RecordSystem

Software Architecture Document

Version 1.1

12/4/2018

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Description of Versions / Changes** | **Responsible Party** | **Date** |
| 1.0 | Initial version | Ian C | 11/29/2018 |
| 1.1 | Changes to Add Topic,Manage Grades,View Attendance | Ian C | 04/12/2018 |

**Approval Block**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Comments** | **Responsible Party** | **Date** |
| 1.0.1 | UI Changes | Ian C | 11/26/2018 |
| 1.0.2 | Backend Services | Ian C | 11/27/2018 |
| 1.0.3 | Combined system | Ian C | 11/28/2018 |
| 1.0 | Completed System | Ian C | 11/29/2018 |
| 1.1 | System enchantments | Ian C | 04/12/2018 |

Table of Contents

[1. Introduction 1](#_Toc353470033)

[1.1. Purpose 1](#_Toc353470034)

[1.2. Scope 1](#_Toc353470035)

[1.3. Definitions, Acronyms, and Abbreviations 2](#_Toc353470036)

[1.4. References 2](#_Toc353470037)

[1.5. Overview 2](#_Toc353470038)

[2. Architectural Representation 3](#_Toc353470039)

[3. Architectural Goals and Constraints 4](#_Toc353470040)

[4. Use-Case View 4](#_Toc353470041)

[4.1. Actors 4](#_Toc353470042)

[4.2. Use-Case Realizations 5](#_Toc353470043)

[4.2.1. Login 5](#_Toc353470044)

[4.2.2. Request Analysis (Get Report) 7](#_Toc353470045)

[4.2.3. Retrieve Last Report 9](#_Toc353470046)

[4.2.4. Print Report 10](#_Toc353470047)

[4.2.5. Email Report 11](#_Toc353470048)

[5. Logical View 12](#_Toc353470049)

[5.1. Overview 12](#_Toc353470050)

[5.2. Interface Definitions 14](#_Toc353470051)

[6. Data View 23](#_Toc353470052)

[7. Deployment View 25](#_Toc353470053)

Software Architecture Document

# Introduction

This document provides a high-level overview and explains the architecture of the RecordSystem software solution.

The document defines goals of the architecture, the use cases supported by the system, architectural styles and components that have been selected. The document provides a rationale for the architecture and design decisions made from the conceptual idea to its implementation.

## Purpose

The Software Architecture Document (SAD) provides a comprehensive architectural overview of the Distributed Development Monitoring and Mining system (DMM). It presents a number of different architectural views to depict the different aspects of the system.

## Scope

The scope of this SAD is to explain the architecture of the RecordSystem software solution.

Stakeholders who require a technical understanding of the RecordSystem software solution system are encouraged to start by reading the Project Proposal, Concept of Operations and Software Requirements Specification documents provided by the client.

## References

ICTPRG\_501\_502\_503\_AT1\_PE.docx

ICTPRG527\_AT1\_PE\_Elan.docx

PRELIMINARY BACKGROUND INFO.docx

ICTPRG\_501\_502\_503\_AT1\_MC.docx

## Overview

In order to fully document all the aspects of the architecture, the Software Architecture Document contains the following subsections.

Section 2: describes the use of each view & server components

Section 3: describes the architectural goals and constrains

Section 4: describes main functionalities

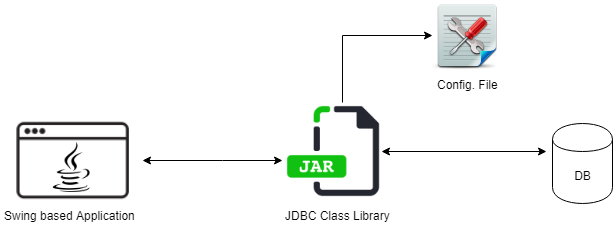
Section 5: describes view component of the system including interface and operation definitions.

Section 6: describes how the data tier.

Section 7: describes how the system will be deployed.

# Architectural Representation

The system is divided into two main components: Front End and Back End.



Front End: Developed as a Java Swing based application. Responsible of handing user interactions to the system.

Back End: Developed as a Java System Library. Responsible on connecting to the persistent tire.

# Architectural Goals and main functionalities

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

1. System needs to be developed in Java SE.
2. System must run on cross-platform.
3. System needs to connect to MySQL DB.
4. System should read configuration from external file.

# Main Functionalities

## Manage Teacher Details

### Teacher Login

### Edit Teacher Details

## Manage Student Details:

### Add New student

### View existing students (Table View)

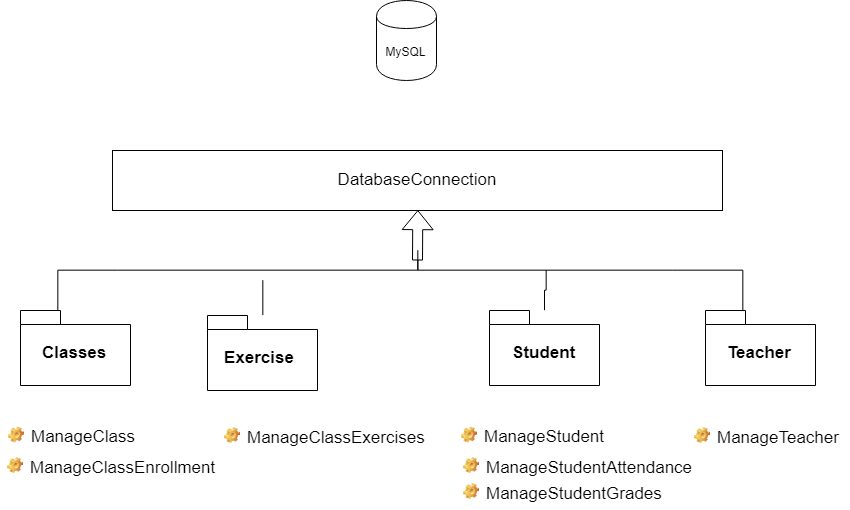
### Edit Existing student details

## Manage Classes & Attendance:

### Add classes

### Link Students to Classes

### Add student attendance



**Figure 2** Main component implementation

# View Component

## Overview

The main goal of the view component is to define the main UI screens and the process flow of the system.

## Main UI screens

### Welcome Screen

### Teacher Login

#### Change Teacher Login details

### Manage Students

#### Add new Student

#### Edit Existing Students

#### View Existing Students

### Manage Classes

#### Add new classes

#### Link existing students to classes

#### Mark student Attendance

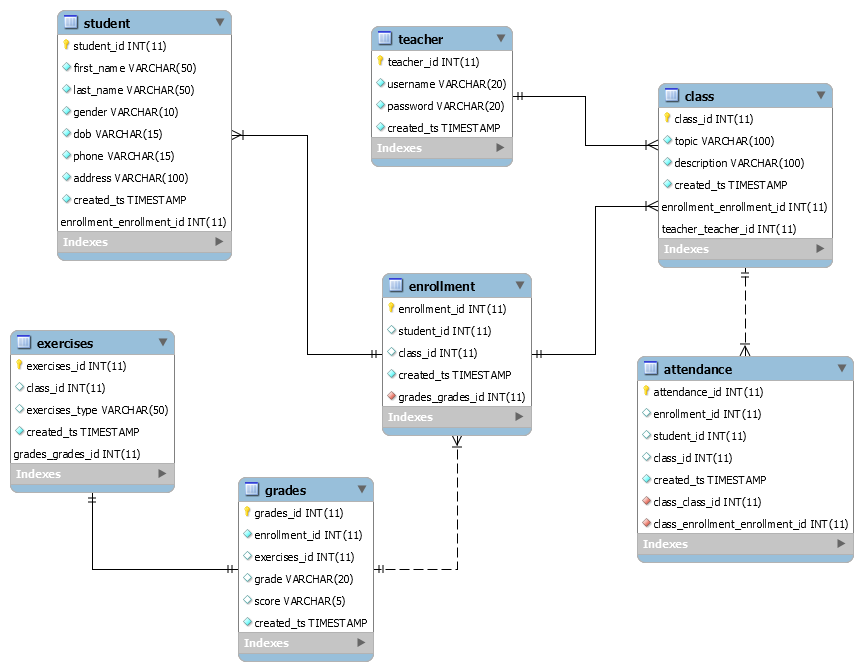
### Manage Class Exercises & Grades

#### Add new exercises to existing classes

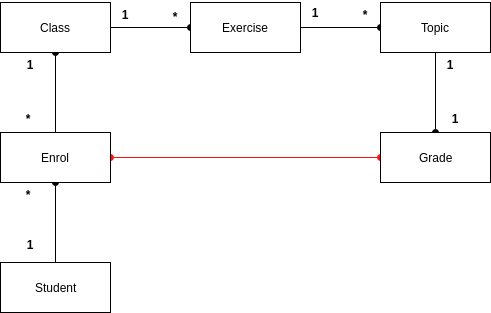
#### View Grades & Exercises

# Data View

This below diagram illustrates the data structure and relationships of the main entities that will be stored by the application in its database. Each element nominally represents a database table. Relationship cardinality is denoted with UML multiplicity notation.



New changes are added to the student Grades:



# Deployment View

The Application is packaged as a JAR file : RecordSystemApp.jar

To Execute the file type :

java -jar RecordSystemApp.jar <config file path>

The configuration file should look something below:

|  |
| --- |
| # Application Information  APP\_NAME=Record System App  APP\_VERSION=Version 1.1  APP\_AUTHOR=TEST USER  # Database configuration  DB\_URL=jdbc:mysql://localhost:3306/classroom\_records  DB\_USER=<>  DB\_PASSWORD=<>  DB\_DATE\_FORMAT=yyyy-MM-dd |

NOTE: The DB\_DATA\_FORMAT property is used to set the DB date time format.