

University timetabling (unitime) maintenance

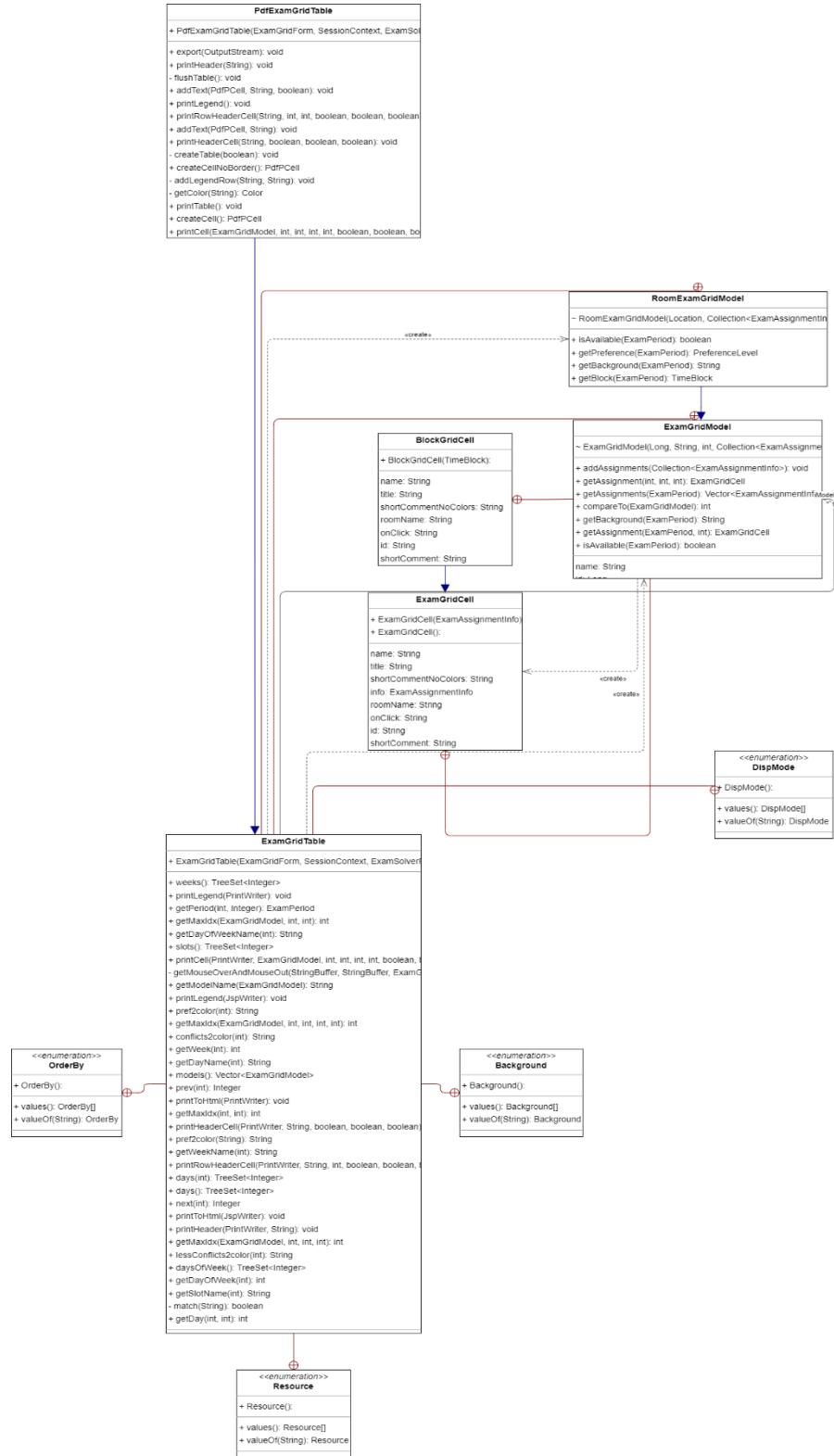
Phase 2

Name	ID
Youssef Hussein Mahmoud	20206111
Ahmed Nasr Hassan	20206129
Omar Sayed Ramadan	20206042
Mahmoud Hassan	20206071

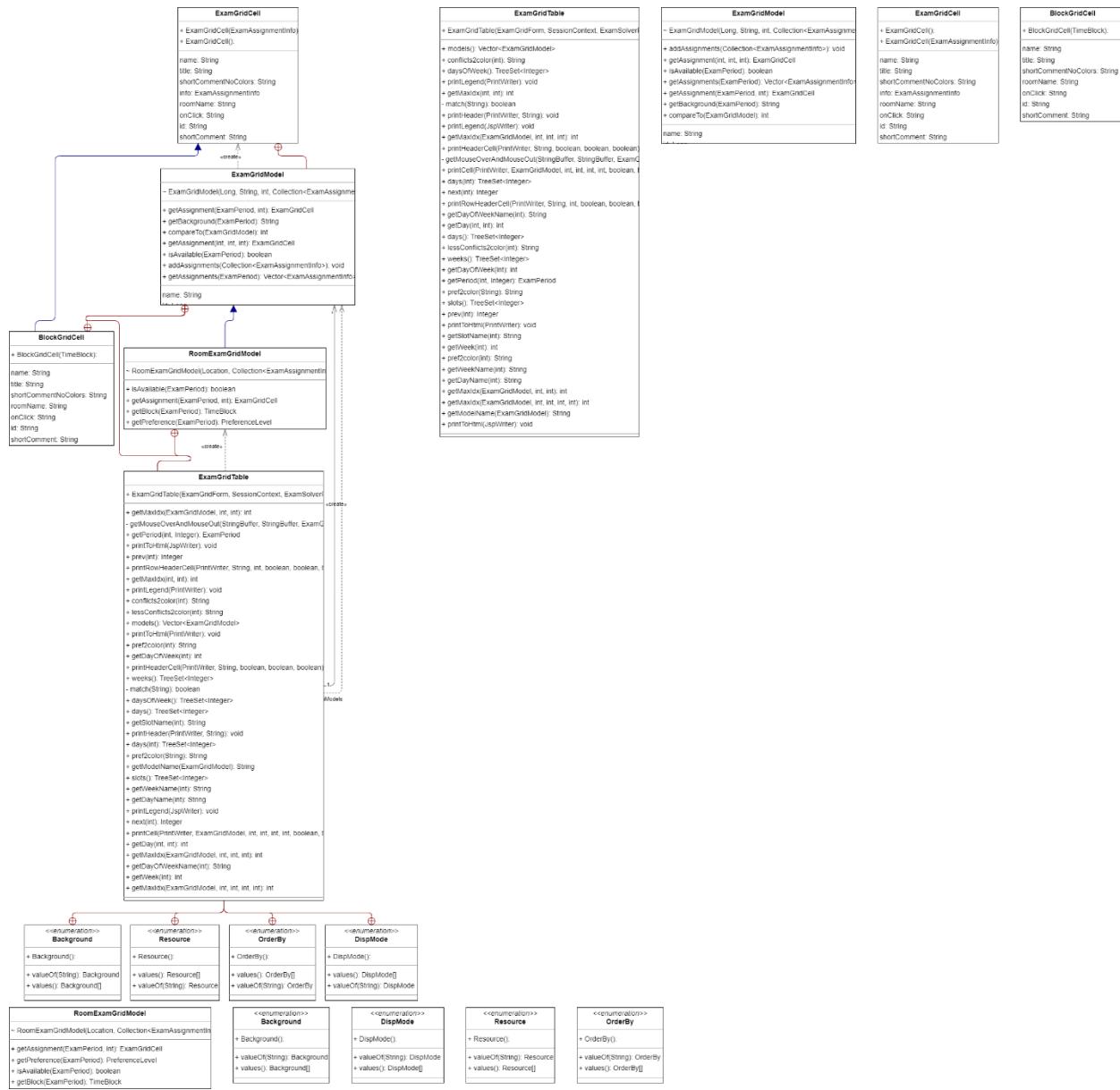
GitHub repository link: <https://github.com/Youssef-Wahba/University-Timetabling-Maintainice>

Jira Project: <https://youssefhussein1.atlassian.net/jira/core/projects/UT/board>

UML diagram before resolving bugs:



UML diagram after resolving bugs:



Static Analysis Report before changes:

The image displays two screenshots of the SonarQube interface, showing the 'Issues' and 'Measures' sections for the 'unitime' project.

Issues Report (localhost:9000/project/issues?resolved=false&id=unitime):

- Overview:** Shows a summary of issues: 969 Bugs, 136 Vulnerabilities, 51k Code Smells, 205 Blockers, 14k Critical, 17k Major, 52k Main code, 0 Test code, 52k Unresolved, 0 False Positive, 0 Fixed, 0 Removed, 0 Won't Fix.
- Issues List:** A list of recent findings:
 - Remove this commented out code. (Code Smell, Major, Open, Not assigned, 5min effort, Comment) - 13 hours ago, L88, unused.
 - Remove this commented out code. (Code Smell, Major, Open, Not assigned, 5min effort, Comment) - 13 hours ago, L75, unused.
 - Remove this commented out code. (Code Smell, Major, Open, Not assigned, 5min effort, Comment) - 13 hours ago, L60, unused.
 - Remove this commented out code. (Code Smell, Major, Open, Not assigned, 5min effort, Comment) - 13 hours ago, L128, unused.
 - Complete the task associated to this "TODO" comment. (Code Smell, Info, Open, Not assigned, 0min effort, Comment) - 13 hours ago, L178, cwe.
 - Complete the task associated to this "TODO" comment. (Code Smell, Info, Open, Not assigned, 0min effort, Comment) - 13 hours ago, L157, unused.

Measures Report (localhost:9000/component_measures?id=unitime):

- Overview:** Shows a Project Overview with sections: Reliability, Security, Security Review, Maintainability, Coverage, Duplications, Size, Complexity, and Issues.
- Measures List:** A chart titled 'unitime' showing Risk vs. Coverage. The chart uses color coding for reliability and security ratings (A-E) and bubble size for lines of code. A large green bubble is visible near the bottom center, indicating high risk and low coverage.

The screenshot displays the SonarQube web interface for the 'unitime' project. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and a search bar. A banner at the top right indicates that the last analysis has warnings, with a 'See details' link and a 'Version 4.8' badge.

The main content area shows the 'Code' tab selected, displaying metrics for the 'unitime' project. The table includes columns for Lines of Code, Bugs, Vulnerabilities, Code Smells, Security Hotspots, Coverage, and Duplications. The data shows 543,029 lines of code, 969 bugs, 136 vulnerabilities, 50,625 code smells, 472 security hotspots, 0.0% coverage, and 12.6% duplications. A note at the bottom of this section states: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine."

The bottom section shows the 'Projects' tab selected, listing the 'unitime' project with a status of 'Passed'. It provides a summary of bugs (969), vulnerabilities (136), hotspots reviewed (0.0%), code smells (51k), coverage (0.0%), and duplications (12.6%). The project is identified as Java XML. On the left, filters are available for Duplications, Size, Languages, and Tags. A note at the bottom of this section is identical to the one above: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine."

The screenshot shows the SonarQube dashboard for the 'unitime' project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and a search bar. A message indicates that the last analysis has warnings, with a link to 'See details'. The version of the dashboard is 4.8.

The main content area is divided into sections: 'QUALITY GATE STATUS' and 'MEASURES'.

QUALITY GATE STATUS: Shows a green box labeled 'Passed' with the subtext 'All conditions passed.'

MEASURES:

- New Code:** Since May 5, 2023. Started 13 hours ago. Status: E
- Overall Code:** 969 Bugs. Status: E
- 136 Vulnerabilitys.** Status: E
- 472 Security Hotspots.** 0.0% Reviewed. Status: E
- 1301d** Debt. Status: A
- 51k Code Smells.** Status: A
- Maintainability:** A

At the bottom, there's a taskbar with icons for weather (24°C), search, and various system applications. The date and time are also displayed.

Static Analysis Report after changes:

The screenshot shows the SonarQube interface for the 'unitime-after-resolve' project. The 'Code' tab is selected, displaying a detailed report for the 'timegrid' component. The report includes a table with metrics for three files: 'timegrid', 'ExamGridTable.java', and 'PdfExamGridTable.java'. A note at the bottom indicates that an embedded database should be used for evaluation purposes only.

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
timegrid	1,510	0	0	171	0	0.0%	4.2%
ExamGridTable.java	1,052	0	0	155	0	0.0%	4.8%
PdfExamGridTable.java	458	0	0	16	0	0.0%	2.7%

Below the table, a note states: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine."

The SonarQube interface also shows the following details:
Community Edition - Version 10.0 (build 59432) - GPLV3 - Community - Documentation - Plugins - Web API
34°C 5/17/2023 5:03 PM

This screenshot shows the SonarQube interface for the 'unitime-after-resolve' project. The 'Code' tab is selected, displaying a report for the 'unitime-after-resolve' component. The report includes a table with metrics for the 'JavaSource' component. A note at the bottom indicates that an embedded database should be used for evaluation purposes only.

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
unitime-after-resolve	543,028	918	136	50,627	472	0.0%	12.6%

Below the table, a note states: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine."

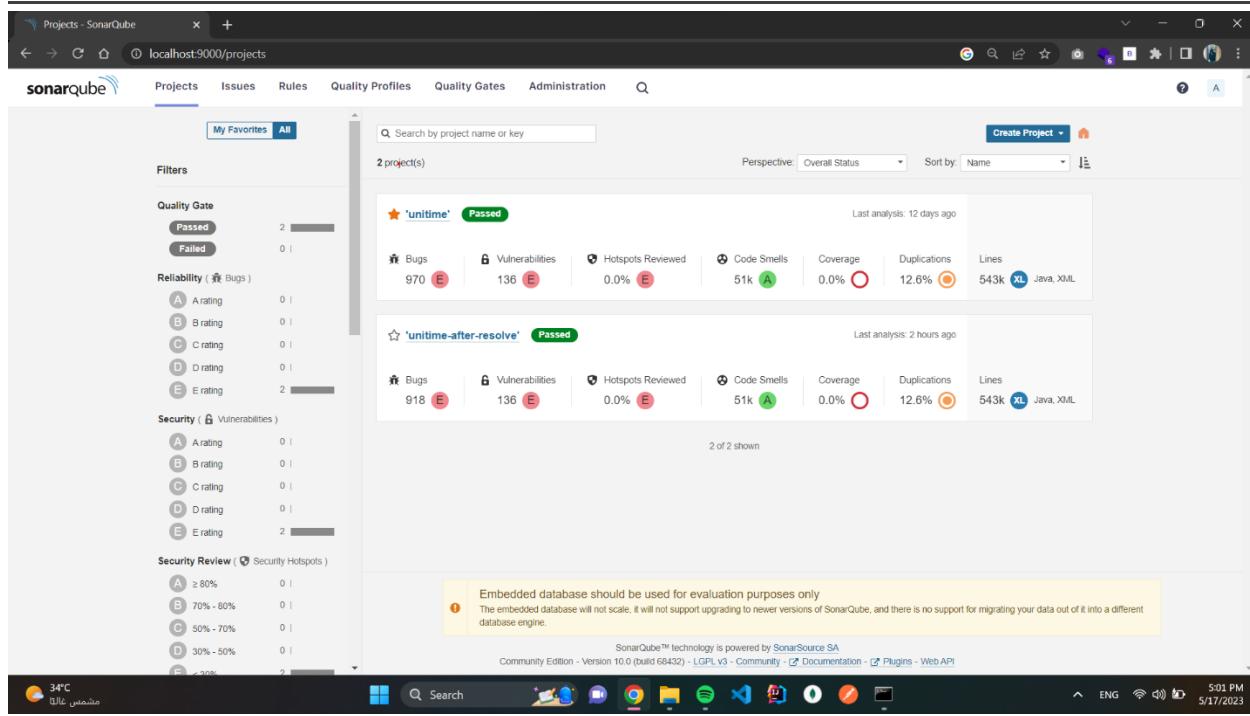
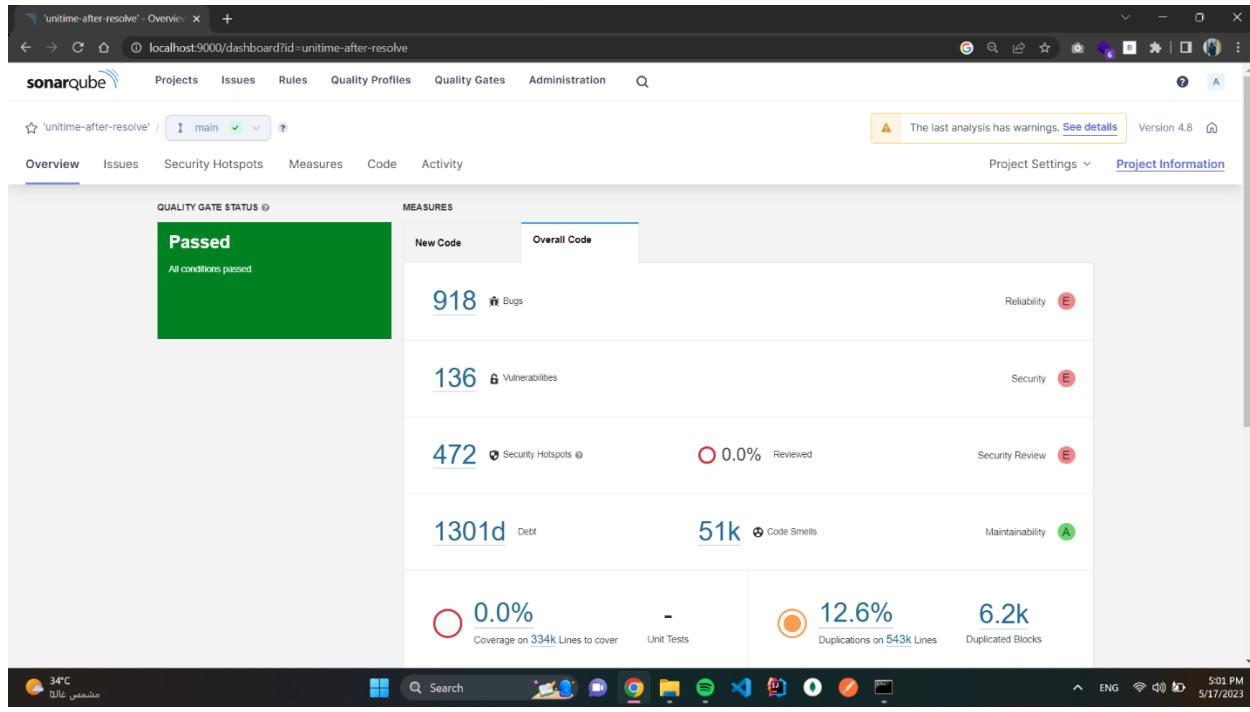
The SonarQube interface also shows the following details:
Community Edition - Version 10.0 (build 59432) - GPLV3 - Community - Documentation - Plugins - Web API
34°C 5/17/2023 5:03 PM

SonarQube Project Overview - 'unitime-after-resolve'

The screenshot shows the SonarQube interface for the project 'unitime-after-resolve'. The top navigation bar includes 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', 'Administration', and a search bar. A banner at the top right indicates 'The last analysis has warnings. See details'.

Measures Tab: This tab is selected, showing a 'Project Overview' sidebar with categories like Reliability, Security, and Coverage. The main area displays a 'Risk' chart where nodes are colored by Reliability Rating and Security Rating, and sized by Lines of Code. A bubble chart below shows 'Coverage' from 0.0% to 100%.

Issues Tab: This tab is also present, showing a list of 51,681 issues across various files. The list includes items such as 'Remove this commented out code.' and 'Complete the task associated to this "TODO" comment.', each with details like severity (e.g., Major, Minor, Info), status (e.g., Open, Not assigned), and effort.



Jira issue tickets and status:

The screenshot shows the Jira Board interface for the "University Timetabling" project. The board has three columns: "TO DO" (0), "IN PROGRESS" (0), and "DONE" (4). The "DONE" column contains a single issue titled "Fixing .equals() issues in file 'JavaSource/org/unitime/timetable/webutil/timegrid/ExamGridTable.java'". The issue details are as follows:

Summary	Assignee	Reporter	Priority	Labels	Due date	Time tracking	Start date	Category
Fixing .equals() issues in file "JavaSource/org/unitime/timetable/webutil/timegrid/ExamGridTable.java"	UT-1	Youssef Hussein Wahba	High	None	None	No time logged	May 05, 2023	None

The issue description states: "Comparing with strings and boxed types in Java should be done with `equals()` but boxed types comparing in this file is done with the operator `==`".

The screenshot shows the Jira Issue view for issue "UT-3". The issue title is "Fixing .equals() issues in file 'JavaSource/org/unitime/timetable/webutil/timegrid/ExamGridTable.java'". The issue details are identical to the ones shown in the Board view.

A screenshot of a Jira issue page for UT-2. The issue title is "merge if statement in 'JavaSource/org/unitime/timetable/webutil/timegrid/PdfExamGridTable.java'". The description states: "merge the **if statement** in line 253 with the enclosing one to increase performance". The activity tab shows comments, history, and work log. A right-click context menu is open over the issue summary, showing options like "Done", "Actions", "Details", "More", and "Configure". The "Details" panel is expanded, showing the following fields:

Field	Value
Assignee	Youssef Hussein Wahba
Reporter	Youssef Hussein Wahba
Priority	High
Labels	Java replace
Due date	None
Time tracking	No time logged
Start date	May 05, 2023
Category	None

Created May 5, 2023 at 7:31 PM Updated 16 hours ago

A screenshot of a Jira issue page for UT-4. The issue title is "use the original value instead 'JavaSource/org/unitime/timetable/webutil/timegrid/ExamGridTable.java'". The description says: "use the original value instead 'JavaSource/org/unitime/timetable/webutil/timegrid/ExamGridTable.java'" and includes a note: "• Unsaved changes". The activity tab shows comments, history, and work log. A right-click context menu is open over the issue summary, showing options like "Done", "Actions", "Details", "More", and "Configure". The "Details" panel is expanded, showing the following fields:

Field	Value
Assignee	Omar Sayed
Reporter	Youssef Hussein Wahba
Priority	Low
Labels	None
Due date	None
Time tracking	No time logged
Start date	May 14, 2023
Category	None

Created 15 hours ago Updated 1 minute ago

The screenshot shows a Jira issue details page for a project named "University". The issue is titled "Fixing .equals() issues in file JavaSource/org/unitime/timetable/webutil/timegrid/PdfExamGridTable.java". The issue has one comment from Ahmed Nasr. The issue details panel shows the following information:

Assignee	Ahmed Nasr
Reporter	Youssef Hussein Wahiba
Priority	High
Labels	Java comparison
Due date	None
Time tracking	No time logged
Start date	May 05, 2023
Category	None
Created May 5, 2023 at 7:12 PM	
Updated 16 hours ago	

The issue was created on May 5, 2023, at 7:12 PM and was updated 16 hours ago. The issue is currently labeled as "Done".

File 1(JavaSource/org/unitime/timetable/webutil/timegrid/ExamGridTable.java):

Before

The screenshot shows two instances of the SonarQube web interface, both displaying analysis results for the file `ExamGridTable.java`.

Top Window:

- Title Bar:** Board - University Timetabling - Use the original value instead -
- SonarQube Header:** Projects Issues Rules Quality Profiles Quality Gates Administration
- Breadcrumbs:** ★ 'unitime' / main
- Status Bar:** The last analysis has warnings. See details Version 4.8
- Issues List:** A sidebar on the left lists several issues under the heading "Use the original value instead." These include:
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Use the original value instead.
- Code Editor:** The main area shows the Java code for `ExamGridTable.java`, with the first few lines of the `compareTo` method highlighted:

```
public int compareTo(ExamGridModel model) {
    switch (OrderBy.values()[iForm.getOrder()]) {
        case NameAsc :
            return getIName().compareTo(model.getName());
        case NameDesc :
            return -getIName().compareTo(model.getName());
    }
    return getId().compareTo(model.getId());
}
```
- Bottom Status Bar:** 24°C, 5/17/2023, 2:16 AM

Bottom Window:

- Title Bar:** Board - University Timetabling - Strings and Boxed types should be compared using "equals()".
- SonarQube Header:** Projects Issues Rules Quality Profiles Quality Gates Administration
- Breadcrumbs:** ★ 'unitime' / main
- Status Bar:** The last analysis has warnings. See details Version 4.8
- Issues List:** A sidebar on the left lists several issues under the heading "Strings and Boxed types should be compared using "equals()". These include:
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
 - Strings and Boxed types should be compared using "equals()".
- Code Editor:** The main area shows the Java code for `ExamGridTable.java`, with the `printHeaderCell` method highlighted:

```
private void printHeaderCell(out, name, vertical, false, false);
TreeSet<Integer> days = days(), slots = slots(), weeks = weeks(), daysOfWeek = daysOfWeek();
if (iForm.getDispMode() == DispMode.InRowHorizontal.ordinal()) {
    for (Integer day : days) {
        for (Integer slot : slots) {
            boolean end = (slot == slots.last());
            boolean eol = (end && day == days.last());
            if (iForm.getDispMode() == DispMode.InRowVertical.ordinal())
                end = true;
            printHeaderCell(out, getDayName(day) + "\r\n" + getSlotName(slot), vertical, eol, eol);
        }
    }
} else if (iForm.getDispMode() == DispMode.InRowVertical.ordinal()) {
    for (Integer day : days) {
        for (Integer slot : slots) {
            boolean end = (slot == slots.last());
            boolean eol = (end && day == days.last());
            if (iForm.getDispMode() == DispMode.InRowVertical.ordinal())
                end = true;
            printHeaderCell(out, getDayName(day) + "\r\n" + getSlotName(slot), vertical, eol, eol);
        }
    }
}
```
- Bottom Status Bar:** 24°C, 5/17/2023, 2:17 AM

This screenshot shows the SonarQube Issues page for the 'unitime' project. The main interface includes a navigation bar with 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', 'Administration', and a search bar. A banner at the top right indicates 'The last analysis has warnings. See details'. The 'Issues' tab is selected, showing a list of 26 issues across 11 days ago. The issues are categorized by file: JavaSource.../timetable/webutil/timegrid/ExamGridTable.java. Most issues are of type 'Bug' and severity 'Major'. The list includes several entries like 'Strings and Boxed types should be compared using "equals()"' and 'Strings and Boxed types should be compared using "equals()"'.

After

This screenshot shows the SonarQube Issues page for the 'unitime-after-resolve' project. The interface is identical to the previous one, but the results are significantly different. The banner still shows 'The last analysis has warnings. See details'. The 'Issues' tab is selected, but the list is empty, displaying '0 issues 0 effort'. A message in the center of the page states 'We couldn't find any results matching selected criteria.' Below this, there is a note: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale; it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.' The footer of the page provides version information: 'Community Edition - Version 10.0 (build 68432) - LGPLv3 - Community - Documentation - Plugins - Web API'.

The screenshot shows the SonarQube interface for the project 'unitime-after-resolve'. The 'Code' tab is selected. A banner at the top right indicates 'The last analysis has warnings. See details'.

Project Settings and Project Information buttons are visible on the right. The code editor displays the file 'ExamGridTable.java' with the following statistics:

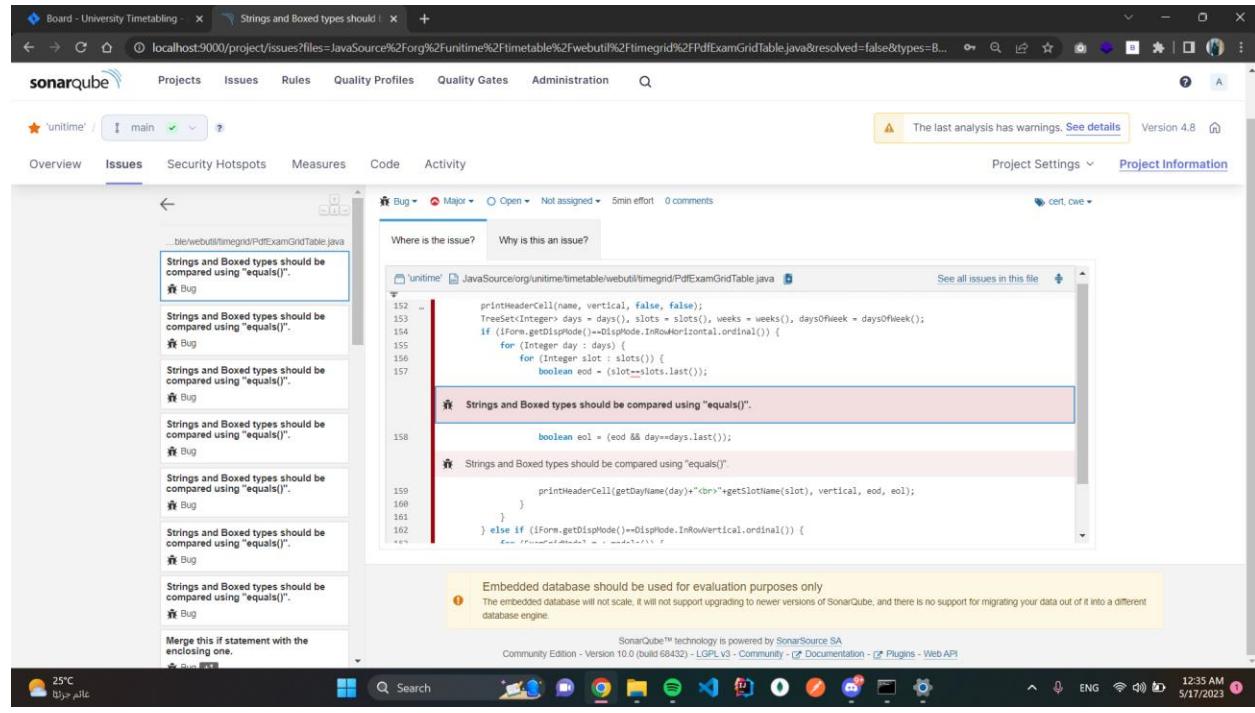
	Lines	Coverage	Duplications	Bug	Vulnerability	Code Smell	Security Hotspot
	1.2k	0.0%	4.8%	0	0	155	0

The code itself is a Java class with an Apache License 2.0 header and imports from java.util.*.

```
1  /*
2   * Licensed to The Apache Foundation under one or more contributor License
3   * agreements. See the NOTICE file distributed with this work for
4   * additional information regarding copyright ownership.
5   *
6   * The Apache Foundation Licenses this file to you under the Apache License,
7   * Version 2.0 (the "License"); you may not use this file except in
8   * compliance with the License. You may obtain a copy of the License at
9   *
10  * http://www.apache.org/licenses/LICENSE-2.0
11  *
12  * Unless required by applicable Law or agreed to in writing, software
13  * distributed under the License is distributed on an "AS IS" BASIS,
14  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
15  *
16  * See the License for the specific language governing permissions and
17  * limitations under the License.
18  */
19 package org.unitime.timetable.webutil.timegrid;
20
21 import java.io.PrintWriter;
22 import java.util.Calendar;
23 import java.util.Collection;
24 import java.util.Collections;
25 import java.util.Date;
26 import java.util.Enumeration;
27 import java.util.Hashable;
28 import java.util.Iterator;
```

File 2(JavaSource/org/unitime/timetable/webutil/timegrid/PdfExamGridTable.java):

Before



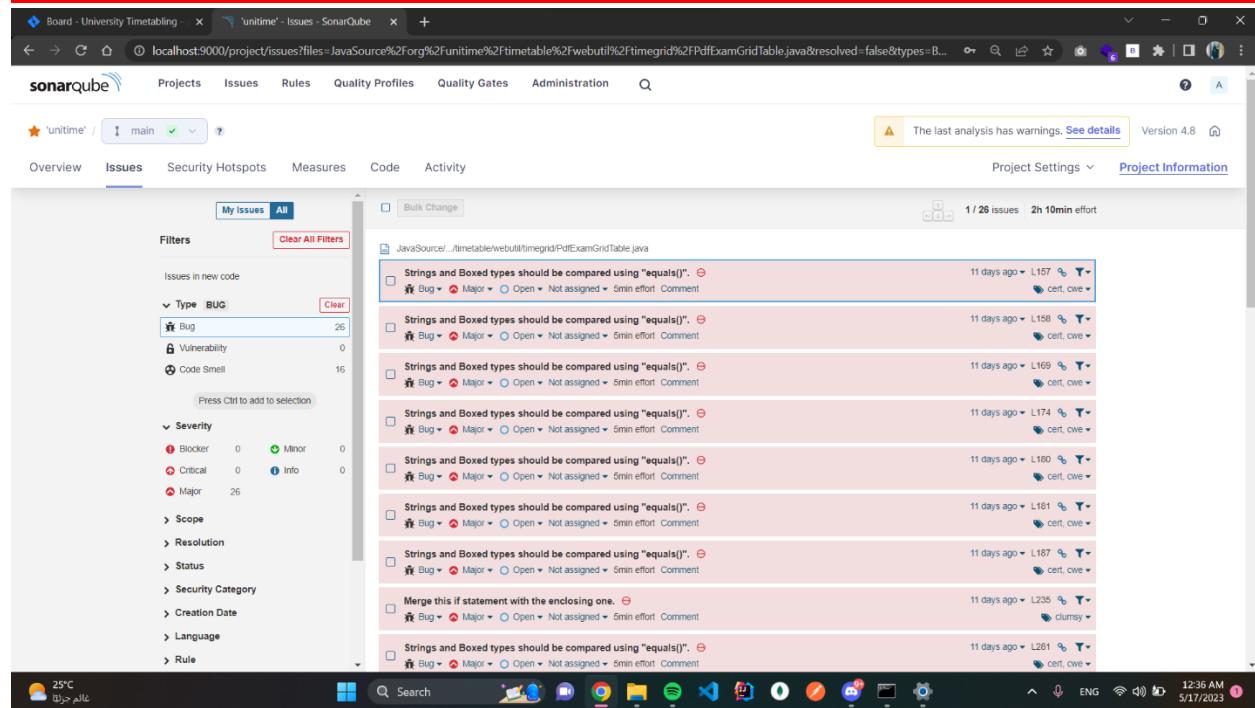
The screenshot shows the SonarQube interface for the file `JavaSource/org/unitime/timetable/webutil/timegrid/PdfExamGridTable.java`. On the left, a sidebar lists several code review comments, each with a red star icon indicating a bug. The main panel displays the Java code with these comments overlaid. One comment highlights a specific line of code:

```
152     printHeaderCell(name, vertical, false);
153     TreeSet<Integer> days = days();
154     slots = slots();
155     weeks = weeks();
156     daysOfWeek = daysOfWeek();
157     if (ifForm.getDispMode() == DispMode.InRowHorizontal.ordinal()) {
158         for (Integer day : days) {
159             for (Integer slot : slots) {
160                 boolean eod = (slot == slots.last());
161             }
162         }
163     } else if (ifForm.getDispMode() == DispMode.InRowVertical.ordinal()) {
164         for (Integer week : weeks) {
165             for (Integer day : days) {
166                 for (Integer slot : slots) {
167                     boolean eod = (slot == slots.last());
168                 }
169             }
170         }
171     }
```

A callout box points to the line `boolean eod = (slot == slots.last());` with the message: "Strings and Boxed types should be compared using 'equals()'".

At the bottom of the code panel, there is a warning about the embedded database:

Embeddable database should be used for evaluation purposes only
The embedded database will not scale. It will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.



The second screenshot shows the SonarQube Issues page for the same file. The sidebar on the left shows a summary of issues: 11 days ago, L157, 26 bugs, 0 vulnerabilities, and 16 code smells. The main panel lists 11 issues, all of which are bugs, each with a red star icon. The first few issues are identical to those shown in the code review view, pointing to the same line of code.

SonarQube Issues page for 'unitime' project. The main issue is titled "Merge this if statement with the enclosing one." It has several sub-comments pointing to specific lines of code in `PdfExamGridTable.java`. One comment is highlighted with a red box around line 234.

```

229     c.setBackgroundColor(getColor(bgColor));
230     c.setBorderWidthBottom(idx==maxIdx ? 0 : 1);
231     addText(c, " ");
232   } else {
233     String bgColor = cell.getBackground();
234   } if (!form.getBackgroundColor()=>backgroundColor.None.ordinal() && !s8gColorNotAvailable.equals(bgColor)) {
235     if (!model.isAvailable(period))
236       bgColor = s8gColorNotAvailableButAssigned;
237     if (bgColor!=null)
238       c.setBackgroundColor(getColor(bgColor));
239     addText(c, cell.getName());
240     if (!form.getResource()>Resource.Room.ordinal())
241       addText(c, cell.getRoomName());
242

```

A tooltip message at the bottom right of the code editor area states: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine."

After

SonarQube Issues page for 'unitime-after-resolve' project. The Java file `PdfExamGridTable.java` now includes an Apache License header at the top. The review comments from the previous screenshot are no longer visible.

```

1 /*
2  * Licensed to the Apache Foundation under one or more contributor license
3  * agreements. See the NOTICE file distributed with this work for
4  * additional information regarding copyright ownership.
5  *
6  * The Apache Foundation licenses this file to you under the Apache License,
7  * Version 2.0 (the "License"); you may not use this file except in
8  * compliance with the License. You may obtain a copy of the License at
9  *
10 * http://www.apache.org/licenses/LICENSE-2.0
11 *
12 * Unless required by applicable law or agreed to in writing, software
13 * distributed under the License is distributed on an "AS IS" BASIS,
14 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
15 *
16 * See the License for the specific language governing permissions and
17 * limitations under the License.
18 */
19 package org.unitime.timetable.webutil.timegrid;
20
21 import java.awt.Color;
22 import java.io.OutputStream;
23 import java.util.StringTokenizer;
24 import java.util.TreeSet;
25
26 import org.unitime.timetable.form.ExamGridForm;
27 import org.unitime.timetable.model.ExamPeriod;
28 import org.unitime.timetable.model.PreferenceLevel;

```

The screenshot shows the SonarQube interface for the project 'unitime-after-resolve'. The 'Issues' tab is selected. In the left sidebar under 'Filters', the 'Type' dropdown is set to 'BUG'. The main content area displays a message: 'We couldn't find any results matching selected criteria.' Below this message is a note: 'Try to change filters to get some results.' At the bottom of the page, there is a warning message: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.' The status bar at the bottom right shows the date and time as '5/17/2023 5:04 PM'.