Design Documentation

J2EE  
To-do Application

Mike LeBlanc

W0247871

Table of Contents

[Introduction 3](#_Toc159579012)

[Overview 3](#_Toc159579013)

[Scope 3](#_Toc159579014)

[Architectural Overview 3](#_Toc159579015)

[Technologies Used 4](#_Toc159579016)

[System Components 4](#_Toc159579017)

[Database Design 4](#_Toc159579018)

# Introduction

## Overview

This application was developed to satisfy the assignment criteria and serve as a personal list of things necessary to complete.

## Scope

This application is intended to be lightweight and easy to navigate. Having just a handful of pages to navigate through. This was a project to learn the fundamentals of J2EE as they are.

# Architectural Overview

A diagram of a computer

Description automatically generated

Client can access the JSP of the application (Views) the controllers access the required data through the help of the models and serve it back to the JSP.  
The architectural pattern that was employed with this application was the Model, View, Controller (MVC) pattern.

The Models are plain object used as a framework to hold information needed.

The Controllers fetch information from the database and serve it to the user though the Java Server Pages.

Views act as the interface that the users and clients can interface with and see the information served to them.

# Technologies Used

With this application, we have employed Java 2 Enterprise Edition, Tomcat application server, and MySQL (MariaDB) Database.

These technologies were selected because J2EE was a requirement for the Assignment. MySQL was selected for data storage because it is cost effective, powerful, and the most comfortable. These are also the reasons for Tomcat application server.

# System Components

The components used in this To-do application range from JavaServer Pages, Servlets, and web Components.

JavaServer Pages are used for the user to interface with and consume the information they requested.

Servlets with help from the context of the request, fetch the information and serve it back to the user.

Web Components respond to HTTP requests made by the web-browser and thus the user. JavaServer Pages, servlets, and web listeners make up the web components.

# Database Design

A green and black box with black text

Description automatically generated

The tables shown above strived for simplicity. The To-do table is tasked with storing the title and description of the To-do, as well as the username of who created it, and the ID of the entry.

The users table holds the data of those who’ve decided to sign up to this application. It holds the first name, last name, their chosen username, and password, as well as the ID number of the entry.

The relationship between the tables is linked through the username. When the query pulls To-Dos from the database, it will only return the entries related to the user who has logged in.

This page intentionally left blank.