SOFTENG754 ASSIGNMENT 3

Background

In Assignment 2, you have finished the user requirements analysis and identified a set of user stories/application features. Now, it is the time to implement the identified functionality of the application. You can use existing machine learning libraries – suggestions will be provided in the next lecture.

Tasks

You are required to use the Test-Driven Development (TDD) method to implement the application. Given the time limit, just the **backend** part of a subset of the application features generated in Assignment 2 are selected for the TDD implementation. Note that some functionality has been slightly modified from the previous assignments to reduce complexity. For example, you are not required to provide single sign on using existing accounts. Therefore, you will need to allow users to register.

Selected Feature List

1. Support two types of users: administrator and developer
2. User registration
3. User log in and off

4. Group similar forum questions using machine learning algorithms (e.g., clustering). You should store the following details for each cluster:
    a. Title
    b. Related forum posts
5. Store issue details. Each issue detail is a summary of multiple similar original forum questions (generated from the clustering). You should store the following information for each issue:
    a. Summary of the issue (external text summarisation algorithms might be required)
    b. Number of related posts
    c. Number of users (total across all posts)
    d. Cause of issue (context). Note that each related question may have a different context, so you should also use clustering techniques on the context to group these within each problem cluster. For each context, show its summary and the number of users reporting this context.
    e. Assignee(s).
    f. Current status.
6. Sort issues. Sort categories by priority (based on the number of forum posts related to the issue).

7. An administrator can add a forum post to a cluster.
8. An administrator can remove forum post from a cluster.
9. An administrator can delete a cluster – all related forum posts should be put into single question clusters.

10. An administrator can assign an issue to a developer
11. An administrator can unassign a developer from an issue

12. A developer can mark an issue as resolved

13. Data should be stored in database

Submission

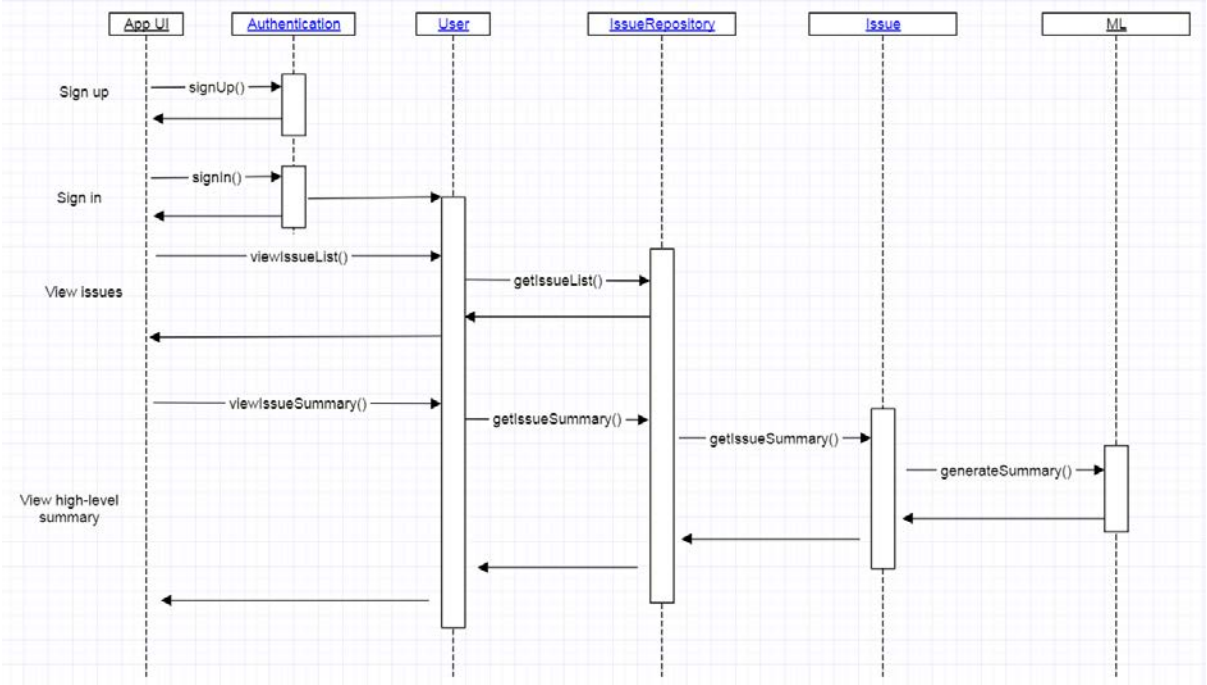
You will submit a report which includes:

- a link to the GitHub repo with your code,
- a matrix which links each implemented requirement to its associated test, and
- a brief discussion on your experiences using TDD and CI.

Your team will be required to present a demo of your implementation and discuss your experiences in using TDD and CI. Each group will present for 15 minutes, and all team members must take part in the presentation.
You should submit all documents before 11:59PM on the required date. All documents should have a title page which has on it the course number and name, the title of your project; the name of your group, a list of members indicating who was responsible for which parts, and the date. All submissions should be done via Canvas.

TDD Simple Example:

Note that this is just an example. Your tests/specifications and implementation can be totally differently from the classes and methods shown here.



Exemplary test/specification for the sign-up functionality

Requirement: The string length of password should not be less than 8

```
      @Test
      public void
shouldThrowPasswordFormatExceptionWhenPasswordLengthIsLessThan8() {
            Authentication authentication = new Authentication();

            expected.expect(PasswordFormatException.class);
            expected.expectMessage("Password length is less than 8!");

            authentication.signup("username", "123456");
      }
```

Database:

xero-dump.sql

```
CREATE TABLE question (
    question_id integer DEFAULT nextval('question_id_seq'::regclass) NOT NULL,
    text text,
    date date,
    author character varying(256),
    forum_details_id integer,
    content text,
    url character varying(256)
);
```

```
forumdb=# \dt
                List of relations
 Schema |          Name          | Type  |  Owner
--------+------------------------+-------+----------
 public | community              | table | postgres
 public | forum_details          | table | postgres
 public | keyword_statistics     | table | postgres
 public | question               | table | postgres
 public | reply                  | table | postgres
 public | training_data          | table | postgres
 public | training_data_category | table | postgres
(7 rows)

forumdb=#
forumdb=#
forumdb=# SELECT COUNT(*) FROM question;
 count
-------
 17984
(1 row)
```

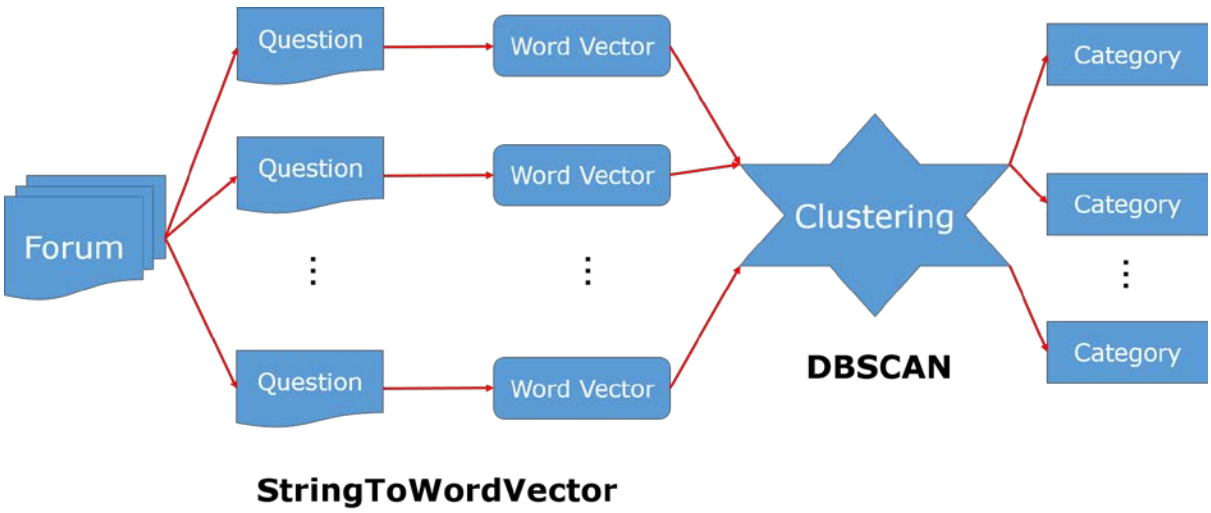
Sample:

sample10.txt

First row:

44330,"xero-php API, how to structure query",1/06/2017,Rajiv Pardiwala,2286,"I'm try to request bank transactions from a specific account from a certain date using the xero-php API.$xero = new PrivateApplication($config);print_r($xero->load('Accounting\BankTransaction')->setParameter('AccountID','XXXXXXX-4097-4028-9b99-XXXXXXXXX')->fromDate(new DateTime('2016-06-01 00:00:00'))->execute());which becomes this:""https://api.xero.com/api.xro/2.0/BankTransactions?AccountID=XXXXXXX-4097-4028-9b99-XXXXXXXXX&fromDate=2016-06-01""however, it is not filtering by AccountID and instead returns transactions for both of my accounts. The fromDate is ignored as well. What is the correct url to do what I want?",https://community.xero.com/developer/discussion/50795838/

Machine Learning:



Java:

Weka data mining lib: https://www.cs.waikato.ac.nz/ml/weka/

DBSCAN clustering algorithm Java implementation and usage

```
import weka.clusterers.DBSCAN;

...

      DBSCAN dbscan = new DBSCAN();
      dbscan.setEpsilon(0.12);
      dbscan.setMinPoints(5);
      dbscan.buildClusterer(data);

...
```

```
<!--
https://mvnrepository.com/artifact/nz.ac.waikato.cms.wek
a/weka-stable -->
     <dependency>
          <groupId>nz.ac.waikato.cms.weka</groupId>
          <artifactId>weka-stable</artifactId>
          <version>3.6.14</version>
     </dependency>
```

Python:

Scikit learn lib: http://scikit-learn.org/stable/

DBSCAN clustering algorithm python implementation and usage

```
from sklearn.cluster import  DBSCAN
...
      db = DBSCAN(eps=0.3, min_samples=10).fit(X)
...
```

TDD in Python:

PyDev Plug-in for Eclipse IDE

xUnit Frameowrks: PyUnit/unittest

```
import unittest
...
class Test(unittest.TestCase):
        def test_XXX(self):
                ...
...
```

For the assignment, either **Java** or **Python** is acceptable.