|  |
| --- |
| Software Architecture Document for Grade Performance Analyzer |
|  |
|  |

**Revision History**



DJ Grader

Version 1.0

Prepared by A.Y Dissanayake

Index No: 120126K Date: 12/07/2015

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/07/2015 | 1.0 | Initial Version | A.Y Diisanayake |
| 07/09/2015 | 1.1 | 2nd Version | A.Y Diisanayake |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introduction 5](#_Toc394093119)

[1.1 Purpose 5](#_Toc394093120)

[1.2 Scope 6](#_Toc394093121)

[1.3 Definitions, Acronyms, and Abbreviations 6](#_Toc394093122)

[1.4 References 6](#_Toc394093123)

[1.5 Overview 6](#_Toc394093124)

[2. Architectural Representation 7](#_Toc394093125)

[3. Architectural Goals and Constraints 8](#_Toc394093126)

[4. Use-Case View 9](#_Toc394093127)

[5. Logical View 13](#_Toc394093128)

[5.1 Overview 13](#_Toc394093129)

[6. Process View 15](#_Toc394093130)

[7. Deployment View 20](#_Toc394093131)

[8. Implementation View 21](#_Toc394093132)

[9. Data View 22](#_Toc394093133)

[10. Size and Performance 22](#_Toc394093134)

[11. Quality 22](#_Toc394093135)

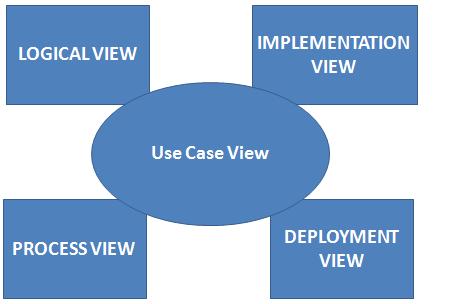
# Introduction

This document provides a high level overview and explains the whole architecture of MyGrader, the application for lecturers. It provides a high-level description of the goals of the architecture, the use cases support by the system and the architectural styles and components that have been selected to best achieve the use cases. The document is intended to capture and convey the architectural decisions that have been made and elaborates on aspects of the system that are considered to be architecturally significant.

## Purpose

The software architecture document is intended to show how the software system will be structured to satisfy the requirements previously identified in the software requirements specification. It will depict the translation of requirements into a description of the software structure, software components, interfaces and data necessary for the implementation phase of the Rational Unified Process. The document is targeted for the implementation and hence provides a blueprint for the activities carried out during this phase.

The document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. For the convenience of the different stakeholders, the document will be based on the “4+1” model view of architecture. The model will make it easier for the stakeholders to find what they need in the software architecture.



The 4+1 view model describes the architecture of the system in terms of basically five views. They are **Use Case view, Logical view, Process view, Implementation view** and **Deployment view.** In addition to these the document uses three more views namely, **Data view**, **Size & Performance** and **Quality**. Detailed descriptions of these views will be included in a future section. The purpose of using different views is to makes it possible for stakeholders to get the parts of the model that are relevant to them. The design decisions mentioned in the document will be directly used for implementation of the application.

## Scope

This document applies to the overall design of the application. It contains information related to the architectural design of the software, the structure of the database etc. The project is using the agile development methodology and therefore is subject to change. The document will evolve and additional details will get added depending on the decisions taken during the implementation activities.

## Definitions, Acronyms, and Abbreviations

Some of the technical terms used throughout the document and their meanings are explained below.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| RUP | Rational Unified Process |
| UML | Unified Modeling Language |
| SAD | Software Architecture Document |

## References

The following document/s should be read in conjunction with the SAD.

|  |  |  |
| --- | --- | --- |
| **Document/Reference** | **Version** | **Date** |
| Software Requirements Specification | 1.0 | 12/07/2015 |

## Overview

The basic purpose of this document is to help the reader visualize the solution to the project presented in the software requirements specification. For further convenience, the document has been organized into different sections each one providing details of a specific aspect of the software application. The sections included are as mentioned below.

**Section 2**: Describes the use of each view in 4+1 model view.

**Section 3**: Describes the architectural constraints of the system

**Section 4**:Illustratesthe functional requirements which have a significant impact on the architecture.

**Section 5**: Describes the architecturally significant layers of the application and the packages within those layers.

**Section 6**: Describes design’s concurrency aspects. It is concerned about the processes.

**Section 7**: Describes how the system will be deployed. It contains the Deployment Model.

**Section 8**:Describes the components that the application comprises of and their interactions.

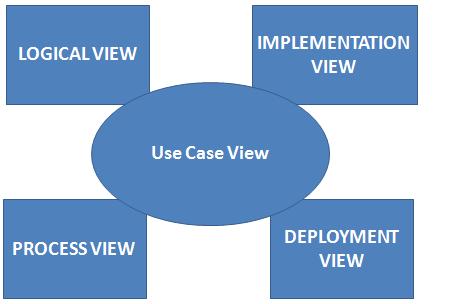
**Section 9**: Describes thestructure of persistent data. It contains the Data Model.

**Section 10**: Describes any performance issues and constraints.

**Section 11**: Describes any aspects related to the quality of service (QoS) attributes.

# Architectural Representation

As mentioned in the preceding section, this document details the architecture using the views defined in the “4+1” model. The following views will be used to discuss the architecture of the application with appropriate UML diagrams where necessary.



|  |  |  |
| --- | --- | --- |
| **Views** | **Representation** | **UML Diagrams** |
| Architectural Goals and Constraint | This view describes the architecturally significant goals and constraints for the application. | None |
| Use-case View | This view describes the functionality of the application from the perspective of the outside world. | UML Use-case Diagrams. |
| Logical View | Concerned with the functionality that the system provides to end users. | UML 3-tier Layered Diagram |
| Process View | It encompasses some non-functional requirements such as performance and availability.. | UML Activity Diagram |
| Deployment View | Shows how the application is physically configured (deployed). Describes the system from a system engineer’s perspective. | UML Deployment Diagram |
| Implementation View | Illustrates the system form the programmer’s perspective. It is concerned with software management. | UML Component Diagram |
| Data View | Describes the architecturally significant persistent elements of the system. | None |
| Size and Performance | Shows data volumes and expected performance figures | None |
| Quality | Shows expected quality metrics | None |

# Architectural Goals and Constraints

This section describes the software requirements and objectives that have some significant impact on the architecture.

**Architectural Goals:**

* Good response time for special features such as result prediction.
* Data persistence should be addressed using a relational database in the computer.
* Authentication supported using username and password.
* Search queries should return %90 of the time below 5 sec.
* The architecture should be flexible and extensible, ensure reusability for the next phase of the development.

**Architectural Constraints:**

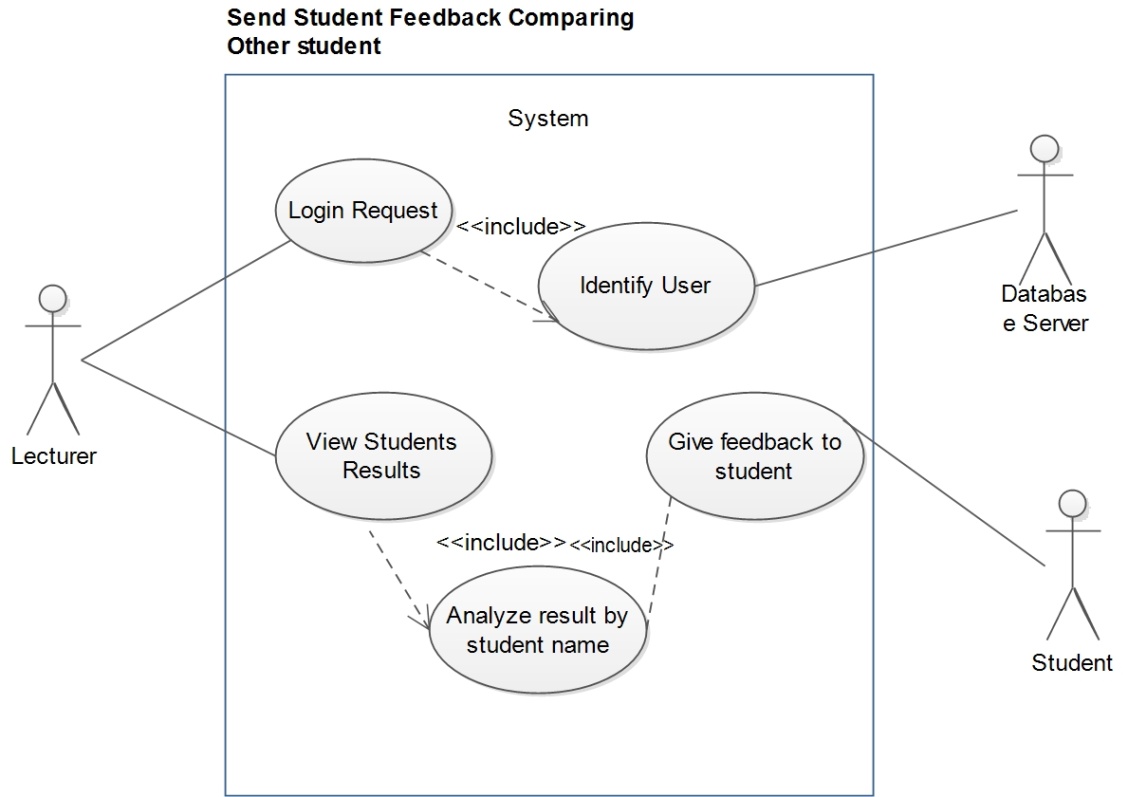
* Potential performance impact by large dataset.
* Response time constraints due to less processing power.

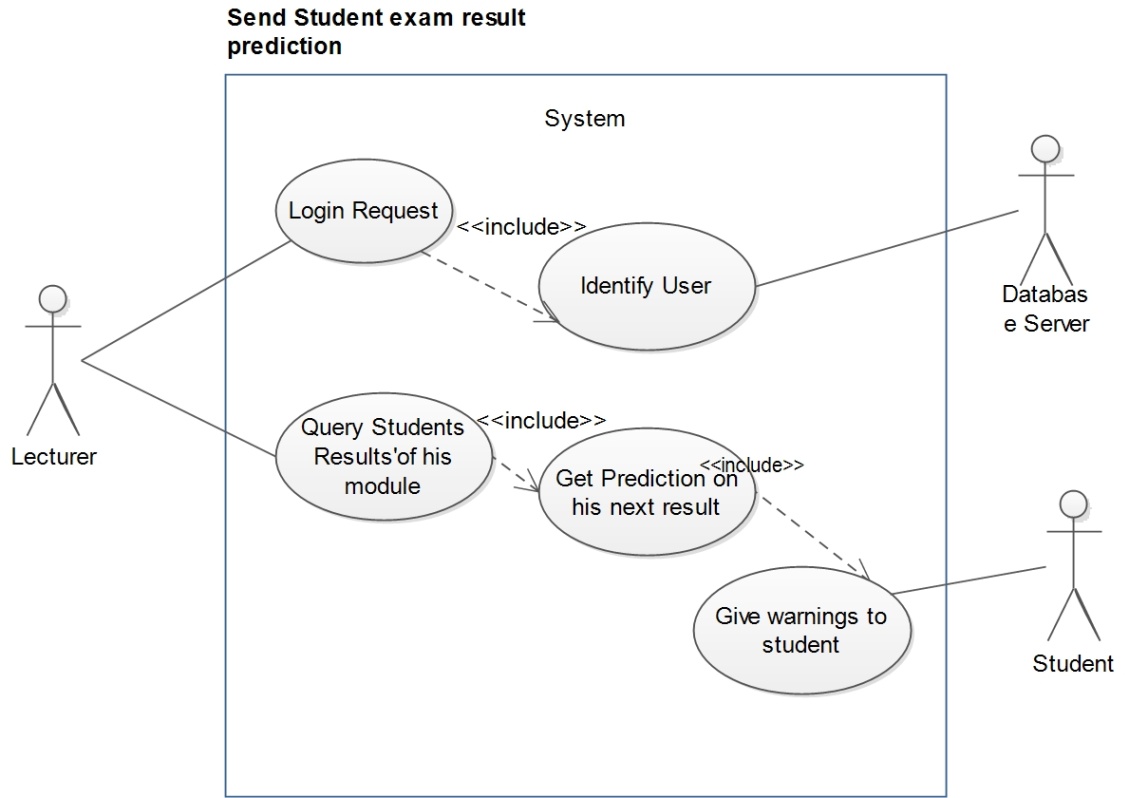
# Use-Case View

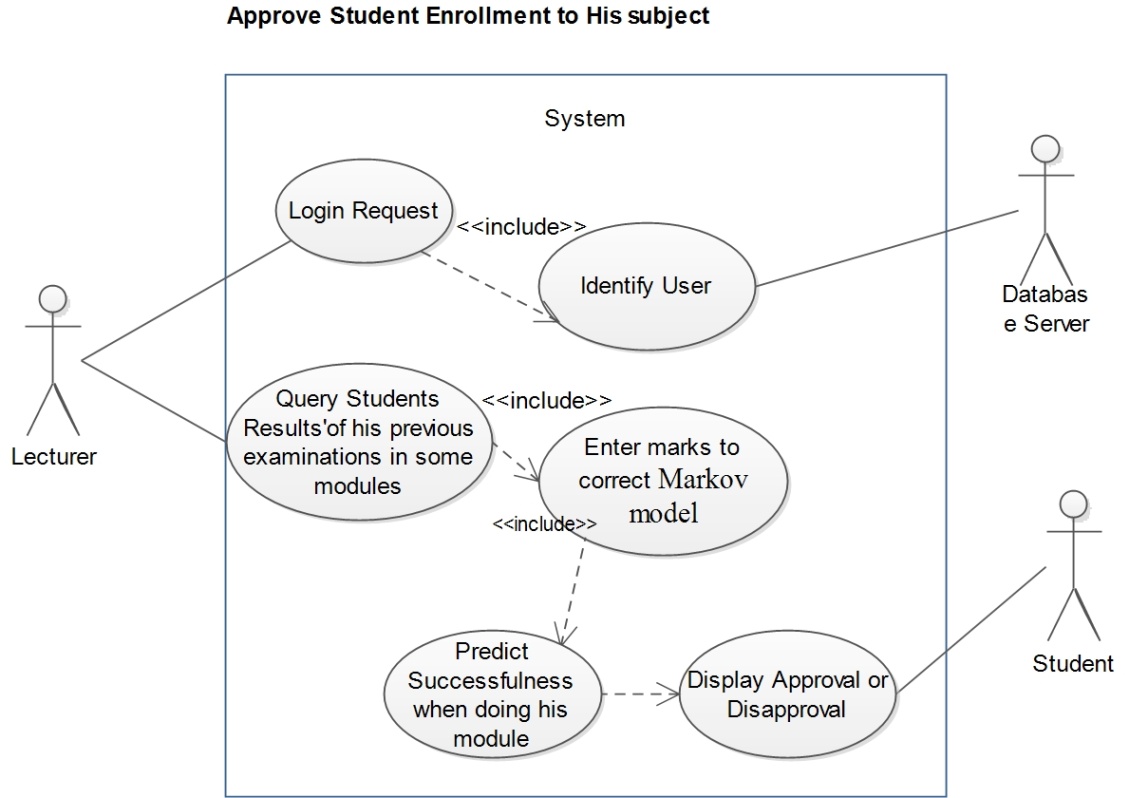
This section lists use cases or scenarios from the use-case model if they represent some significant, central functionality of the application MyGrader or if they have a large architectural coverage - they exercise many architectural elements, or if they stress or illustrate a specific, delicate point of the architecture

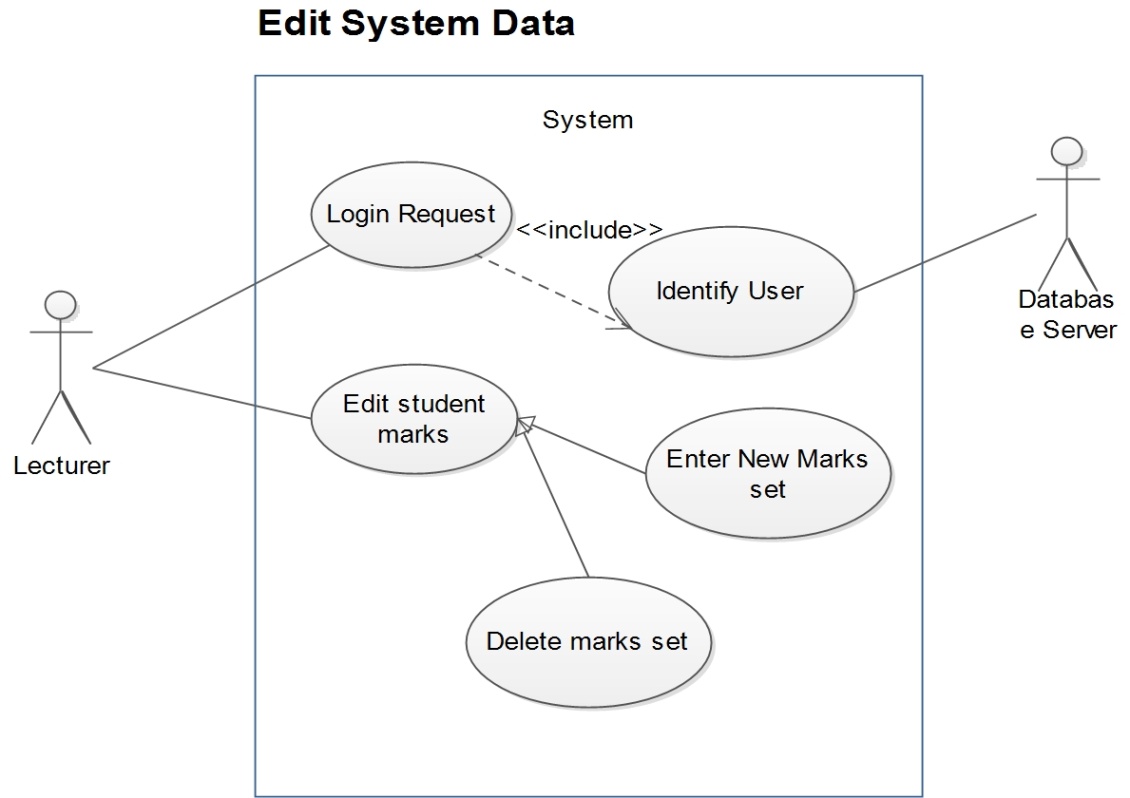
### data reporting

### 







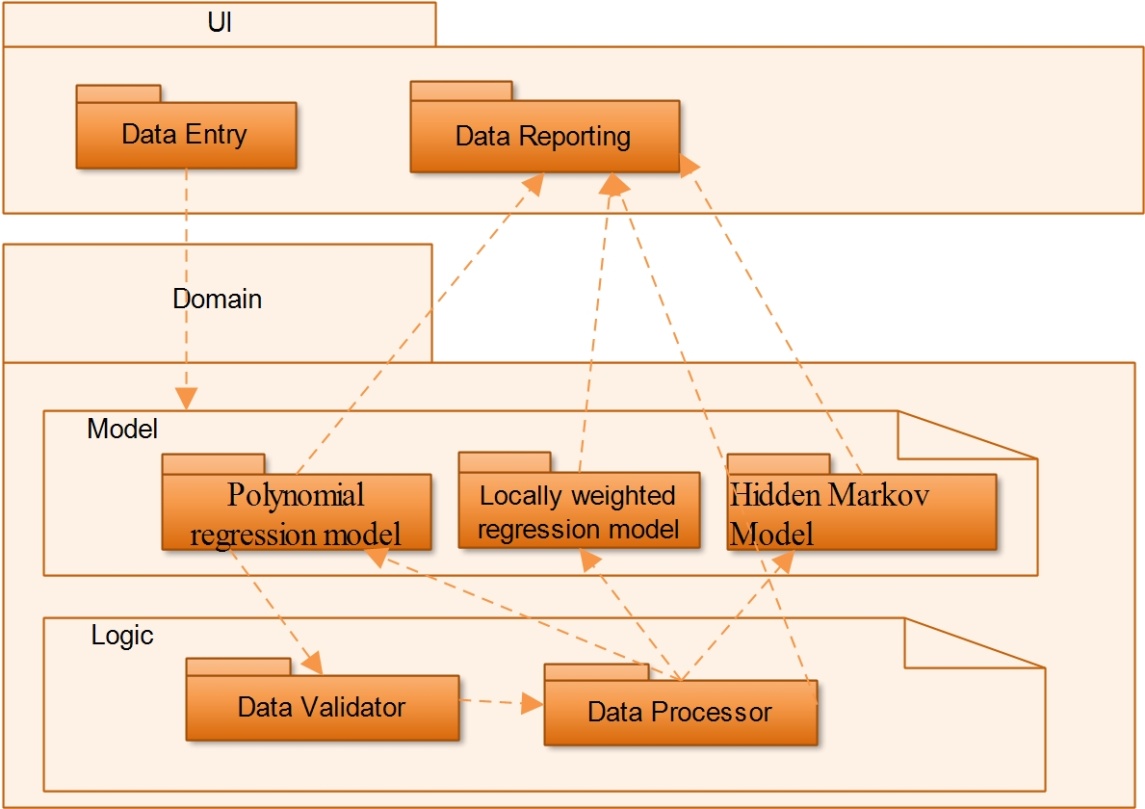


# Logical View

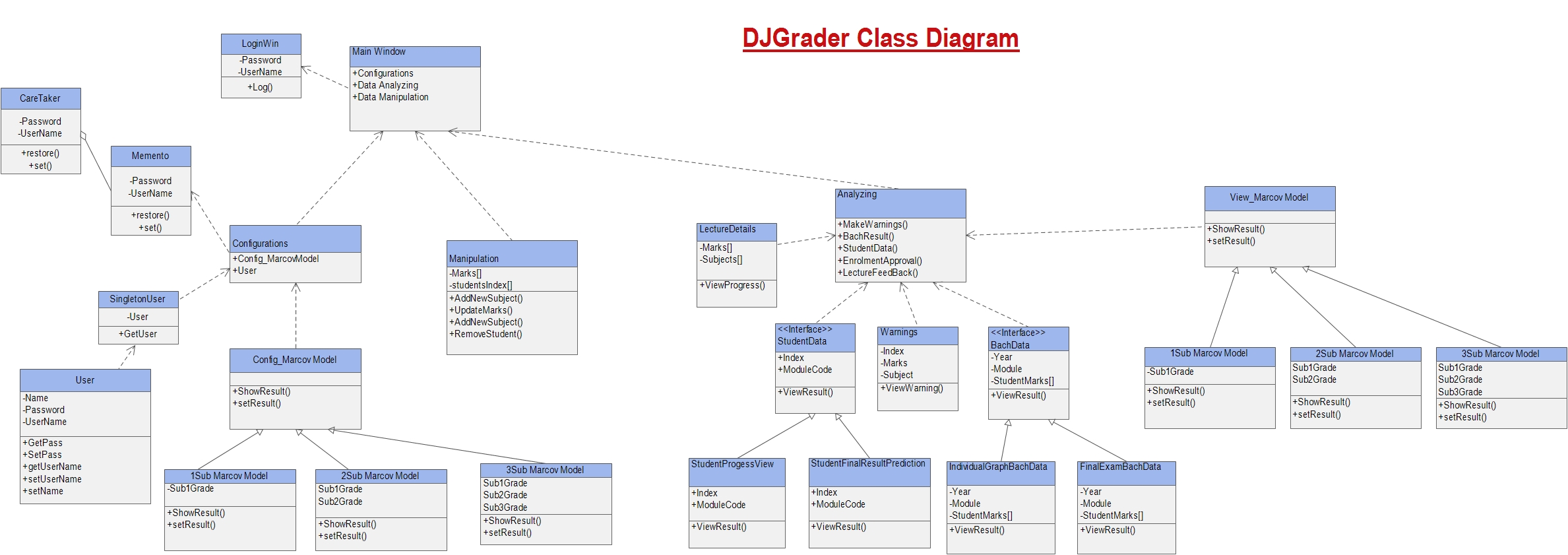
As mentioned previously, the logical view of the architecture is expected to illustrate the functionalities provided to the end users.

## Overview

The following UML 3-tier architecture diagram shows the main layers of the architecture of the application, the packages within each one of them and interactions between these packages.



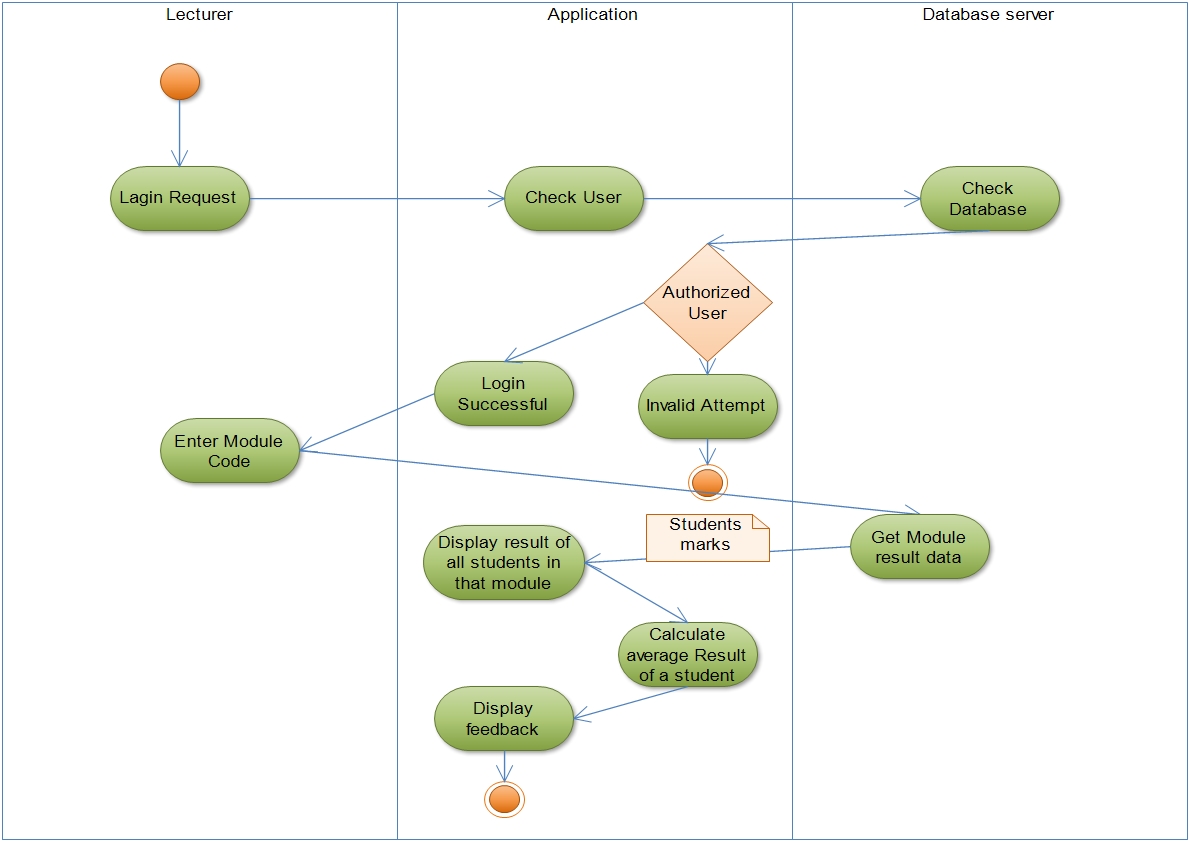
## Class Diagram



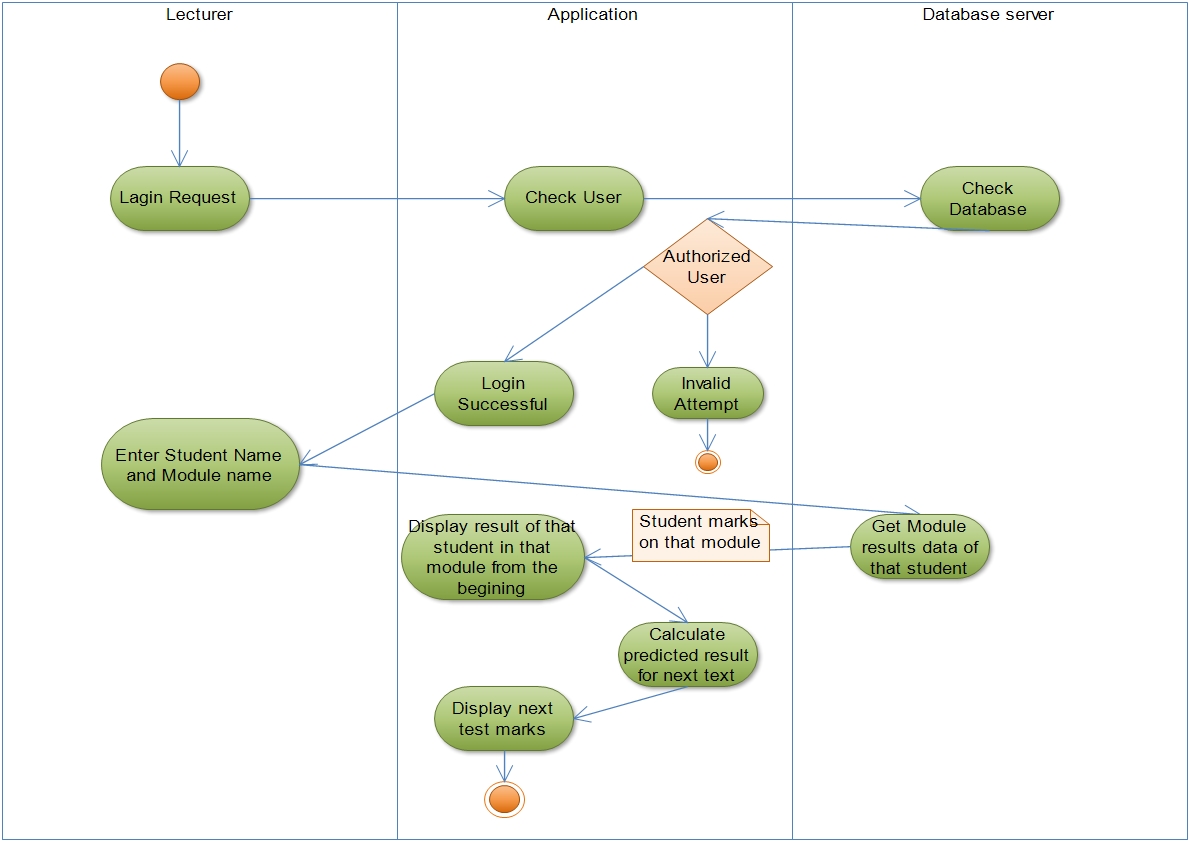
# Process View

Illustrated below are only the processes that get added due to the extension of the existing application. Already existing processes are not shown. Also it is important to note that except for the process that keeps track of the time since the last checkup, there are no other concurrently running processes. This time tracking process will run concurrently with all the other processes a user may start. The application is expected to run one of the following processes at a time depending on the user’s demand.

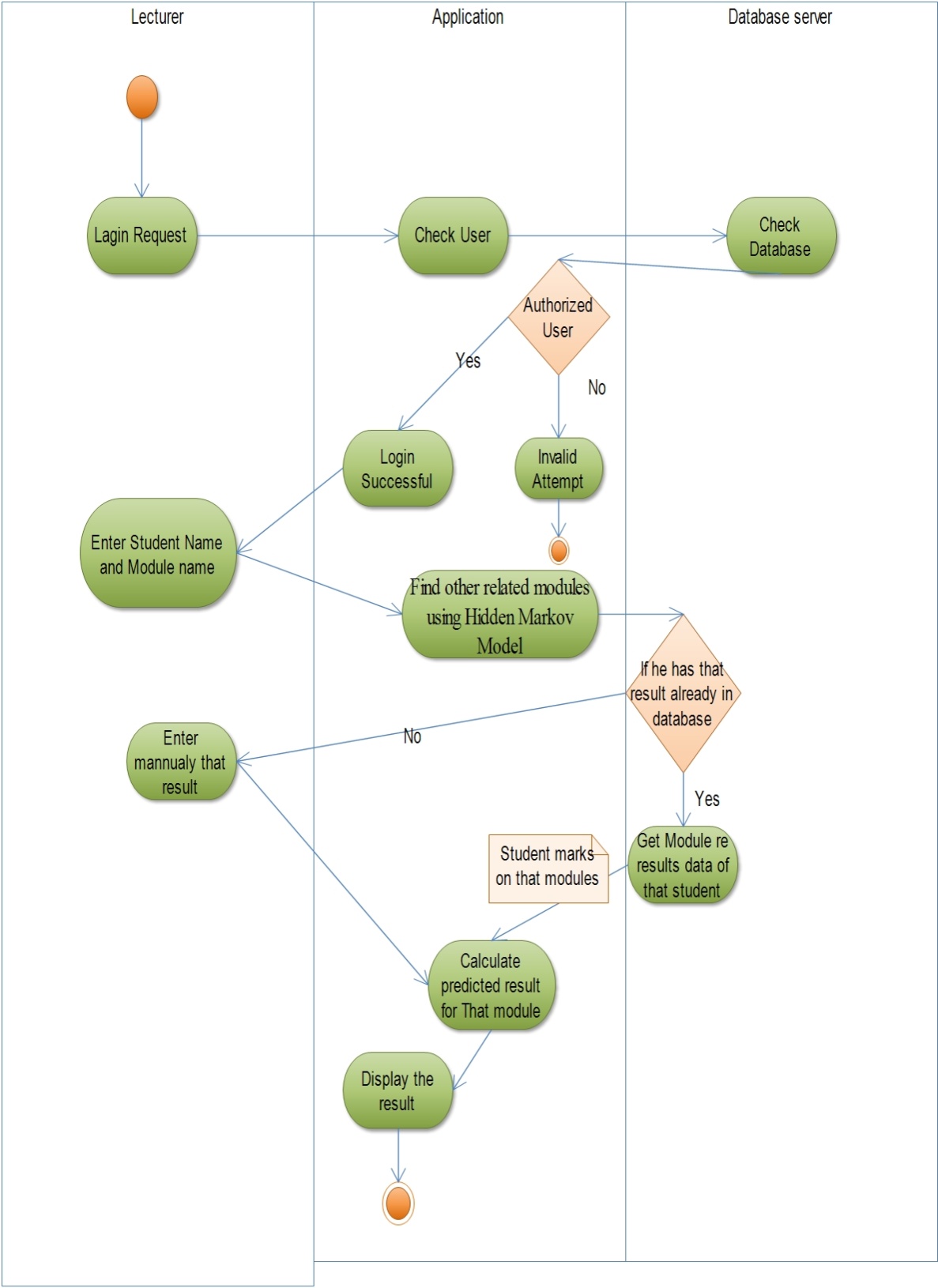
**Get His Lecturing feedback**



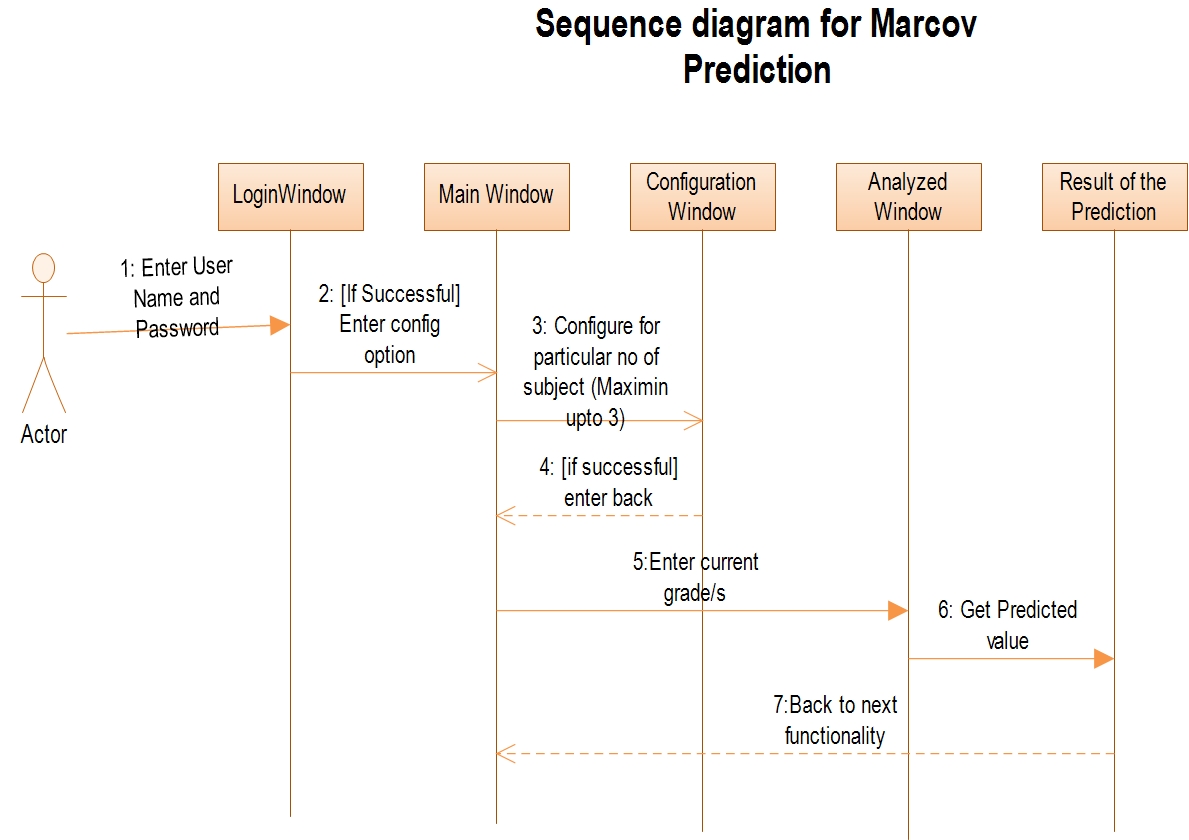
**Get Student’s predicted result for next examination**



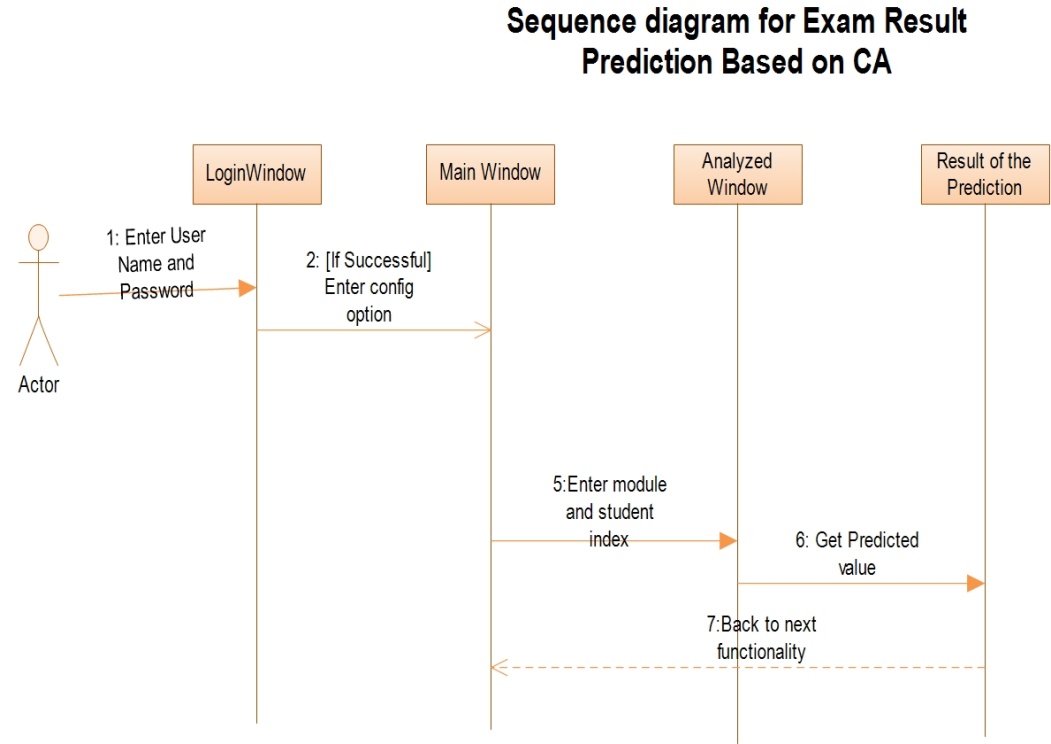
**Approve Student module registration considering past completed module’s result**



## Sequence diagram

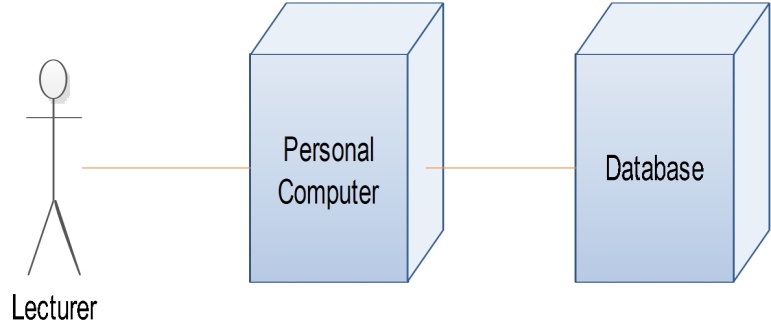


# 

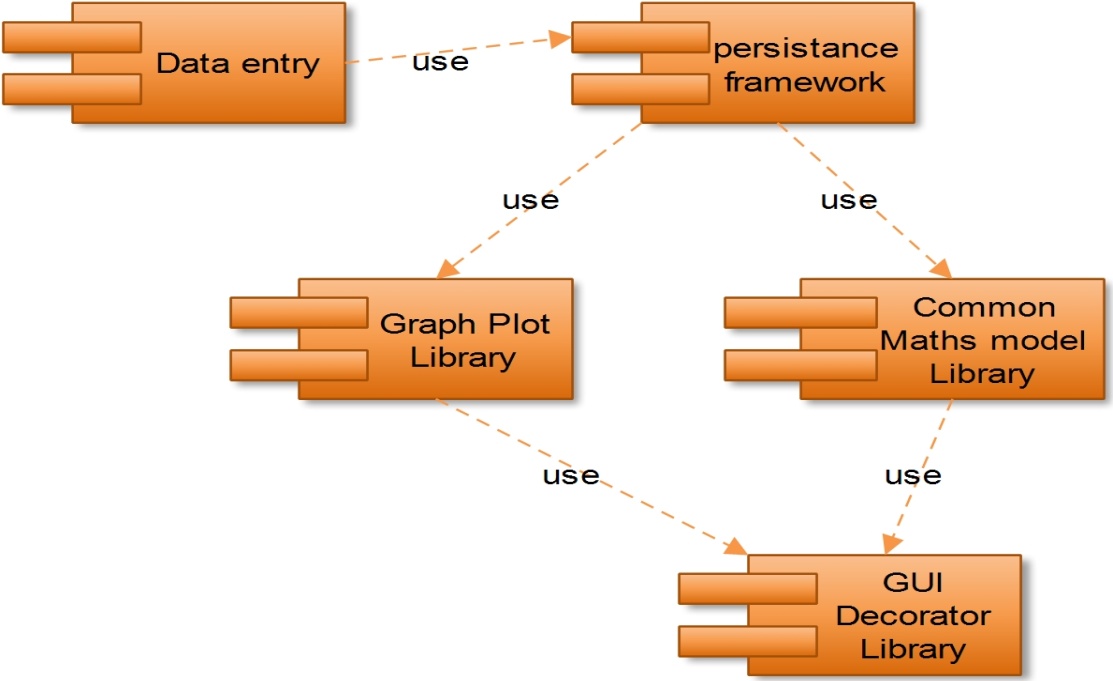


# seq33.jpg

# Deployment View

When mentioning about the deployment view, since this is a standalone system which uses for lecturers personnel uses this system don’t connect with internet. But when entering module data if the data base amount is too big for keeping history of some modules of nearly 100 student within 4 years it is good to have cloud backup system .

# Implementation View

The structure of the implementation model can be understood in terms of the packages of the application, by referring to the logical view of the architecture displayed in [section 5](#_Overview).

The component view of the architecture is shown below.

# Data View

The data view of the system architecture refers to all the details related to the persistent elements of the application. As mentioned previously, a DBMS is used in the application basically to store marks of a students in modules which he lecturing and if that module is useful for later semester module that results should be keeping until that batch is out. And also other pre requests module data should be added for taking enrolment decision. Furthermore, the same database will be used to store authentication details of different users(usernames and passwords) and customization details.

# Size and Performance

Personal computer always have low amount of hardware resources. But architecture design always try to reduce system consumption so that it can achieve maximum perform with limited hardware in the device. Auto deleting unnecessary data when after the academic year using events handing help to effective usage of memory capacity.

# Quality

This software specially considers about the reliability. User will face unexpected events and reduce their marks unusually. By using FFT metrology this system ignores unusual pattern of results and dray the curve fitting

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End of Document\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*