Design and Development Deliverable

For

MBA Student Profile System

(Advanced System Project CIS5690)

Submitted to

Dr. Kerry Henson

Project Management (CIS 5675)

Department of Computer Science and Information System

Submitted by:

Eak Narayan Paudel

Email:exp10910@ucmo.edu

Student Id:700641091

1 Narrative description of your design

In order to address Business problem with in MBA department, I designed and developed the application entitled with”MBA Student Profile Management System”. The main purpose of the application is to track all the student related activities, assigned courses, assigned prerequisites, and concerned advisor . After spending several days in brainstorming, discussion with friends, internet search, studying similar project, I started designing the system.

I am assuming there can be three kinds of users who actively use the system. They are Director (may be head of department), Advisor (any professor) and Student. Director acts as super admin role, which is responsible for adding student, advisor, courses, prerequisites courses. Director creates user credentials for all kind of user. Advisor uses the userid and password provided by Director to login to the system. Director can access the list of student related to advisor’s concentration. Advisor is responsible to approve and comment upon student prerequisites courses. Student can access the system after the email with link send from director. Student should go through questionnaire and read the academic code document. Student response to the questions is reviewed by the Advisor and enter note if required. All students’ course status and approval can be viewed by Director.

I have designed the User Interface in responsive pattern to give both web and mobile feel. I am using encryption security for login credentials. This system is designed to build transparency between advisors and Directors, which can minimize the misunderstanding about courses among students and department.

2 Narrative description of your system

I have designed UML diagram to figure out the workflow of the system

2.1 Class diagram for the system

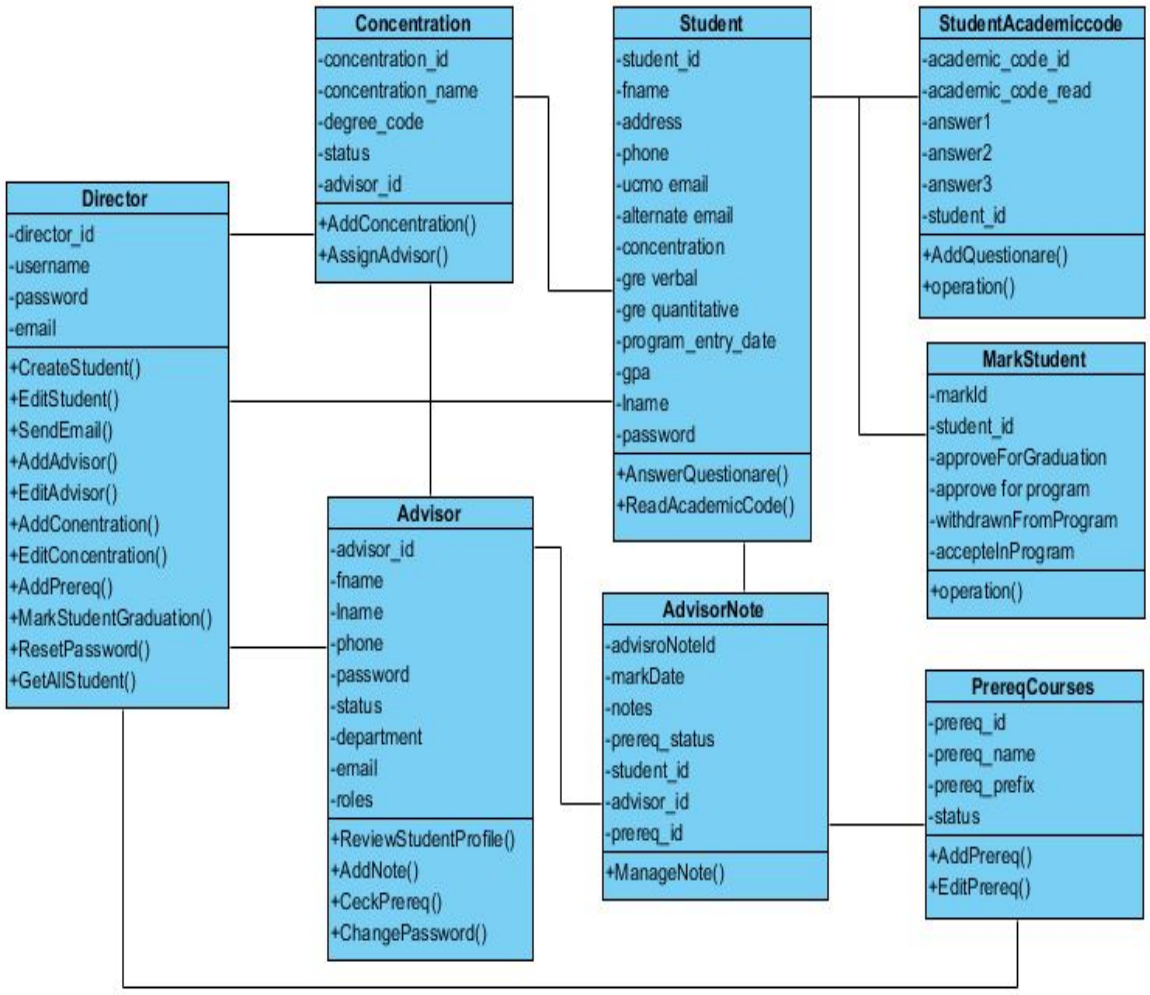


Fig: Class Diagram for MBA Profile System

2.2 Activity Diagram

This Activity diagram is for overall flow of the activity from login to inner page.

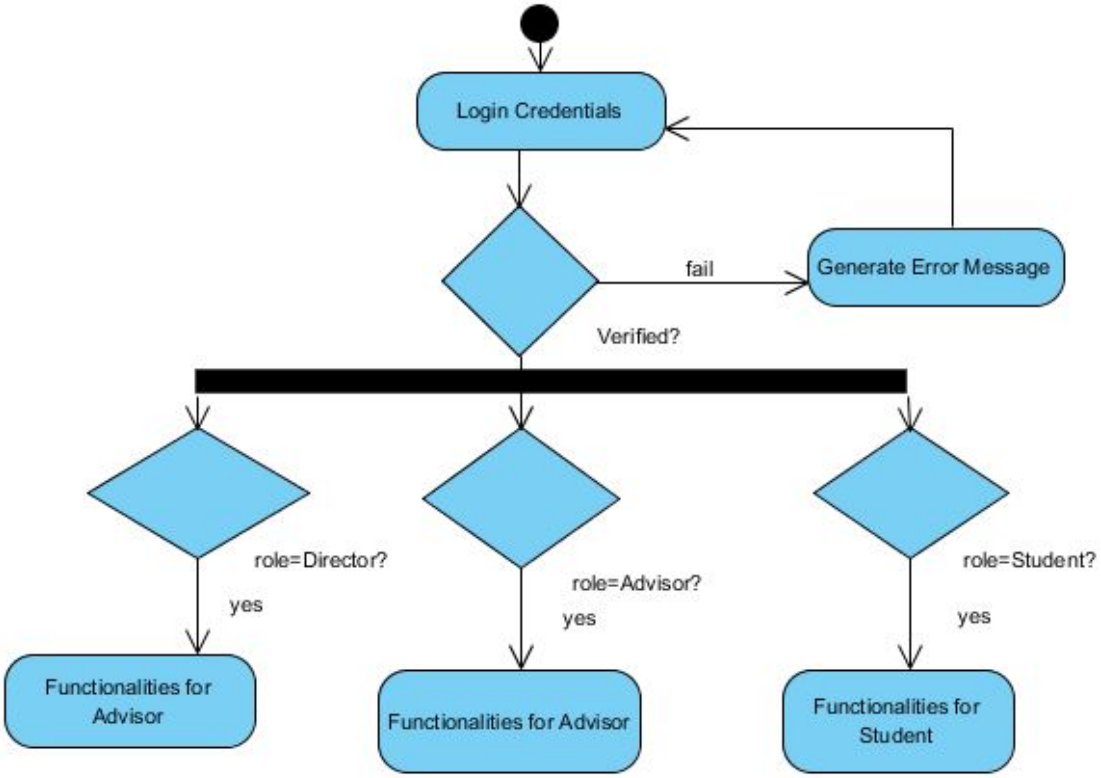


Fig: Activity diagram For MBA Student Profile System

Activity diagram for Director's Role

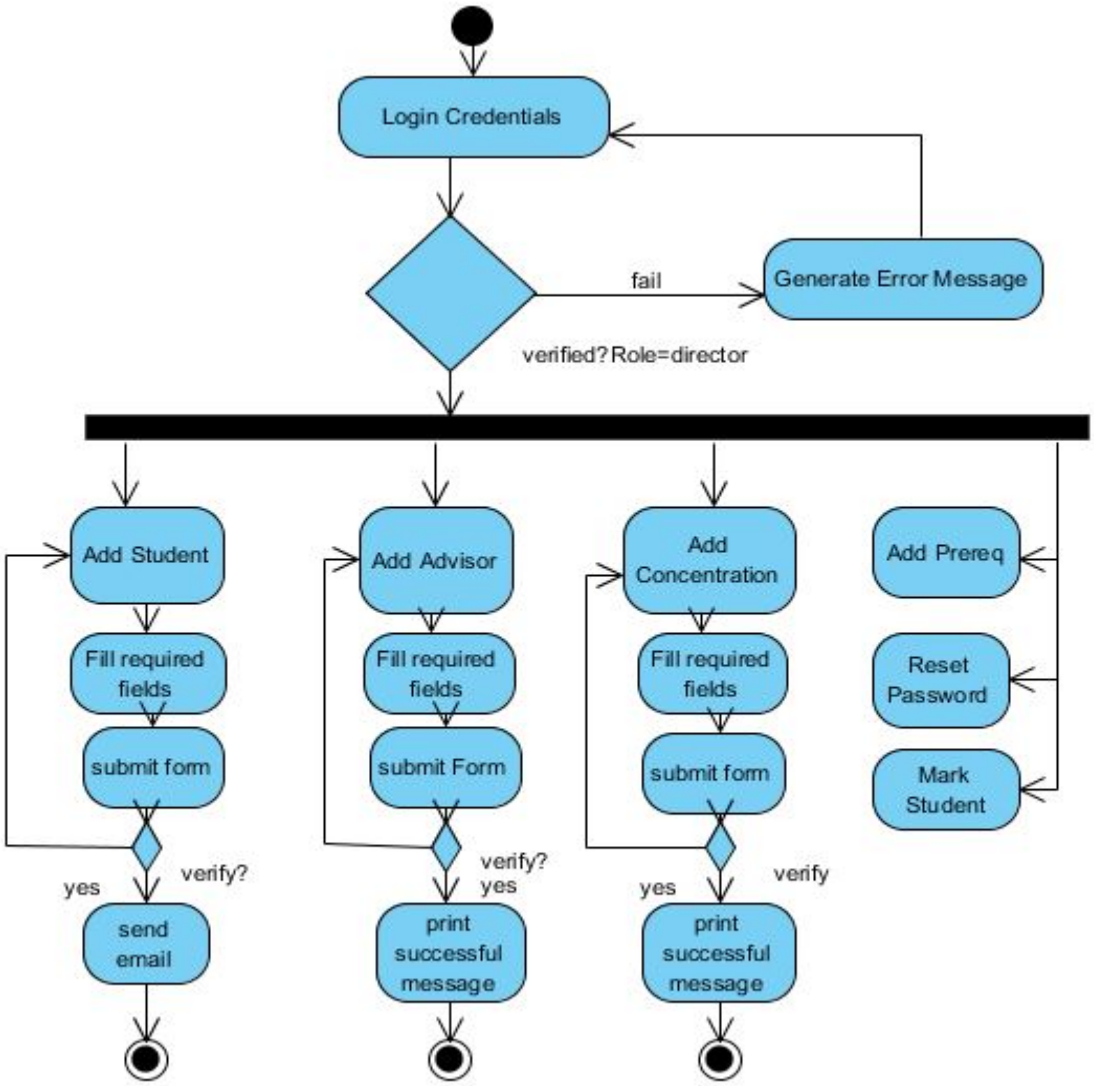


Fig: Activity diagram to understand Director's roles.

Activity Diagram for Advisor's Role

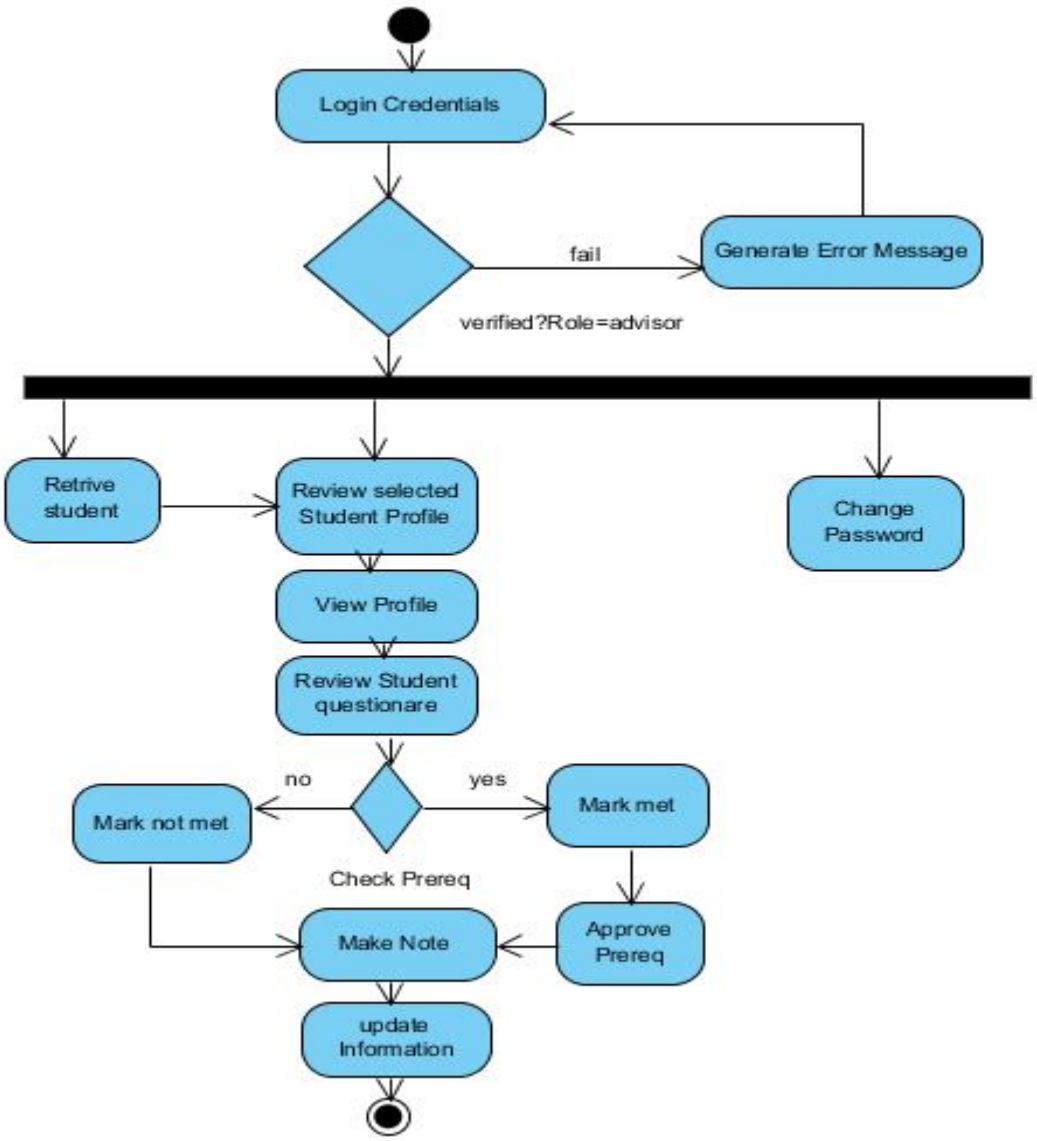


Fig: Activity Diagram for understanding Advisor's role

Activity Diagram for Student's Role

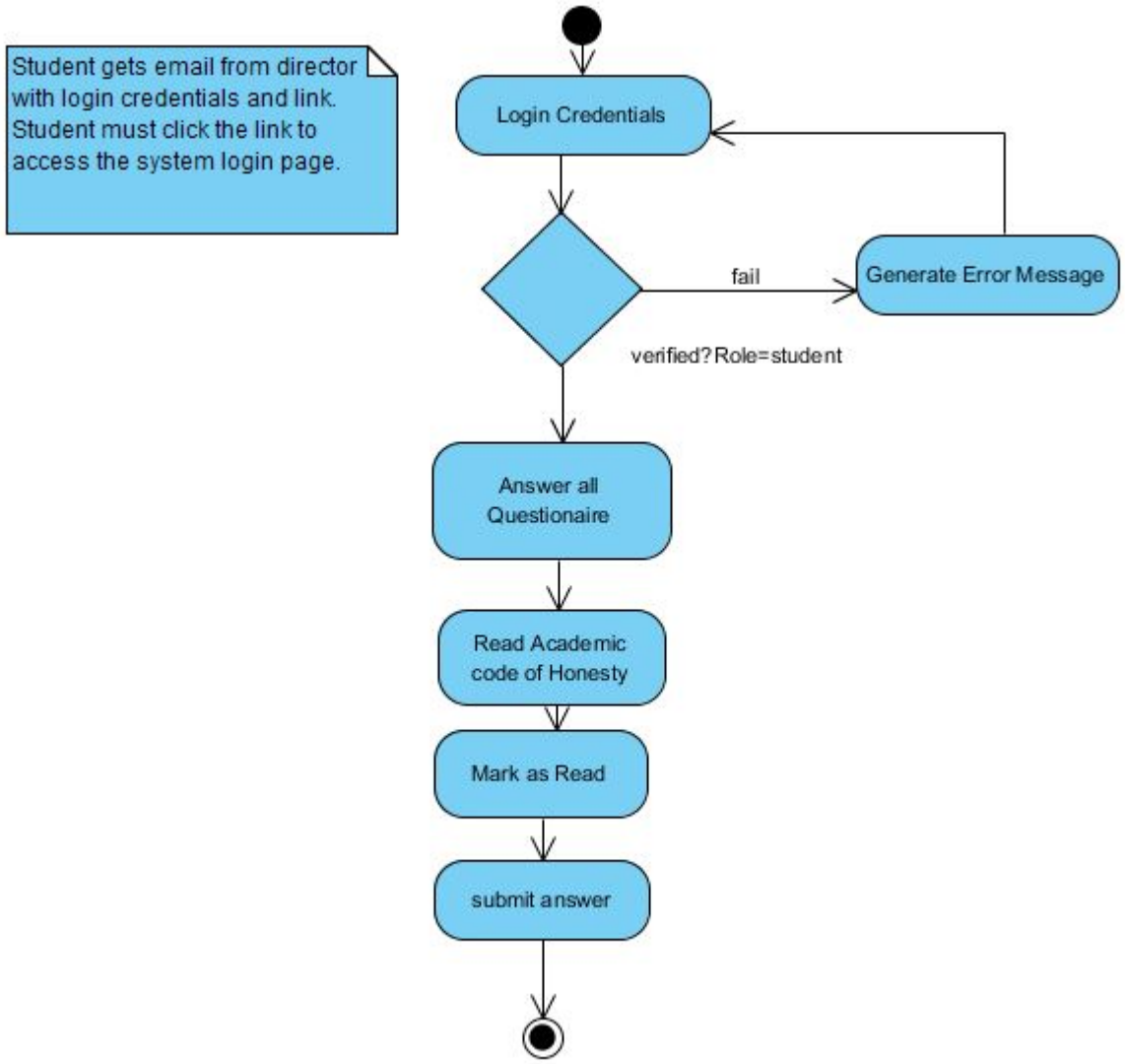


Fig: Activity Diagram to understand Student's Role

2.3 Sequence Diagram

Sequence diagram gives the flow of interaction between objects within the system.

Director's Login

The login process for all users is same. Here, I am representing director as Actor.

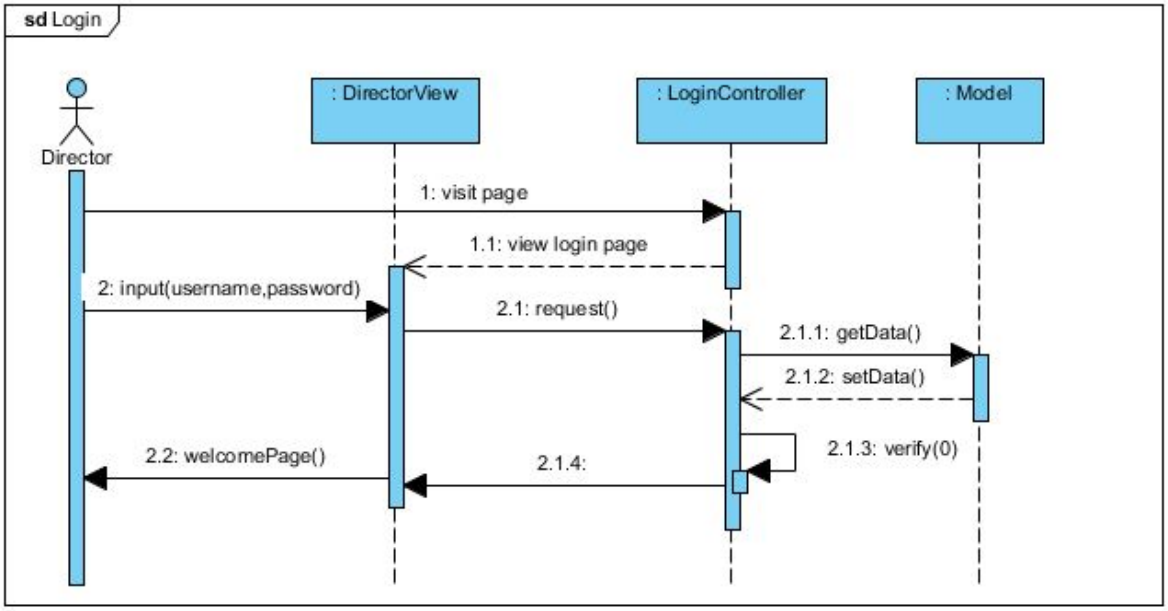


Fig: Sequence diagram for System login

Add Student

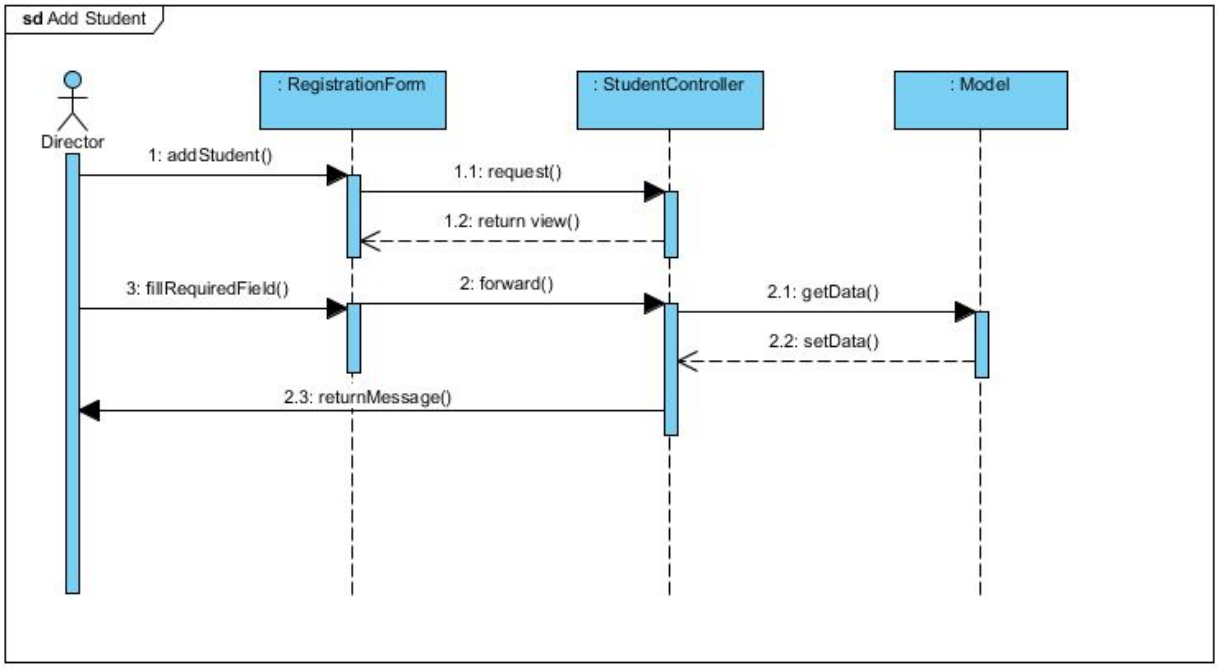


Fig: Sequence diagram to Add Student

Retrieve profile by Advisor

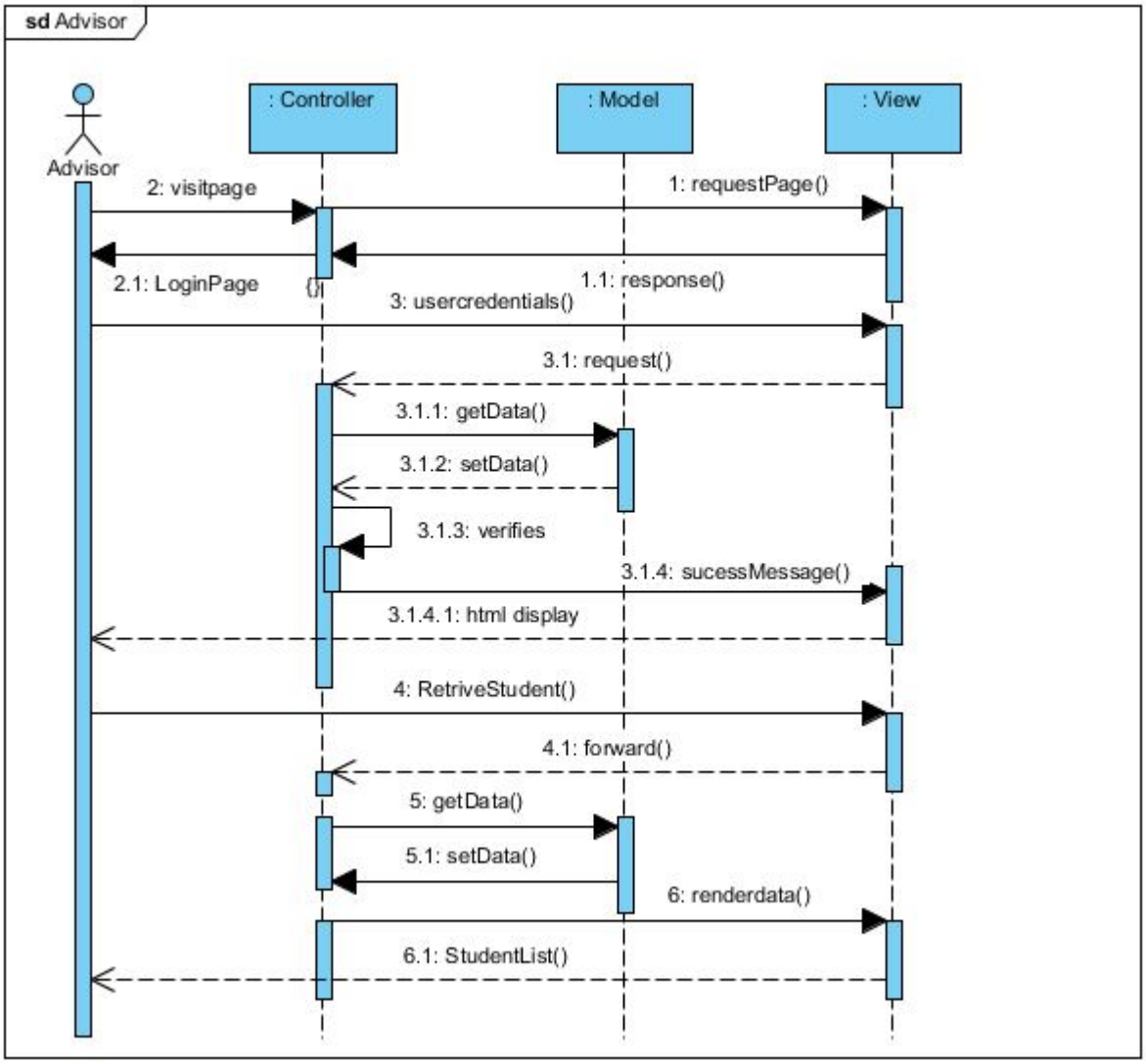


Fig: Sequence diagram for Advisor's role

2.4 Use Case Package diagram

Package Diagram give high level view of the system. In below diagram I am grouping the similar use case in single package related to the actors.

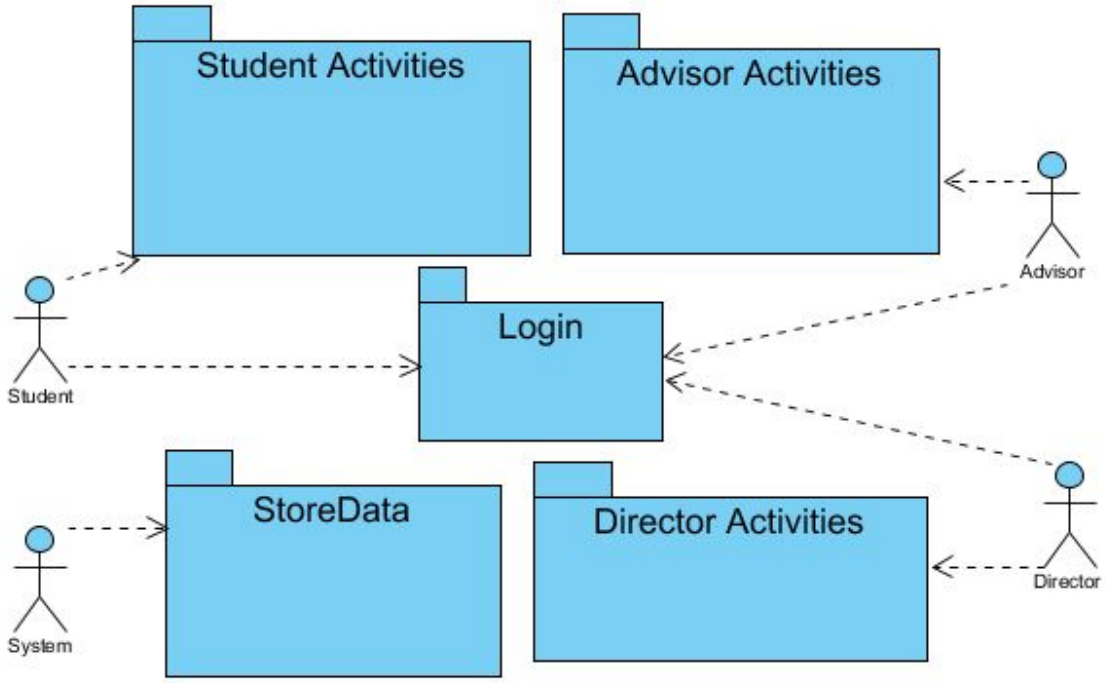


Fig: Use Case Package Diagram for System

2.5 Database Design

I am using MySQL database to store all the information regarding Advisor, Director and Student. I have normalized the table to reduce data redundancy. Primary key is used in all tables to give the unique identification to table element. As I am using hibernate, the entity defined in the entity model is creating own table, so no need to design the database table explicitly in the MySQL.

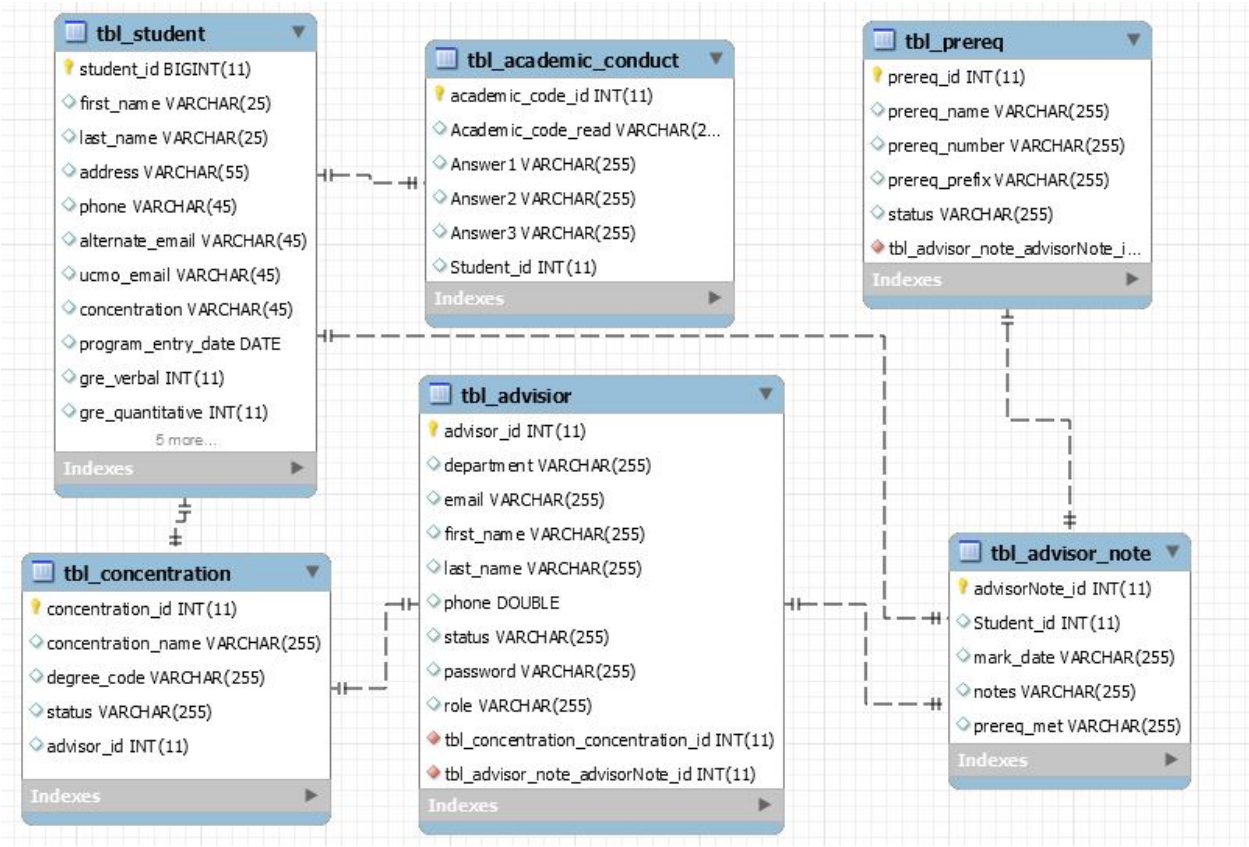


Fig: Schema Diagram for the system

Data Dictionary

There are six different table created in the database. Table entities with their data types and properties are shown below in figures. I have used the snap sort of the table from the database.

Table (tbl_student)

This table stores data for student.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
student_id	BIGINT(11)	☑	☑	☐	☐	☐	☐	☑	☐	
first_name	VARCHAR(25)	☐	☑	☐	☐	☐	☐	☐	☐	
last_name	VARCHAR(25)	☐	☑	☐	☐	☐	☐	☐	☐	
address	VARCHAR(55)	☐	☑	☐	☐	☐	☐	☐	☐	
phone	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
alternate_email	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
ucmo_email	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
concentration	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
program_entry_date	DATE	☐	☑	☐	☐	☐	☐	☐	☐	
gre_verbal	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
gre_quantitative	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
gpa	FLOAT	☐	☑	☐	☐	☐	☐	☐	☐	
password	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

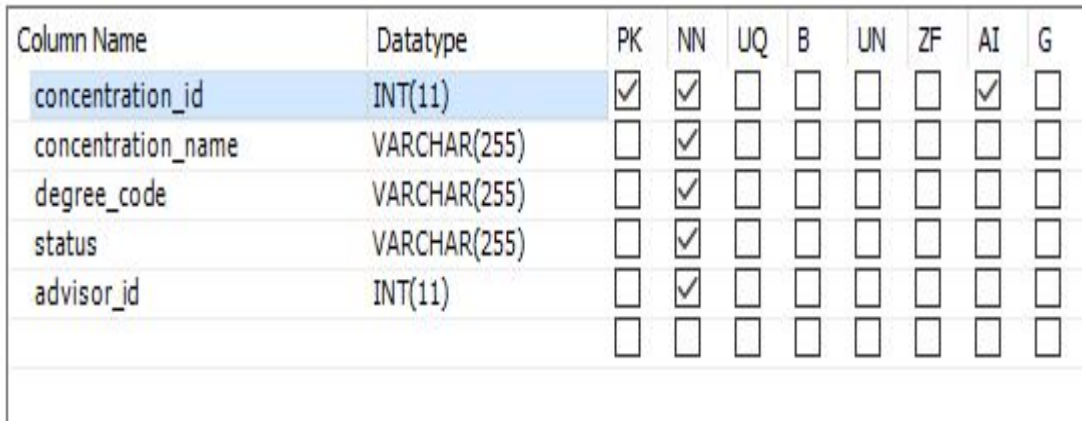
Table(tbl_advisor)

This Table stores the data for both advisor and Director.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
advisor_id	INT(11)	☑	☑	☐	☐	☐	☐	☑	☐	
department	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
email	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
first_name	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
last_name	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
phone	DOUBLE	☐	☑	☐	☐	☐	☐	☐	☐	
status	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
password	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
role	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

Table (tbl_concentration)

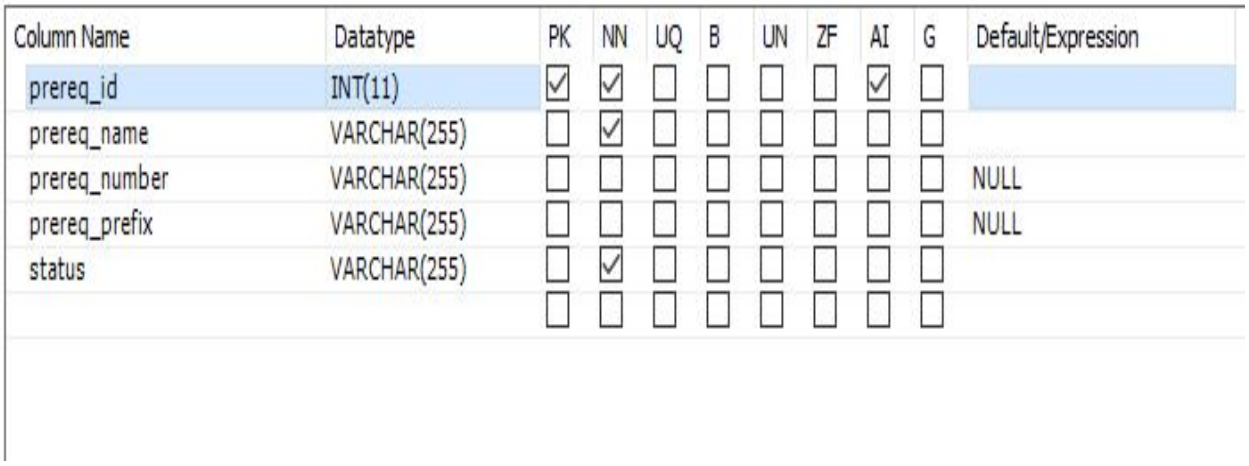
This table stores data for concentration. Advisor ID is used from advisor table to assign advisor to concentration. I am using ID vs ID approach to define the relationship.



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
concentration_id	INT(11)	☑	☑	☐	☐	☐	☐	☑	☐
concentration_name	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐
degree_code	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐
status	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐
advisor_id	INT(11)	☐	☑	☐	☐	☐	☐	☐	☐
		☐	☐	☐	☐	☐	☐	☐	☐

Table (tbl_prereq)

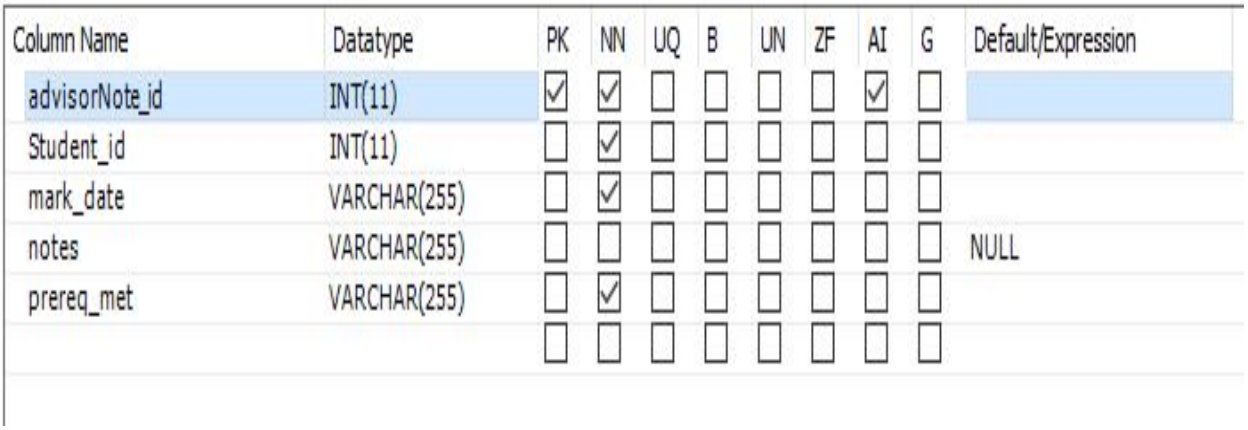
This table Stores the data for prerequisites courses.



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
prereq_id	INT(11)	☑	☑	☐	☐	☐	☐	☑	☐	
prereq_name	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
prereq_number	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
prereq_prefix	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
status	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

Table(tbl_advisor_note)

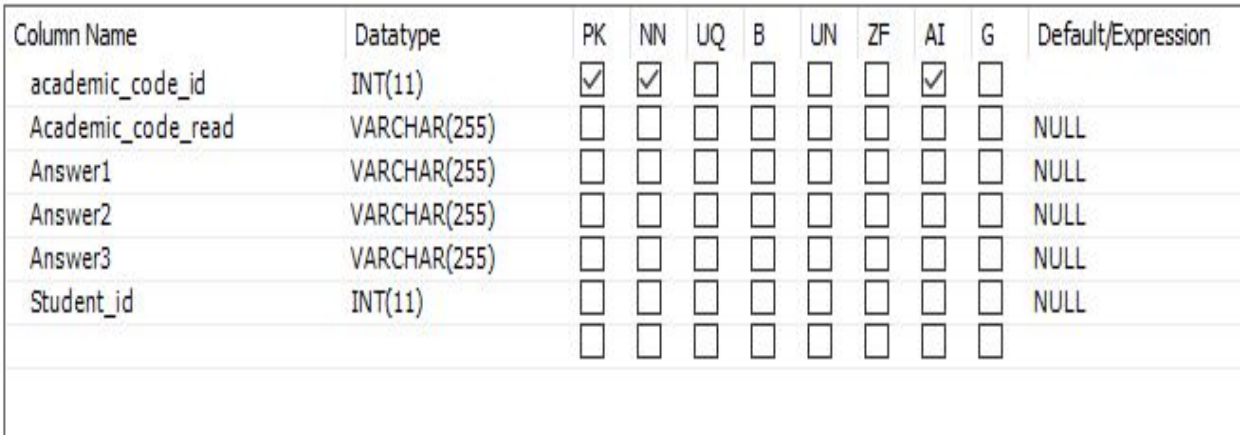
As Advisor approves the prerequisites for the student, updated information is saved in this table with advisor note.



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
advisorNote_id	INT(11)	☑	☑	☐	☐	☐	☐	☑	☐	
Student_id	INT(11)	☐	☑	☐	☐	☐	☐	☐	☐	
mark_date	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
notes	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
prereq_met	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

Table(tbl_academic_conduct)

This table stores the answers submitted by student and Academic code of conduct for each student.



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
academic_code_id	INT(11)	☑	☑	☐	☐	☐	☐	☑	☐	
Academic_code_read	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Answer1	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Answer2	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Answer3	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Student_id	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	☐	

SQL Query

```
CREATE TABLE `tbl_student` (
  `student_id` bigint(11) NOT NULL AUTO_INCREMENT,
  `first_name` varchar(25) DEFAULT NULL,
  `last_name` varchar(25) DEFAULT NULL,
  `address` varchar(55) DEFAULT NULL,
  `phone` varchar(45) DEFAULT NULL,
  `alternate_email` varchar(45) DEFAULT NULL,
  `ucmo_email` varchar(45) DEFAULT NULL,
  `concentration` varchar(45) DEFAULT NULL,
  `program_entry_date` date DEFAULT NULL,
  `gre_verbal` int(11) DEFAULT NULL,
  `gre_quantitative` int(11) DEFAULT NULL,
  `gpa` float DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`student_id`)
) ENGINE=InnoDB AUTO_INCREMENT=700018 DEFAULT CHARSET=utf8;
```

SQL query to create tbl_advisor

```
CREATE TABLE `tbl_advisior` (
  `advisor_id` int(11) NOT NULL AUTO_INCREMENT,
  `department` varchar(255) DEFAULT NULL,
  `email` varchar(255) DEFAULT NULL,
  `first_name` varchar(255) DEFAULT NULL,
  `last_name` varchar(255) DEFAULT NULL,
  `phone` double DEFAULT NULL,
  `status` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  `role` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`advisor_id`)
) ENGINE=InnoDB AUTO_INCREMENT=702 DEFAULT CHARSET=utf8;
```

Sql query to create tbl_concentration

```
CREATE TABLE `tbl_concentration` (
  `concentration_id` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `concentration_name` varchar(255) DEFAULT NULL,
  `degree_code` varchar(255) DEFAULT NULL,
  `status` varchar(255) DEFAULT NULL,
  `advisor_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`concentration_id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;
```

Sql query to create tbl_prerreq

```
CREATE TABLE `tbl_prereq` (
  `prereq_id` int(11) NOT NULL AUTO_INCREMENT,
  `prereq_name` varchar(255) DEFAULT NULL,
  `prereq_number` varchar(255) DEFAULT NULL,
  `prereq_prefix` varchar(255) DEFAULT NULL,
  `status` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`prereq_id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
```

Select Query

SELECT * FROM mbaportfoliosystem.tbl_student;

Insert Query

INSERT INTO `mbaportfoliosystem`.`tbl_student`

(`student_id`,`first_name`,`last_name`,`address`,`phone`,`alternate_email`,`ucmo_email`,`concentration`,`program_entry_date`,`gre_verbal`,`gre_quantitative`,`gpa`,`password`)

VALUES

(<{student_id: }>,<{first_name: }>,<{last_name: }>,<{address: }>,<{phone: }>,<{alternate_email: }>,

<{ucmo_email: }>,<{concentration: }>,<{program_entry_date: }>,<{gre_verbal: }>,<{gre_quantitative: }>,

<{gpa: }>,<{password: }>);

Update Query:

UPDATE `mbaportfoliosystem`.`tbl_student` SET `ucmo_email`='shiva@gmail.com' WHERE `student_id`='700013';

Delete Query:

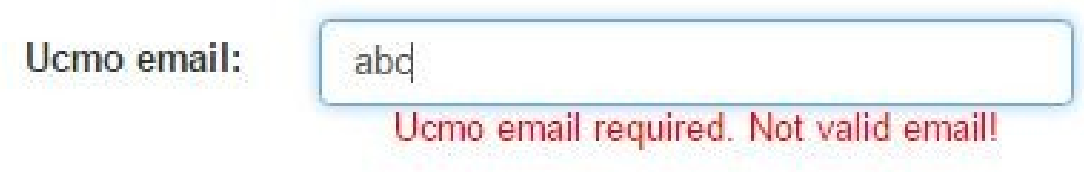
DELETE FROM `mbaportfoliosystem`.`tbl_student` WHERE `student_id`='700013';

2.6 Security Issues

There are several security threats in web application. Issues can be from bad user input, session objects, web services, SQL injection, third party library file and many more. In this project I am more concern about user's login credentials, user's input and session attributes. I am implementing several security approach to obtain high tolerance toward vulnerable.

AngularJS Security: I have used AngularJS for form validation along with HTML form validation. Every User should fill the mandatory field in order to put their request. If the form is not filled, error message is displayed .No empty forms can be submitted. This approach helps to protect from bad user input.

For example, if email is not in email format, error message is displayed as "Not Valid email".



Server side Validation: Server side security is used to validate model data. Although Client side validation is used, there is high risk of malicious user to gain access in server. Use of server side validation protect from such kind of users.

Password Encryption/Decryption: I am using MD5 hash function to encrypt and decrypt the password. Passwords are kept secure by changing it original value.

program_entry_date	gre_verbal	gre_quantitative	gpa	password
NULL	123	213	4	99d077188b3e8f4da041ce6fb5488a8f
NULL	NULL	NULL	NULL	NULL

Fig: how password data are stored in database

2.7 Justification for design

Overall design for the system is satisfactory. Further refinement is needed to increase better performance of the system. UML diagram in document clarifies more about the design. Database Schema diagram give the details design of the database. While designing database, I am avoiding the use of foreign key .I am addressing the relation between entities through the concept of “table vs table” data.

For Example, advisor_id(primary key of tbl_advisor) is used in tbl_concentration(as advisor_id).The relationship is defined in business logic.

3 Development with Testing and Implementation

I took development and designing parallel.

3.1 External Package and Libraries used

This java project used following jar files. All jar files are externally invoked into the system. In this project I have defined all jar files in POM.XML, which manage all the jar files.

1. **Spring framework 4.1.5 RELEASE**
   Used for comprehensive programming and configuration in any kind of deployment platform.
2. **Hibernate Version 4.3.8. Final**
   Powerful, high performance object/relational persistence and query service
3. **Mysql connector 5.1.10**
   Official JDBC drive for MySQL. Use to connect database
4. **Junit -4.11**
   Unit Testing Framework
5. **Jstl-version 1.2**
   Use to bind the data from model
6. **Java 1.8**
   Development environment
7. **Google-api-services-gmail v1-rev-1.22.0**
   Use google gmail server
8. **Mail (javax.mail)**
   Send mail
9. **Commons-codec-1.9**
   Used in password encryption
10. **Maven 2.3**
    Build manager for Java projects
11. **Tomcat 7.0**
    Web server
12. **Bootstrap**
    Used in front end development
13. **AngularJS**
    Used in Form validation

3.2 Testing

Testing is the process of evaluating the application's performance weather it satisfies the user requirements. It helps to identify the loop hole in the program. I started the testing process after

gathering the requirements. After the preliminary research and gathered data, I figure out the first cut of the system. I continuously work on it, test the components and refine it to give the final design. I followed manual testing strategy. As my project does not contain any complex data, I did not use any kinds of tools to test the bug. There are few stages which I followed during Testing.

Unit Testing: Unit testing is the process of testing individual unit of source code. I used to put test data to execute each function.

Integration Testing: In this process bulk of code, were tested to check their functionality. Multiple blocks of code are combine to make component, these component are integrated to build the system. During the Integration Testing is done to ensure the components are functioning correctly.

System Testing: Once all components are combined, system testing is done. System testing is very important to make the standard application.

User acceptance Testing: I ask my friend to use the application. He used the application and suggest be to fix the bug. During this test:

- Some of the hyperlinks were not functioning
- One SQL Query was returning null value.
- Exception was not handled in Login page.

After testing I change the code to resolve the generated errors.

Testing for Student Registration

Case	Description	Expected Results	Actual Results	Passed

Number				
1.	Enter empty for First Name and submit	Display error message ”First name required”	Displays error message ”First name required”	Yes
2.	Enter empty for last name	Display error message ”Last name required”	Displays error message ”Last name required”	Yes
3.	Enter empty for email address	Display error message ”email required”	Displays error message ”email required”	Yes
4.	Enter text in email address field	Display error message “email format did not match”	Displays error message “email format did not match”	Yes
5.	Enter empty for mailing address	Display error message “address required”	Displays error message “address required”	Yes
6.	Do not Select concentration	Display error message “ select concentration”	Displays error message “ select concentration”	Yes
7.	Enter empty for GRE Verbal	No error	No error	Yes
8.	Enter empty for GRE Quant	No error	No error	Yes
9.	Enter empty for GPA	Display error message “ GPA required”	Display error message “ GPA required”	Yes

10.	Submit empty form	Display error for all required field	Error message is displayed in all required field	Yes

For Advisor Registration

Case Number	Description	Expected Results	Actual Results	Passed
1.	Enter empty for First Name and submit	Display error message ”First name required”	Displays error message ”First name required”	Yes
2.	Enter empty for last name	Display error message ”Last name required”	Displays error message ”Last name required”	Yes
3.	Enter empty for email address	Display error message ”email required”	Displays error message ”email required”	Yes
4.	Enter text in email address field	Display error message “email format did not match”	Displays error message “email format did not match”	Yes
5.	Enter empty for mailing address	No error	No error	Yes
6.	Empty for Department field	Display error message “ enter Department name”	Displays error message “ enter Department name”	Yes
7.	Enter empty for	Display error ”password required”	Displays error	Yes

	password		”password required”	
8.	Enter empty for Confirm password	Display error ”password required”	Displays error ”password required”	Yes
9.	Enter different value in password and confirm password field	Display error message “ password did not match	Displays error message “ password did not match	Yes
10.	Do not select active/Inactive	Insert default value as ”active”	value =active	Yes
11.	Do not select role	Display error” must be selected”	Displays error” must be selected”	Yes
12.	Submit empty form	Display error for all required field	Error in all required field	Yes

For Adding Concentration

Case Number	**Description**	**Expected Results**	**Actual Results**	**Passed**
1.	Submit empty	Display error message in all	Error message is	Yes

	form	required field	displayed in all required field	
2.	Enter empty for Degree Code	Display error message ”concentration code required”	Displays error message ”concentration code required”	Yes
3.	Do not select status	Insert default value as “active”	Value=active	Yes
4.	Do not select advisor	Display error message “ must be selected”	Error! Must be selected	Yes

For Adding Prerequisites courses

Case Number	**Description**	**Expected Results**	**Actual Results**	**Passed**
1.	Submit empty form	Display error message in all required field	Error message is displayed in all required field	Yes
2.	Enter empty Prereq Name	Display error message ”prereq name required”	Error message ”prereq name required” is sseen in screen	Yes
3.	Enter empty prereq prefix	Display error message ”prefix name required”	Error message ”prefix name required” is seen	Yes
4.	Enter empty for prereq code	Display error message “ prereq code required”	Error message “prereq code required is seen”	Yes

5.	Do not select status	Insert default value as “active”	Value=active	Yes
6.	Do not select advisor	Display error message “ must be selected”	Error message is displayed in all required field	Yes

To change Password

Case Number	**Description**	**Expected Results**	**Actual Results**	**Passed**
1.	Submit empty form	Display error message in all required field	Error message is displayed in all required field	Yes
2.	Enter empty New Password	Display error message ”password required”	Error message “password is required ”is seen in screen	Yes
3.	Enter random text in confirm Password	Display error message ”password miss match”	Displays error message ”password miss match”	Yes
4.	Enter password	Password must be visible as *****	*******	Yes

3.3 Implementations

To deploy the system in other machine following specification is required.

Hardware Specification:

Operating system: No later than windows 7

System memory: Minimum 40 MB

Internet Access: Need Internet to Access

Software Requirement:

IDE: Eclipse

Framework: Spring

Database: MySQL, Hibernate

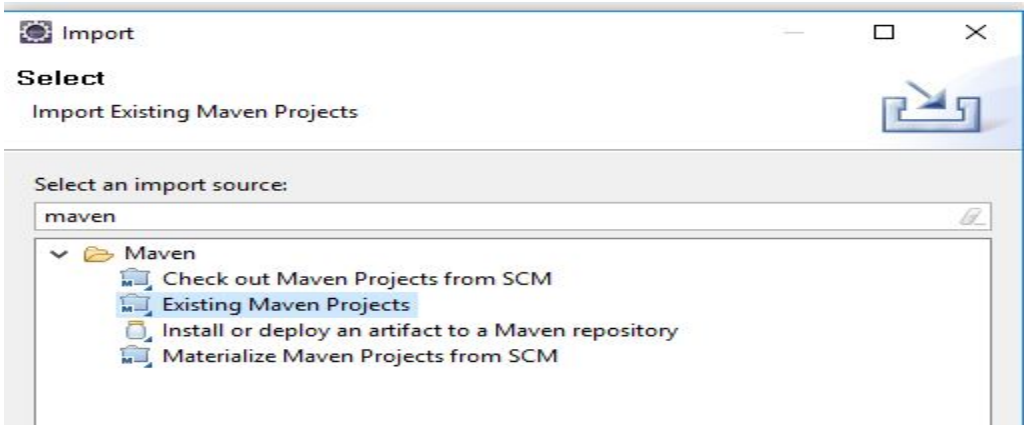
Server: Apache tomcat 7.0

Web Browser: Any (recommended Google chrome)

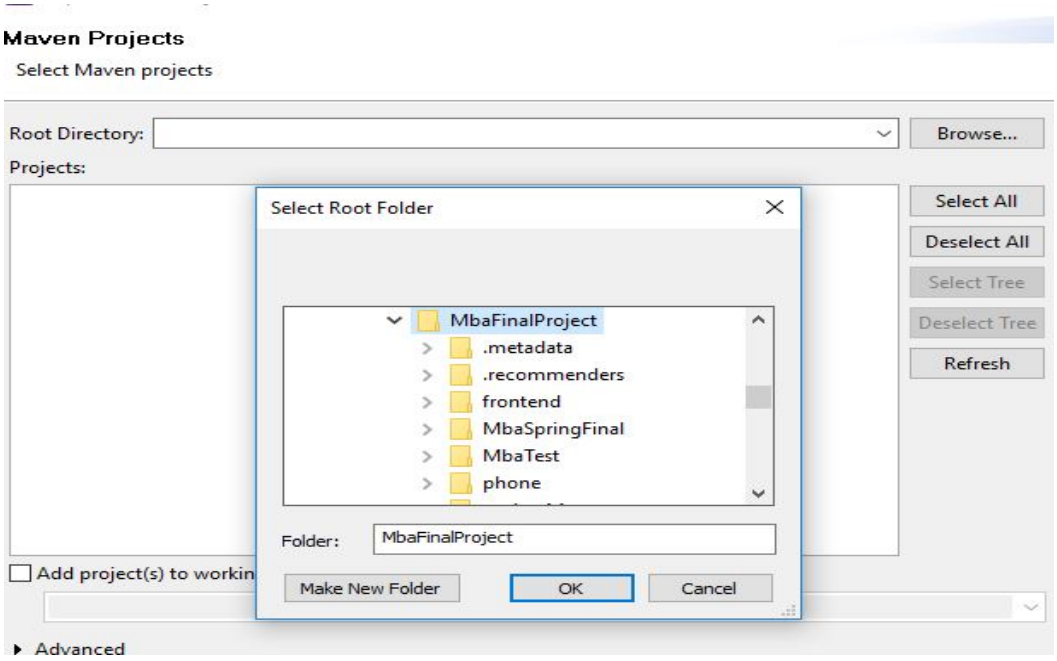
Steps to deploy in the system

1. Install the Eclipse IDE in your system (recommended version- mars)
2. Open Eclipse
3. Click File> Import

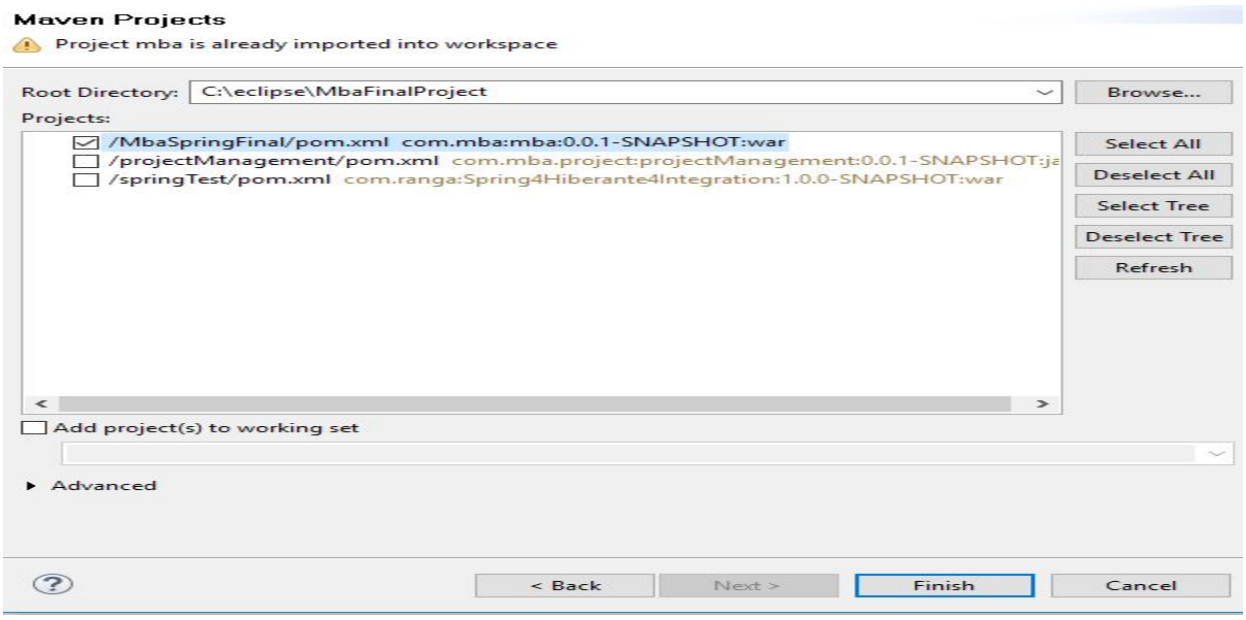
4. Type **Maven** in the search box under **Select an import source:**



5. Select **Existing Maven Projects**
6. Click **Next**
7. Click **Browse** and select the folder that is the root of the Maven project (probabl

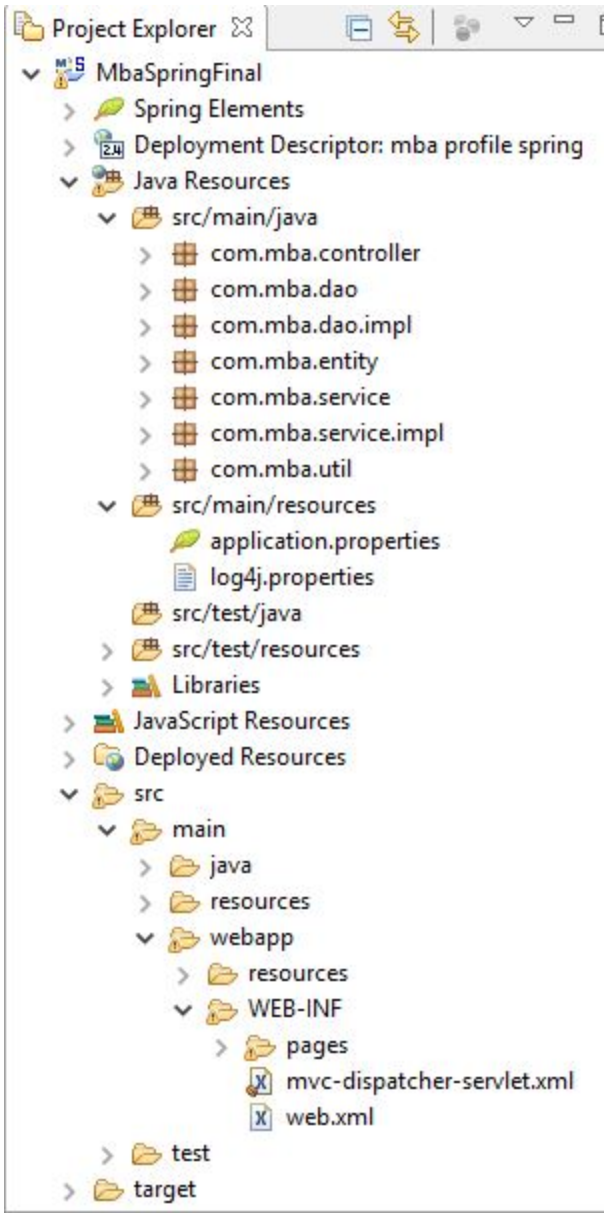


8. contains the pom.xml file)
9. Click **OK**
10. Under **Projects:** click the **checkbox** next to the pom.xml file



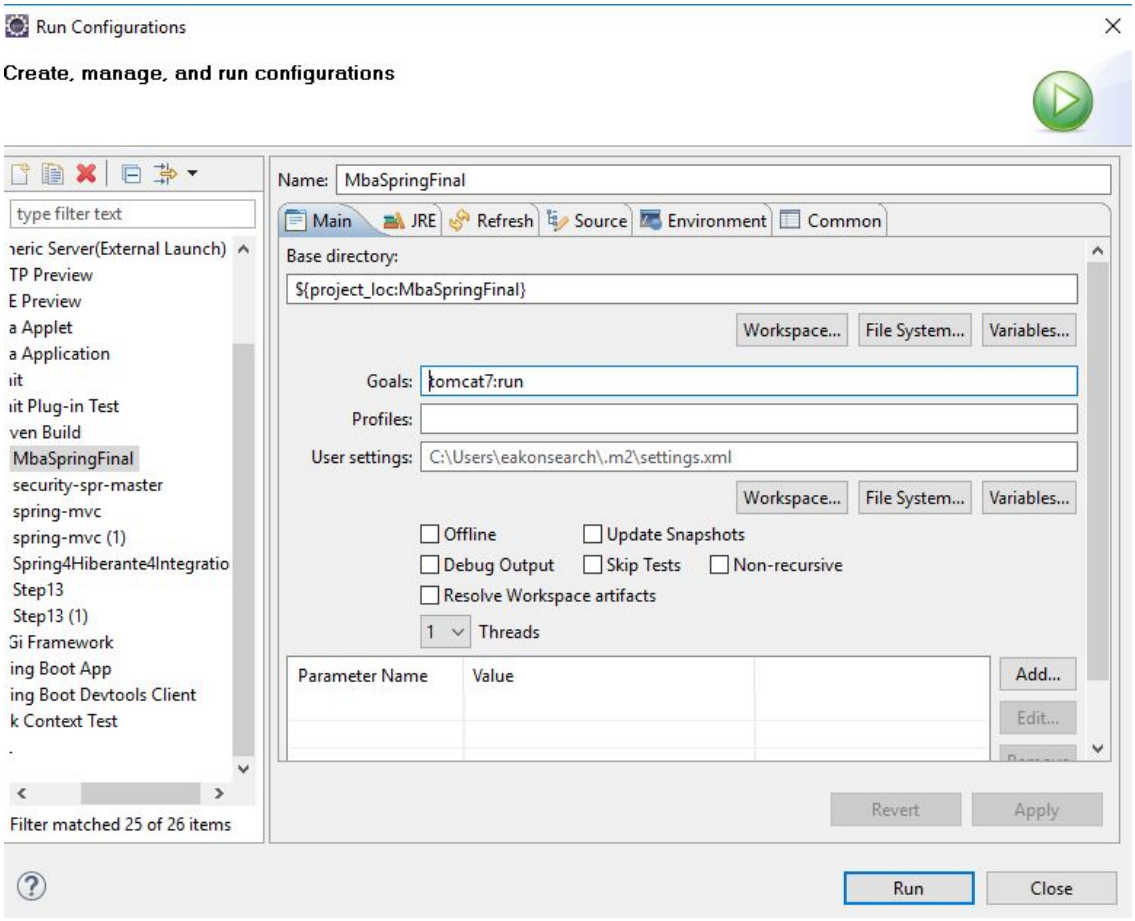
11. Click **Finish**

Project Folder Structure for the project



To execute the program

Right click on project, Click run as, select Run configuration and type “tomcat7:run” in goal and press run



4 Summary

4.1 Problematic Issues

1.0 Difficulties during setting the system (Configuration Issues): It was really very hard to start the project. As I was in early stage of learning spring, Hibernate, and MVC pattern, I struggled for long time to configure the system. As java is open source software, it uses many external files to operate. The files version was not matching the system. System refused to run the application on server.

Steps taken to solve the issues:

1. Download the sample spring template.
2. Create new Maven Project.
3. Write all required dependencies jar files in pom.xml.
4. Import the sample template in maven build project.
5. Right click the properties of project, go to project facts, change run time to Apache tomcat (as I was using tomcat 7.0 as server)
6. Run the project as maven build.
7. Program executed successfully.

2.0 Configuration in XML: Communication between different layers is managed by xml. I need to define each entity and controller in xml in order to execute the program. Small error in the xml file fails the execution of program.

Solution approach: After some research, I found @Annotation is best alternatives to xml. Just few words in the same page was mapping to other file. No need to write more xml configuration.

3.0 Generic queries: As I am using Generic hibernate queries, I was able to use predefined general queries. I got stocked when I need to write customized queries.

Solution approach: One solution was implementing JPA over hibernate. But I tried different approach and it really worked. I fire the query in DAO implementation file using Data Source. I manually get and set the entity model.

The below figure is solution approach that I Implemented:

```
@Autowired
DataSource dataSource;
public DataSource getDataSource() {
    return dataSource;
}
public void setDataSource(DataSource dataSource) {
    this.dataSource = dataSource;
}
@Override
public Student isValidUser(Integer UserId, String password){
    Student s=new Student();
    String query="select * from tbl_student where student_id=? and password=?";
    try{
    PreparedStatement ps=dataSource.getConnection().prepareStatement(query);
    ps.setInt(1, UserId);
    ps.setString(2,password);
    ResultSet rs=ps.executeQuery();
    if(rs.next())
    {
        String firstName=rs.getString("first_name");
        Integer student_id=rs.getInt("student_id");
        s.setStudent_id(student_id);
        s.setFirst_name(firstName);
        return s;
    }
```

4.0 Null Pointer Exception: Every time I run the program, null pointer exception was first error to publish in console. Some time it took me a whole day to figure out the error. Although all the code was fine, null pointer exception was continuously taking my time.

Solution Approach: There are several approaches to figure out this problem:

1. @Autowired annotation was missing in "studentServiceImpl" file. solved
2. No data was passed from form to controller. Name mismatch
3. Need to restart the server, even eclipse, some time machine.

4.2 Evaluation

How would you approach this project differently next time?

Regarding this project, I tried my best to develop fully functional application. The technology I use was new for me. I tried to use current market trends and approaches to develop the system. But in some case I am not able to implement the functionalities which should be written in different way. Next time definitely I will have different approach and methodology to complete this project.

1. I will set up all required configuration files, so that it will not create problem later.
2. I will be using generic functions to implement the hibernate queries. I will try to implement JPA with hibernate.
3. As, I found entity is more important in framework, I will spend more time in designing entity models, database and inter component relations.
4. I will be using spring security for login security.
5. I will focus in more responsive and interactive user interface. I will be using AngularJS for validation and data modeling.
6. I will try to find the standard approach to write the code.

Describe what you have learned from this project experience

This project “MBA Student Profile System” gave me real time experience in software development. I understand the way of building the good software. The best thing I learn is the flow of the phases and process which had significant role through the development. Designing is the vital phase in development, better your design part, better your project what I feel through this project. Changes cannot be restricted but should be handling, I learn through the implementation phase.

I got time to learn new technology. The concept of spring MVC is really magical, which let me to code in different label. The availability of external jar files, way of invoking them in the system, single word used in mapping two class made be exciting to learn more. Using APIs , configuration files, helped be for easy setup. I got clear concept of Model, View, and Controller. Hibernate is other technology which I implemented in the project. It provides data persistency, high performance and creates database. I knew about generic query, their purpose and use.

I learn about AngularJS, and purpose of using it. I tried to implement it my project, which is motivating me too use in further projects. I really enjoyed angular features and functionalities.

Hence, with the technology, I got really good experience about software development life cycle.

References

Alversion. “UML sequence diagram for process view ”. Spring,2007, Accessed November 08, 2016. http://courses.cs.washington.edu/courses/cse403/07sp/lectures/SequenceUML.pdf

Mysql.”mysql work bench”. Downloaded from http://www.mysql.com/products/workbench/

Visual Paradism.”Download Visual paradism tool ”. Download from https://www.visual-paradigm.com/solution/freeumltool/?

gclid=Cj0KEQiA9ZXBBRC29cPdu7yuvrQBEiQAhyQZ9GJAparFyNpREcX1fLt_w_v2TYvFqZ1zM65otBftWUoaAiz38P8HAQ