Centralized Test Automation System for Web Applications

Buddhika Bhageashwara Alwis

E131041006

Faculty of Information Technology

University of Moratuwa

March 2017

Centralized Test Automation System for Web Applications

Buddhika Bhageashwara Alwis

E131041006

Faculty of Information Technology

University of Moratuwa

March 2017

Declaration

We declare that this thesis is our own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student Signature of Student

Date:

Supervised by

Name of Supervisor Signature of Supervisor

Date:

Acknowledgement

Specially I would like to take this as an opportunity to express my deep sense of gratitude and profound feeling of admiration to my supervisor, Mr. ransara wijethunga for guiding me through-out this project. Additionally I would like to thank my officemate Mr kasun vimukthi disanayaka for giving me a great support to successfully carryout the project. I also wish to gratefully thank all other faculty members including coordinator of Bachelor of Information Technology (BIT) and Project Examination board of University of moratuwa for their guidance provided for successfully carrying out the project. Again I wish to forward my thanks to my batch mates for providing various support and encouragement to complete the project. Finally I send my thanks from deep in my heart to my parents for encouraging me do the degree and supporting to carry out the entire program.

Abstract

In Sri Lanka most of the IT companies are used decentralized free test automation tools to test GUI functionalities of web applications. Decentralized attribute is caused to couple of problems for companies who are willing to test GUI functionalities of web applications. Test case management is poor within these free tools. There is no interconnection among testing tools. Because of that web application testing is error prone and management cost is high even though tool is free. Free tools are limit the team work activities. To solve these kind of problems, IT companies are used few solutions. As a frequent solution, create test cases and execute by a free tool and reporting and analysing part are done by using distinct tool. Data gather is done by through emails. Checking the progress of the test case execution is not possible time to time until the report is submitted. As well as require occupy additional time for reporting of each test run .during test case execution if something happen to local machine that tester have to wait until the machine is repaired because tester created test cases are stored in the local machine. Productivity, efficiency, effectiveness of testers will be challenged because of decentralized testing tools. Testers can’t show maximum performance. Sometime testers may could cheat on test case execution. Testers can mark test case execution as done or can be marked mistakenly.

Centralized characteristic is capable of reduce the most of the inconvenient and problems in current test automation process. Pointing on advantages of centralized characteristic this Centralized Test Automation System is come up with Centralized characteristic. System is expected couple of inputs from the user to process and give execution details. Respectively system is required to create a product by an authorized user. Created product is needed to be attached to a test plan. That test plan should have test cases. Inside one of test case should have test case steps. After user inputs are entered correctly, user will be able to execute test cases according to way of test cases are lined up. After execution was finished system will be generated reports. And also can be shared the details of the execution simply without an extra effort.

According to implementation concerns here used most of the technologies are free and open source software and technologies. MySQL community server for centralized database to be installed and for development of testing tool is used J2EE platform.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Centralized Test Automation System

1.1 Introduction

Web applications are leading application category in many domains such as ecommerce, medicine, finance, people management and education. There are many free and commercial testing tools available in the market for automating testing of web applications. One of the major drawback of available tools is free tools are not provided features to work as a team. Main reason for this limitation is that all the tester's tools are placed in theirs computers locally and there is no interconnection among tools. In some cases, the test report is created using Microsoft Excel. In addition to above analysing past test case results become cumbersome with traditional local testing system lack of centralized test and test result management system open an opportunity to testers to cheat on test case execution and result reporting. Although the tester may state that the test case was executed, it might not be executed against the new release of the software. Another emerging aspect in the web application quality assurance is test automation. When automating web application testing, testers have to go through coding activities. They have to know a programing language like java, ruby, PHP, Perl or python. These automation data is also saved in tester's local computer most of the times. When the local machine hard disk stops performing due to defect or damage, then the tester's whole data will lost.

As mentioned in above paragraphs, lot of problems are occurring because testing tool is located in the tester's machine. This project aims to introduce a centralized web application testing and test automation system to overcome above mentioned problems. In a centralized system, managers and supervisors don't need to wait until email or the excel sheet comes to them. They can check it through the system. Testers can't cheat on testing and reporting. Testers are not needed to do code because the proposed tool will provide a GUI for the tester to automate the test. In any defect or failures at the local machine, no data will be lost since data is stored and backed up in the centralized server.

1.2 Background & Motivation

When the automating the web application testing is not centralized or doesn’t have a GUI [5], testers will be faced to few problems. Even though test cases are executed analyse test results are not an easy task for users. Complication of the report analyse procedure (checking all emails and excel sheets), managers and supervisors who in charge, may miss important details and it can be misleading when taking decisions. Most of the time manual calculations are needed to be carried out by counting passed and failed test cases of test runs. Generally reporting part is done by excel sheets and share using emails. However tester are good at testing, testers may do mistakes or testers may cheat on test case execution. One of significant problem in test automating is testers have to familiar with a programing language. It will be a nightmare to the testers. Testers have to use loops, if conditions, check errors etc. And also testers will be made mistakes when create scripts for automate web application testing. In a case of a failure in a local machine, tester’s data will be lost or can’t continue the process until machine is fixed.

In Sri Lankan IT companies are not used centralized test automation systems due to few reasons. First reason is there is no proper free tools for test automations. Whatever there are exist free tools that tools are restricted mostly requires functionalities from user. That features are come with the commercial version of that tools or just only for a limited time period.

1.3 Aim and Objectives

1.3.1 Aim

Centralized test automation system for web applications.

1.3.2 Objectives

- Give a GUI to proposed tool to avoid the coding approaches for test automations.
- Give the centralized characteristic to proposed tool for overcome difficulties of non-centralized test automation systems for web applications.

- Add better test case management features by letting to create, delete, update and execute test cases.

1.4 Proposed Solution

Centralized Test Automation System (CTAS) for Web Application is about to check graphical user interface (GUI) functionalities of web applications and generate report for test cases executions and share the execution details among authorized users. Basically system is consisted with two components.

- Testing Tool (deploy in each of user's machines, testing tools are needed to connect to the Centralized Database)
- Centralized Database (deploy in a common database server, all testing tool are connected to Centralized Database)

Test case create and execution can be done connecting these two key components together. System is followed the two tier architecture. Business logic and presentation are together and database is separated from others.

Following procedure have to follow by the user to act on the system,

At the very first time user should give database connection settings for connect to the centralized database. And also should have an account for the user to pass the authentication process. After user is authorized to login to the system, user can create Test Cases within a Test Plan and also have to create Test Case Steps within a Test Case. After completing the Test Case creation, user can execute that created Test Cases. Test Case execute results can be viewed from reporting module according to user access levels.

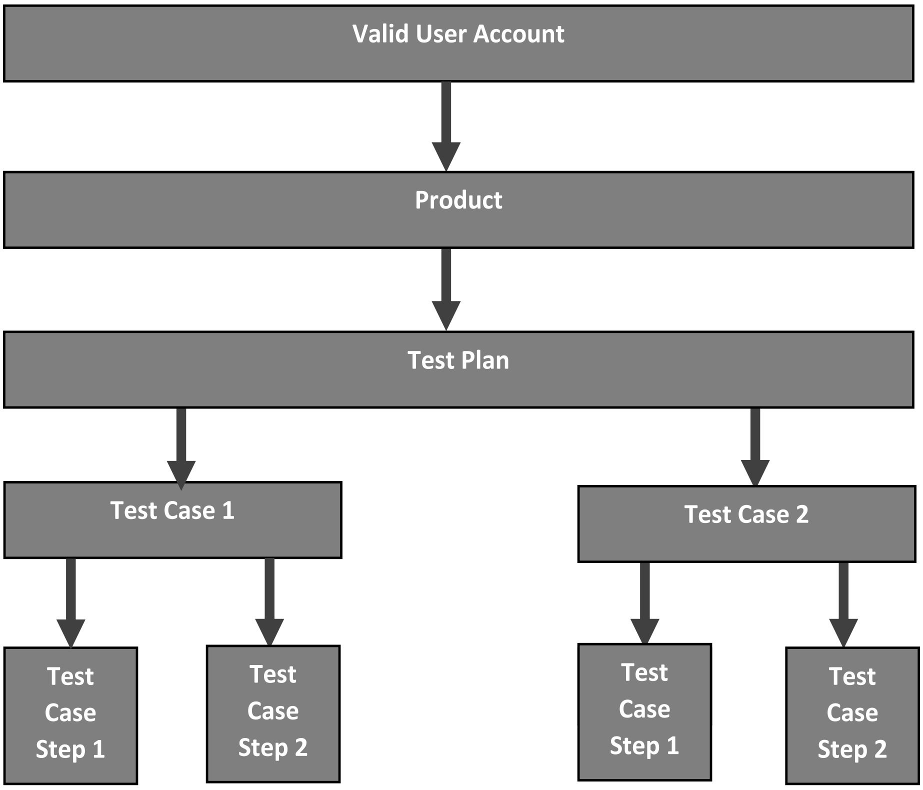


Figure 1: Test Case Creation Flow

1.5 Summery

From above chapter discussed about the problems of decentralized systems and way of address the problem. Proposed system is the solution for decentralized system's problems. Proposed system capable enough to reduce the difficulties of decentralize systems.

Chapter 2 is based on literature survey. It is about the similar system studies for which are mentioned problems and solutions in chapter 1.

Chapter 2

Status of Decentralized Test Automation Tools

2.1 Introduction

In previous chapter was given a clear idea about the background and the motivation of the project, proposed solution for the decentralized systems, aims and objectives of project. Purpose of this chapter is expressed a full description about background information of the project and state about similar other's approaches to solve decentralized problem and including other problems.

2.2 Proposed Centralized System vs Decentralized Systems

Centralized System	Decentralized System
Store test case execution results and test case details in a centralized database.	Create and store test case execution results in local machine.
Test case execution results available for every authorized users. No need to request test case execution results.	Execution details should have to send by an email.
Can't cheat on test case execution. If a test case wasn't run, system shows it as a not executed test case.	Can marked as a executed test case even though it is not executed
Can check progress of the test case execution anytime	Until report is submitted by tester, supervisor and manager cant check the progress
If need to hand over the work of a tester to another, process is easier.	Work hand over process will be complicated
If a failure was occurred in local machine that is not effected on the test case execution. User can start the test from another machine.	Test case data will be lost or have to wait until repair the machine

Table 1 Proposed Centralized System vs Decentralized Systems

2.3 Review of Other Similar Systems

Currently IT companies in sri lanka such as hSenidmobile, MIT, Virtusa are used selenium web driver [4] for GUI test automations. This web driver doesn’t have a GUI and testers have to create test cases by creating scripts. In earlier time that companies were used selenium IDE and selenium RC for test automation.

2.4 Limitations of the similar systems found

- Centralized characteristic is not in similar systems[8]
- Doesn’t have a GUI.[5]
- Rare to find free tools without limitations for basic testing[6]

2.5 Summery

Widely use test automation tools are not going to fulfil all requirements neither centralized systems nor decentralized system. But on the whole, centralized systems are more suitable for test case executions. This chapter is included limitation details of decentralized systems and also how that problems are handled by centralized systems.

Next chapter is reserved for talk about the technologies are used for the system development and identify what are the reasons to chosen that technologies

Chapter 3

Centralized Test Automation System (CTAS) Moreover Decentralized Systems

3.1 Introduction

Chapter 3 about to describe nature of the system users, technologies and tools are used for develop the CTAS with reasons to adopt that technologies for development process.

3.2 Nature of the Users in the System

Especially CTAS is designed for manner structured companies. From this system companies can manage their test automation process as expected way. If a company may not have a hierarchy, some of user levels are not required to use. As an example if the company haven't managers, that company isn't need to use level 2 type users. Level 1 and Level 3 type users are enough to carrying out the test automation. Single users also can use this system to do test automation. In that kind of a situation Level 1 type users are enough for test automation.

3.3 Technologies and Tools

Most of the developers are required free and open source software and technologies. If use free and open source softwares and technologies there is no need to worry about the legal barriers. Because of this concern most of the selected technologies and tools are free and open source for this development also.

3.3.1 Technologies

3.3.1.1 Object orient programing (OOP)

Increase the code readability and reusability and also easy to do modifications. (Appendix A)

3.3.1.2 Linux Over Windows Environment

Linux Operating System is chosen for development environment by considering. Security (virus attacks are less) matters and RAM, CPU usages are lesser than the windows OS.

3.3.1.3 Version Control System (VCS)

Able to keep the project in cloud to take the project from anywhere. Storing versions, restoring previous versions, backup are the other few benefits from a VCS. Used Bitbucket as VCS. (Appendix B)

3.3.1.4 MD5 Encryption for Encrypt the Password

Helps to prevent from MySQL injection.

3.3.1.5 J2EE (Java 2 Platform Enterprise Edition)

Need to integrate better security and error handling features for the system and also to development process. That concerns are covered by the J2EE platform. Other thing is platform independent behaviour in J2EE. That will be easier for development process.

3.3.1.6 Maven

Maven is used to build the project easily without dependency missing issues [3](Appendix C). Can make a war file of the project and using the war file can deploy the system after correctly configured with database.

3.3.1.7 MySQL Community Server

Most popular database in the world. Free and open source. [2]. Support on J2EE.

3.3.1.8 Bootstrap

Use full for create user friendly interfaces. Web elements are arranged automatically according to screen sizes. No need an extra effort.

3.3.1.9 DAO (Data Access Object)

Can access database data systematically. Easy to manage when needed a modification. Others also can easily understand the code even though who are not familiar with the project. (Appendix D)

3.3.1.10 MVC

Separated view, controller and model are useful for untangle the files within the project. Allow a better approach for modifications. If a user interface problem came just need to look at the view only. No need to deal with controller or model. Can find the problem file quickly.

3.3.1.11 Scaler Vector Graphic (SVG) with D3 (Data Driven Documnt)

Line charts, pie charts in the reporting module are drawn by the SVG[13] and use the D3 library[12] to dynamic interactive data visualization in web browser.

3.3.1.12 Log4j

Used as logging framework to manage the login. [11]

3.3.2 Tools

3.3.2.1 Apache Tomcat

War file is can deploy on server and help full for run the java files meanwhile developing the system.

3.3.2.2 Eclipse (Neon version)

Eclipse also one of most popular free and open source IDE. Easy to code. Lot of development support plugins and features are included with in it.

3.4 Summery

These technologies which described in this chapter are helped to develop the system with effectively, efficiency, quicker and also with high security. End product will be an optimal product.

Chapter 4

Adopt the Centralized Property for Decentralized Test Automation Systems

4.1 Introduction

technologies are explained in previous chapter for the system development with reasons to choose that technologies and this chapter is for explain the basic process of the system, system functionalities, user behaviours and input, output, process of the system.

4.2 Background and benefits of the solution and Technologies to Solve the Problem

With centralized property, test automation systems are worthy more than the decentralized systems. Centralized test automation systems are reduced the risk of data loses, increase the security of data, make the data shareable, allocate huge amount of storage capacity for data. Availability of the data also another advantage of centralized systems. Which users are authorized to access the database, that user are capable to access data anytime, anywhere when they want by passing the authentication process. Data integrity is maintain highly in centralized systems. There is no possibility to repeat same data again and again. That problems are handled by the system. All users are connecting to a centralized database. That mean every records of users are checked for each and every time when users are trying to store records within the database. When same record is needed among users of system, no need to redundant the data for share data. Instead of that users can share the records without having redundant data. This behaviour is ensure by letting do the concurrent transections. Database level changes are can apply easily. One change mean all user are affected from change. No need to change all databases like in decentralized systems.

Following technologies are used to develop the system

- Design – visual paradigm, star UML
- Server Side Programing Language – java
- Client Side Language – JavaScript
- GUI Design – bootstrap , Data Driven Documents (D3)
- Dependency management – Maven
- Logging - Log4j
- Database Management System- MySQL
- Layout – bootstrap, HTML, CSS
- Hosting – Apache Tomcat

4.3 User Involvement in the System

Proposed system is accessed by three type of users. Each and every type users have different responsibilities and act differently within the system. System user types are,

- Level 1, Level 2, Level 3

User	Functions and Responsibilities	Access levels
Level 1 user (Admin)	Create and update users Created users are allocated to test plans according to managers, supervisors requests with proper access levels for create, delete, update test plans, test case and test case steps. Create test plans Create products	Unlimited within the system

	Deactivate user’s status change to the active status Activate user’s status change to the deactivate status Take care of the centralized database Reset password. Access levels can be reduce as well as increase for test plans, test cases and test case steps	
Level 2 user (Manager, Supervisor)	Analyse test case execution results Order to allocate users for test plans. Track the test case execution process of authorized test plans	If a level 1 user give access, can create test cases and test case steps. Can delete, update, execute too. But only within the authorized test plans. Can view test plan execution results and logs and also allocated users for the test plan. But only within the authorized test plans.

		* Can’t create test plans and products.
Level 3 user (tester)	Execute test cases. Create test cases and test steps	If a level 1 user give, access, can create test cases and test case steps. Can delete, update, execute too. But only within the authorized test plans. Can view test plan execution results for the test plan. But only within the authorized test plans. * Can’t create test plans and products.

Table 2 User Function and Responsibilities with Access Levels

User access levels can be vary from user to user whatever users are in same level user type except level 1 users. As an example two level 2 or two level 3 users can have different access levels. It will be depend on the access levels given by the level 1 users whatever the users are in same level. Dynamic access levels will be maintained for level 2 and level 3 users. That access level are for create delete update execute test cases, test case steps and also for update, delete test plans. Level 1 users have static access levels by default. That access levels can’t be changed. Report module access levels are static. Please refer Table 2

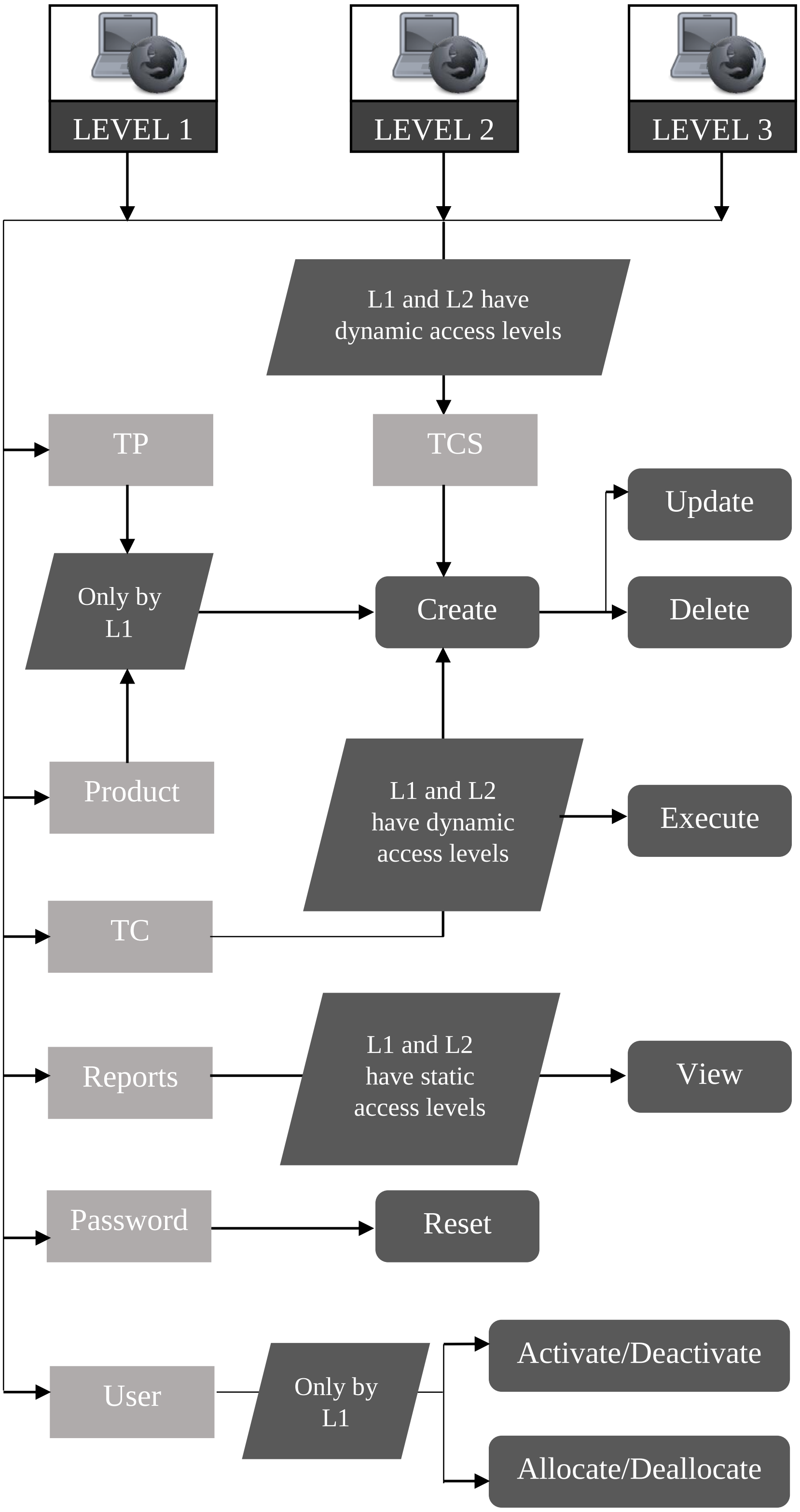


Figure 2 Access Level Matrix

4.4 Input, Outputs and Processes of the System

4.4.1 Input

- Create TC
- Create TP
- Create TCS
- Create users
- Create products
- allocate users to TP s

4.4.2 Processers

- Delete, Update TC
- Delete, Update TP
- Delete, Update TCS
- Update users
- Delete, Update products
- Execute TC
- Allocate, deallocate, and reallocate users from TP
- Deactivate/Activate users

4.4.3 Out put

- TC execution results
- TP, TC, TCS logs
- User profile details
- Allocate details

4.5 Summery

Within this chapter was discussed about the summary of the system process, user involvement on system and also about the input, outputs and processes for the system.

Next chapter will reserve for talk about the analysis and design of the proposed system.

Chapter 5

Analysis and Design Approach for the CTAS

5.1 Introduction

This chapter will define how the system is going to implement. Objects that are defined in the analysis phase will refine in this chapter. Data base design is illustrated. Use case, sequence, activity diagrams and other relevant UML diagrams are also included in purpose of design.

As one of most important topic in analysis and design chapter is top level architecture of the system. That topic also going to explain in this chapter.

5.2 Analysis Approach for CTAS

5.2.1 Alternate Solutions and Feasibility

This system is designed to run in a common platform where in any given time, an authorized person could be able to login to the system's centralized database when the user have a testing tool. In future this system will be able to deal with the WAN connections from the LAN connections. Web based approach was chosen for future expanding of the systems. Therefore J2EE based technology is the most flexible way of implementing the system.

5.2.2 Functional Requirements

Functional requirements are the tasks which should be performed to complete an action of a system. Use case diagrams in the design chapter will describe further more about functional requirements of the system.

Functional requirements of each modules are listed below,

- **Admin module**
  Create, Update users
  Active/Deactivate users
  Create test plans
  Create/delete/update products
  Allocate / deallocate / Reallocate to test plan

- **User Profile Module**
  No functional requirements

- **3T Management Module**
  Delete, Update test plans
  Create, Delete, Update test cases
  Create, Delete, update test case steps
  Execute test cases

- **Reporting module**
  View test case executed results

5.2.3 Non-Functional Requirement

Non-functional requirements of a system are critical requirements where overall system may be fall in to a failure when one or more of them were not fulfilled. These types of functions are not directly deal with the functions of the system and non-functional requirements will gain the maximum system utility and optimal performance from the system.

5.2.3.1 Accessibility

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is available to as many people as possible. Accessibility can be viewed as the ability to access and benefit from some system or entity. Since this is centralized, accessibility will no longer be a problem to the users. Additionally functions such as password reset will be attached to admin module to increase the accessibility of the system.

5.2.3.2 Accuracy

Accuracy defines the correctness of the information. Since this system involved with the test case automation details of the system should be accurate where it can avoid unnecessary conflicts.

5.2.3.3 Security

Security of the centralized test automation system is a critical non-functional requirement since it holds many sensitive web application information and security details of web application. The security. Password protection can be implemented to enhance the security of the system. User can only login to system by providing username and password.

In order to ensure the system security, proposed Centralized Test Automation System will adhere to the following. Password policy is applicable for all users who are accessing the system directly.

- Password length needs to be at least 8 alphanumeric characters long.
- Password must contain at least 1 numeric character.
- Passwords shall be encrypted and stored. MD5 encoding scheme will be used.
- User account shall be disabled after 3 consecutive invalid attempts.
- Password reset shall be available for a Level 1 user

5.2.3.4 User Friendly

Since this is a system very close to the managers, supervisors, testers, the system should be very user friendly to increase the usage of the system. company need to increase the usage of the system where they can have updated test case execution information of web applications what going to test using this system.

5.3 Design Approach for the CTAS

5.3.1 Top Level Architecture

The top level architecture of CTAS is based on fat client-server architecture. Business logic and client both are placed in one machine. Centralized database is separated from business logic and client located tire.it is existed in a separated tire.

Business logic and client are represented by the testing tool and the browser. DBMS tire is represented by the centralized database.

User should request information through a web browser using testing tool and request sent to the centralized database server. Centralized database server is provided details according to the request. After details are received to testing tool, testing tool is started to test the web application’s GUI functionalities.

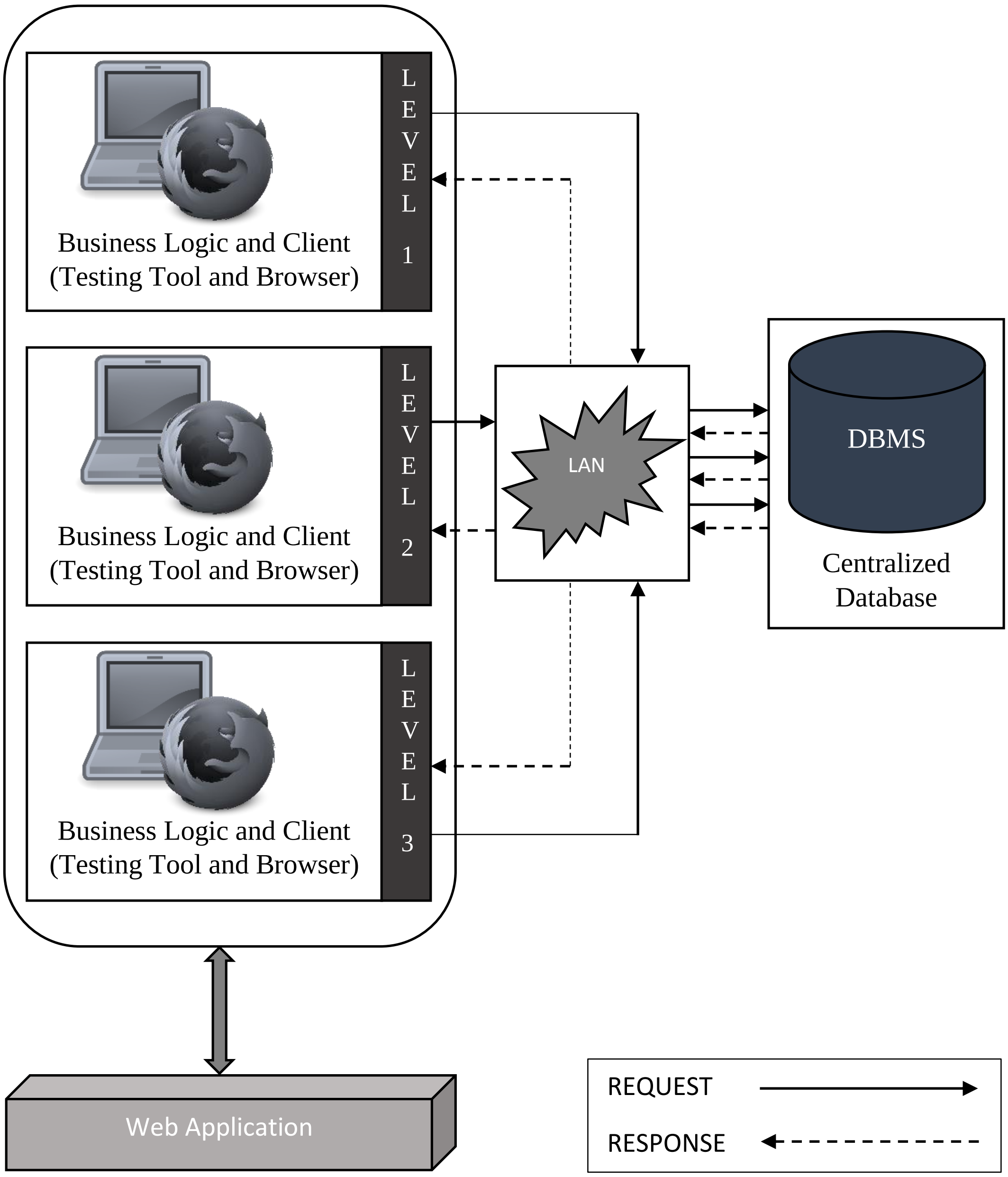

Business Logic and Client (Testing Tool and Browser)
LEVEL 1
Business Logic and Client (Testing Tool and Browser)
LEVEL 2
Business Logic and Client (Testing Tool and Browser)
LEVEL 3
LAN
DBMS
Centralized Database
Web Application
REQUEST
RESPONSE

Figure 3: Top Level Architecture

5.3.2 UML Diagrams

Since this project is developing through J2EE. It supports object oriented approach very well. Therefore Unified Modelling Language is the most suitable design technique to use. Use case diagrams, Activity diagrams, Sequence diagrams and class diagrams are most suitable for design a system like this. Where use case diagrams shows the functions of the system where as sequence diagrams shows interactions of objects along with the time and class diagrams shows the intra object relationships.

Activity diagrams are about to graphical presentation of workflows of step by step activities and actions. EER diagram is to represent the database entities and attribute with relevant relationships what are needed to CTAS.

5.3.2.1 Use Case Diagram

This Use case diagram for entire CTAS. See Appendix E for use case descriptions

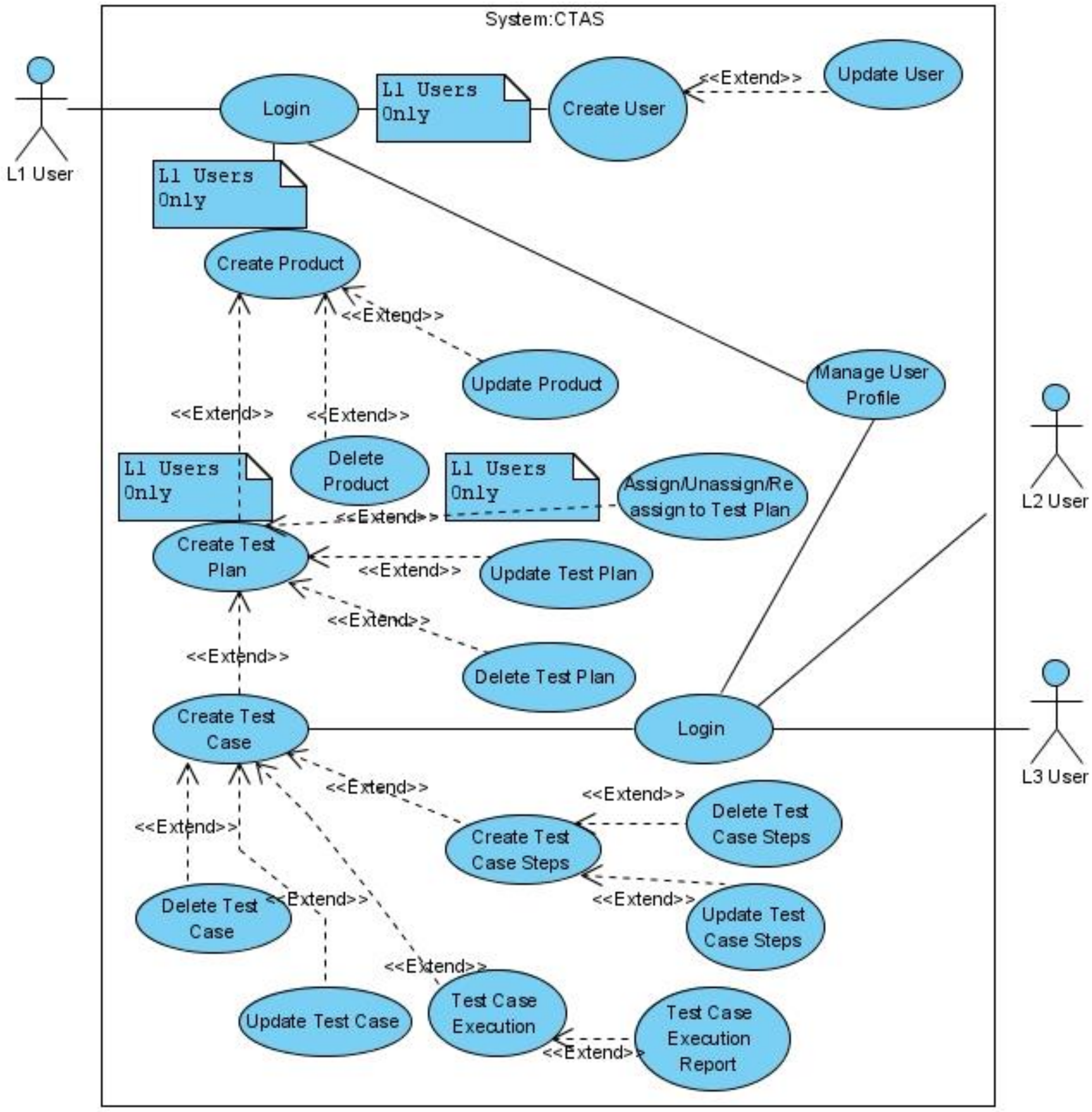


Figure 4: Use Case Diagram for Entire CTAS System

5.3.2.2 Activity Diagram

These activity diagrams are a flow chart to represent the flow form one activity to another activity of centralized system.

5.3.2.2.1 User Login

Below activity diagram for understand the login flow of the system. Users will be blocked by the system after three time attempt by using a wrong password. If database settings were not set, system is provided a page to set the database settings.

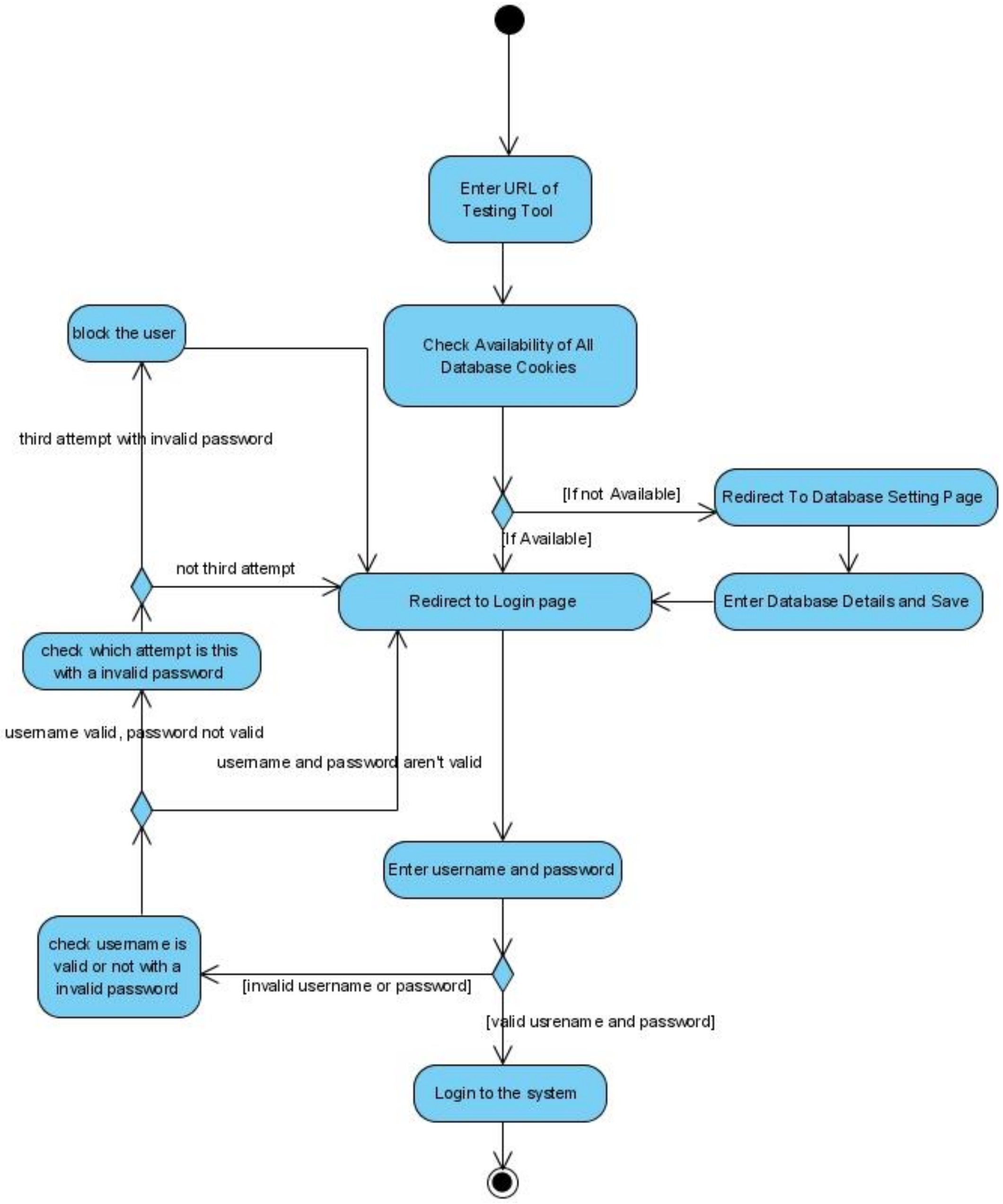

Enter URL of Testing Tool
Check Availability of All Database Cookies
block the user
third attempt with invalid password
[If not Available]
Redirect To Database Setting Page
[If Available]
not third attempt
Redirect to Login page
Enter Database Details and Save
check which attempt is this with a invalid password
username valid, password not valid
username and password aren't valid
Enter username and password
check usernam e is valid or not with a invalid password
[invalid username or password]
[valid usrenam e and password]
Login to the system

Figure 5 Activity Diagram for User Login

5.3.2.2.2 Activity Diagram for 3T Management

3T Management activity diagram for explain activities from create test cases to execute test cases according to user access levels.

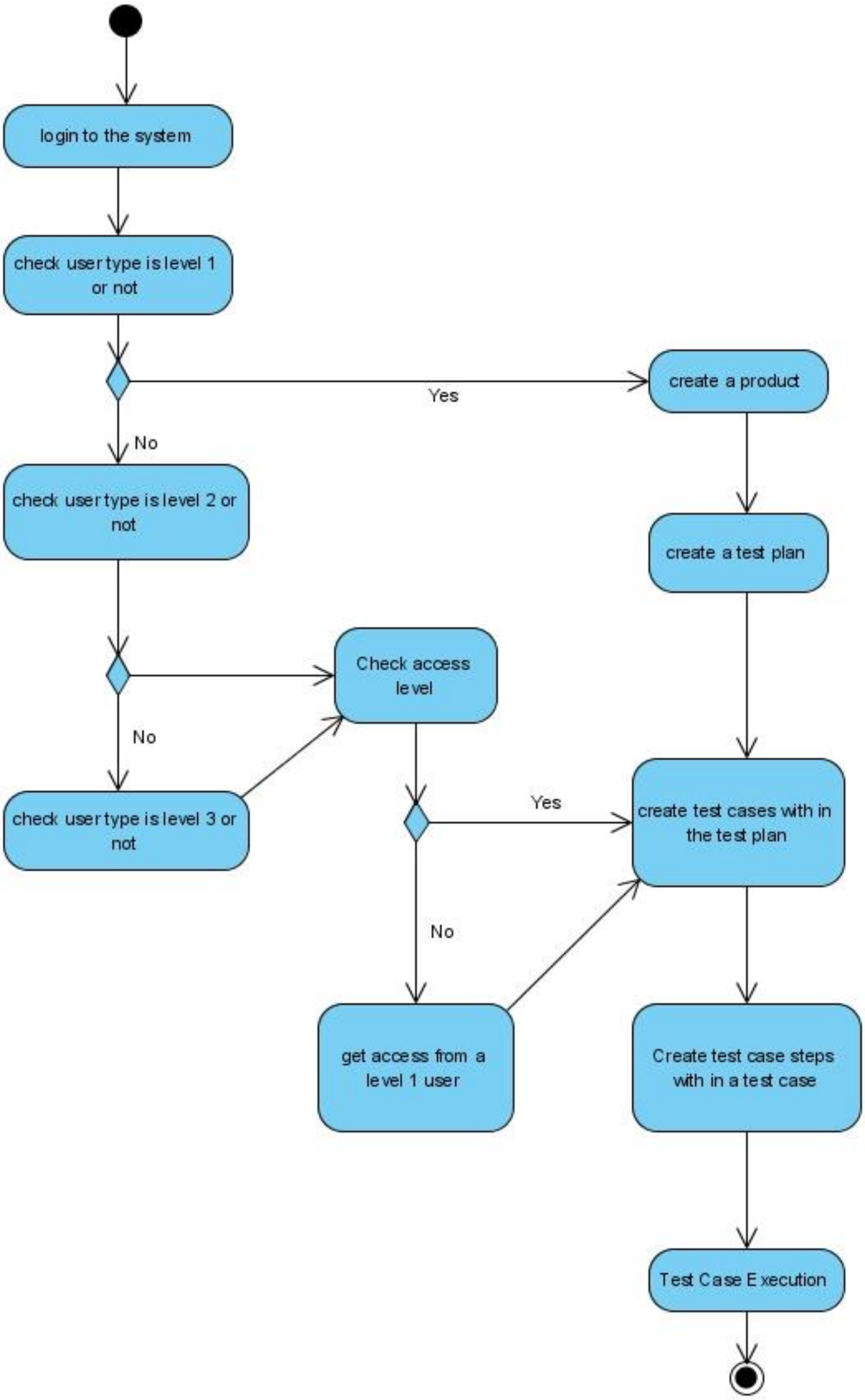

login to the system
check user type is level 1 or not
Yes
create a product
No
check user type is level 2 or not
create a test plan
Check access level
No
check user type is level 3 or not
Yes
create test cases with in the test plan
No
get access from a level 1 user
Create test case steps with in a test case
Test Case Execution

Figure 6 Activity Diagram for 3T Management

5.3.2.3 Sequence Diagrams

5.3.2.3.1 Sequence Diagram for Login

Login sequence diagram represents the interaction between classes and objects according to time for login.

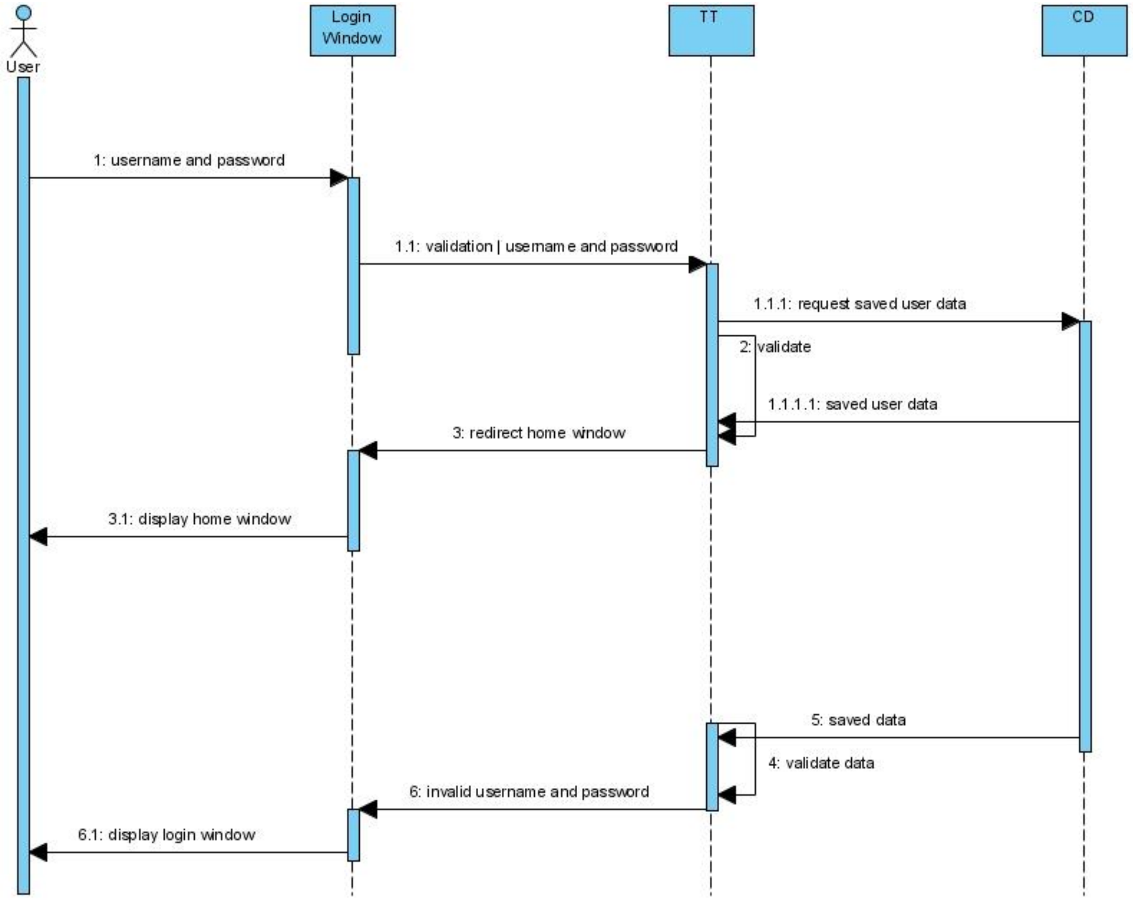


Figure 7: Sequence Diagram for Login

5.3.2.3.2 Sequence diagram for 3T Management

3T Management sequence diagram represents the interaction between classes and objects according to time for 3T Management.

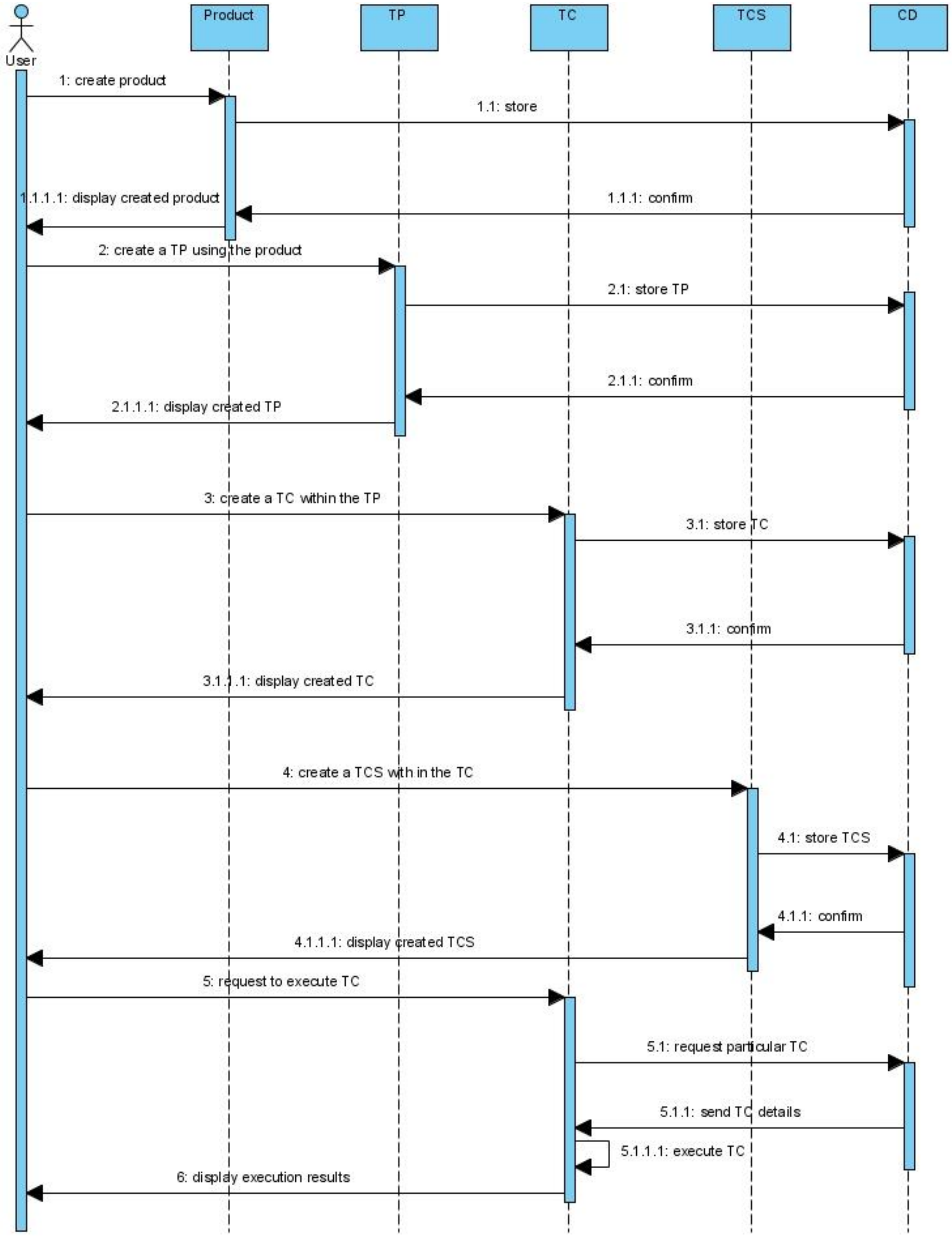

User
Product
TP
TC
TCS
CD
1: create product
1.1: store
1.1.1: confirm
1.1.1.1: display created product
2: create a TP using the product
2.1: store TP
2.1.1: confirm
2.1.1.1: display created TP
3: create a TC within the TP
3.1: store TC
3.1.1: confirm
3.1.1.1: display created TC
4: create a TCS with in the TC
4.1: store TCS
4.1.1: confirm
4.1.1.1: display created TCS
5: request to execute TC
5.1: request particular TC
5.1.1: send TC details
5.1.1.1: execute TC
6: display execution results

Figure 8: Sequence Diagram for 3T Management

5.3.2.4 EER Diagram

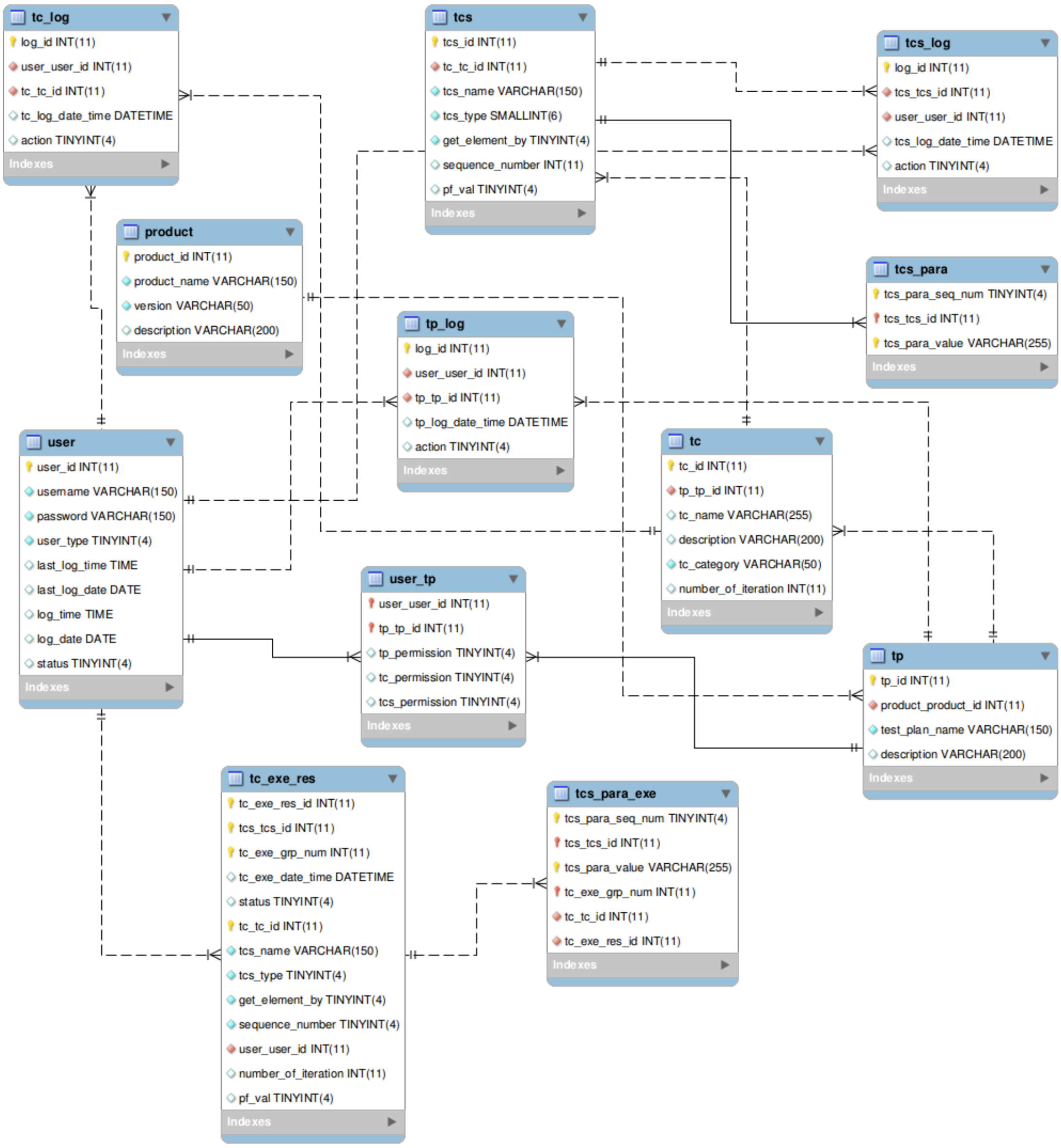


Figure 9 EER Diagram

Below assumptions were taken when draw the EER diagram

- A User has many test plans and a test plan has many users.
- A user has none or many test plan logs. A particular test plan log belong to one user only. Test plan log is become unique by the log_id.
- A user has none or many test case logs. A particular test case log belong to one user only. Test case log is became unique by the log_id.
- A user has none or many test case step logs. A particular test case step log belong to one user only. A test case step log is became unique by the log_id.
- A user execute none or many test case steps. But a test case execution result belong to one user only. A test case step result is became unique by tc_exe_res_id. Need to have execution records about the test case steps also. Keep test case step execution time and date are helped to take time durations among one steps to other.
- A test plan has one product and a product can has none or many test plans. Without a product, there can’t be existed a test plan.
- A test plan has none or many test cases. A test case has one test plan only. A test case is became unique by the tc_id.
- A test case has none or many test case steps. A test case step has one test case only. A test case step is became unique by the tcs_id.
- A test case step has at least one or many parameters and a parameter has one test case step. A parameter is became unique by the tcs_id and tcs_para_values.
- A test case execution result has none or many tc execution para. A tc execution para has one tc execution result.

5.3.2.5 Class Diagram

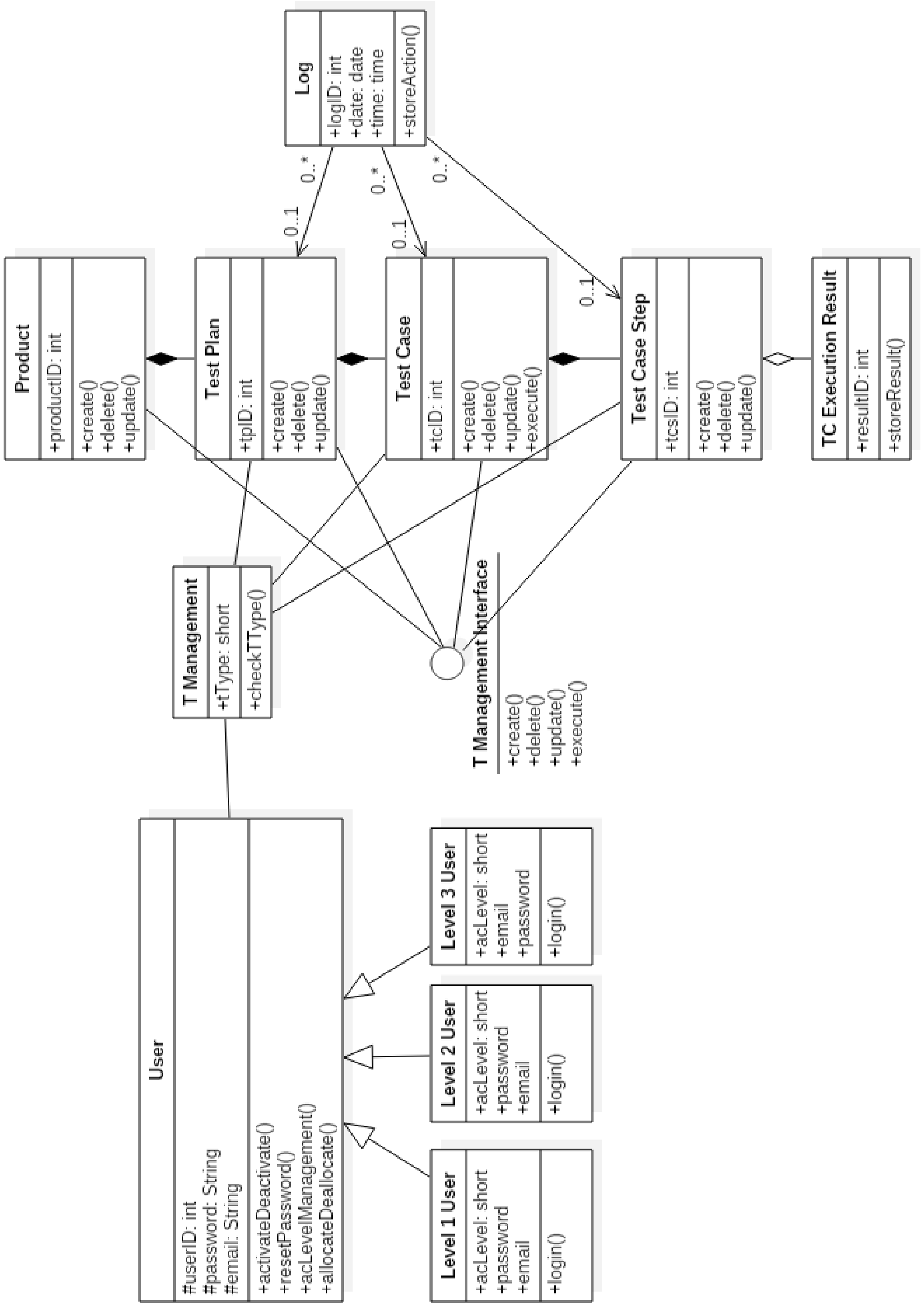


Figure 10 Class Diagram

- Level 1, 2, 3 users has user class inheritances. (acLevel, password, email)
- User involve with test plan, test case, and test case step classes through the T management class.
- Without a product class there is not a test plan object. Because of that composition relationship is maintained between the test plan class and product class.
- Without a test plan class there is not a test case object .because of that composition relationship is maintained between the test plan class and test case class.
- Without a test case class there is not a test case step object .because of that composition relationship is maintained between the test case class and test case step class.
- Test plan, test case, test case step are connected with T Management interface for get the update, delete, create, execute abstract methods. Execute method implementation going on in test case class only.
- Log class has directional association relationship with test plan, test case and test case step classes.

5.4 Summery

Design and analysis chapter was given a brief explanation about the way of design and analysis of centralized test automation systems by using various diagrams such as use case diagrams, activity diagram, sequence diagrams, class diagrams, and EER diagram. In next chapter is about the implementation approach using these design approach. That chapter rich with the actual coding for the system.

Chapter 6

Implementation of Centralized Test Automation System

6.1 Introduction

Implementation is the process of converting the system specification into a working system and this is the place where any system gets its real life shape. Implementation processes transform the specification to an executable program, which are, most of the time interleaved. This system has implemented in order to fulfil the requirements identified so far in the system analysis phase and to satisfy all the main stake holders of the system.

Modern well known and highly using language and appropriate tools were chosen in the process of development and coding. The codes were written and arranged in a human readable and understandable format, supporting details with comments, hoping to produce software that will be maintainable in further future developments.

6.2 Implementation Environment

Some unique features and specified needs of the system were taken in to consideration when selecting an implementation Environment. When selecting the development software and tools, most of them were free and open source which will not cause much trouble when getting the copyrights of the system. Some of these technologies are targeted at a specific application domain (e.g., testing tool design and implementation); others focus on a technology domain (e.g., object-oriented systems).

6.3 Major Code Segments

TC Executer: - TC execution part is handle by this code segment. According to the TCS type, appropriate action is taken. Set the execution start date time and pass fail determination are other responsibilities of this code segment.

```
public short executeTC(TManagementTCExecutionModel tMTCEM, WebDriver driver) {
        try { //Date date = new Date();
            tMTCEM.setDateTime(new Date());
            //1 Open //2 Click //3 Input //4 Match //5 Wait //6 Clear //7 Close //8 isRedirected
            if (tMTCEM.gettCSType() == 1) {
                //1 Open   //1 para --> URL
                driver.get(tMTCEM.getParaValuesArrayList().get(0));
                tMTCEM.setStatus((short) 1); // pass
                return 1; // pass
            } else if (tMTCEM.gettCSType() == 2) {
                //2 Click   //  1 para --> element name by type (css, xpath, class, id etc)
                WebElementGen.getWebElement(tMTCEM, driver).click();
                tMTCEM.setStatus((short) 1); // pass
                return 1; // pass
            } else if (tMTCEM.gettCSType() == 3) {
                //3 Input //  1 para --> element name by type (css, xpath, class, id etc)   //  2 para --> input
text
                WebElement elemnt = WebElementGen.getWebElement(tMTCEM, driver);
                elemnt.sendKeys((tMTCEM.getParaValuesArrayList().get(1)));
                tMTCEM.setStatus((short) 1); // pass
                return 1; // pass
            } else if (tMTCEM.gettCSType() == 4) {
                //4 Match  //  1 para --> element name by type (css, xpath, class, id etc)  //  2 para -->
Expected text
```

Figure 11 TC Executer

Web Element Provider: - When executing a TC, system is required a web element. If user was defined xpath, class or id type elements, system should provide the relevant web element to TC execution. That service is provided by this section.

```
if (tMTCEM.getGetElementBy() == 1) {
            // By Class
            WebElement element =
driver.findElement(By.className(tMTCEM.getParaValuesArrayList().get(0)));
            //List<WebElement>  element =
driver.findElements(By.className(tMTCEM.getParaValuesArrayList().get(0)));
            return element;
        } else if (tMTCEM.getGetElementBy() == 2) {
            // By ID
            //System.out.println("inside ement ID"+driver.getCurrentUrl());
            WebElement element =
driver.findElement(By.id(tMTCEM.getParaValuesArrayList().get(0)));
            //WebElement element =
driver.findElements(By.id(tMTCEM.getParaValuesArrayList().get(0)));
            return element;
        } else if (tMTCEM.getGetElementBy() == 3) {
            // By XPath
            WebElement element =
driver.findElement(By.xpath((tMTCEM.getParaValuesArrayList().get(0))));
            return element;
        } else if (tMTCEM.getGetElementBy() == 4) {
            // By Tag Name
            WebElement element =
driver.findElement(By.tagName(tMTCEM.getParaValuesArrayList().get(0)));
            return element;
        }
```

Figure 12 Web Element Provider

Please refer the appendix F for other code segments

6.4 Database Implementation

Database implementation is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. (Appendix G)

Database implementation improves the functionalities of the proposed system. It represents attributes of the each class in the system. Data base is normalized and naming method is carried out to enhance the manipulation of the system.

6.5 User Interface Design

The main source of interaction between a user and the system are the user interfaces where the user is allowed to work with the system without bothering about the back-end logics which are neatly hidden from the user. User interface of a web based system is more user friendly because of the flexible styles and navigation through dynamic hyperlinks.

Since the centralized test automation system adapted MVC framework, where the system divide in to 4 main modules namely

- Admin module
- User Profile Module
- 3T Management Module(Test plan, Test case, Test case steps Management Module)
- Reporting module

Furthermore login module and database setting module are included.

Then users can easily navigate through system module to get their job done without navigate unwanted pages.

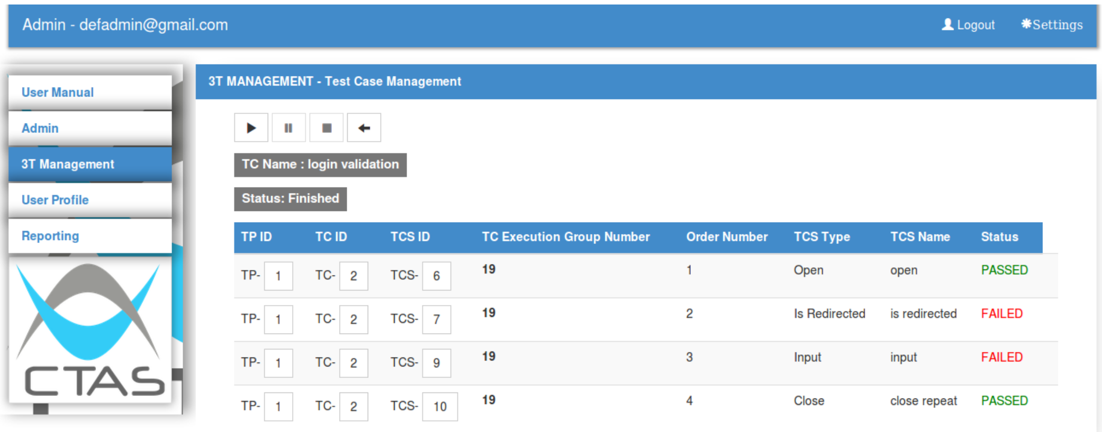
The main advantage of MVC is that we separate presentation from logic, keeping everything much cleaner. Views make it easy for developers to present a flexible and consistent interface, and to change it if the user requests without worrying about the internal business rules.

Some main user interface design rules were followed while designing the system. As specified below.

- Provide clear and consistent navigation along with easy access.
- Logical grouping of tasks where a collection of links are provided for each module.
- Display visual elements regarding their priority, by following special organization which allows for quick location and interpretation.
- Build interfaces for the concepts from the system’s domain.
- Choose a font and the text-size which is clear and readable.
- Display meaningful error messages and proper instructions when the user encounters any errors.
- Be flexible, auto select options as well as let the users select options.

6.5.1 User Interfaces

TC Execution Page in TC Management Page: - responsible for show the live TC Execution details as well as execute TC.



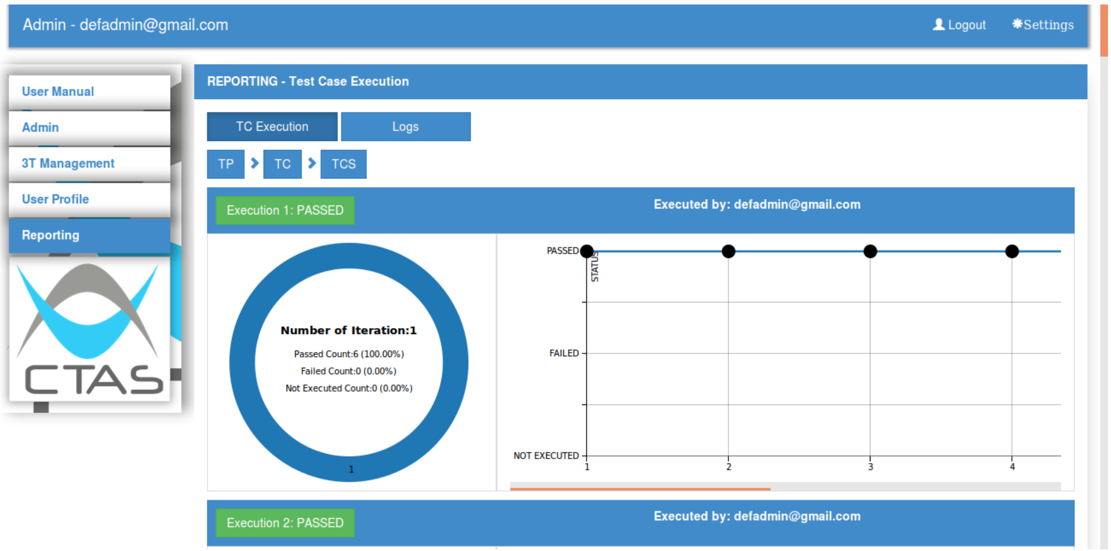
TC Name : login validation

Status: Finished

TP ID	TC ID	TCS ID	TC Execution Group Number	Order Number	TCS Type	TCS Name	Status
TP- 1	TC- 2	TCS- 6	19	1	Open	open	PASSED
TP- 1	TC- 2	TCS- 7	19	2	Is Redirected	is redirected	FAILED
TP- 1	TC- 2	TCS- 9	19	3	Input	input	FAILED
TP- 1	TC- 2	TCS- 10	19	4	Close	close repeat	PASSED

Figure 13 TC Execution Page in TC Management Page

Detailed TC Execution Results in Reporting Page: - pie charts and line charts are within this page to show complicated details in user understandable way.



Execution 1: PASSED

Executed by: defadmin@gmail.com

Number of Iteration:1

Passed Count:6 (100.00%)

Failed Count:0 (0.00%)

Not Executed Count:0 (0.00%)

Execution 2: PASSED

Executed by: defadmin@gmail.com

Figure 14 Detailed TC Execution Results in Reporting Page

Please refer Appendix H for other user interfaces

6.6 Hardware Requirement

Following assumptions have been considered when deriving the hardware requirements for the system.

Client Side Computer (for testing tool)

- CPU: intel core i5 or higher
- RAM: 8GB

Centralized data base server

For 25 Concurrent Users

- CPU: Quad 2GHz+ CPU
- RAM: 4GB
- Minimum database space: 100GB

6.7 Software Requirement

Following assumptions have been considered when deriving the software requirements for the system.

- Operating Systems – Windows (10, 8.1, 8) or LINUX.
- Web Browser - Firefox, Google Chrome, Internet Explorer.
- Web Server – J2EE application servers like apache tomcat, glassfish
- Database Server – MySQL (Version 5.6 or later).

6.8 Performance Requirements

Database server can withstand even though many number of user requests desired service. LAN connectivity will be caused for better performance of the system.

6.9 Summery

This chapter emphasis on discussing about the technologies that is used for the implementation of CTAS. Also shows some user interface designs of the system. Further, database implementation was explained and demonstrated. The next chapter is a discussion on summary about the report and a brief description of further work.

Chapter 7

Evaluation and Testing of CTAS

7.1 Introduction

Software testing is a critical element of software quality assurance that represents the ultimate review of specification, design, and code generation. The testing is comprised of Verification and Validation methods. Verification – refers to the techniques of finding out whether the built software meets all the customer requirements and Validation- refers to the techniques that ensure the software meets all the specified functional requirements. The testing process is considered dynamic; given that the process is carried out on an executable function of the system. Testing is a vital process to ensure a perfectly working system with fewer errors is delivered to the customer [9].

7.2 Testing

Testing process was carried out to ensure the requirement are satisfied by the CTAS and identify the corner case issues within the developed system. In addition to that will be able to minimize the CRB(Customer Reported Bugs) after deliver the system to end users. GUI functionality testing, sanity testing, concurrent testing, security testing, installation testing, integration testing, unit testing, database performance test were taken in testing process.

7.2.1 GUI Functionality Testing

This testing was proceeded by manually doing the GUI functionalities, by selenium web driver and as well as by CTAS. Please refer 7.3 section for test case details which were executed under this testing.

7.2.2 Sanity Testing

Before go for the high level testing, this testing was carried out and verified the basic functionalities of the CTAS were functioned as expected way. Please refer 7.3 section for test case details which were executed under this testing.

7.2.3 Concurrent Testing

CTAS is allowing multiple users to work concurrently at the same time while accessing the same centralized database by everybody. Therefore, concurrent testing was given a higher priority and tested vey will in this project. Lost update problems (eg:- if user 1 is executed TC with current TC details and before end the TC execution of the user 1, user 2 is update the TC with new TC details, in this case current TC details will be overwritten by user 2's TC details. This will effect on user 1's execution), incorrect summary problems (eg:- when user 1 is executing a TC and that TC is included 5 TCS, meanwhile the execution is going if another user was deleted two TCS , user 1's execution will get into a trouble because at the begin it was read the TCS count as 5 and end of the TC execution it is counting as 3 TCS. These results are display in the reporting also) are the common and often problems when concurrent ability has a system. These problems were tested from this testing and pass the test. Please refer 7.3 section for test case details which were executed under this testing.

7.2.4 Security Testing

Security is the most one of important fact a system should has. Security measures of the CTAS were taken from this test. Please refer 7.3 section for test case details which were executed under this testing.

7.2.5 Installation Testing

End of the project, product need to be deployed in user's machines. This proses should be checked before introduce the system to users who are willing to use the system. Installation was tested successfully in Linux (Ubuntu) and Windows(8.1) operating systems. Please refer 7.3 section for TC details which were executed under this testing.

7.2.6 Unit Testing

Testing was done during the implementation phase. Tested the expected results from each and every methods.

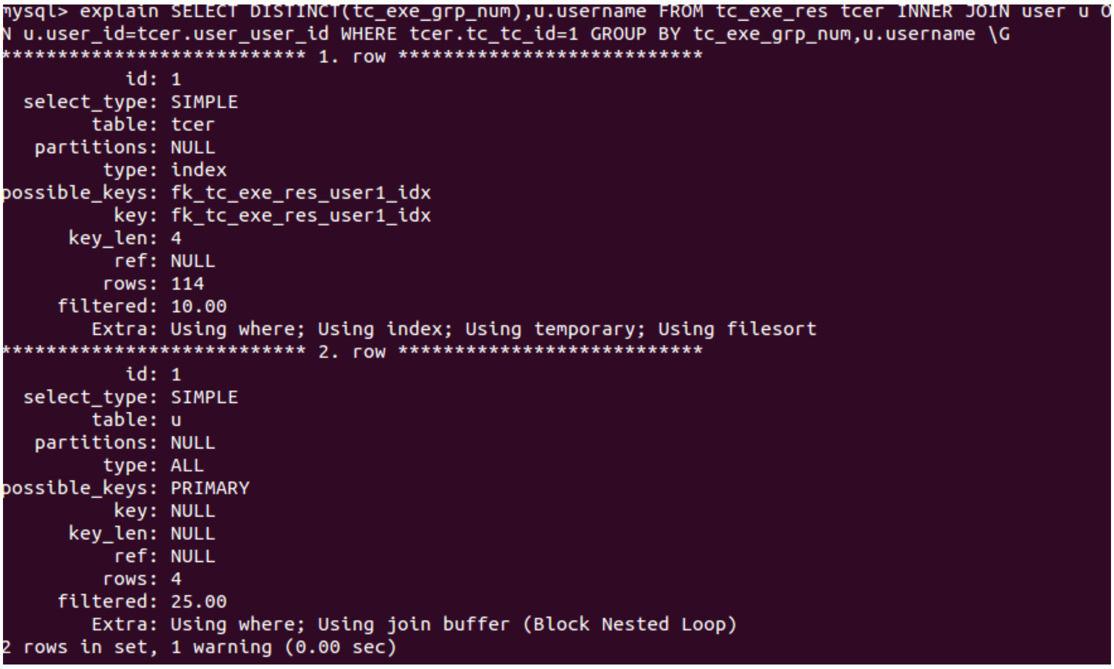
7.2.7 Integration Testing

After end the unit test, this testing was proceeded by taking all modules together.

7.2.8 Database Performance Testing

In database level there are too many transactions need to be handled by the database. Specially read operations are required from database and response time is needed to as soon as less rather than for the write operations. Because of this reason indexes are added without using the default indexes to improve the performance of database. From this testing each and every table performance were examined when the lot of records are existed in tables. MySQL "explain" quarries was used for this testing.

Eg:- explain SELECT DISTINCT(tc_exe_grp_num),u.username FROM tc_exe_res tcer INNER JOIN user u ON u.user_id=tcer.user_user_id WHERE tcer.tc_tc_id=1 GROUP BY tc_exe_grp_num,u.username;



```
mysql> explain SELECT DISTINCT(tc_exe_grp_num),u.username FROM tc_exe_res tcer INNER JOIN user u O
N u.user_id=tcer.user_user_id WHERE tcer.tc_tc_id=1 GROUP BY tc_exe_grp_num,u.username \G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: tcer
   partitions: NULL
         type: index
possible_keys: fk_tc_exe_res_user1_idx
          key: fk_tc_exe_res_user1_idx
      key_len: 4
          ref: NULL
         rows: 114
     filtered: 10.00
        Extra: Using where; Using index; Using temporary; Using filesort
*************************** 2. row ***************************
           id: 1
  select_type: SIMPLE
        table: u
   partitions: NULL
         type: ALL
possible_keys: PRIMARY
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 4
     filtered: 25.00
        Extra: Using where; Using join buffer (Block Nested Loop)
2 rows in set, 1 warning (0.00 sec)
```

Figure 15 Quarry Performance for Default Indexes

7.3 Test Cases

These test cases were created to test the CTAS under GUI functionality testing, sanity testing, concurrent testing, security testing, installation testing.

7.3.1 User Login Test Case

Test Case ID	Test Case Description	Action	Expected Result
1	Check login validation	Enter less than 8 alphanumeric characters password in password field, should not run the MySQL server	Appropriate error message should show, should not connect to database. (Check logs)
		Enter 8 alphanumeric characters password and should not have a numeric character in password field, should not run the MySQL server	Appropriate error message should show, should not connect to database. (Check logs)
		three consecutive invalid attempts within a minute from valid username	User account should block and Appropriate error message should show
		Enter correct username password	Should be able to logged
		Enter invalid format username	Appropriate error message should show, should not connect to

			database. (Check logs)

Table 3 User Login Test Case

7.3.2 Database Setting Test Case

Test Case ID	Test Case Description	Action	Expected Result
2	Check database setting validation	Try to access the login page at very first time	User should redirect to database setting page
		Fill all required fields and click save button	Should redirect to login page
		Update database settings and try to login	Tool need to connect for database according to which are saved new database settings

Table 4 Database Setting Test Case

7.3.3 3T Management Test Case

Test Case ID	Test Case Description	Action	Expected Result
3	Check functionalities of	Create a test case	Test case should create and appear in TC list

	3T management module	Update test case	Should update the test case and new TC details should show
		Delete test case	Should delete the test case and should not show the test case within the test case list
		Create test case step	Test case step should create and appear in TCS list
		Update test case step	Should update the test case step and new TCS details should show
		Delete test case step	Should delete the test case step and should not show the test case step within the test case step list
		Execute test case	Should execute the test case
		Shift two test case steps	Two test case steps are should shifted each other
		Delete a test case when have test case steps within it	should not allow to delete test case, should show the error message instead

		Click pause button in test case execution	Test case execution should pause
		Click resume button in test case execution	Test case execution should resume from stopped step
		Click stop button in test case execution	Test case execution should stop

Table 5 3T Management Test Case

7.3.4 Admin Test Case

Test Case ID	Test Case Description	Action	Expected Result
4	Check functionalities of admin module	Create a user	New user should save in database
		Enter required type password in user create section with appropriate field values for other fields and save	Password should save in database from md5 format
		Update a user from user update section	User should update with new details
		Create a product from create product section	Product should create and save in database
		Delete a product from product delete/update section	Product should delete

		Update a product from product delete/update section	Product should update
		Create a TP from TP create section	TP should create and save in database
		Allocate and deallocate users for TPs from permission management section	User should allocate / deallocate

Table 6 Admin Test Case

7.3.5 User Profile Test Case

Test Case ID	Test Case Description	Action	Expected Result
4	Check details in user profile	Check user details	Other user details should not show, only logged user’s details should show

Table 7 User Profile Test Case

7.3.6 Reporting Test Case

Test Case ID	Test Case Description	Action	Expected Result
5	Check reporting details	Check test case execution details	Pass/Fail status and appropriate details of a test case should show in line chart and in pie chart
		Check log details	Log details of 3T managements should show

Table 8 Reporting Test Case

7.3.7 User Manual Test Case

Test Case ID	Test Case Description	Action	Expected Result
6	Check user manual	Check user manual details are corrected	Should match the manual details with CTAS actual behaviour

Table 9 User Manual Test Case

7.4 Evaluation

10 people were involved with the evaluation and asked questions in evaluation are below.

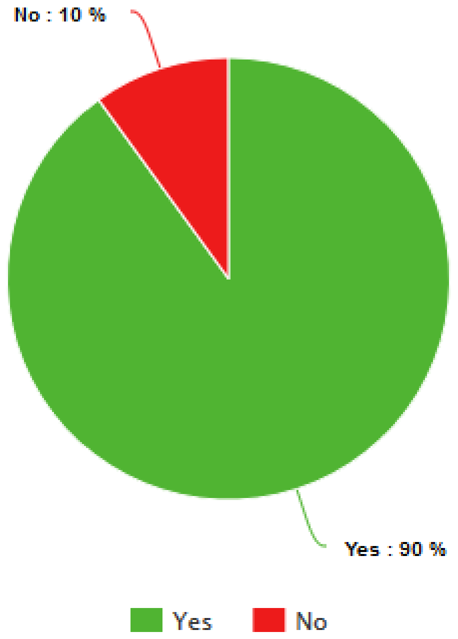
Question 1:- Are you comfortable with CTAS rather than manual test automation


Table 10 Question 1

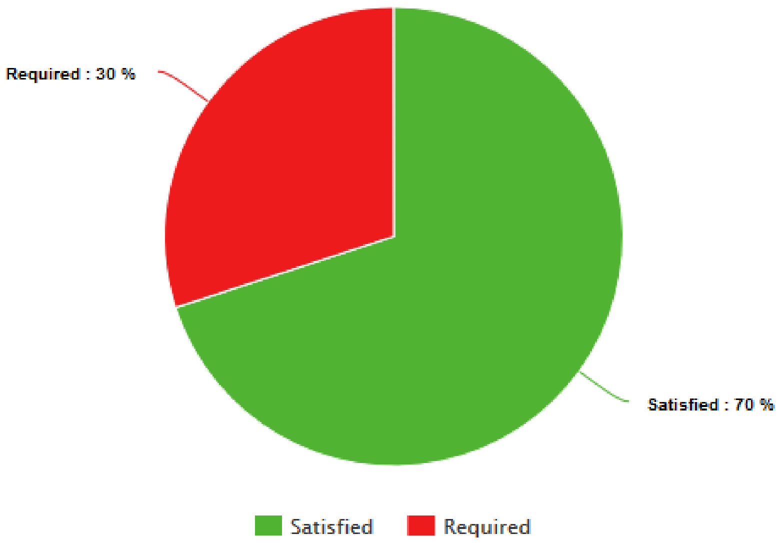
Question 2:- are you required more features to this system or are you satisfied with existed basic features of CTAS?


Table 11 Question 2

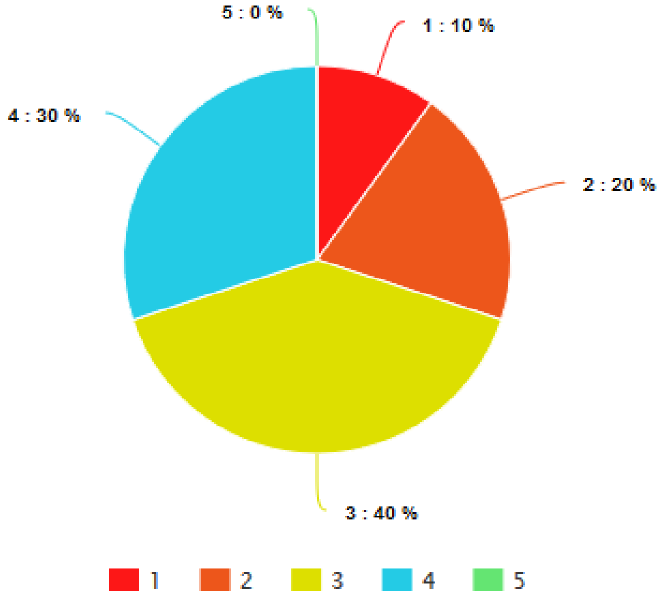
Question 3:- what is the rate number from 1 to 5 you like to give for CTAS as whole?


Table 12 Question 3

Question 4:- Are you accepted the accuracy of the CTAS execution results

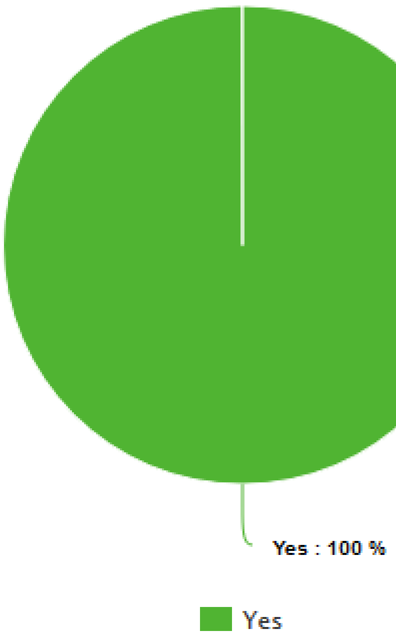


Table 13 Question 4

Question 5:- how are you expressed the user friendliness of the CTAS from 1 to 5 number

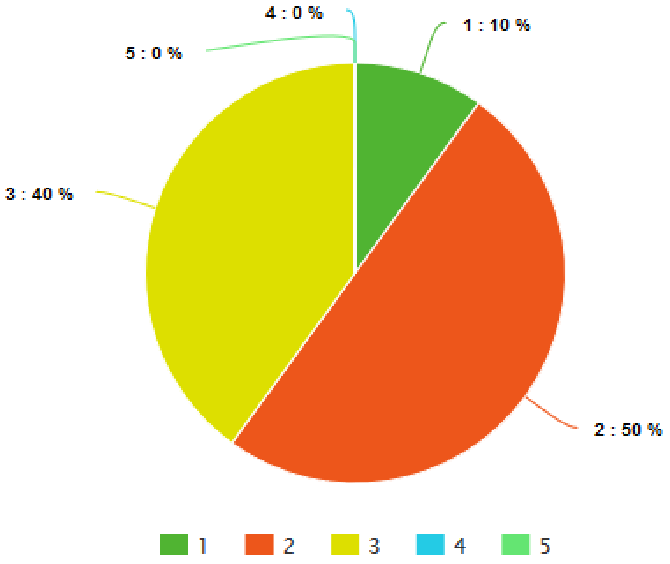


Table 14 Question 5

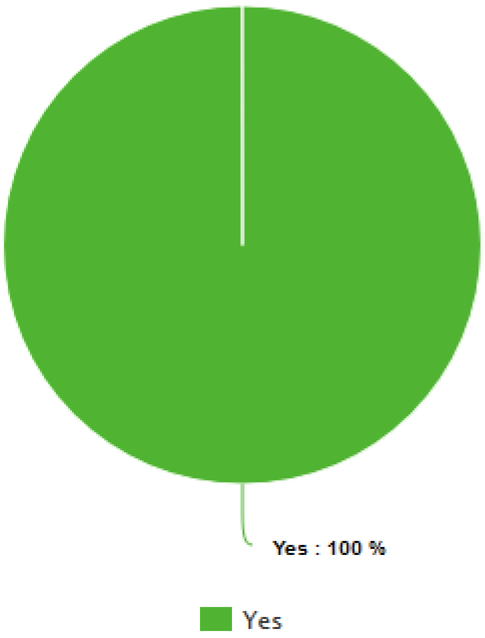
Question 6:- are you agreed with the centralized property for CTAS?


Table 15 Question 6

7.5 Summery

From this chapter was discussed about varied testing were taken to ensure the quality of the product and evaluation process. Final chapter is received to talk conclusion and further works of the CTAS.

Chapter 8

Conclusion and Further Work

8.1 Introduction

This chapter is about to talk overall achievement of objective, what kind of problems were faced during the project start to end, what kind of further works can be taken in future and conclusion about the CTAS.

8.2 Overall Achievement of Objectives

Sri Lankan IT companies such as hSenidmobile, MIT, Virtusa are used tools and libraries to do GUI test automations. These tools and libraries are not included at least one of below features.

- GUI
- Centralized
- Test case management
- Free

Bring these features to CTAS are the objectives of this project and eventually able to bring these features to CTAS.

8.3 Problems Encountered During the Project

Time is the main problem here was encountered. Have to deliver a quality product to users by managing deadlines. Problem was able to control because of the supervisor's guidance and advisers. Reduce the concurrent issues also one of major problem. Have to take actions from database level as well as from the application level to reduce concurrent issues.

8.4 Limitations and Further works

End product of this project is not capable to carry out test automations beyond the GUI test automations. In near future hope to integrate few new test automation types such as load testing, security testing to CTAS. According to development perspective, it is better to use a framework for development like spring, spring boot, hibernate etc. that can be helped to improve the performance, security of the product. After completing load testing the author wish to make this project available to the general public as a free and open source software and improve it further as they wish.

8.5 Conclusion

There are two types of difficulties which are tried to address from this project. They are namely decentralized test automation system's difficulties and manual test automation difficulties. IT companies are seek suitable and possible answers for these difficulties. As one of answer, CTAS can be proposed to Sri Lankan IT companies. CTAS is handy to reduce these difficulties because it has centralized, GUI, test case management, free features.

References

[1] Wikipedia. “Test Automation”.

Internet: https://en.wikipedia.org/wiki/Test_automation, Sep. 08, 2016 [Aug. 08, 2016]

[2] MySQL. “MySQL Community Server”.

Internet: https://dev.mysql.com/downloads/mysql/, Sep. 13, 2016 [Sep. 02, 2016]

[3] Maven. “Maven Life Cycle”.

Internet: https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html, 2016 [Sep. 02, 2016]

[4] Selenium. “Selenium Web Driver”.

Internet: http://docs.seleniumhq.org/docs/03_webdriver.jsp, Sep. 08, 2016 [Aug. 08, 2016]

[5] Wikipedia. “List of GUI Testing Tools”.

Internet: https://en.wikipedia.org/wiki/List_of_GUI_testing_tools Sep. 12, 2016 [Aug. 29, 2016]

[6] Telerik. “A Good Tool, But not a Free Tool”.

Internet: http://www.telerik.com/teststudio, Aug. 31, 2016 [Sep. 10, 2016]

[7] Stepinforum. “Centralized vs Decentralized”.

Internet: http://www.stepinforum.org/2012-03-26-11-42-19/2012-03-27-06-10-31/test-management/293-centralized-vs-decentralized-test-automation, Jan, 2008 [Aug. 08, 2016]

[8] SlideShare. “What is Centralized test automation”.

Internet: http://www.slideshare.net/krishnbharathi/centralized-test-automationframeworkimplementation, Mar. 12, 2012 [Sep. 10, 2016]

[9] [Ian Sommerville, 2007] I.Sommerville, Software Engineering, 8thedition, Addison-Wesley, 2007.

[10] Testopia. “Genaric tool for tracking test cases”.

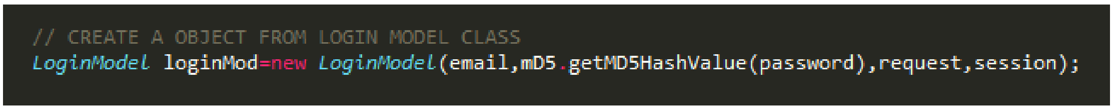
Internet: https://developer.mozilla.org/en-US/docs/Mozilla/Bugzilla/Testopia, Jan. 01, 2017 [Nov. 20, 2016]

[11] Apache. “Log4j - Apache”.
Internet: https://logging.apache.org/log4j/2.x/, Jan. 01, 2017 [Dec. 08, 2016]

[12] D3. “Data Driven Document”.
Internet: https://d3js.org/, Jan. 01, 2017 [Jan. 05, 2017]

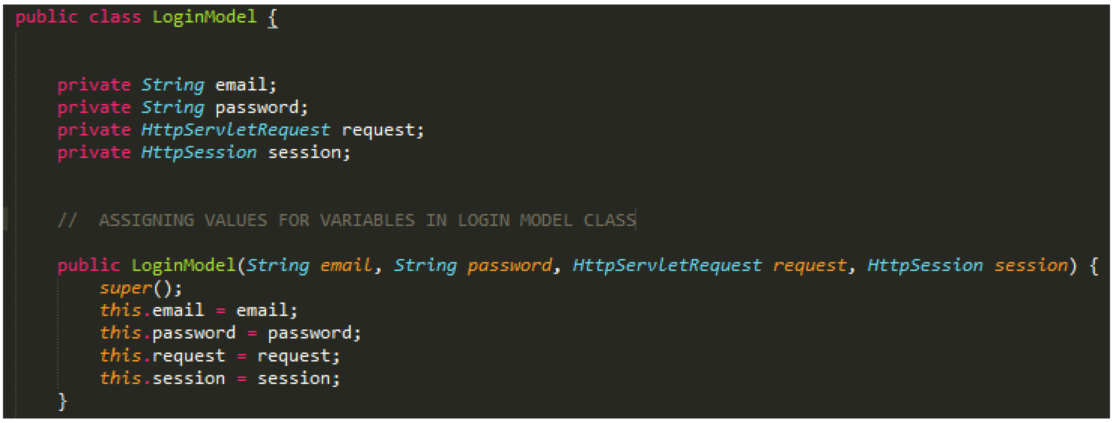
[13] W3Scool. “HTML5 SVG”.
Internet: https://www.w3schools.com/html/html5_svg.asp, Jan. 01, 2017 [Jan. 05, 2017]

Appendix A – Some of OOP Coding in Login Controller and Login Model Classes



```
// CREATE A OBJECT FROM LOGIN MODEL CLASS
LoginModel loginMod=new LoginModel(email,mD5.getMD5HashValue(password),request,session);
```

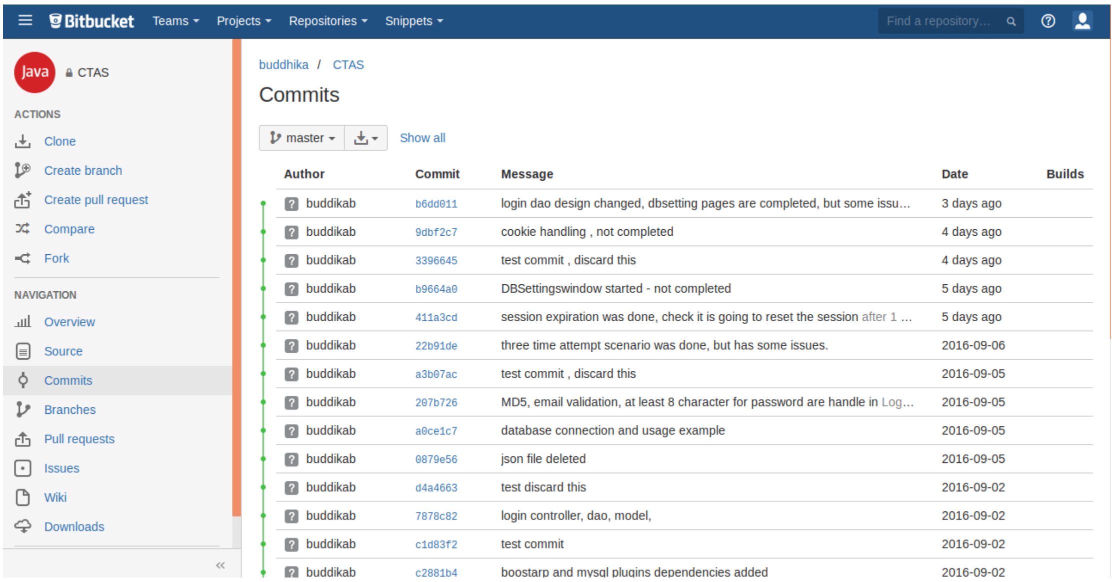
Figure 16 OOP in LoginController.java



```
public class LoginModel {


    private String email;
    private String password;
    private HttpServletRequest request;
    private HttpSession session;


    //  ASSIGNING VALUES FOR VARIABLES IN LOGIN MODEL CLASS

    public LoginModel(String email, String password, HttpServletRequest request, HttpSession session) {
        super();
        this.email = email;
        this.password = password;
        this.request = request;
        this.session = session;
    }
```

Figure 17 OOP in LoginModel.java

Appendix B – Bitbucket as VCS



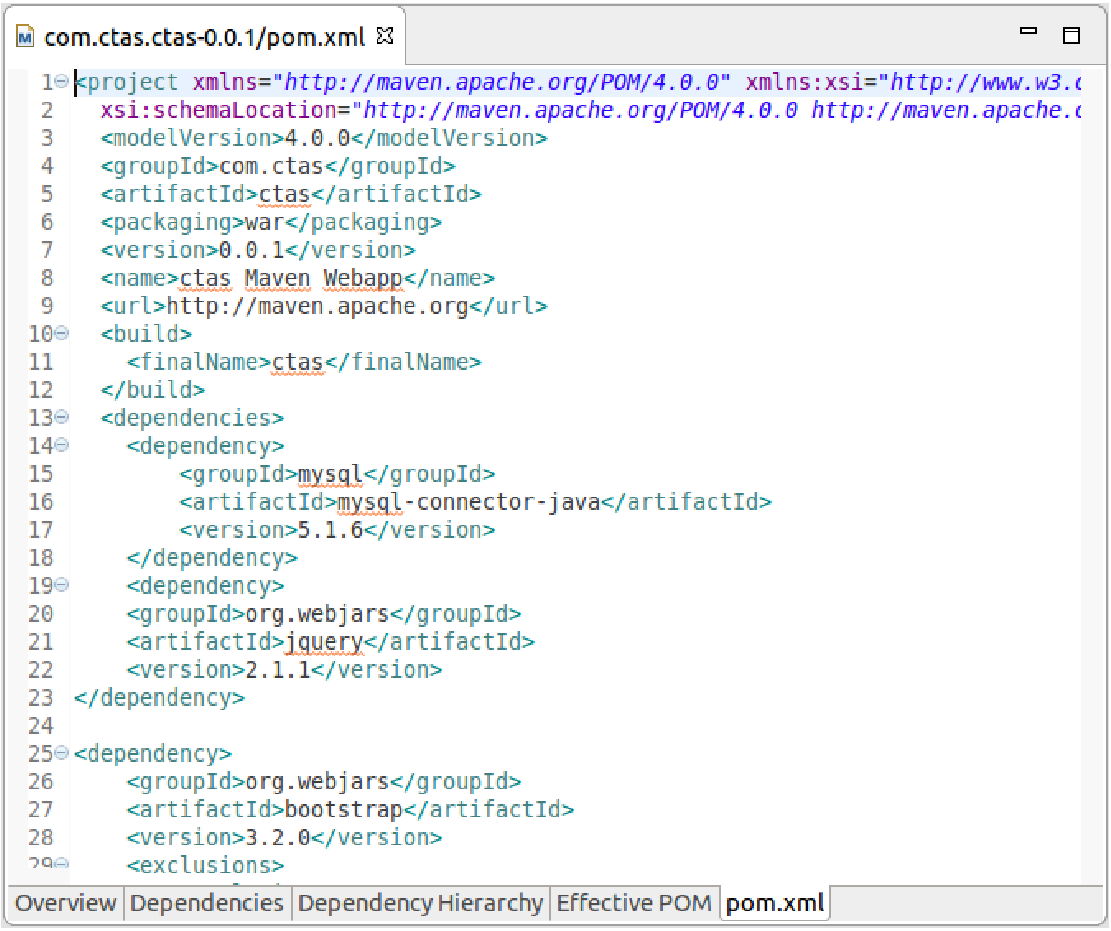
buddhika / CTAS

Commits

Author	Commit	Message	Date	Builds
buddikab	b6dd011	login dao design changed, dbsetting pages are completed, but some issu…	3 days ago	
buddikab	9dbf2c7	cookie handling , not completed	4 days ago	
buddikab	3396645	test commit , discard this	4 days ago	
buddikab	b9664a0	DBSettingswindow started - not completed	5 days ago	
buddikab	411a3cd	session expiration was done, check it is going to reset the session after 1 …	5 days ago	
buddikab	22b91de	three time attempt scenario was done, but has some issues.	2016-09-06	
buddikab	a3b07ac	test commit , discard this	2016-09-05	
buddikab	207b726	MD5, email validation, at least 8 character for password are handle in Log…	2016-09-05	
buddikab	a0ce1c7	database connection and usage example	2016-09-05	
buddikab	0879e56	json file deleted	2016-09-05	
buddikab	d4a4663	test discard this	2016-09-02	
buddikab	7878c82	login controller, dao, model,	2016-09-02	
buddikab	c1d83f2	test commit	2016-09-02	
buddikab	c2881b4	boostarp and mysql plugins dependencies added	2016-09-02	

Figure 18 Bitbucket as VCS

Appendix C – Dependency Management Details in POM File

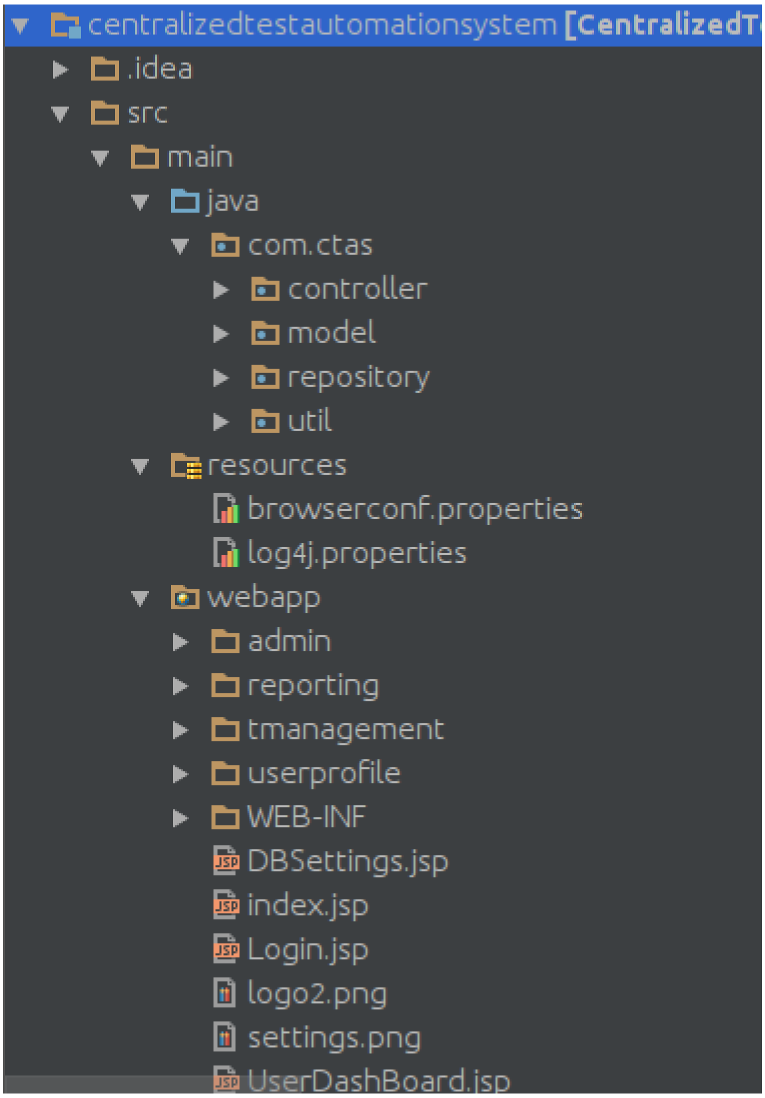


```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.o
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.o
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.ctas</groupId>
  <artifactId>ctas</artifactId>
  <packaging>war</packaging>
  <version>0.0.1</version>
  <name>ctas Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <build>
    <finalName>ctas</finalName>
  </build>
  <dependencies>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>5.1.6</version>
    </dependency>
    <dependency>
    <groupId>org.webjars</groupId>
    <artifactId>jquery</artifactId>
    <version>2.1.1</version>
</dependency>

<dependency>
    <groupId>org.webjars</groupId>
    <artifactId>bootstrap</artifactId>
    <version>3.2.0</version>
    <exclusions>
```

Figure 19 POM.xml file

Appendix D – DAO File Structure

- **Data Access Object Interface** –repository
- **Data Access Object concrete class** – repository → impl
- **Model Object or Value Object** –model



```
▼ centralizedtestautomationsystem [CentralizedT
    ▶ .idea
    ▼ src
        ▼ main
            ▼ java
                ▼ com.ctas
                    ▶ controller
                    ▶ model
                    ▶ repository
                    ▶ util
            ▼ resources
                    browserconf.properties
                    log4j.properties
            ▼ webapp
                ▶ admin
                ▶ reporting
                ▶ tmanagement
                ▶ userprofile
                ▶ WEB-INF
                    DBSettings.jsp
                    index.jsp
                    Login.jsp
                    logo2.png
                    settings.png
                    UserDashBoard.jsp
```

Figure 20 DAO Design Pattern

Appendix E – Use Case Descriptions

Use Case Description for Login Management

Use Case Name	Login Management
Actors	L1 User, L2 User, L3 User
Overview	User can login to the system using username and password
Pre-Condition	User should have an active status account
Flow of Events	1. Open TT through the client web browser 2. Enter username 3. Enter password 4. Click on login button 5. If username and password are correct user can be logged in to the system, if it is not error message will be shown. If user try to login to the system with invalid username or password more than 2 times, user account will be blocked. And also can be reset the password if needed.
Post Condition	User should be able to logged into the system

Table 16: Test Case Description for Login Management

Use Case Description for User Profile Management

Use Case Name	User Profile Management
Actors	L1 User, L2 User, L3 User
Overview	User can see user profile details and can be changed current password
Pre-Condition	User should have an active status account
Flow of Events	1. Login to the system 2. Enter to the user module 3. Check user profile details or if needed to change password, user have to give current password and click on update button
Post Condition	User should be able to check user profile details including allocated TP details and should be able to change the current password

Table 17: Test Case Description for User Profile Management

Use Case Description for User Management

Use Case Name	User Management
Actors	L1 User
Overview	L1 user can create users, update and activate/deactivate user
Pre-Condition	User should be a L1 type user User should log into the system
Flow of Events	1. Enter to admin module 2. Fill required fields with valid values and click on create button for create a user 3. Do modifications and click save button for update (including activate/deactivate user) a user.
Post Condition	New users will be created Existing users will be updated (activate/deactivate user too)

Table 18: Use Case Description for User Management

Use Case Description for Product Management

Use Case Name	Product Management
Actors	L1 User
Overview	L1 user can create, update, delete product
Pre-Condition	User should be a L1 type user User should log into the system
Flow of Events	1. Enter to admin module 2. Fill required fields with valid values and click on create button for create a product 3. Select an existing product and click on delete button 4. Select an existing product and do some modifications. Click on update button
Post Condition	New products will be created Existing products will be updated Existing products will be deleted

Table 19: Use Case Description for Product Management

Use Case Description for TP Management

Use Case Name	TP Management
Actors	L1 User, L2 User, L3 User
Overview	users can be created, updated, deleted TP users can be allocated/unallocated/reallocated users to TP with access levels
Pre-Condition	User should be a L1 type user if want to create TP and allocate/deallocate/reallocate users to TP. User should log into the system
Flow of Events	1. Enter to admin module 2. Fill required fields with valid values and click on create button for create a TP 3. Allocate / Deallocate / Reallocate users for TP with create, delete, update access levels for TC, TCS, TP (not required access levels for create TP). 4. Enter to 3T module 5. Select an existing TP and click on delete button 6. Select an existing TP and do some modifications. Click on update button
Post Condition	New TP will be created Existing TP will be updated Existing TP will be deleted Users will be Allocate / Deallocate / Reallocate to TP

Table 20: Use Case Description for TP Management

Use Case Description for TC Management

Use Case Name	TC Management
Actors	L1 User, L2 User, L3 User
Overview	user can create, update, delete TC
Pre-Condition	User should be a L1 type user Or if user is another type user, that user should have access to TP and for perform delete, update, create actions user need permissions. User should log into the system
Flow of Events	1. Enter to 3T module 2. Select a TP 3. Fill required fields with valid values and click on create button for create a TC 4. Select an existing TC and click on delete button 5. Select an existing TC and do some modifications. Click on update button
Post Condition	New TC will be created Existing TC will be updated Existing TC will be deleted

Table 21: Use Case Diagram for TC Management

Use Case Description for TCS Management

Use Case Name	TCS Management
Actors	L1 User, L2 User, L3 User
Overview	user can create, update, delete TCS
Pre-Condition	User should be a L1 type user Or if user is another type user, that user should have access to TP and for perform delete, update, create actions user need permissions. User should log into the system
Flow of Events	1. Enter to 3T module 2. Select a TP 3. Select a TC 4. Fill required fields with valid values and click on create button for create a TCS 5. Select an existing TCS and click on delete button 6. Select an existing TCS and do some modifications. Click on update button
Post Condition	New TCS will be created Existing TCS will be updated Existing TCS will be deleted

Table 22: Use Case Description for TCS Management

Use Case Description for Reporting

Use Case Name	Reporting
Actors	L1 User, L2 User, L3 User
Overview	User can be checked reports
Pre-Condition	L2 and L3 users will be can see limited reports User should login to the system
Flow of Events	1. Enter to reporting module 2. Set filter fields and search
Post Condition	User relevant reports will be shown

Table 23: Use Case Description for Reporting

Appendix F – Code Segments

Web Element Converter: - database haven't web element names are saved as string. All types are saved as numbers. When system need to show element types in UI, should need a web element converter. That service is provided from this code segment

```
if (typeNumber == 1) {

            return "Class";

        } else if (typeNumber == 2) {

            return "ID";

        } else if (typeNumber == 3) {

            return "XPath";

        } else if (typeNumber == 4) {

            return "Tag Name";

        } else if (typeNumber == 5) {

            return "Name";

        } else if (typeNumber == 6) {

            return "Link Text";

        } else if (typeNumber == 7) {
```

Figure 21 Web Element Converter

AES Encrypt and Decrypt: - cookies need to encrypt for security reasons when keep storing cookies. Vice versa are also required when need have the cookie data. These responsibilities are taken by this code segment.

```
public static String encryptCookies(String value) throws Exception {


        Key key = generateKey();
        Cipher cipher = Cipher.getInstance(AESEncryptDecrypt.ALGORITHM);
        cipher.init(Cipher.ENCRYPT_MODE, key);
        byte[] encryptedByteValue = cipher.doFinal(value.getBytes("utf-8"));
        String encryptedValue64 = new BASE64Encoder().encode(encryptedByteValue);
        return encryptedValue64;

    }



    public static String decryptCookies(String value) throws Exception {
        Key key = generateKey();
        Cipher cipher = Cipher.getInstance(AESEncryptDecrypt.ALGORITHM);
        cipher.init(Cipher.DECRYPT_MODE, key);
        byte[] decryptedValue64 = new BASE64Decoder().decodeBuffer(value);
        byte[] decryptedByteValue = cipher.doFinal(decryptedValue64);
        String decryptedValue = new String(decryptedByteValue, "utf-8");
        return decryptedValue;

    }



    private static Key generateKey() throws Exception {
        Key key = new SecretKeySpec(AESEncryptDecrypt.KEY.getBytes(),
AESEncryptDecrypt.ALGORITHM);
        return key;
```

Figure 22 AES Encrypt and Decrypt

Three Time Attempt Handler – if a user try to loin to the system with three time consecutive invalid password within a minute, user need to be blocked.

```
public class ThreeTimeAttempt {



        LoginModel logingMod;



        public ThreeTimeAttempt(LoginModel logingMod) {

                this.logingMod = logingMod;
        }



        public short createThreeTimeAttemptSession(){



                 short visitCount = 1;
           String visitCountKey="visitCount"+logingMod.getEmail();
           String userID = logingMod.getEmail();



           if(logingMod.getSession()!=null){

                 long currentTime = Calendar.getInstance().getTimeInMillis(); // miliseconds

                 long expireTime=((logingMod.getSession().getCreationTime())+60000);//
milisceonds
```

Figure 23 Three Time Attempt Handler Part 1

```
            Date ttttt1=new Date(currentTime);
            Date ttttt2=new Date(expireTime * 1000);



            // 60000 miliseconds --> 1 miniutes --> 60 seconds

                if(expireTime < currentTime){

                    logingMod.getSession().invalidate(); // distroy old session
                    HttpSession session =
logingMod.getRequest().getSession(true);
                    logingMod.setSession(session);
                }
        }
        // is session new
        if (logingMod.getSession().getAttribute(userID)==null){



            //set the userID attribute to the session
            logingMod.getSession().setAttribute(userID, userID);
            logingMod.getSession().setAttribute(visitCountKey,  visitCount);
        } else {



            String temp=logingMod.getSession().getAttribute(visitCountKey).toString();
            int tempttoint=Integer.parseInt(temp);
```

Figure 24 Three Time Attempt Handler Part 2

```
                    //object convert to string object and string object convert to integer

                    //is visitCount less than the 3

                            if((logingMod.getSession().getAttribute(visitCountKey)!=null) &&
(tempttoint < 2 )  ){

if(logingMod.getSession().getAttribute(visitCountKey)==null){

//set the visitCountKey attribute to the session

logingMod.getSession().setAttribute(visitCountKey,  visitCount);

                                                }

                                            visitCount = (Short)
logingMod.getSession().getAttribute(visitCountKey);

                                            visitCount = (short) (visitCount + 1);

                                            userID =
(String)logingMod.getSession().getAttribute(userID);

                                            logingMod.getSession().setAttribute(visitCountKey,
visitCount);

                            }else{

                                      visitCount = (Short)
logingMod.getSession().getAttribute(visitCountKey);

                                 visitCount = (short) (visitCount + 1);

                                      //Destroy session after three time attempt (reset session)

                                      logingMod.getSession().invalidate();

                                      return visitCount;

                            }

               }


                    return visitCount;

           }


}
```

Figure 25 Three Time Attempt Handler Part 3

Login Module – MD5HashService.java

```
public class MD5HashService {

    public String getMD5HashValue(String password){

        //Generates a MessageDigest object implementing the specified algorithm, as supplied from the specified provider, if such an algorithm is available from the provider.

        MessageDigest md5;

        try {

            md5 = MessageDigest.getInstance("MD5");

            //Resets the digest for further use.

            md5.reset();

            //Updates the digest using the specified byte

            md5.update(password.getBytes());


            //Completes the hash computation by performing final operations such as padding.

            byte[] digest = md5.digest();

            //This constructor is used to translate the sign-magnitude representation of a BigInteger into a BigInteger.

            BigInteger bigInt = new BigInteger(1,digest);

            String hashtext = bigInt.toString(16);

            // Now we need to zero pad it if you actually want the full 32 chars.


            while(hashtext.length() < 32 ){

              hashtext = "0"+hashtext;

            }

            return hashtext;

        } catch (NoSuchAlgorithmException e) {

            e.printStackTrace();

        }

        return null;

    }
```

Figure 26 MD5HashService.java Part 1

Database Settings Module – DBSettingsWindowController.java

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {


                    String host=request.getParameter("host");
                    Short port=Short.parseShort(request.getParameter("port"));
                    String dbname=request.getParameter("dbname");
                    String username=request.getParameter("username");
                    String password=request.getParameter("password");

                    Cookie[] cookies = request.getCookies();
                    String[] cookieNameContainer=new String[5];

                    cookieNameContainer[0]="tTDBSettingsHost";
                    cookieNameContainer[1]="tTDBSettingsPort";
                    cookieNameContainer[2]="tTDBSettingsDBName";
                    cookieNameContainer[3]="tTDBSettingsUsername";
                    cookieNameContainer[4]="tTDBSettingsPassword";
                    BDSettingsWindowModel dBSWMod=new BDSettingsWindowModel();

                    dBSWMod.setHost(host);
                    dBSWMod.setPort(port);
                    dBSWMod.setDbname(dbname);
                    dBSWMod.setUsername(username);
                    dBSWMod.setPassword(password);
                    dBSWMod.setRequest(request);
                    dBSWMod.setResponse(response);
```

Figure 27 DBSettingsWindowController.java part 1

```
                    dBSWMod.setNotHostCookieExist(true);
                    dBSWMod.setNotPortCookieExist(true);
                    dBSWMod.setNotDBNameCookieExist(true);
                    dBSWMod.setNotUsernameCookieExist(true);
                    dBSWMod.setNotPasswordCookieExist(true);

                    dBSWMod.setCookies(cookies);

                    dBSWMod.setCookieNameContainer(cookieNameContainer);



                    DBSettingsWindowImpl dBSWImpl=new DBSettingsWindowImpl();



                    dBSWImpl.cookiesHandle(dBSWMod);



                    RequestDispatcher rd=request.getRequestDispatcher("Login.jsp");
                    rd.forward(request, response);




          }

}
```

Figure 28 DBSettingsWindowController.java part 2

Database Settings Module – DBSettingsWindowImpl.java

```
public void cookieSetter(BDSettingsWindowModel dBSWMod) {

                    String ii=dBSWMod.getRequest().getParameter(dBSWMod.getHost());

                    if( dBSWMod.isNotHostCookieExist() ){

                              Cookie cookie1 = new 
Cookie(dBSWMod.getCookieNameContainer()[0], ii );

                    dBSWMod.getResponse().addCookie(cookie1);

             }else{

                    //overwrite the new value for cookie

                    Cookie cookie2 = new Cookie(dBSWMod.getCookieNameContainer()[0], 
(dBSWMod.getRequest().getParameter("host")));

                              cookie2.setValue(dBSWMod.getHost());

                              dBSWMod.getResponse().addCookie(cookie2);

             }



                    if( dBSWMod.isNotPortCookieExist() ){



                              Cookie cookie = new Cookie(dBSWMod.getCookieNameContainer()[1], 
(dBSWMod.getRequest().getParameter(dBSWMod.getPort().toString())));

                    dBSWMod.getResponse().addCookie(cookie);



             }else{



                    Cookie cookie = new Cookie(dBSWMod.getCookieNameContainer()[1], 
(dBSWMod.getRequest().getParameter(dBSWMod.getPort().toString())));

                              cookie.setValue(dBSWMod.getPort().toString());

                              dBSWMod.getResponse().addCookie(cookie);



             }
```

Figure 29 DBSettingsWindowImpl.java Part 1

Database Settings Module – BDSettingsWindowModel.java

```
public class BDSettingsWindowModel {
        private String host;
        private Short port;
        private String dbname;
        private String username;
        private String password;
        private HttpServletRequest request;
        private HttpServletResponse response;
        private Cookie[] cookies;
        private boolean isNotHostCookieExist;
        private boolean isNotPortCookieExist;
        private boolean isNotDBNameCookieExist;
        private boolean isNotUsernameCookieExist;
        private boolean isNotPasswordCookieExist;

        private String[] cookieNameContainer;



        public String[] getCookieNameContainer() {
                return cookieNameContainer;
        }
        public void setCookieNameContainer(String[] cookieNameContainer) {
                this.cookieNameContainer = cookieNameContainer;
        }
        public Cookie[] getCookies() {
                return cookies;
        }
```

Figure 30 BDSettingsWindowModel.java Part 1

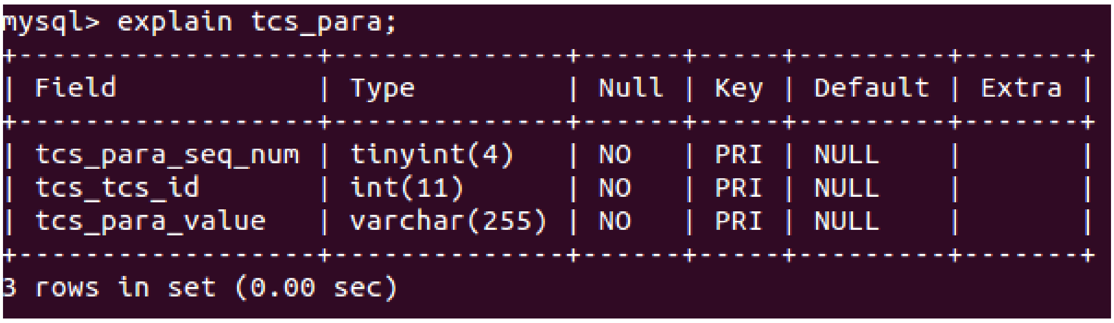
Database Settings Module – DBSettingsWindowInterface.java

```
package com.ctas.dbsettingswindow.model;


public interface DBSettingsWindowInterface {



        void cookiesHandle(BDSettingsWindowModel dBSWMod);

        void cookieSetter(BDSettingsWindowModel dBSWMod);


}
```

Figure 31 DBSettingsWindowInterface.java Part 1

Appendix G – Database Implementation

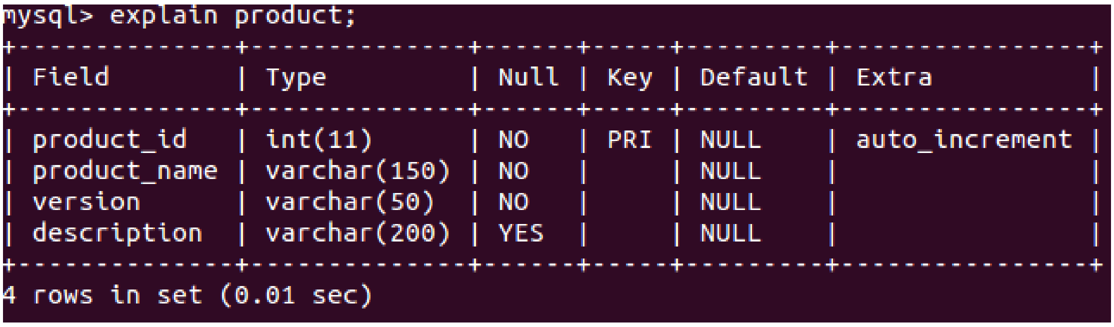
tcs_para



```
mysql> explain tcs_para;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| tcs_para_seq_num | tinyint(4)    | NO   | PRI | NULL    |       |
| tcs_tcs_id       | int(11)       | NO   | PRI | NULL    |       |
| tcs_para_value   | varchar(255)  | NO   | PRI | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Figure 32 tcs_para

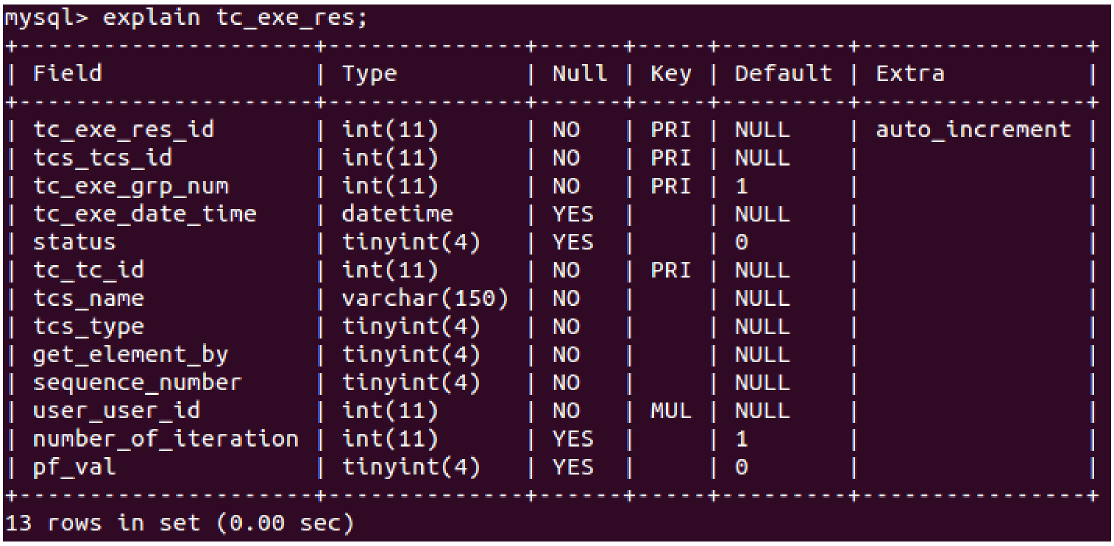
product



```
mysql> explain product;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| product_id   | int(11)      | NO   | PRI | NULL    | auto_increment |
| product_name | varchar(150) | NO   |     | NULL    |                |
| version      | varchar(50)  | NO   |     | NULL    |                |
| description  | varchar(200) | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

Figure 33 product

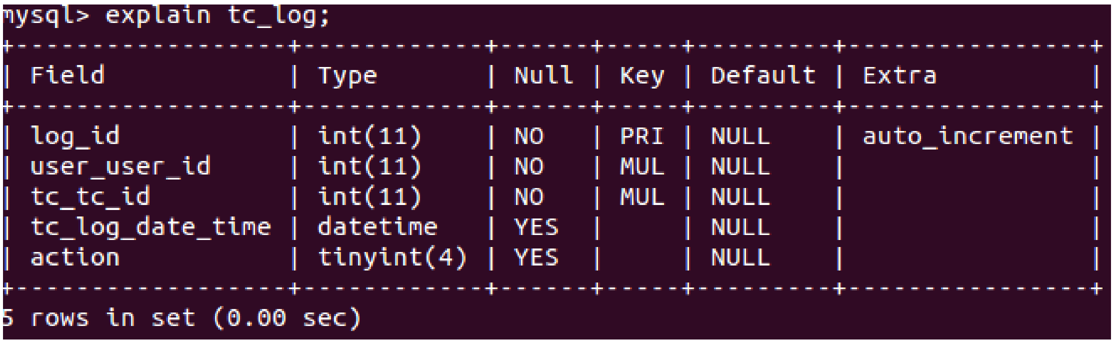
tc_exe_res



```
mysql> explain tc_exe_res;
+---------------------+--------------+------+-----+---------+----------------+
| Field               | Type         | Null | Key | Default | Extra          |
+---------------------+--------------+------+-----+---------+----------------+
| tc_exe_res_id       | int(11)      | NO   | PRI | NULL    | auto_increment |
| tcs_tcs_id          | int(11)      | NO   | PRI | NULL    |                |
| tc_exe_grp_num      | int(11)      | NO   | PRI | 1       |                |
| tc_exe_date_time    | datetime     | YES  |     | NULL    |                |
| status              | tinyint(4)   | YES  |     | 0       |                |
| tc_tc_id            | int(11)      | NO   | PRI | NULL    |                |
| tcs_name            | varchar(150) | NO   |     | NULL    |                |
| tcs_type            | tinyint(4)   | NO   |     | NULL    |                |
| get_element_by      | tinyint(4)   | NO   |     | NULL    |                |
| sequence_number     | tinyint(4)   | NO   |     | NULL    |                |
| user_user_id        | int(11)      | NO   | MUL | NULL    |                |
| number_of_iteration | int(11)      | YES  |     | 1       |                |
| pf_val              | tinyint(4)   | YES  |     | 0       |                |
+---------------------+--------------+------+-----+---------+----------------+
13 rows in set (0.00 sec)
```

Figure 34 tc_exe_res

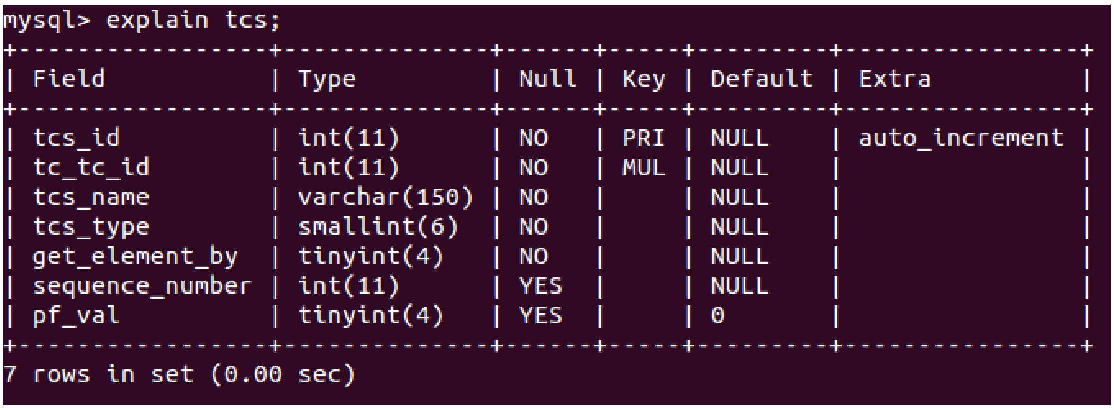
tc_logs



```
mysql> explain tc_log;
+-------------------+------------+------+-----+---------+----------------+
| Field             | Type       | Null | Key | Default | Extra          |
+-------------------+------------+------+-----+---------+----------------+
| log_id            | int(11)    | NO   | PRI | NULL    | auto_increment |
| user_user_id      | int(11)    | NO   | MUL | NULL    |                |
| tc_tc_id          | int(11)    | NO   | MUL | NULL    |                |
| tc_log_date_time  | datetime   | YES  |     | NULL    |                |
| action            | tinyint(4) | YES  |     | NULL    |                |
+-------------------+------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Figure 35 tc_logs

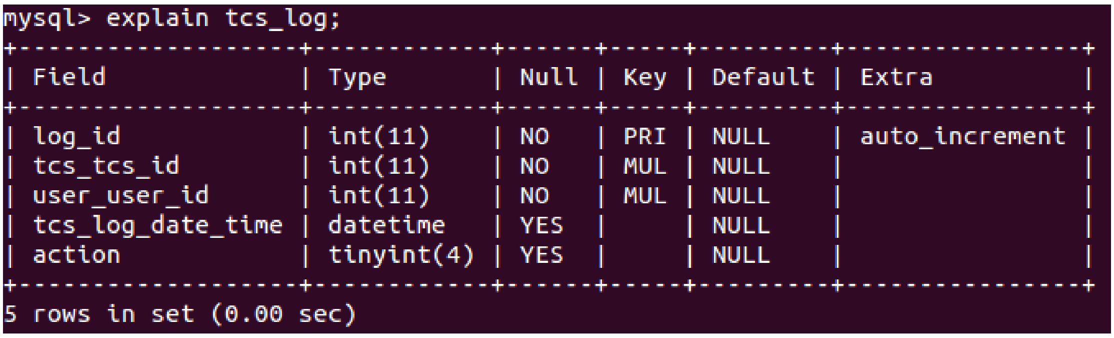
tcs



```
mysql> explain tcs;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| tcs_id          | int(11)      | NO   | PRI | NULL    | auto_increment |
| tc_tc_id        | int(11)      | NO   | MUL | NULL    |                |
| tcs_name        | varchar(150) | NO   |     | NULL    |                |
| tcs_type        | smallint(6)  | NO   |     | NULL    |                |
| get_element_by  | tinyint(4)   | NO   |     | NULL    |                |
| sequence_number | int(11)      | YES  |     | NULL    |                |
| pf_val          | tinyint(4)   | YES  |     | 0       |                |
+-----------------+--------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

Figure 36 tcs

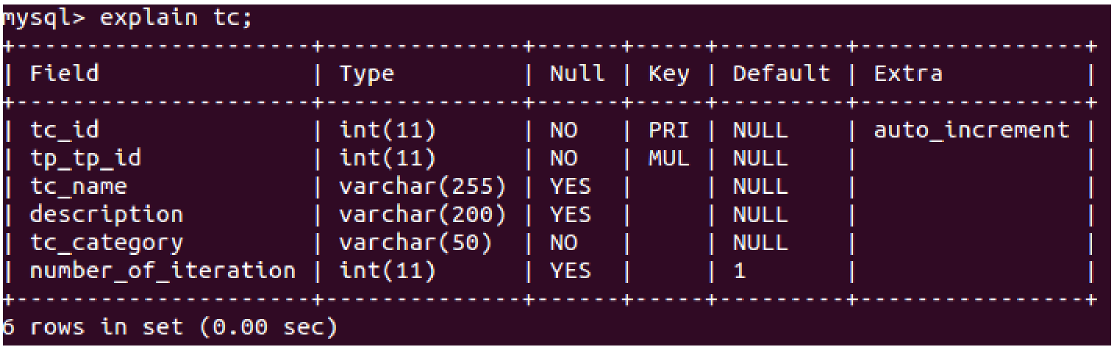
tcs_logs



```
mysql> explain tcs_log;
+-------------------+------------+------+-----+---------+----------------+
| Field             | Type       | Null | Key | Default | Extra          |
+-------------------+------------+------+-----+---------+----------------+
| log_id            | int(11)    | NO   | PRI | NULL    | auto_increment |
| tcs_tcs_id        | int(11)    | NO   | MUL | NULL    |                |
| user_user_id      | int(11)    | NO   | MUL | NULL    |                |
| tcs_log_date_time | datetime   | YES  |     | NULL    |                |
| action            | tinyint(4) | YES  |     | NULL    |                |
+-------------------+------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Figure 37 tcs_log

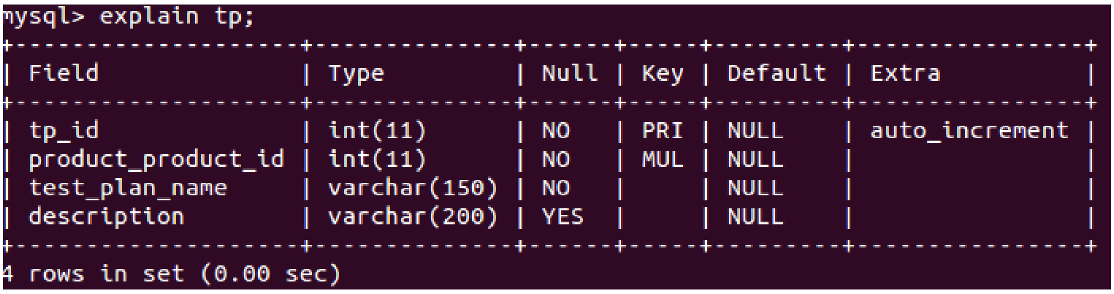
tc



```
mysql> explain tc;
+---------------------+--------------+------+-----+---------+----------------+
| Field               | Type         | Null | Key | Default | Extra          |
+---------------------+--------------+------+-----+---------+----------------+
| tc_id               | int(11)      | NO   | PRI | NULL    | auto_increment |
| tp_tp_id            | int(11)      | NO   | MUL | NULL    |                |
| tc_name             | varchar(255) | YES  |     | NULL    |                |
| description         | varchar(200) | YES  |     | NULL    |                |
| tc_category         | varchar(50)  | NO   |     | NULL    |                |
| number_of_iteration | int(11)      | YES  |     | 1       |                |
+---------------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

Figure 38 tc

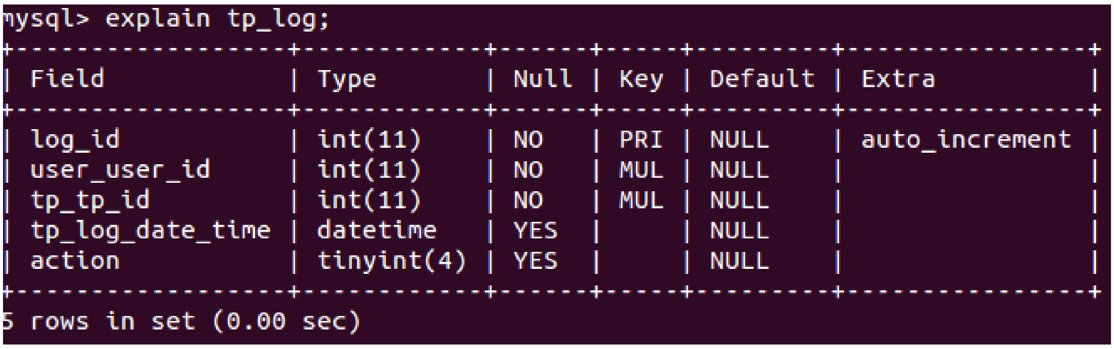
tp



```
mysql> explain tp;
+--------------------+--------------+------+-----+---------+----------------+
| Field              | Type         | Null | Key | Default | Extra          |
+--------------------+--------------+------+-----+---------+----------------+
| tp_id              | int(11)      | NO   | PRI | NULL    | auto_increment |
| product_product_id | int(11)      | NO   | MUL | NULL    |                |
| test_plan_name     | varchar(150) | NO   |     | NULL    |                |
| description        | varchar(200) | YES  |     | NULL    |                |
+--------------------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Figure 39 tp

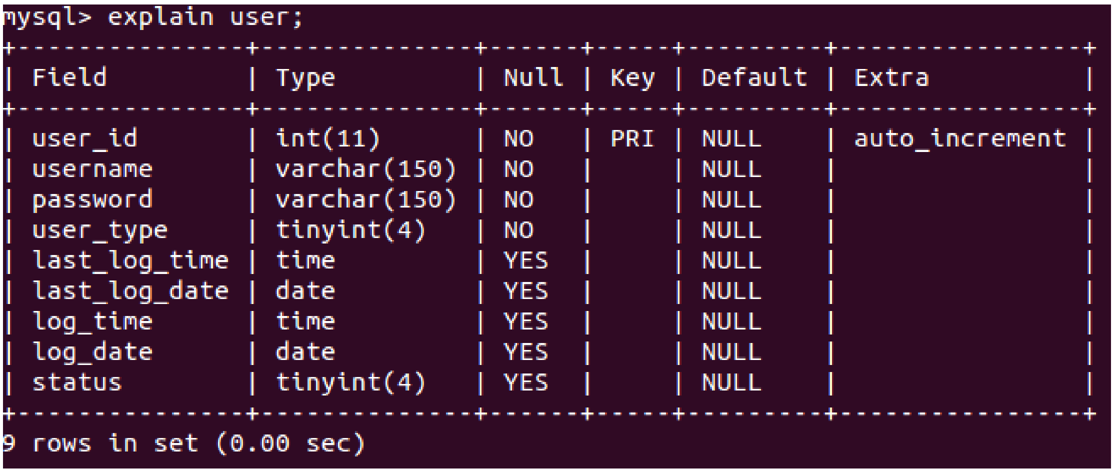
tp_logs



```
mysql> explain tp_log;
+------------------+------------+------+-----+---------+----------------+
| Field            | Type       | Null | Key | Default | Extra          |
+------------------+------------+------+-----+---------+----------------+
| log_id           | int(11)    | NO   | PRI | NULL    | auto_increment |
| user_user_id     | int(11)    | NO   | MUL | NULL    |                |
| tp_tp_id         | int(11)    | NO   | MUL | NULL    |                |
| tp_log_date_time | datetime   | YES  |     | NULL    |                |
| action           | tinyint(4) | YES  |     | NULL    |                |
+------------------+------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Figure 40 tp_log

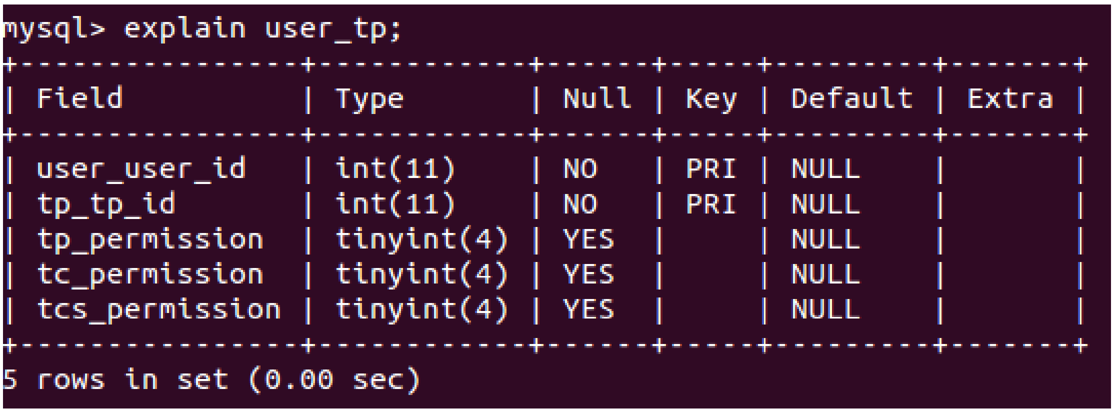
user



```
mysql> explain user;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| user_id       | int(11)      | NO   | PRI | NULL    | auto_increment |
| username      | varchar(150) | NO   |     | NULL    |                |
| password      | varchar(150) | NO   |     | NULL    |                |
| user_type     | tinyint(4)   | NO   |     | NULL    |                |
| last_log_time | time         | YES  |     | NULL    |                |
| last_log_date | date         | YES  |     | NULL    |                |
| log_time      | time         | YES  |     | NULL    |                |
| log_date      | date         | YES  |     | NULL    |                |
| status        | tinyint(4)   | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
9 rows in set (0.00 sec)
```

Figure 41 user

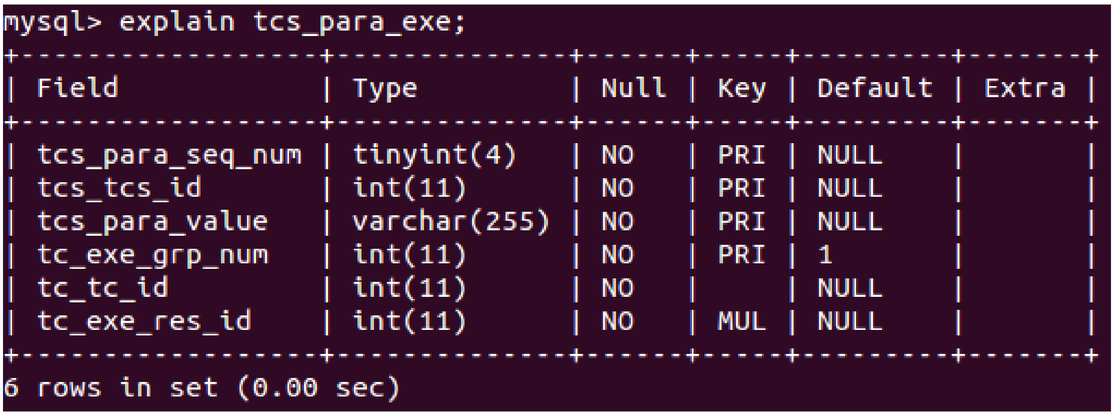
user_tp



```
mysql> explain user_tp;
+----------------+------------+------+-----+---------+-------+
| Field          | Type       | Null | Key | Default | Extra |
+----------------+------------+------+-----+---------+-------+
| user_user_id   | int(11)    | NO   | PRI | NULL    |       |
| tp_tp_id       | int(11)    | NO   | PRI | NULL    |       |
| tp_permission  | tinyint(4) | YES  |     | NULL    |       |
| tc_permission  | tinyint(4) | YES  |     | NULL    |       |
| tcs_permission | tinyint(4) | YES  |     | NULL    |       |
+----------------+------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

Figure 42 user_tp

tcs_para_exe



```
mysql> explain tcs_para_exe;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| tcs_para_seq_num | tinyint(4)   | NO   | PRI | NULL    |       |
| tcs_tcs_id       | int(11)      | NO   | PRI | NULL    |       |
| tcs_para_value   | varchar(255) | NO   | PRI | NULL    |       |
| tc_exe_grp_num   | int(11)      | NO   | PRI | 1       |       |
| tc_tc_id         | int(11)      | NO   |     | NULL    |       |
| tc_exe_res_id    | int(11)      | NO   | MUL | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Figure 43 tc_para_exe

Appendix H – User Interfaces

Login

Carried on the authentication process.

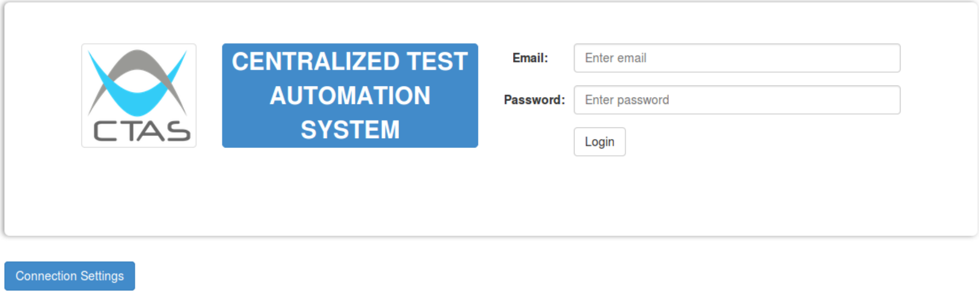


Figure 44 Login Interface

Database Settings

Let user to make the database connection to centralized database.

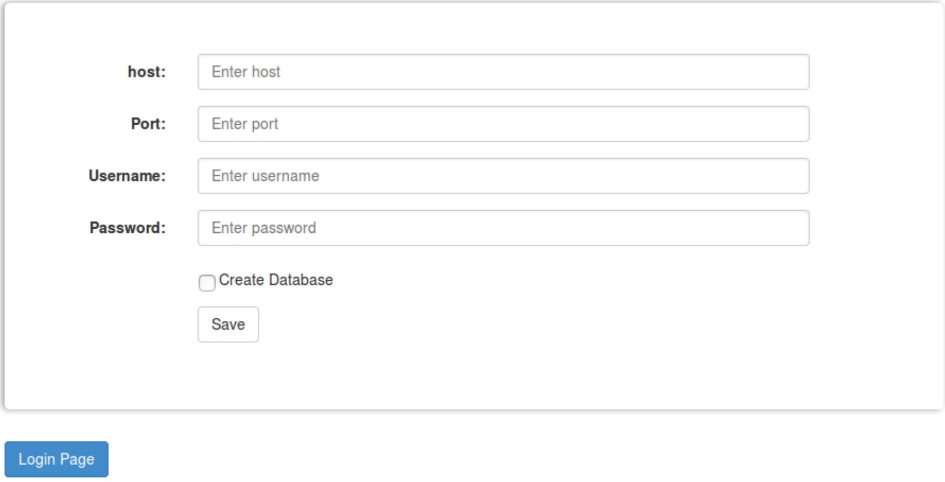
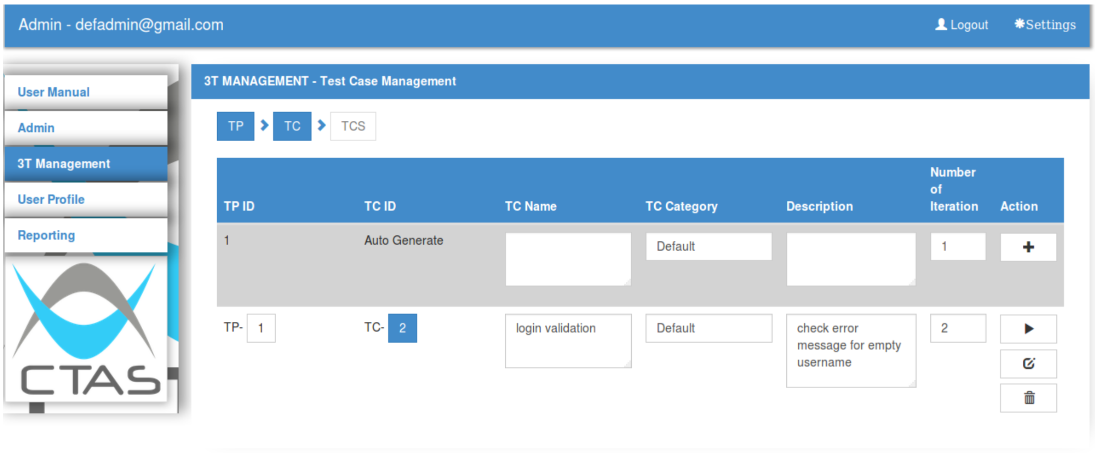


Figure 45 Database Settings Interface

TC Management in 3T Management

TC create, delete, update and execution actions can be taken from this interface



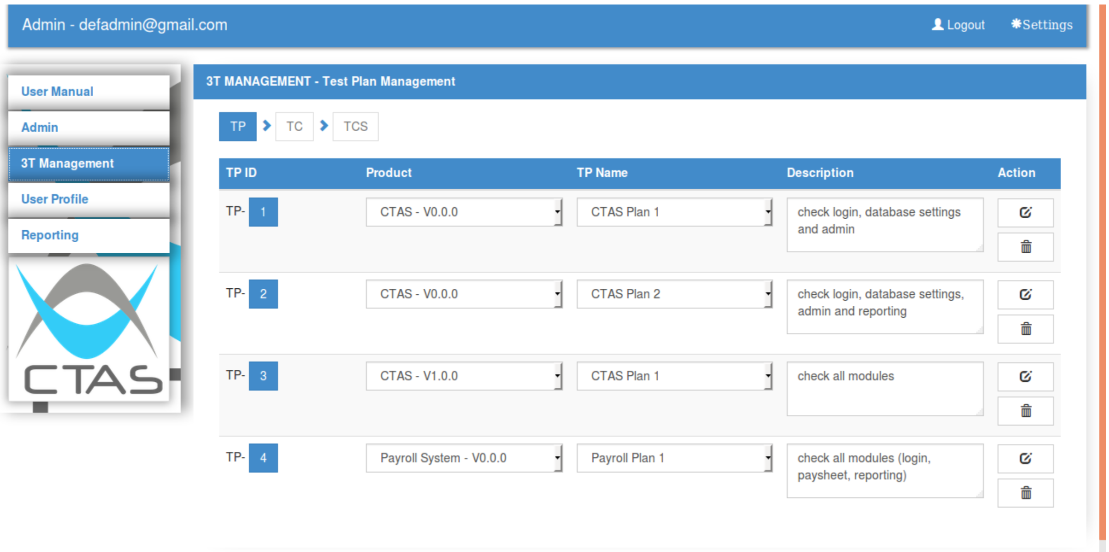
3T MANAGEMENT - Test Case Management

TP ID	TC ID	TC Name	TC Category	Description	Number of Iteration	Action
1	Auto Generate		Default		1	+
TP- 1	TC- 2	login validation	Default	check error message for empty username	2	▶

Figure 46 TC Management in 3T Management

TP Management in 3T Management

TP delete, update actions can be taken from this interface



3T MANAGEMENT - Test Plan Management

TP ID	Product	TP Name	Description	Action
TP- 1	CTAS - V0.0.0	CTAS Plan 1	check login, database settings and admin	
TP- 2	CTAS - V0.0.0	CTAS Plan 2	check login, database settings, admin and reporting	
TP- 3	CTAS - V1.0.0	CTAS Plan 1	check all modules	
TP- 4	Payroll System - V0.0.0	Payroll Plan 1	check all modules (login, paysheet, reporting)	

Figure 47 TP Management in 3T Management