**CSC 540 – Project 1**

**Spring 2015**

**Due dates: Demo – March 26th & 27th. Final Report - April 2nd.**

**DATABASE APPLICATION PROJECT DESCRIPTION**

**University Student Housing**

The Director of the University Accommodation Office requires you to design a database to assist with the administration of the office. The requirements collection and analysis phase of the database design process has provided the following data requirements specification for the University Housing Office database followed by examples of query transactions that should be supported by the database.

*Note: Slight revisions are expected to be made to description for clarification based on questions that come up.*

**Data Requirements**

**Students**

The data stored on each full-time student includes: the student number, name (first and last name), phone number, alternate phone, home address (street, city, postcode), date of birth, sex, category of student (for example, first year undergraduate, postgraduate), nationality, smoker (yes or no), special needs, any additional comments, current status (placed/waiting), and what course the student is studying. If the student is a family student, then names and dates of birth for family members to reside in housing are also stored. Information about next of kin for a student is stored as well (relationship, address – street, city, postcode, and contact telephone number).

The university also allows university *guests* such as visiting scholars to request housing on campus. Such guests must first be approved and are assigned an approval id and similar information as students is stored as well as approval id.

**Housing Options**

There are different types of housing options on campus, including accommodation for full-time single students, couples and families.

Residence halls: each has a name, address, telephone number, and a hall manager who supervises the operation of the hall. A hall provides single rooms each with a room number, a place number, and monthly rent rate. The place number uniquely identifies each room in all halls controlled by the Housing office and is used when renting a room to a student.

Apartments: are either general student apartments or family apartments and are all fully furnished. General apartments provide single-room accommodation for groups of three or four students. The information held on student apartments includes an apartment number, address, and the number of single bedrooms and bathrooms available in each apartment. The apartment number uniquely identifies each apartment. Each bedroom in a apartment has a monthly rent rate, room number, and a place number. The place number uniquely identifies each room available in all apartments and is used when renting a room to a student. Family apartments are rented as a whole and are either one, two or three bedroom apartments. They have similar information stored as other apartments except for slight differences such as rate is for entire apartment, and they do not have place numbers.

The housing office also maintains of private accommodation - from rooms in shared houses, to apartments and family homes, for students who prefer to live off campus in the local community. However, this is not an option available to freshmen.

Students may rent a room in a hall of residence or apartment. Only university students and approved guests are eligible for university housing. Some specific residence halls are open to only graduate students and upperclass students.

**Leases**

A student may rent a room in a hall or student flat for various periods of time.

New lease agreements are negotiated at the start of each academic year with a minimum rental period of one semester and a maximum rental period of one year, which includes Semesters 1, 2, and the Summer Semester. **Students submit a request for accommodation along with up to three housing preferences (specific halls, or alternative housing options)**. The request is given a status of *Pending* and entered into Housing Office’s request list. If it is approved, its status is changed to *Processed*, and after it is signed by student, the status is changed to *InProgress* and after lease is over it is changed to *Completed.*

Each individual lease agreement between a student and the Accommodation Office is uniquely identified using a lease number. The data stored on each lease includes the lease number, duration of the lease (given as semesters), **name and matriculation number of the student, place number, room number, address details of the hall or student apartment, and the date the student wishes to enter the room, and the date the student wishes to leave the room (if known)**. Information about the terms of lease agreement is also included such as payment schedule options, monthly or once a semester, security deposit, and penalty for

early lease termination. Early termination penalty is certain percentage of the remaining lease before cut-off date after which the entire amount of remaining rent is the penalty.

To terminate a lease, a student sends a request with a reason and the date he/she wants to terminate. A termination request passes through the similar states as a lease request, *pending*, *processed* and *completed*. As part of processing a lease termination request, an inspection date is included to determine any incidentals or damages and to assign fees. Relevant information about the termination request is stored and given a unique request number.

**Parking**

The university accommodation also offers students parking using some dedicated parking lots that each have a limited number of parking spots. Each parking lot is identified by a unique number and a list of nearby housing options is associated with each parking lot. Parking spots are identified uniquely globally and some parking spots have special classification, handicapped, small car, large car and bike and each have different rental fees. As part of a lease request, a student may request a parking spot, denote if a special classification is needed and if a non-nearby spot assignment will be accepted. A parking spot nearby the student’s assigned housing option with the right classification is selected if possible and a unique parking permit id is assigned. Parking for housing students is only available to campus residents. Students may also opt for the more general student parking lots. The parking spots are given based on the availability and “first come first serve” basis.

**Invoices**

A student is issued an invoice monthly and includes at least two lineitems: the monthly housing rent, monthly parking rent and then a total. If the student opted for a single semester payment, then only a single invoice in the first month of lease. (Additional lineitems may also be added on the invoice by the housing administration such as late fees, or other incidental charges etc). Each invoice has a unique invoice number which is stored along with associated information such as lease number, payment due, due date, student’s full name and number, place number, room number, and name of hall or apartment. Additional data is also held on the payment of the invoice and includes the date the invoice was paid, the method of payment (check, cash, credit card, etc.). A final invoice is generated at the end of lease and it deducts any pending late fees, penalty, and damage charges from initial deposit. Each student should be able access a list of all invoices and status, paid or billed.

**Maintenance**

In case of any problem in the apartment, students should be able to raise a ticket for maintenance. When a student accesses the maintenance ticketing options, a list of ticket types are presented. In the system, each ticket type is associated with a severity, *Low*, *Medium* and *High* (not displayed to student) and a ticket with a unique number and associated information (type, date, student, location) is stored. Tickets are inserted into a ticket list in order that they are received but ordered by severity (i.e. higher severity tickets will be placed higher on list but after earlier same severity tickets)

**Housing Office Staff**

Some information is also held on members of staff of the Housing Office and includes the staff number, name (first and last name), home address (street, city, postcode), date of birth, sex, position (for example, Hall Manager, Administrative Assistant, Cleaner) and location (for example, Housing Office or Hall). Housing staff can access the list of pending lease, lease termination requests, and maintenance tickets and process them (approve, enter inspection date, damage charges, change status of tickets, etc). The information about which staff took any action on a lease or ticket and date is stored.

**Extra Credit** **(**max points 15): Implement a roommate matching functionality that will allow students with complimentary interests to be assigned to the same room. No specific requirements will be given for this extra credit functionality. It will be graded based on how comprehensive the functionality is. You may look up the NCSU housing website to see what kinds of features are typical of such a functionality.

**Application Requirements**

Most of the requirements of the application are embedded in the above description. In general, students and guests can view available housing options based on different criteria, initiate and check status of lease related requests, invoices and maintenance tickets. Housing staff can view and act on the different types of requests. A graphical user interface is not required (but will be accepted if team wants to implement one). A menu driven interface that presents lists of correct menu options based on role will be satisfactory. For the menu-based interface, a supplementary document will be provided in order to assure uniformity of menu option and their numbering. A sample of part of the supplementary document from previous projects is included below for your planning purposes.

You will be expected to implement about 8 - 10 queries which will largely fall under two categories (i) *reporting queries* and *retrieval queries* (this is in addition to basic insert and update functions).

**Note:** *these are representative. Final queries will be included in supplementary document.*

**Reporting Queries**

1. For each hall or apartment, display the total number of nearby parking spots (both available and not available)
2. For each hall, display total number of graduate students on the request list that do not yet have approved leases.
3. For each student renting, print their next invoice.
4. For each available parking spot, print its information (id, lot, location and type)

**Retrieval Queries**

1. Find the most popular hall or apartment (hall or apartment with the most number of requests approved, pending, or denied)
2. Find all students who have paid their rent after the due date for any of the past three months.
3. Print information about pending lease requests where student have not requested a parking spot.

**Sample of part of program flow supplement document from previous project (e-learning application). .**



**Tentative Grading Scheme**

E-R model – 25 pts

Create& Insert table (SQL) statements – 20 pts

SQL queries – 30 pts

Application specifications – 25pts

**To Submit:** You are to submit a zip file named following the convention “unityid1-unityid2-unityid3-unityid4-540S15-1”. It should consist of the following files+.

1. An E-R diagram and a short paragraph describing which constraints are not captured in the diagram**.**
2. A text file of create table statements capturing the E-R diagram in SQL and a short paragraph describing what constraints are captured and how e.g. using CHECK constraints or triggers or privileges etc.
3. A text file including the table population statements that you used i.e. INSERT statements to add data to your tables.
4. A text file including the SQL queries to the four query descriptions that will be provided to you. (TBA)

\*\*\*\* Learn how to use scripts so that recreating databases can be done easily.

We will have an in class demonstration of your projects. Schedule and instructions will be posted later.

Is matriculation number the same with student id.