As part of the BILL project, you will have to read in student records and users from a set of databases (functionally, these are actually flat files in the JSON format). We have included samples in this ZIP, and these tables will listing the fields that may appear, and examples of the type of data that might appear in the those fields.

You will need to implement Java classes to contain and represent this data.

Student Record Mapping:

<table>
<tr><th>Key</th><th>Value Type</th></tr>
<tr><td>student</td><td>Student class, contains fields:
<table>
<tr><th>Key</th><th>Value Type</th></tr>
<tr><td>id</td><td>string (ex: ggay)</td></tr>
<tr><td>firstName</td><td>string (ex: Greg)</td></tr>
<tr><td>lastName</td><td>string (ex: Gay)</td></tr>
<tr><td>phone</td><td>string, three integers, dash, three integers, dash, four integers (ex: 999-999-9999)</td></tr>
<tr><td>emailAddress</td><td>string (ex: mhunt@mailbox.sc.edu)</td></tr>
<tr><td>addressStreet</td><td>string (ex: 221B Baker St.)</td></tr>
<tr><td>addressCity</td><td>string (ex: Pittsburgh)</td></tr>
<tr><td>addressState</td><td>string (ex: SC)</td></tr>
<tr><td>addressPostalCode</td><td>String (ex: 29205)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>college</td><td>One of: {ARTS_AND_SCIENCES, ENGINEERING_AND_COMPUTING, GRADUATE_SCHOOL}</td></tr>
<tr><td>termBegan</td><td>Term class, contains fields:
<table>
<tr><th>Key</th><th>Value Type</th></tr>
<tr><td>semester</td><td>One of: {FALL, SPRING, SUMMER}</td></tr>
<tr><td>year</td><td>four digit number (ex: 2010)</td></tr>
</table></td></tr>
<tr><td>capstoneEnrolled</td><td>Term class, contains fields:
<table>
<tr><th>Key</th><th>Value Type</th></tr>
<tr><td>semester</td><td>One of: {FALL, SPRING, SUMMER}</td></tr>
<tr><td>year</td><td>four digit number (ex: 2010)</td></tr>
</table>
Allowed to be null (not enrolled in Capstone)</td></tr>
<tr><td>classStatus</td><td>One of: {FRESHMAN, SOPHOMORE, JUNIOR, SENIOR, MASTERS, PHD, GRADUATED}</td></tr>
<tr><td>gradAssistant</td><td>boolean</td></tr>
<tr><td>international</td><td>boolean</td></tr>
<tr><td>internationalStatus</td><td>One of: {SHORT_TERM, SPONSORED, NONE}</td></tr>
<tr><td>resident</td><td>boolean</td></tr>
<tr><td>activeDuty</td><td>boolean</td></tr>
</table>

<table>
<tr><td>veteran</td><td>boolean</td></tr>
<tr><td>freeTuition</td><td>boolean</td></tr>
<tr><td>scholarship</td><td>One of: {WOODROW, DEPARTMENTAL, GENERAL, ATHLETIC, SIMS, NONE}</td></tr>
<tr><td>studyAbroad</td><td>One of: {REGULAR, COHORT, NONE}</td></tr>
<tr><td>nationalStudentExchange</td><td>boolean</td></tr>
<tr><td>outsideInsurance</td><td>boolean</td></tr>
<tr><td>courses</td><td>List, containing 0 or more courses, each with the following fields:
<table>
<tr><th>Key</th><th>Value Type</th></tr>
<tr><td>name</td><td>string (ex: Machine Learning)</td></tr>
<tr><td>id</td><td>string (ex: csce740)</td></tr>
<tr><td>numCredits</td><td>integer</td></tr>
<tr><td>online</td><td>boolean</td></tr>
</table>
</td></tr>
<tr><td>transactions</td><td>List containing 0 or more transaction. Each transaction has the following fields:
<table>
<tr><th>Key</th><th>Value Type</th></tr>
<tr><td>type</td><td>One of: {PAYMENT, CHARGE}</td></tr>
<tr><td>transactionDate</td><td>Instance of Date class, containing:</td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td></td>
<td>
<table>
<tr>
<td></td>
<td>
<table>
<tr><th>Key</th><th>Value Type</th></tr>
<tr><td>month</td><td>two digit number (1-12)</td></tr>
<tr><td>day</td><td>number (1-31)</td></tr>
<tr><td>year</td><td>four digit number (ex: 2010)</td></tr>
</table>
</td>
</tr>
<tr><td>amount</td><td>double</td></tr>
<tr><td>note</td><td>String, intended to explain the transaction</td></tr>
</table>
</td>
</tr>
</table>

You will also need to produce bills in a similar format. An example has been included, but here is a guide to the fields that should appear.

Bill Mapping:

<table>
<tr><th>Key</th><th>Value Type</th></tr>
<tr><td>student</td><td>Student class, contains fields:
<table>
<tr><th>Key</th><th>Value Type</th></tr>
<tr><td>id</td><td>string (ex: ggay)</td></tr>
<tr><td>firstName</td><td>string (ex: Greg)</td></tr>
<tr><td>lastName</td><td>string (ex: Gay)</td></tr>
<tr><td>phone</td><td>string, three integers, dash, three integers, dash, four integers (ex: 999-999-9999)</td></tr>
<tr><td>emailAddress</td><td>string (ex: mhunt@mailbox.sc.edu)</td></tr>
<tr><td>addressStreet</td><td>string (ex: 221B Baker St.)</td></tr>
<tr><td>addressCity</td><td>string (ex: Pittsburgh)</td></tr>
<tr><td>addressState</td><td>string (ex: SC)</td></tr>
<tr><td>addressPostalCode</td><td>String (ex: 29205)</td></tr>
</table>
</td></tr>
<tr><td>college</td><td>One of: {ARTS_AND_SCIENCES, ENGINEERING_AND_COMPUTING, GRADUATE_SCHOOL}</td></tr>
<tr><td>classStatus</td><td>One of: {FRESHMAN, SOPHOMORE, JUNIOR, SENIOR, MASTERS, PHD, GRADUATED}</td></tr>
</table>

<table>
<tr><td>balance</td><td>double</td></tr>
<tr><td>transactions</td><td>List containing 0 or more transaction. Each transaction has the following fields:
<table>
<tr><th>Key</th><th>Value Type</th></tr>
<tr><td>type</td><td>One of: {PAYMENT, CHARGE}</td></tr>
<tr><td>transactionDate</td><td>Instance of Date class, containing:
<table>
<tr><th>Key</th><th>Value Type</th></tr>
<tr><td>month</td><td>two digit number (1-12)</td></tr>
<tr><td>day</td><td>number (1-31)</td></tr>
<tr><td>year</td><td>four digit number (ex: 2010)</td></tr>
</table>
</td></tr>
<tr><td>amount</td><td>double</td></tr>
<tr><td>note</td><td>String, intended to explain the transaction</td></tr>
</table>
</td></tr>
</table>

Two additional data sets also must be loaded into your system. One contains a full data set of the courses that the department offers currently, along with the number of credit hours:

Courses Mappings:

Key	Value Type
name	String, (example: Operating Systems)
id	String, four lowercase letters followed by three numbers (example: csce740)
numCredits	String, either a single number or a range (ex: 3, ex: 1-3)

Note - this is only a list of CSCE courses. Courses from other departments do not need to be validated (you can assume out-of-department courses are accurate).

The other is a simple permissions data set:

Permissions Mappings:

Key	**Value Type**
id	string of letters and numbers (ex: gayxx067)
firstName	string (ex: Greg)
lastName	string (ex: Gay)
role	String, one of {STUDENT, GRADUATE_PROGRAM_COORDINATOR}
department	String, all uppercase, with underscores instead of spaces (example: COMPUTER_SCIENCE)