**INTERVIEW CALENDER**

1. **FRAMEWORK**

The application is implemented as .Net console application in Visual Studio 2017 by using .Net framework 4.6.1. SQLite is used as database engine.

1. **DATA MODEL**

**Tables**

**USERS**: Table used for storing interviewer and candidates.

ID: Primary key, stores unique user id values

USER\_NAME: Name of the user

ROLE: Role of the user. Value definitions are stored as enum (1 for Interviewer, 2 for Candidate)

**AVAILABLE\_TIME\_SLOTS**: Table used for storing the available time slot values for interviewers.

ID: Primary key

USER\_ID: User id of the interviewer

START\_TIME: Available slot time start value (as datetime) for interviewer

END\_TIME: Available slot time end value (as datetime) for interviewer

Start – end time pairs are kept as slot by slot. For example if the interviewer is available on Monday between 10 and 12 am, there are two rows in the table one for slot 10-11 and other for slot 11-12. Consecutive slots are inserted in different rows; because when one slot is reserved for a candidate that slot will be deleted from the database.

**REQUESTED\_TIME\_SLOTS**: Table used for storing the requested time slot values for candidates.

ID: Primary key

USER\_ID: User id of the candidate

START\_TIME: Available slot time start value (as datetime) for candidate

END\_TIME: Available slot time end value (as datetime) for candidate

1. **BUSINESS MODEL**

**InterviewCalender.Business layer:**

Interviewer and Candidate classes are inherited from a base User class. Inherited objects are constructed with factory pattern using EnumRole value (1 for interviewer, 2 for candidate).

**InterviewCalender.Common layer:**

Enum values and utility classes are stored in this layer.

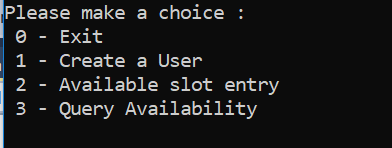
**InterviewCalender.Data layer:**

There is a data repository class for each table in the database. Database operations like insert, select etc are done on these classes by using SQLite functions.

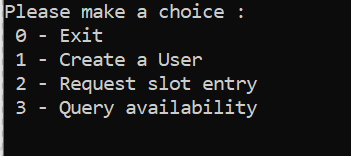
1. **APPLICATION**

Application is implemented as a console application. Both interviewers and candidates can use the application by using command line. Application is started by selecting a role (interviewer/candidate)

Start menu for interviewer:



Start menu for candidate:



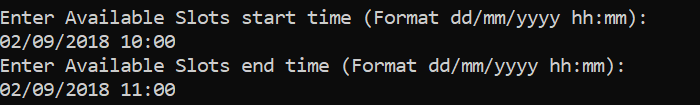
**Create User:**

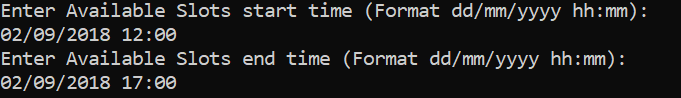
Depending on the role candidate/interviewer user is created. A user can be created by simply entering a non-empty user name. Unique user id of the newly created user is informed to the console.

**Available slot entry:**

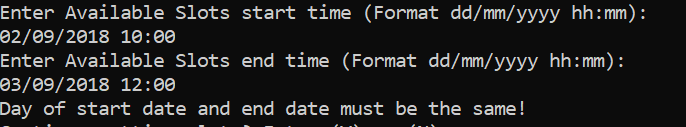
This menu option can be used by the interviewer role. User id, slot start and end time values are taken from the user. The user must specify the start and end time for each day separately. Consecutive slot times for the same day can be entered together.

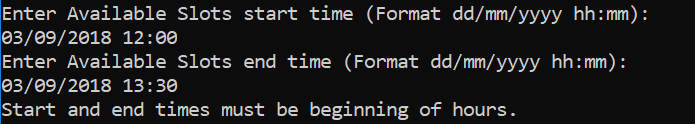
Valid slot entries:

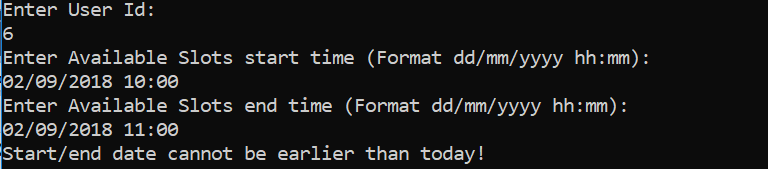




Invalid slot entries:







**Request slot entry:**

This menu option can be used by candidate role. User id, slot start and end time values are taken from the user.

**Query availabilities:**

Each role can query available slots. User is prompted to enter candidate user id and interviewer user id. More than one interviewer can be queried by separating the interviewer ids by comma (,). If there exists a time slot which is available for both candidate and any of the interviewers, time slot start and end time are informed to the user.

