|  |
| --- |
| University of Houston Enterprise Systems – Application Developer position |
| Appointment System |
|  |

|  |
| --- |
| Krishna Madhuri Kompella  2-22-2017 |

## **Document Purpose**

The document address the requirements implemented with the design assumptions and constraints of the system.

## **Requirements Addressed**

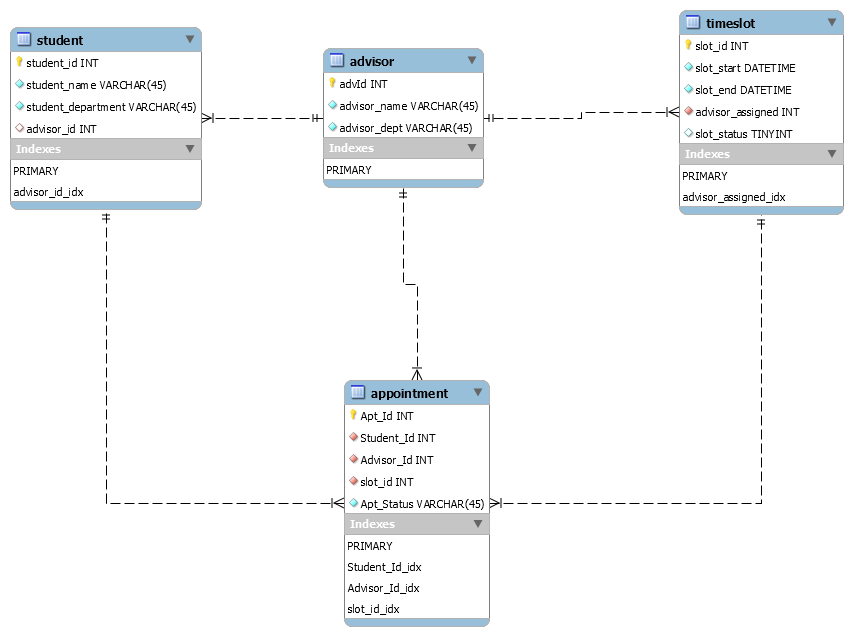
|  |  |  |
| --- | --- | --- |
| # | Requirement | Notes |
| req1 | Student needs to be able to schedule appointments with an advisor | An advisor shall be assigned to the student prior to this implementation |
| req2 | Advisor needs to be able to schedule appointments with a student | The advisor shall have the assigned student Id information. |
| req3 | Student needs to be able to cancel, modify and view his/her appointments | Student shall have the privilege to perform modify operation on the appointments made |
| req4 | Advisor needs to be able to cancel, modify and view his/ her appointments. | Advisor shall be shared with the privilege to modify appointments made. |

## **Modules**

The entire system is implemented in two modules.

* Student -> Implementing req1 and req3
* Advisor -> Implementing req2 and req4.

Database Schema:



**Database Tables:**

* Student Table: Stores information related to the student entity.
* Advisor Table: Stores information related to the advisor entity.
* Appointment Table: To store the appointment details and appointment status: Upcoming or Completed is updated by the advisor.
* TimeSlot Table: Stores the timeslot information of advisor.
* The attribute slot\_status in the table stores 0 and 1 value where 0 signifies that the slot is blocked and 1 is for slot availability.

## **Detailed Design**

**Assumptions:**

Every student is assigned with an Advisor.

Every Advisor have time slots assigned for advising.

The time slots are updated periodically.

**Implementation Package Structure:**

* App Package: The classes with main method to trigger the respective modules.

**Main Class:**

* + Student Client class to perform CRUD operations on Appointment accessible with studentId as first step to enter the system.
  + Advisor Client class to verify the advisor Id and make the CRUD operations accessible.
* DAO Package: The classes for interacting with the database and store the result set. The student and advisor functionalities are implemented in this package and returned the values to move across the system module.
* DTO Package: The classes to store the object values returned from result set and to use within the method calls.
* Exception: The exception classes of Invalid Student Id and Invalid Advisor Id are organized under this package.
* Service: The service interfaces required to implement the student and advisor modules are provided in this package.
* Impl: This package consists classes implement the service interface.
* Connection: The class to establish database connection is implemented with getConnection method in this package.

## **Unit Tests**

The Junit Test suite to test the functionality implemented in DAO package. The test suite runs the StudentDAOTest and AdvisorDAOTest. The package daoTest consists of the test classes.

## **Tools and Languages used**

* Java | Eclipse IDE kepler
* MySQL
* Junit