# Database Schema

* **The ‘runs’ table**: For storing the *url* and *urlid* with the timestamp for when crawling happened.

|  |  |  |
| --- | --- | --- |
| urlid | ts | url |
| 1 | 2016-10-22 00:00:00 | [www.yahoo.com](http://www.yahoo.com) |
| 2 | 2016-10-22 01:03:30 | [www.wikipedia.com](http://www.wikipedia.com) |
| … | … | …. |

CREATE TABLE runs (

urlid BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT,

ts TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

urlname VARCHAR(1083) NOT NULL,

PRIMARY KEY (urlid)

);

* **The ‘wordcount’ table**: Stores the word count for each *urlid* with the primary key (*urlid*, *word*).

|  |  |  |
| --- | --- | --- |
| urlid | word | count |
| 1 | article | 2 |
| 1 | been | 1 |
| 1 | created | 2 |
| … | … | … |
| 2 | also | 2 |
| 2 | the | 3 |
| … | … | … |

CREATE TABLE wordcount (

urlid BIGINT UNSIGNED NOT NULL,

word VARCHAR(1024),

count INT DEFAULT 0,

PRIMARY KEY (urlid, word)

);

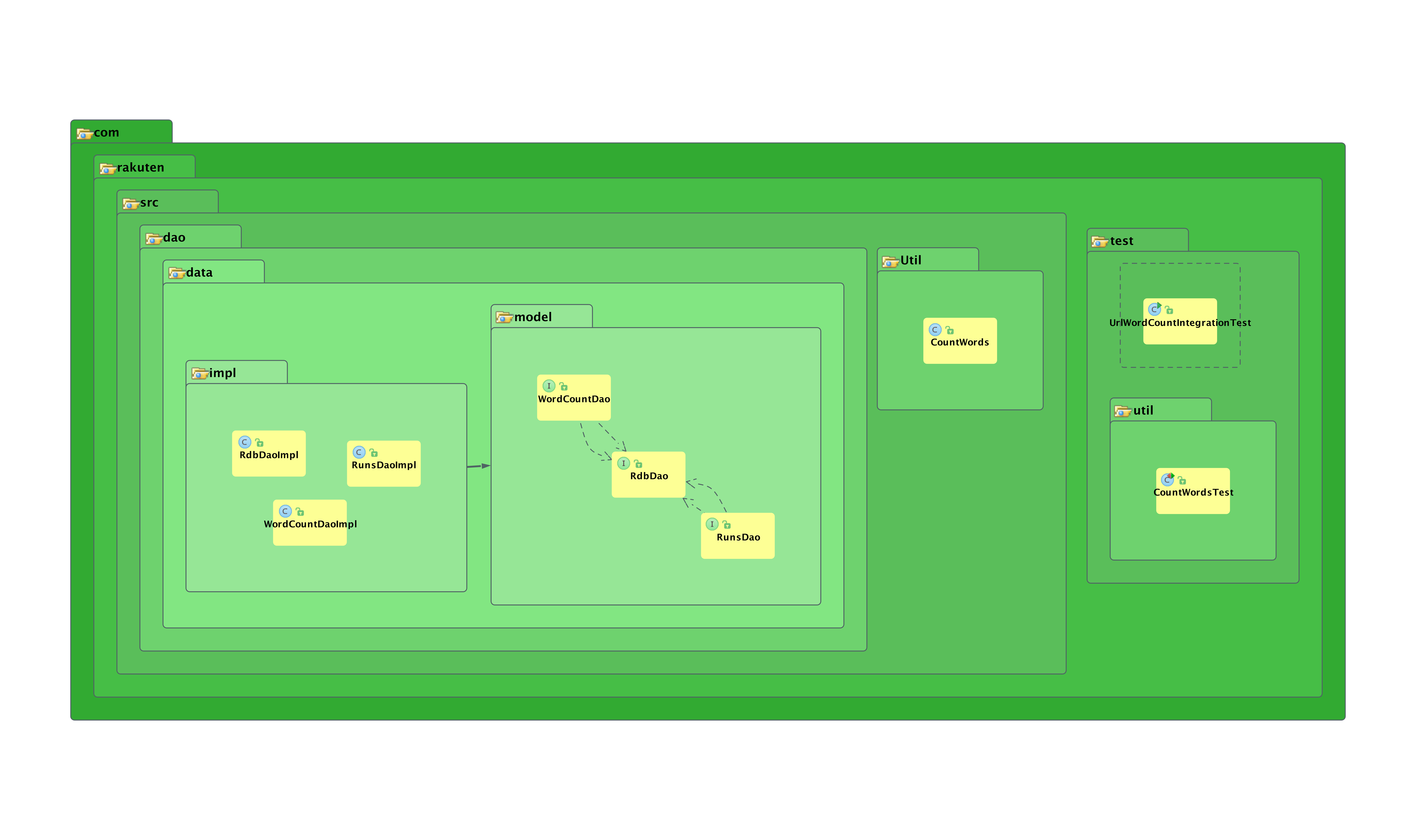
# Source code

[Attached]

* Contains JUnit unit tests for word counts in CountWordsTest.java
* Integration test UrlWordCountIntegrationTest.java.
* Project used was IntelliJ with libraries installed using maven.

# Design

* **UML diagram**



* All words are stored in lower case with their positive count.

# Libraries used

1. **com.robbypond:boilerpipe:1.2.32:** A fork of the boilerpipe package used to extract HTML text from an webpage.
2. **mysql:mysql-connector-java:5.1.402:** MySQL connector for storing records in the database.

# Scripts used, other information.

1. Install MySQL from <http://dev.mysql.com/doc/mysql-getting-started/en/>
2. Create the needed database and tables (script attached)
3. Add the above libraries using maven in IntelliJ.
4. <http://dev.mysql.com/doc/connector-j/5.1/en/connector-j-installing.html>

# Future improvements

* Multithread the application for multiple page crawling.
* Separate concerns of grabbing the page into a separate class given URL.
* Store database credentials separately in a encrypted config file or pass as command line parameters.
* Handle redirection of URL appropriately to avoid multiple redundant entries in the database.
* Handle multi-language tokenization.