MILESTONE 3

PROJECT TITLE: SMART-ASSET VAULT ENTERPRISE

Submitted by

S. No	Name	Student ID
1	Aiswarya Antu	0983439
2	Aiswarya Roy	0996480
3	Priya Meelu	0994395
4	Shenglin Meng	1004708
5	Umashankar Sundarrajan	0992253
6	Venkata Pisupati – (Team Leader)	1014937

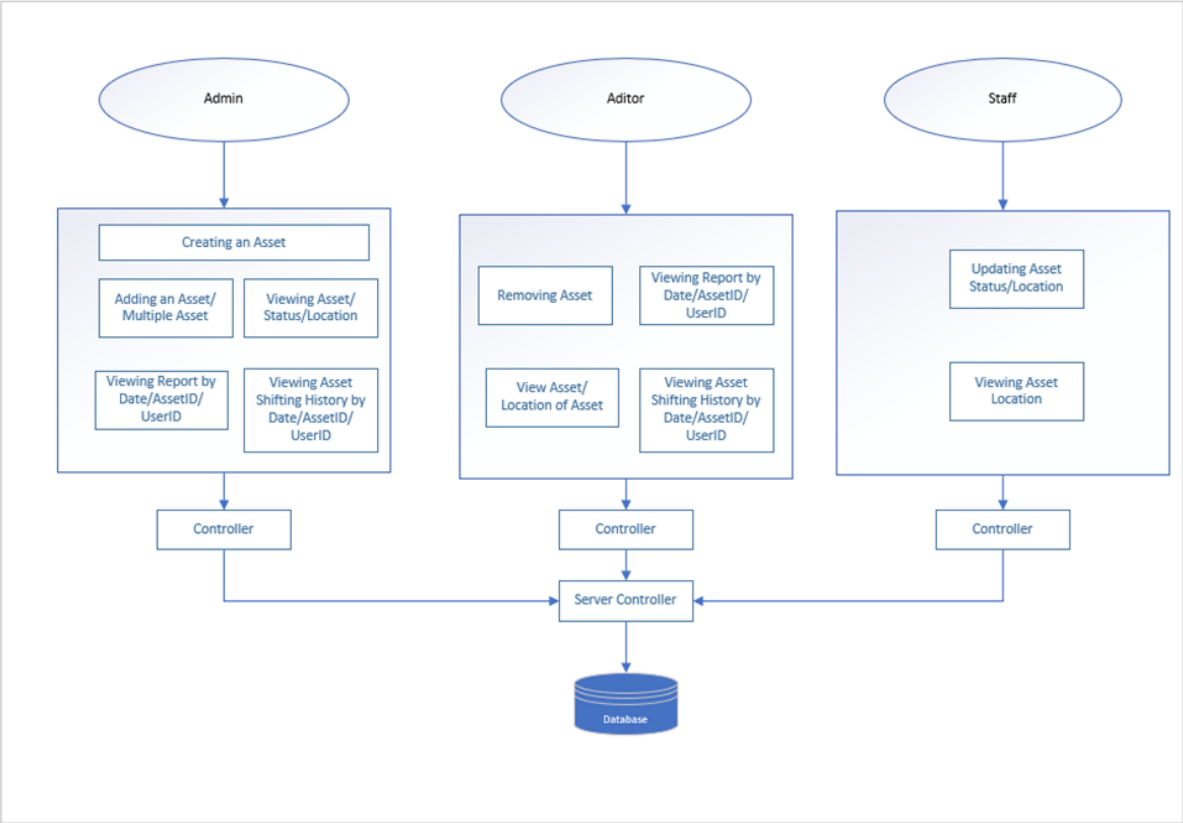
Professor – Mona Abou Taka

Submission Date – 20th Mar 2022

Contents

1 System Architecture

1.1 Architecture Diagram



An architecture diagram is a visual representation of all the elements that make up part, or all, of a system. Above all, it helps the engineers, designers, stakeholders, and anyone else involved in the project, understand a system layout. There are three main actors including Admin, Auditor, and the Staff.

Admin can do the following activities:

1. Create an Asset: Create new asset which is needed for the enterprise using asset id.

2. Adding an Asset/ Multiple Asset: Adding the received asset in the system, which can be a single asset or multiple assets or a group of assets.

3. Viewing Asset/Status/Location: The admin can view the asset once it is added to the system and the current status of the system, also the location where the asset is added.

4. Admin can access the report by filtering using the date, AssetID or userID. - Optional

5. Viewing Asset shifting history: It means once an asset is added to the location and someone else changed the asset from that location and moved to some other room, based on the AssetID admin can track the actual place of the asset and also the current location of the asset. So, admin can access the shifting history of the asset.

Auditor is the third-party person or the stakeholder for managing and viewing the asset. Auditor can do the following activities:

1. Remove Asset: Auditor can remove the asset which is no longer needed.

2. Viewing Asset/Status/Location: The auditor can view the asset once it is added to the system and the status of the system, also the location where the asset is added.

3. Auditor can access the report by filtering using the date, AssetID or userID. - Optional

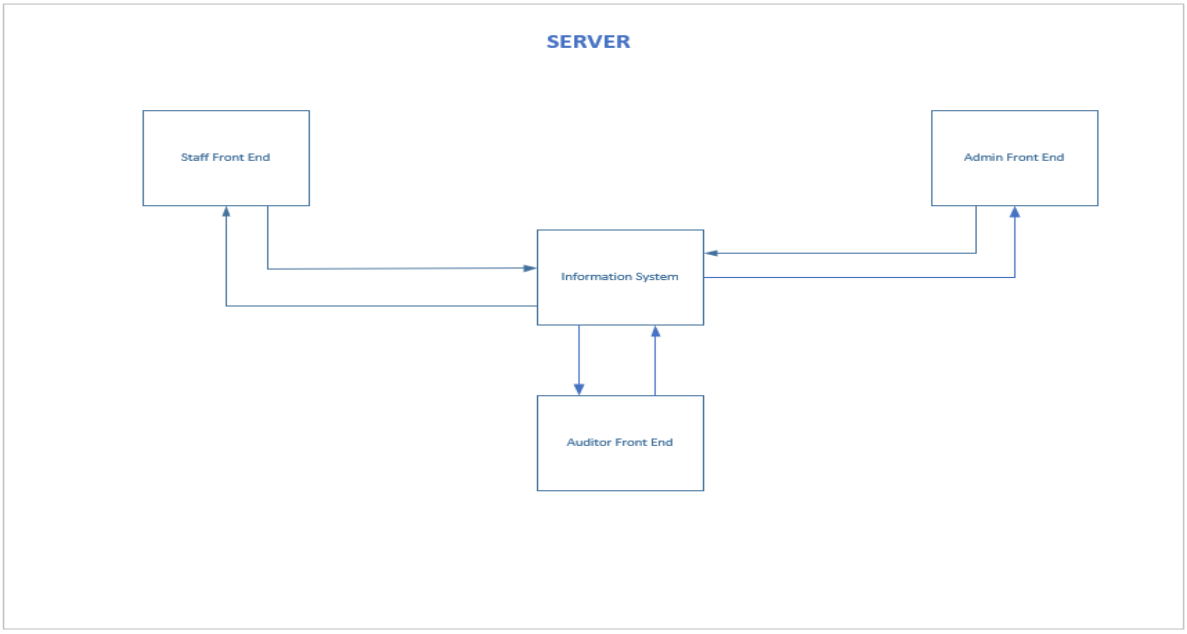
4. Viewing Asset shifting history: Shifting history can be viewed by the auditor same as the admin.

Staff is the other employees in that organization who have the access to edit or view the details of assets.

1. Update the asset/status/ location: any changes to the asset the staff can edit and update to the system.

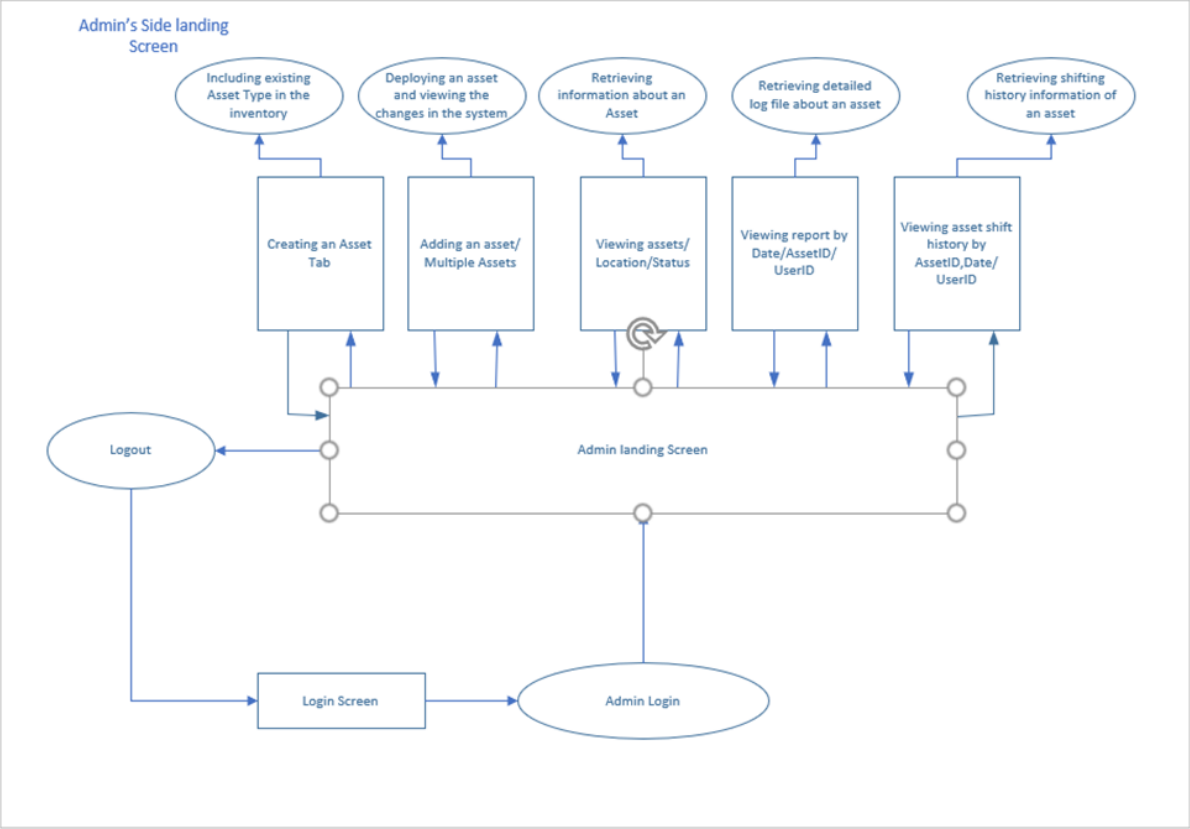
2. View the asset/location/status: They also have the permission to view the location of the asset location.

1.2 Server Diagram



The figure depicts a Server Diagram used mapping out your elements and device interactions. Data entry and data extraction can be performed by all users of the Asset management system - Staff, Administrator, and Auditor - from the front end. Each user has a request and response from the Information system as shown in the figure.

1.3 Admin Landing Screen



It's the detailed view of the admin landing screen. Here the admin has to login the system using the admin credentials. So, the admin can view the multiple tabs for the different functions such as:

1. Creating asset tab for creating new assets and to include the existing asset type in the inventory.

2. Adding an Asset/Multiple Asset where the admin can deploy assets and view the changes in the system.

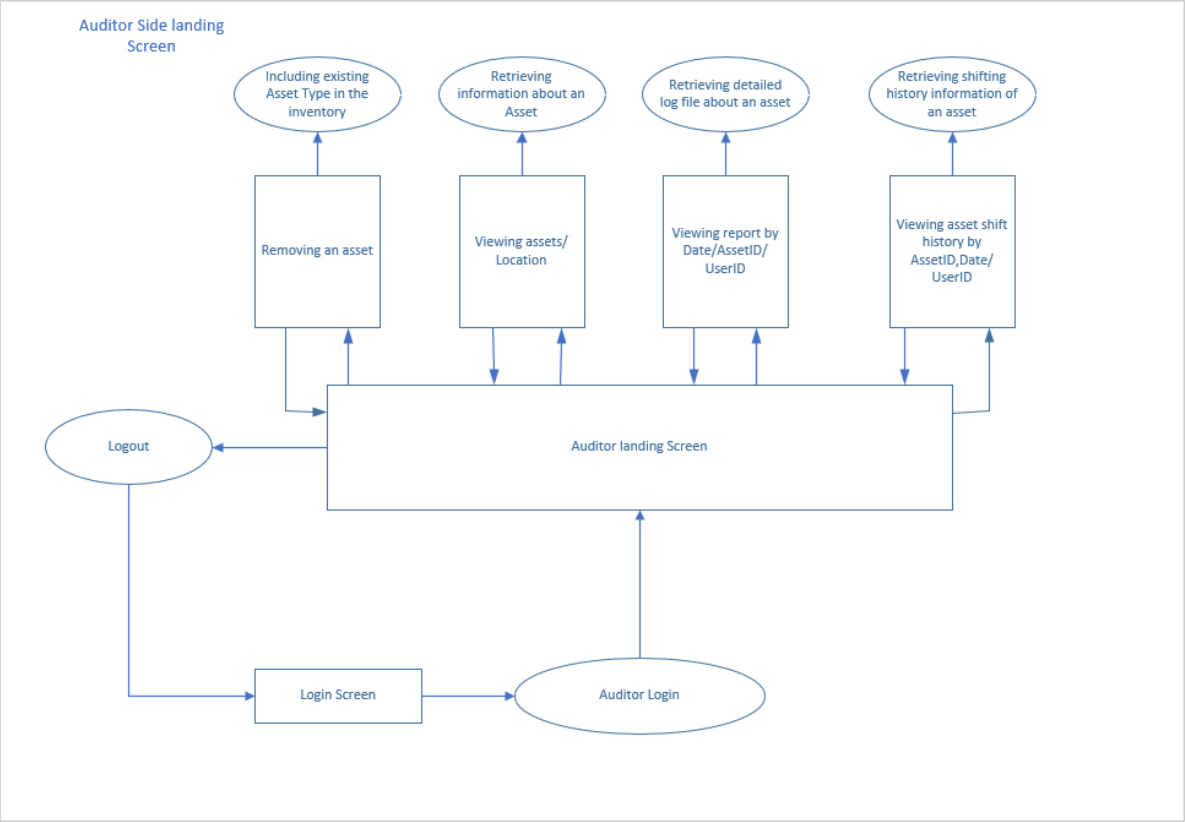
3. Retrieving information about an asset such as type of asset, location and status can be done by view asset/Location/Status Tab.

4. Detailed log files about assets can be identified by the viewing report tab using userID, AssetID, and Date. - Optional

5. Retrieving shifting history information of assets can be viewed by asset shift history tab by AssetID, Date and UserID.

6. Admin can log out once the activities are done.

1.4 Auditor Landing Screen



The figure indicates the landing screen of the auditor. The auditor has to login to the system using the auditor credentials.

1. The Auditor page includes the remove asset tab for removing the asset

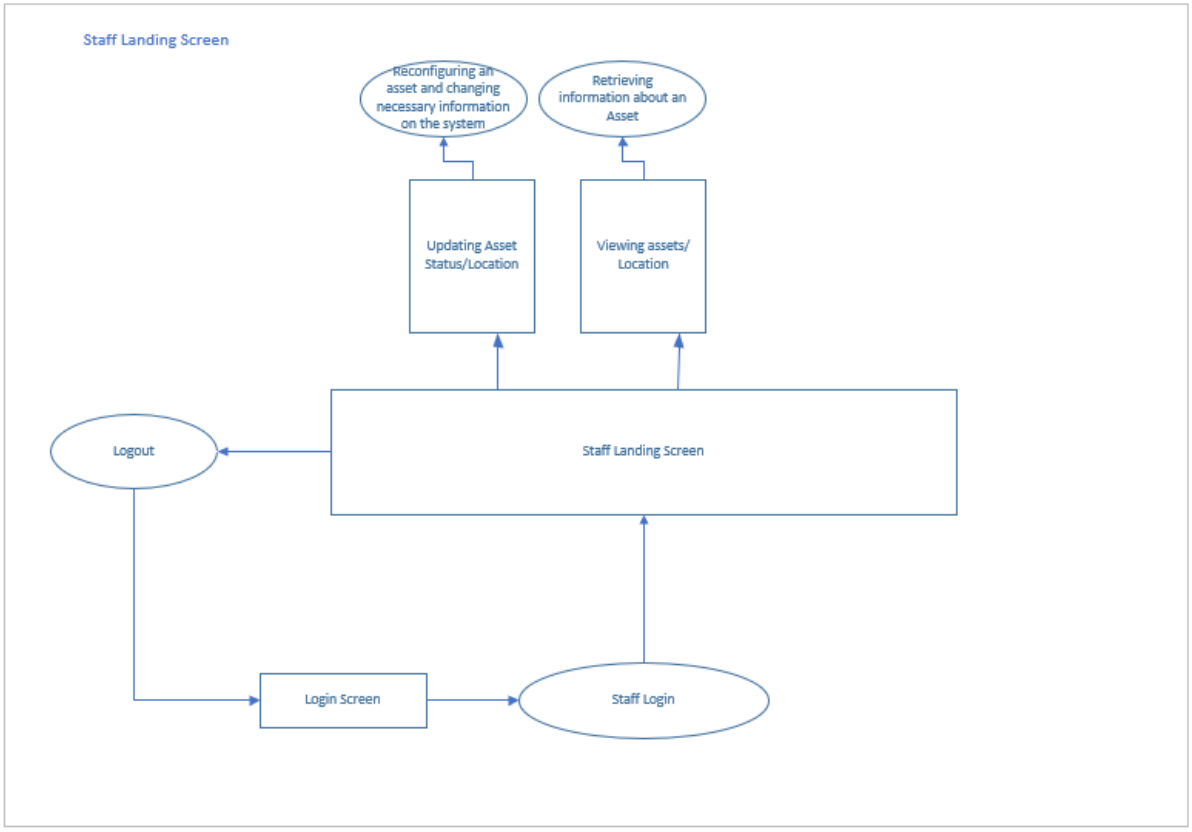
2. Retrieving information about an asset can be viewed by the view asset/Location Tab.

3. Detailed log files about assets can be identified by the viewing report tab using userID, AssetID, and Date. - Optional

4. Retrieving shifting history information of assets can be viewed by asset shift history tab by AssetID, Date and UserID.

5. Auditors can log out once the activities are done.

1.5 Staff Landing Screen

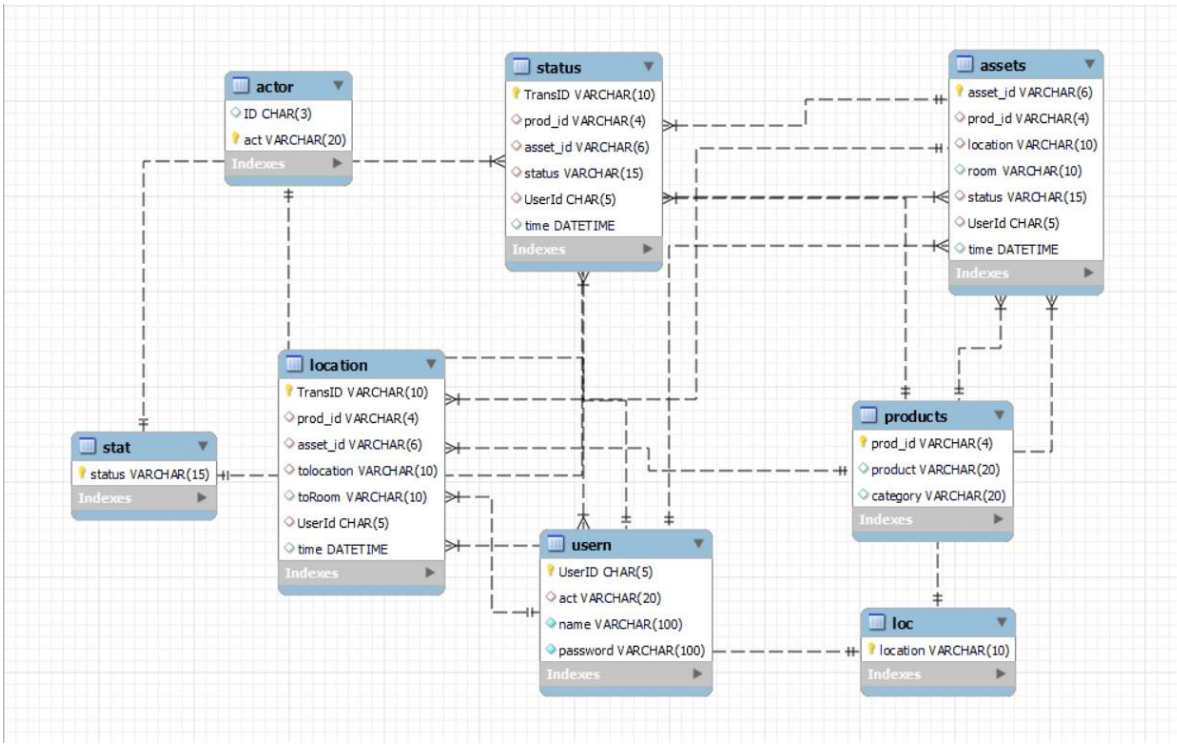


The figure pictures the Staff landing screen of SAVE. The staff has to login using staff credentials from the login page to be navigated to the staff landing page. Each staff are authorized to perform following activities:

1. Staff can view asset status/location to retrieve information about an asset, including its type, location, and status.

2. Staff can update asset status/location to reconfigure the asset information in the system.

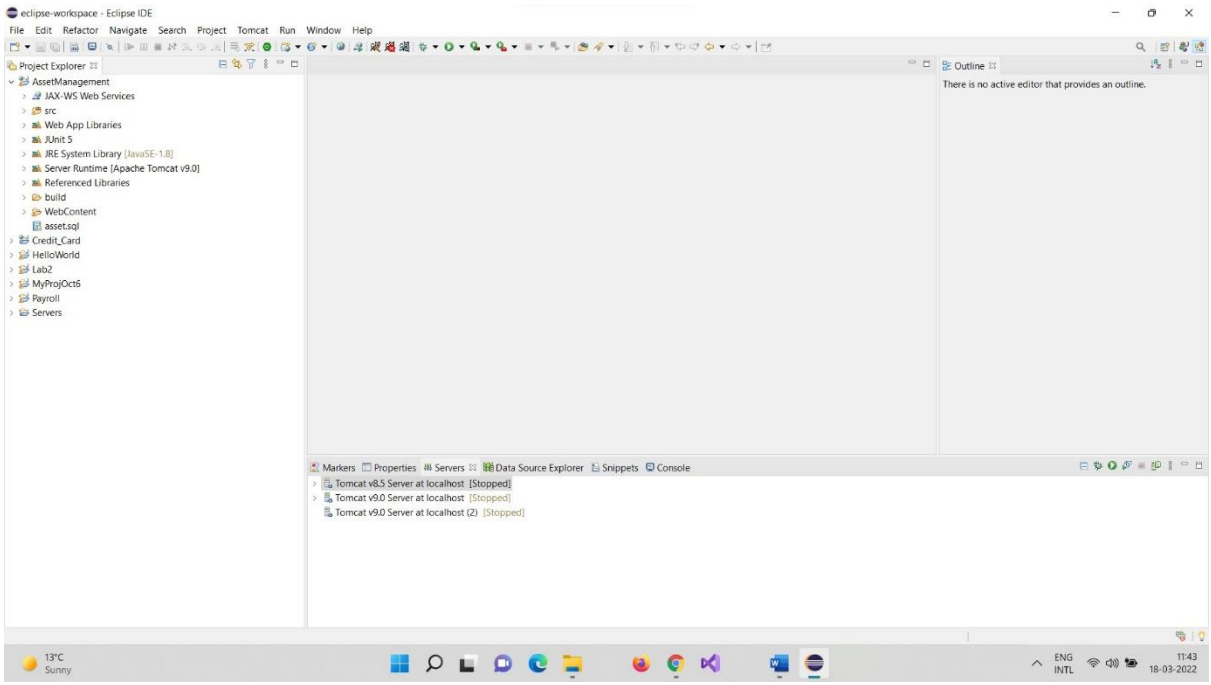
2 ER Diagram



- ✓ The status table is the main table in the er diagram. The TransID is the primary key in that table.
- ✓ In status table it provides details about product, asset, status, UserID, time and date.
- ✓ The location is the second major table which interlinks rest other tables.
- ✓ The TransID acts as the secondary key here.
- ✓ The other tables are also created for better object relational mapping and automation of other tasks like location, assets and other tasks.

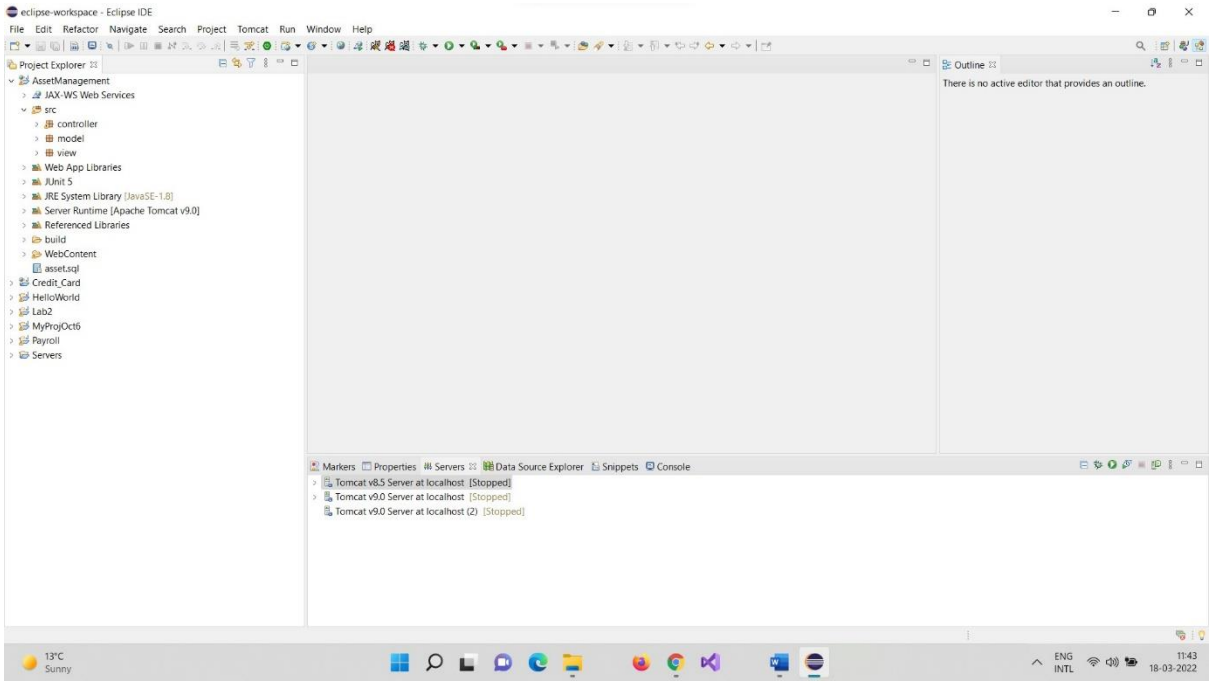
3 Draft of API

3.1 Project Creation



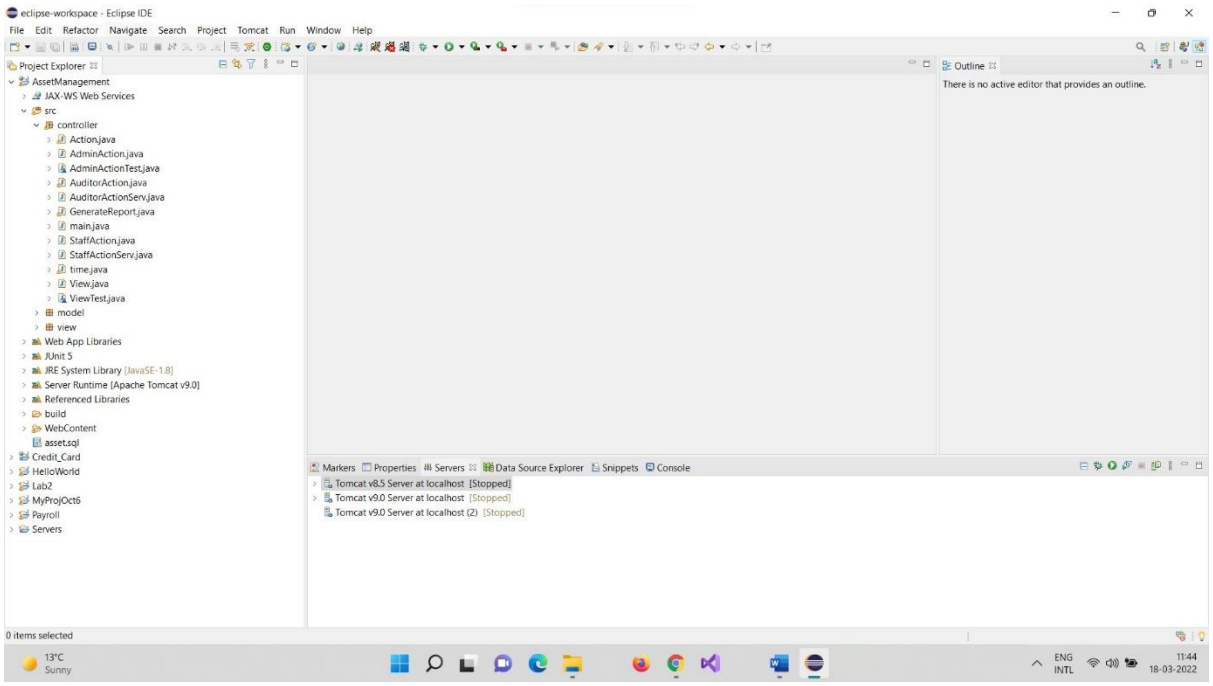
- ✓ A project creation is a set of documents that usually consists of a project description, Tasks, deliverables and risks.
- ✓ It is the initial creation of the project.
- ✓ The above eclipse workspace shows the details view of it.
- ✓ The left pane briefly shows the files and paths.

3.2 MVC model



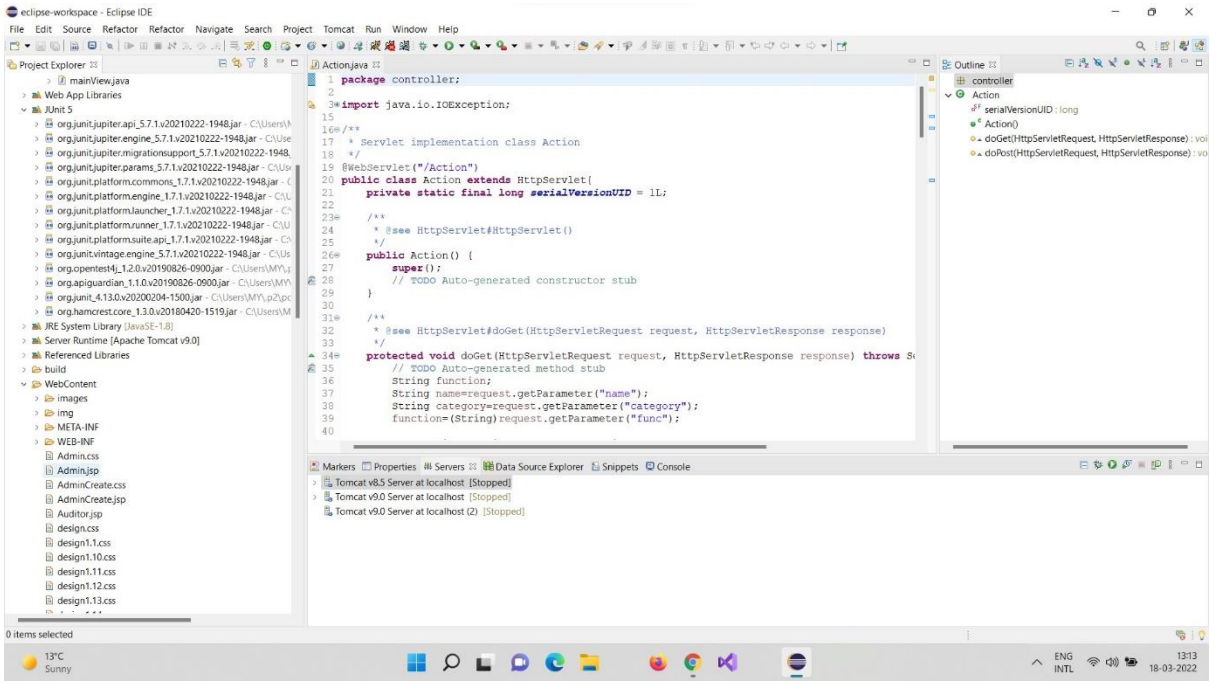
- ✓ The MVC model is the second phase in the draft of API.
- ✓ Model–view–controller (MVC) is a software design pattern commonly used for developing user interfaces that divide the related program logic into three interconnected elements.
- ✓ Under the source folder controller, model and view are listed.

3.3 Java files



- ✓ The Java files is the third phase in the draft of API.
- ✓ Once the controller has been expanded, we can they see the list of JAVA files.
- ✓ It plays a major role in processing the code and able to provide the required output of the product.

3.4 J-unit jar files and Web Content

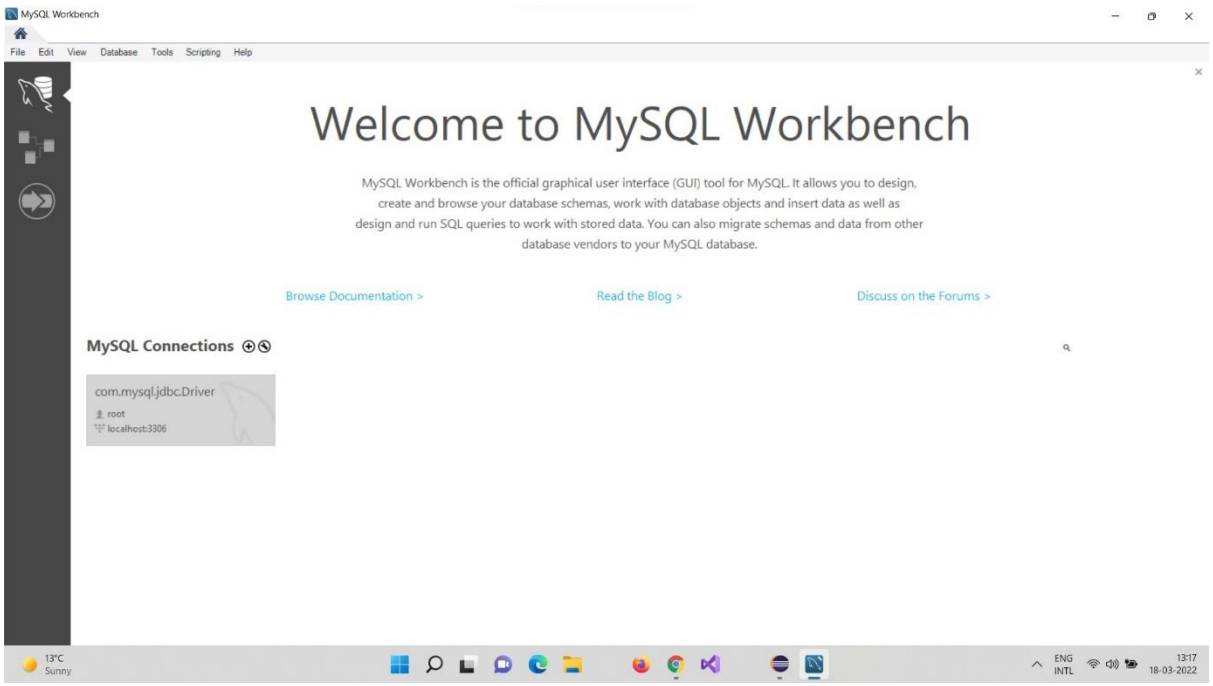


- ✓ The J- unit jar files and web content are the fourth phase in the draft of API.

- ✓ JUnit is a Java unit testing framework that's one of the best test methods for regression testing.

- ✓ An open-source framework, it is used to write and run repeatable automated tests.

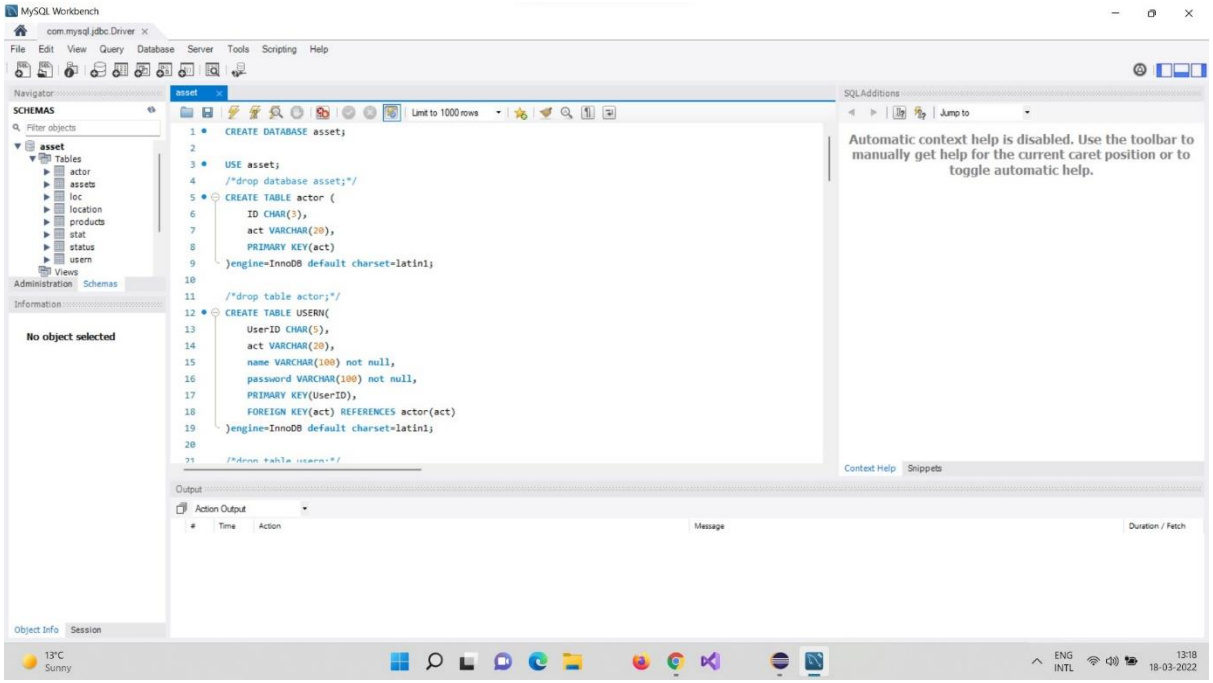
3.5 Database Connection Established



- ✓ After completing the above code and executing it we will be connecting to our database.

- ✓ The database connection will be established.

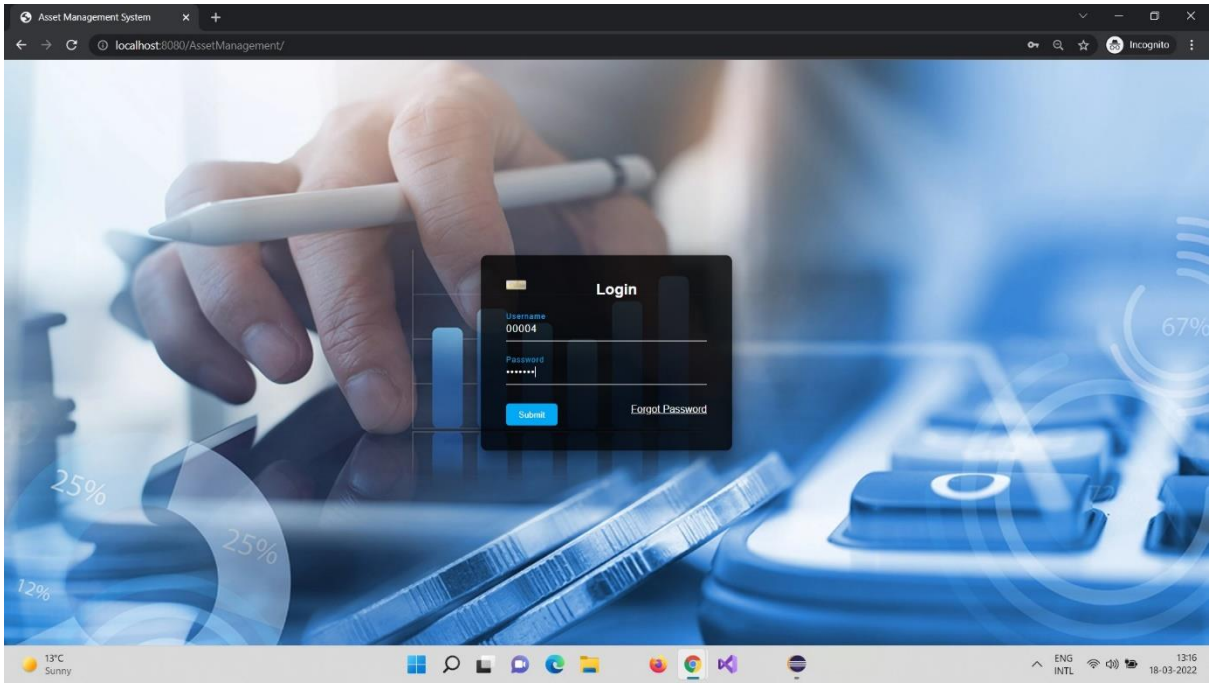
3.6 MySQL Code



- ✓ The above SQL code explains about the username, id and rest other information.
- ✓ If the query is executed the results will be displayed.

4 Prototypes

4.1 Login Page

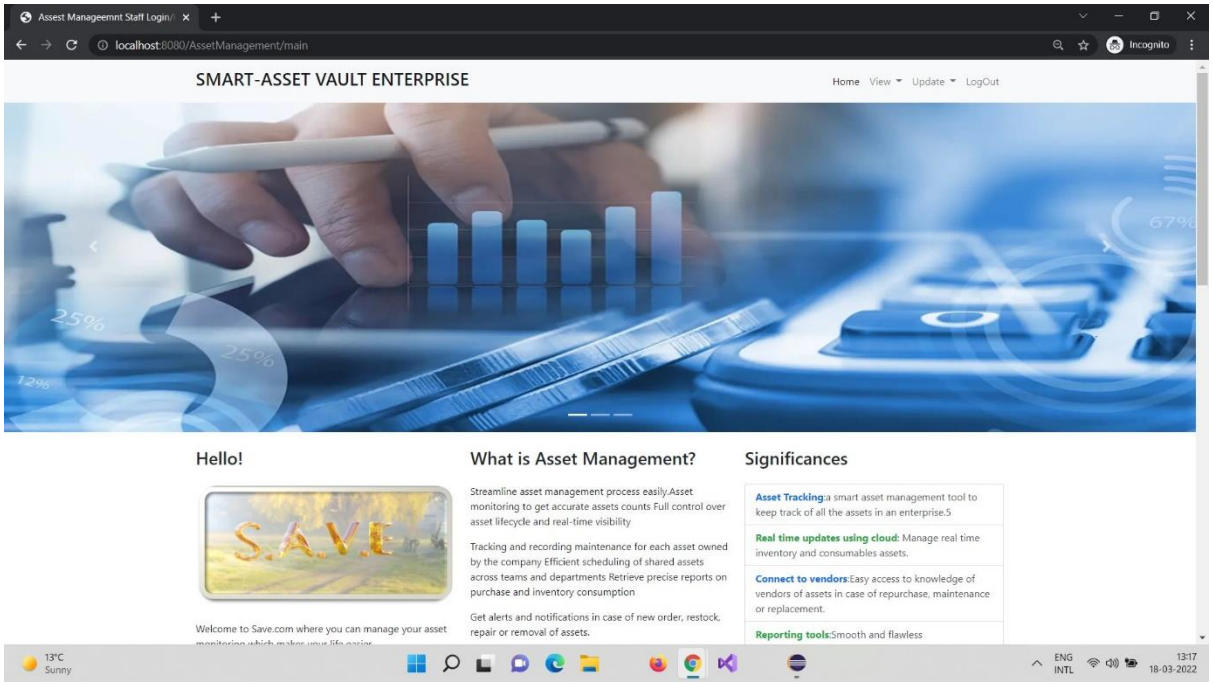


The figure shows the login page of Asset management system. There are three users who can login to the asset management system.

- ✓ The admin
- ✓ The Auditor
- ✓ The Staff

The users can login to the system using their credentials. Once the user login to the system, the actions on the home page is different for Admin, Auditor and Staff. The actions which can be performed by the Staff is less compared to Admin and Auditor.

4.2 Home Page

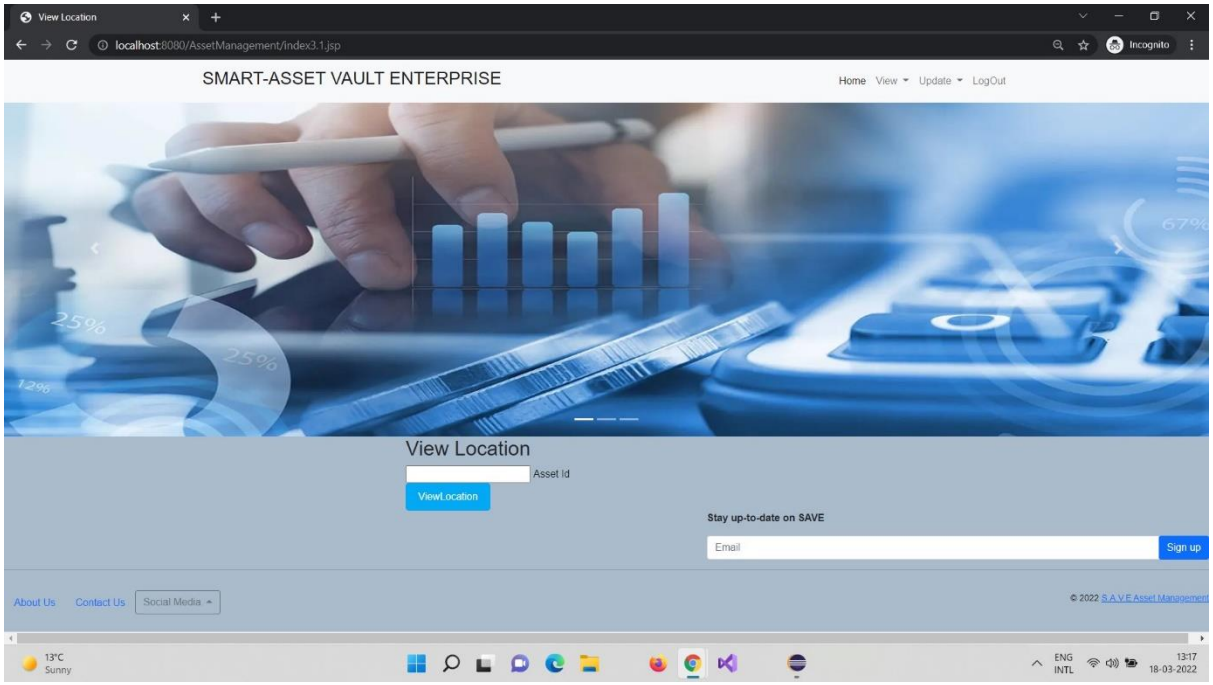


The figure shows the home page of Asset management System. Once the admin login to the system,

The admin can see different tabs which include

1. Home Tab: It displays the basic idea of what is Asset Management? the Logo of the project, and the significance of Asset Management.
2. View Tab: It displays the location of the asset ID after entering the AssetID.
3. Update: It displays a form to update the asset details.
4. Logout

4.3 Draft View Location Page



The figure shows the View Location page of Asset management System. When admin click on the view tab it displays the location of the asset ID after entering the Asset ID. Based on the location of asset the admin can change or update the location of the asset using the update tab.

5 Testing

5.1 Login Test Case

Test Case ID	Test Scenario	Test Steps	Expected Results	Expected Results	Pass/Fail
TC_LOG_1	Login with valid data	1. Enter username 2. Enter password 3. Click Submit button	User should login successfully and redirected to home page	As Expected,	Pass
TC_LOG_2	Login with invalid data	1. Enter invalid username 2. Enter invalid password 3. Click Submit button	User could not login. ‘Enter the valid details' error message will display	As Expected,	Pass
TC_LOG_3	Login without username and password	1. Do not enter username 2. Do not enter password 3. Click on Submit button	User could not login. ‘Enter the valid details' error message will display	As Expected,	Pass
TC_LOG_4	Login with valid username and invalid password	1. Enter username 2. Enter invalid password 3. Click on Submit button	User could not login. ‘Enter the valid details' error message will display	As Expected,	Pass
TC_LOG_5	Login with invalid username and valid password	1. Enter invalid username 2. Enter valid password 3. Click on Submit button	User could not login. ‘Enter the valid details' error message will display	As Expected,	Pass

5.2 View Test Case

Test Case ID	Test Scenario	Test Steps	Expected Results	Expected Results	Pass/Fail
TC_STATUS_1	Enter with valid Asset Id	1. Enter a valid asset ID 2. Click on view status button	Display message showing status: working	As Expected,	Pass
TC_STATUS_2	Enter with invalid Asset Id	1. Enter invalid asset ID	Show warning message showing status: invalid	As Expected,	Pass
TC_STATUS_3	Enter without asset ID	1. Do not enter any asset ID 2. Click on view status button	The pointer shows to the asset field showing "This field is required"	As Expected,	Pass

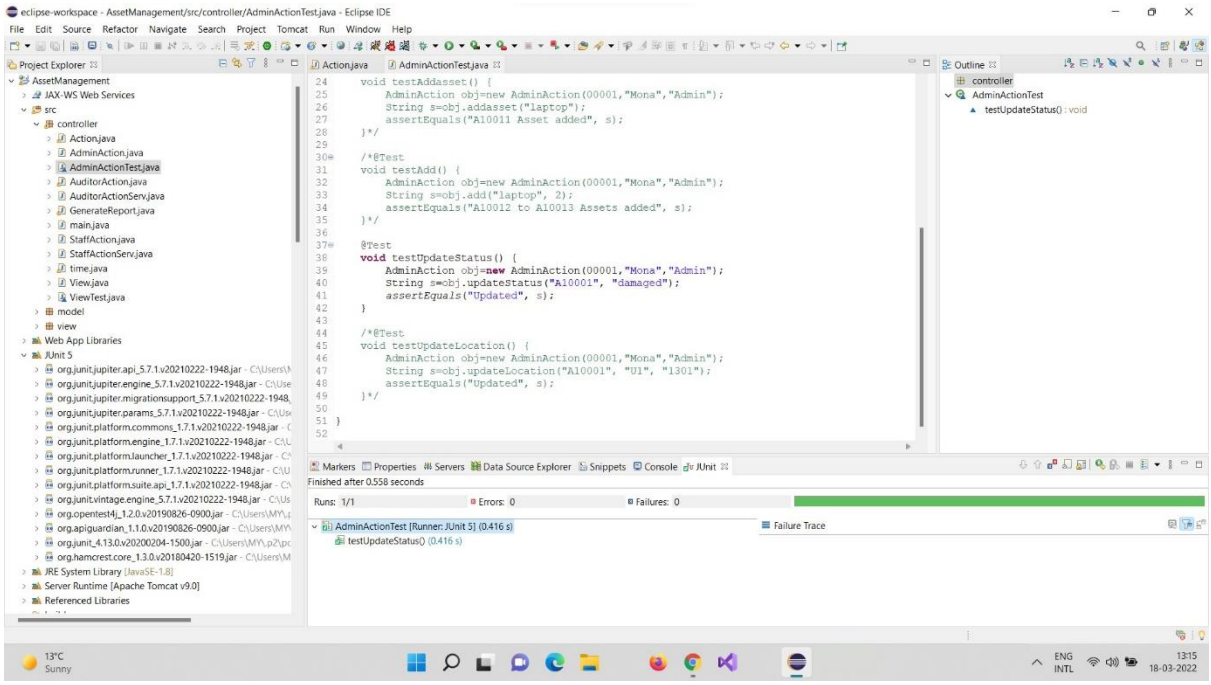
5.3 Update Test Case

Test Case ID	Test Scenario	Test Steps	Expected Results	Expected Results	Pass/Fail
TC_UPDATE_1	Enter with valid Asset Id and Submit valid Status	1. Enter a valid asset ID 2. Enter new status 3. Click on Update status button	Display message 'Updated'	As Expected,	Pass
TC_UPDATE _2	Enter with valid Asset Id and Submit Invalid Status	1. Enter a valid asset ID 2. Enter invalid status 3. Click on Update status button	Show warning message displaying status: 'error occurred'	As Expected,	Pass
TC_UPDATE_3	Enter with Invalid Asset Id and Submit valid Status	1. Enter a valid asset ID 2. Enter invalid status 3. Click on Update status button	Show warning message displaying status: 'error occurred'	As Expected,	Pass

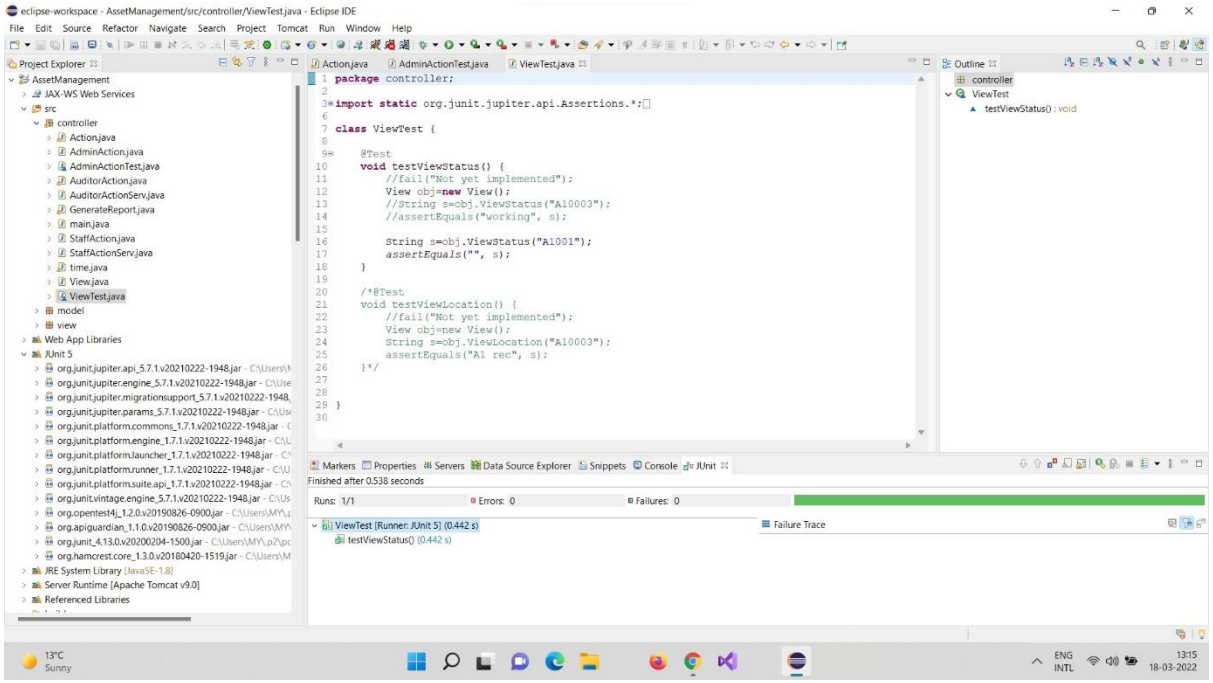
TC_UPDATE_4	Enter with valid Asset Id and without entering a status.	1. Enter a valid asset ID 2. Leave status field empty 3. Click on Update status button	Show warning message displaying status: 'error occurred'	As Expected,	Pass

6 Junit Testing using Jupiter API

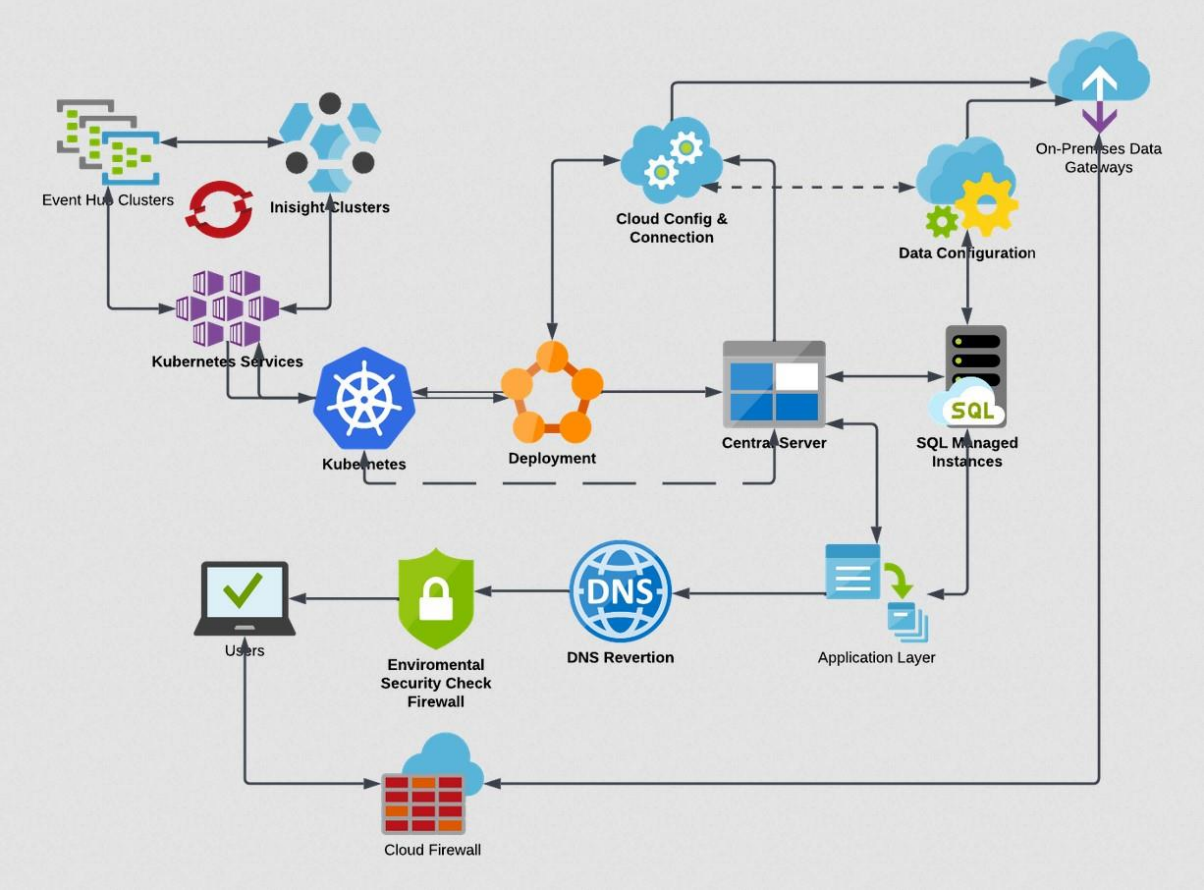
6.1 Admin Action Test



6.2 View Test


eclipse-workspace - AssetManagement/src/controller/ViewTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Project Explorer
AssetManagement
JAX-WS Web Services
src
controller
Action.java
AdminAction.java
AdminActionTest.java
AuditorAction.java
AuditorActionServ.java
GenerateReport.java
main.java
StaffAction.java
StaffActionServ.java
time.java
View.java
ViewTest.java
model
view
Web App Libraries
JUnit 5
JRE System Library [JavaSE-1.8]
Server Runtime [Apache Tomcat v9.0]
Referenced Libraries
Action.java
AdminActionTest.java
ViewTest.java
1 package controller;
3 import static org.junit.jupiter.api.Assertions.*;
7 class ViewTest {
9 @Test
10 void testViewStatus() {
11 //fail("Not yet implemented");
12 View obj=new View();
13 //String s=obj.ViewStatus("A10003");
14 //assertEquals("working", s);
16 String s=obj.ViewStatus("A1001");
17 assertEquals("", s);
18 }
20 /*@Test
21 void testViewLocation() {
22 //fail("Not yet implemented");
23 View obj=new View();
24 String s=obj.ViewLocation("A10003");
25 assertEquals("A1 rec", s);
26 }*/
29 }
Outline
controller
ViewTest
testViewStatus() : void
Markers Properties Servers Data Source Explorer Snippets Console JUnit
Finished after 0.538 seconds
Runs: 1/1
Errors: 0
Failures: 0
ViewTest [Runner: JUnit 5] (0.442 s)
testViewStatus() (0.442 s)
Failure Trace
13°C
Sunny
ENG
INTL
13:15
18-03-2022

7 Deployment Diagram

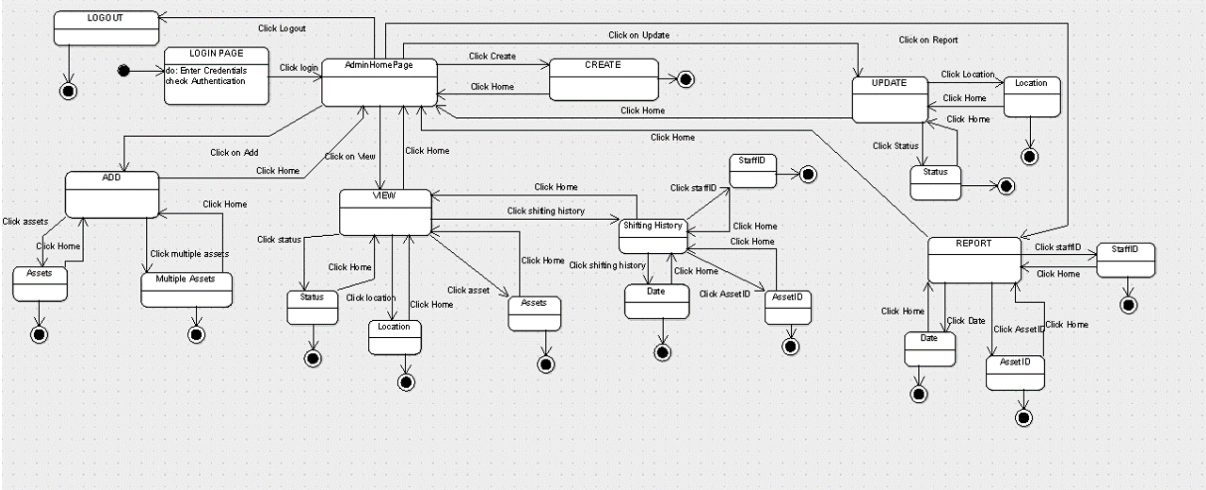


The internal relationship between software, hardware as well as the could platform is demonstrated by the deployment diagram. The diagram is used to deploy the system onto cloud server/platform; hence the figures represent the essential components of the project.

For the Cloud configuration, Kubernetes configuration is used to keep the server flawless. The data servers are spread into insight clusters and the pager /process session ID dictionaries servers are spread into Event Hub Cluster. Where both clusters can communicate and are connected by Kubernetes Services API from Kubernetes MVT controller (Model, View, Template). The central server gathers data from both SQL managed instances (servers) and the Cloud and transmit the data into CI/CD pipeline before sending to the Kubernetes. The Cloud Configuration and connection is an API that monitors and controls the dataflow from the central servers to the deployment pipeline. The Cloud Data Configuration will then control the data uploaded from SQL server to Cloud.

The application then will retrieve requested data from both SQL data servers and the central server upon users' requests. Through a DNS reversion and a firewall, the data packet is then safely handed to the end users. Alternatively, if the user deemed to store / retrieve data from Cloud directly, the On-premises cloud gateway will process the request and retrieve the requested data from the cloud platform, then go through the firewall and hand the data to the end user.

8 State Diagram



Initial state is being shown by starting with a black dot. Final State is being shown by the black dot surrounded by an empty circle.

9 Execution Architecture

Runtime environment required is any device supporting Java and an OS of Windows 7 or later versions. The whole project was built using the MVC architecture in OOPs for a better code reusability as well as better understanding. Testing and development of code is easier and can be done any non-member of the group easily jus y reading this document as well as Software Requirements document provided earlier.

10 Pseudo Codes

10.1 Create ():

1. Make connection with database

2. Sql1="Select prod_id from products;”

3. PreparedStatement ps=(PreparedStatement) con.prepareStatement(sql1);

4. ResultSet rs=ps.executeQuery();

5. id=rs.getString(1);

6. id=id+1;

7. Execute the following statement to store the new product details in database 8. Insert into products values(id,name,category);

10.2 ViewStatus ():

1. Make connection with database

2. Take asset_id1 as input

3. sql1="Select status from assets where asset_id= asset_id1;”

4. rs = ps.executeQuery();

5. While rs.next() is TRUE repeat Step 6

6. status=rs.getString(1);

10.3 ViewLocation ():

1. Make connection with database

2. Take asset_id1 as input

3. sql1="Select location and room from assets where asset_id= asset_id1;”

4. rs = ps.executeQuery();

5. While rs.next() is TRUE repeat Step 6

6. location=rs.getString(1)+" "+rs.getString(2);

10.4 ViewAsset ():

1. Make connection with database
2. Take asset_id1 as input
3. sql1="Select * from assets where asset_id=asset_id1";
4. rs = ps.executeQuery();
5. While rs.next() is TRUE repeat Step 6
6. asset=rs.getString(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getString(4)+" "+rs.getString(5)+" "+rs.getString(6)+" "+rs.getTimestamp(7);

10.5 Remove ():

1. Make connection with database
2. Take asset_id
3. Sql=" Remove assets where assetId=asset_id;"
4. Executing the Sql statement will remove the asset from database.

10.6 Logout ():

1. Get session object from request
2. Session.invalidate();

10.7 Login ():

1. Make connection with database
2. Take user_name and password as input

3. sql="Select * from UserN where UserID=user_name and password=aes_encrypt(password,key);”

4. rs = ps.executeQuery();

5. While rs.next() is TRUE Repeat Steps 6 to Step 11

6. actor=rs.getString(2);

7. name=rs.getString(3);

8. if(actor==null)

9. Print(“username or password is wrong”);

10.Else

11.Print(“Successful login”);

11 Data Dictionary

Data Dictionary plays a crucial role in explanation of the technicality of the data and information which are the main components of any system. The website planned for Automation of Lab Squad gathers a lot of data from its users which will be manipulated at backend and ER Diagrams a pictorial demonstration for the same, but to represent the business value out of it, Data Dictionary comes into play.

11.1 Actor

FIELD NAME	DATA TYPE	DATA SIZE	DESCRIPTION	NULL VALUE	REQUIRED	DEFAULT
ID	CHAR	3	Id of the actor	NO	YES	None
ACT	VARCHAR	20	Primary key	NO	YES	None.

11.2 Status

FIELD NAME	DATA TYPE	DATA SIZE	DESCRIPTION	NULL VALUE	REQUIRED	DEFAULT
TransID	VARCHAR	10	Primary key	NO	YES	None
Prod_id	VARCHAR	4	Product ID of the status	NO	YES	None.
Asset_id	VARCHAR	6	Asset ID of the status	NO	YES	None
status	VARCHAR	15	Status information to show whether the issue has been solved or not	NO	YES	None.
UserID	CHAR	5	Status user ID	NO	YES	None
time	DATETIME	255	Date and time choose to resolve the issue	NO	YES	None.

11.3 Assets

FIELD NAME	DATA TYPE	DATA SIZE	DESCRIPTION	NULL VALUE	REQUIRED	DEFAULT
Asset_id	VARCHAR	6	Primary key	NO	YES	None
Prod_id	VARCHAR	4	ID of the assest	NO	YES	None.
Location	VARCHAR	10	Location of asset	NO	YES	None
room	VARCHAR	10	To indicate which room the assets are in.	NO	YES	None.
status	VARCHAR	15	Status information to show whether the issue has been solved or not	NO	YES	None

UserId	CHAR	5	To mention the assets Userid	NO	YES	None.
time	DATETIME	255	Date and time choose to resolve the issue	NO	YES	None

11.4 stat

FIELD NAME	DATA TYPE	DATA SIZE	DESCRIPTION	NULL VALUE	REQUIRED	DEFAULT
status	VARCHAR	15	Primary key	NO	YES	None

11.5 location

FIELD NAME	DATA TYPE	DATA SIZE	DESCRIPTION	NULL VALUE	REQUIRED	DEFAULT
TransID	VARCHAR	10	Primary key	NO	YES	None
Prod_id	VARCHAR	4	Product ID of the status	NO	YES	None
Asset_id	VARCHAR	6	Asset ID of the status	NO	YES	None
tolocation	VARCHAR	10	To location the data is being sent	NO	YES	None
toroom	VARCHAR	10	To which room the data has been sent	NO	YES	None
UserId	CHAR	5	To mention the location Userid	NO	YES	None
time	DATETIME	255	Date and time choose to resolve the issue	NO	YES	None

11.6 usern

FIELD NAME	DATA TYPE	DATA SIZE	DESCRIPTION	NULL VALUE	REQUIRED	DEFAULT
UserID	VARCHAR	5	Primary key	NO	YES	None
Act	VARCHAR	20	Which act will be performed	NO	YES	None

Name	VARCHAR	100	Name of the usern	NO	YES	None
password	VARCHAR	100	To protect the data password is used.	NO	YES	None

11.7 Products

FIELD NAME	DATA TYPE	DATA SIZE	DESCRIPTION	NULL VALUE	REQUIRED	DEFAULT
Prod_id	VARCHAR	4	Primary key	NO	YES	None
Product	VARCHAR	20	Products that are displayed.	NO	YES	None
category	VARCHAR	20	Categories of the product to identify	NO	YES	None

11.8 Loc

FIELD NAME	DATA TYPE	DATA SIZE	DESCRIPTION	NULL VALUE	REQUIRED	DEFAULT
location	VARCHAR	10	Primary key	NO	YES	None