**2013**

Scheduler Developer’s Manual



Dipen Patel (Scrum Master)

Shalin Banjara (Team Member)

Sanket Patel (Team Member)

Devraj Valecha (Team Member)

Ruby Verma (Team Member)

Sonny Kr (Team Member)

\sonny \kr

Sonny Kr (Team Member)

Contents

[Executive Summary 3](#_Toc374130001)

[Problem Statement 3](#_Toc374130002)

[Proposed Solution 3](#_Toc374130003)

[High Level Deliverables 3](#_Toc374130004)

[Features 4](#_Toc374130005)

[Project Architecture 6](#_Toc374130006)

[High Level Architecture Diagram 6](#_Toc374130007)

[Component Diagram 7](#_Toc374130008)

[Use Cases 8](#_Toc374130009)

[Client Onboarding 8](#_Toc374130010)

[Client Onboarding 10](#_Toc374130011)

[Appointment Schedule 11](#_Toc374130012)

[Meeting Official 12](#_Toc374130013)

[Authentication 14](#_Toc374130014)

[Announcement and Notification 15](#_Toc374130015)

[FAQs 16](#_Toc374130016)

[Database 18](#_Toc374130017)

[SQL Script 18](#_Toc374130018)

[Database Diagram 30](#_Toc374130019)

[Classes 31](#_Toc374130020)

[Web Application 31](#_Toc374130021)

[Class Diagrams 31](#_Toc374130022)

[Object Diagram 33](#_Toc374130023)

[Sequence Diagrams 37](#_Toc374130024)

[State Diagram 42](#_Toc374130025)

# Executive Summary

## Problem Statement

Very long queue ups in front of Government organizations and places like college enrolment department is a normal scene even in this 21st century, even when people uses computers and internet to do each and everything. Even after standing in such long queues to schedule an appointment, people has to wait hours to meet the counsellor, hoping to call their name to be called out. As students of the technology department of Centennial College, we would like to make proper implementation of technology to solve this problem.

## Proposed Solution

Scheduler Application tries to overcome the existing scenario of scheduling appointments, by allocating tokens to users via internet and estimating the time of appointment. Alert will be given to the users’ android device, which will save time in waiting in front of the office to see the official. Also the app will provide necessary information about other services provided by the institution, which may keep the users from scheduling appointments at all.

## High Level Deliverables

1. Web App for Administrators and officials (E.g.: Student counsellors).

2. Android mobile application to be used to book an appointment.

## Features

The application keeps tracks of various appointments scheduled and smartly informs the users about their upcoming appointments. It also calculates the average wait time and informs the user about their expected meeting time.

1. Makes appointment with counsellors in any department of Centennial College.
2. Analyses the average meeting time and calculates the expected meeting time and notifies user’s mobile device.
3. Admin can add departments, and officials/Counsellors.
4. Provides FAQs for general enquiries that avoid scheduling meetings with Counsellor.
5. Push announcements from the department to users’ android mobile.
6. Ask Questions to officials and get answers.

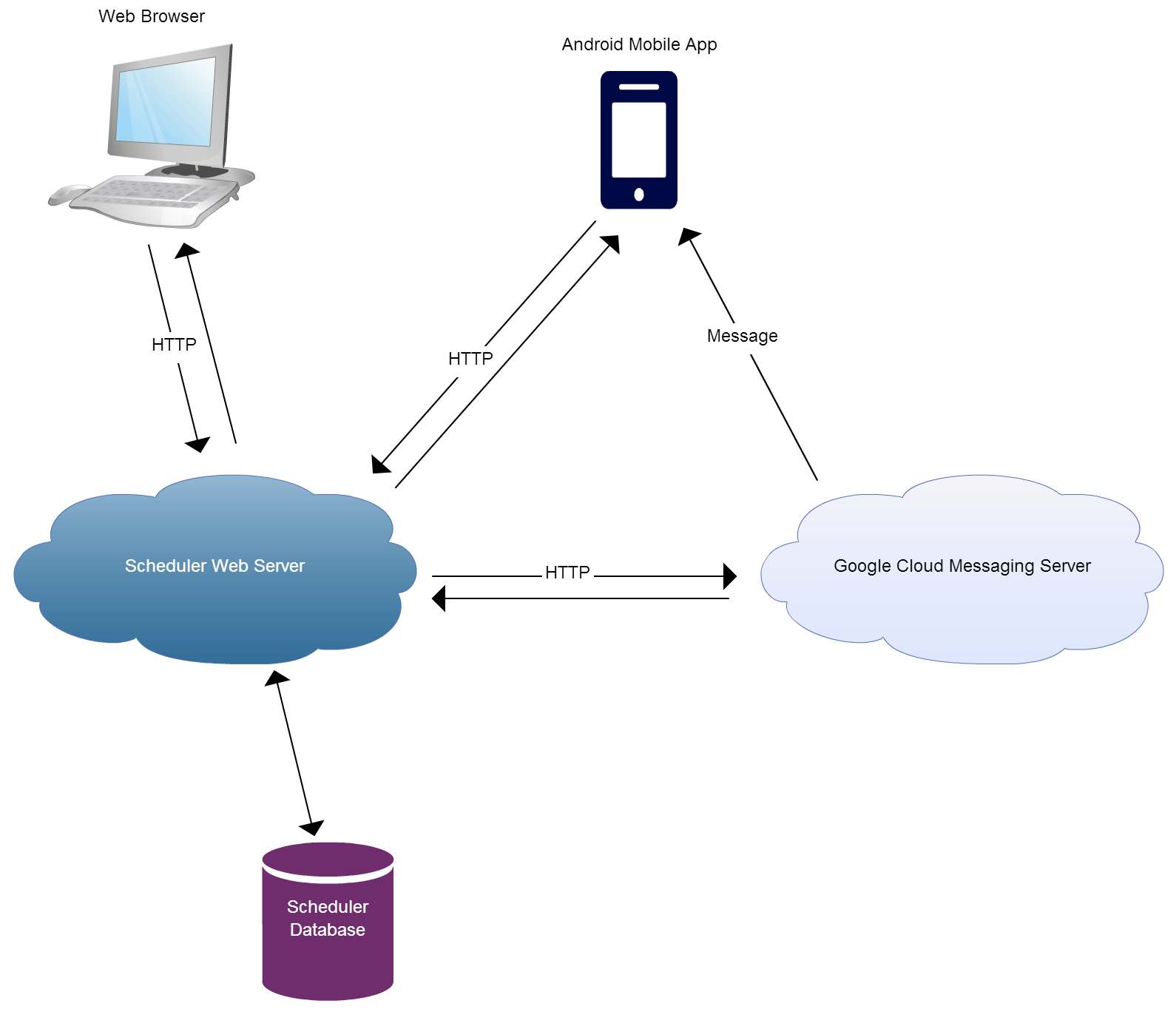
The products to be produced are:

1. Web App for Administrators and officials (E.g.: Student Counsellors).
   1. All business logic resides in Web app backend, and data will be exposed to mobile app through APIs.
   2. Mainly used by admin and student Counsellors to manage the system.
   3. Facility to see the dashboard, which will have the summary of users in queue, waiting times etc.
   4. Counsellors will start and end the appointment time, which will be used to calculate expected waiting and appointment time for students.
2. Android mobile application to be used to book an appointment.
3. Will have the ability to book appointment and receive the token.
4. Will provide a notification about the scheduled time and create a reminder 5 minutes before the appointment due time.
5. Able to view FAQs about services provided by college.

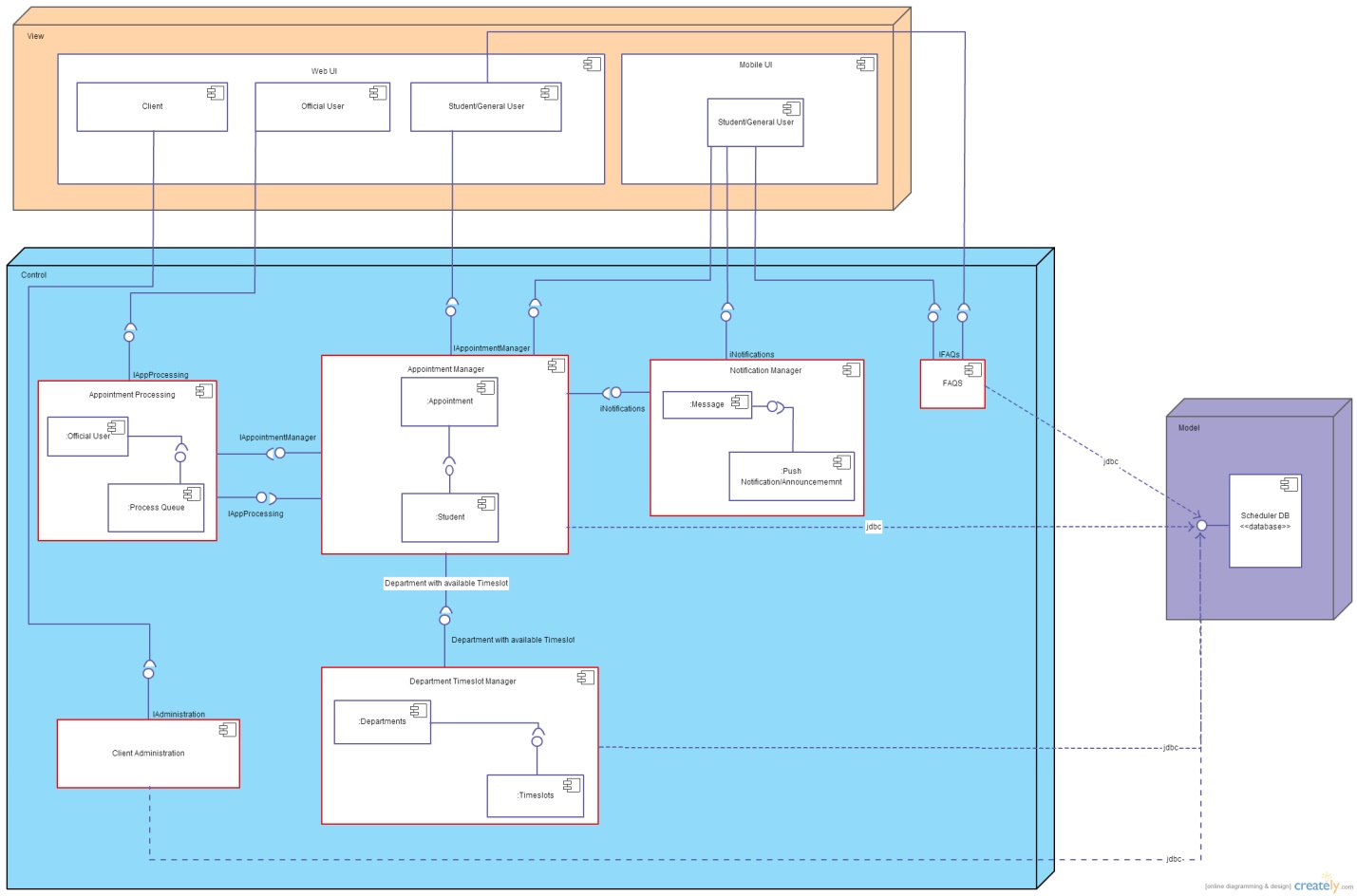
# Project Architecture

Git Hub Repository Link: https://github.com/dipen2512/Scheduler.git.

## High Level Architecture Diagram



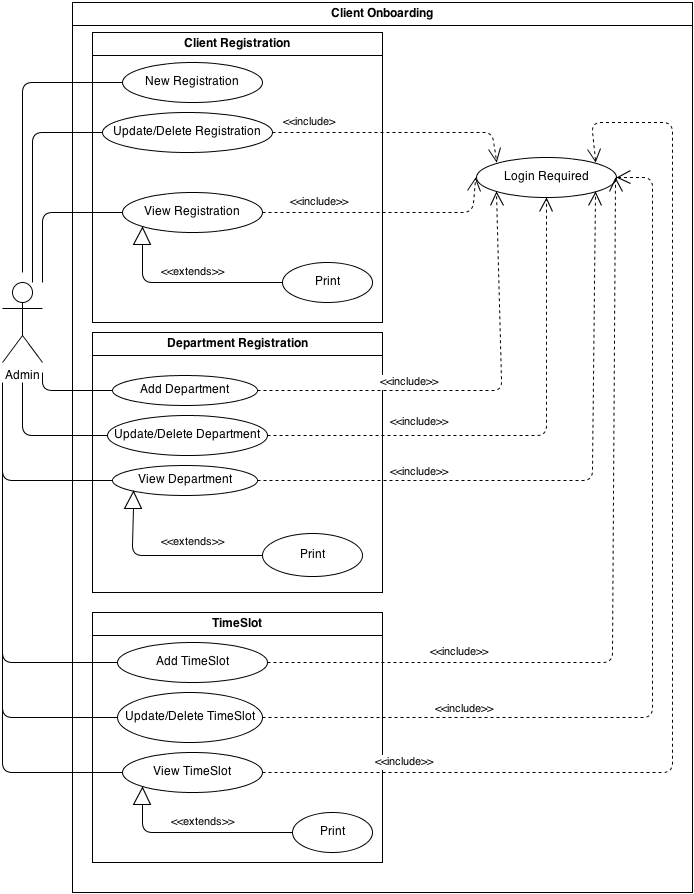
## Component Diagram



Description:- Component diagram consist of 3 layer, namely Model, View and Control layer. View Layer consist of users view. Controller layer consist of spring controllers which are mapped with Model layer through interfaces. Model layer consist of business logic classes and persistence classes and xml file.

# Use Cases

## Client Onboarding



1) Client Registration

a) New Registration: Client is able to register on the system.

b) Update/Delete Registration: Client is able to update or delete their registration details on the system.

c) View Registration: Client is able to view their registration details and able to print them if required.

2) Department Registration

a) Add Department: Client is able to add department to the system.

b) Update/Delete Department: Client is able to update or delete department from the system.

c) View Department: Client is able to view department information and print them if required.

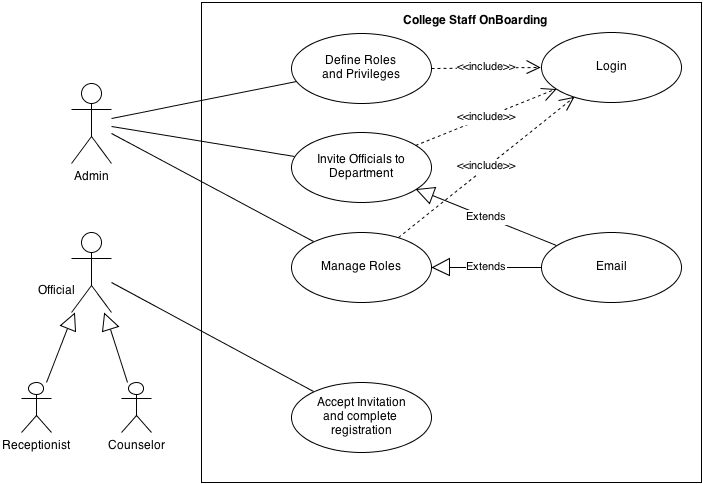
3) Timeslot

a) Add Timeslot: Client is able to add timeslot to the system.

b) Update/Delete Timeslot: Client is able to update or delete a timeslot from the system.

c) View Timeslot: Client is able to view timeslot information and print them if required.

## Client Onboarding



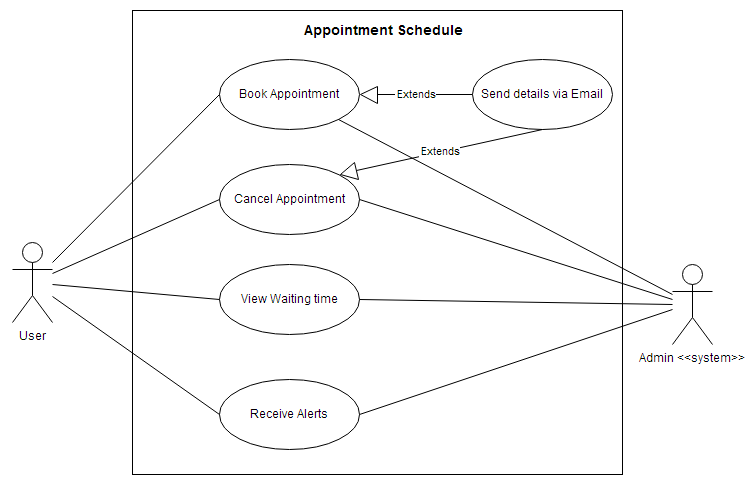
**Define Roles and Privileges:** Admin will define the roles and privileges that are given to each one (eg. Receptionist and Counselor are roles but Counselor can manage the queue whereas Receptionist cannot)

**Invite officials to Department:** Admin invites all officials to come and register in the system.

**Manage Roles:** Admin can manage (Add, Update, and Delete) the roles for staff and send them emails regarding updates.

**Accept invitation and complete registration:** Officials accept the invitation and completes the registration process and on boards to the system.

## Appointment Schedule



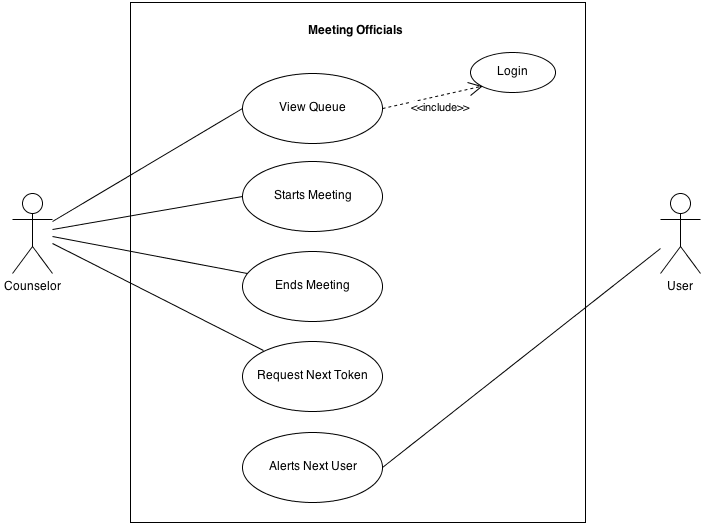
**UC1. Book Appointment:** A regular user book appointment on available days through the mobile or web app. When an appointment request is received, it is added to the backend system and confirmation is sent to user. User has the privilege to share this with any email address.

**UC2. Cancel Appointment:** If user finds out that he/she does not need to consult the counselor, they are eligible to cancel the appointment. Cancel confirmation will be sent to users mobile device or email address.

**UC3. View Waiting Time:** User can open up the app and see the expected visiting time based on the calculation from the historic visiting time of each user.

**UC4. Receive Alerts:** User will receive notifications regarding any updates of his/her appointment and also receive broadcasted message from the system administrator.

## Meeting Official



**UC1. View Queue:** - Counselor views the queue and gets the next available open token in order to start the meeting process. For that, he will login into the system first.

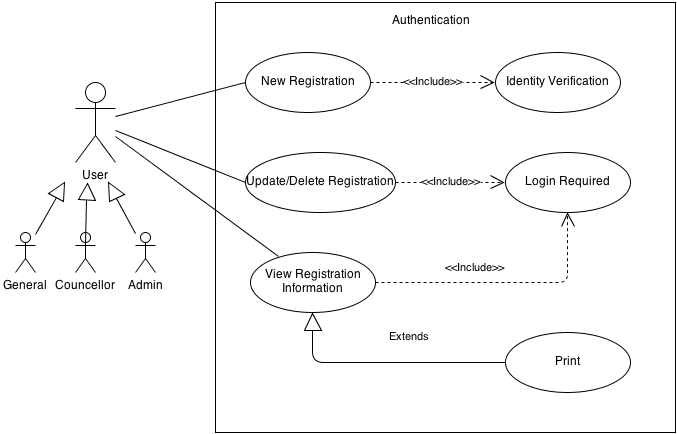
**UC2. Starts Meeting: -** After receiving the token from the queue Counselor starts the meeting and attends the student waiting for the counseling.

**UC3. Ends Meeting: -** After finishing the consultation with the current student the counselor will finalizes the documentation and end the meeting

**UC4. Requests Next Token: -** Counselor views the updated token queue and requests for the next token as soon as the current token is utilized.

**UC5. Alerts Next User: -** As soon as the counselor requests for the next token the system will automatically intimate the next user in the queue about the updates and alerts.

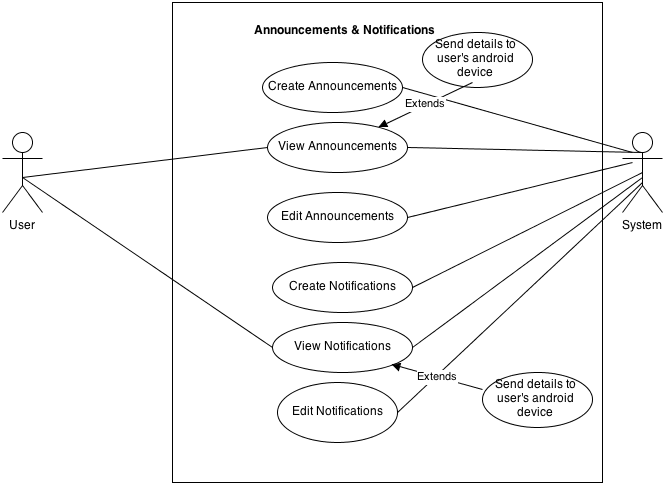
## Authentication



Authentication:

1. New Registration: User is able to register on the system.
2. Update/Delete Registration: User is able to update or delete their registration details on the system.
3. View Registration: User is able to view their registration details and able to print them if required.

## Announcement and Notification



**UC1. Create Announcements: -** This is created in case there are some changes in schedule of the department notifying all the users.

**UC2. View Announcements: -** This notifies the users via their Android device, in case there are some changes in schedule of the department thereby informing in advance.

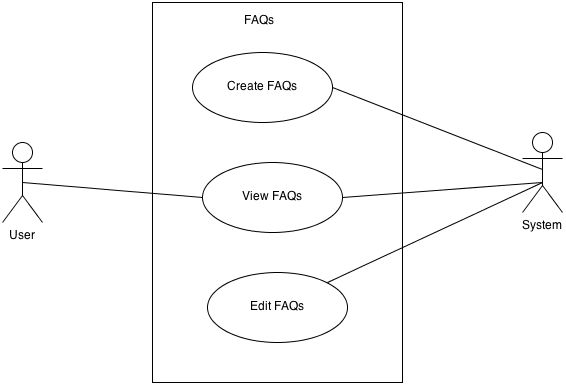
**UC3. Edit Announcements: -** This is done when there are further changes to schedule with regards to the department.

**UC4. Create Notifications: -** This is created in case there are some changes in schedule of the user's appointment time.

**UC5. View Notifications: -** This notifies the user via his/her Android device, in case there are some changes in schedule of the appointment time.

**UC6. Edit Notifications: -** This is done when there are further changes to the appointment time of the user.

## FAQs

******UC1. Create FAQs: -** The admin creates a list of frequently asked questions that addresses common queries of students (users). This reduces the number of appointments scheduled, thereby solving the problems or issues at one point of time.

**UC2. View FAQs: -** The FAQs created can be viewed by the user and system for quick references to some common queries.

**UC3. Edit FAQs: -** The admin at some point of time can edit the FAQs due to changes in system or new policy rules. The changes can then be viewed by both the user and system.

# Database

## SQL Script

CREATE DATABASE IF NOT EXISTS `scheduler` /\*!40100 DEFAULT CHARACTER SET latin1 \*/;

USE `scheduler`;

-- MySQL dump 10.13 Distrib 5.6.13, for Win32 (x86)

--

-- Host: localhost Database: scheduler

-- ------------------------------------------------------

-- Server version 5.6.14-log

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `announcement`

--

DROP TABLE IF EXISTS `announcement`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `announcement` (

`announcementId` int(50) NOT NULL AUTO\_INCREMENT,

`officialId` int(50) NOT NULL,

`announcementHeader` varchar(100) NOT NULL,

`announcementDescription` varchar(255) DEFAULT NULL,

`gcmMessageId` text,

`announcementDate` datetime NOT NULL,

PRIMARY KEY (`announcementId`)

) ENGINE=InnoDB AUTO\_INCREMENT=33 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `announcement`

--

LOCK TABLES `announcement` WRITE;

/\*!40000 ALTER TABLE `announcement` DISABLE KEYS \*/;

INSERT INTO `announcement` VALUES (32,1234,'Hello Students','Department will be closed by today afternoon',NULL,'2013-10-29 00:48:52');

/\*!40000 ALTER TABLE `announcement` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `appointment`

--

DROP TABLE IF EXISTS `appointment`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `appointment` (

`appointmentId` int(50) NOT NULL AUTO\_INCREMENT,

`departmentTimeId` int(50) NOT NULL,

`userId` int(50) NOT NULL,

`officialId` int(50) DEFAULT NULL,

`purposeOfVisit` varchar(255) DEFAULT NULL,

`startTime` time DEFAULT NULL,

`endTime` time DEFAULT NULL,

`meetingFinished` varchar(1) DEFAULT NULL,

`meetingNotes` text,

`dateCreated` datetime NOT NULL,

`appointmentDate` date NOT NULL,

PRIMARY KEY (`appointmentId`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `appointment`

--

LOCK TABLES `appointment` WRITE;

/\*!40000 ALTER TABLE `appointment` DISABLE KEYS \*/;

INSERT INTO `appointment` VALUES (1,1,1,3,'Course Enquiry','16:18:52','08:15:00','L','Counselled','2013-11-11 08:16:45','2013-10-28'),(2,1,2,1234,'Course Enquiry Details','14:09:16','08:30:00','N','Counselled','2013-11-12 08:30:45','2013-10-28'),(3,1,3,6,'Program Enquiry','00:00:00','08:45:00','N','Counselled','2013-11-13 08:45:45','2013-10-28'),(4,4,4,1,'Program Enquiry','10:29:13','10:29:34','Y','Hello','2013-11-13 09:45:45','2013-12-03'),(5,4,5,1,'Course Enquiry','10:53:49','10:12:00','N','Counselled','2013-11-13 10:13:00','2013-12-03'),(6,6,6,2,'Course Enquiry','10:30:00','10:48:15','N','Counselled','2013-11-13 10:50:00','2013-11-13');

/\*!40000 ALTER TABLE `appointment` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `campus`

--

DROP TABLE IF EXISTS `campus`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `campus` (

`campusId` int(50) NOT NULL AUTO\_INCREMENT,

`clientId` int(50) NOT NULL,

`campusName` varchar(50) NOT NULL,

`campusAddress` varchar(100) DEFAULT NULL,

`dateCreated` datetime NOT NULL,

PRIMARY KEY (`campusId`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `campus`

--

LOCK TABLES `campus` WRITE;

/\*!40000 ALTER TABLE `campus` DISABLE KEYS \*/;

INSERT INTO `campus` VALUES (1,1,'Progress Campus','940 Progress Campus','2013-10-18 14:30:22'),(2,1,'Morningside Campus','939 Morningside Campus','2013-10-19 14:45:22'),(3,1,'Ashtonbee Campus','938 Ashtonbee Campus','2013-10-20 14:05:22'),(4,1,'CCC Campus','937 CCC Campus','2013-10-21 13:05:22');

/\*!40000 ALTER TABLE `campus` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `category`

--

DROP TABLE IF EXISTS `category`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `category` (

`categoryId` int(50) NOT NULL AUTO\_INCREMENT,

`officialId` int(50) NOT NULL,

`categoryName` varchar(50) NOT NULL,

PRIMARY KEY (`categoryId`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `category`

--

LOCK TABLES `category` WRITE;

/\*!40000 ALTER TABLE `category` DISABLE KEYS \*/;

INSERT INTO `category` VALUES (1,1,'Course FAQs'),(2,2,'Admission FAQs'),(3,3,'Course FAQs'),(4,4,'Admission FAQs'),(5,5,'Course FAQs'),(6,6,'Admission FAQs');

/\*!40000 ALTER TABLE `category` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `client`

--

DROP TABLE IF EXISTS `client`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `client` (

`clientId` int(10) NOT NULL AUTO\_INCREMENT,

`clientName` varchar(50) NOT NULL,

`userName` varchar(50) NOT NULL,

`password` varchar(30) NOT NULL,

`email` varchar(30) NOT NULL,

`address` varchar(100) NOT NULL,

`memo` varchar(255) DEFAULT NULL,

`logo` varchar(100) DEFAULT NULL,

`phone1` varchar(15) NOT NULL,

`phone2` varchar(15) DEFAULT NULL,

`phone3` varchar(15) DEFAULT NULL,

`contactPerson` varchar(50) DEFAULT NULL,

`website` varchar(100) DEFAULT NULL,

`token` varchar(100) NOT NULL,

`emailVerified` tinyint(1) DEFAULT NULL,

`dateJoined` datetime NOT NULL,

PRIMARY KEY (`clientId`),

UNIQUE KEY `UNIQUE` (`userName`,`email`)

) ENGINE=InnoDB AUTO\_INCREMENT=10 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `client`

--

LOCK TABLES `client` WRITE;

/\*!40000 ALTER TABLE `client` DISABLE KEYS \*/;

INSERT INTO `client` VALUES (1,'Centennial College','root','admin','Centennial@mycentennial.ca','941 Progress Campus','College','centlogo-1.JPEG','416-723-5501','416-723-5523','416-456-2637','Ann Buller','www.centennialcollege.ca','55757377',1,'2013-10-16 14:45:16'),(2,'fd','dfa','fajsl','123@124.com','dfalk','asdf','','12132','13213','13213','31dfa','http://www.google.com','123',1,'2013-11-11 00:00:00'),(9,'Patel','dsfa','fsad','techrockers@yahoo.co.in','102, Padmavati Palace','dsf',NULL,'919909031761','919909031761','919909031761','fsdf','http://www.google.com','679910',NULL,'2013-11-11 00:00:00');

/\*!40000 ALTER TABLE `client` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `department`

--

DROP TABLE IF EXISTS `department`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `department` (

`departmentId` int(50) NOT NULL AUTO\_INCREMENT,

`campusId` int(50) NOT NULL,

`departmentName` varchar(50) NOT NULL,

`departmentHod` varchar(50) NOT NULL,

`contactInfo` int(15) NOT NULL,

`departmentDescription` varchar(255) DEFAULT NULL,

`dateCreated` datetime NOT NULL,

PRIMARY KEY (`departmentId`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `department`

--

LOCK TABLES `department` WRITE;

/\*!40000 ALTER TABLE `department` DISABLE KEYS \*/;

INSERT INTO `department` VALUES (1,1,'School of Engineering and Technology','Ashley Giles',416,'Enginnering Department','2013-10-20 14:04:22'),(2,1,'School of Business','Matt Giles',416,'Business Department','2013-10-20 14:00:22'),(3,1,'School of Continuing Education','Jeff Giles',416,'Continuing Education Department','2013-10-20 14:04:15'),(4,2,'School of Community and Health Studies','Michael Giles',416,'Community and Health Department','2013-10-20 13:04:22'),(5,2,'School of Advancement','Mitchel Slater',416,'Advancement Department','2013-10-20 12:08:22'),(6,3,'School of Hospitality','Clinton Slater',416,'Hospitality Department','2013-10-20 12:03:22'),(7,3,'School of ChildCare','Arjun Slater',416,'Childcare Department','2013-10-20 12:03:25'),(8,4,'School of Aviation','Randy Slater',416,'Aviation Department','2013-10-22 12:03:25'),(9,4,'Applied Research and Innovation Center','Amy Slater',416,'Research Department','2013-10-22 12:03:45'),(10,4,'jasdlfjsadfk','Sanket Patel',123136516,'Final Testing','2013-11-27 17:31:15');

/\*!40000 ALTER TABLE `department` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `departmenttimeslot`

--

DROP TABLE IF EXISTS `departmenttimeslot`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `departmenttimeslot` (

`departmentTimeId` int(50) NOT NULL AUTO\_INCREMENT,

`departmentId` int(50) NOT NULL,

`timeslotId` int(50) NOT NULL,

`weekdays` varchar(7) NOT NULL,

`capacity` int(5) NOT NULL,

PRIMARY KEY (`departmentTimeId`)

) ENGINE=InnoDB AUTO\_INCREMENT=18 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `departmenttimeslot`

--

LOCK TABLES `departmenttimeslot` WRITE;

/\*!40000 ALTER TABLE `departmenttimeslot` DISABLE KEYS \*/;

INSERT INTO `departmenttimeslot` VALUES (1,1,1,'1111000',30),(2,2,1,'1111100',35),(3,4,2,'1110000',25),(4,7,2,'1111100',28),(5,1,3,'0000100',10),(6,1,4,'0000010',10),(7,10,1,'1000000',1),(8,10,1,'0010000',1),(9,10,1,'0000100',1),(10,10,2,'1000000',1),(11,10,2,'0010000',1),(12,10,2,'0000100',1),(13,10,7,'1000000',1),(14,10,7,'0100000',1),(15,10,7,'0001000',1),(16,10,9,'1000000',1),(17,10,9,'0010000',1);

/\*!40000 ALTER TABLE `departmenttimeslot` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `faq`

--

DROP TABLE IF EXISTS `faq`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `faq` (

`faqId` int(50) NOT NULL AUTO\_INCREMENT,

`categoryId` int(50) NOT NULL,

`officialId` int(50) NOT NULL,

`faqQuestion` varchar(255) NOT NULL,

`faqAnswer` varchar(255) NOT NULL,

`dateCreated` datetime NOT NULL,

PRIMARY KEY (`faqId`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `faq`

--

LOCK TABLES `faq` WRITE;

/\*!40000 ALTER TABLE `faq` DISABLE KEYS \*/;

INSERT INTO `faq` VALUES (1,1,1,'How to change course','Navigate to Home => Change Course.','2013-11-11 12:34:23'),(2,2,2,'How to register for a program?','Navigate to Home => New Program','2013-11-11 12:23:45');

/\*!40000 ALTER TABLE `faq` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `generaluser`

--

DROP TABLE IF EXISTS `generaluser`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `generaluser` (

`userId` int(50) NOT NULL AUTO\_INCREMENT,

`clientId` int(50) NOT NULL,

`username` varchar(50) NOT NULL,

`password` varchar(50) NOT NULL,

`firstName` varchar(50) NOT NULL,

`lastName` varchar(50) NOT NULL,

`email` varchar(100) NOT NULL,

`dob` date DEFAULT NULL,

`address` varchar(100) DEFAULT NULL,

`gender` varchar(10) DEFAULT NULL,

`token` varchar(100) NOT NULL,

`emailVerified` tinyint(1) DEFAULT NULL,

`gcmRegId` text,

PRIMARY KEY (`userId`),

KEY `UNIQUE` (`username`,`email`)

) ENGINE=InnoDB AUTO\_INCREMENT=20 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `generaluser`

--

LOCK TABLES `generaluser` WRITE;

/\*!40000 ALTER TABLE `generaluser` DISABLE KEYS \*/;

INSERT INTO `generaluser` VALUES (1,1,'dkaith','cc1','Kaith','Devon','dkaith@ccollege.com','1985-12-14','22 Progress Ave','Female','ab12',1,'demoId'),(2,1,'dkemar','cc2','Kemar','Devon','dkemar1@ccollege.com','1985-12-12','46 rudington dr','Male','ac12',1,'APA91bFNiV2wA0nrzr9AmFnmULzBXpg0vuOzEMV36ih9bsWOXiFj8VFFs80cBck55DtVeKo3DFsVgRK8eIu8\_PB\_in0PNnaRW\_vLKiS9lKuMtc6a8m3f5a2tdnHpPmrWRZYCDOIWTr79xWph19Q-PzOfTLOki-oPbg'),(3,1,'pvishal','cc3','Vishal','Punjabi','pvishal@ccollege.com','1986-12-12','45 rudington dr','Male','ad12',1,'APA91bEV6sWyrFiPsF5a4Cb8HVKx7BuE3jgEznRGG5SRODgpTimEI5XWjRUgrARq1t-ZFDGiodfTuwSQYTSGnpfrNXjw17welyABfrHzPbCtaQyU9rYc9X4VCx0UzPn6IGiB43I2Q7QmTk\_94fxEesvWF4Rwdiumug'),(4,1,'vagin','cc4','Vishal','Agin','vagin@cccollege.com','1984-12-12','47 rudington dr','Male','ae12',1,'abcd'),(5,1,'dvalecha','cc5','Devraj','Valecha','dvalecha@ccollege.com','1986-10-16','48 rudington dr','Male','af12',1,'abcd'),(6,1,'sbanjara','cc6','Shalin','Banjara','sbanjara@ccollege.com','1984-08-08','49 rudington dr','Male','ag12',1,'abcd'),(7,1,'ksara','cc7','Kim','Sara','ksara@ccollege.com','1984-08-08','50 rudington dr','Female','ah12',1,'abcd'),(17,1,'techrockers','fasd','Sanket','Patel','sanket.scorp@gmail.com','1986-12-12','fajsdl','Male','347117',1,NULL),(19,1,'fasd','fasd','dfasd','fasd','techrockers@yahoo.co.in','1986-12-12','102, Padmavati Palace','Male','796556',1,NULL);

/\*!40000 ALTER TABLE `generaluser` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `notification`

--

DROP TABLE IF EXISTS `notification`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `notification` (

`notificationId` int(50) NOT NULL AUTO\_INCREMENT,

`officialId` int(50) NOT NULL,

`userId` int(50) NOT NULL,

`gcmMessageId` varchar(100) DEFAULT NULL,

`notificationHeader` varchar(100) DEFAULT NULL,

`notificationDescription` varchar(255) NOT NULL,

`readByUser` tinyint(1) NOT NULL,

`notificationDate` datetime NOT NULL,

PRIMARY KEY (`notificationId`),

UNIQUE KEY `UNIQUE` (`gcmMessageId`)

) ENGINE=InnoDB AUTO\_INCREMENT=39 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `notification`

--

LOCK TABLES `notification` WRITE;

/\*!40000 ALTER TABLE `notification` DISABLE KEYS \*/;

INSERT INTO `notification` VALUES (19,1234,2,'0:1382983756742590%3d747f6b38eb0007','Meeting ready to be started','Please come in to the department',0,'2013-10-28 14:09:16'),(20,1234,3,'0:1382983756848381%3d747f6b38eb0007','Meeting starting soon!','You are the next person in queue',0,'2013-10-28 14:09:17'),(21,3,1,'0:1385587101798893%3d747f6b38eb0007','Meeting ready to be started','Please come in to the department',0,'2013-11-27 16:18:21'),(22,3,3,'0:1385587101977230%3d747f6b38eb0007','Meeting starting soon!','You are the next person in queue',0,'2013-11-27 16:18:21'),(23,3,1,'0:1385587133396606%3d747f6b38eb0007','Meeting ready to be started','Please come in to the department',0,'2013-11-27 16:18:53'),(24,3,3,'0:1385587133554719%3d747f6b38eb0007','Meeting starting soon!','You are the next person in queue',0,'2013-11-27 16:18:53'),(25,1,4,NULL,'Meeting ready to be started','Please come in to the department',0,'2013-12-03 10:12:27'),(26,1,4,NULL,'Meeting ready to be started','Please come in to the department',0,'2013-12-03 10:14:32'),(27,1,4,NULL,'Meeting ready to be started','Please come in to the department',0,'2013-12-03 10:15:49'),(28,1,4,NULL,'Meeting ready to be started','Please come in to the department',0,'2013-12-03 10:19:37'),(29,1,4,NULL,'Meeting ready to be started','Please come in to the department',0,'2013-12-03 10:21:02'),(30,1,5,NULL,'Meeting starting soon!','You are the next person in queue',0,'2013-12-03 10:21:33'),(31,1,4,NULL,'Meeting ready to be started','Please come in to the department',0,'2013-12-03 10:25:55'),(32,1,5,NULL,'Meeting starting soon!','You are the next person in queue',0,'2013-12-03 10:25:59'),(33,1,4,NULL,'Meeting ready to be started','Please come in to the department',0,'2013-12-03 10:29:13'),(34,1,5,NULL,'Meeting starting soon!','You are the next person in queue',0,'2013-12-03 10:29:13'),(35,1,5,NULL,'Meeting ready to be started','Please come in to the department',0,'2013-12-03 10:30:03'),(36,1,5,NULL,'Meeting ready to be started','Please come in to the department',0,'2013-12-03 10:41:37'),(37,1,5,NULL,'Meeting ready to be started','Please come in to the department',0,'2013-12-03 10:43:05'),(38,1,5,NULL,'Meeting ready to be started','Please come in to the department',0,'2013-12-03 10:53:53');

/\*!40000 ALTER TABLE `notification` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `officialuser`

--

DROP TABLE IF EXISTS `officialuser`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `officialuser` (

`officialId` int(50) NOT NULL AUTO\_INCREMENT,

`departmentId` int(50) NOT NULL,

`roleId` int(50) NOT NULL,

`officialName` varchar(50) NOT NULL,

`email` varchar(100) NOT NULL,

`password` varchar(50) NOT NULL,

`firstName` varchar(50) NOT NULL,

`lastName` varchar(50) NOT NULL,

`dateJoined` date NOT NULL,

`lastLogin` datetime NOT NULL,

PRIMARY KEY (`officialId`),

KEY `UNIQUE` (`officialName`,`email`)

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `officialuser`

--

LOCK TABLES `officialuser` WRITE;

/\*!40000 ALTER TABLE `officialuser` DISABLE KEYS \*/;

INSERT INTO `officialuser` VALUES (1,7,2,'Sanket','techrockers@yahoo.co','123','Ashley','Giles','2013-05-22','2013-05-22 12:23:34'),(2,1,3,'Mark Waugh','m2@ccollge.com','admin2','Mark','Waugh','2013-05-22','2013-05-23 12:25:34'),(3,2,2,'Matt Giles','m3@ccollge.com','admin3','Matt','Giles','2013-05-22','2013-05-23 13:25:34'),(4,2,3,'Alex Tudor','m4@ccollge.com','admin4','Alex','Tudor','2013-05-23','2013-05-23 13:25:30'),(5,4,2,'Michael Giles','m5@ccollge.com','admin5','Michael','Giles','2013-05-23','2013-05-24 13:25:30'),(6,4,3,'Tony Greg','m6@ccollge.com','admin6','Tony','Greg','2013-05-24','2013-05-25 13:25:30'),(7,7,2,'Arjun Slater','m7@ccollge.com','admin7','Arjun','Slater','2013-05-25','2013-05-25 12:25:30'),(8,7,3,'Karan Slater','m8@ccollge.com','admin8','Karan','Slater','2013-05-26','2013-05-27 12:25:30');

/\*!40000 ALTER TABLE `officialuser` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `roles`

--

DROP TABLE IF EXISTS `roles`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `roles` (

`roleId` int(50) NOT NULL AUTO\_INCREMENT,

`clientId` int(50) NOT NULL,

`roleName` varchar(50) NOT NULL,

`privilege` varchar(255) NOT NULL,

`description` varchar(255) DEFAULT NULL,

PRIMARY KEY (`roleId`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `roles`

--

LOCK TABLES `roles` WRITE;

/\*!40000 ALTER TABLE `roles` DISABLE KEYS \*/;

INSERT INTO `roles` VALUES (1,1,'President','Universal Access','College Affairs.'),(2,1,'HOD','Department Access','Department Affairs.'),(3,1,'Counsellor','Department Access','Department and Student Affairs');

/\*!40000 ALTER TABLE `roles` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `test\_user`

--

DROP TABLE IF EXISTS `test\_user`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `test\_user` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`firstname` varchar(20) NOT NULL,

`lastname` varchar(20) NOT NULL,

`email` varchar(20) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `test\_user`

--

LOCK TABLES `test\_user` WRITE;

/\*!40000 ALTER TABLE `test\_user` DISABLE KEYS \*/;

INSERT INTO `test\_user` VALUES (1,'Sonny','rr','skr@gmail.com'),(2,'shal','ss','ss@gmail.com');

/\*!40000 ALTER TABLE `test\_user` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `timeslot`

--

DROP TABLE IF EXISTS `timeslot`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `timeslot` (

`timeslotId` int(50) NOT NULL AUTO\_INCREMENT,

`clientId` int(11) NOT NULL,

`startTime` time NOT NULL,

`stopTime` time NOT NULL,

`description` varchar(255) DEFAULT NULL,

PRIMARY KEY (`timeslotId`)

) ENGINE=InnoDB AUTO\_INCREMENT=10 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `timeslot`

--

LOCK TABLES `timeslot` WRITE;

/\*!40000 ALTER TABLE `timeslot` DISABLE KEYS \*/;

INSERT INTO `timeslot` VALUES (1,1,'07:00:00','11:00:00','Slot timings'),(2,1,'08:00:00','12:00:00','Slot timings'),(6,1,'10:00:00','15:00:00','Normal working time'),(7,1,'13:59:00','12:59:00','fkasdjflkajsdflk'),(8,1,'00:12:00','12:12:00','12 Hours'),(9,1,'00:00:00','12:00:00','12 Hours');

/\*!40000 ALTER TABLE `timeslot` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `userannouncements`

--

DROP TABLE IF EXISTS `userannouncements`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `userannouncements` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`announcementId` int(11) NOT NULL,

`userId` int(11) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=105 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `userannouncements`

--

LOCK TABLES `userannouncements` WRITE;

/\*!40000 ALTER TABLE `userannouncements` DISABLE KEYS \*/;

INSERT INTO `userannouncements` VALUES (99,32,1),(100,32,2),(101,32,3),(102,32,4),(103,32,5),(104,32,6);

/\*!40000 ALTER TABLE `userannouncements` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

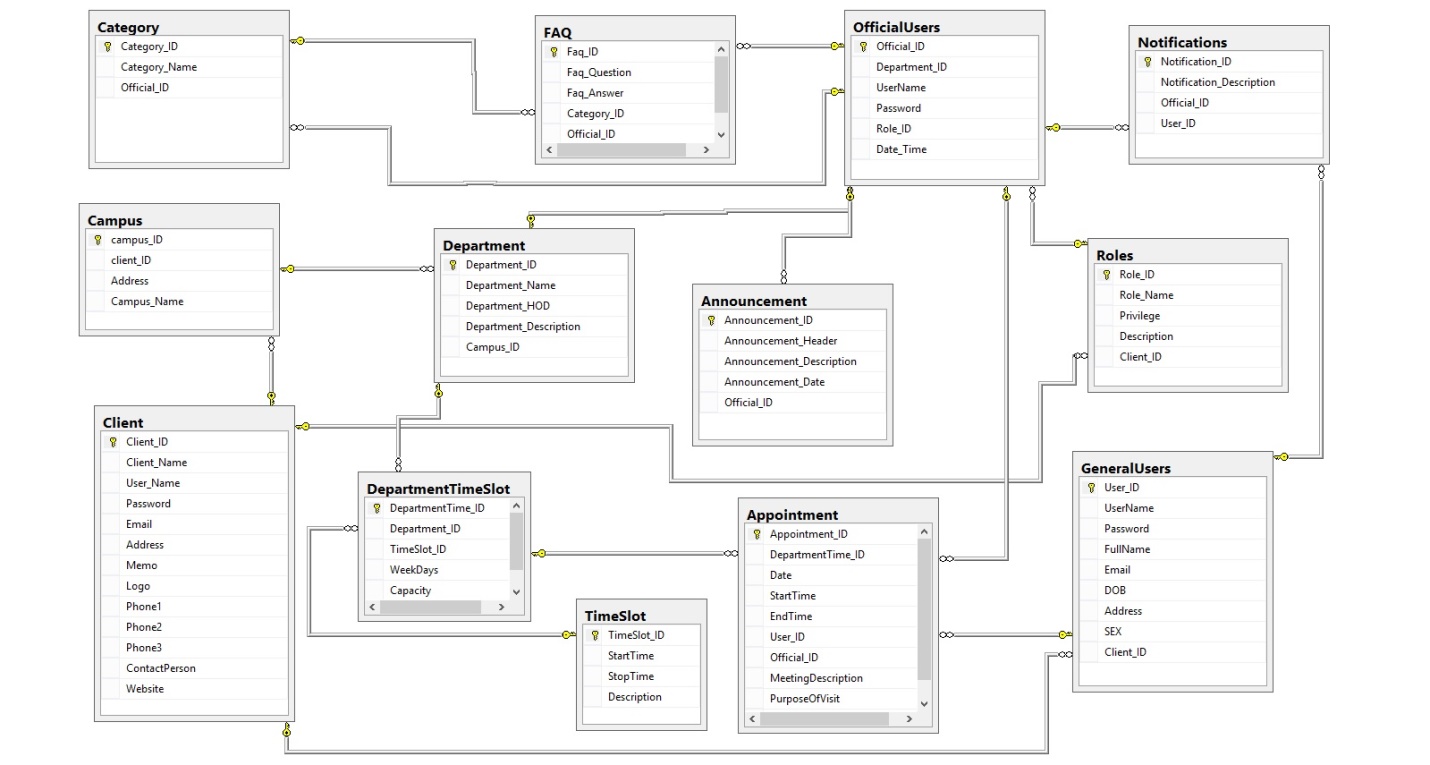
/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2013-12-06 20:47:16

## Database Diagram

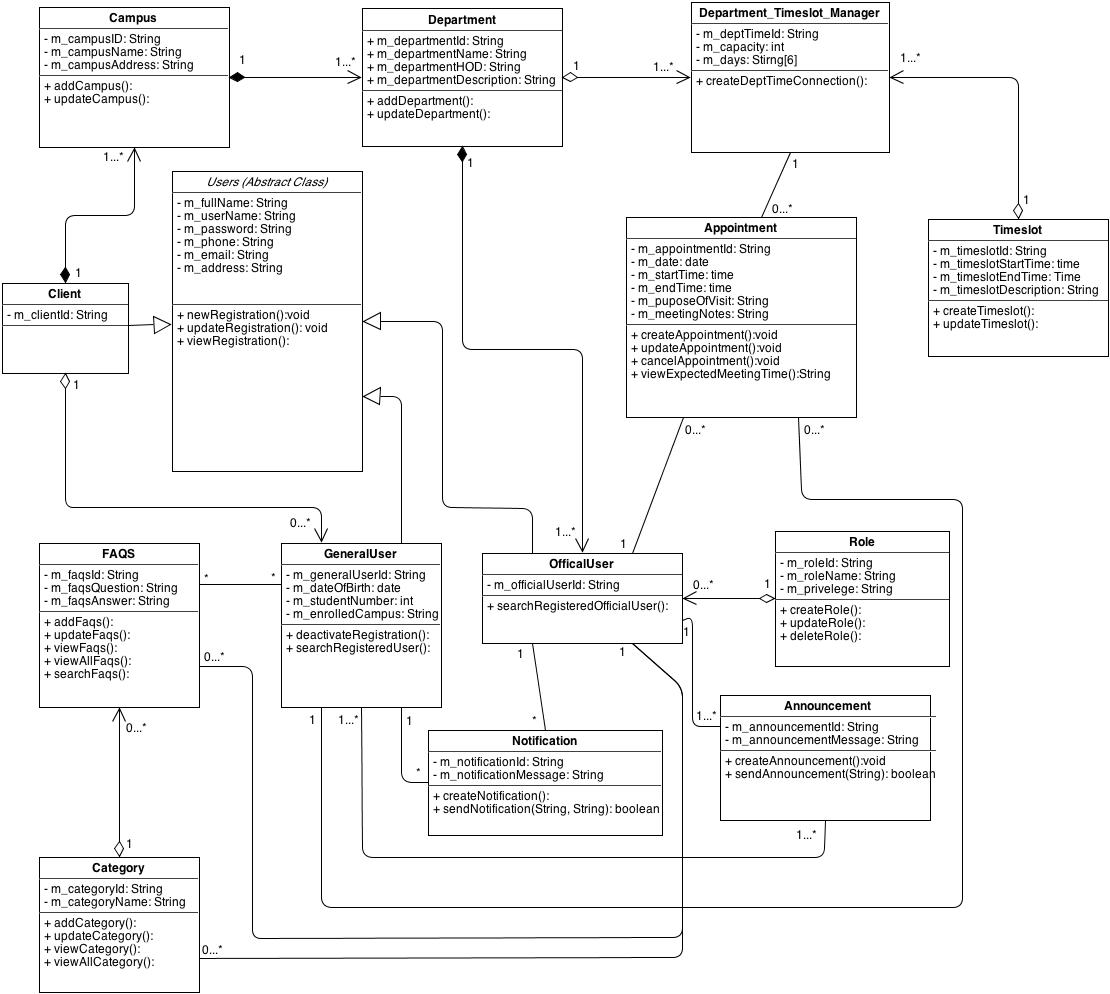


Description: - Database has been designed with a perspective of avoiding data redundancy. So to avoid many to many relationship, junction tables have been created. Also tables are designed in such a manner that no duplicated field exist.

# Classes

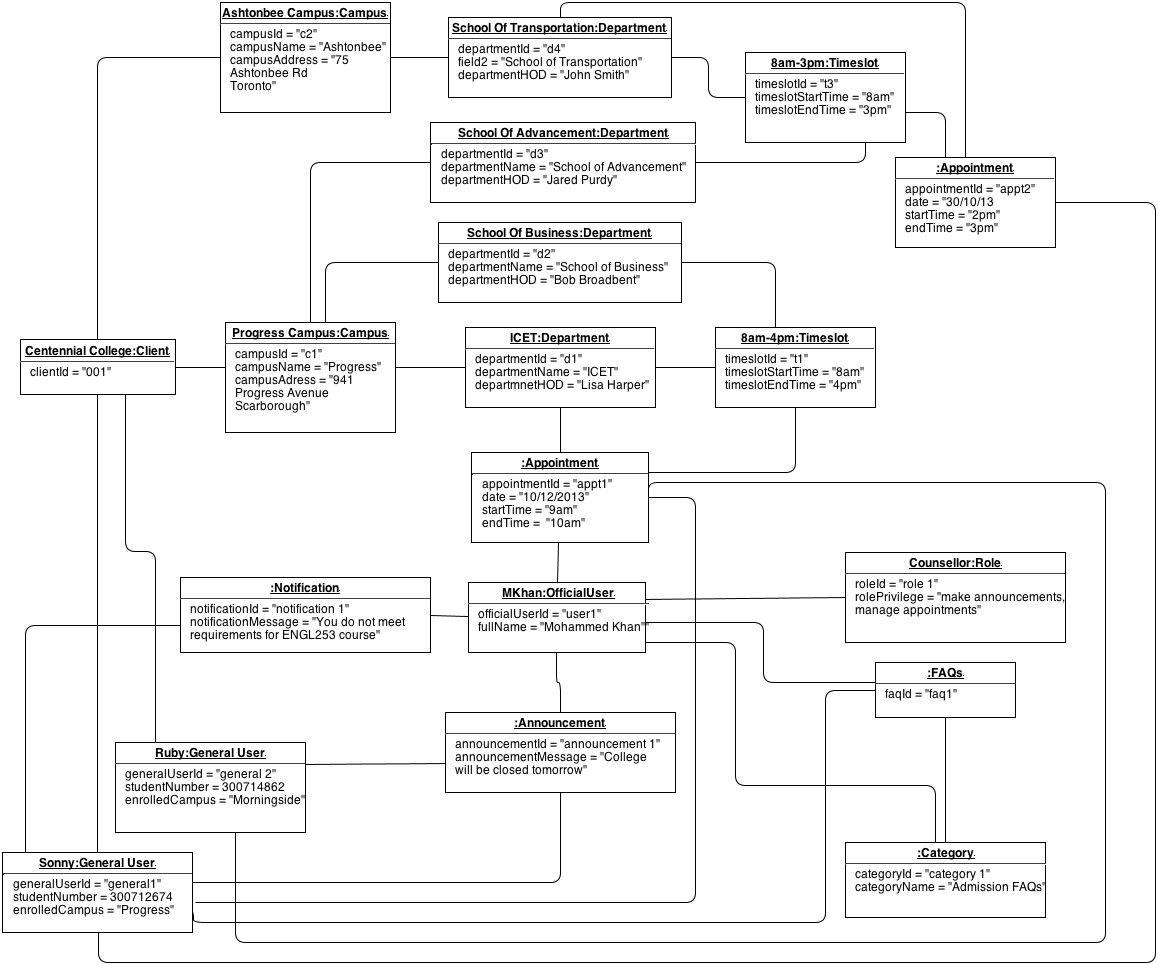
## Web Application

### Class Diagrams



Description:- Class diagram describe the Data Transfer Objects and their mapping based on database diagram. Department\_Timeslot\_Mager class is a junction class for DepartmentTimeslot table. Class diagram describes the One to Many and many to one relationship with the respective classes.

### Object Diagram



The following diagram is an object diagram for the project. It represents the Scheduler - schedule management system the application we developed for our project. The following diagram is an instance of the system at a particular instance of time. It mocks the below mentioned objects stored in the database at that moment. Objects mocked in the diagram are following:-

* Client
* Campus
* Department
* Notification
* General User
* Official User
* Appointment
* Timeslot
* Role
* Faq's'
* Category

Now the Client object (Centennial College) is our main object which is the root of our application and all other objects are associated with it and without the existence of the client none of the objects can exists. After the there are number of objects at the first level after the client is created to make the system working we will go through all the objects one by one. As our target customer's' are the college's' the first thing they need is the campuses' so we have the campus object directly attached to the Campus object diagram takes example of the two campus Ashtonbee and Progress campus. Next is we need user's to manage the application then we have our system users. There are three types of the users in the system General User object which has two user's Ruby and Sonny. General users represents the user objects which are the front users of the system and use the application to schedule the appointments. After the general users we have the Official users objects represented by M. Khan. Official users are the users appointed by the client to handle the admin activities on behalf of him. They are directly associated with the Role object and Official users are assigned the different roles on which basis system runs and applies the system permissions and access roles.

After we have all the first level objects associated to the client we move ahead. Now campus are not useful until they have the working department in them so we have Department object. In the diagram we show four departments attached to the two campuses (School of Transportation, School of Advancement, School of Business and ICET department). Each of them are directly linked to the campus which is linked to our main client.

Each department defines the time slots in which meetings can be booked. Diagram represents two different time slots for four different departments. Timeslot objects are 8am - 3pm and 8am - 4pm which are connected to the Departments School of Transportation and Advancement and the later one with the School of Business and ICET department respectively.

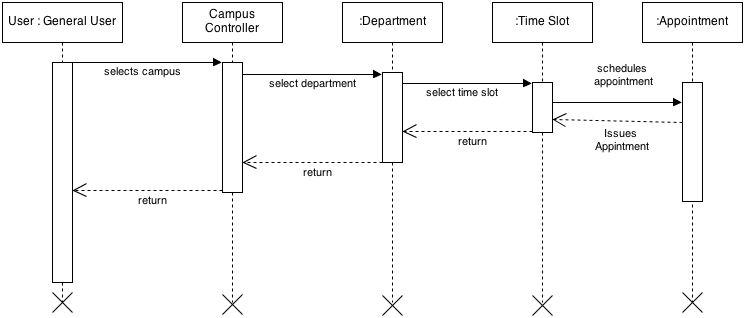
Once we have all the main content set up General users can go and book the appointments and here the Appointment object comes into picture. We have only one Appointment described in the diagram which is assigned to the Official user MKhan and is booked by General user Sonny.

There are some general objects also described in the object diagram. We have the Notification object which is used to give notification to the users about the appointment activity and updates on the appointment. Other two are Faq object and Category object. We have one instance of each of them faq1 and the category1.

Diagram showed the object mappings by mocking the objects and relating them to each other. Diagram represents the actual structure of the application data model at a point of time and shows how they are connected to each other maintaining the data integrity of the application.

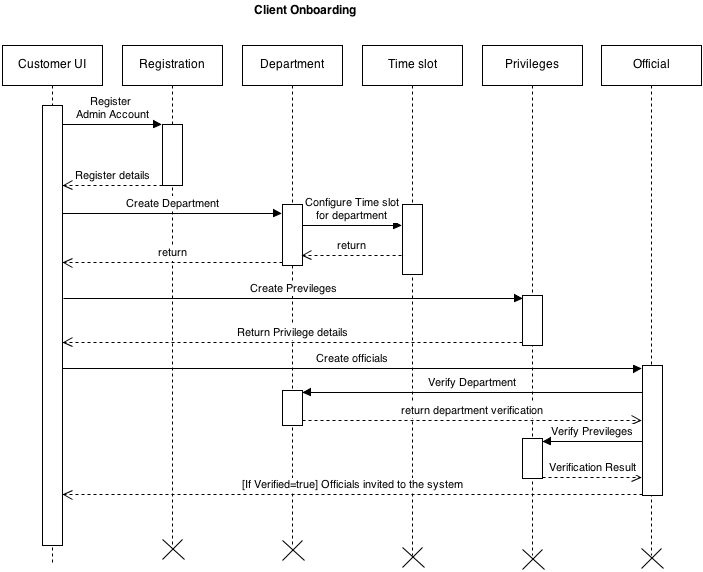
### Sequence Diagrams

#### Appointment Schedule



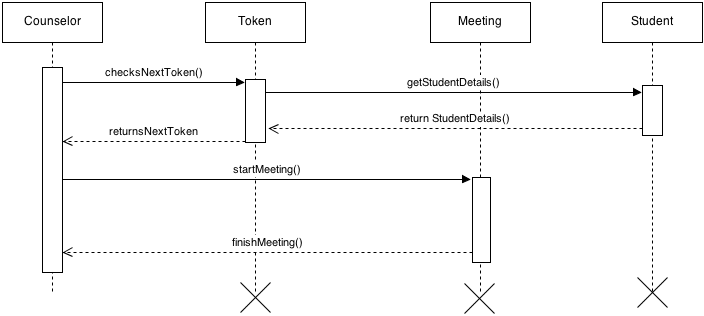
Description:- To create a new appointment, student has to select the campus and department. Once the department is selected, the list of available timeslots is fetched and returned. Once they are selected, an object of appointment class is created, and appointment is saved, and success message is returned back to the student

#### Client Onboarding



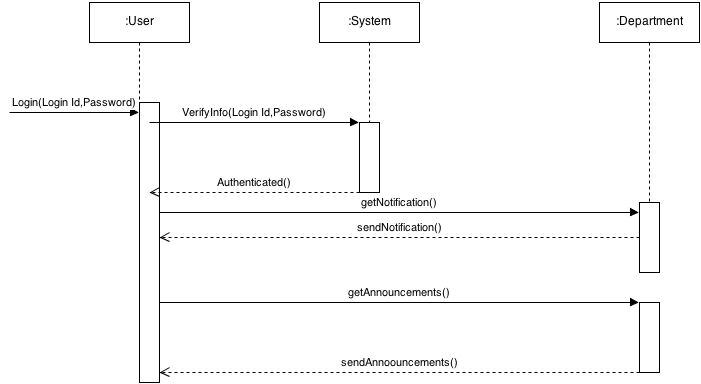
Description:- Once the roles and departments are defined, the admin can go ahead and create new officials and each official has to be attached to a department, and a particular set of privileges. Once the creation of Official is finished and verified, the official themselves can login to the system based on the initial credential provided by the Admin.

#### Meeting Official



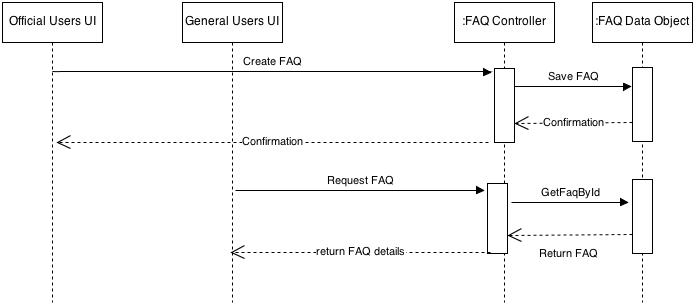
Description: - When the meeting is started, the next token is fetched from database, and corresponding student details is retrieved. Once these details are received, the meeting is started and finished.

#### Announcements Notifications



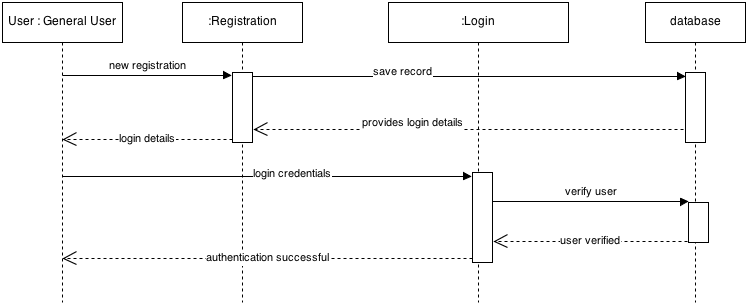
Description:- The student needs to login to the system in order to see the announcement and notification. User access announcement/notification class to send or receive announcement/notification.

#### FAQs



Description:- To create a new FAQ, the official user access the FAQ class and the confirmation is sent back to the official. To retrieve the FAQ, the corresponding faq id is sent and question answer is returned back to the general user

#### Authentication



Description:- Once a student (General user) is registered, the record is saved to the database. Once the student logs in, the credentials are verified against the database using the login data object.

### State Diagram

#### Appointment

#### https://github.com/dipen2512/Scheduler/blob/master/Diagrams/State%20Diagrams/Appointment.png?raw=true

Description: - The student visits the appointment page to set up an appointment at the desired time. Once the appointment is scheduled, the student waits for the appointment which can vary slightly depending upon the average meeting time of the previous student.

#### Meeting Official

#### https://github.com/dipen2512/Scheduler/blob/master/Diagrams/State%20Diagrams/Meeting%20Official.png?raw=true

Description: - An official user can start the first available appointment in the queue. Once appointment started it is in wait mode. If users arrive in time then is goes to in meeting mode and then finished state. If while in wait mode, if marked as late appointment state again rolls back to ready fetching the new user in appointment queue.

#### Client Onboarding:-

#### https://github.com/dipen2512/Scheduler/blob/master/Diagrams/State%20Diagrams/ClientOnBoarding.png?raw=true

Description: - The client is first created when it is registered. Post this, it can either be updated or deleted.

After the client is created, various departments are created post which it can be either updated or deleted.

When all the respective departments are created, time slots are created. This can be either deleted or updated.

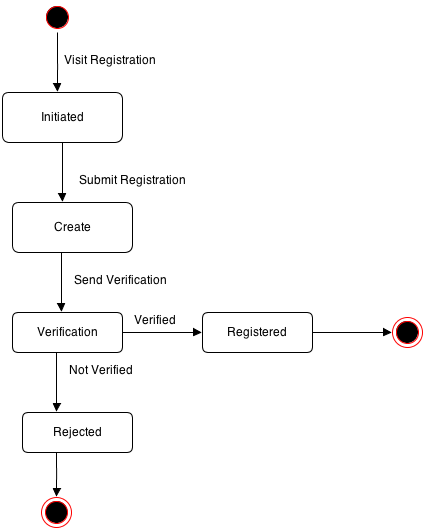
At the end, official users or staff members are created. This too can either be created or deleted.

#### Announcement

#### https://github.com/dipen2512/Scheduler/blob/master/Diagrams/State%20Diagrams/Announcement.png?raw=true

Description: - Official user visits the send announcements page to create an announcement. Once the announcement is created, it can either be submitted to broadcast or cancel.

#### Registration



Description: - The client/official user/ general user visits the registration page. Once the fields are inputted in the form, it is submitted for registration. Post registration, a verification is sent to the email address to confirm the genuinely of the person who registered. Once it is verified, the client/official user/ general user is successfully registered. If it is not verified, registration fails.