



Student Information System

Architecture Document

Contents

1. Introduction	3
1.1. Purpose.....	3
2. Scope	3
2.1. System Context.....	3
2.2. System Quality Attributes.....	3
2.2.1 Security	3
2.2.2 Right to Left Alignment	5
2.2.3 Multi Lingual Support.....	6
2.2.4 Cross browser support	6
3. Technology Stack	6
3.1 UI technology.....	6
3.2 DI Technology Options.....	7
3.3 Database Options:.....	8
3.4 ORM Framework Options.....	8
3.5 REST Service Options.....	8
3.6 Validation Framework.....	9
3.7 Business Rule Engine.....	9
3.8 Reporting Framework.....	10
3.9 SIS Technology Stack.....	11
4. Application Architecture.....	11
4.1 SIS Layered Architecture.....	12
4.2 Architecture Significance Use cases.....	13
4.2.1 Drools Rule Management.....	14
4.2.2 Dynamic Report	14
4.2.3 Messaging.....	14
4.2.4 Advanced Search Criteria Builder	14
5. Integration	15
6. Deployment Architectural.....	16
6.1 Database upgrade.....	16
6.2 Continuous Integration.....	16
6.3 Build Mechanisms.....	17
6.4 Code Versioning Methods.....	17

1. Introduction

1.1. Purpose

This System Architecture document describes the hardware and software environment within which the system application(s) will be implemented. Furthermore, it describes the system architecture of tiers, layers and sub-systems (as appropriate) for Academia Student Information System.

2. Scope

2.1. System Context

Serosoft has educational ERP/SIS product called "Academia" based on JAVA platform. The system will cater to all size of educational organization ranging from a university to small training institutes. The ERP/SIS will cover end-to-end functionality, right from core student lifecycle activities including admission enquiry, selection, admission, program & course enrolment, exams, evaluation management with ability to capture non-core & extracurricular activities. Along with this, the system will also have modules to control back end activities.

The system is compatible with multiple languages and supports Left to right and right to left (Arabic) languages. The system is web-based system that will be integrated with payment gateway for online payment functionality. The system will have multiple users and each user can have multiple roles. The system is integrated with Module to use different collaborative features from there, thus saving time and effort for developing such features.

2.2. System Quality Attributes

2.2.1 2.2.1 Security

Academia Application Security is focused on the internal controls that aim to limit user behavior and privileges. System supports the authentication, authorization mechanisms.

Academia uses spring security and Oauth 2.0 authentication protocol to address the security needs of an application. Oauth 2 protocol gives access token-based security to the resources on the server.

2.2.1.1 Authentication

Authentication is process by which you verify that someone is who they claim they are. When user login in academia by giving their credentials, the user credentials should be verified by system. Protected resource should be accessible only to Authenticated Users.

In Academia only registered users will get the access to resources on server. Registered user needs to login for access. Users will provide their credentials to application which will be verified at server and if credentials are valid then Oauth server will return an access token for the user. This access token is nothing but the string denoting a specific scope, lifetime, access attributes and user authentication. Then user can access the resources by providing this access token to server. If token is valid then only user will get access. In this way only, authenticated users can access the system.

2.2.1.2 AUTHORIZATION

Authorization is a process by which someone is allowed to be where they want to go, or to have information that they want to have. Academia system provides access to its services based on the access rights assigned to users.

Academia uses spring security Expression-Based Access Control to provide controlled access to authenticated users. Users will have access rights based on their type such as For Examination module, student will have only READ permission, but ADMIN will have CRETAE, READ, UDATE, DELETE permission. So, whenever user requests a service to access a resource, their access rights will be checked before providing access.

2.2.1.3 SENSITIVE DATA

Academia provides secure way to handle sensitive information like passwords. It uses the encryption mechanism to secure sensitive fields of entities such as password. These fields will be stored in database in the encrypted format.

2.2.1.4 HTTPS & LINK ENCRYPTION

Academia provides a security to communication links so that confidential information will not be breached by eavesdroppers during communication. It uses HTTPS for communication over network. HTTPS encrypts the data sent and received with SSL (secure sockets layer). With **HTTPS** if anyone in between the sender and the recipient could open the message, they still could not understand it. Only the sender and the recipient, who know the "encryption code," can decipher the message.

2.2.1.5 SERVER & CLIENT-SIDE SECURITY

Academia server is accessible only to valid users and academia client does not allow the user to send invalid input to server.

We use server-side authentication & authorization to provide secure access of academia functionalities only to valid users. For client side, ExtJS has inbuilt customizable validation which we use to accept only valid input from user, so user can not send invalid input to breach the server security. Similarly, user input will also be validated before proceeding request at server side using hibernate validator framework.

2.2.1.6 DATABASE SECURITY

Academia provides database level security for the database systems, stored functions, the database server and the associated network links.

As it uses MySQL database for the Academia, these database systems provide inbuilt security against database access. Only authenticated users will be allowed to access the database. We also use constraints mechanism provided by this system to extend the authority wise security.

2.2.2 Right to Left Alignment

The system is compatible with multiple languages and supports Left to right and right to left languages like (Arabic/Urdu) languages.

ExtJS 4.1 provides functionality of RTL. It has inbuilt libraries & logic of RTL included in its "Ext.rtl.*" namespace. If User wants to Use RTL, then they can import this namespace & will set rtl flag to true in Extjs configuration. All these

settings need to be done during UI development so UI will be backed with all language support with RTL or LTR configuration. During runtime User only switch between languages using option provided.

2.2.3 Multi Lingual Support

The system supports multilingual through which users can switch between the languages.

ExtJS also provide supports to switch between multiple languages. To achieve multiple language compatibility Ext JS comes bundled with localization files for over 40 languages ranging from Indonesian to Macedonian, and it's dead-easy to set up. For more details refer to the [Link](#).

2.2.4 Cross browser support

Academia application supports all major browsers like chrome, IE, Safari and Firefox. Application can be viewed on various screen sizes like desktop, tablet, mobile.

Ext JS 4 supports an incredible variety of browsers and on more operating systems using the same code. On modern browsers, Ext JS 4 utilizes HTML5 features and falls back to alternatives on older browsers. ExtJS supports all major web browsers including IE 6, Firefox 3.6+, Safari 4+, Chrome 10+, Opera 11+.

3. Technology Stack

3.1 UI technology

The following technologies have been considered for the presentation layer:

EXT JS 4.1 is a pure JavaScript application framework that works on all modern browsers from IE6 to the latest version of Chrome. It enables you to create the best cross-platform applications using nothing but a browser and has a phenomenal API.

Academia application is compatible with multiple languages and supports Left to right and right to left (Arabic/Urdu) languages. Since this feature is not supported by the JQuery while ExtJS supports RTL feature, hence we used ExtJS as a UI technology.

Features of EXTJS 4.1

- **EXTJS supports RTL feature**

RTL (Right-to-Left) is a key feature when application needs an RTL language support such as Urdu/Arabic/Hebrew etc. And EXTJS 4 has an in-built support for RTL Feature. Extj has inbuilt library for RTL that include the logic. RTL feature is enabled by simply importing a rtl library & enabling a rtl flag to true. EXTJs automatically render a screen UI based on selected language at runtime as for English LTR & for Urdu RTL. For more details refer a [link](#)

- **MVC Application Architecture**

Ext JS 4 brings a whole new way to build client applications, by introducing the popular model-view-controller pattern to Ext JS. By allowing the separation of data management, logic and interface elements, Ext JS 4 makes it easier for even large development teams to work independently without worrying about stepping on each other's toes. Ext JS 4 ships with a helpful MVC guide to get started.

- **Modern App Framework**

Ext JS 4 is a major step forward for web frameworks. Building on Ext JS 3.3, our latest release adds over 350 new APIs, 50 new classes, and 65% more documentation. An entirely new data package equips developers to leverage features like Infinite Grid Scrolling to build an entirely new level of interactivity to web apps.

- **Plugin free Charting**

No plugins, no problem. Ext JS 4 delivers the most advanced charting and graphing capabilities of any JavaScript framework, without relying on plugins, delivering pixel perfect visuals on any browser on any operating system. Leveraging SVG and VML, the Ext JS 4 charting package lets developers design and programs their graphs once for all browsers

3.2 DI Technology Options

The following technology option has been considered for providing dependency injection for system:

Spring:

The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform. A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments.

3.3 Database Options:

The following Database Option has been considered for the SIS product:

MySQL:

- It's open-source and community supported
- Scalable and secure database platform
- Reduces overall total cost of ownership for deployed solution

3.4 ORM Framework Options

The following ORM framework has been considered for the SIS product:

1. Hibernate:

Hibernate is an object-relational mapping (ORM) library for the Java language, providing a framework for mapping an object-oriented domain model to a traditional relational database. Hibernate solves object-relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling functions. Hibernate is the widely used ORM framework with Java applications

3.5 REST Service Options

REST stands for Representational State Transfer. It relies on a stateless, client-server, cacheable communications protocol i.e. HTTP. Similarly, a REST service also supports HTTPS to secure the communication channel. REST is a lightweight alternative to mechanisms like RPC (Remote Procedure Calls) and Web Services (SOAP, WSDL).

The following REST service option has been considered for SIS solution.

Jersey

Jersey RESTful Web Services framework is open source, production quality and framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) Reference Implementation. Jersey provides its own API that extends the JAX-RS toolkit with additional features and utilities to further simplify RESTful service and client development. In Academia we used jersey rest service as it is mainly devoted to RESTful services and have more advantages over other rest service options.

- Jersey is a Reference Implementation from Sun.
- It is simple & easy to use.
- Jersey makes it easy to build RESTful Web services with Spring using annotations like @GET&@POST & @PUT @DELETE.

3.6 Validation Framework

Validating data is a common task that occurs throughout Academia, from the presentation layer to the persistence layer. I.e. when client send request to server, server will validate the request parameters before proceeding to service the request. Often the same validation logic is implemented in each layer, proving time consuming and error-prone. To avoid duplication of these validations in each layer, we bundled the validation logic directly into the domain model, cluttering domain classes with validation code which is really metadata about the class itself.

Client-Side Validation

In Academia as we used EXTjs UI framework, it supports built in client-side validation such as web form validation. For advanced validation EXTjs provides VType which encapsulates a validation function, an error message (which will be displayed if the validation fails), and a regular expression mask to prevent any undesired characters from being entered the field. Extjs also provides flexibility as user can customize these VTypes and can add their own validation for fields.

Server-Side Validation

In Academia, for server-side validation we used Hibernate Validator framework which is a JSR 303 Reference Implementation. JSR 303 - Bean Validation - defines a metadata model and API for entity validation. The default metadata source is annotations. When request along with data parameters come from presentation layer to server data will be available in a Beans. Server will validate the request parameters before proceeding

to service the request. Hibernate Validator will validate the Beans and reports if any error.

Database Validation

In Academia, System validates data before persisting into the database. Database constraints are used for database validation. They define certain properties that data in a database must comply with. They can apply to a column, a whole table, more than one table or an entire schema. Common kinds of constraints are NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY and CHECK.

3.7 Business Rule Engine

A business rules engine is a software system that executes one or more business rules in a runtime production environment. The rules might come from business needs, organization policy or other sources. A business rule system enables these rules and other operational decisions to be defined, tested, executed and maintained separately from application code. Rule engines typically support rules, facts, priority (score), preconditions, and other functions.

The following Rule Engine option has been considered for SIS solution.

Drools : Drools is a business rule management system (BRMS) with a forward chaining inference based rules engine, more correctly known as a production rule system, using an enhanced implementation of the Rete algorithm. We have used Drools rule engine in academia as it is most popular & open source rule engine.

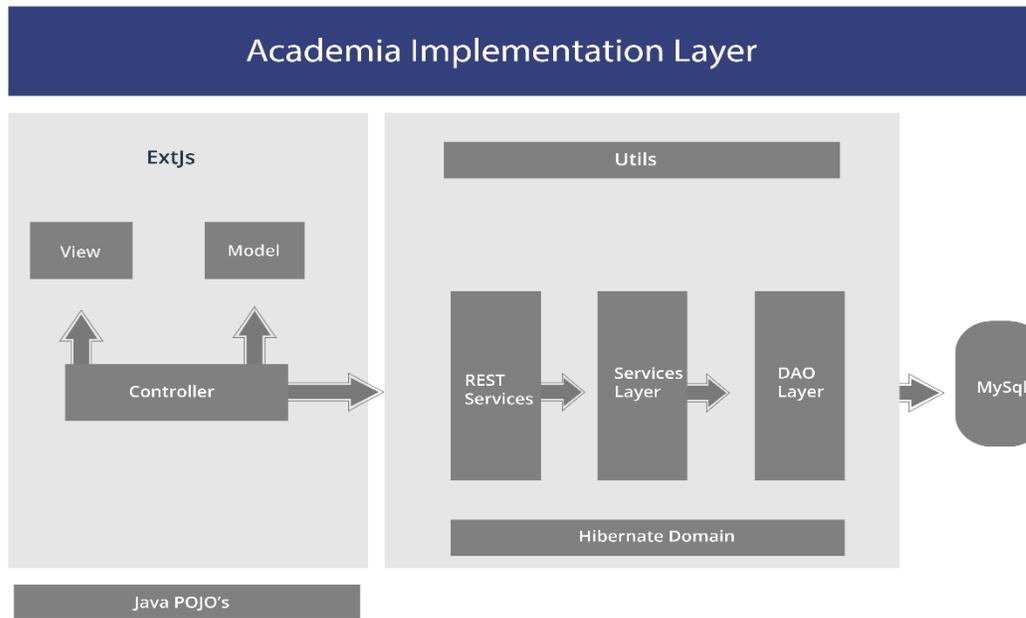
3.8 Reporting Framework

Reporting may refer to any activity that leads to reports. In Academia Jasper Report 4.x framework has been used for reporting. Jasper Reports is an open source reporting library that can be embedded into any Java application. Jasper Reports is the world's most popular open source reporting engine. It can use data coming from any kind of data source and produce pixel-perfect documents that can be viewed, printed or exported in a variety of document formats including HTML, PDF, Excel.

3.9 SIS Technology Stack

UI Technology	Ext Js 4.2.1
Reporting	Jasper Reports 4.0.2
Dynamic Reports	XDocReport 1.0.2
Rules Engine	Drools 5.5.0
Rest Services	Jersey 2.3.1
Di Framework	Spring 3.6
ORM Framework	Hibernate 4.0
Velidator Framework	Hibernate Validator 5.0
Application Server	Tomcat 7.0
Database	MyS41 5.6.14

4. Application Architecture



4.1 SIS Layered Architecture

S. No	Layer	Description
CLIENT		
1	View	A View is nothing more than a Component, usually defined as a subclass of an Ext JS component. Separate from the application logic of your code. It is any type of component like grids, trees and panels are all views.
2	Model	Model is a collection of fields and their data Models know how to persist themselves through the data package and can be linked to other models through associations. Models work a lot like the Ext JS 3 Record

		class, and are normally used with Stores to present data into grids and other components.
3	Controllers	Controllers are special places to put all of the code that makes your app work - whether that's rendering views, instantiating Models, or any other app logic.
SERVER		
4	Utility	Common utility classes which will be shared between the Views, Activities and Custom Components
5	Rest Services	The REST services are designed to use a stateless communication protocol, typically HTTP. Using REST services, clients and servers exchange representations of resources by using a standardized interface and protocol. Rest service will interact with the Service Layer to provide service to client. REST service will hold the logic of conversion of the data from service layer to required format like JSON.
6	Service layer	This layer will hold the business logic of the application. The Service layer will retrieve the Hibernate entities from the DAO layer .
7	DAO layer	This layer will hold the logic to interact with the database. The DAO layer will use the API to interact will the database.
8	Java Domain	This layer will hold the data for communication between the Activity, Service and DAO layers

4.2 Architecture Significance Use cases

4.2.1 Drools Rule Management

Academia application has rules associated with every process in the system. This application is used by different institutions. Every institute may have different rules which will be defined by the business users of the system. So Application needs to provide a way to easily create & manage these rules so that new rules can be added & existing rules can be modified or removed. During such change application code should not need to change. We have used JBoss Drool Rule engine to manage rules.

For implementation details refer to the Drools Document.

4.2.2 Dynamic Report

Academia application provides dynamic documents generation in various processes based on data in that process. Documents or reports will be generated dynamically using predefined templates. Since application is used by many different institutions, every institution may have different formats of such reports. So application needs to provide a way to easily define templates for these documents & using these templates documents should be generated dynamically. Hence, we have used XDocReport API for dynamic document generation.

For the implementation details refer to the documentation of Dynamic Document.

4.2.3 Messaging

In Academia there are multiple modules from where the notifications, alerts, messages or reports can be sent through a communication medium such as email, sms, and links on the screen when a user logs in. For messaging via sms we can use Gateway4sms API and for e-mails we can use Amazon SES API as well as Exchange Server.

For the implementation details refer to the Messaging Document.

4.2.4 Advanced Search Criteria Builder

In academia criteria builder is a generic component that is used in all advanced search (in list screens and dashboards), lookup for a resource, report filters. Advanced search module is consisting of two parts. One is Creation of View of a search result. And another part is criteria builder for the search. Using these two functionality users efficiently get the search results in required format satisfying

given criteria. We have used database driven query builder logic for advanced search filter.

For more details refer to the Advanced Search Criteria Builder Document.

4. Integration

1.	Biometric System (automated attendance recording system)	Every Biometric system has a unique identifier for each user. The academia application provides an additional field on the User Management screens to link the User with the Biometric system. For each User in Academia the administrator can manually fill the Biometric identifier value to link them to the Biometric system.
2.	Payment Gateway	Payment gateways shall be integrated for online credit/debit card, net banking payment options for collection of fees from students and other stakeholders.
3.	Accounting Software's	Academia has got capability to integrate with different Accounting packages like Tally, QuickBooks, Sage, etc.
4.	Integration with Institution Website	Every institute may have its own website and there may be a need to publish various data like timetables, admission list etc. on the website. Since most of such websites may be static in nature and the institutes may not have their developers to maintain the site a very easy solution is required for this integration. The simplest solution would be to redirect the request to the Academia application and integrate the academia screens into the institute website. The academia screens need to match the look and feel of the website and the redirection should be transparent to the end user.
5.	Integration with Amazon Email Services	For seamless email sending from Academia, Amazon SES is suggested as it provides all the necessary tools for emailing requirements for a system of this nature.

6.	Integration with Exchange Server	If required Microsoft exchange server integration for email can be done.
7.	Integration with SMS Gateway	SMS gateways which support API based integration shall be integrated for SMS.
8.	ORM Scanning	OMR based software can also be integrated as optional modules for OMR based examination flows

5. Deployment Architectural

6.1 Database upgrade

In Academia we have used flyway for database upgrades. Flyway updates the database from one version to the next using migrations. Each migration must have a unique version and a description. Flyway support Sql and Java migrations.

During Academia application loading time database connection gets established and next the flyway upgrades the database using migration scripts.

6.2 Continuous Integration

Jenkins

In Academia we have used Jenkins for continuous integration of system. Jenkins is an open source continuous integration tool written in Java. Jenkins provides continuous integration services for software development. It is a server-based system running in a servlet container such as Apache Tomcat. Jenkins focuses on the following two jobs:

1. Building/testing software projects continuously, such as continuous integration system, making it easier for developers to integrate changes to the project.
2. Monitoring executions of externally-run jobs, such as cron jobs.

Jenkins offers the following features:

- Easy installation
- Easy configuration
- Change set support

- Permanent links
- JUnit/TestNG test reporting

- Distributed builds
- Plugin Support

6.3 Build Mechanisms

Apache MAVEN has been used for managing the builds. MAVEN is a tool, used for building and managing any Java-based project. It makes the build process easy. Maven allows a project to build using its project object model (POM) and a set of plugins that are shared by all projects using Maven, providing a uniform build system.

6.4 Code Versioning Methods

SVN: In Academia we have used Apache **Subversion** (SVN) for code versioning. This is a version control system for keeping track of all modifications to project source code files. SVN is widely used in both open source and proprietary software development projects and is generally considered to be the best freely available, full-featured version control tool.

