Table of Contents

[1.0 Introduction 2](#_Toc425678269)

[2.0 System Overview 2](#_Toc425678270)

[3.0 Implementation Overview 4](#_Toc425678271)

[4.0 Module Design Specification 6](#_Toc425678272)

[4.1 Olingo OData API Design 6](#_Toc425678273)

[4.2 User Interface Design 8](#_Toc425678274)

# Introduction

The ATB User Login test project is designed to demonstrate a single user sign-on, where the user name and password should be verified and determine authorized user of system. The main technologies used in this project include JSF, OData, RESTful web service.

In general, a Login process includes the following steps:

* User Name validation
* User Password validation
* User Name verification
* User Password verification

# System Overview

The system is designed to demonstrate the user login procedures listed above. A few assumptions are made to simplify the demonstration as follows.

* The password encryption and decryption are not implemented.
* The user sign-up feature is not implemented. Default users and password are provided.

With these assumptions, the system is designed to have two key components:

* OData web service: It performs all operations required by the user login process,
* Login web server: While handling all user requests, it acts as a OData service client to communicate with the OData service for login procedures.
* Restful request: Retrieve search result from Google Custom Search.

There are two types of communications in the system:

* OData communication between the OData web service and the web server.
* Normal HTTP communication between the user, the web server and other APIs

# Implementation Overview

The Java technology is used for the business logic. Based on the design described above, different frameworks and libraries are chosen for the implementation of the OData service and the web server, as described as follows:

* OData web service

The Apache Olingo OData 4.0 library is used to implement the web service component. The library provides all types of RESTful APIs that follow the OData 4.0 standard, covering data access, data navigation and manipulation. In terms of the sign-on procedures, some of the APIs are implemented and will be presented in next section.

* Login web server

The implementation of the web server component follows a two-tier View-Service architectural design pattern using JEE technology. The JSF framework is used for the view layer following the model-view-controller pattern; and the Apache Olingo client library is used for Odata web service communication.

* Data source

A formatted JSON data file contains built-in user credentials, provides data to oData web service.

# Module Design Specification

This section presents the detailed module design for each component, including the API design of the Olingo OData service and the user interface design for the web server.

## Olingo OData API Design

The OData service requires defining the types of entities and entity sets that can be used in the RESTful API calls. The following lists the types of entity and entity set that are defined in odata.service package.

* Entity type: UserCredential
* Entity set type: UerCredentials

RESTful web service provides APIs to support CUD (Create, Update and Delete) operations. In terms of the project functional requirement, the following lists all supported RESTful API calls available from the OData web service.

* The OData service document
  + URL: <serviceroot>/
  + Method: HTTP GET
* The OData service metadata document
  + URL: <serviceroot>/$metadata
  + Method: HTTP GET
* Retrieving user credential
  + URL: <serviceroot>/odata
  + Method: HTTP GET

## Google Custom Search API Design

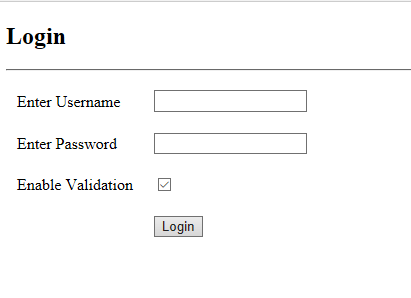
Google Custom Search enables people to create a search engine for own project. Using JSON/Atom Custom Search API to retrieve and display search results in JSON format from Google Custom Search by issuing RESTful requests

* Create Custom Search Engine from Google API Console, obtain the API key and engine ID
* Retrieve results for a particular search by sending an HTTP GET request to URI
* URI: https://www.googleapis.com/customsearch/v1?key= **YOUR\_API\_KEY**&cx=**YOUR\_ENGINE\_ID**&q=**YOUR\_SEARCH\_QUERY**

## User Interface Design

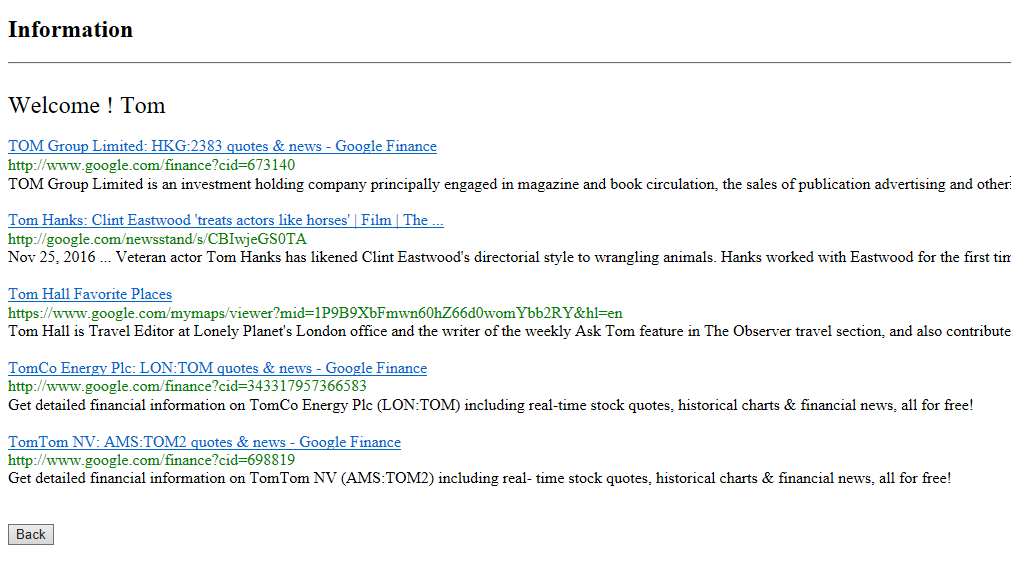
The User Login test project is a web-based application. Users interact with the application using a web browser. According to the functional requirements, a list of screens are identified and developed. The following shows all screen screenshots with business rules defined for each screen.

* User login screen



Business rules:

* + The user name has to be 3 -10 characters and has to be a real name
  + The password has to be 8-12 characters long with at least one capital, lower case, and one of the symbols: %#\*&!@ , and password cannot have more than 3 letters together.
  + A validation checkbox for switching validation process between client and server side.
  + A corresponding error message is displayed for breaking the user name and password rule, authorized user or invalid password.
  + A successful login navigates the user to the user information screen.
* The user information screen after successful login



Business rules:

* + The logged-in use name is shown on the top of the screen.
  + List first 5 google results for the user name. The search results from Custom Search Engine (free edition) are different with Google Site Search (paid edition) which is using on Google site.
  + Clicking on the ‘back’ button navigates the user back to the Login screen.