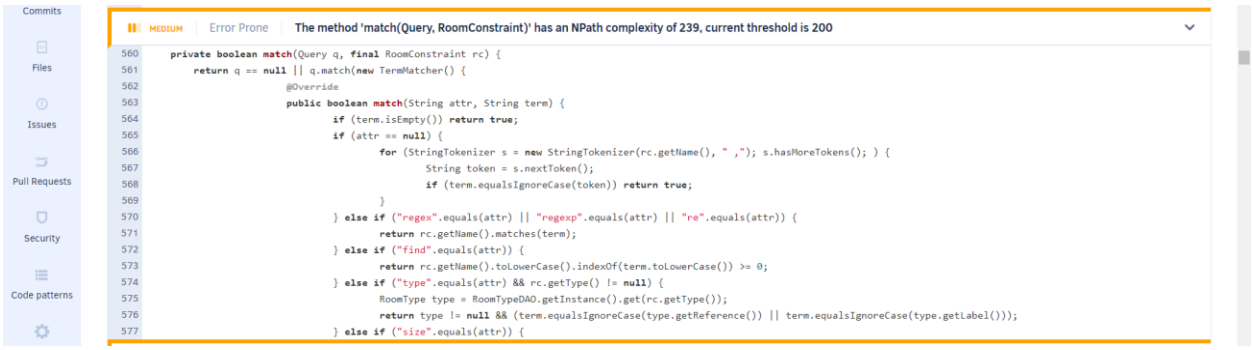
1- JavaSource/org/unitime/timetable/solver/TimetableSolver.java

A- The method 'match(Query, RoomConstraint)' has an NPath complexity of 239, current threshold is 200

ISSUELINK:
https://app.codacy.com/gh/MennaHG/unitime/file/88692826674/issues/source?bid=35165038&fileBranchId=35165038#l560

With refactoring (Replacing Nested Conditionals with guard clauses + decomposing conditions into separate methods) NPathComplexity can be minimized which in turn increases maintainability, readability.



B- The method 'assign (Collection)' has an NPath complexity of 271, current threshold is 200.

ISSUELINK:

https://app.codacy.com/gh/MennaHG/unitime/file/88692826674/issues/source?bid35165038&fileBranchId=35165038#l649

Refactoring can also be done here to reduce NPathComplexity by decreasing the number of acyclic execution paths method can go through and lessen the complexity of statements in the same block like sorting out the nested if else conditions

```
648  @Override
649  public void assign(Collection hints) {
650      Lock lock = currentSolution().getLock().writeLock();
651          lock.lock();
652          try {
653              Hashtable initialAssignments = new Hashtable();
654              for (Placement placement: currentSolution().getAssignment().assignedValues()) {
655                  initialAssignments.put(placement.variable(), placement);
656              }
657              AssignmentRecord record = new AssignmentRecord(this);
658              for (Iterator i=hints.iterator();i.hasNext();) {
659                  Hint hint = (Hint)i.next();
660                  Placement p = hint.getPlacement((TimetableModel)currentSolution().getModel());
661                  if (p!=null) {
662                      Placement ini = (Placement)initialAssignments.get(p.variable());
663                      record.add(ini,p);
664                      Progress.getInstance(currentSolution().getModel()).info(p.variable().getName()+": "+(ini==null?"not assigned":ini.getLongName(useAmPm()))+" &rarr; "+p.getLongName(useAmPm()));
665          if (ini!=null) currentSolution().getAssignment().unassign(0, p.variable());
666                  } else if (hint.getDays() == 0) {
667                      Lecture lecture = null;
668                      for (Lecture l: currentSolution().getModel().variables())
669                          if (l.getClassId().equals(hint.getClassId())) {
670                              lecture = l;
671                          }
672                      if (lecture != null && !lecture.isCommitted())
673                          currentSolution().getAssignment().unassign(0, lecture);
674                  }
675              }
676          for (Iterator i=hints.iterator();i.hasNext();) {
677              Hint hint = (Hint)i.next();
678              Placement p = hint.getPlacement((TimetableModel)currentSolution().getModel());
679              if (p!=null) currentSolution().getAssignment().assign(0,p);
680          }
681              for (Lecture lec: currentSolution().getModel().unassignedVariables(currentSolution().getAssignment())) {
682                  Placement p = (Placement)initialAssignments.get(lec);
683                  if (p!=null) {
684                      record.add(p,null);
685                      Progress.getInstance(currentSolution().getModel()).info(p.variable().getName()+": "+p.getLongName(useAmPm())+" &rarr; not assigned");
686                  }
```

C- Avoid using a branching statement as the last in a loop. Method 'match(Query, String)'

ISSUELINK:
https://app.codacy.com/gh/MennaHG/unitime/file/88692826674/issues/source?bid=35165038&fileBranchId=35165038#l543

The outer loop at line 537 is not necessary since it will only iterate once and reach the return statement at line 543 unless conditions are added in between.

```
543                     return false;
544                 }
545                 return true;
546             } else if ("regex".equals(attr) || "regexp".equals(attr) || "re".equals(attr)) {
547                 return name.matches(term);
548             } else if ("find".equals(attr)) {
549                 return name.toLowerCase().indexOf(term.toLowerCase()) >= 0;
550             }
551             return false;
552         }
553     });
554 }
555
556 private static enum Size {
557         eq, lt, gt, le, ge
558     };
```

```
530
531     private boolean match(Query q, final String name) {
532         return q == null || q.match(new TermMatcher() {
533                     @Override
534                     public boolean match(String attr, String term) {
535                         if (term.isEmpty()) return true;
536                         if (attr == null) {
537                             term: for (StringTokenizer s = new StringTokenizer(term, " ,"); s.hasMoreTokens(); ) {
538                                 String termToken = s.nextToken();
539                                 for (StringTokenizer t = new StringTokenizer(name, " ,"); t.hasMoreTokens(); ) {
540                                     String token = t.nextToken();
541                                     if (token.toLowerCase().startsWith(termToken.toLowerCase())) continue term;
542                                 }
543                                 return false;
544                             }
545                             return true;
546                         } else if ("regex".equals(attr) || "regexp".equals(attr) || "re".equals(attr)) {
547                             return name.matches(term);
548                         } else if ("find".equals(attr)) {
549                             return name.toLowerCase().indexOf(term.toLowerCase()) >= 0;
550                         }
551                         return false;
552                     }
553             });
554         }
```

2- JavaSource/org/unitime/timetable/model/Assignment.java

Avoid using unnecessary if else statements when returning Booleans.

Method: isInConflict(Assignment , Assignment boolean)

ISSUELINK:
https://app.codacy.com/gh/MennaHG/unitime/file/88692831302/issues/source?bid=35165038&fileBranchId=35165038#l333

MEDIUM Error Prone **Avoid** unnecessary if..then..else statements when returning booleans

```
333             if (distance <= a1.getSolution().getProperties().getPropertyDouble("Student.DistanceLimit75min",100.0) && (
334                 (t1.getLength()==18 && s1+t1.getLength()==s2) ||
335                 (t2.getLength()==18 && s2+t2.getLength()==s1)))
336                 return false;
337             return true;
338 }
339
340     public int getMinutesPerMeeting() {
341             TimePattern pattern = getTimePattern();
342             if (pattern.isExactTime()) {
343                     DurationModel dm = getClazz().getSchedulingSubpart().getInstrOfferingConfig().getDurationModel();
```

Line 332:

if (distance <= a1.getSolution().getProperties().getPropertyDouble("Student.DistanceLimit75min",100.0) && ((t1.getLength()==18 && s1+t1.getLength()==s2) ||

(t2.getLength()==18 && s2+t2.getLength()==s1)))

return false;

can simply be replaced with

boolean flag = (distance <= a1.getSolution().getProperties().getPropertyDouble("Student.DistanceLimit75min",100.0);

flag = flag && (t1.getLength()==18 && s1+t1.getLength()==s2) || (t2.getLength()==18 && s2+t2.getLength()==s1);

return flag;

As simple it is it improves Readability, Maintainability of method

3- JavaSource/org/unitime/timetable/gwt/server/UniTimePrincipal.java

ISSUELINK:

https://app.codacy.com/gh/MennaHG/unitime/file/88692827340/issues/source?bid=35165038&fileBranchId=35165038#l43

Avoid throwing null pointer exceptions.

```
42  public UniTimePrincipal(String externalId, String studentExternalId, String name) {
MEDIUM  Error Prone  Avoid throwing null pointer exceptions.
43      if (externalId == null) throw new NullPointerException();
44      iExternalId = externalId;
45      iStudentExternalId = studentExternalId;
46      iName = name;
47
```

Line 43. if (externalId == null) throw new NullPointerException();

Throwing NPEs is not a recommended practice because it would be hard to tell code-thrown NPEs from ones thrown from the VM.

Alternatives:

1- Throw IlegalArgumentException

2- use requrieNonNull --> externalId.requireNonNull(externalId, "Object externalID is Null");

4- JavaSource/org/unitime/timetable/util/MemoryCounter.java

You should not modify visibility of constructors, methods or fields using setAccessible()

ISSUELINK:

https://app.codacy.com/gh/MennaHG/unitime/file/88692831786/issues/source?bid=35165038&fileBranchId=35165038#l164

setAccessible(true) makes private fields accessible from outside the class which poses a risk to data protection and ruins the purpose of encapsulation. Instead, Setters should be used.

```
Security
Code patterns
Settings
163  Field f = uc.getDeclaredField("theUnsafe");
MEDIUM  Error Prone  You should not modify visibility of constructors, methods or fields using setAccessible()
164  f.setAccessible(true);
165  theUnsafe = f.get(uc);
166  } catch (Exception e) { exception = e; }
167  UNSAFE = (sun.misc.Unsafe) theUnsafe;
MINOR  Code Style  Line is longer than 80 characters (found 109).
```

5- WebContent/loginRequired.jsp

Using unsanitized JSP expression can lead to Cross Site Scripting (XSS) attacks.

ISSUELINK:
https://app.codacy.com/gh/MennaHG/unitime/file/88692829714/issues/source?bid=35165038&fileBranchId=35165038#l30

unsanitized jsp expressions lead to Cross Site Scripting which makes website vulnerable to attacks that inject malicious HTML/JS or steal data as the expression would always be

interpreted directly by the browser. Thus, EL expressions should not be directly used and used with c: out as an alternative or escape it using fn.