

OpenRefine

SOFTWARE ARCHITECTURE AND DESIGN

Group 7



Agenda

หัวข้อที่จะนำเสนอ

01

What is OpenRefine

02

Architecture

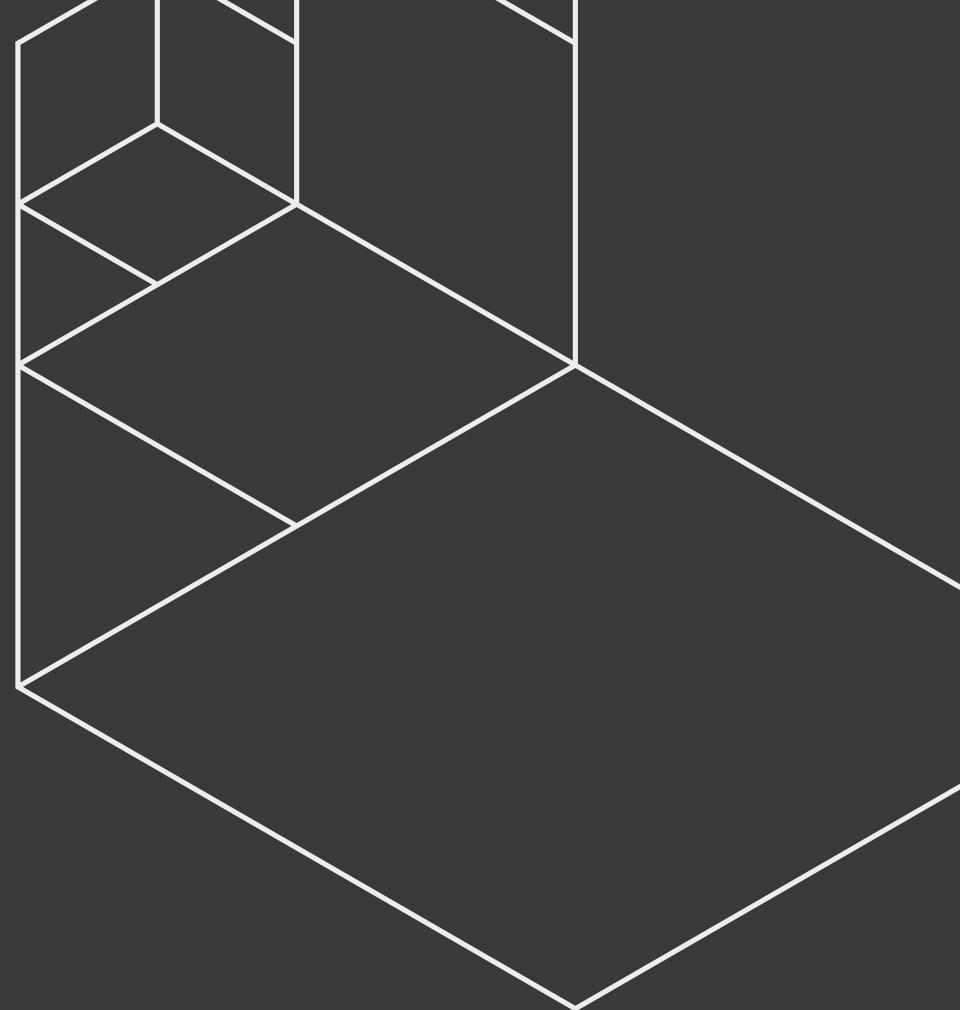
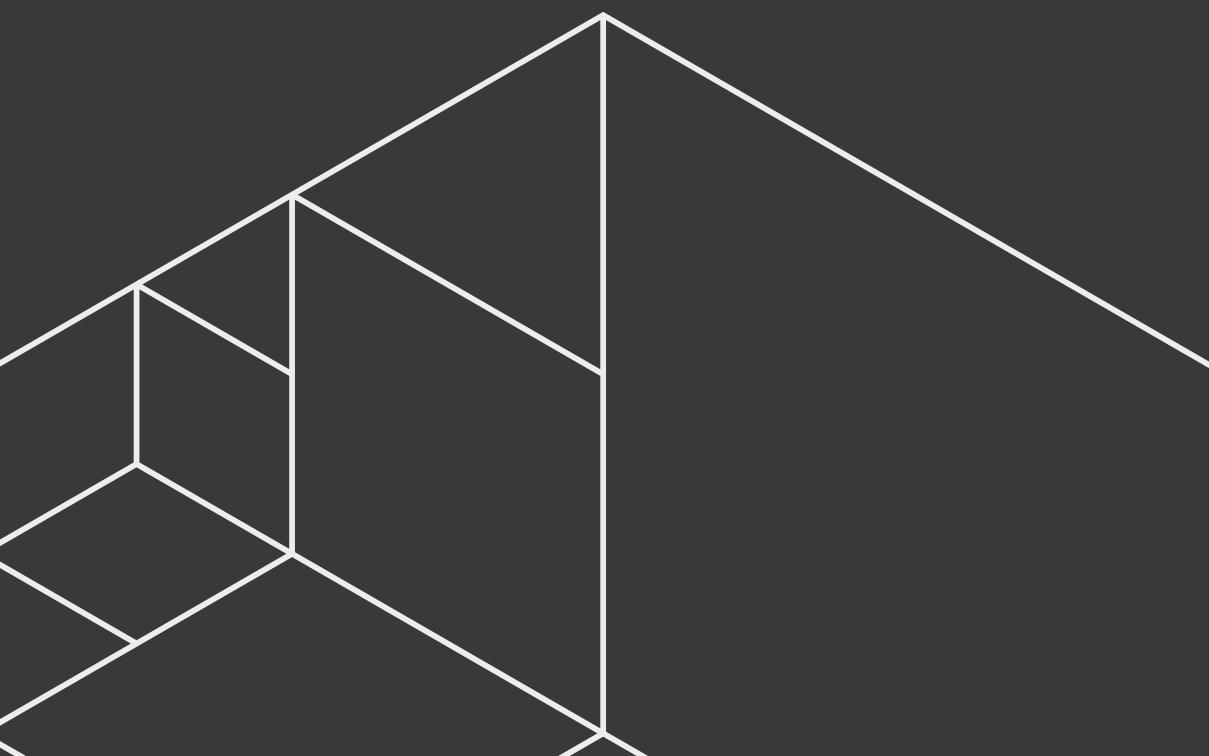
03

Quality Attribute

04

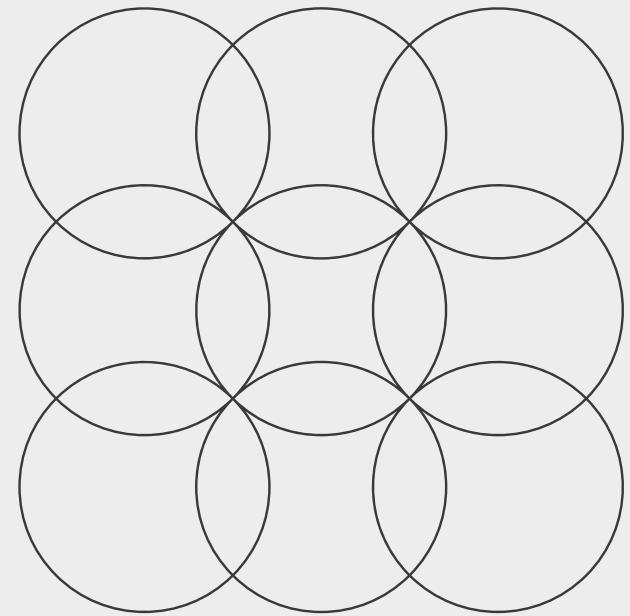
Design Pattern

OpenRefine คืออะไร ?



OpenRefine

- คือโปรแกรม (open-source) ที่ใช้สำหรับจัดการข้อมูล , กำความสะอาดข้อมูล ,
เปลี่ยนแปลงข้อมูลจากรูปแบบหนึ่งเป็นอีกรูปแบบหนึ่ง ซึ่งเรียกโดยทั่วไปว่า Data wrangling
- ตัวโปรแกรมมีลักษณะคล้ายกับแอพพลิเคชัน spreadsheet และสามารถจัดการกับรูปแบบไฟล์ spreadsheet เช่น csv ได้ แต่จะมีลักษณะเหมือนฐานข้อมูลมากกว่า
- ก่อนหน้านี้ได้รับการพัฒนาโดย Google (ภายใต้ชื่อ Google Refine)



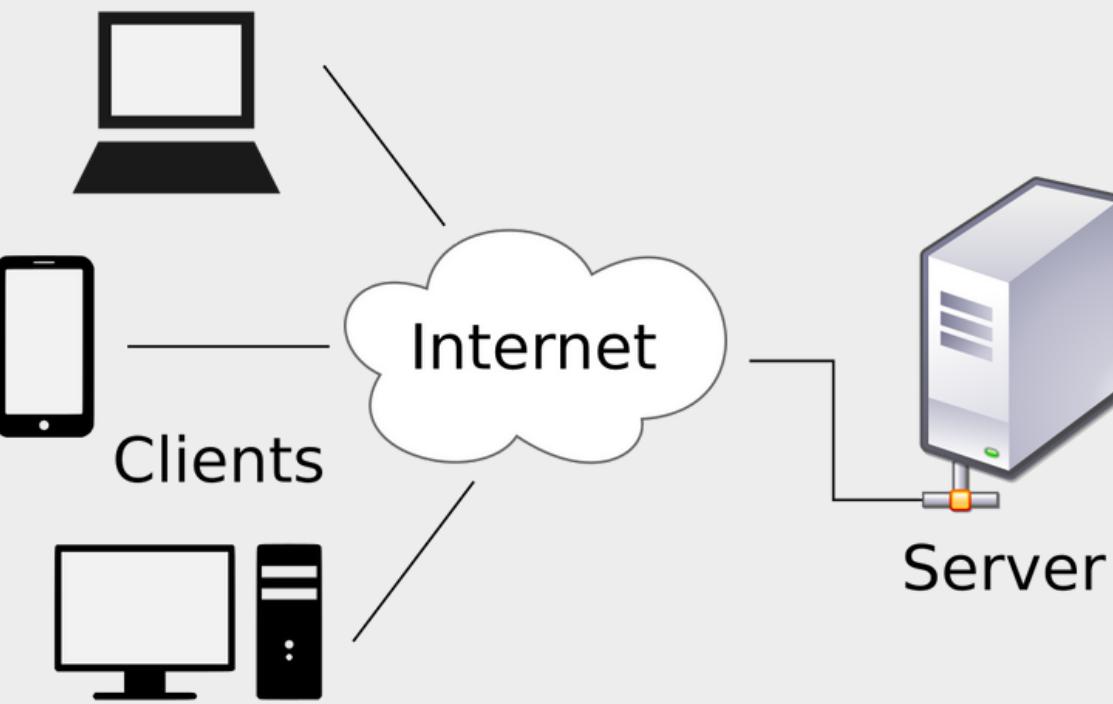
Commodity_Cod	Commodity_Des	Country_Code	Country_Name	Market_Year
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969
813300	Meal, Cottonseed	BR	Brazil	1969

OpenRefine



Architecture

Architecture



- **Client - Server Architecture**

OpenRefine เป็น Web application แต่ได้รับการออกแบบให้ทำงานบนเครื่อง โดยผู้ใช้งานจะเป็นตัวจัดการสถานะของข้อมูล เช่น เก็บประวัติการเลิกทำ/ทำซ้ำ (undo/redo), กระบวนการที่ใช้เวลาทำงานนาน เป็นต้น ในขณะที่ผู้ใช้งาน เป็นตัวจัดการสถานะของส่วนที่เชื่อมต่อกับผู้ใช้ (User Interface) เช่น การเลือกข้อมูล การแบ่งหน้า เป็นต้น และยังมีการเรียก ajax (GET, POST) เพื่อเกิดการเปลี่ยนแปลงกับข้อมูลและมีการดึงข้อมูลและสถานะที่เกี่ยวข้องกับข้อมูลจากผู้ใช้งาน

Architecture

- **Pros**

Architecture ดังกล่าว สามารถช่วยแยกการทำงานต่างๆออกจากกันได้ดี (เช่น ข้อมูลกับ UI) และยังมีการใช้เทคโนโลยีของ web ที่มีการใช้กันไป (เช่น HTML, CSS, JavaScript) สำหรับการนำ features ของ UI มาใช้งาน และการใช้งานผ่าน Server สามารถเรียกใช้ด้วยซอฟแวร์ third-party ได้โดยผ่าน GET, POST Operation ที่เป็นมาตรฐาน

- **Cons**

ข้อเสียของสถาปัตยกรรมแบบ client-server คือ หากเซิร์ฟเวอร์หลักหยุดทำงาน สถาปัตยกรรมทั้งหมดจะหยุดชะงัก รวมถึงการดำเนินการมีค่าใช้จ่ายสูงเนื่องจากต้นทุนของเครื่องมือฮาร์ดแวร์และซอฟต์แวร์จำนวนมาก และหากมีผู้ใช้งานจำนวนมากเกินไปในเวลาเดียวกันอาจทำให้เกิดปัญหา traffic congestion ได้ง่าย



Quality Attributes

Performance

Create Project [Start Over](#) [Configure Parsing Options](#)

[Open Project](#) [Import Project](#)

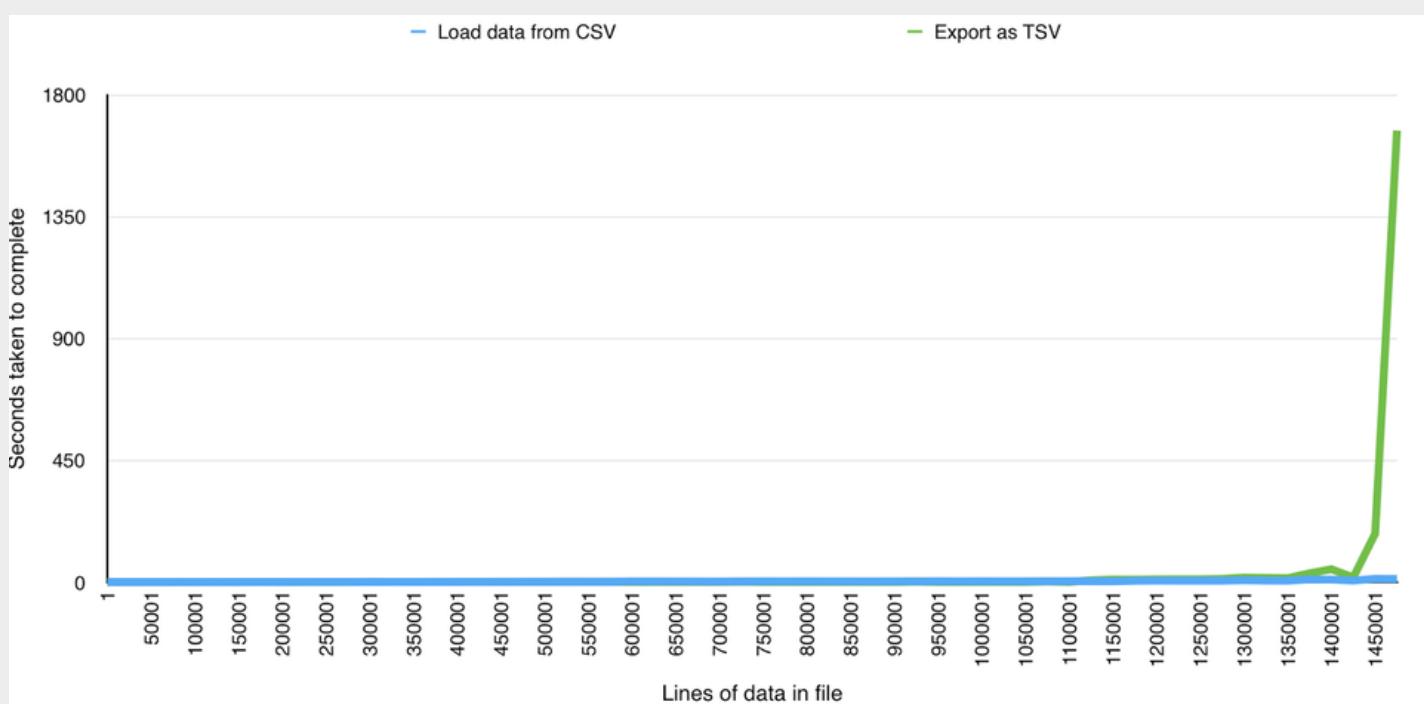
	navn	sum-rettighetshaver	sum-myndighet	avvik-rettighetshaver	avvik-myndighet	avklart	uavklart
1.	4Sea Energy AS	-3941	-5038	1097	0	1097	0
2.	A/S Norske Shell	9368192	9368192	0	0	0	0
3.	Agora Oil & Gas AS	-32092	-32092	0	0	0	0
4.	Altinex Oil Norway AS	105173	105173	0	0	0	0
5.	Bayerngas Norge AS	-545019	-545019	0	0	0	0
6.	Bayerngas Produksjon Norge AS	0	0	0	0	0	0
7.	BG Norge AS	-1220538	-1220538	0	0	0	0
8.	BP Norge AS	541041	541041	0	0	0	0
9.	Bridge Energy Norge AS	-141840	-141840	0	0	0	0
10.	Centrica Resources (Norge) AS	353999	354000	-1	0	-1	0
11.	Chevron Norge AS	246081	245726	355	0	355	0
12.	Concedo ASA	-83429	-83429	0	0	0	0
13.	ConocoPhillips Skandinavia AS	15157183	15158371	-1188	0	-1188	0
14.	Dana Petroleum Norway AS	-68343	-67292	-1051	0	-1051	0
15.	Det norske oljeselskap ASA	-2256144	-2300038	43892	0	43892	0
16.	Det norske oljeselskap AS	-67427	-68739	1312	0	1312	0
17.	DONG E&P Norge AS	1743717	1743717	0	0	0	0
18.	E.ON E&P Norge AS	-242115	-242115	0	0	0	0
19.	Edison International Norway Branch	-158298	-158298	0	0	0	0
20.	Eni Norge AS	9309843	9309843	0	0	0	0
21.	E&P Holding AS	-32906	-32906	0	0	0	0
22.	Enterprise Oil Norge AS	795364	795364	0	0	0	0
23.	ExxonMobil Exploration & Production Norway AS	33610336	33655138	-44802	0	-44802	0
24.	Faroë Petroleum Norge AS	-265316	-260282	4966	0	4966	0
25.	Fortis Petroleum Norway AS	-36135	-36135	0	0	0	0
26.	Front Exploration AS	-96879	-96764	1885	0	1885	0

Parse data as

Excel files Worksheets to Import: Sheet1: 65 rows Ignore first 0 line(s) at beginning of file Store blank rows Store blank cells as nulls
 Parse next 1 line(s) as column headers Store file source (file names, URLs) in each row
 Discard 0 initial row(s) of data Load at 0
 Line-based text files
 CSV / TSV / separator-based files
 Fixed-width field text files
 PC-Axis text files
 RDF/N3 files
 XML files

Help Version 2.5 [r2407]

เนื่องจาก OpenRefine เป็นซอฟต์แวร์ที่ต้องจัดการกับข้อมูลต่าง ๆ ที่อาจมีขนาดที่ใหญ่ ซึ่งจำเป็นต้องมีประสิทธิภาพในการทำงานของพังก์ชันต่าง ๆ ได้ดี ซึ่งจะส่งผลลับโดยตรงต่อประสิทธิภาพในการใช้งานต่อผู้ใช้ หากตัวซอฟต์แวร์นี้การใช้เวลามากในการประมวลผล



Interopability

The screenshot shows the OpenRefine interface with the title "OpenRefine 100 Most Populous Cities from Wikidata". On the left, there's a facet panel for "countryLabel" with a list of countries and their counts. The main area displays a table titled "100 rows" with columns: cityLabel, population, and countryLabel. The data includes the top 10 most populous cities:

Rank	cityLabel	population	countryLabel
1.	Shanghai	23390000	People's Republic of China
2.	Beijing	21710000	People's Republic of China
3.	Lagos	21324000	Nigeria
4.	Dhaka	16800000	Bangladesh
5.	Mumbai	15414288	India
6.	Istanbul	14657434	Turkey
7.	Tokyo	13942856	Japan
8.	Tianjin	13245000	People's Republic of China
9.	Guangzhou	13080500	People's Republic of China
10.	São Paulo	12106920	Brazil

หน้าจอหลัก โดยมีการเชื่อมต่อข้อมูล
กับ wikidata ซึ่งทำงานร่วมกัน

OpenRefine สามารถขยายด้วยบริการเว็บและข้อมูลภายนอกได้
และจะเก็บข้อมูลของคุณเป็นส่วนตัวบนคอมพิวเตอร์ของคุณเสมอ
จนกว่าคุณจะต้องการแบ่งปันหรือทำงานร่วมกัน ข้อมูลส่วนตัวของ
คุณจะไม่ออกจากคอมพิวเตอร์ของคุณเว็บแต่คุณต้องการ

OpenRefine สามารถใช้งานได้ทั้งบน Windows, Linux และ
MacOS เพียงแต่ว่าใช้งานและคำสั่งต่างๆ จะต่างกันไม่มากเท่านั้น

จะเห็นได้ว่า OpenRefine มีการทำงานร่วมกับคอมพิวเตอร์และผู้
ใช้ได้ดี และอำนวยความสะดวกให้กับผู้ใช้และคอมฯ มากขึ้นด้วย

Availability

55 rows			
Show as: rows records Show: 5 10 25 50 rows Sort ▾			
All	Author	Title	Country
24.	A. S. Byatt	Possession	United Kingdom
39.	Alan Hollinghurst	The Line of Beauty	
18.	Anita Brookner	Hotel du Lac	
53.	Anna Burns	Milkman	
42.	Anne Enright	The Gathering	
43.	Aravind Adiga	The White Tiger	India
32.	Arundhati Roy	The God of Small Things	India
27.	Barry Unsworth	Sacred Hunger	United Kingdom
25.	Ben Okri	The Famished Road	Nigeria
55.	Bernardine Evaristo	Girl, Woman, Other	United Kingdom
2.	Bernice Rubens	The Elected Member	United Kingdom
10.	David Storey	Saville	United Kingdom
38.	DBC Pierre	Vernon God Little	Australia
48.	Eleanor Catton	The Luminaries	New Zealand
52.	George Saunders	Lincoln in the Bardo	United States of America
31.	Graham Swift	Last Orders	United Kingdom
44.	Hilary Mantel	Wolf Hall	United Kingdom
47.	Hilary Mantel	Bring Up the Bodies	United Kingdom
45.	Howard Jacobson	The Finkler Question	United Kingdom
33.	Ian McEwan	Amsterdam	United Kingdom
12.	Iris Murdoch	The Sea, the Sea	Ireland
3.	J. G. Farrell	Troubles	United Kingdom
6.	J. G. Farrell	The Siege of Krishnapur	United Kingdom
17.	J. M. Coetzee	Life & Times of Michael K	South Africa

ตัวอย่างหน้าจอ ในการจัดข้อมูลที่ import เข้ามาในโปรแกรม

OpenRefine เป็นโปรแกรมที่สามารถโหลดข้อมูล จัดระเบียบ เพิ่มลดเปลี่ยน แก้ไขข้อมูลที่ไม่เป็นระเบียบ แปลง format ไฟล์ได้ แม้จะมีขนาดและจำนวนไฟล์ที่มากได้ โดยโปรเจคของ OpenRefine จะเป็นการนำเข้าข้อมูลที่มีอยู่ ทั้งจากไฟล์ในคอมพิวเตอร์ หรือไฟล์ใน server นอกจากนี้ ยังสามารถขยายด้วยบริการเว็บและข้อมูลภายนอกได้ด้วย

OpenRefine สามารถใช้งานโดยไม่ต้องเชื่อมต่ออินเทอร์เน็ตในการใช้งาน ยกเว้นตอน import ข้อมูลจาก web browser โปรแกรมจึงพร้อมใช้งานได้ตลอด

Testability

การทดสอบก้าวไป
การทดสอบ OpenRefine ก้าวไปเริ่มต้นด้วยรหัสต่อไปนี้

```
it('Ensure cells are blanked down', function () {
  cy.loadAndVisitProject('food.mini');
  cy.get('.viewpanel-sorting a').contains('Sort').click();
  cy.get('.viewpanel').should('to.contain', 'Something');
});
```

การทดสอบด้วยภาพ

การทดสอบด้วยภาพเป็นวิธีการทดสอบโดยการเปรียบเทียบภาพ
ภาพหน้าจออ้างอิงจะถูกถ่ายในครั้งแรกที่การทดสอบทำงาน และการ
เรียกใช้ในครั้งต่อๆ ไปจะเปรียบเทียบภาพหน้าจอใหม่กับข้อมูลอ้างอิงที่
ระดับพิเศษ

- การทดสอบรูปภาพที่สร้างโดยเบราว์เซอร์ OpenRefine (เช่น
แผนภาพกระจาย)

ภาพหน้าจออ้างอิง จะเก็บไว้ใน /cypress.snapshots และสามารถ
ถ่ายสแนปช็อตสำหรับทึ่งหน้าหรือเพียงส่วนเดียวของหน้าก็ได้

Testability

OpenRefine interface ได้รับการทดสอบด้วย
Cypress framework

การทดสอบจะดำเนินการยืนยันโดยใช้เบราว์เซอร์จริง
เช่นเดียวกับที่ผู้ใช้จริงจะใช้ซอฟต์แวร์
สามารถเรียกใช้โดย

- ใช้ Cypress test runner
(development mode)
- ใช้ command line (CI/CD mode)

แนวทางการทดสอบ

- cy.wait ควรใช้ในสถานการณ์ทางเลือกสุดท้าย
ถือเป็นการปฏิบัติที่ไม่ดี แม้บางครั้งไม่มีทาง
เลือกอื่น
- การทดสอบควรแยกออกจากกัน ทางที่ดีควร
ลองใช้ฟีเจอร์อย่างใดอย่างหนึ่งในขณะบัน
- การทดสอบควรเริ่มต้นด้วยโครงการใหม่
- ซื้อของไฟล์ควรสะท้อนองค์กร OpenRefine UI



Design Patterns

Singleton

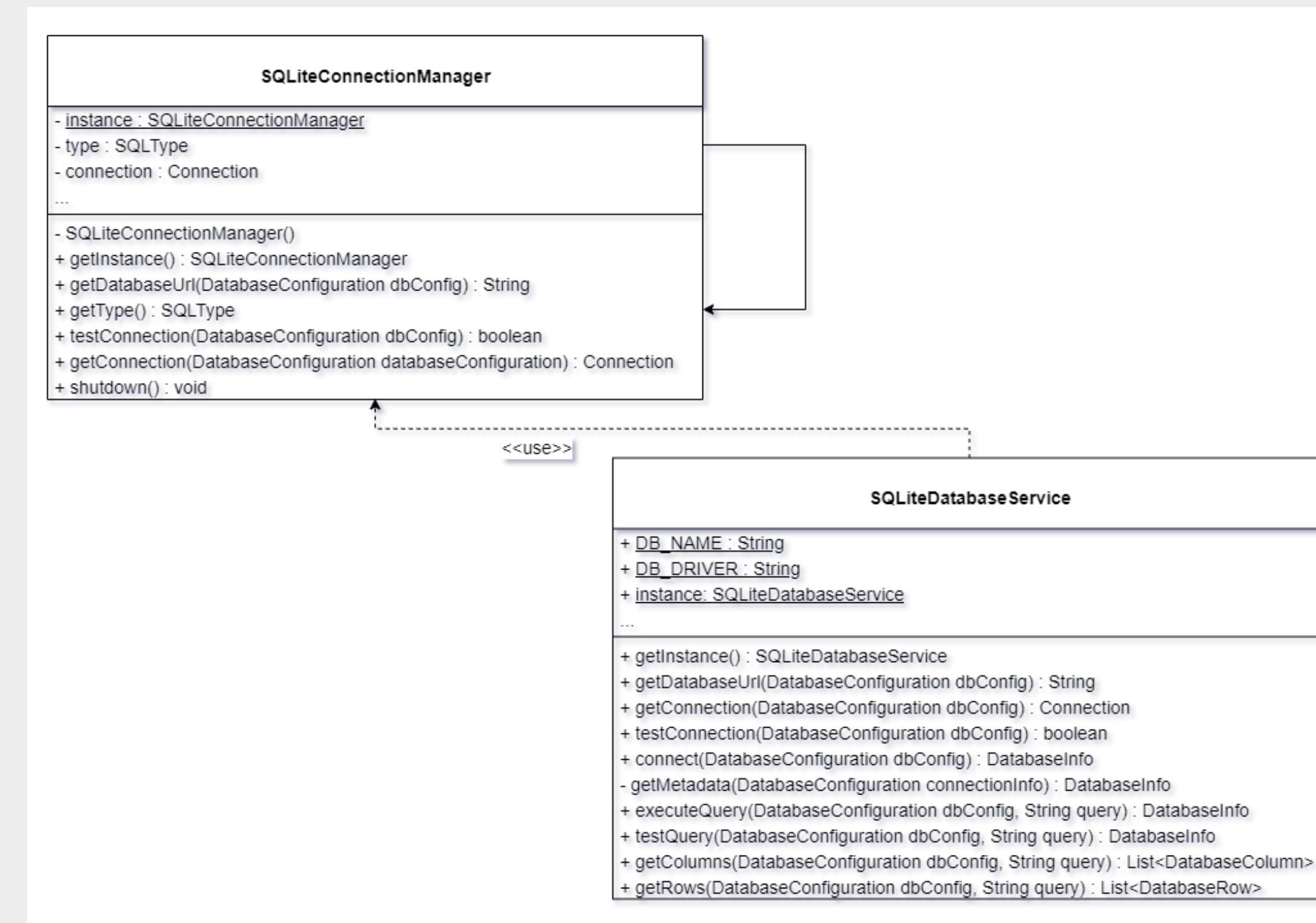
SQLiteConnectionManager.java

```
42 public class SQLiteConnectionManager {  
43  
44     private static final Logger logger = LoggerFactory.getLogger("SQLiteConnectionManager");  
45     private static SQLiteConnectionManager instance;  
46     private final SQLType type;  
47     private Connection connection;  
48  
49     private SQLiteConnectionManager() {  
50         type = SQLType.forName(SQLiteDatabaseService.DB_NAME);  
51     }  
  
53     /**  
54      * Create a new instance of this connection manager.  
55      *  
56      * @return an instance of the manager  
57      */  
58     public static SQLiteConnectionManager getInstance() {  
59         if (instance == null) {  
60             if (logger.isDebugEnabled()) {  
61                 logger.debug(":Creating new SQLite ConnectionManager :");  
62             }  
63             instance = new SQLiteConnectionManager();  
64         }  
65         return instance;  
66     }  
67 }
```

SQLiteDatabaseService.java

```
44 public class SQLiteDatabaseService extends DatabaseService {  
45  
46     public static final String DB_NAME = "sqlite";  
47     public static final String DB_DRIVER = "org.sqlite.JDBC";  
48     private static final Logger logger = LoggerFactory.getLogger("SQLiteDatabaseService");  
49     private static SQLiteDatabaseService instance;  
50  
51  
67     @Override  
68     public Connection getConnection(DatabaseConfiguration dbConfig) throws DatabaseServiceException {  
69         return SQLiteConnectionManager.getInstance().getConnection(dbConfig);  
70     }  
71  
72     @Override  
73     public boolean testConnection(DatabaseConfiguration dbConfig) throws DatabaseServiceException {  
74         return SQLiteConnectionManager.getInstance().testConnection(dbConfig);  
75     }  
76  
77     @Override  
78     public DatabaseInfo connect(DatabaseConfiguration dbConfig) throws DatabaseServiceException {  
79         return getMetadata(dbConfig);  
80     }  
81  
82     /**  
83      * @param dbConfig  
84      * @return  
85      * @throws DatabaseServiceException  
86      */  
87     private DatabaseInfo getMetadata(DatabaseConfiguration connectionInfo) throws DatabaseServiceException {  
88         try {  
89             Connection connection = SQLiteConnectionManager.getInstance().getConnection(connectionInfo);  
90             if (connection != null) {  
91                 java.sql.DatabaseMetaData metadata = connection.getMetaData();  
92                 int dbMajorVersion = metadata.getDatabaseMajorVersion();  
93                 int dbMinorVersion = metadata.getDatabaseMinorVersion();  
94             }  
95         } catch (SQLException e) {  
96             logger.error("Error connecting to database: " + e.getMessage());  
97         }  
98     }  
99 }
```

Singleton



TemplateMethod

ProjectManager.java

```
64  /**
65  * ProjectManager is responsible for loading and saving the workspace and projects.
66  *
67  */
68  */
69  public abstract class ProjectManager {
70     // last n expressions used across all projects
71     static public final int EXPRESSION_HISTORY_MAX = 100;
72
73     // If a project has been idle this long, flush it from memory
74     static protected final int PROJECT_FLUSH_DELAY = 1000 * 60 * 15; // 15 minutes
75
76     // Don't spend more than this much time saving projects if doing a quick save
77     static protected final int QUICK_SAVE_MAX_TIME = 1000 * 30; // 30 secs
78
79
80     protected Map<Long, ProjectMetadata> _projectsMetadata;
81     protected Map<String, Integer> _projectsTags; // TagName, number of projects have
82     protected PreferenceStore           _preferenceStore;
83 }
```

```
230  */
231  public void save(boolean allModified) {
232      if (allModified || _busy == 0) {
233          saveProjects(allModified);
234          saveWorkspace();
235      }
236  }
237
238  /**
239   * Saves the workspace to the data store
240   */
241  protected abstract void saveWorkspace();
```

ProjectManagerStub.java

```
49  /*
50  public class ProjectManagerStub extends ProjectManager {
51
52     @Override
53     public void deleteProject(long projectID) {
54         // empty
55     }
56 }
```

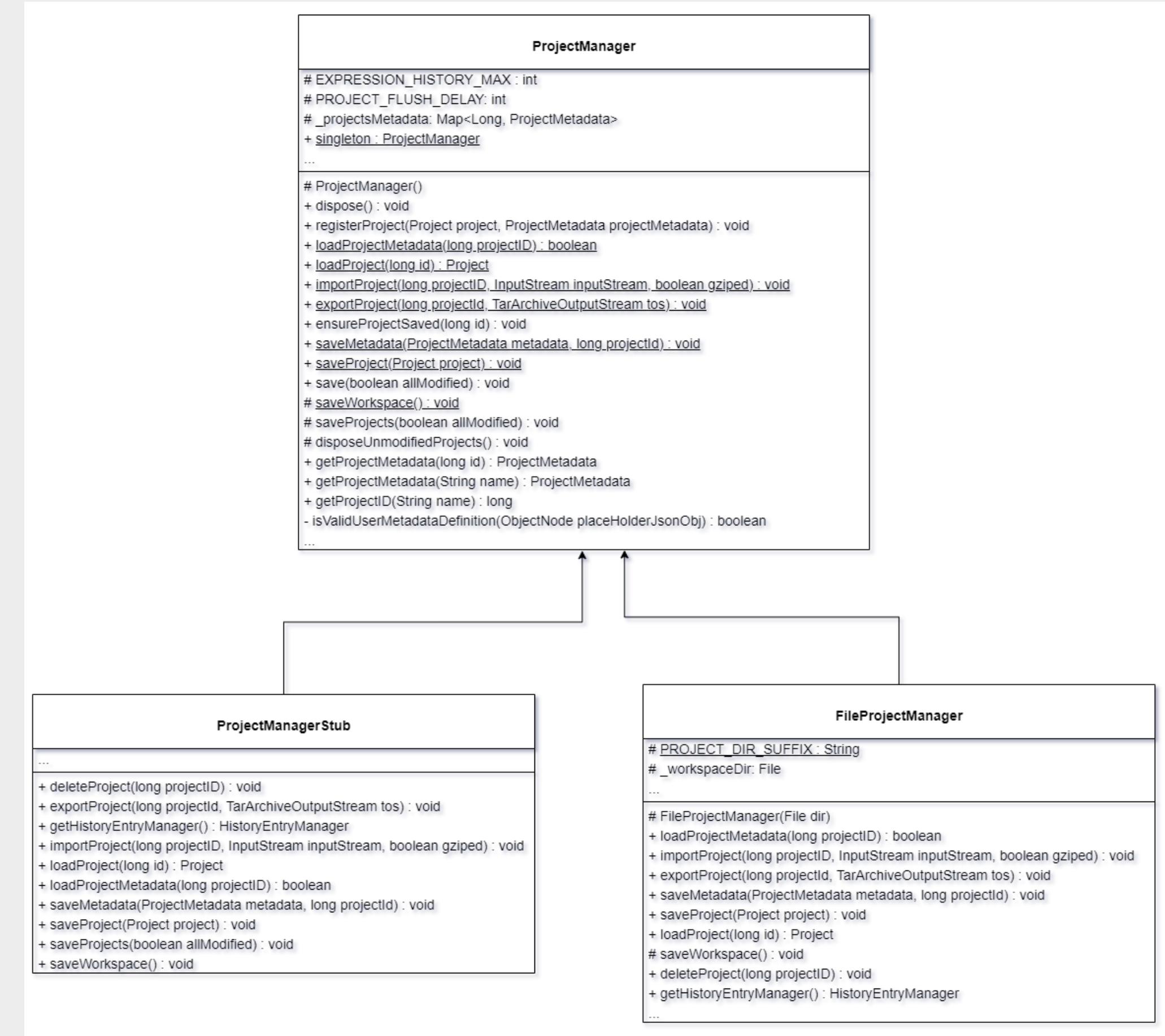
```
96     //Overridden to make public for testing
97     @Override
98     public void saveProjects(boolean allModified){
99         super.saveProjects(allModified);
100    }
101
102    @Override
103    protected void saveWorkspace() {
104        // empty
105    }
106 }
```

FileProjectManager.java

```
67  public class FileProjectManager extends ProjectManager {
68      final static protected String PROJECT_DIR_SUFFIX = ".project";
69
70      protected File           _workspaceDir;
71
72      final static Logger logger = LoggerFactory.getLogger("FileProjectManager");
73
74      static public synchronized void initialize(File dir) {
75          if (singleton != null) {
76              logger.warn("Overwriting singleton already set: " + singleton);
77          }
78          logger.info("Using workspace directory: {}", dir.getAbsolutePath());
79          singleton = new FileProjectManager(dir);
80          // This needs our singleton set, thus the unconventional control flow
81          ((FileProjectManager) singleton).recover();
82      }
83 }
```

```
259     @Override
260     protected void saveWorkspace() {
261         synchronized (this) {
262             // TODO refactor this so that we check if the save is needed before
263             // doing it
264             File tempFile = new File(_workspaceDir, child: "workspace.temp.json");
265             try {
266                 if (!saveToFile(tempFile)) {
267                     // If the save wasn't really needed, just keep what we had
268                     tempFile.delete();
269                     logger.info("Skipping unnecessary workspace save");
270                     return;
271                 }
272             } catch (Exception e) {
273                 e.printStackTrace();
274             }
275             logger.warn("Failed to save workspace");
276         }
277     }
278 }
```

TemplateMethod



OpenRefine-3.5.0\main\src\com\google\refine\ProjectManager.java

OpenRefine-3.5.0\main\tests\server\src\com\google\refine\ProjectManagerStub.java

OpenRefine-3.5.0\main\src\com\google\refine\io\FileProjectManager.java

Visitor

ExpressionNumericValueBinner.java

```
50 public class ExpressionNumericValueBinner implements RowVisitor, RecordVisitor {
51     /*
52      * Configuration
53      */
54     final protected RowEvaluable _rowEvaluable;
55     final protected NumericBinIndex _index; // base bins
56
57     /*
58      * Computed results
59      */
60     final public int[] bins;
61     public int numericCount;
62     public int nonNumericCount;
63     public int blankCount;
64     public int errorCount;
65
66     /*
67      * Scratchpad variables
68      */
69     protected boolean hasError;
70     protected boolean hasBlank;
```

```
@Override
public void start(Project project) {
    // nothing to do
}

@Override
public void end(Project project) {
    // nothing to do
}

@Override
public boolean visit(Project project, int rowIndex, Row row) {
    resetFlags();

    Properties bindings = ExpressionUtils.createBindings(project);
    processRow(project, rowIndex, row, bindings);

    updateCounts();

    return false;
}
```

RowVisitor.java

```
43 public interface RowVisitor {
44
45     /**
46      * Called before any visit() call.
47      *
48      * @param project
49      */
50     public void start(Project project);
51
52     /**
53      * @param project project
54      * @param rowIndex zero-based row index
55      * @param row row
56      * @return true to abort visitation early - no further visit calls will be made
57      */
58     public boolean visit(
59             Project project,
60             int rowIndex,
61             Row row
62     );
63
64     /**
65      * Called after all visit() calls.
66      *
67      * @param project
68      */
69     public void end(Project project);
70 }
```

Visitor

FilteredRows.java

```
42 public interface FilteredRows {  
43     /**  
44      * Go through the rows of the given project, determine wh  
45      * and call visitor.visit() on those that match, and poss  
46      * dependent rows.  
47      *  
48      * @param project  
49      * @param visitor  
50      */  
51     public void accept(Project project, RowVisitor visitor);  
52 }  
53
```

ConjunctionFilteredRows.java

```
49 public class ConjunctionFilteredRows implements FilteredRows {  
50     final protected List<RowFilter> _rowFilters = new LinkedList<RowFilter>();  
51  
52     public void add(RowFilter rowFilter) {  
53         _rowFilters.add(rowFilter);  
54     }  
55  
56     @Override  
57     public void accept(Project project, RowVisitor visitor) {  
58         try {  
59             visitor.start(project);  
60  
61             int c = project.rows.size();  
62             for (int rowIndex = 0; rowIndex < c; rowIndex++) {  
63                 Row row = project.rows.get(rowIndex);  
64                 if (matchRow(project, rowIndex, row)) {  
65                     visitor.visit(row);  
66                 }  
67             }  
68         } catch (Exception e) {  
69             throw new RuntimeException(e);  
70         }  
71     }  
72 }
```

FilteredRecordsAsFilteredRows.java

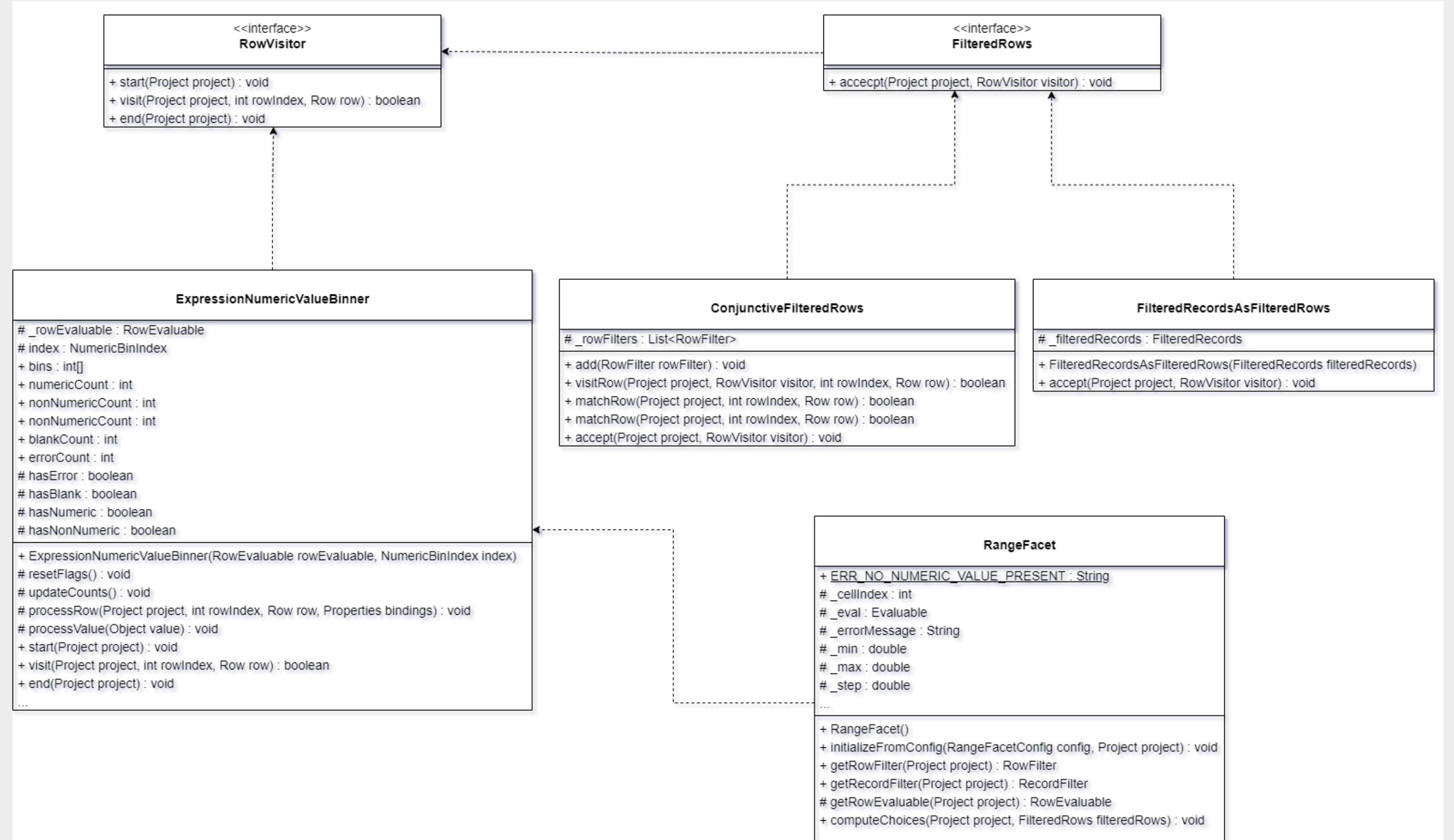
```
41 public class FilteredRecordsAsFilteredRows implements FilteredRows {  
42     final protected FilteredRecords _filteredRecords;  
43  
44     public FilteredRecordsAsFilteredRows(FilteredRecords filteredRecords) {  
45         _filteredRecords = filteredRecords;  
46     }  
47  
48     @Override  
49     public void accept(Project project, RowVisitor visitor) {  
50         _filteredRecords.accept(project, new RowVisitorAsRecordVisitor(visitor));  
51     }  
52  
53 }  
54
```

Visitor

RangeFacet.java

```
317     @Override
318     public void computeChoices(Project project, FilteredRows filteredRows) {
319         if (_eval != null && _errorMessage == null) {
320             RowEvaluable rowEvaluable = getRowEvaluable(project);
321
322             Column column = project.columnModel.getColumnByCellIndex(_cellIndex);
323             String key = "numeric-bin:row-based:" + _config._expression;
324             NumericBinIndex index = (NumericBinIndex) column.getPrecompute(key);
325             if (index == null) {
326                 index = new NumericBinRowIndex(project, rowEvaluable);
327                 column.setPrecompute(key, index);
328             }
329
330             retrieveDataFromBaseBinIndex(index);
331
332             ExpressionNumericValueBinner binner =
333                 new ExpressionNumericValueBinner(rowEvaluable, index);
334
335             filteredRows.accept(project, binner);
336             retrieveDataFromBinner(binner);
337         }
338     }
```

Visitor



OpenRefine-3.5.0\main\src\com\google\refine\browsing\facets\RangeFacet.java

OpenRefine-3.5.0\main\src\com\google\refine\browsing\util\ConjunctiveFilteredRows.java

OpenRefine-3.5.0\main\src\com\google\refine\browsing\util\ExpressionNumericValueBinner.java

OpenRefine-3.5.0\main\src\com\google\refine\browsing\FilteredRows.java

OpenRefine-3.5.0\main\src\com\google\refine\browsing\RowVisitor.java

OpenRefine-3.5.0\main\src\com\google\refine\browsing\util\FilteredRecordsAsFilteredRows.java

Team members

Group 7 - OpenRefine (v 3.5.0)

62010052 กันต์ มากกรพยสิน

62010243 ณพวัฒน์ คำยอด

62010309 ตาณ ชัยวงศ์ศรีอรุณ

62010355 suwa คำเพราะ

62010404 ธรรมบุญ กิจเลิศไฟโรมัน

62010462 นาพร ตึ้งใจ