A REPORT

ON

AUTOMATION OF SEVERAL EXISTING PROCEDURES

By

Name of the student	**ID No.**
Hritik Soni	**2014A2PS0480P**

AT

JDA Software Solutions, Bangalore

A Practice School II Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(September, 2017)

A REPORT

ON

AUTOMATION OF SEVERAL EXISTING PROCEDURES

By

Name of the student	ID No.	Discipline
Hritik Soni	2014A2PS0480P	Civil Engineering

Prepared in the partial fulfillment of the

Practice School II Course

AT

JDA Software, Bangalore

A Practice School II Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(September, 2017)

Acknowledgements:

There is no project without sources of motivation. In my case I would first like to thank BITS Pilani for giving me the opportunity for an Internship here at JDA Software, the HR Manager for accepting me as an intern, Mr. Raghuram Thaiyar (My Manager) for providing an excellent mentor for my project, Mr. Harish Ramapur Narayanappa(My Primary Mentor), Gaston Mascarenhas (My Secondary Mentor), Mr. Anand Yadav for providing me guidance for my project and finally my parents and everyone else who indirectly helped me during the course of my project.

Abstract Sheet

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)

Practice School Division

Station:	**JDA Software**	**Centre:**	**Bangalore**
Duration:	**6 Months**	**Date of Start:**	**4th July 2017**

Date of Submission: **12th September 2017**

Title of the Project: **AUTOMATION OF SEVERAL EXISTING PROCEDURES**

ID No./Name(s)/: Discipline(s) of the Student(s) **2014A2PS0480P/Hritik Soni/B.E. Civil**

Name(s) and: Designation(s) of the expert(s)
Harish Ramapur Narayanappa / Enterprise Senior Architect
Gaston Mascarenhas / Project Manager

Name(s) of the: PS Faculty **Mr. Vineet Garg**

Key Words: **Supply chain management, SCPO, Log Analyzer, Upgrade Automation**

Project Areas: **Development of Automation Interfaces**

Abstract: **My Project is a combination of three subprojects which achieve different goals with same concept i.e. Automation of Redundant and Time-Consuming Activities using GUI interface.**

Following are my three projects with a brief description:

1. **Automated Log Analyzer: Helps in finding causes of incidents by analyzing log files for errors by preparing frequency histograms of all kinds of errors.**
2. **JDA Server Manager: Web Interface for managing services running on JDA Servers and also classifying Servers based on their related Customer.**
3. **Upgrade Manager: All-in-one interface for performing one step migration while adhering to official procedures.**

Signature(s) of Student(s) Signature of PS Faculty

Date 9th September 2017

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)

PRACTICE SCHOOL DIVISION

Response Option Sheet

Station: JDA Software Center: Bangalore

ID No. & Name(s): Hritik Soni

Title of the Project: **AUTOMATION OF SEVERAL EXISTING PROCEDURES**

Usefulness of the project to the on-campus courses of study in various disciplines. Project should be scrutinized keeping in view the following response options. Write Course No. and Course Name against the option under which the project comes.

Refer Bulletin for Course No. and Course Name.

Code No.	Response Option	Course No.(s) & Name
1.	A new course can be designed out of this project.	No
2.	The project can help modification of the course content of some of the existing Courses	No
3.	The project can be used directly in some of the existing Compulsory Discipline Courses (CDC)/ Discipline Courses Other than Compulsory (DCOC)/ Emerging Area (EA), etc. Courses	No
4.	The project can be used in preparatory courses like Analysis and Application Oriented Courses (AAOC)/ Engineering Science (ES)/ Technical Art (TA) and Core Courses.	No
5.	This project cannot come under any of the above mentioned options as it relates to the professional work of the host organization.	Yes

Signature of Student
Date:

Signature of Faculty
Date:

FORMAT FOR NO DUES CERTIFICATE

PS II station at: JDA Software, Bangalore Date: _________________

Name: Hritik Soni ID No. : 2014A2PS480P will be completing his/her Practice School Program on15th Dec' 2017. In case he /she has any dues, please report it below against your name. Incase he/she has no dues please write NO DUES and sign.

1. Organization Coordinator :______________________________________
2. Professional Expert :______________________________________
3. Librarian :______________________________________
4. Accounts Section :______________________________________
5. PS Faculty :______________________________________
6. Any Other :______________________________________
7. Any Other :______________________________________

Signature of PS Faculty

Table of Contents

INTRODUCTION

JDA is an application meant for interested customers to use it. Beyond development of the actual utility comes maintenance which is perhaps the most challenging part as unexpected situations can arise and they need to be dealt immediately. The Project comprises of three sub projects each serving to help the maintainability of the architecture of JDA in different aspects.

1. Automated Log File Analyzer:

   Whenever an error/incident is encountered, database logs are the main source of its identification (in many cases) and thus need to be inspected thoroughly. My first Project attempts to ease the inspection significantly. It has the following major functionalities:-

   1. GUI based mode for easy to use and self-explanatory functioning.
   2. Automatic/Script based mode for batch processing.
   3. Capability to handle several logs and several patterns of errors.
   4. Frequency histograms of errors found in Bar Chart based formats.
   5. Automatic mailing of the output charts at every specified duration.

2. JDA Server Manager:

   A key part of the maintenance is the ability to control services on remote and servers and being able to grab details like RAM free, CPU usage, etc. on them and also able to seamlessly do all this for any desired customer without much effort. This part of the project attempts to solve this problem via. providing a web interface to do all mentioned above.

3. Upgrade Manager:

   It is nature of every application to evolve and gain more abilities and thus the JDA application also gains new versions and the most challenging part is the ability to migrate existing customers to the latest platform. Long procedures already exist to do the same and are very time consuming. This part of the project attempts to simplify all the steps via a GUI based automated procedure.

Automated Log Analyzer

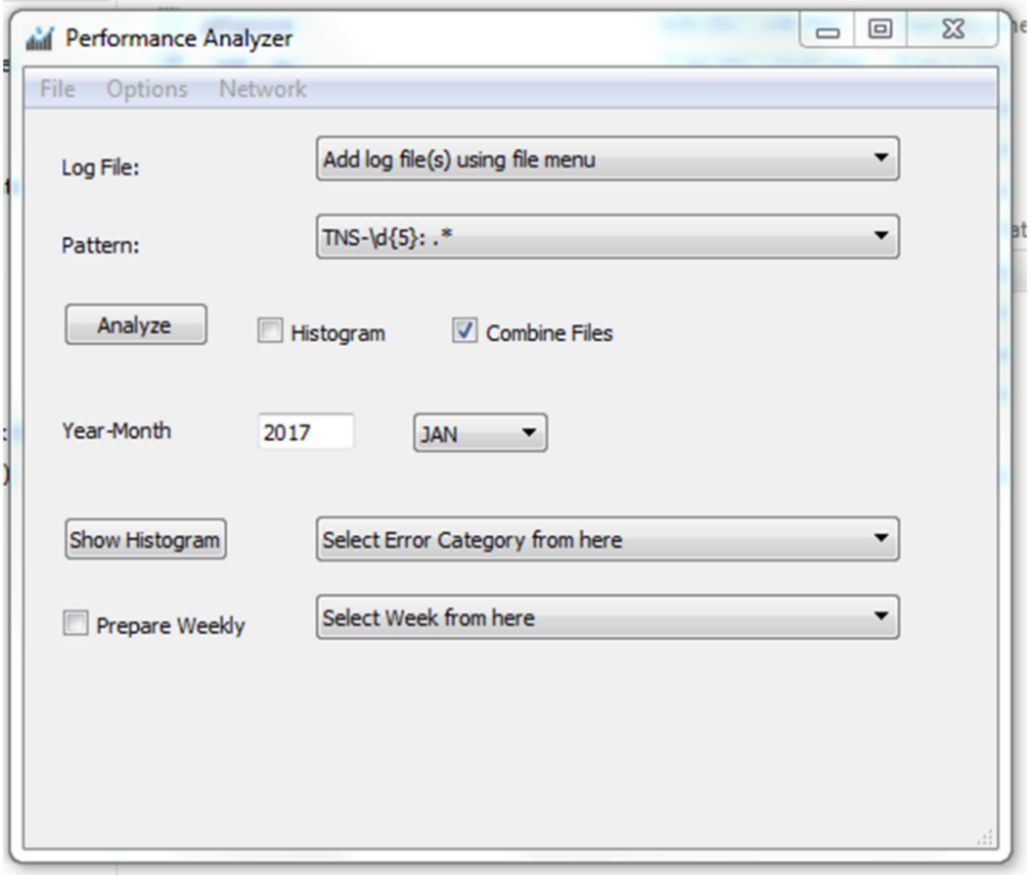
Every JDA application is associated with a database and every database has its own server just like the application. The choice of database for JDA happens to be Oracle which is an expensive database management system but very robust. During maintenance phase of the application, unexpected situations can arise. These situations are termed as incidents and a lot of incidents have their solutions hidden in the database logs. As of now, these logs are analyzed manually via. 3rd party tools but even through them it is quite time consuming.

Every error has a certain pattern and using creative programming all of these tasks can be turned to an automated task.

A Sample Error Would be:

```
TNS-12547: TNS:lost contact
    ns secondary err code: 12560
    nt main err code: 0
    nt secondary err code: 0

    nt OS err code: 0
```

There is a little bit more information related to the error like data information and the above is sufficient to demonstrate the point. As we can observe the error starts with TNS-….. This trend is common for every error and thus can be exploited to find several errors within thousands of lines in a fraction of a second. The following is a screenshot of the opening screen.



Log File: Add log file(s) using file menu

Pattern: TNS-\d{5}: .*

□ Histogram ☑ Combine Files

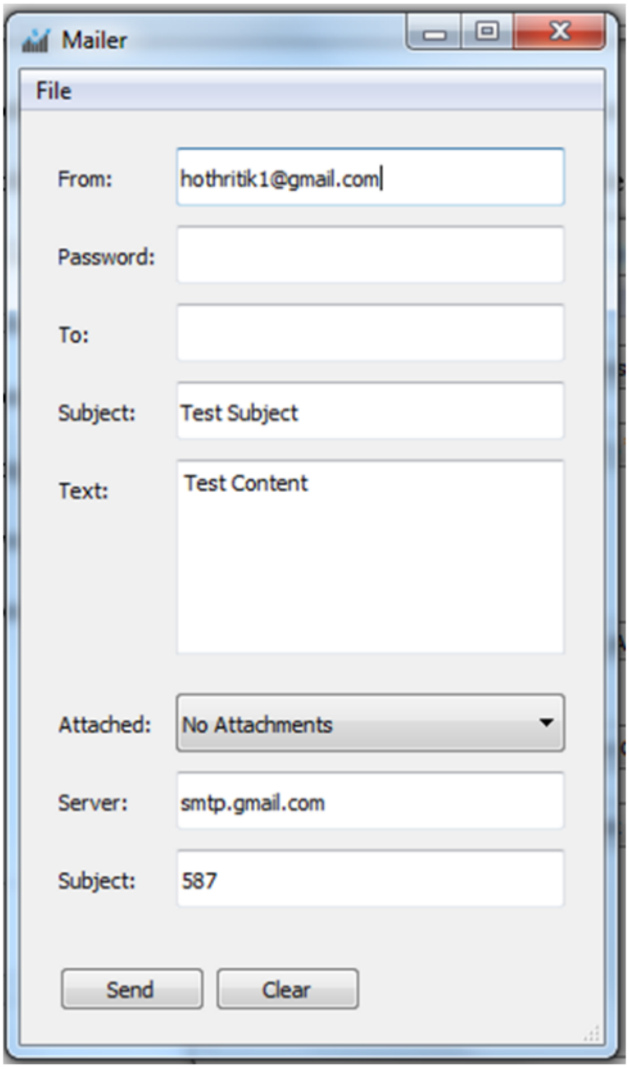
Year-Month 2017 JAN

Select Error Category from here

□ Prepare Weekly Select Week from here

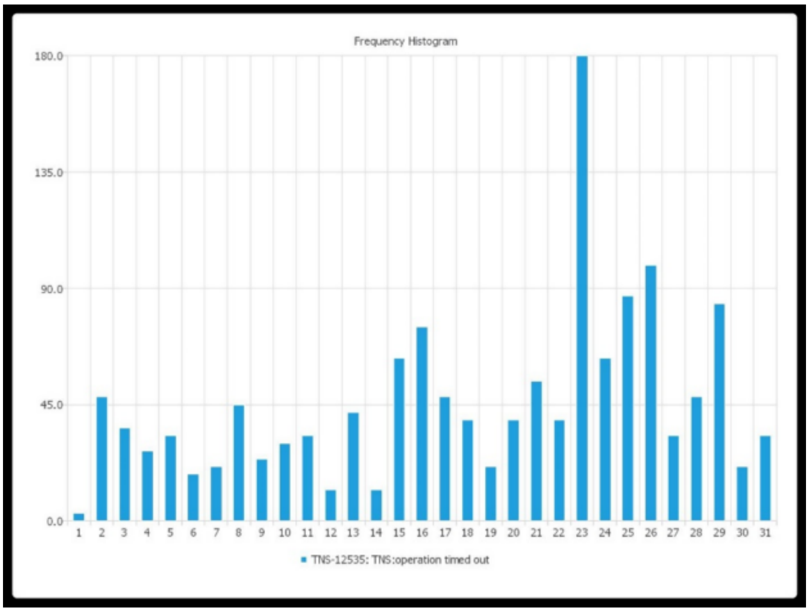
It looks very similar to a professional interface with a standard menu bar and widgets that serve the objective. Log file(s) can be selected from the File Menu and will appear in front of the Log File Label. Any custom error pattern can be entered or can be loaded via. Options menu. The combine files option gives the tool an ability to combine outputs of several log files together.

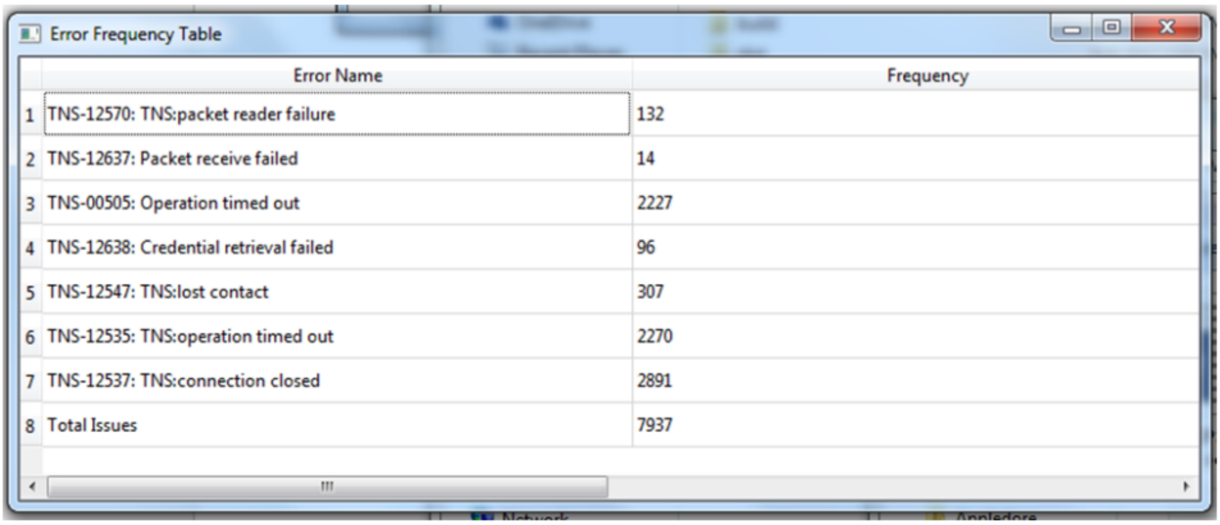
If someone is interested in more than finding the frequencies of the error he can also check the histogram check box to exploit date based information to prepare frequency histograms (monthly or weekly). The Network menu has a mailer tool for mailing the outputs generated to a certain someone. A view of the mailer is shown below:



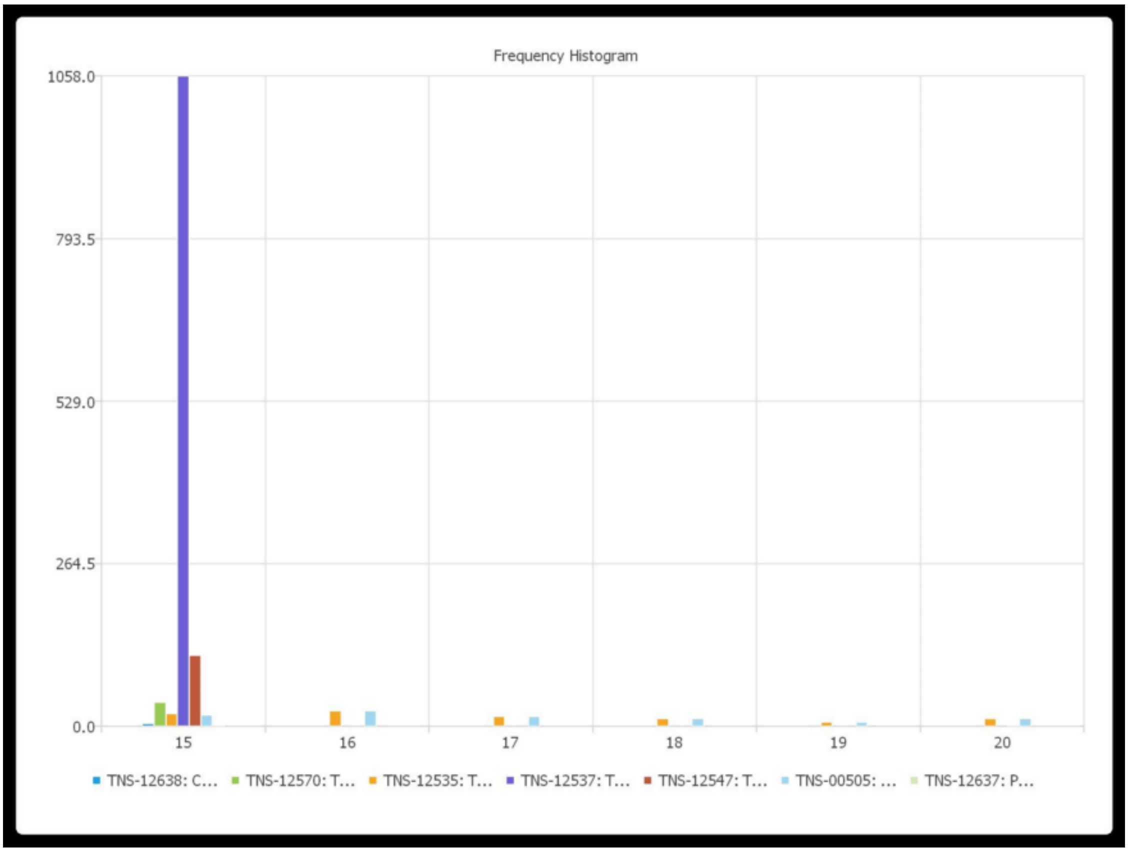
From: hothritik1@gmail.com
Password:
To:
Subject: Test Subject
Text: Test Content
Attached: No Attachments
Server: smtp.gmail.com
Subject: 587

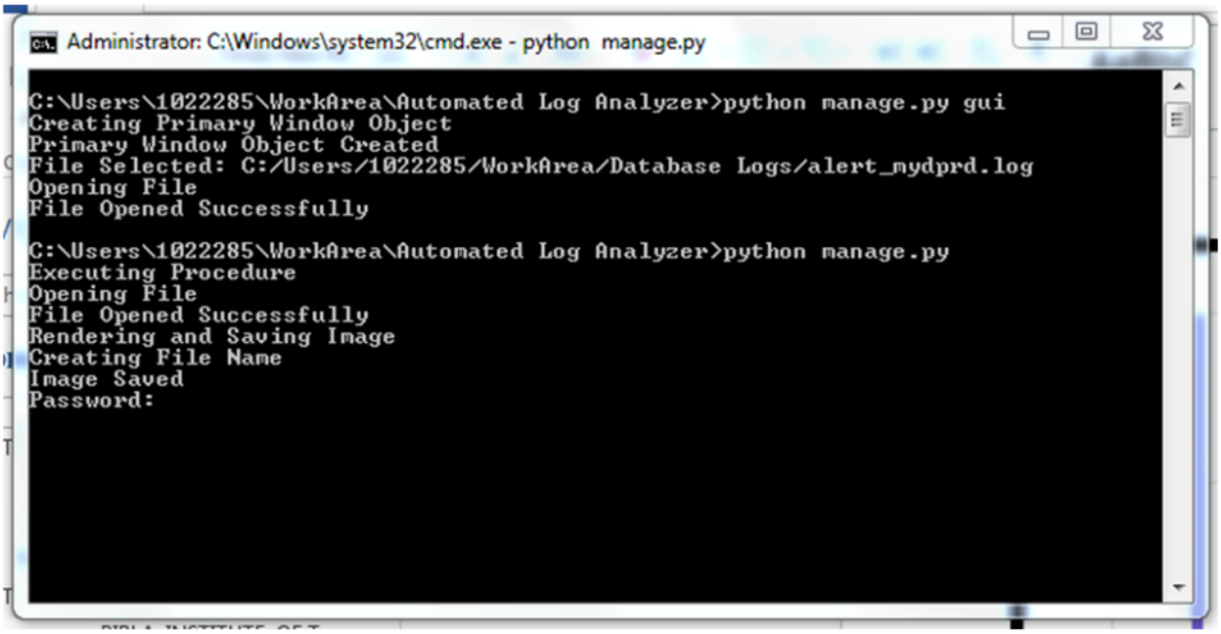
The mailer tool is just a side tool developed if for some reason someone wanted to mail something using a GUI interface. The main reason to develop it was mainly to develop the associated functions so that the mailing can be done without user intervention in console mode. In fact, the whole tool namely Automated Log Analyzer can run in auto mode without any user intervention. A few screenshots of the outputs and its functioning in auto mode is displayed below.





	Error Name	Frequency
1	TNS-12570: TNS:packet reader failure	132
2	TNS-12637: Packet receive failed	14
3	TNS-00505: Operation timed out	2227
4	TNS-12638: Credential retrieval failed	96
5	TNS-12547: TNS:lost contact	307
6	TNS-12535: TNS:operation timed out	2270
7	TNS-12537: TNS:connection closed	2891
8	Total Issues	7937




Administrator: C:\Windows\system32\cmd.exe - python manage.py

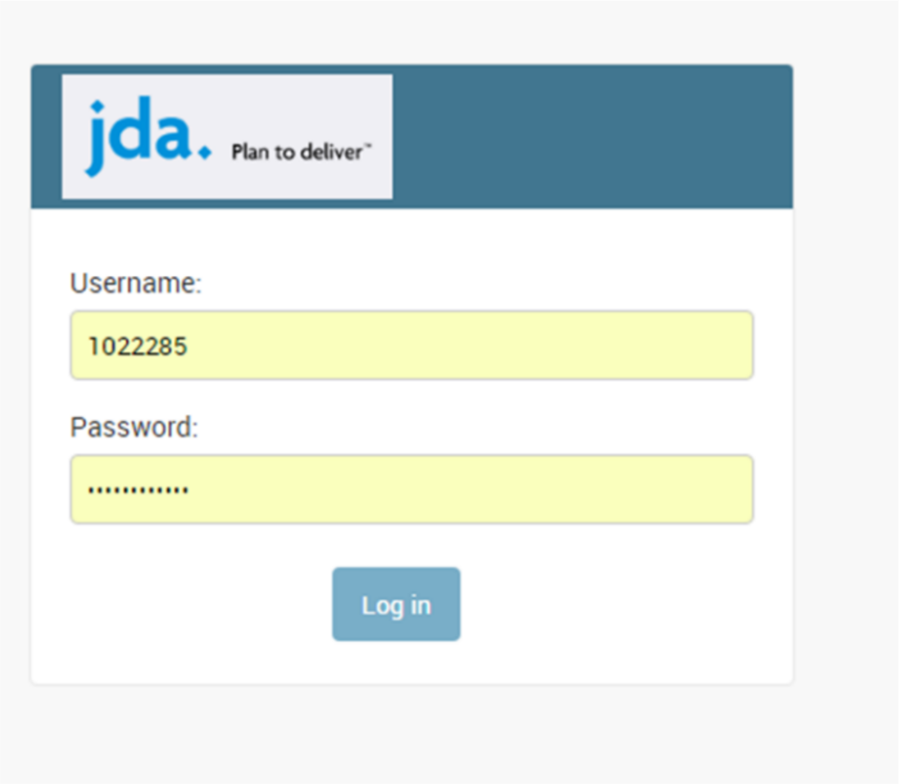
```
C:\Users\1022285\WorkArea\Automated Log Analyzer>python manage.py gui
Creating Primary Window Object
Primary Window Object Created
File Selected: C:/Users/1022285/WorkArea/Database Logs/alert_mydprd.log
Opening File
File Opened Successfully

C:\Users\1022285\WorkArea\Automated Log Analyzer>python manage.py
Executing Procedure
Opening File
File Opened Successfully
Rendering and Saving Image
Creating File Name
Image Saved
Password:
```

Future Planning

The tool is ready to be deployed on a centralized server to collect logs from various sources, analyze them and shoot mails on a pre-defined manner. The idea has been submitted and as soon as the community approves, this project will be deployed.

JDA Server Manager

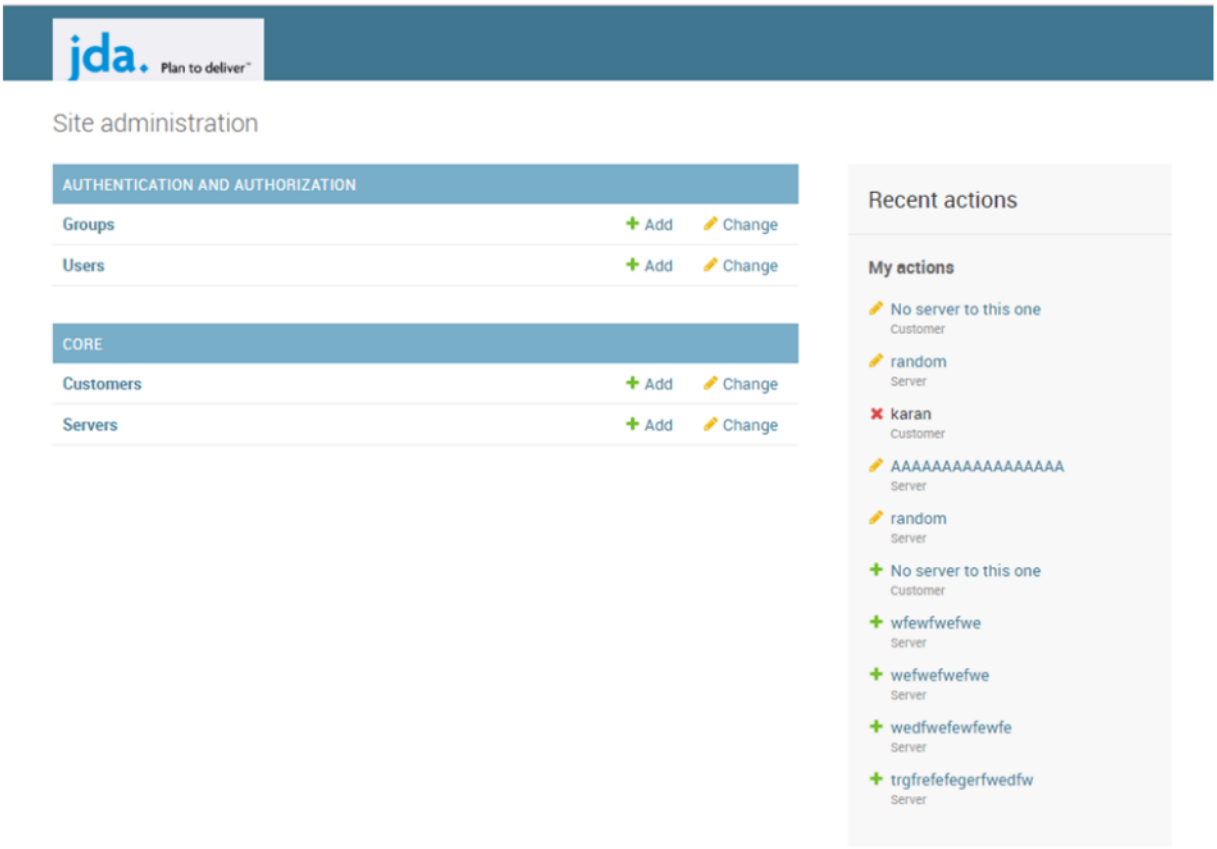
A key part of maintaining the application is to maintain the servers that run it. Also being able to easily control different parts of the application (services) is also important. This part of the project aims to simplify all of the above via a unified interface. This was named JDA Server Manager / JDAServerMan. An opening screen of the interface is shown below:



Username: 1022285

Password: ••••••••••••

The interface opens up with a login form asking for user credentials. Now, to explain its purpose, we can say that every application (in general) has some users. In many cases you want those users to be able to perform only specific tasks while some others might be allowed to perform some of the rest and so on. This is why the Server Manager is going to have a super-user (one with all rights) and some general user who can only see the servers and maybe able to start and stop services while the super-user has the ability to do all of this plus create new users, add new customers, add new servers to those customer and so on. Below is a screenshot with a logged in super-user:



Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	✎ Change
Users	+ Add	✎ Change

CORE		
Customers	+ Add	✎ Change
Servers	+ Add	✎ Change

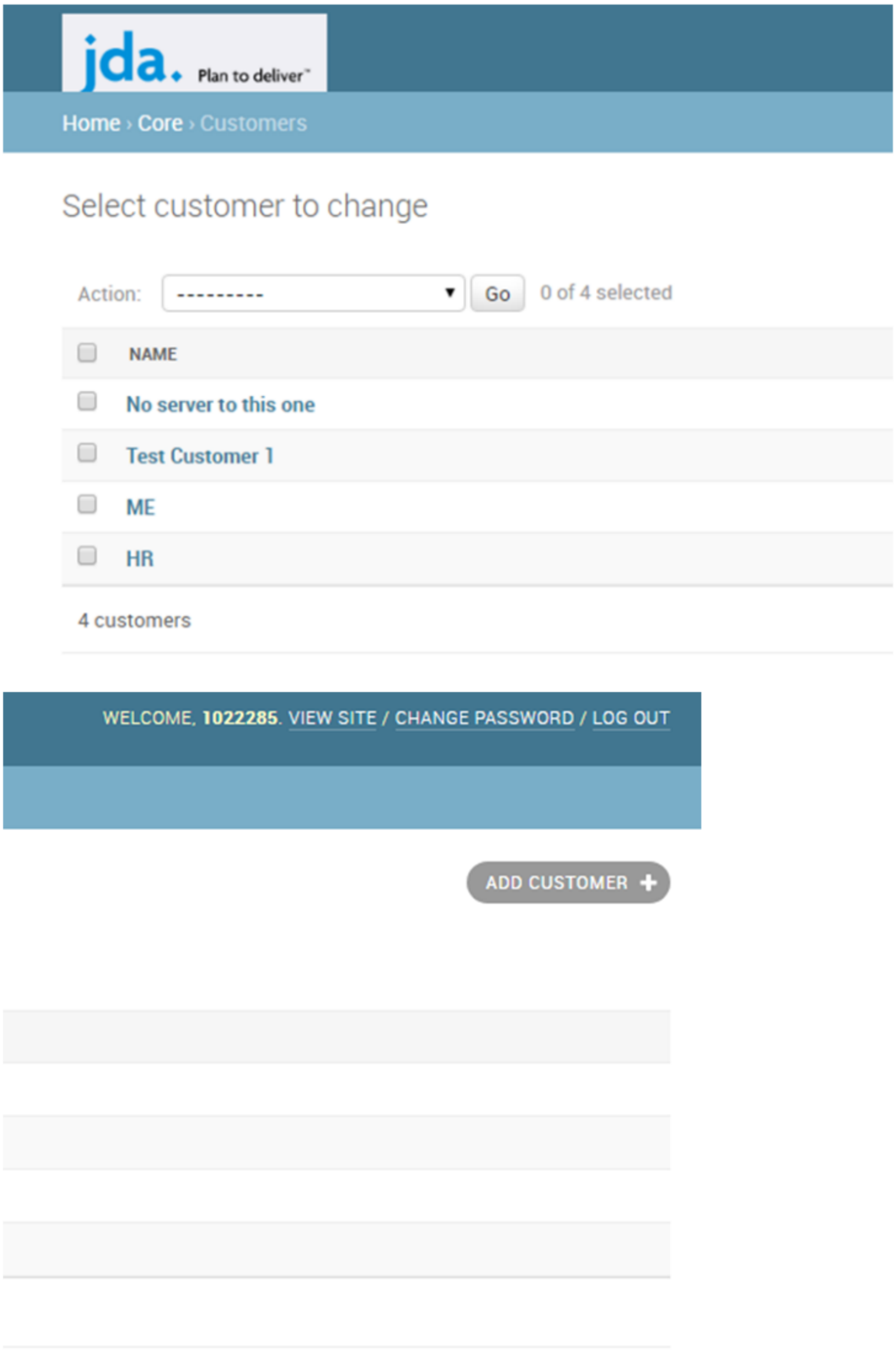
Recent actions

My actions

✎ No server to this one
Customer
✎ random
Server
✖ karan
Customer
✎ AAAAAAAAAAAAAAAA
Server
✎ random
Server
+ No server to this one
Customer
+ wfewfwefwe
Server
+ wefwefwefwe
Server
+ wedfwefewfewfe
Server
+ trgfrefefegerfwedfw
Server

As it can be seen above that the super-user can deal with users and everything else.

As mentioned previously, every server is going to be owned by a specific customer. So if we were to create a new customer we would go to the Customer section under core, click on the add customer button and add the servers associated with it there itself as shown below:

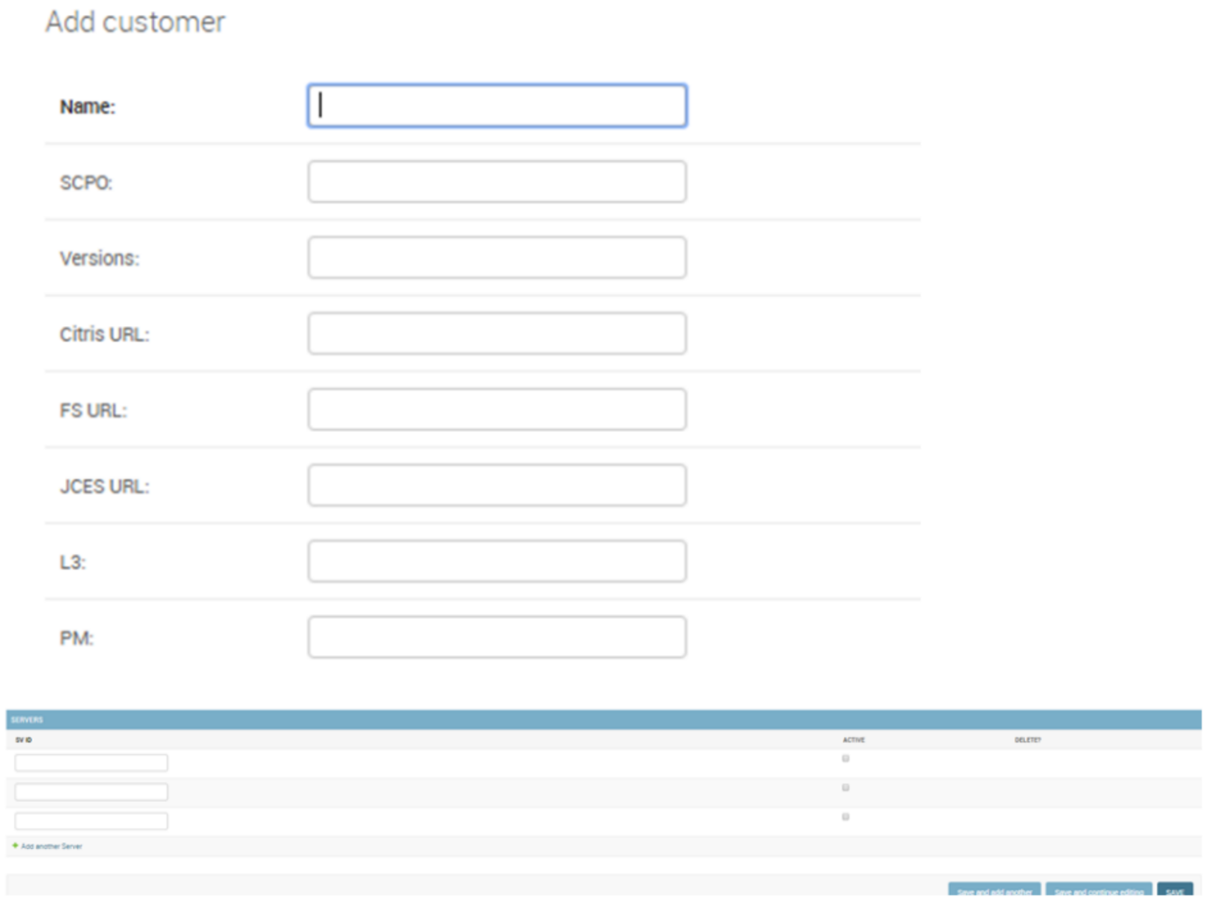


Select customer to change

0 of 4 selected

□	NAME
□	No server to this one
□	Test Customer 1
□	ME
□	HR

4 customers



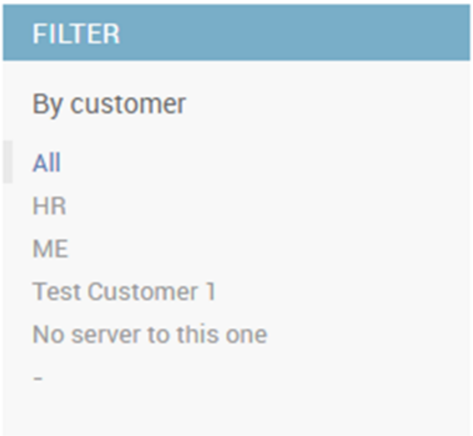
Add customer

Name:	
SCPO:	
Versions:	
Citris URL:	
FS URL:	
JCES URL:	
L3:	
PM:	

SERVERS

SV ID	ACTIVE	DELETE?
	□	
	□	
	□	

Now, to see what all servers are owned by a specific customer we would go to the Servers section under Core and use the customer filter to filter the required servers among the rest as shown:



FILTER

By customer

All

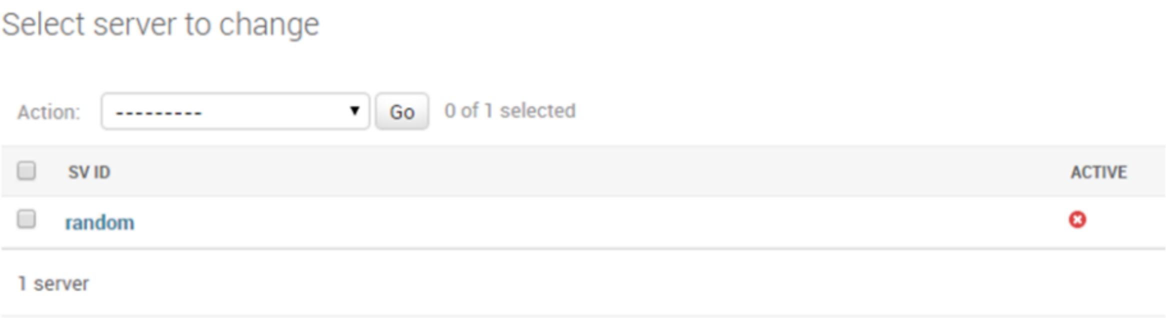
HR

ME

Test Customer 1

No server to this one

-



Select server to change

Action: --------- ▾ Go 0 of 1 selected

□	SV ID	ACTIVE
□	random	☒

1 server

FUTURE PLANNING

As of now the interface has the sole ability to filter servers by customers and being able to manage just the details of the servers and customer. The real challenge left is to fetch information from the servers like RAM Usage, etc. and the ability to monitor services running on remote servers. Once all this is done, the project should be ready to get deployed.

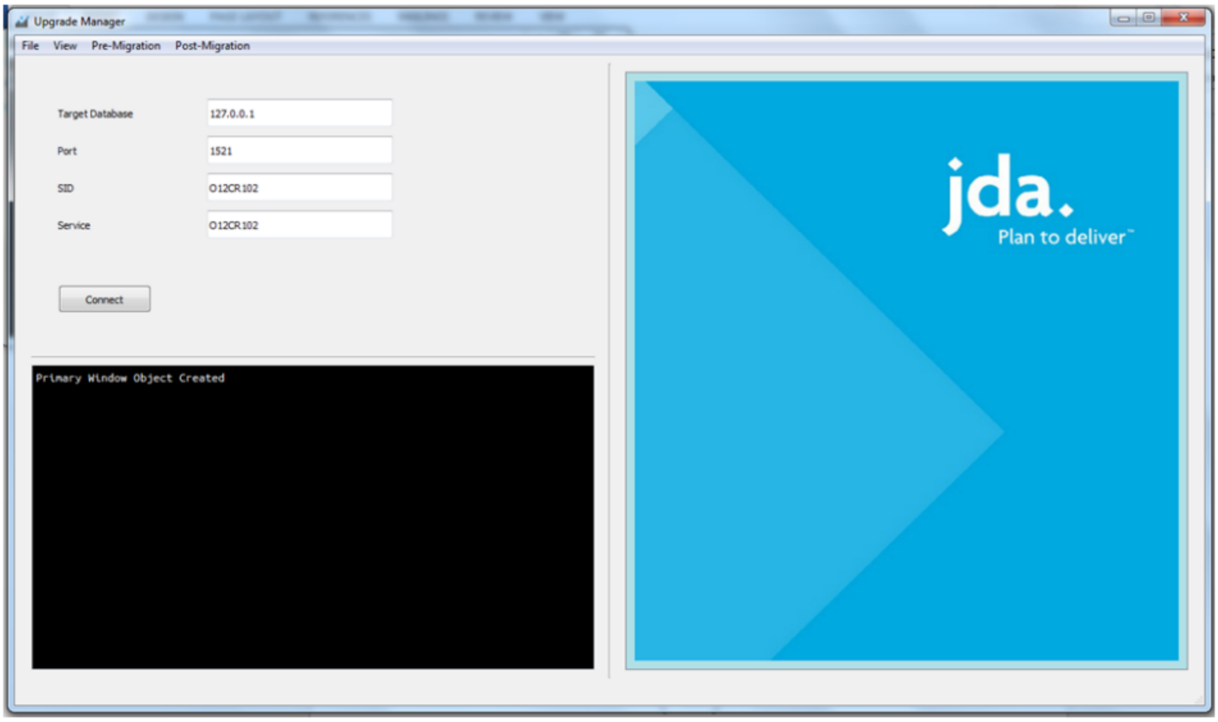
UPGRADE MANAGER

Every once a while changes are made to the JDA application which bring in new features and performance improvements. But a number of the JDA customers are already running older versions of the JDA application and sooner or later with advancing technology those older versions might become obsolete. This is why there is a strong demand to upgrade the existing customers to the newer version so that they can enjoy the new features and improved performance.

Procedures already exist to upgrade any JDA application to a newer version but the problem is that the procedures are quite lengthy. If it were a single customer there would no problem since we'd have to do it once but there are like hundreds of customers and every once a while newer versions of JDA application might come in and we'd have to do everything over again.

This part of the project attempts to solve the issues described above. The result is a unified interface which would do everything described above automagically without much user intervention. Also it has the capability to auto fix the error if it is told how to deal with a specific type of error it can solve all similar errors on its own. So in short, we'd have to fix errors for a single customer and every similar error for all others will be fixed.

A view of the opening screen is shown below:



Target Database 127.0.0.1

Port 1521

SID O12CR102

Service O12CR102

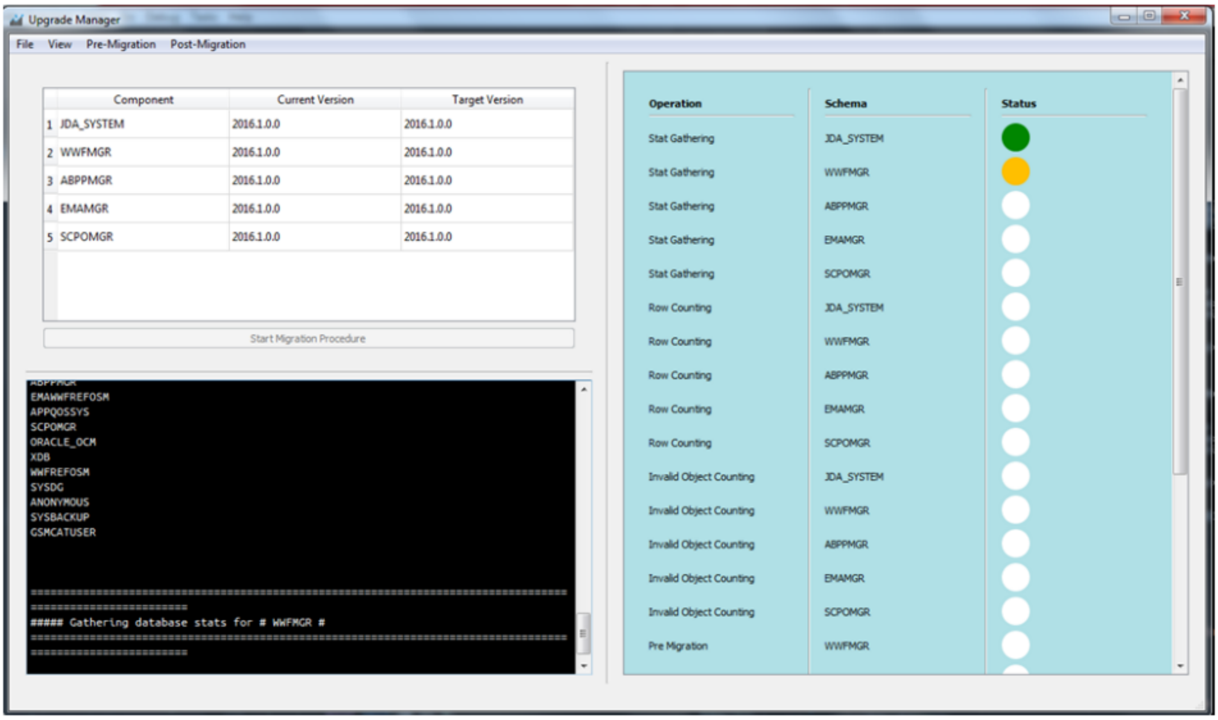
```
Primary Window Object Created
```

The unified layout has a bunch of components. On the top, we have a menu bar which has a couple of options for the user to customize and monitor the migration. On top-left is the action screen, all visual inputs from the user will be taken over there, as of now the action screen is set to have a connection screen which has a button that will try to connection to various schemas required for performing migration.

On the bottom-left, we have the console output screen where most of the raw output can be seen. On the right, we have a display screen which is meant for processed output for the user to see various activities during migration in a user friendly manner.

After successfully performing the connection test, it will prepare the plan of migration i.e. the tasks that need to be performed in order and a button will wait for the user to press that will initiate the migration procedure.

Every task is accompanied by a status label which is basically an image. The image is white initially when nothing has been done. It turns amber while in progress, green when finished and red if something went wrong. If it is red, it is going to wait for the user to fix something or ignore it so it can move on and do the rest of the tasks. A view while migration is in progress is shown below:



	Component	Current Version	Target Version
1	JDA_SYSTEM	2016.1.0.0	2016.1.0.0
2	WWFMGR	2016.1.0.0	2016.1.0.0
3	ABPPMGR	2016.1.0.0	2016.1.0.0
4	EMAMGR	2016.1.0.0	2016.1.0.0
5	SCPOMGR	2016.1.0.0	2016.1.0.0

```
ABPPMGR
EMAWWFREFOSM
APPQOSSYS
SCPOMGR
ORACLE_OCM
XDB
WWFREFOSM
SYSDG
ANONYMOUS
SYSBACKUP
GSMCATUSER



==========================================================================================
========================
##### Gathering database stats for # WWFMGR #
==========================================================================================
========================
```

Operation	Schema	Status
Stat Gathering	JDA_SYSTEM	🟢
Stat Gathering	WWFMGR	🟠
Stat Gathering	ABPPMGR	⚪
Stat Gathering	EMAMGR	⚪
Stat Gathering	SCPOMGR	⚪
Row Counting	JDA_SYSTEM	⚪
Row Counting	WWFMGR	⚪
Row Counting	ABPPMGR	⚪
Row Counting	EMAMGR	⚪
Row Counting	SCPOMGR	⚪
Invalid Object Counting	JDA_SYSTEM	⚪
Invalid Object Counting	WWFMGR	⚪
Invalid Object Counting	ABPPMGR	⚪
Invalid Object Counting	EMAMGR	⚪
Invalid Object Counting	SCPOMGR	⚪
Pre Migration	WWFMGR	⚪

FUTURE PLANNING

This is the biggest project among the others and scope is quite vast. Migration involves many components and it will take quite some time to include all of them but on the other hand reward is huge in terms of reducing the time and manpower it takes for migration activities.

CONCLUSION

Man power is a valuable resource and should be utilized only wherever needed. Doing repetitive tasks which can be automated should never be done manually. The Project’s aim is to eradicate this man power wastage by automating procedures wherever possible and hence in turn benefit the organization ethically and financially. The sub-projects aren’t yet being approved for usage and will take quite some time for them to get mature so that the community may accept them. Aside from the actual implementation, I had a lot of fun in the learning aspects as further developments like these will be done in much faster time by my hands.