UniTime

UniTime is a comprehensive university timetable scheduling system. UniTime provides a web-based interface for students, faculty, and staff to view and manage course schedules, exams, and other academic events. The system uses advanced algorithms and techniques to optimize course schedules and minimize conflicts between student schedules.

Components

UniTime has 5 major components, which are:

1. **Course timetabling**

    > This component allows organizations to create a schedule for their courses, taking into account various constraints such as room availability, instructor availability, and time conflicts. The system also allows for automatic scheduling and optimization based on user-defined preferences.

    An example of this component is a **code chunk** that can be found in the *JavaSource\org\unitime\timetable\reports\exam\ScheduleByCourseReport.java* the main function of the ScheduleByCourseReport class, which generates the PDF report by printing the schedule of the courses.

    ```
    public void printReport() throws DocumentException {
        sLog.debug(MSG.statusSortingSections());
        // omitted code
    }
    ```

2. **Examination timetabling**

    > This component helps organize and schedule exams for courses, taking into account various factors such as room availability, student schedules, and conflicts between different exams.

    An example of this component is a **code chunk** that can be found in the *JavaSource\org\unitime\timetable\model\ExamConflict.java* where a function resides that displays exams that conflict with each other.

    ```
    public boolean isDirectConflict() {
        return sConflictTypeDirect==getConflictType();
    }
    ```

3. **Student scheduling**

    > This component allows students to create and manage their own schedules for their courses, taking into account various constraints such as course availability, time conflicts, and prerequisites.

An example of this component is a **code chunk** that can be found in *JavaSource\org\unitime\timetable\model\StudentClassEnrollment.java* where all availabe enrollments for a student are queried from the database.

```
  public static Iterator findAllForStudent(Long studentId) {
     return new StudentClassEnrollmentDAO().getSession().createQuery(
             "select e from StudentClassEnrollment e where "+
             "e.student.uniqueId=:studentId").
             setLong("studentId",
studentId.longValue()).setCacheable(true).list().iterator();
 }
```

4. **Event manager**

> This component allows organizations to manage and schedule various events such as meetings, seminars, and conferences. The system can help with room scheduling, participant registration, and other related tasks.

An example of this component is a **code chunk** that can be found in *JavaSource\org\unitime\timetable\model\Event.java* where we can deduce what type of events the system supports to organize from this enumeration string.

```
public static final String[] sEventTypes = new String[] {
     "Class Event",
     "Final Examination Event",
     "Midterm Examination Event",
     "Course Related Event",
     "Special Event",
     "Not Available Event"
 };
```

5. **Instructor's assignments load handler:**

> This component helps administrators assign instructors to courses based on their qualifications and availability. It also provides tools to manage their workload and ensure equitable distribution of teaching assignments.

An example of this component is a **code Beacon** (function definition) that can be found in *target\src\org\unitime\timetable\form\ClassInstructorAssignmentForm.java* that allows us to deduce an instructor can add an assignment to classes from a GUI form.

```
public void addToClasses(Class_ cls, Boolean isReadOnly, String indent,
String nameFormat)
```

Features

Features it provides include but are not limited to:

- Advanced algorithmic scheduling: UniTime uses advanced algorithms to optimize course schedules and minimize conflicts between student schedules.

- Student scheduling: The system enables students to view and manipulate their course schedules, and also offers features for student sectioning.

- Faculty/instructor scheduling: UniTime provides tools for scheduling faculty and instructors, assigning them to courses and ensuring availability for teaching.

- Resource & room scheduling: The system includes features for scheduling resources such as classrooms, lecture halls, and equipment, ensuring availability and minimizing conflicts.

- Exam scheduling: UniTime provides tools for scheduling exams, assigning them to exam rooms, and ensuring there are no conflicts between exams.

- Automatic conflict resolution: UniTime's algorithms automatically resolve scheduling conflicts to ensure that classes, exams, and resources do not overlap or conflict with each other.

- Web-based interface: The UniTime system can be accessed via a web-based interface, enabling students, faculty, and staff to access their schedules and perform scheduling tasks from any device with internet access.

Code Comprehension Technique applied:

> ***Opportunistic Approach(Top-down + Bottom-up):***
>
> We used this approach because it deemed the most efficient to allow us to get a good overview of the system in little time comapritively.

- First our team analyzed the project's website which contained a comprehensive description of the project and all its prominent features.
- Then we reviewed the project structure analyzing the various packages looking for ***Beacons*** in the **package names** allowing us to pinpoint what type of source code each package contains
- After that we anaylyzed the ***names of the java source files*** trying to pinpoint the classes that perform the core functionality of the system that is mentioned above.
- Moreover we analyzed various java source files looking for more ***Beacons*** (in the form of function names or known programming idioms) in each class that would help us understand what are the most important functions and what a class is generally made to do.

Done by:

- Karim Amr Hamdy: 20206138
- Mostafa Hesham Allam: 2020126
- Youssef Ahmad Abdulghafar: 20206152
- Fares Karim El Kholy: 20206143