**Group members**

Dang Thi Thao My - 15905067

Bui Nguyen Thien Khanh - 15907679

Nguyen Anh Quan - 15905072

Le Huynh Anh Tuan - 1325907

**Supervisor:** MSc. Truong Phuoc Loc

**Product Owner:** Mr. To Hoa Duy Man

**System Design**

**Performance testing dashboard**

Version 1.0 – Released date: 11/10/2016

**Status**:

Approved by:

Released by: Capstone team (ETI1)

Internal

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Changed by | Modifications |
| 1.0 | 11.10.2016 | Team | Initial version |
|  |  |  |  |

List of Abbreviations

CCR **C**ritical **C**omputer **R**esources  
CI **C**onfiguration **I**tem  
CM **C**onfiguration **M**anagement  
HLD **H**igh **L**evel **D**esign

Contents

[1. Introduction 1](#_Toc482697494)

[1.1. About Apache Jmeter 1](#_Toc482697495)

[1.2. About testing management 2](#_Toc482697496)

[1.3. Testing management and Jmeter in one 2](#_Toc482697497)

[2. Objectives and features 4](#_Toc482697498)

[2.1. Benefits 4](#_Toc482697499)

[2.2. Features 4](#_Toc482697500)

[2.3. Functional Scope 5](#_Toc482697501)

[3. System architecture diagram 6](#_Toc482697502)

[4. UML diagrams 7](#_Toc482697503)

[4.1. User case for administrators 7](#_Toc482697504)

[4.2. Activity diagram for administrators 8](#_Toc482697505)

[4.3. Sequence diagram for administrators 10](#_Toc482697506)

[4.4. Use case for testers 11](#_Toc482697507)

[4.5. Activity diagram for testers 12](#_Toc482697508)

[4.6. Sequence diagram for testers 16](#_Toc482697509)

[5. System design 17](#_Toc482697510)

[5.1. Database schema 17](#_Toc482697511)

[5.2. Dashboard sitemap 18](#_Toc482697512)

# Introduction

The objective of Performance Testing Dashboard for the RBVH’s Validation Team is to create an integrated system for managing and executing performance test cases using Jmeter as the test engine.

Performance Testing Dashboard is a web-based solution for testers to manage and keep track of test results and scripts. It is also a management tool for all testers to handle report and share to customers.

Therefore, the objectives of this project are as follows:

* **Objective 1:** The PTD need to have ability to show performance testing result both in offline-mode and in real-time.
* **Objective 2**: Have the ability to zoom in and zoom out, show the value at the specific point on graph when test execution is complete.
* **Objective 3:** Need to be able to manage the Report. Have some features such as report comparison, report export, etc.
* **Objective 4:** Need a user management which can be linked with Bosch user and also able to create new user.
* **Objective 5:** Need a Script management which use to manage performance script. It also need a File Uploader where user can upload their Script.
* **Objective 6:** The P.T.D also require an automation system to co-operate with. This automation system will have ability to automate run performance test from the Jmx script which user can upload from dashboard. It also have ability to automate collect performance testing report, depending on the requirement of the customer then it automate send the result back to the customer when test execution is complete.

## About Apache Jmeter

Jmeter is a desktop application used for measuring and analyzing the performance of various services focusing on web applications. Jmeter is developed and maintained by Apache Software Foundation, a non-profit organization. Jmeter is provided as an open-source application with large supporting community.

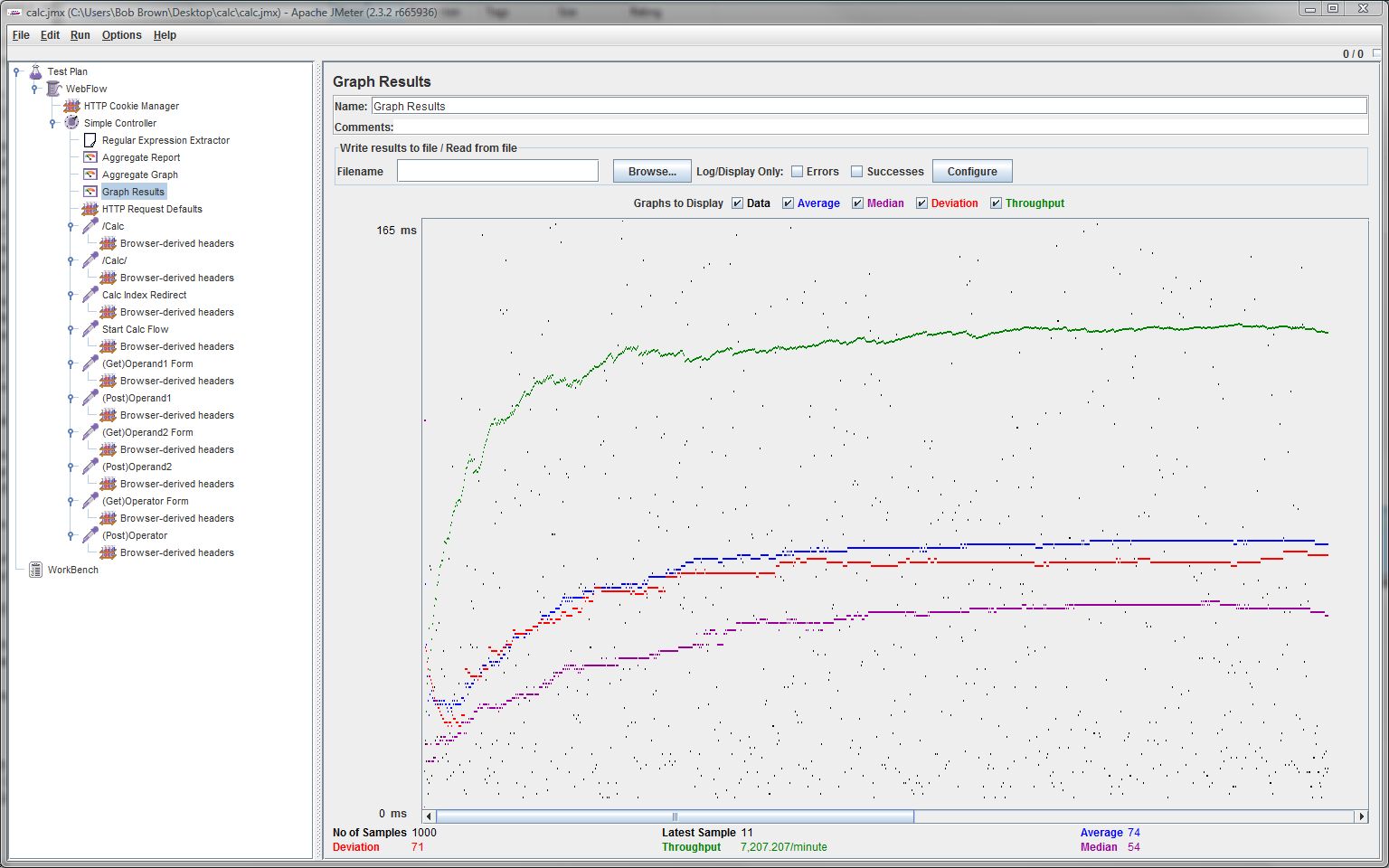


Figure 1 Jmeter application interface

## About testing management

Testing management is an essential process for managing test cases, test results and collaboration among testers of a QC team. It involves using technology to configure and automate testing processes. Testing management dashboard can help store all the testing script and testing results, and access them easily from different locations.

## Testing management and Jmeter in one

Jmeter is a widely used tool for performance testing especially with small and medium-sized projects. People choose Jmeter for its richness of features and for the various benefits coming with an open source application as opposed to costly commercial solutions. However, Jmeter lacks management features for the test scripts, test results and run history. So, it is hard to manage testing workflow and get an overview of testing status of all test cases.

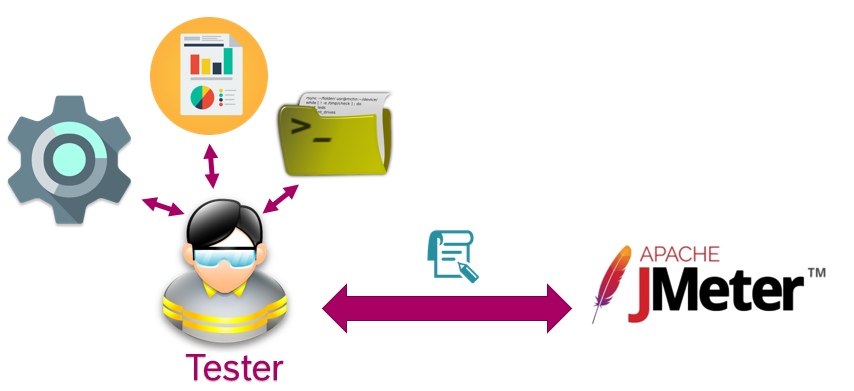


Figure 2 Performance testing process using Jmeter

When testers use Jmter and manage everything on their own, performance testing will become a time and cost consuming process. To overcome this problem and work more effectively, they need a management system which can integrate with Jmeter testing tool. Thus, a performance testing dashboard is the system they want to develop.

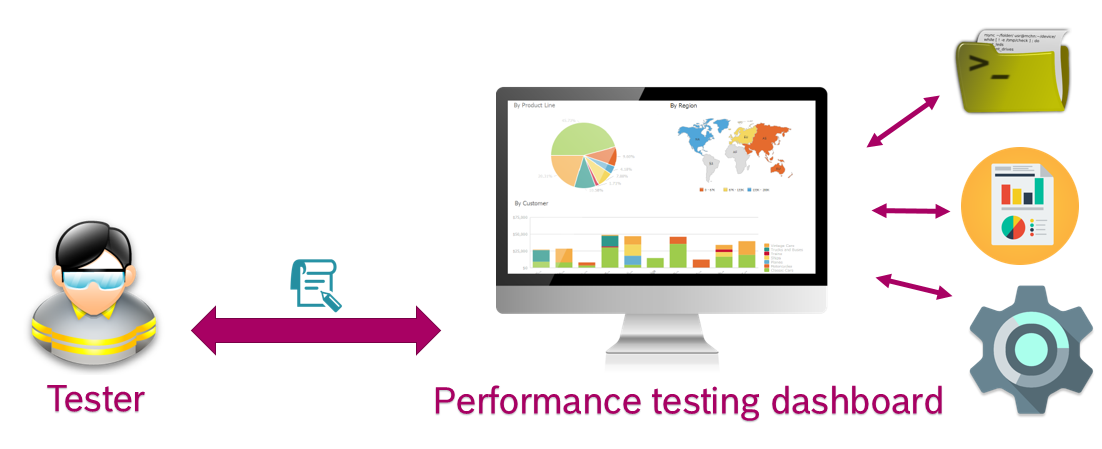


Figure 3 Testing process using Dashboard

# Objectives and features

## Benefits

The Performance testing dashboard integrated with Jmeter provides a variety of benefits to the organization.

## Features

Manage projects: the dashboard keeps track of all test projects and its status. Testers and managers will be able to project details and status at any time.

Authenticate and authorize users by groups: the dashboard provide a mechanism to for user authentication and authorization by assigning users into different groups and setting user enabled statuses.

Manage test scenario with test script: a feature that enables users to upload test scripts to create test scenarios and also modify the test script parameters in a graphical interface.

Integrate with Jmeter for running test scripts: the dashboard becomes a connection point with Jmeter. This allows users run test scripts directly in the dashboard board interface without caring about Jmeter installation on their computers.

Manage distributed system of multiple computers: the dashboard server plays the role as a master of an agent system of many computers. This is an enhancement for Jmeter that allows testers to increase the number of virtual users in performance testing to meet performance requirements of any applications under testing.

Track real-time results while tests are running and runtime history: this feature helps testers watch the performance results in real-time with graphical interface and analysis tools and view runtime history of each test scenario. Multiple users of the dashboard can watch the same test results at the same time by accessing to the result link, making the sharing so much easier than before.

## Functional Scope

|  |  |
| --- | --- |
| User | Task |
| Admin | * Manage users * Manage user groups * Manage privileges * Manage projects * Manage test scenarios * Manage test results * Manage agents * Manage configurations |
| Testers | * Create and test cases * Upload and edit test scripts * View performance reports in real- time * View complete reports * View runtime history of test cases |

# System architecture diagram

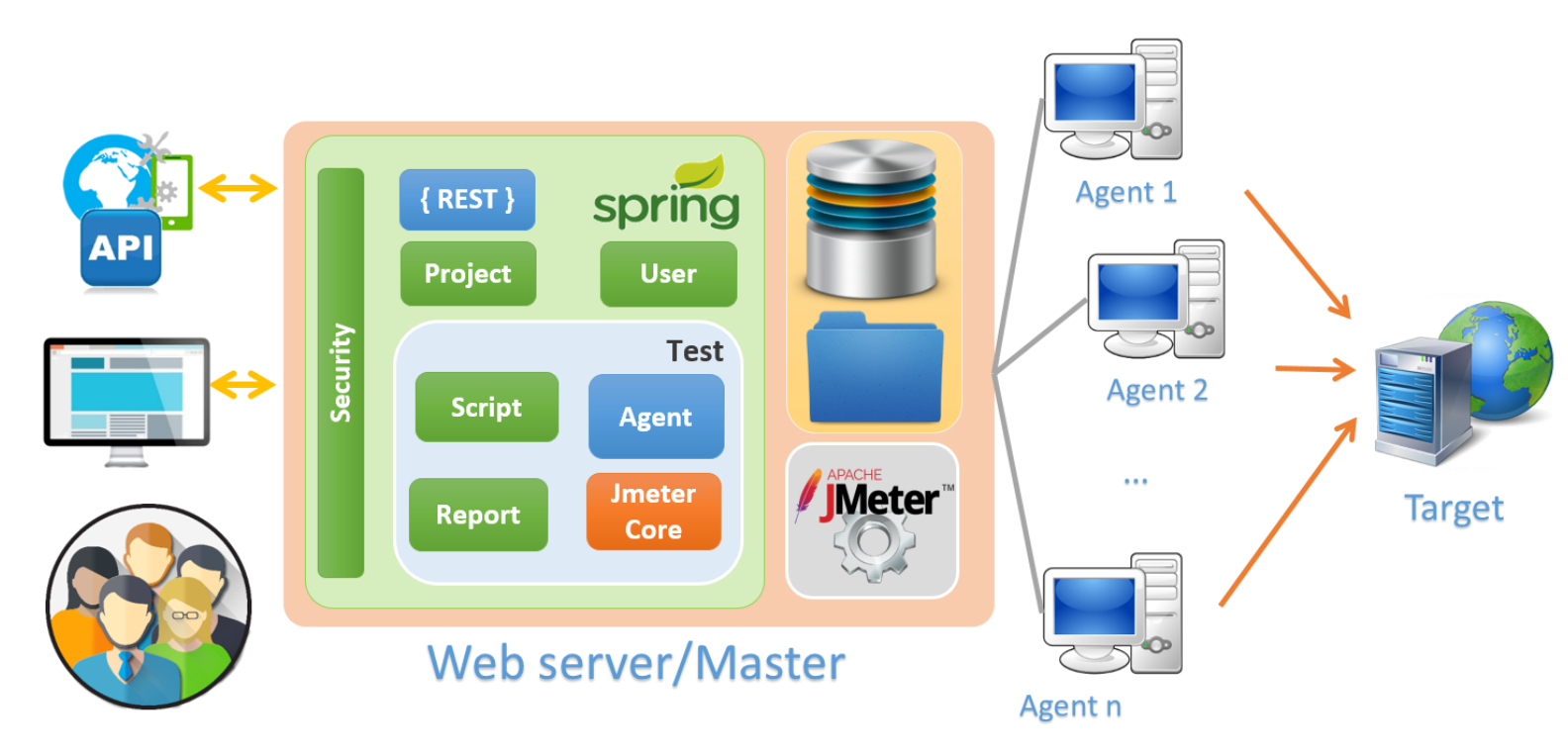


Figure 4 Distributed system architecture

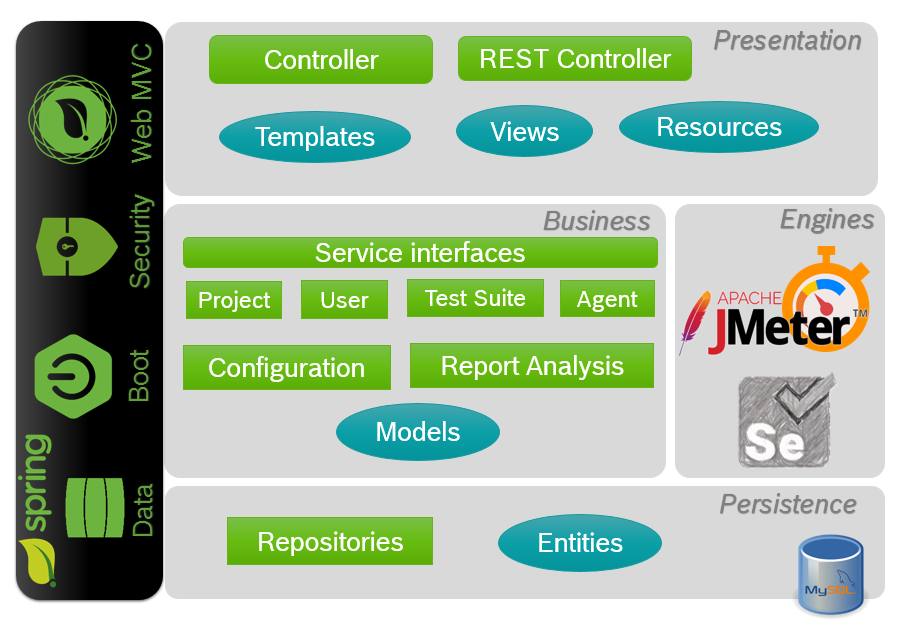


Figure 5 Layer architecture of the dashboard

The distributed system architecture diagram visualizes the basic idea of the whole system from a general perspective. It shows that The Performance Dashboard is divided into two main components, namely, the management component and the Jmeter engine component.

The management component is divided into different modules which represent projects, test suites, agents, users, reports and configurations.

The management component is also designed as three main layers: persistence, business and presentation.

# UML diagrams

## User case for administrators

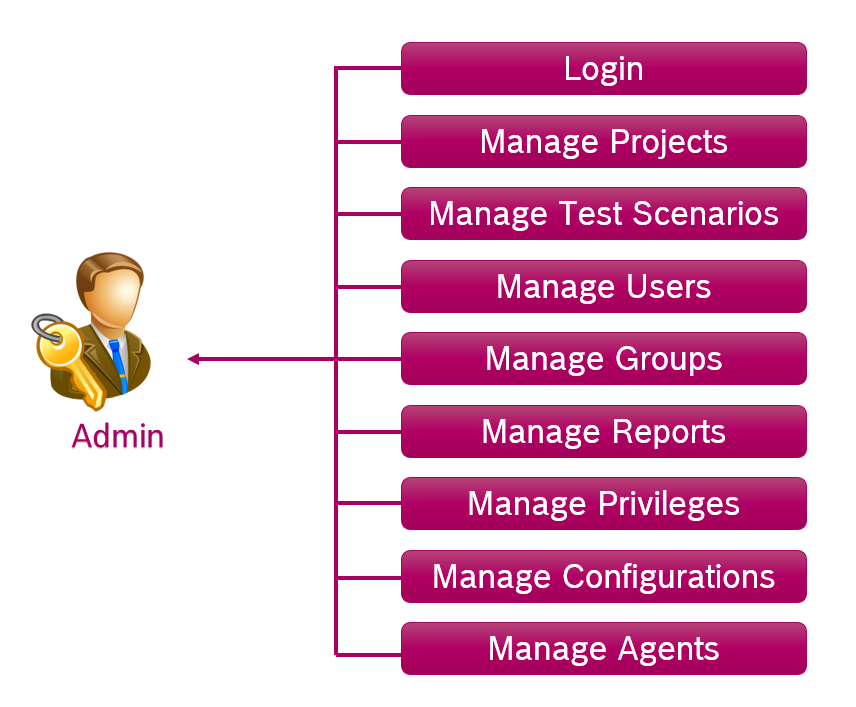


Figure 6 Admin's use cases

## Activity diagram for administrators

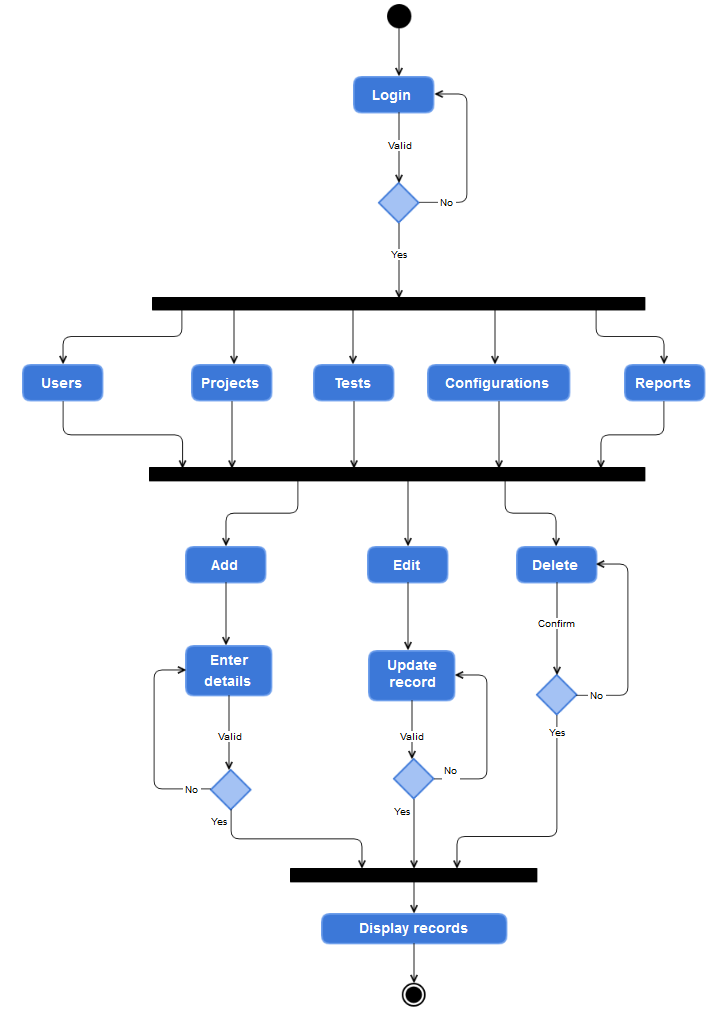


Figure 7 Admin's activities

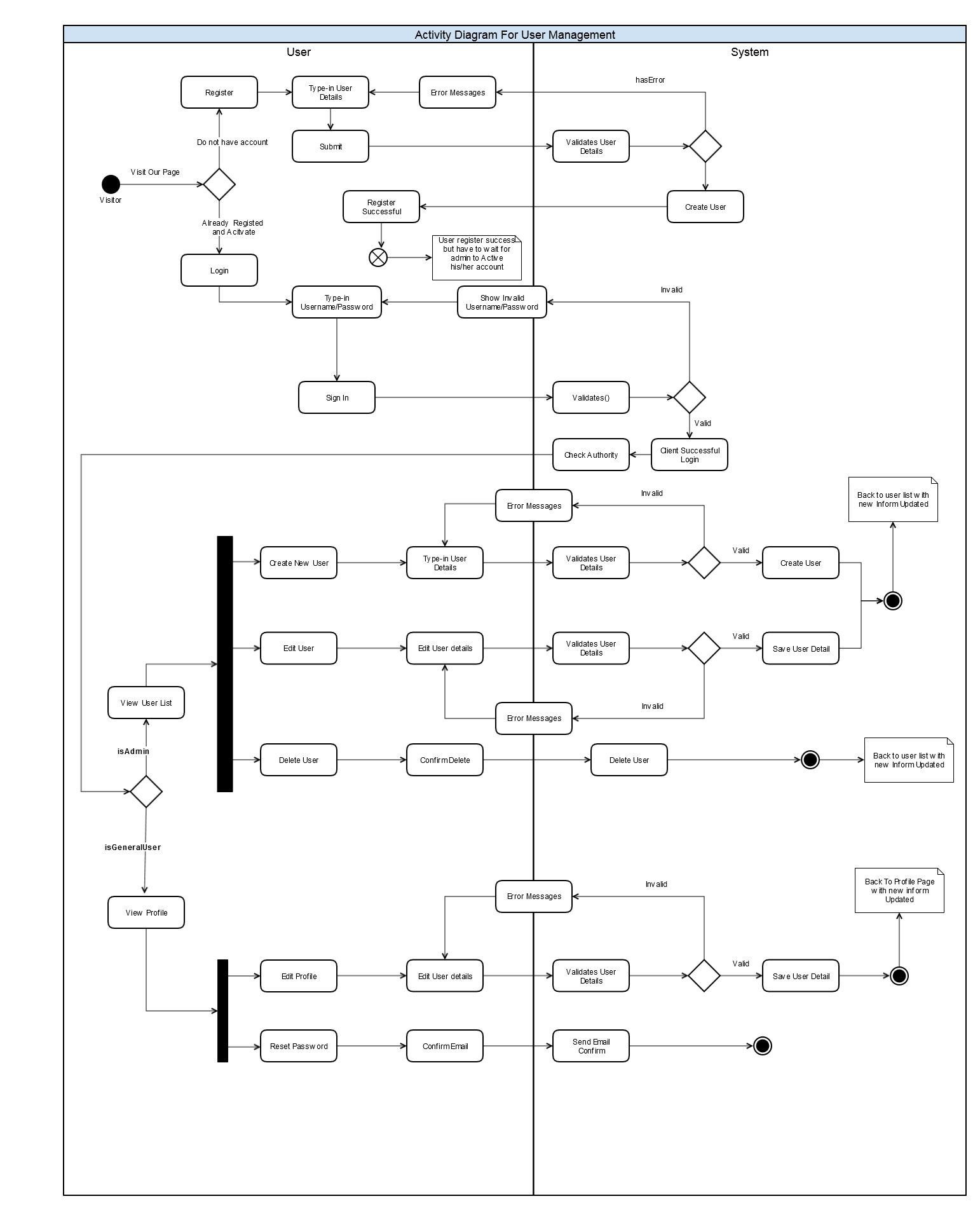


Figure 8 User management activity diagram

## Sequence diagram for administrators

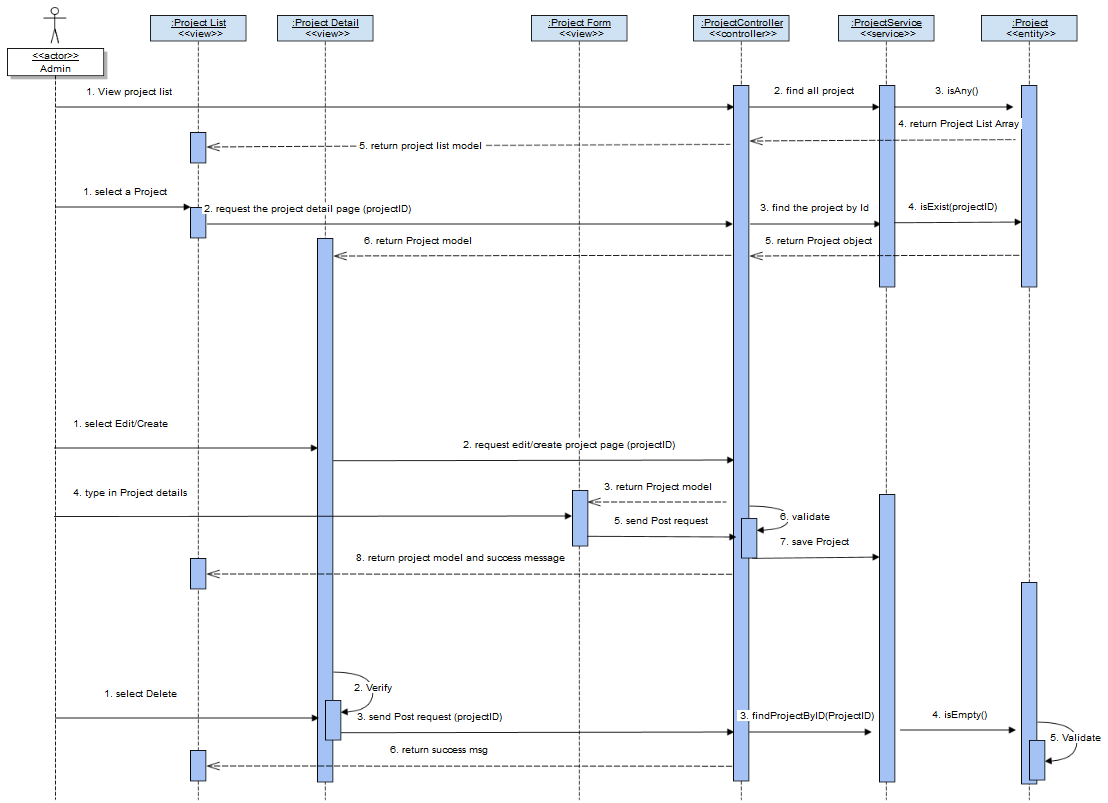


Figure 9 Admin's project management sequence diagram

## Use case for testers

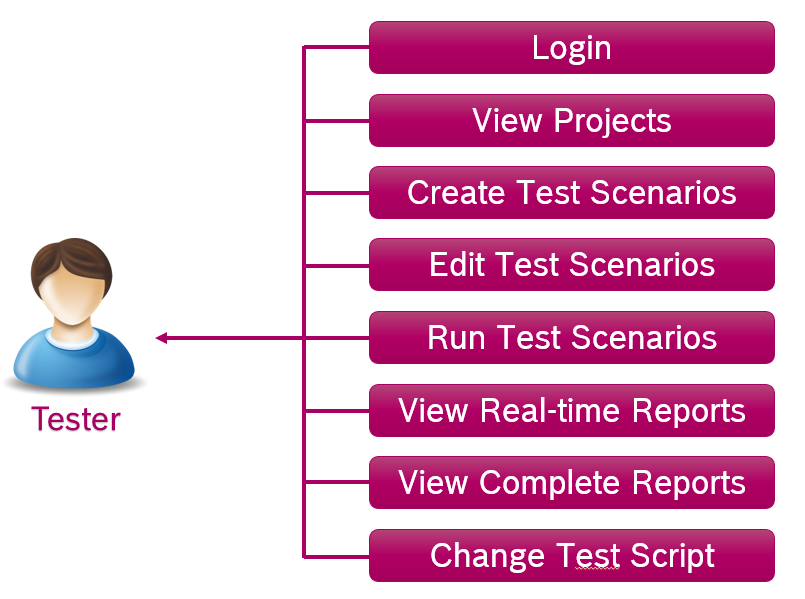


Figure 10 Tester's use cases

## Activity diagram for testers

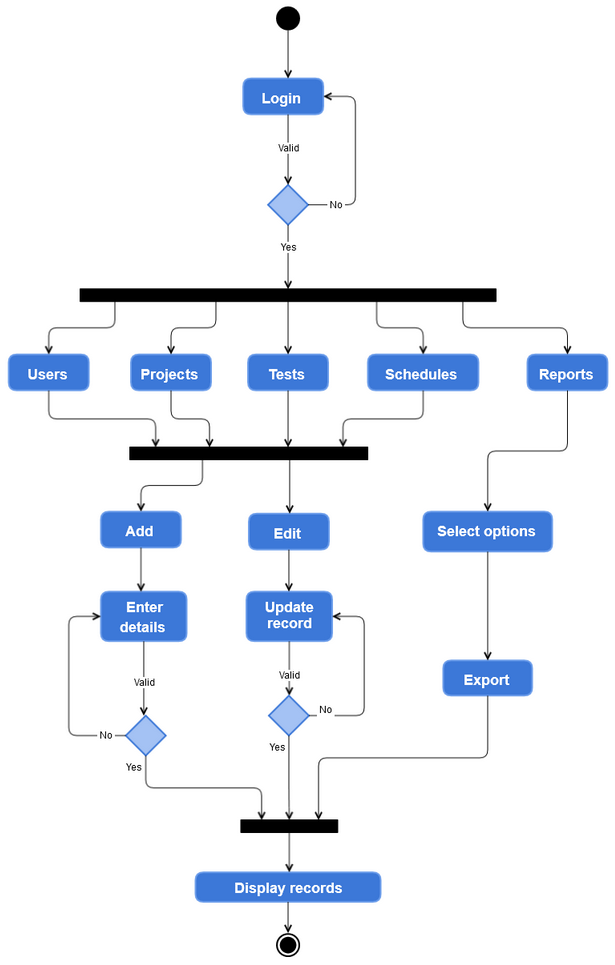


Figure 11 Tester's activities

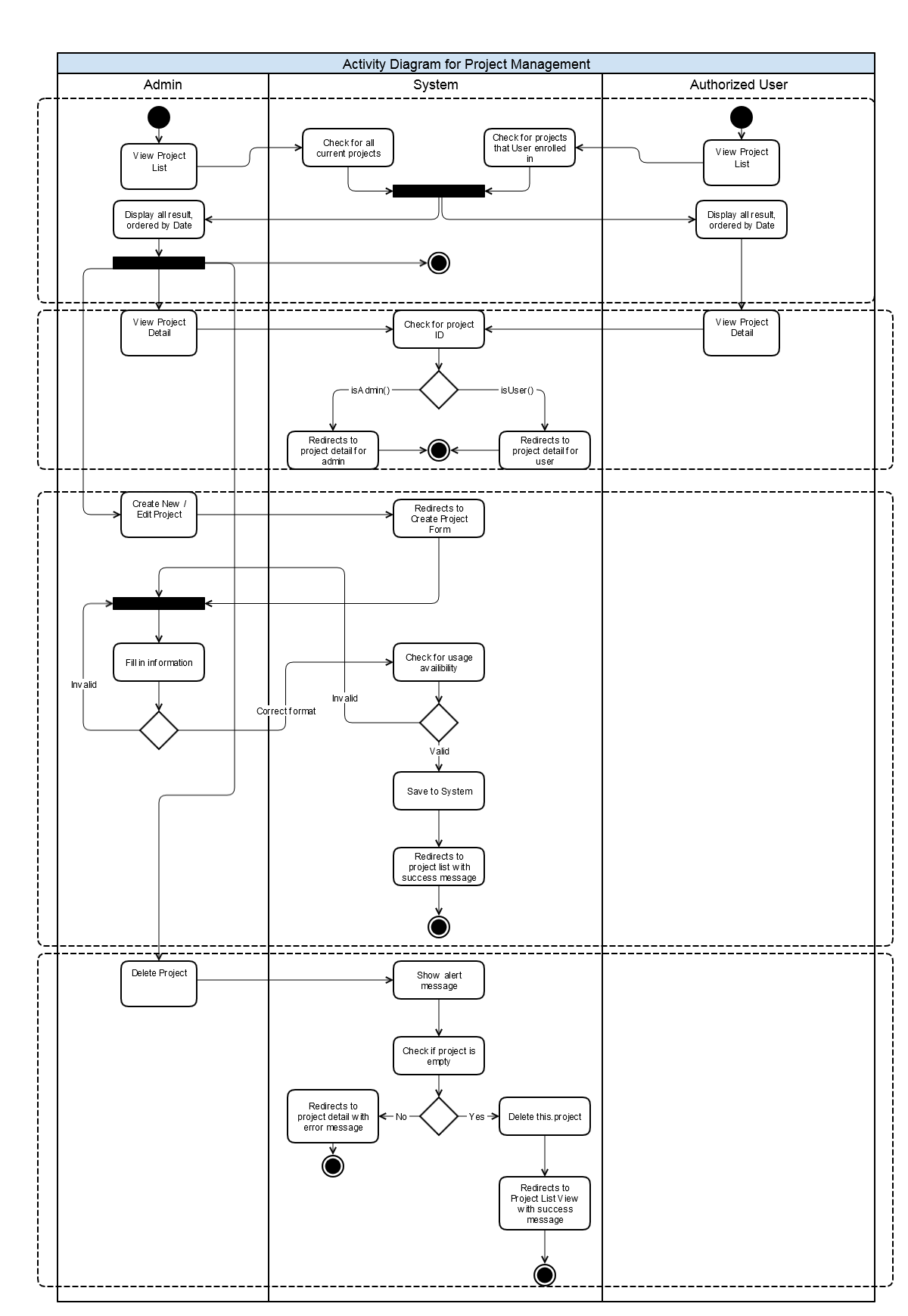


Figure 12 Project management activity diagram

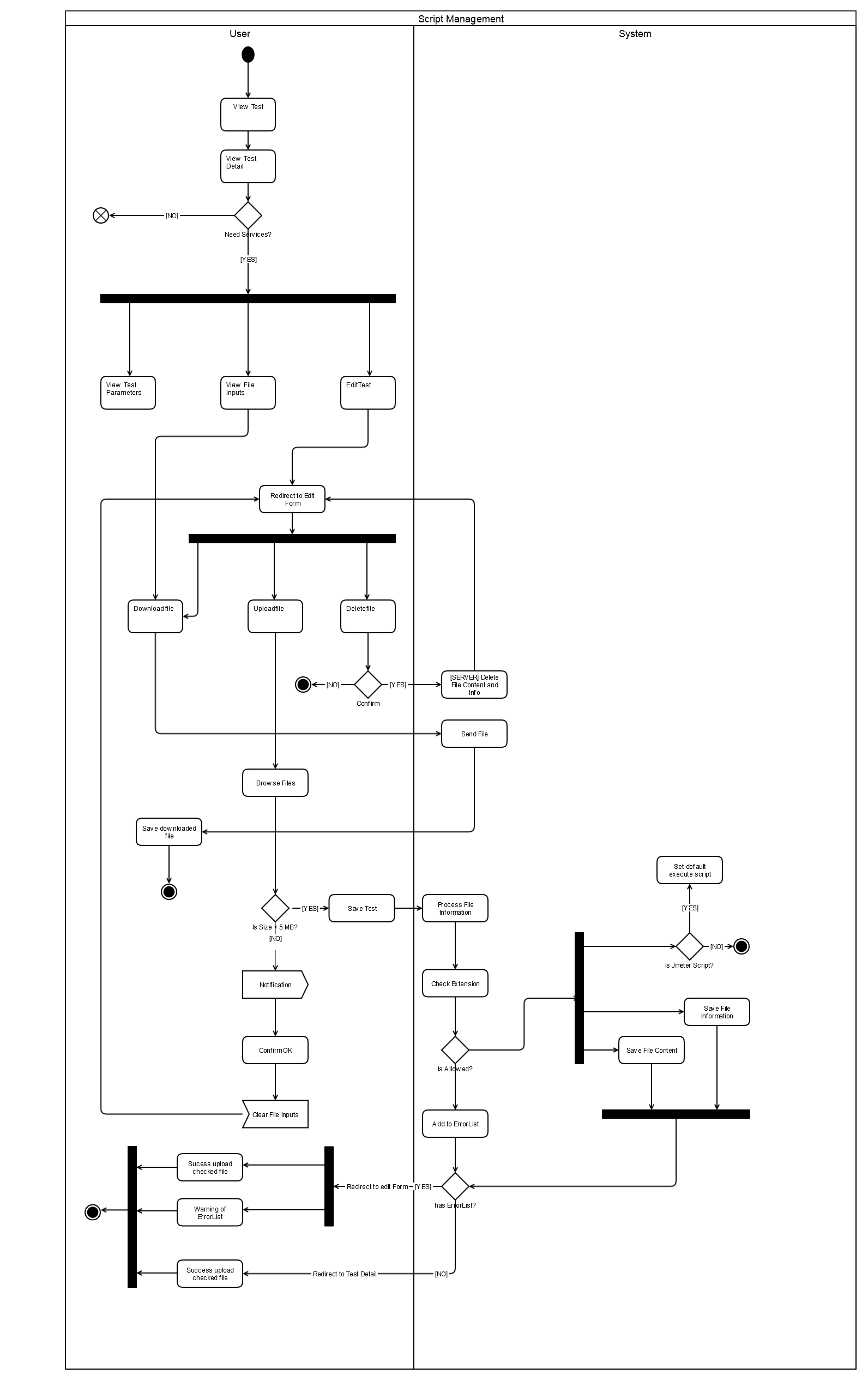


Figure 13 Script management activity diagram

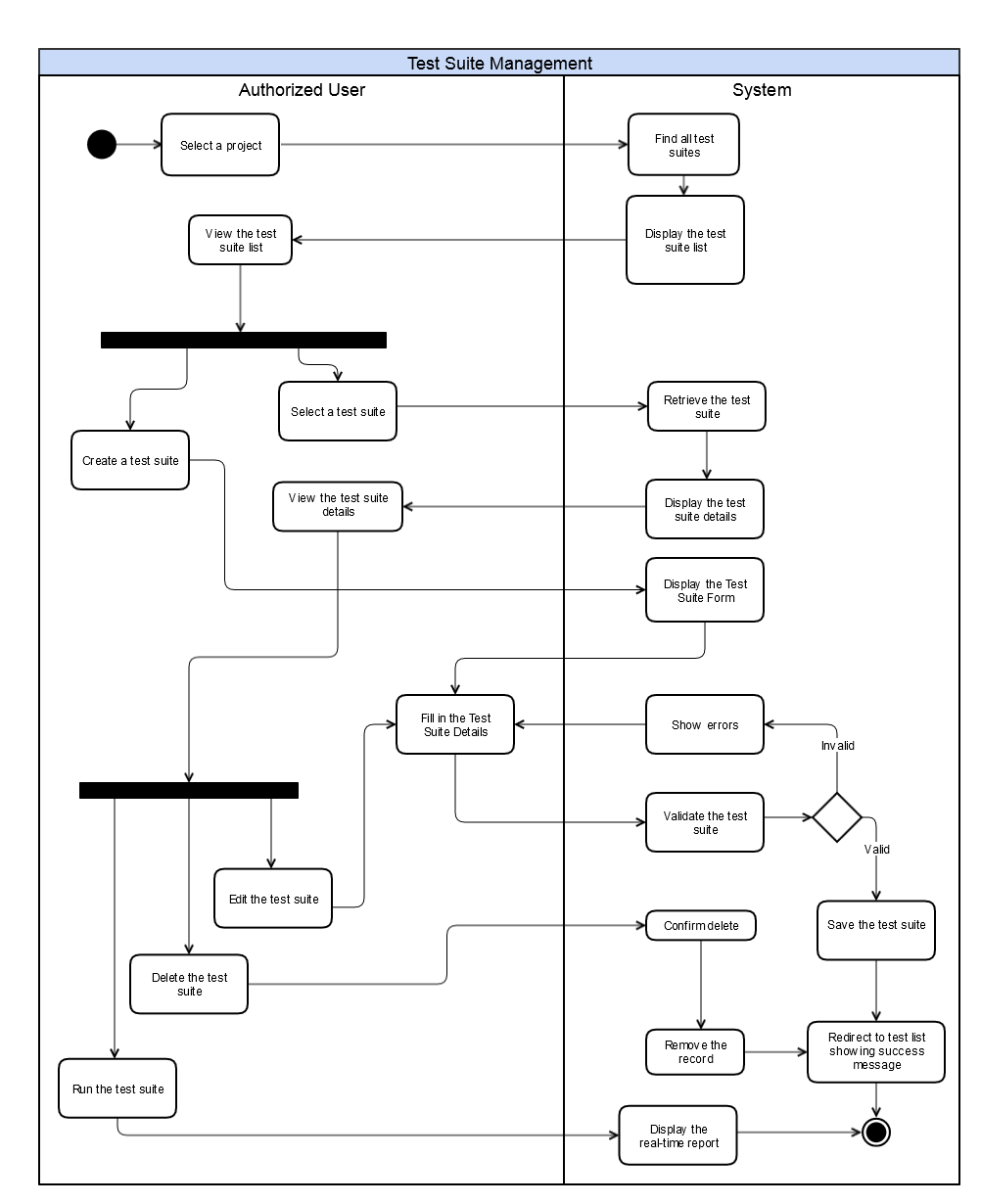


Figure 14 Test Suite management activity diagram

## Sequence diagram for testers

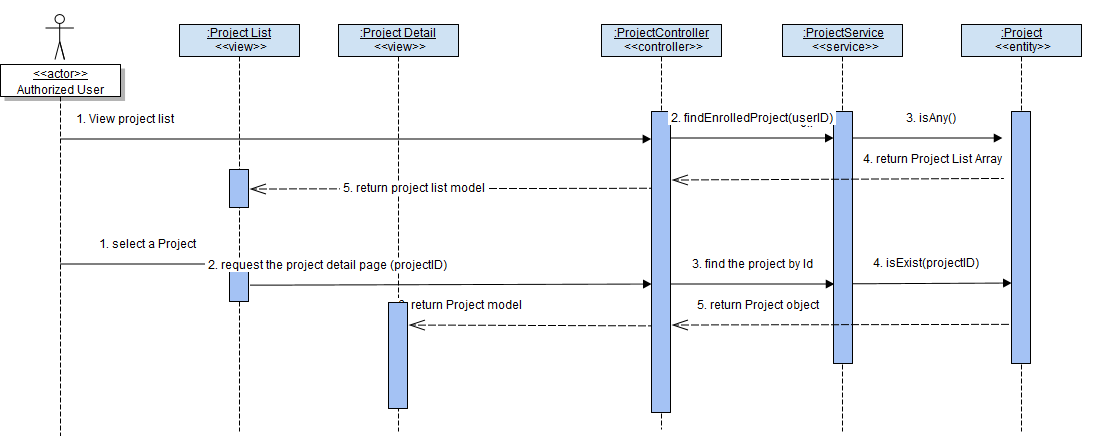


Figure 15 View project sequence diagram

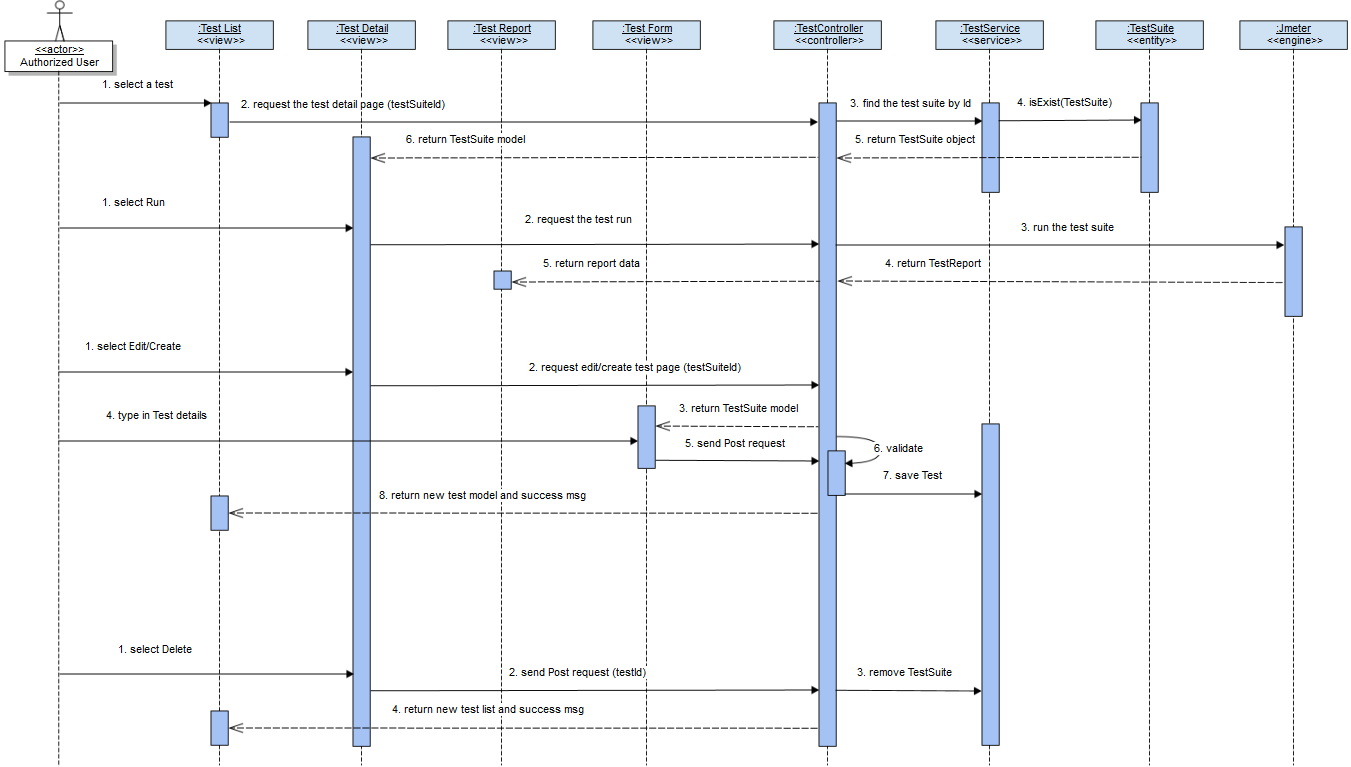


Figure 16 Test management sequence diagram

# System design

## Database schema

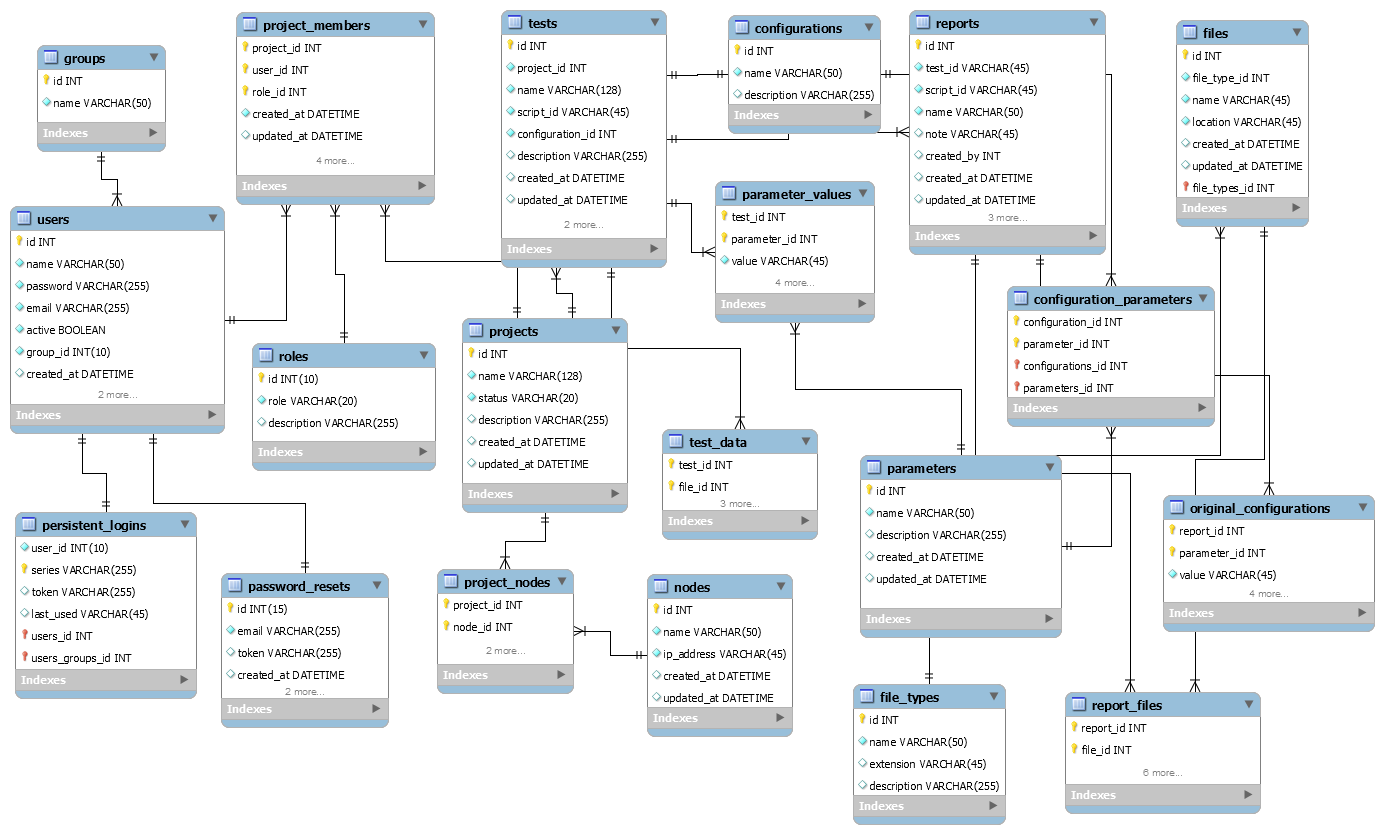


Figure 17 The Dashboard database schema

|  |  |
| --- | --- |
| Tables | Description |
| Groups | Group of users of the system |
| Users | User information |
| Persistent\_logins | Store remember me credentials |
| Password\_resets | Store reset password credentials |
| Project\_members | Members of projects |
| Tests | Test suite information |
| Roles | User roles |
| Projects | Project information |
| Project\_nodes | Link agent with projects |
| Nodes | Agent information |
| Test\_data | Data file of test suites |
| Parameter\_values | Values of parameters of test suites |
| Parameters | Parameters of Test suites |
| Reports | Result of test runtimes |
| Files | Script files of test suites |

## Dashboard sitemap

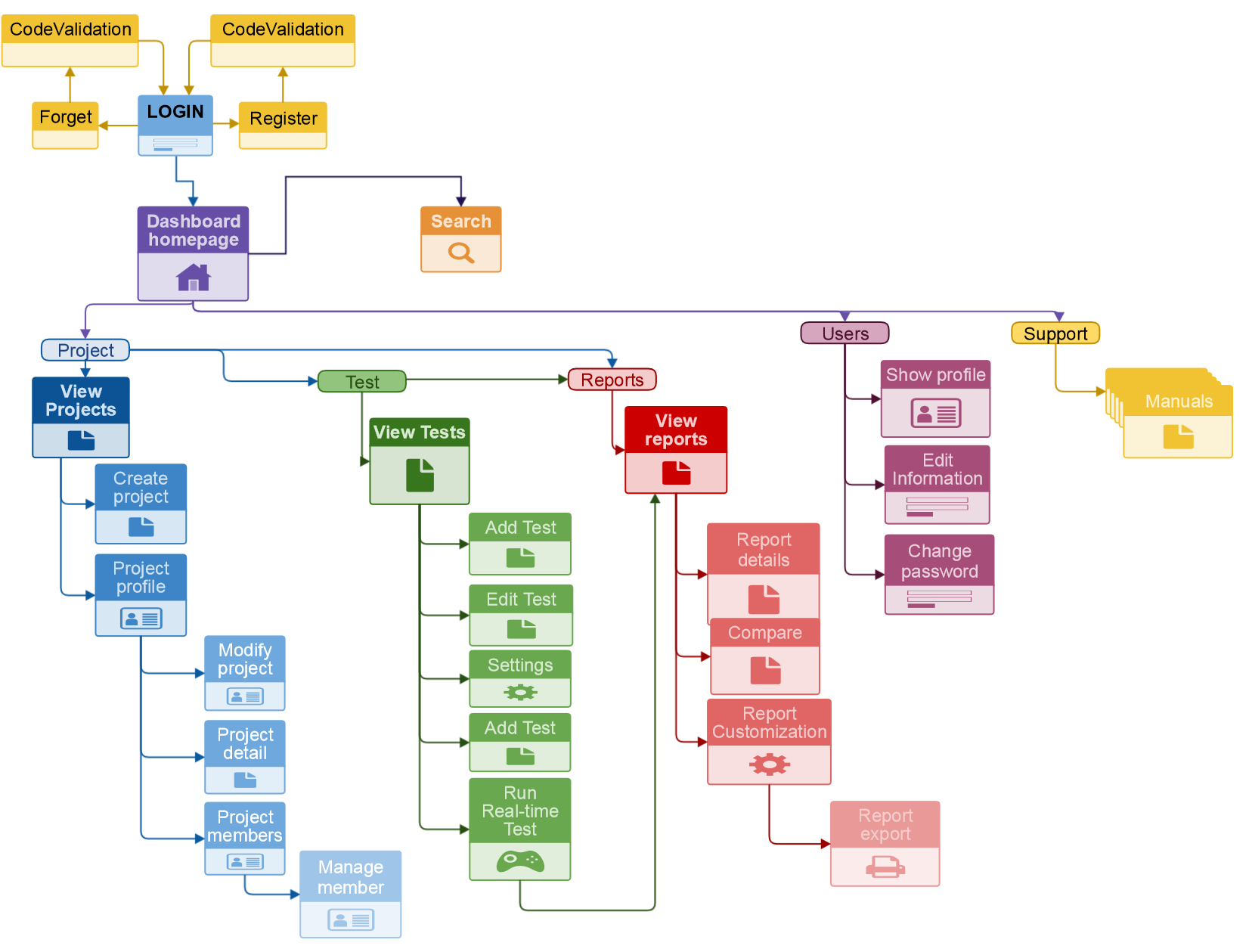


Figure 18 Dashboard sitemap