

“Java Advanced Project”

Software documentation



BY

Sandesh VAKALE

Contents

Subject description	3
Goal	Error! Bookmark not defined.
Use case	Error! Bookmark not defined.
Subject analysis	4
Major features	4
Application Feasibility	4
Data description	5
Expected results	5
Algorithms study	5
[I] Saving data	5
[II] Searching data	5
[III] Website Responsiveness	5
[IV] Packaging & Architecture	5
[V] Authorization & Security	5
[VI] Software configuration management	5
Scope of the application	5
Conception	5
Chosen algorithm	5
[I] Saving data	5
[II] Searching data	6
[III] Website Responsiveness	6
[IV] Packaging & Architecture	6
[V] Authorization & Security	Error! Bookmark not defined.
[VI] Software configuration management	6
Global application flow	7
Global schema and major features schema	Error! Bookmark not defined.
GUI description	Error! Bookmark not defined.
LOGIN	Error! Bookmark not defined.
Configuration instructions	7
Database configuration	7
Open in browser	7

Subject description

AIM

This project is created as a mean to show sufficient knowledge of java which is acquired during the Advanced Java classes at EPITA.

During these java classes a range of technologies and best practices related to java is shown, this knowledge is implemented in this project to show sufficient knowledge about the technology and how to apply them in practice.

Scenario

As a Scenario a Student management system is created which includes all the technologies and best practices acquired during java class. This system is an extension of the developed application during the previous java class.

The hospital management system which is created should be a web application. The main features are analyzed in the next chapter.

Subject analysis

The minimal requirements for the system are extracted from the specification provided on the Advanced Java webpage. The requirements are applied to the use case.

Major features

The following 4 categories of features and associated requirements are extracted:

Main application features

- Login page for authentication
- Welcome page, as a sort of dashboard
 - Disconnect options, takes the user to the login page
- Creation page
- Manage page
 - Insert option
 - Update option
 - Delete
 - Search

Technology requirements

- Use Hibernate or JDBC to store data models

Process requirements

- Use Junit to test
- Scalability

Additional features

- Make it possible to Search Any Fields of Table from single input.

Application Feasibility

To determine the feasibility we will analyze the requirements and determine the risk for the realization of this requirement

[1]	The creation of webpages including the communication with servlets and backend processing is covered during class and can be used in this project.	Average
[2]	The data models are simple and Hibernate and JDBC are known technologies.	Average
[3]	The context in this project is different from the learned content during class. Since we are working with Spring and JUnit on servlets this could provide issues while testing and can result in additional time spend.	Medium
[4]	Scalability in the context of web development is a new area which knowledge needs to be accumulated, this can result in potential extra work	Medium
[5]	The addition of new fields in a web application is Searching Any data from Table Typing in One Field. Which is Web Services & JavaScript Features and it is very User Friendly	Average

Data description

The project mainly manages student data. Below the two data types are displayed.

Student	
Field	Type
Uid	int
StudentName	String
EmailID	String
UserName	String

Expected results

The result of this project should contain the features and extracted requirements. There should be documentation which is clear enough to get the main idea of the application and its structure.

Algorithms study

The different possibilities to realize certain functions are analyzed to result in the best applicable solution

[I] Saving data

To Student Data Model previously a specific DAO implementation is used. This implementation could handle different data models, but cannot handle dynamic data. Since the application should be scalable 0, it is preferred to have a more dynamic solution. Hibernate can be a better solution to provide this 0.

[II] Searching data

The application should feature a search method. Since the data should be scalable 0 & 0, the search should be able to handle dynamic fields. JavaScript can be the better solution for fast and friendly response

[III] Website Responsiveness

It should be able to do operations on Students like create, modify and delete without unnecessary page reloads.

[IV] Packaging & Architecture

The different features required should be organized in a way that is maintainable. The basic functionality from the previous java semester can be used where necessary.

[V] Authorization & Security

To provide security an authorization mechanism should be provided. The mechanism of the previous project will be used. A method should be created to save the User credentials and status for each connected in parallel.

[VI] Software configuration management

To control the different version of the used packages a system needs to be used.

Scope of the application

The application is restricted to the features specified in chapter *Major features*.

Conception

This chapter provides the chosen design for the application

Chosen algorithm

[I] Saving data

Saving data is done using the hybernate technology, this ensures the flexibility required

[II] Searching data

To provide the very smooth and Fast Results Searching Data is done by JavaScript. It will filters all the data of table according to requested data into search fields. It gives very Fast Response.

[III] Website Responsiveness

To provide a responsive flow to the user without unnecessary refreshes of pages the AJAX technology will be used.

Sending data between Pages and servlets will be done using JSON objects. This provides an efficient and scalable solution for data transfer.

[IV] Packaging & Architecture

To organize the different features the project is subdivided in the packages shown below, each package is a project.

[VI] Software configuration management

Maven

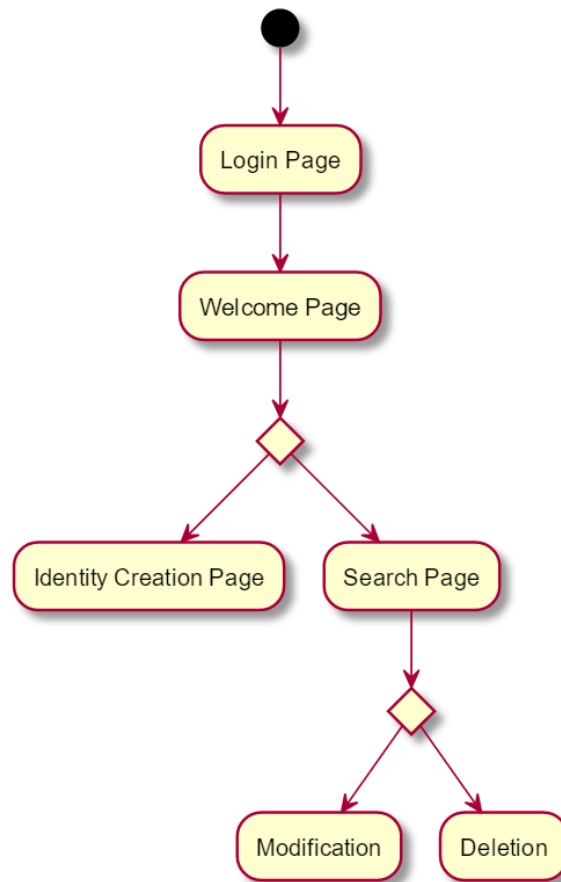
To provide a configuration environment Maven is used. The project will comply with the company versions specified in the company Maven “super POM”.

Spring

To provide a flexible software application the dependency injection framework Spring is used. Interfaces will be created to be able to inject different implementations.

Global application flow

The global application flow is based on the specified requirements [REQ 1-4]. The flow is depicted below:



Configuration instructions

Database configuration

Open in browser

The website is generally run on: <http://localhost:8080/lamWeb-Spring/index.jsp>