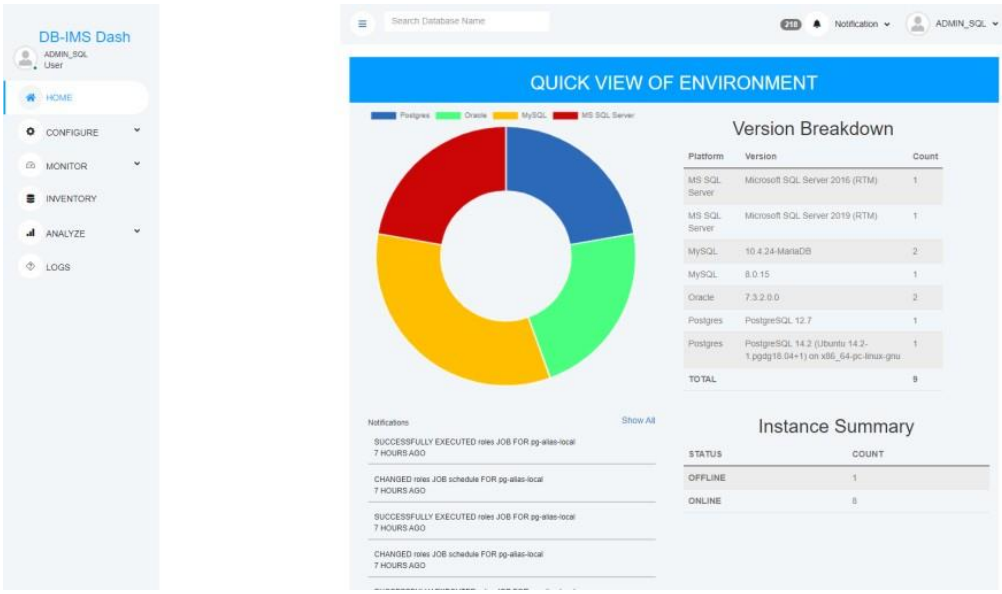
C.3: Sign Up / Login Page



C.4: Change Password

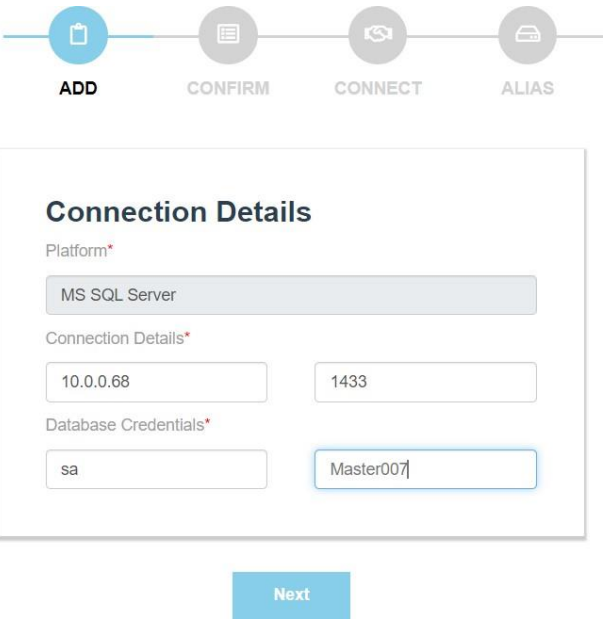


C.5: Dashboard

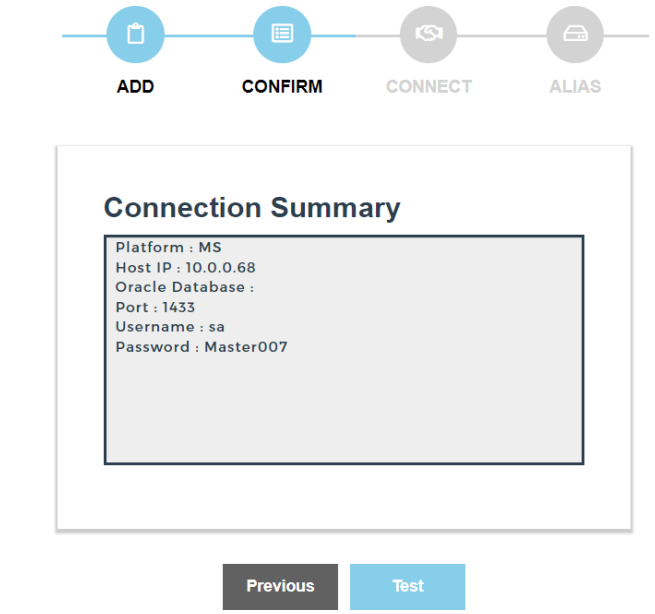


C.6: Configuration > ADD

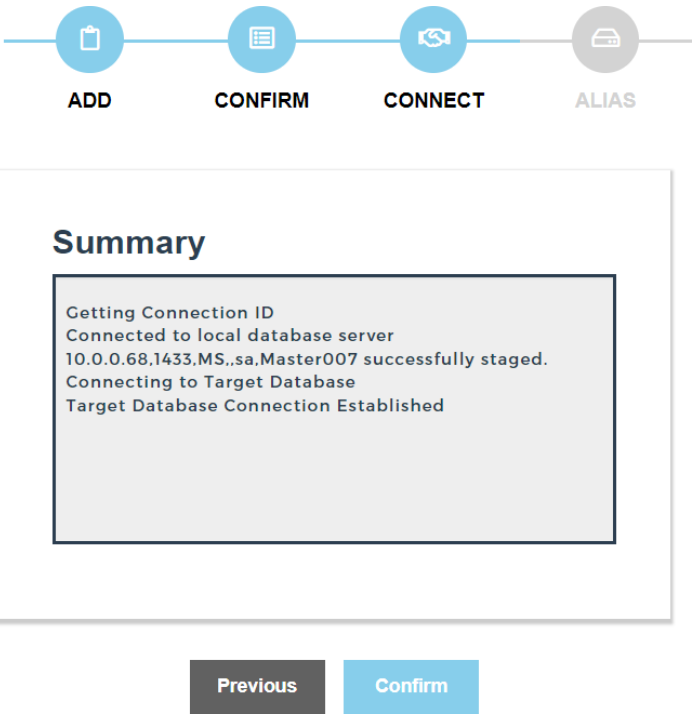
C.6.A: *Connection Details – ADD*



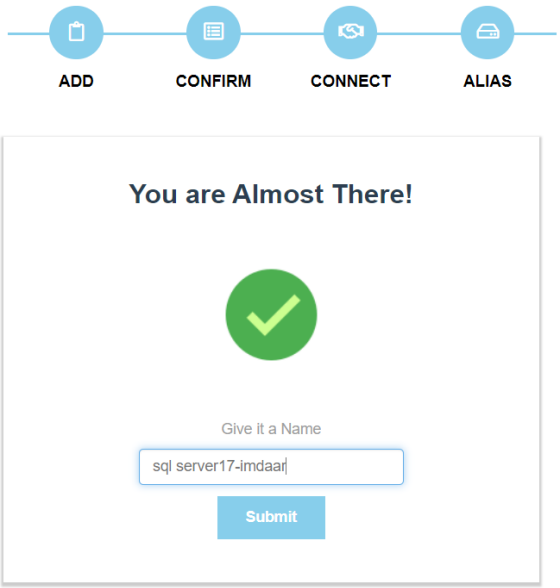
C.6.B: *Connection Details – Confirm*



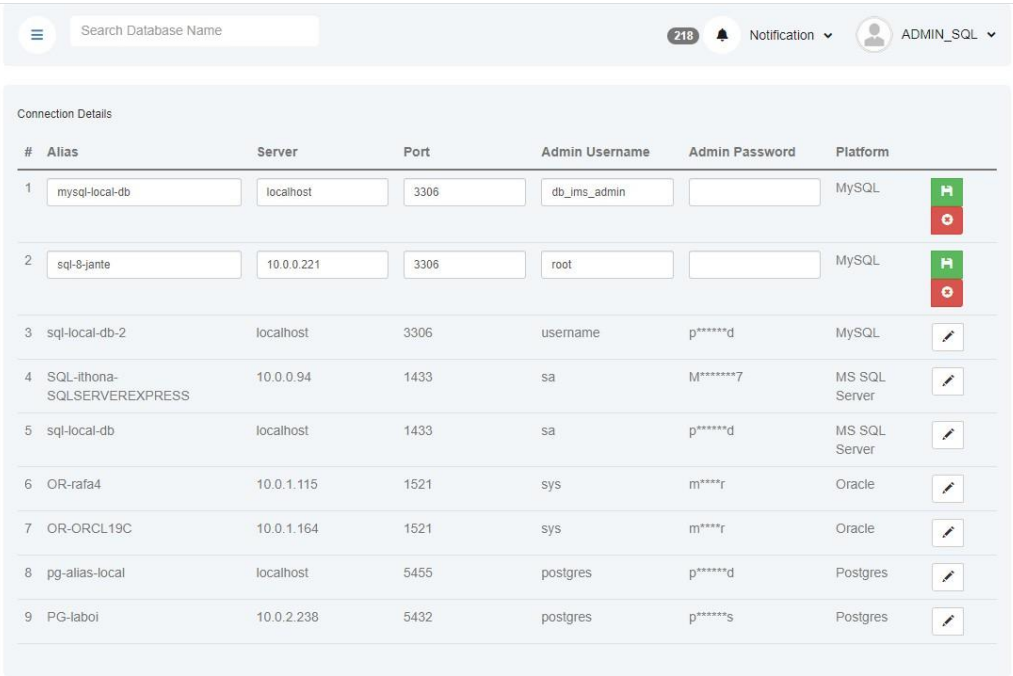
C.6.C: *Connection Details – Connect*



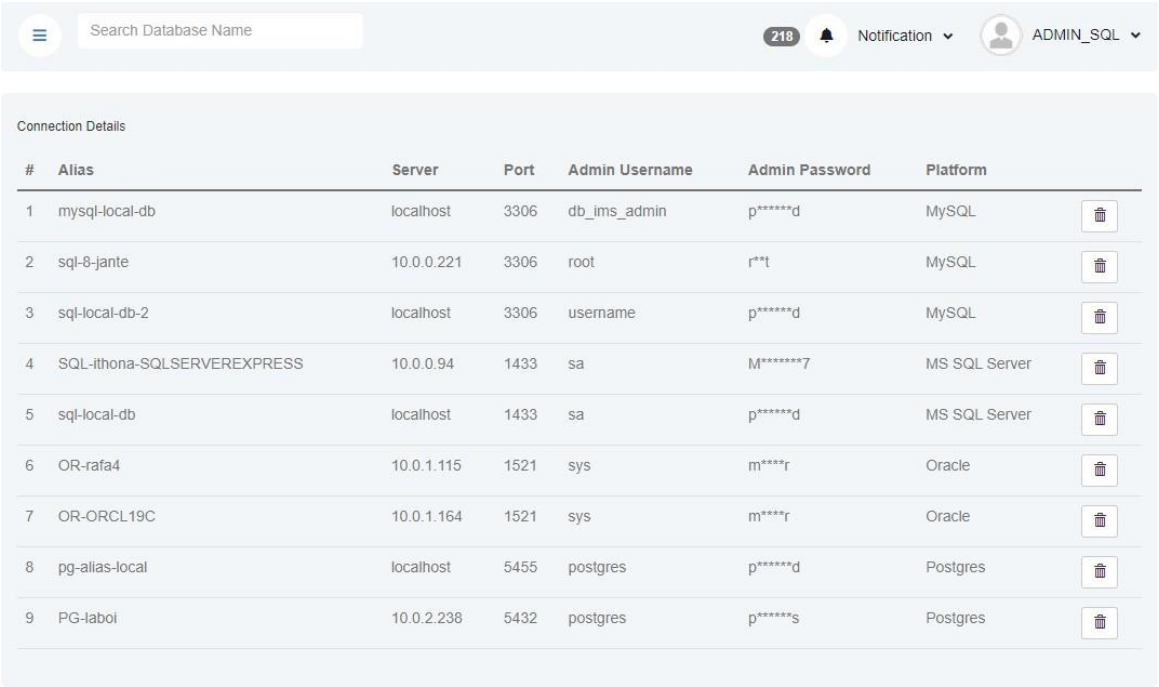
C.6.D: *Connection Details – Alias*



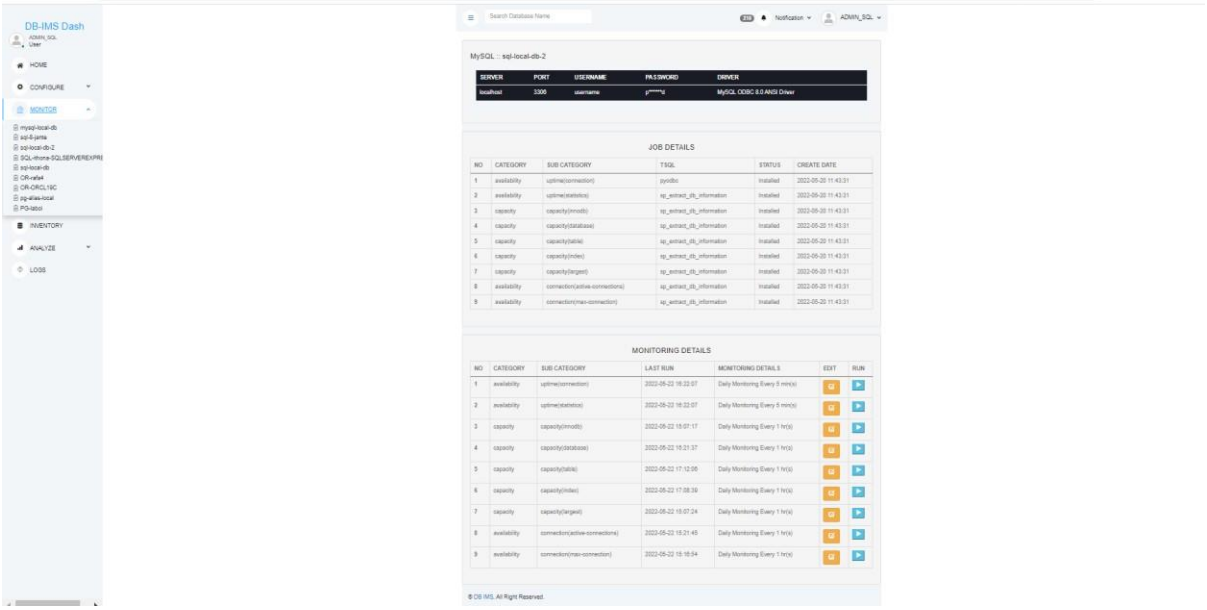
C.7: Configuration > Edit



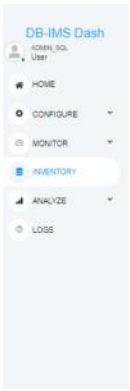
C.8: Configuration > Remove

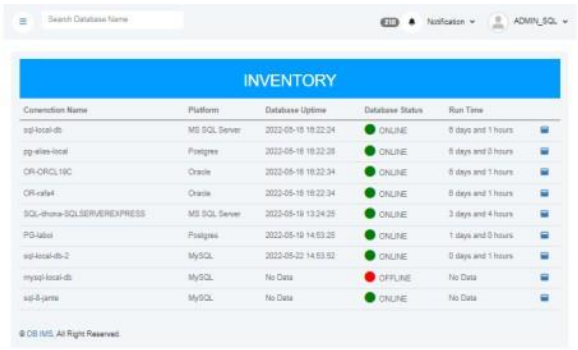


C.9: MONITOR > [CONNECTION]

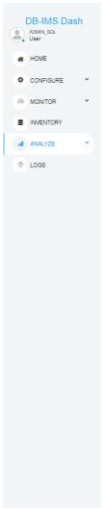


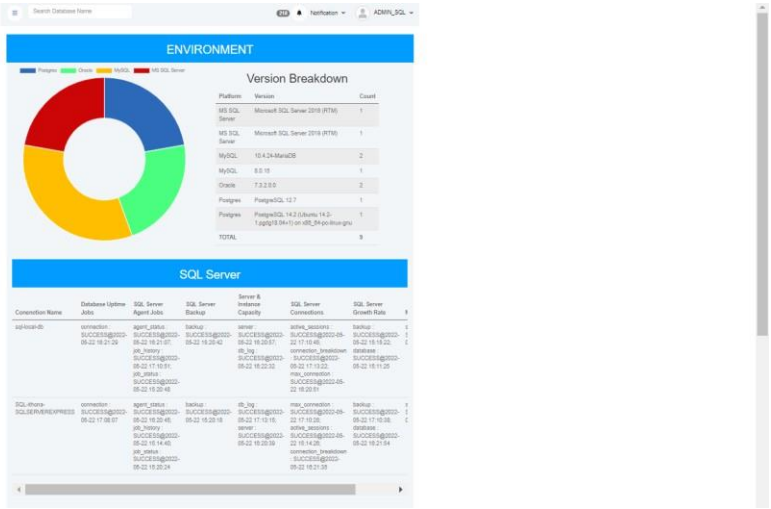
C.10: INVENTORY



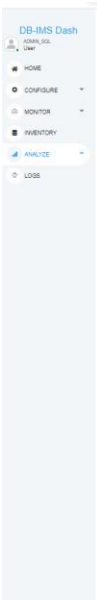


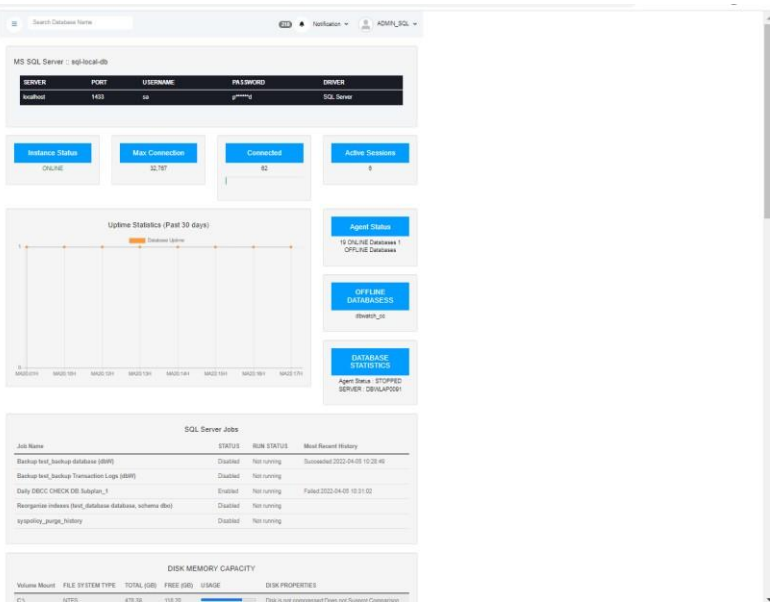
C.11: ANALYZE > Environment





C.12: Analyze > [CONNECTION]

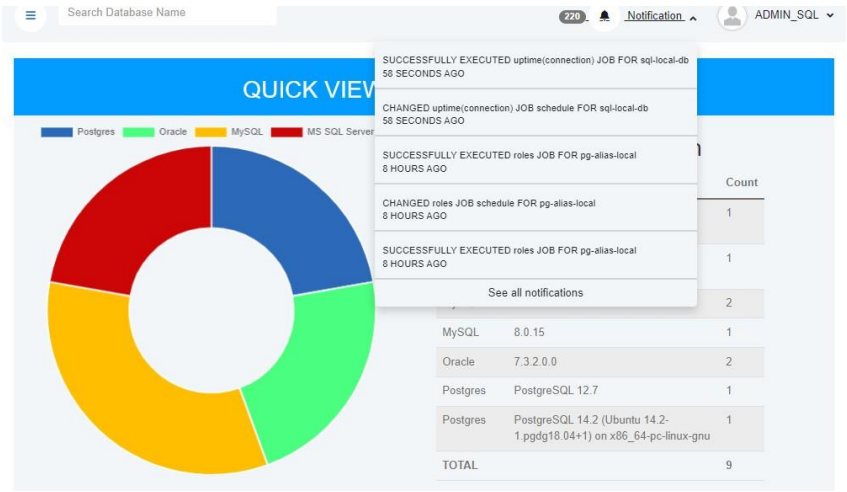




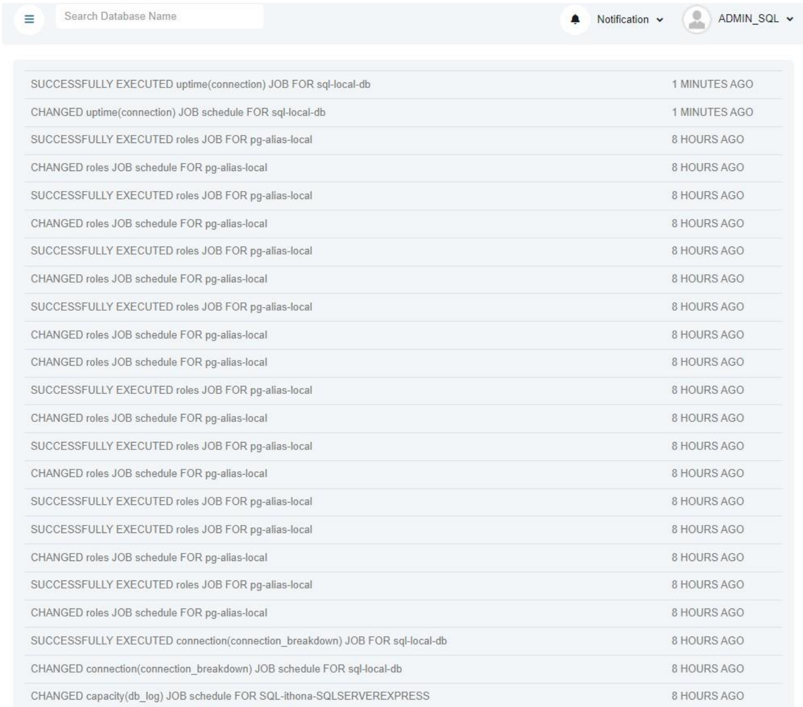
C.13: LOGS

C.14: Notification

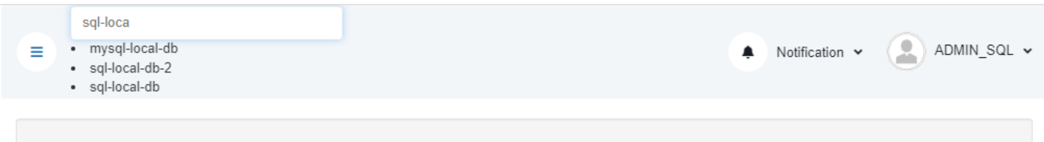
C.14.A: Notification Bar



C.14.B: See All Notifications

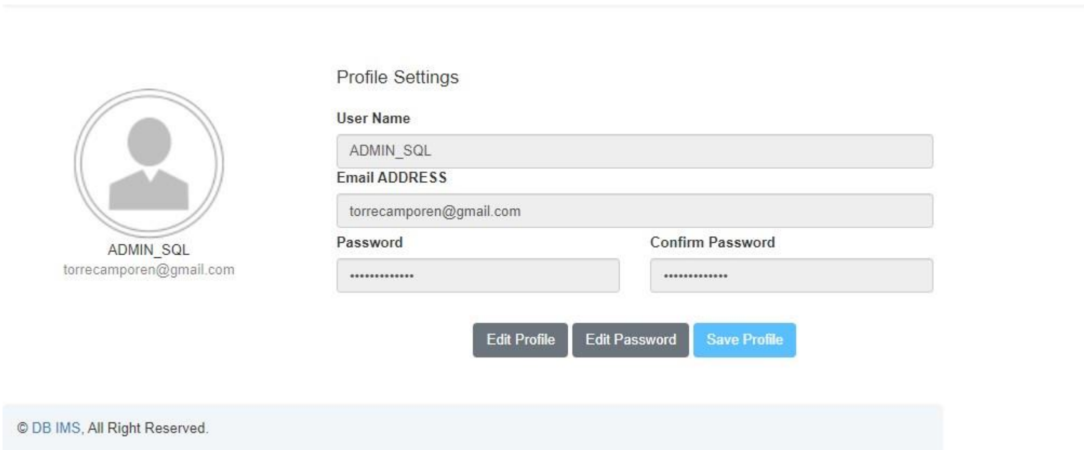


C.15: Search Bar

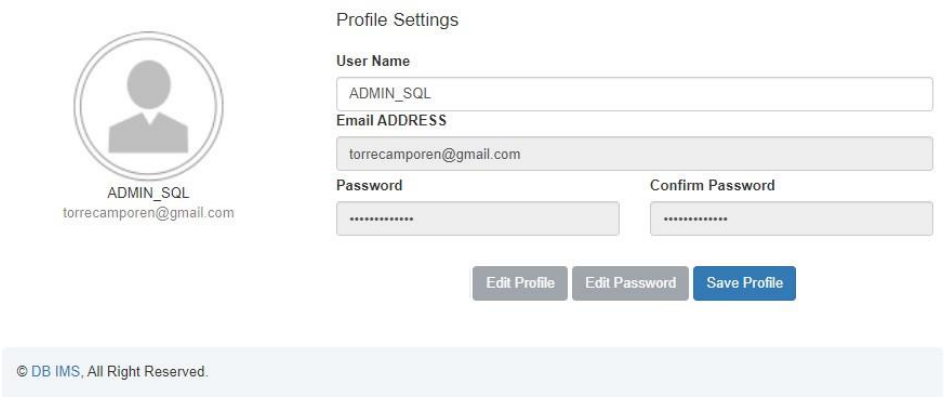


C.16: My Profile Page

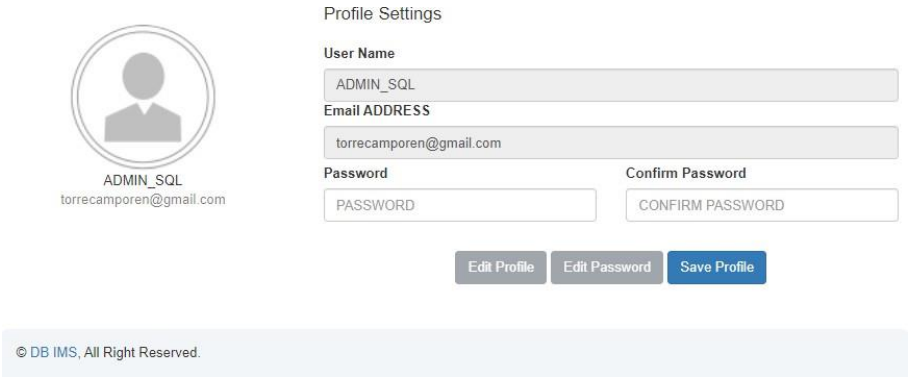
C.16.A Profile Page



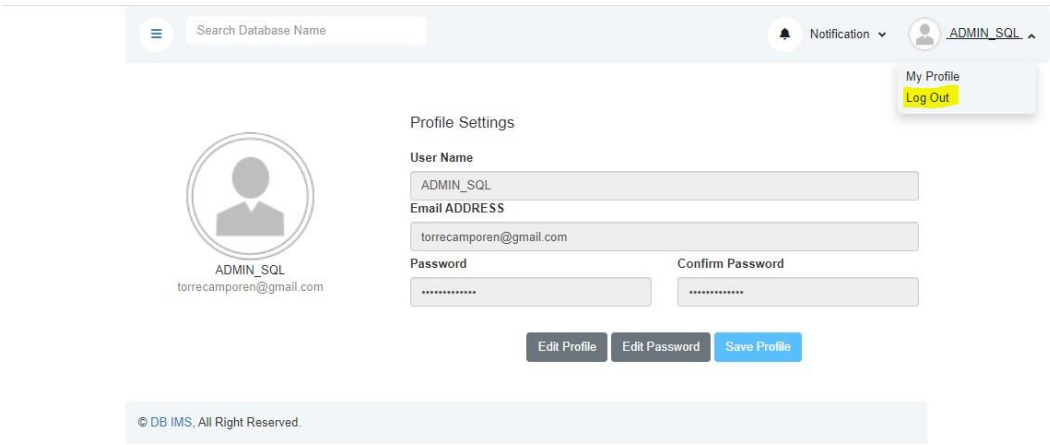
C.16.B Edit Profile



C.16.C Edit Password



C.17: Log Out



Appendix D: Actual Project Progress

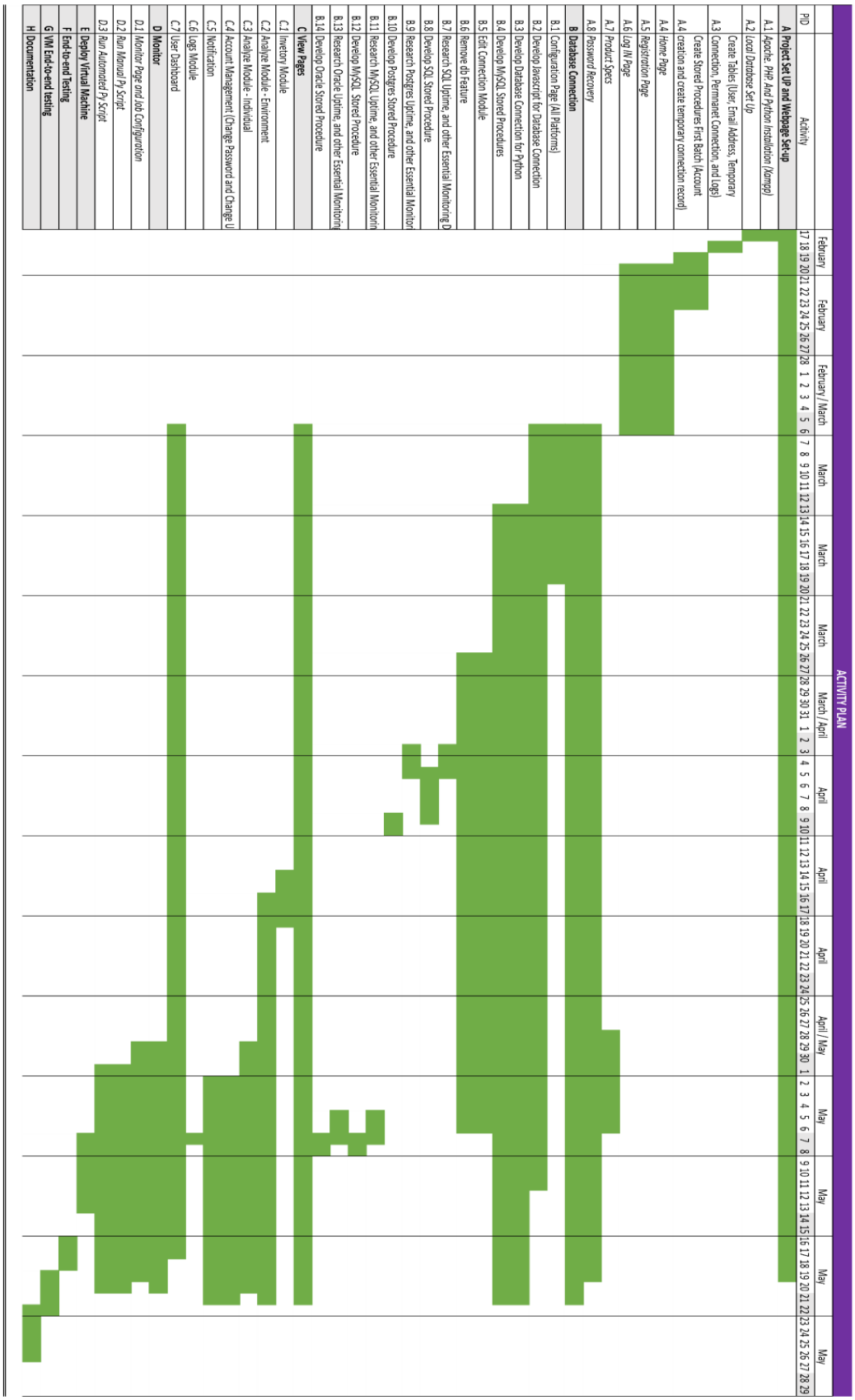


Figure 10: Gantt Chart for Project Implementation

Appendix E: User Case Model

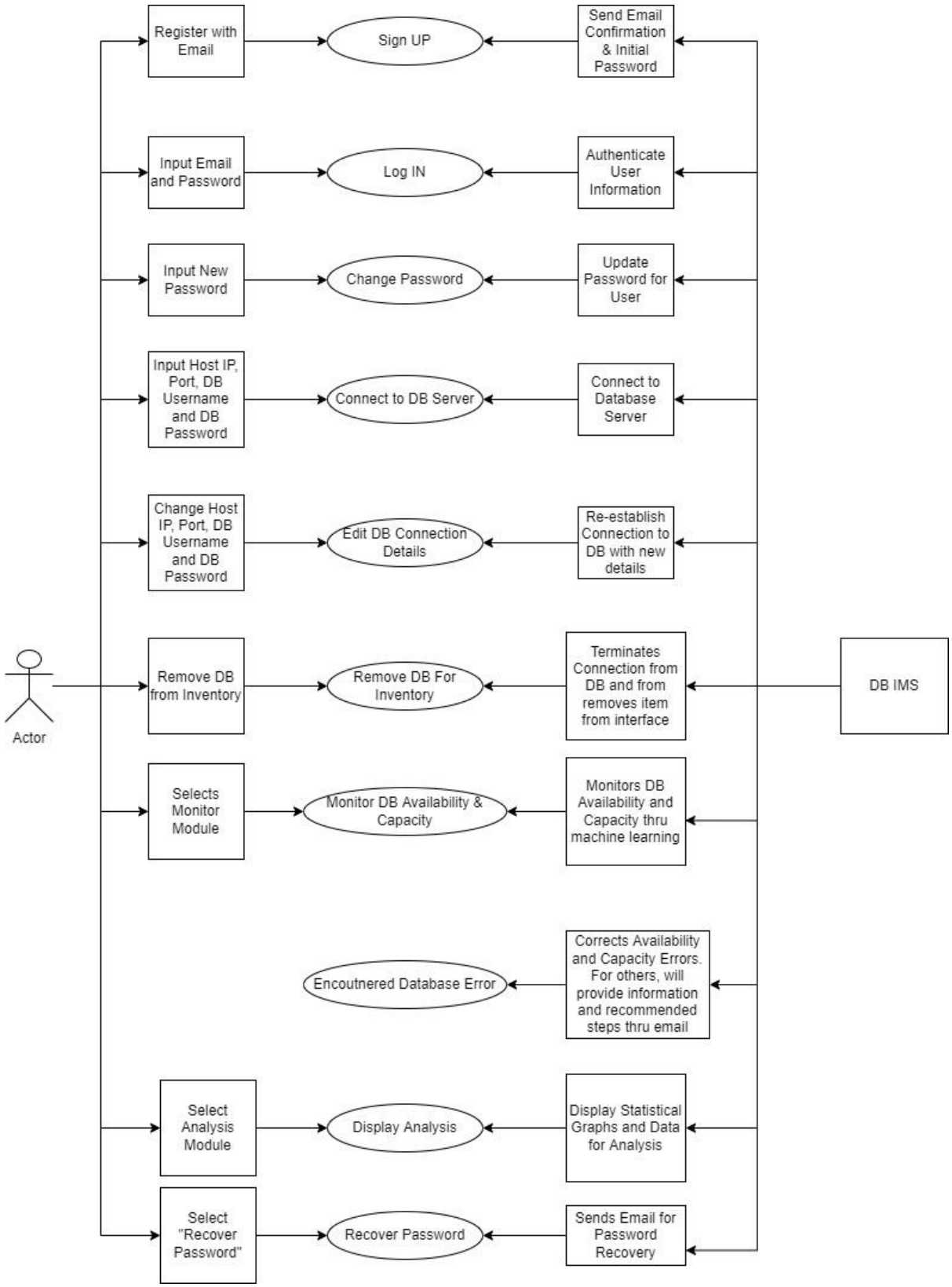
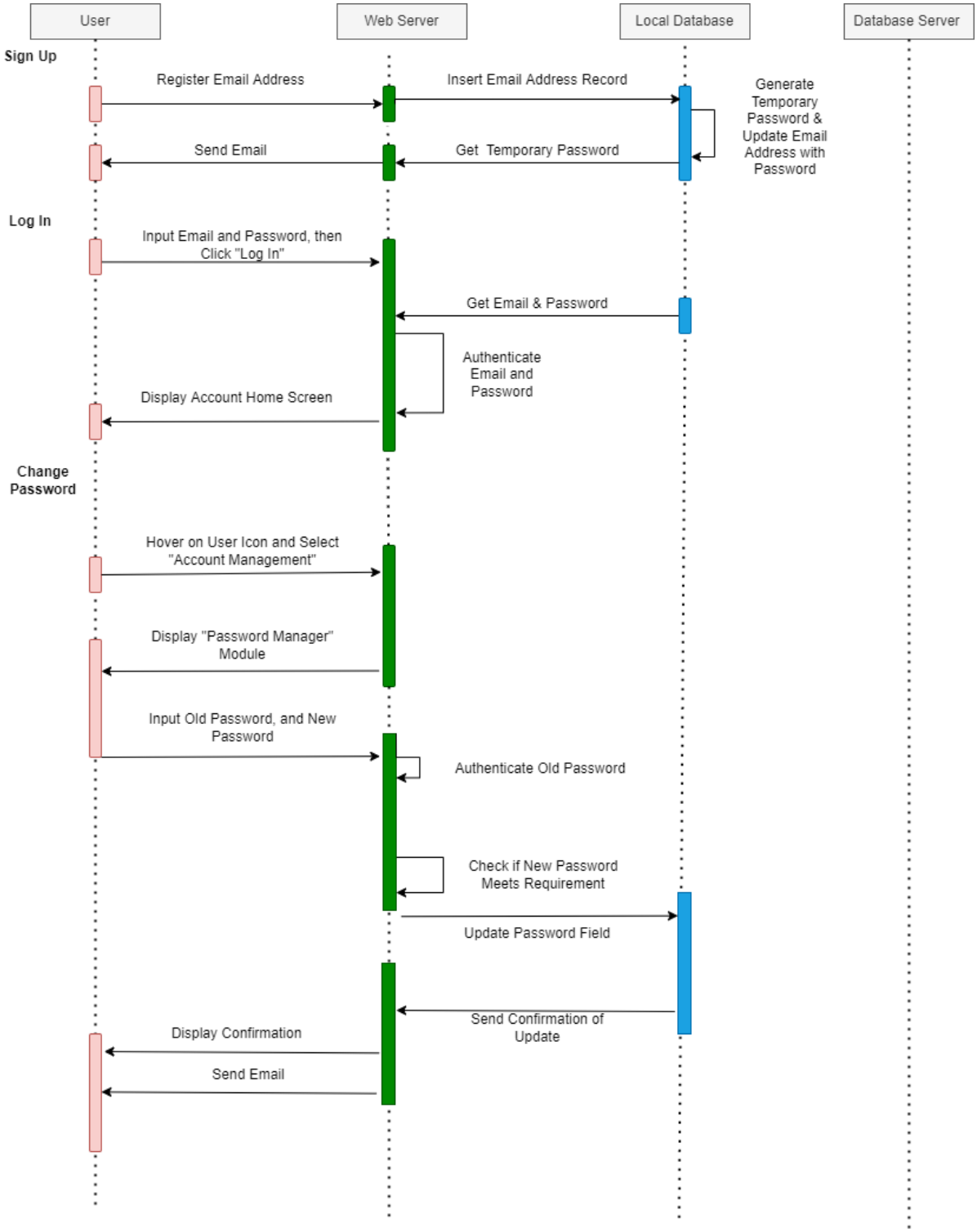


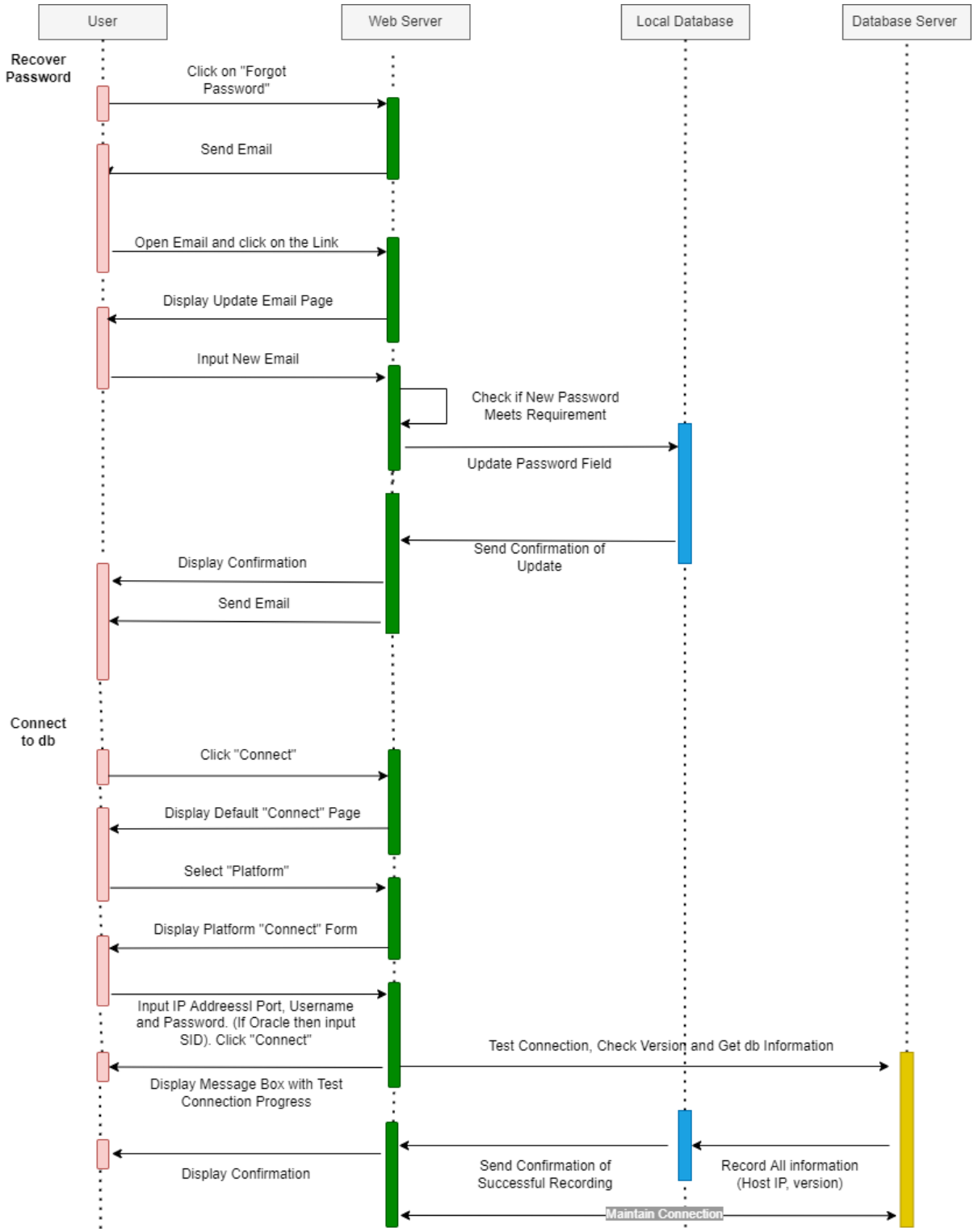
Figure 11: Use Case Model

Appendix F: Sequence Diagrams

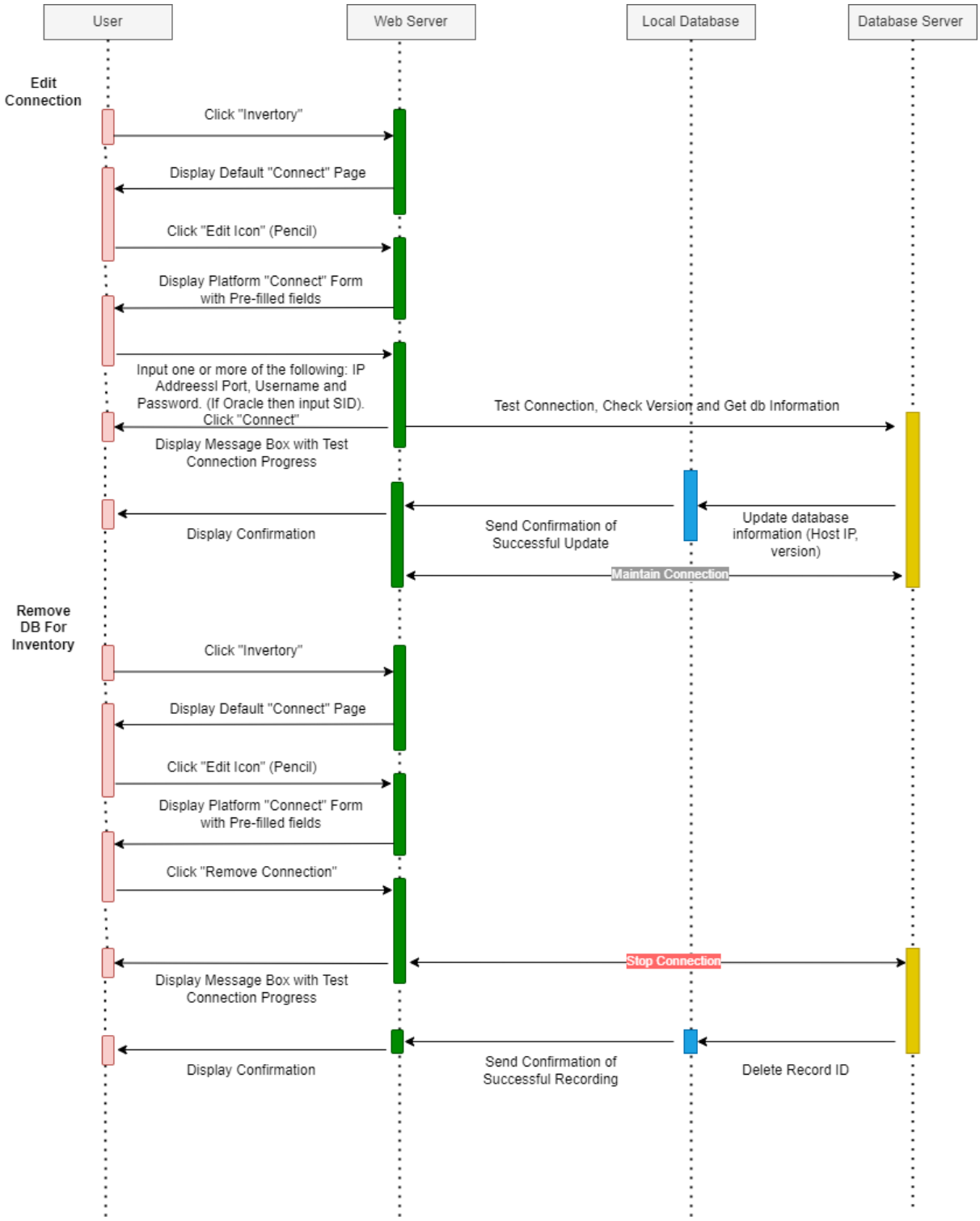
F.1: Account Creation, Log In and Change Password



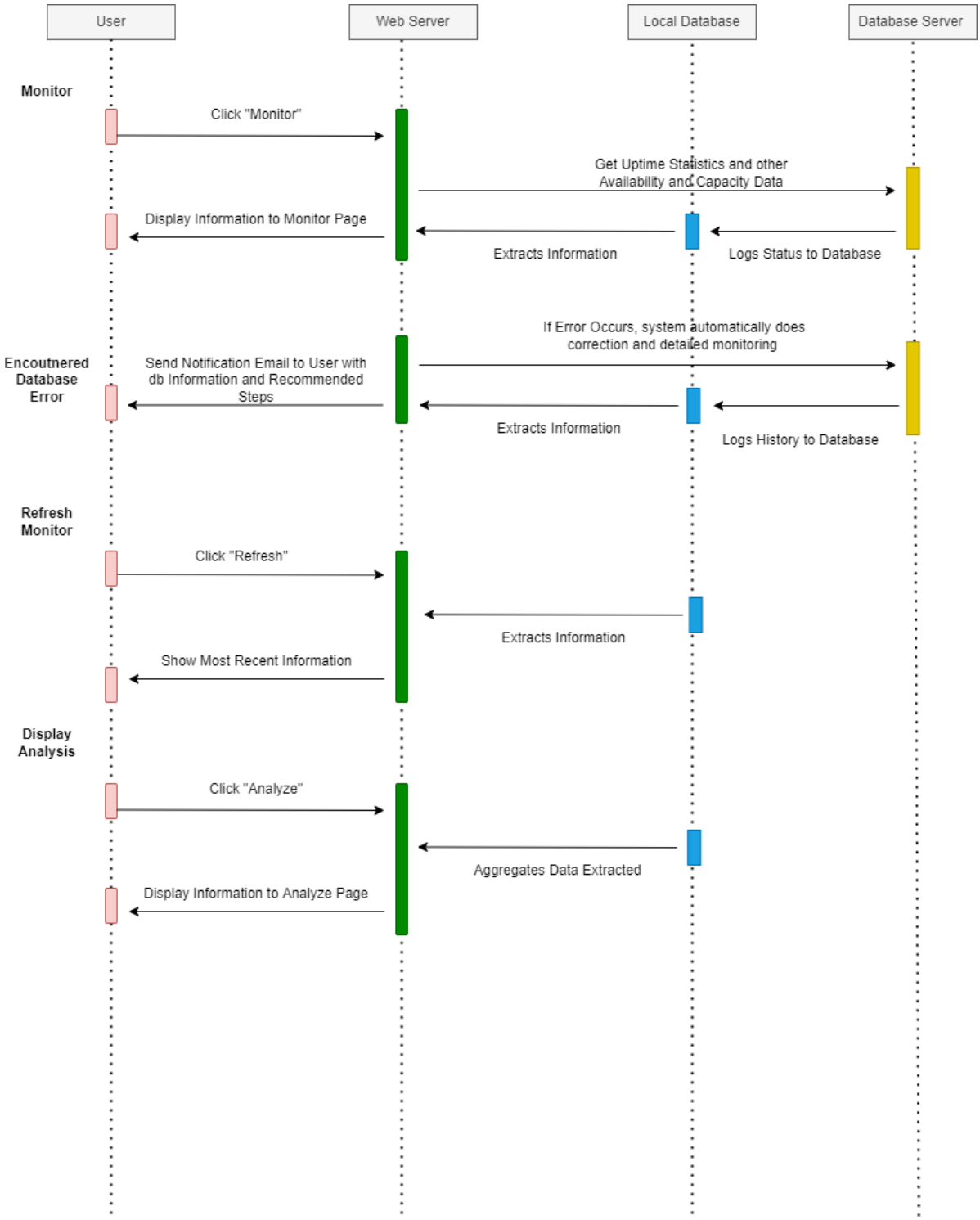
F.2: Recover Password and Connect to database



F.3: Edit Connection and Remove Connection



F.4: Monitor database, logging, and refresh database connection



Appendix G: List of Databases

G.1: List of Database for Integration Testing

	Type	Database type	DB Version	Hostname	Host IP	SID	Port	OS Type	DB user	DB password
1	Local	MySQL	[redacted]	[redacted]	127.0.0.1		3306	Windows 10	root	p@ssw0rd
2	Local	SQL Server	[redacted]	[redacted]	127.0.0.1		1433	Windows 10	postgres	p@ssw0rd
3	Local	PostgreSQL	[redacted]	[redacted]	127.0.0.1		5455	Windows 10	sa	p@ssw0rd
4	Approved	MariaDB	5.5.41	iqobal	10.0.0.86	NA	3306	Ubuntu 14.04 LTS 64bit	root	root
5	Approved	MySQL	8.0.15	jante	10.0.0.221	NA	3306	Ubuntu 16.04 LTS 64bit	root	root
6	Approved	Oracle	11.2.0.1.0 Enterprise Edition	rafa4	10.0.1.115	rafa4	1521	Oracle Enterprise Linux 5.10 64bit	sys	master
7	Approved	Oracle	19.3.0.0.0 Enterprise Edition	jodaka	10.0.1.164	ORCL19C	1521	Oracle Enterprise Linux 7.5 64bit	sys	master
8	Approved	PostgreSQL	14.2.0	laboi	10.0.2.238	NA	5432	Ubuntu 18.04.3 LTS64bit	postgres	postgres
9	Approved	SQL Server	2005 Express Edition	hijado	10.0.0.61	SQLEXPRESS	1433	Windows Server 2003Enterprise 32bit	sa	Master007
10	Approved	SQL Server	2016 Standard Edition	ithona	10.0.0.94	MSSQLSERVER	1433	Windows Server 2016Enterprise 64bit	sa	Master007

Table 3: List of Database and Access Information for Integration Testing

Note: Local databases are found in my local machine, while approved databases are found on our company's server. They can only be accessed with a VPN provided by our company.

G.2: List of Database for Staging Area Testing

	Database type	DB Version	Hostname	Host IP	SID	Port	DB user	DB password
1	MariaDB	5.5.41	hirsi	10.0.0.62	NA	3306	root	root
2	MySQL	8.0.13	janguine	10.0.0.187	NA	3306	root	root
3	MySQL	5.0.90	halmad	10.0.0.54	NA	3306	root	root
4	Oracle	9.2.0.1 EnterpriseEdition	gand	10.0.0.48	ORA92	1521	sys	master
5	Oracle	18.3.0.0.0 Enterprise Edition	jiaan	10.0.1.163	ORCL18C	1521	sys	master
6	Oracle	18.3.0.0.0 Enterprise Edition	jiaan	10.0.1.163	ORCL18C	1521	sys	master
7	Oracle	19.3.0.0.0 Enterprise Edition	jerne	10.0.1.70	ORCL19C	1521	sys as sysdba	master
8	PostgreSQL	8.1.23	herego	10.0.0.60	NA	5432	postgres	postgres
9	PostgreSQL	8.3.9	belsavis	10.0.0.39	NA	5432	postgres	postgres
10	PostgreSQL	8.4.10	budpock	10.0.0.40	NA	5432	postgres	postgres
11	PostgreSQL	9.0.6	bunduki	10.0.0.41	NA	5432	postgres	postgres
12	PostgreSQL	10.4.0	jangelle	10.0.0.183	NA	5432	postgres	postgres
13	PostgreSQL	12.5.0	kanzi	10.0.1.191	NA	5432	postgres	postgres
14	PostgreSQL	13.1.0	kardoa	10.0.1.192	NA	5432	postgres	postgres
15	SQL Server	2005 Express Edition	gyndine	10.0.0.51	SQLEXPRESS	1433	sa	Master007
16	SQL Server	2008 Express Edition	cona	10.0.0.45	SQL2008EX	1433	sa	Master007
17	SQL Server	2012 EnterpriseEdition	imdaar	10.0.0.68	MSSQLSERVER	1433	sa	Master007

18	SQL Server	2014 Standard Edition	indupar	10.0.0.76	MSSQLSERVER	1433	sa	Master007
19	SQL Server	2014 EnterpriseEdition	indikir	10.0.0.73	MSSQLSERVER	1433	sa	Master007
20	SQL Server	2014 Express Edition	intran	10.0.0.79	SQLEXPRESS	1433	sa	Master007
21	SQL Server	2016 EnterpriseEdition	ivatch	10.0.0.95	SQLA	1433	sa	Master007
22	SQL Server	2017 Standard Edition	ixtlar	10.0.0.99	MSSQLSERVER	1433	sa	Master007
23	SQL Server	2017 EnterpriseEdition	jaminere	10.0.0.121	MSSQLSERVER	1433	sa	Master007
24	SQL Server	2019 RC1 Developer Edition	jaroona	10.0.1.53	MSSQLSERVER	1433	sa	Master007

Table 4: List of Databases and Access Information for Staging Area Testing

Note: All listed databases are found in our company's network. They can only be accessed through a VPN provided by our company. Here are the complete details of the list of databases.

Appendix H: Test Scenarios

H.1: All test cases

Case ID	Suite Title	Title	Description	Type	Automation status
1	Account Details	Create an Account	As a user, I should be able to create an account through Email	Functional	Not automated
2	Account Details	Duplicate Email Address	Users cannot have similar emails	Functional	Not automated
3	Account Details	Recover Password	As a user, I should be able to recover my Password.	Functional	Not automated
4	Account Details	Change Username	As a user, I should be able to change my Username	Functional	Not automated
5	Account Details	Change Password in "My Profile"	As a user, I should be able to change my Password on the Profile Page	Functional	Not automated
6	Connect Db Connections	Add Database Connection	As a user I should be able to connect to a database	Functional	Not automated
7	Edit / Remove Connection	[Save] Edit Connection Details	As a user I should be able to edit the current connection details	Functional	Not automated
8	Edit / Remove Connection	[Confirm] Remove Connection	As a user, I should be able to remove any connection	Functional	Not automated
9	Edit / Remove Connection	[Cancel] Remove Connection	As a user, I should be able to remove any connection	Functional	Not automated
10	Edit / Remove Connection	[Cancel] Edit Connection Details	As a user I should be able to edit the current connection details	Functional	Not automated
11	View all Connections	View Inventory	As a user I should see my database inventory	Functional	Not automated
12	View all Connections	Search Conenction View Inventory	As a user, I should be able to search my database connection with name	Functional	Not automated
13	Account Details	[Individual] See Notifications	As a user I should be able to be notified by the system	Functional	Not automated
14	Account Details	[All] See Notifications	As a user I should be able to be notified by the system	Functional	Not automated
15	View all Connections	See Logs	As a user, I should be able to see the log file	Functional	Not automated
17	View all Connections	View Environment	As a user I should be able to see if jobs were executed and their status	Functional	Not automated

H.2: All test cases and scenarios

Test Category	Case Title	Action	Input Data	Expected Result
Account Details	Create an Account	Go to Register Page		Directed to Registered Link
Account Details	Create an Account	Type in Email	renpogiman@gmail.com	Email is Successfully Sent Confirmation
Account Details	Create an Account	Go to Email Account		Email should show Username and Password similar to that in database.
Account Details	Create an Account	Click Log in to dashboard		Transition to log in page
Account Details	Create an Account	Log in Username and Password	Login = (Username) / Password = (Password)	redirect to dashboard
Account Details	Duplicate Email Address	Open Registration Page		Open Registration Page
Account Details	Duplicate Email Address	Input Existing Email	renpogiman@gmail.com	Flag as duplicate email
Account Details	Recover Password	Click Forget Password in Log in Page		Redirected to Password Recovery Page

Account Details	Recover Password	Input Email Address	renpogiman@gmail .com	Click Submit and Email should be sent
Account Details	Recover Password	Open Email Account		Recover Password Email
Account Details	Recover Password	Open Email Link		Redirected to Password Recovery Page
Account Details	Recover Password	Input New Password	password1@ in both input bar	Change Password is successful
Account Details	Recover Password	Go back to the registration Log-in Page		Redirected to Log in Page
Account Details	Recover Password	Input email and new Password	Given Username or email address / password1@	Redirect to Home Page
Account Details	Change Username	Go to My Profile		Redirect to My Profile Page
Account Details	Change Username	Click Change Username button		Password button disabled, Username field enabled and Save Button Enabled
Account Details	Change Username	Change Username	New_Username	
Account Details	Change Username	Click Save Button		Reload Page and New_username should appear
Account Details	Change Password in "My Profile"	Click on My Profile		redirected to My Profile Page
Account Details	Change Password in "My Profile"	Click "Change Password"		Enable Password inputs, and Save button
Account Details	Change Password in "My Profile"	Input New Password	p@ssw0rd4w	
Account Details	Change Password in "My Profile"	Click Save		Reload Page and changes should be seen
Account Details	[Individual] See Notifications	Click on notification		Dropdown list of new notifications
Account Details	[Individual] See Notifications	Click on notification		Direct to relevant notification link
Account Details	[All] See Notifications	Click on notification		Dropdown list of new notifications
Account Details	[All] See Notifications	Click on "See All Notifications"		redirected to notification page
Connect Db Connections	Add Database Connection	Click on Add Database		Redirect to Add Connection Form
Connect Db Connections	Add Database Connection	Click on Platform		If Oracle (Oracle db field should be seen); else it should be the same form
Connect Db Connections	Add Database Connection	Add Connection Details	host ip / port/ username / password /(Oracle db if oracle is chosen)	
Connect Db Connections	Add Database Connection	Click Next		See Summary in Text Box

Connect Db Connections	Add Database Connection	Click Test		Test connection details; if correct then CONFIRM should be enable, else only previous should be enabled
Connect Db Connections	Add Database Connection	Add Alias	input database alias	
Connect Db Connections	Add Database Connection	Click Save		Save Alias for database name
View all Connections	View Inventory	Click Inventory Link in Dashboard		Directed to Inventory Page
View all Connections	View Inventory	See Inventory Page		Inventory should list all instances connected
View all Connections	Search Conenction View Inventory	Click on search bar	Input Connection Name	Dropdown list of like instances
View all Connections	Search Conenction View Inventory	Click on the instance		Directed to Monitor Page
View all Connections	See Logs	Click on LOGS		redirected to log file
View all Connections	See Logs	View Log File		Log file is shown in iframe
View all Connections	View Environment	Open Analyze > Environment		Analyze Environment Page
View all Connections	View Environment	See Details of Analyze page		Display Analyze Page must be correct
Edit / Remove Connection	[Save] Edit Connection Details	Click Edit		Redirected to Edit Web Page
Edit / Remove Connection	[Save] Edit Connection Details	Click on one of the records		Open up other connection details
Edit / Remove Connection	[Save] Edit Connection Details	Edit Connection Details	New Connection Details and Connection Name	
Edit / Remove Connection	[Save] Edit Connection Details	Click Save		Refresh Page
Edit / Remove Connection	[Confirm] Remove Connection	Click on Remove		Directed to Remove Page
Edit / Remove Connection	[Confirm] Remove Connection	Click on the row with delete button		Pop up saying to proceed
Edit / Remove Connection	[Confirm] Remove Connection	Click Confirm		Refresh Page
Edit / Remove Connection	[Cancel] Remove Connection	Click on Remove		Directed to Remove Page
Edit / Remove Connection	[Cancel] Remove Connection	Click on the row with delete button		Pop saying to proceed
Edit / Remove Connection	[Cancel] Remove Connection	Click Cancel		Should not remove connection
Edit / Remove Connection	[Cancel] Edit Connection Details	Click Edit		Redirected to Edit Web Page
Edit / Remove Connection	[Cancel] Edit Connection Details	Click on one of the records		Open up other connection details

Edit / Remove Connection	[Cancel] Edit Connection Details	Edit Connection Details	New Connection Details and Connection Name	
Edit / Remove Connection	[Cancel] Edit Connection Details	Click Cancel		No Action Done

H.3: Severity Criteria

When a bug is detected, testing will continue until all test scenarios are done unless the bug affects the testing (see table 2 for expected bugs that might affect testing). For each bug, the following severity will be observed:

Severity	Action
High	Stop Testing and address bugs; Highest Priority to be addressed immediately
Medium	Continue Testing until all test scenarios are covered; not too much game-breaking and can be deployed in the next release
Low	Continue testing until all test scenarios are covered; least priority and primarily for recommendation

Table 5: Bug Triage - Severity Criteria

Below are High Priority Bugs that need to be addressed immediately:

Unable to proceed due to invalid Username or Password
Non-Clickable links
Zero connected instances for a particular platform (except for Oracle)
Wrong web pages displayed
Unresponsive python script, javascript, or other components

Appendix I: Test Scenarios

I.1: Create an Account

Description

Password in db: 1c752!BDE0
Password in Email: **1c752 BDE0.**

Attachments

Test run results

Test case	Author	Found in run
DI-1 Create an Account	Rey Lawrence Torrecampo	R-2 - Integrated Environment Testing

Comments

RE **Rey Lawrence Torrecampo** Software Engineer
removed '!' and set a more appropriate password
return concat(LEFT(MD5(RAND()),5),ucase(LEFT(MD5(RAND()),5)));

5 days ago

Status
Resolved

Severity
Minor

Reporter
RE Rey Lawrence Torrecampo

Assignee
RE Rey Lawrence Torrecampo

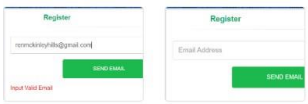
Created
17 May 2022 at 20:18:38

I.2: Duplicate Email Address

Description

Wrong Error. Should be Email is already a duplicate.

Attachments



Test run results

Test case	Author	Found in run
DI-2 Duplicate Email Address	Rey Lawrence Torrecampo	R-2 - Integrated Environment Testing

Comments

RE **Rey Lawrence Torrecampo** Software Engineer
Resolved.

5 days ago

Status
Resolved

Reporter
RE Rey Lawrence Torrecampo

Assignee
RE Rey Lawrence Torrecampo

Created
17 May 2022 at 20:22:39

I.3: Recover Password

Description

Minor Bugs:

- Redirect to Login Page After submission
- Subject Line not found in recovery
- No Email Sent or Confirmation.

Showstoppers:

- Password Recovery should work; after submitting new password, it does not save in db level

Attachments

Test run results

Test case	Author	Found in run
DI-3 Recover Password	Rey Lawrence Torrecampo	R-2 - Integrated Environment Testing

Status
Resolved

Reporter
RE Rey Lawrence Torrecampo

Assignee
Unassigned

Created
17 May 2022 at 20:28:39

I.4: Add Database Connection

Description

Check Python Connection of error

Attachments

xlsx

Test run results

Test case	Author	Found in run
DI-6 Add Database Connection	Rey Lawrence Torrecampo	R-2 - Integrated Environment Testing

Comments

RE **Rey Lawrence Torrecampo** Software Engineer
Done. Able to conenct to databases
5 days ago

RE **Rey Lawrence Torrecampo** Software Engineer
Resolved.

Status
Resolved

Severity
Critical

Reporter
RE Rey Lawrence Torrecampo

Assignee
RE Rey Lawrence Torrecampo

Created
19 May 2022 at 19:55:32

I.5: View Environment

Description

Only Postgres and SQL Server. Not accurate depiction.

- Remove other SQL Server and Postgres Views
- Add Oracel and MySQL Environment results

Attachments

Test run results

Test case	Author	Found in run
DI-17 View Environment	Rey Lawrence Torrecampo	R-3 - Integration Testing 05/20/2022

Status
Resolved

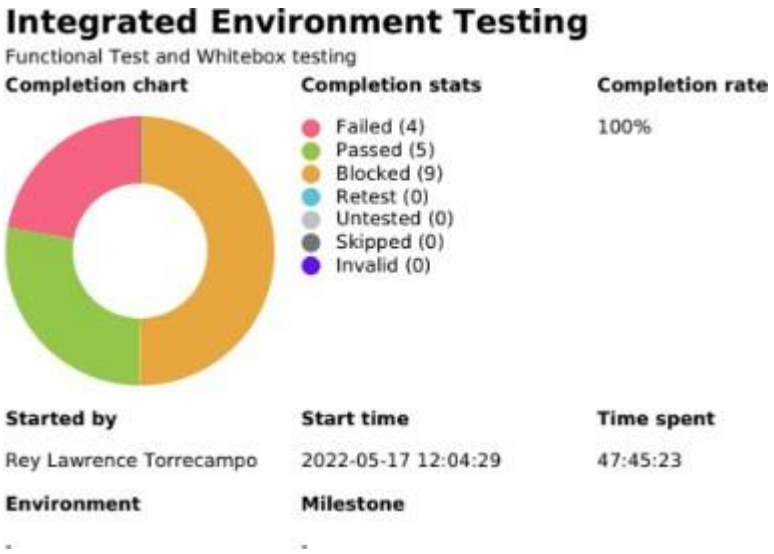
Reporter
RE Rey Lawrence Torrecampo

Assignee
Unassigned

Created
20 May 2022 at 01:09:44

Appendix J: Testing Results

J.1: Integration Testing – First Attempt



Link for complete documentation: [First Attempt (Full Documentation)]

J.2: Integration Testing – First Attempt (Database ConnectionResults)

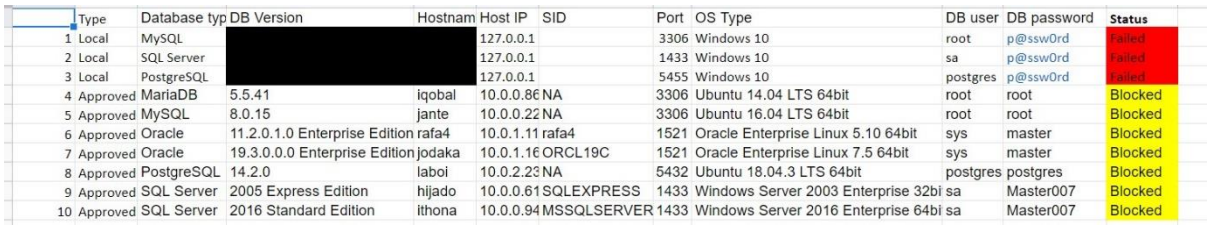
	Type	Database typ	DB Version	Hostnam	Host IP	SID	Port	OS Type	DB user	DB password	Status
1	Local	MySQL	[redacted]		127.0.0.1		3306	Windows 10	root	p@ssw0rd	Failed
2	Local	SQL Server			127.0.0.1		1433	Windows 10	sa	p@ssw0rd	Failed
3	Local	PostgreSQL			127.0.0.1		5455	Windows 10	postgres	p@ssw0rd	Failed
4	Approved	MariaDB	5.5.41	iqobal	10.0.0.86	NA	3306	Ubuntu 14.04 LTS 64bit	root	root	Blocked
5	Approved	MySQL	8.0.15	jante	10.0.0.22	NA	3306	Ubuntu 16.04 LTS 64bit	root	root	Blocked
6	Approved	Oracle	11.2.0.1.0 Enterprise Edition	rafa4	10.0.1.11	rafa4	1521	Oracle Enterprise Linux 5.10 64bit	sys	master	Blocked
7	Approved	Oracle	19.3.0.0.0 Enterprise Edition	jodaka	10.0.1.16	ORCL19C	1521	Oracle Enterprise Linux 7.5 64bit	sys	master	Blocked
8	Approved	PostgreSQL	14.2.0	laboi	10.0.2.23	NA	5432	Ubuntu 18.04.3 LTS 64bit	postgres	postgres	Blocked
9	Approved	SQL Server	2005 Express Edition	hijado	10.0.0.61	SQLEXPRESS	1433	Windows Server 2003 Enterprise 32bi	sa	Master007	Blocked
10	Approved	SQL Server	2016 Standard Edition	ithona	10.0.0.94	MSSQLSERVER	1433	Windows Server 2016 Enterprise 64bi	sa	Master007	Blocked

Link for complete documentation: [First Attempt – db connection]

J.3: Integration Testing – Second Attempt

Link for complete documentation: [Second Attempt (Full Documentation)]

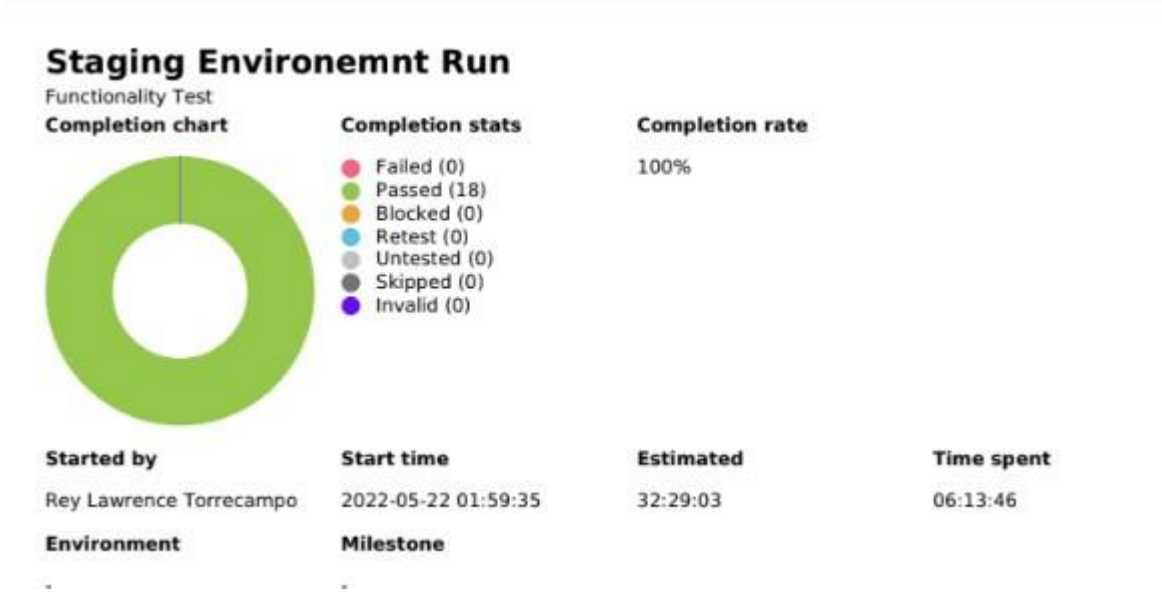
J.4: Integration Testing – Second Attempt (Database ConnectionResults)



	Type	Database typ	DB Version	Hostnam	Host IP	SID	Port	OS Type	DB user	DB password	Status
1	Local	MySQL	[redacted]		127.0.0.1		3306	Windows 10	root	p@ssw0rd	Failed
2	Local	SQL Server			127.0.0.1		1433	Windows 10	sa	p@ssw0rd	Failed
3	Local	PostgreSQL			127.0.0.1		5455	Windows 10	postgres	p@ssw0rd	Failed
4	Approved	MariaDB	5.5.41	iqobal	10.0.0.86	NA	3306	Ubuntu 14.04 LTS 64bit	root	root	Blocked
5	Approved	MySQL	8.0.15	jante	10.0.0.22	NA	3306	Ubuntu 16.04 LTS 64bit	root	root	Blocked
6	Approved	Oracle	11.2.0.1.0 Enterprise Edition	rafa4	10.0.1.11	rafa4	1521	Oracle Enterprise Linux 5.10 64bit	sys	master	Blocked
7	Approved	Oracle	19.3.0.0.0 Enterprise Edition	jodaka	10.0.1.16	ORCL19C	1521	Oracle Enterprise Linux 7.5 64bit	sys	master	Blocked
8	Approved	PostgreSQL	14.2.0	laboi	10.0.2.23	NA	5432	Ubuntu 18.04.3 LTS 64bit	postgres	postgres	Blocked
9	Approved	SQL Server	2005 Express Edition	hijado	10.0.0.61	SQLEXPRESS	1433	Windows Server 2003 Enterprise 32bi	sa	Master007	Blocked
10	Approved	SQL Server	2016 Standard Edition	ithona	10.0.0.94	MSSQLSERVER	1433	Windows Server 2016 Enterprise 64bi	sa	Master007	Blocked

Link for complete documentation: [Second Attempt – db connection]

J.5: Staging Environment Testing



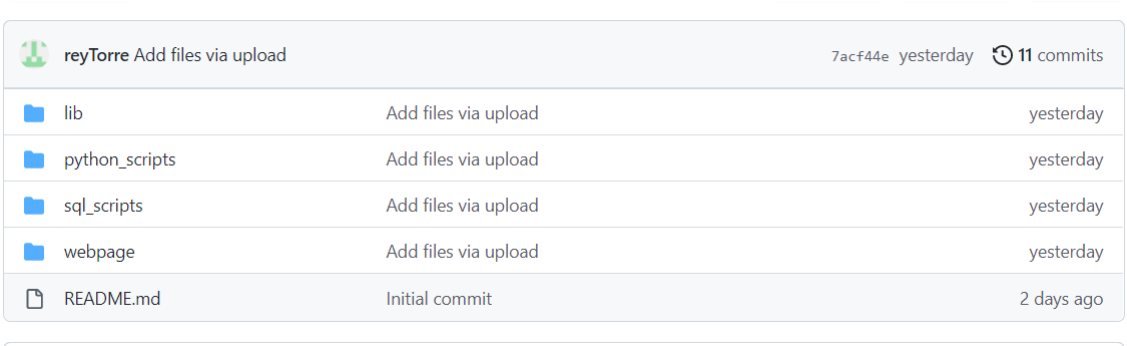
Link for complete documentation: [Staging Environment (Full Documentation)]

J.6: Staging Environment Testing (Database Connection Results)

	Type	Database typ	DB Version	Hostnam	Host IP	SID	Port	OS Type	DB user	DB password	Status
1	Local	MySQL	[redacted]		127.0.0.1		3306	Windows 10	root	p@ssw0rd	Failed
2	Local	SQL Server			127.0.0.1		1433	Windows 10	sa	p@ssw0rd	Failed
3	Local	PostgreSQL			127.0.0.1		5455	Windows 10	postgres	p@ssw0rd	Failed
4	Approved	MariaDB	5.5.41	iqobal	10.0.0.86	NA	3306	Ubuntu 14.04 LTS 64bit	root	root	Blocked
5	Approved	MySQL	8.0.15	jante	10.0.0.22	NA	3306	Ubuntu 16.04 LTS 64bit	root	root	Blocked
6	Approved	Oracle	11.2.0.1.0 Enterprise Edition	rafa4	10.0.1.11	rafa4	1521	Oracle Enterprise Linux 5.10 64bit	sys	master	Blocked
7	Approved	Oracle	19.3.0.0.0 Enterprise Edition	jodaka	10.0.1.16	ORCL19C	1521	Oracle Enterprise Linux 7.5 64bit	sys	master	Blocked
8	Approved	PostgreSQL	14.2.0	laboi	10.0.2.23	NA	5432	Ubuntu 18.04.3 LTS 64bit	postgres	postgres	Blocked
9	Approved	SQL Server	2005 Express Edition	hijado	10.0.0.61	SQLEXPRESS	1433	Windows Server 2003 Enterprise 32bi	sa	Master007	Blocked
10	Approved	SQL Server	2016 Standard Edition	ithona	10.0.0.94	MSSQLSERVER	1433	Windows Server 2016 Enterprise 64bi	sa	Master007	Blocked

Link for complete documentation: [Staging Environment db connection]

Appendix K: Code



reyTorre Add files via upload	7acf44e yesterday	11 commits
lib	Add files via upload	yesterday
python_scripts	Add files via upload	yesterday
sql_scripts	Add files via upload	yesterday
webpage	Add files via upload	yesterday
README.md	Initial commit	2 days ago

Link to all files: https://github.com/reyTorre/db_ims_final

Appendix L: ZAP Alerts

Alert	Affected HTTP
Cross Site Scripting	https://db-ims-module.online/idx_registration.php?qres=%3C%2Fp%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E%3Cp%3E
Absence of Anti-CSRF Tokens	https://db-ims-module.online/add_connection_details.php https://db-ims-module.online/add_connection_details.php?db_alias=ZAP&host_address=ZAP&orcdb=ZAP&password=ZAP&platform=MY&port=1&username=ZAP https://db-ims-module.online/analyze_env.php https://db-ims-module.online/db_connection_edit.php?q=edit https://db-ims-module.online/db_connection_edit.php?q=remove https://db-ims-module.online/home.php https://db-ims-module.online/idx_recovery.php https://db-ims-module.online/idx_recovery.php?email=foo-bar%40example.com https://db-ims-module.online/idx_registration.php https://db-ims-module.online/idx_registration.php?qres=Duplicate%20Email https://db-ims-module.online/idx_registration.php?qres=Failed%20to%20Sent%20Email https://db-ims-module.online/idx_registration.php?qres=Invalid%20Email https://db-ims-module.online/inventory.php https://db-ims-module.online/logs.php https://db-ims-module.online/profile.php?qpg=notification https://db-ims-module.online/profile.php?qpg=profile
Application Error Disclosure	https://db-ims-module.online/add_connection_details.php https://db-ims-module.online/add_connection_details.php?db_alias=ZAP&host_address=ZAP&orcdb=ZAP&password=ZAP&platform=MY&port=1&username=ZAP https://db-ims-module.online/analyze_env.php https://db-ims-module.online/db_connection_edit.php?q=edit https://db-ims-module.online/db_connection_edit.php?q=remove https://db-ims-module.online/home.php https://db-ims-module.online/inventory.php https://db-ims-module.online/logs.php https://db-ims-module.online/profile.php?qpg=notification https://db-ims-module.online/profile.php?qpg=profile https://db-ims-module.online/auto_user_action.php?q=login
Content Security Policy (CSP)Header Not Set	https://db-ims-module.online https://db-ims-module.online/ https://db-ims-module.online/add_connection_details.php https://db-ims-module.online/add_connection_details.php?db_alias=ZAP&host_address=ZAP&orcdb=ZAP&password=ZAP&platform=MY&port=1&username=ZAP https://db-ims-module.online/analyze_env.php https://db-ims-module.online/db_connection_edit.php?q=edit https://db-ims-module.online/db_connection_edit.php?q=remove https://db-ims-module.online/db_ims-connection-logs-May-25-2022.log https://db-ims-module.online/favicon.ico https://db-ims-module.online/home.php https://db-ims-module.online/idx_product_specs.php https://db-ims-module.online/idx_recovery.php https://db-ims-module.online/idx_recovery.php?email=foo-bar%40example.com https://db-ims-module.online/idx_registration.php https://db-ims-module.online/idx_registration.php?qres=Duplicate%20Email https://db-ims-module.online/idx_registration.php?qres=Failed%20to%20Sent%20Email https://db-ims-module.online/idx_registration.php?qres=Invalid%20Email https://db-ims-module.online/index.php https://db-ims-module.online/inventory.php https://db-ims-module.online/logs.php https://db-ims-module.online/profile.php?qpg=notification https://db-ims-module.online/profile.php?qpg=profile https://db-ims-module.online/robots.txt https://db-ims-module.online/sitemap.xml https://db-ims-module.online/auto_user_action.php?q=login
Directory Browsing	https://db-ims-module.online/css/ https://db-ims-module.online/Images/ https://db-ims-module.online/js/ https://db-ims-module.online/Styles/

Missing Anti-clickjackingHeader	https://db-ims-module.online https://db-ims-module.online/ https://db-ims-module.online/add_connection_details.php https://db-ims-module.online/add_connection_details.php?db_alias=ZAP&host_address=ZAP&orcdb=ZAP&password=ZAP&platform=MY&port=1&username=ZAP https://db-ims-module.online/analyze_env.php https://db-ims-module.online/db_connection_edit.php?q=edit https://db-ims-module.online/db_connection_edit.php?q=remove https://db-ims-module.online/home.php https://db-ims-module.online/idx_product_specs.php https://db-ims-module.online/idx_recovery.php https://db-ims-module.online/idx_recovery.php?email=foo-bar%40example.com https://db-ims-module.online/idx_registration.php https://db-ims-module.online/idx_registration.php?qres=Duplicate%20Email https://db-ims-module.online/idx_registration.php?qres=Failed%20to%20Sent%20Email https://db-ims-module.online/idx_registration.php?qres=Invalid%20Email https://db-ims-module.online/index.php https://db-ims-module.online/inventory.php https://db-ims-module.online/logs.php https://db-ims-module.online/profile.php?qpg=notification https://db-ims-module.online/profile.php?qpg=profile https://db-ims-module.online/auto_user_action.php?q=login
Parameter Tampering	https://db-ims-module.online/auto_user_action.php?q=login https://db-ims-module.online/auto_user_action.php?q=login
Cookie No HttpOnly Flag	https://db-ims-module.online/auto_user_action.php?q=login https://db-ims-module.online/auto_user_action.php?q=login
Cookie Without Secure Flag	https://db-ims-module.online/home.php
Cookie without SameSite Attribute	https://db-ims-module.online/home.php
Cross-Domain JavaScriptSource File Inclusion	https://db-ims-module.online/add_connection_details.php https://db-ims-module.online/add_connection_details.php?db_alias=ZAP&host_address=ZAP&orcdb=ZAP&password=ZAP&platform=MY&port=1&username=ZAP https://db-ims-module.online/analyze_env.php https://db-ims-module.online/db_connection_edit.php?q=edit https://db-ims-module.online/db_connection_edit.php?q=remove https://db-ims-module.online/home.php https://db-ims-module.online/idx_registration.php https://db-ims-module.online/idx_registration.php?qres=Duplicate%20Email https://db-ims-module.online/idx_registration.php?qres=Failed%20to%20Sent%20Email https://db-ims-module.online/idx_registration.php?qres=Invalid%20Email https://db-ims-module.online/inventory.php https://db-ims-module.online/logs.php https://db-ims-module.online/profile.php?qpg=notification https://db-ims-module.online/profile.php?qpg=profile
Server Leaks Information via"X-Powered-By" HTTP Response Header Field(s)	https://db-ims-module.online https://db-ims-module.online/ https://db-ims-module.online/add_connection_details.php https://db-ims-module.online/add_connection_details.php?db_alias=ZAP&host_address=ZAP&orcdb=ZAP&password=ZAP&platform=MY&port=1&username=ZAP https://db-ims-module.online/analyze_env.php https://db-ims-module.online/auto_user_action.php?loc=index.php&q=logout https://db-ims-module.online/db_connection_edit.php?q=edit https://db-ims-module.online/db_connection_edit.php?q=remove https://db-ims-module.online/home.php https://db-ims-module.online/idx_product_specs.php https://db-ims-module.online/idx_recovery.php https://db-ims-module.online/idx_recovery.php?email=foo-bar%40example.com https://db-ims-module.online/idx_registration.php https://db-ims-module.online/idx_registration.php?qres=Duplicate%20Email https://db-ims-module.online/idx_registration.php?qres=Failed%20to%20Sent%20Email https://db-ims-module.online/idx_registration.php?qres=Invalid%20Email https://db-ims-module.online/index.php https://db-ims-module.online/inventory.php https://db-ims-module.online/logs.php https://db-ims-module.online/profile.php?qpg=notification https://db-ims-module.online/profile.php?qpg=profile https://db-ims-module.online/auto_user_action.php?q=login https://db-ims-module.online/auto_user_action.php?q=register

Timestamp Disclosure - Unix	https://db-ims-module.online/css/bootstrap.min.css https://db-ims-module.online/lib/tempusdominus/js/moment-timezone.min.js
X-Content-Type-OptionsHeader Missing	https://db-ims-module.online https://db-ims-module.online/ https://db-ims-module.online/add_connection_details.php https://db-ims-module.online/add_connection_details.php?db_alias=ZAP&host_address=ZAP&orcdb=ZAP&password=ZAP&platform=MY&port=1&username=ZAP https://db-ims-module.online/analyze_env.php https://db-ims-module.online/css/bootstrap.min.css https://db-ims-module.online/css/connectionform.css https://db-ims-module.online/css/recoveryform.css https://db-ims-module.online/css/regform.css https://db-ims-module.online/css/style.css https://db-ims-module.online/db_connection_edit.php?q=edit https://db-ims-module.online/db_connection_edit.php?q=remove https://db-ims-module.online/home.php https://db-ims-module.online/idx_product_specs.php https://db-ims-module.online/idx_recovery.php https://db-ims-module.online/idx_recovery.php?email=foo-bar%40example.com https://db-ims-module.online/idx_registration.php https://db-ims-module.online/idx_registration.php?qres=Duplicate%20Email https://db-ims-module.online/idx_registration.php?qres=Failed%20to%20Sent%20Email https://db-ims-module.online/idx_registration.php?qres=Invalid%20Email https://db-ims-module.online/Images/background_dbims.png https://db-ims-module.online/Images/dbims_banner.gif https://db-ims-module.online/Images/dbims_logo.png https://db-ims-module.online/Images/product_specifications.png https://db-ims-module.online/Images/step1_adding_db_instance.png https://db-ims-module.online/Images/step2_install_stored_procedure.png https://db-ims-module.online/Images/step3_triger_stored_procedure.pnghttps://db-ims-module.online/Images/step4_display_analysis.png https://db-ims-module.online/Images/user_profiles/default_profile.png https://db-ims-module.online/Images/why_dbims.png https://db-ims-module.online/index.php https://db-ims-module.online/inventory.php https://db-ims-module.online/js/auth_user_js.js https://db-ims-module.online/js/change_user_js.jshttps://db-ims-module.online/js/connectdb_js.js https://db-ims-module.online/js/delete_connection_js.js https://db-ims-module.online/js/edit_connection.js https://db-ims-module.online/js/inventory_js.js https://db-ims-module.online/js/main.js https://db-ims-module.online/js/recover_email_js.js https://db-ims-module.online/lib/easing/easing.min.js https://db-ims-module.online/lib/owlcarousel/assets/owl.carousel.min.css https://db-ims-module.online/lib/owlcarousel/owl.carousel.min.js https://db-ims-module.online/lib/tempusdominus/css/tempusdominus-bootstrap-4.min.css https://db-ims-module.online/lib/tempusdominus/js/moment-timezone.min.js https://db-ims-module.online/lib/tempusdominus/js/moment.min.js https://db-ims-module.online/lib/tempusdominus/js/tempusdominus-bootstrap-4.min.js https://db-ims-module.online/lib/waypoints/waypoints.min.js https://db-ims-module.online/logs.php https://db-ims-module.online/profile.php?qpg=notification https://db-ims-module.online/profile.php?qpg=profile https://db-ims-module.online/Styles/Stylesheet.css https://db-ims-module.online/auto_user_action.php?q=login
Information Disclosure - Sensitive Information in URL	https://db-ims-module.online/add_connection_details.php?db_alias=ZAP&host_address=ZAP&orcdb=ZAP&password=ZAP&platform=MY&port=1&username=ZAP https://db-ims-module.online/idx_recovery.php?email=foo-bar%40example.com

Information Disclosure -Suspicious Comments	https://db-ims-module.online/js/change_user_js.jshttps://db-ims-module.online/js/connectdb_js.js https://db-ims-module.online/js/inventory_js.js https://db-ims-module.online/lib/tempusdominus/js/moment-timezone.min.js https://db-ims-module.online/lib/tempusdominus/js/moment.min.js https://db-ims-module.online/lib/tempusdominus/js/tempusdominus-bootstrap-4.min.js https://db-ims-module.online/add_connection_details.php https://db-ims-module.online/add_connection_details.php?db_alias=ZAP&host_address=ZAP&orcdb=ZAP&password=ZAP&platform=MY&port=1&username=ZAP
Re-examine Cache-control Directives	https://db-ims-module.online https://db-ims-module.online/ https://db-ims-module.online/idx_product_specs.php https://db-ims-module.online/idx_recovery.php https://db-ims-module.online/idx_recovery.php?email=foo-bar%40example.com https://db-ims-module.online/idx_registration.php https://db-ims-module.online/idx_registration.php?qres=Duplicate%20Email https://db-ims-module.online/idx_registration.php?qres=Failed%20to%20Sent%20Email https://db-ims-module.online/idx_registration.php?qres=Invalid%20Email https://db-ims-module.online/index.php https://db-ims-module.online/auto_user_action.php?q=login

Appendix L: List of Files

M.1: List of Stored Procedures

No	Name	Object Name	Description	Type
1	Generate Unique User ID	generate_unique_user_id	Automatically Generate User ID, 32 characters that must not be similar to any User ID	Function
2	Validate Email Address	validate_email	Validate Email Address. Must follow: [address]@[domain].[com. etc.]	Function
3	Generate Username	generate_username	Creates Username following the format: USER + (3 digit Count Number) + (3 random number)	Function
4	Check if Email Address Exists	check_email_address_exist	Check if Email Address Exists	Function
5	Create User Record	create_user_record	Insert New Record to User and Email Address Tables. If Email Address is correct then: Check if Email exists in Email Address Table If not exists: 1) Insert generated Username, generated Password, and generated userid to the Users table 2) Insert generated userid, Email, and generated Password to the Email Address table Else Do nothing	Proceedure
6	Validate Record Creation	validate_record_creation	Check if create_user_record is successful or not	Function
7	Generate Initial Password	generate_initial_password	Automatically Generates 10 character password with combination of alphanumeric characters with a Special Character and Upper Case	Function
8	Checkdb connection details	check_db_connection_details	Check Format IP Address Format, Port Format and Platform	Function
9	Get User ID	get_user_id	Get User ID using email or Username	Function
10	Create Temporary Connection Details	create_connection_details	Insert New Record to temporary connection table. Get UserID using user information. Get DB to check if the format is ok. if UserID =/= NULL and dbCheck is correct: insert connection details to temporary table	Procedure
11	Get Temporary Connection ID	get_temp_con_id	Gets Temporary Connection ID from parameters	Function
12	Change Connection Status	change_connection_status	Updates connection detail status. If 1 --> Change status to Connected, 0 ---> Change status to Failed	Procedure
13	Generate Connection ID for	generate_connection_id	Generate Permanent Connection ID	Function

	Permanent Table			
14	Transfer Connection Details	transfer_connection_details	Transfer Connection Details from Temp Table to Permanent Table	Procedure
15	Create Salt for Password	generate_random_string	Random Encoded String for salting	Function
16	Encypted Password	get_salted_password	Extract Password from Database	Function
17	Edit Connection Details	edit_connection_details	Edit Connection Details	Procedure
18	Delete Connection Details	remove_connection_details	Delete Connection Details	Procedure
19	Get Username and Password	get_username_and_password	Get Username and Password from Users using email address	Procedure
20	Delete User	remove_user	Delete user from database	Procedure
21	Remove Temp Connection	remove_temporary_connection_details	Remove Temporary Connection ID	Procedure
22	Transfer to Permanent Connection	transfer_to_permanent_connection_details	Copy Temporary Connection details and create a permanent connection	Procedure
23	Get New Password	get_new_password	Get Password Details	Function
24	Rename db Connection	rename_db_connection	Rename Permanent Connection Details	Procedure
25	Get uptime Cycle	get_uptime_cycle	Get current uptime cycle for SQL Server Connection	Function
26	Deactivate other uptime	sp_deactive_other_uptime	Change Uptime Status from 'A' (Active) to 'X' (Deactivated)	Procedure
27	App Uptime Check	sp_add_uptime_check	Add Uptime Record	Procedure
28	Validate Uptime Check	sp_validate_uptime_check	Verify if uptime > db uptime	Procedure
29	Check Lastdb Uptime	sp_check_last_db_uptime	check if latest and uptime follows a chronological flow in inserting records	Procedure
30	Validate Uptime Records	validate_uptime_records	validate and executes procedures when conidtions are met	Procedure
31	Add Database Statuses	sp_add_database_statuses	Add SQL Server database statuses	Procedure
32	Update Database Details	sp_update_database_details	Update JSON field for SQL Server uptime records	Procedure
33	De-jsonify Uptime Records	dejson_current_uptime_records	De-jsonify SQL Server uptime record	Procedure
34	Insert Max Connection	insert_db_max_connection_details	Extract and record Max Connection Record for SQL Server	Procedure

35	Stage Breakdown Logs	stage_breakdown_logs	Extract and record Connections for SQL Server	Procedure
36	SP Log to System log table	sp_log_to_systems_log_tbl	Record System Logs	Procedure
37	Stored SP History	sp_store_to_sp_history	Record SP History	Procedure
38	Get active Sessions	sp_get_active_sessions	Extract and record all active sessions for SQL Server	Procedure
39	Get Breakdown Connections	sp_get_breakdown_connections	Extract and record all connections for SQL Server	Procedure
40	Get Table Name	fxn_get_tbl_name	Get corresponding table name from SP	Function
41	Table Reference	sp_table_reference	Get corresponding table name from SP and Platform	Procedure
42	Mask Password	fxn_mask_password	Mask Password upon display to front end	Function
43	Insert Uptime Status	sp_insert_status	Record Uptime Status	Procedure
44	Get Username	get_username	Get Username from UserID	Function
45	Generate Growth rate Backup Pivot Table	generate_growthrate_back_up_pivot_tbl	Create Pivot Table for Growth Rate	Procedure
46	Convert To Bytes	fxn_convert_to_bytes	Convert Memory String to bytes	Function
47	Log Py DB Details	sp_log_py_db_details	Log Python SP Response Time and pyodbc details	Procedure
48	Get Database Version	fx_get_db_version	Get Database Version MS: substring to nearest '-' PG: substring to nearest ',' Others: Get the actual version	Function
49	Update User password	update_user_password	Update User password and salt using email address as indicator	Procedure
50	Generate Job Default Configuration	sp_create_default_job_configuration	Populate DBIMS jobs table with default configuration	Procedure
51	Get Unchecked Installation Status	get_unchecked_installation_status	Check Job installation Status	Function
52	Alter Job Installation Status	alter_job_installation_status	Update installation status to Uninstalled if (A or N) else Installed	Procedure
53	Get Number of Unchecked Status	get_number_of_unchecked_status	Get Number of Unchecked jobs	Function
54	Decipher Data string	fx_decipher_datastring	Outputs scheduled or occurrence values from data string	Function
55	Convert Occurrence String to table	sp_convert_occurance_str_to_tbl	Changes Scheduling Occurrence to its table equivalence	Procedure
56	Check Time Format	fx_check_time_format	Validates if Time Format is correct (Frequency [FQ] or ST [Set Time])	Function

57	Extract Number from String	fx_extract_number_from_string	Converts Time string to its Integer Equivalence by parts	Function
58	Convert String to time	fx_convert_string_to_time	Converts Time string to its time equivalence	Function
59	Convert String to Table	sp_convert_time_str_to_tbl	Converts Time String to its table equivalence	Procedure
60	Update Job Configuration	sp_update_job_configuration	Update Job Configuration	Procedure
61	Update Last Run	update_last_run	Update Job with last run	Procedure
62	Populate Automated Table	sp_populate_automated_tbl	Populate Automated Table for reference during automated run	Procedure
63	Get Priority Queue	fx_get_priority_queue	Get top most list for automated run	Function
64	Generate Notification ID	generate_notification_id	Generate Notification ID	Function
65	Get User ID from Connection String	get_user_id_from_con_string	Get Userd ID from db_con_string_id	Function
66	Update Username	sp_update_user_name	Update Username	Procedure
67	Archive Connection Details	archive_connection_details	Delete all records from a platform using db con string id	Procedure
68	Oracle Record Connections	or_record_connections	Insert Connection Records for Oracle	Procedure
69	Oracle Instance Information	or_instance_information	Insert Instance Information for Oracle	Procedure
70	Get Username	get_user_name	Get Username from UserID	Procedure
71	Read All Notifications	sp_read_all	set all notifications to read	Procedure
72	Generate SQL ID	generate_sql_id	Create SQL ID to link to DE-JSONIFIED table	Function

M.2: List of Views

No	Object name	Description
1	vw_userinfo	Displays complete user information
2	vw_db_connection_details_complete	Displays db connection along with user information
3	vw_db_connection_details_temporary	Displays temporary db connection along with user information
4	vw_table_details	table details extract from local database information schema
5	vw_uptime_check	Displays uptime connection details
6	vw_sql_connection_details	Displays most recent connection details extract from SQL Server
7	vw_sql_backup_growth_rate_six_months	displays the backup growthrate for the last six months
8	vw_sql_database_growth_rate_90_days	displays the growth rate for the last 90 days
9	vw_db_uptime_logs_graph	displays uptime data to be translated to a line graph

10	vw_db_status_check	displays SQL Server database statuses
11	vw_job_status_history	displays SQL Server job history
12	vw_sql_server_memory_capacity	aggregates data to the most recent sever memory for SQL SERVER
13	vw_sql_db_logs_cap	displays log capacity for SQL SERVER
14	vw_sql_db_logs_cap_size_comp	displays log capacity in percentage for SQL SERVER
15	vw_sql_backup_details	displays SQL SERVER backup details
16	vw_pg_connection_details	displays the most recent connection statistics for Postgres
17	vw_pg_longest_running_queries	displays the top 3 longest running queries for Postgres
18	vw_pg_active_sessions	displays Postgres active sessions
19	vw_pg_cache_hit_ratio_and_memory	displays critical hit ratio for indexes and tables (Postgres)
20	vw_pg_tbl_index_sizes	displays table index sizes for Postgres
21	vw_pg_tbl_cache_hit_ratio_below_80	displays cache hit ratio below 80 percent (table) for Postgres
22	vw_pg_idx_cache_hit_ratio_below_80	displays cache hit ratio below 80 percent (index) for Postgres
23	vw_pg_vacuum_summary	displays vacuum summary for Postgres
24	vw_pg_inherentance_map	displays Postgres role and inheritance map
25	chk_db_status_per_ten_mins	check database status every ten minutes
26	db_ims_job_summary	DB IMS job summary
27	vw_db_invetory	displays database inventory for all platforms
28	vw_sql_job_summary	displays SQL SERVER DB IMS job status
29	vw_pg_job_summary	displays Postgres DB IMS job status
30	vw_mysql_job_summary	displays MySQL DB IMS job status
31	vw_oracle_job_summary	displays Oracle DB IMS job status
32	vw_monitoring_details	displays monitornig details per platform and job
33	vw_mysql_to_py_steps	Converts mysql tsql to py function step
34	vw_job_reference_tbl	displays monitornig details per platform and job
35	vw_job_reference_tbl_filtered_view	filters jobs to be executed (automation)
36	vw_all_con_details	reference table jobs, database conenction and user
37	notify_user	displays notifications for user
38	vw_db_invetory_summary	displays inventory summary for all platforms
39	vw_my_connection_details	displays latest connection details for MySQL
40	table_size_and_index_info	displays table size and index information for MySQL
41	vw_my_active_sessions	displays active sessions for MySQL
42	vw_or_connection_details	displays latest connection details for Oracle
43	vw_check_read_notif	provides a tag for read and unread notifications