جامعة القاهرة

Cairo University
Faculty of Computers and Artificial Intelligence

Software Maintenance and Evolution: Lab Assignment
Phase - 1

Delivered To: Dr.Lamia Abo Zaid

Prepared By:

Mootaz Medhat Ezzat Abdelwahab	(20206074)
Yousef Essam Aboelyazed Esmail	(20206163)
Abdulrahman Ashraf Elmahdy	(20186012)
Amro Adel Farid	(20206145)

Academic Year: 2022/2023

Second Semester

➢ CONTENTS:

➢ General Knowledge:-

- What The System Does?
- System Name.
- System Domain.
- System Main Features.
- System Type and Licenses.
- Supported Operating Systems.

➢ System Comprehension:-

- Knowledge Kind.
- Comprehension Goal.
- Comprehension Scope.
- Comprehension Approach:

  - Top-Down.
  - Bottom-Up.
  - Opportunistic.

➢ Static Analysis Report

➢ Change Request:-

- Change Request Type.
- Change Request Description.

➢ Links:-

- Unitime.org.
- Apereo Foundation.
- Github Repository.
- Ticketing System (Jira).

➢ General Knowledge

➥ What The System Dose?

UniTime is a comprehensive educational scheduling system that supports developing course and exam timetables, managing changes to these timetables, sharing rooms with other events, and scheduling students to individual classes. It is a distributed system that allows multiple university and departmental schedule managers to coordinate efforts to build and modify a schedule that meets their diverse organizational needs while allowing for minimization of student course conflicts. It can be used alone to create and maintain a school's schedule of classes and/or exams, or interfaced with an existing student information system.

➥ System Name:

- **Comprehensive University Timetabling System (UniTime)**

➥ System Domain:

- **Scheduling.**
- **Education.**

➥ System Main Features:

- **Student Scheduling:**
  - Constructing a demand-based timetables and optimize the number of students who receive the needed courses.
  - Determining the expected need for individual course sections using knowledge based on curricular course requirements or historical course requests and the existing timetable.
- **Event Management:**
  Creating events for all class meetings and examinations in the events calendar for the academic session (i.e., guest speakers, club meetings, study sessions, and other activities that need to be scheduled to a variety of campus spaces).
- **Course Timetabling:**
  Placing each course at a time (or set of times) which does not conflict with the time(s) assigned to any other course required by the students attending it.
- **Course Management:**
  Allowing users to easily search for alternatives that have a minimal impact on the overall timetable, make changes to the class timetable, and communicate these changes to affected students and other systems.
- **Examination Timetabling:**
  Creating a complete (mid-term and final) exam schedule for each term considering minimizing:
  - The number of conflicting exam placements for all students.
  - The number of occurrences of back-to-back exams or students with more than a given number of exams in a day.

➥ **System Type and Licenses:**

- The system was originally developed as a **collaborative effort** by faculty, students, and staff at universities in **North America** and **Europe**.
- The software is distributed free under an **open source license** in hopes that other colleges and universities can benefit their students through better scheduling or wish to contribute to ongoing research in this area.
- The **UniTime** project has become a sponsored project of the **Apereo Foundation** in **March 2015.**

➥ **Supported Operating Systems:**

- **Linux.**
- **Mac.**
- **Windows**

➢ System Comprehension

➥ Knowledge Kind:

- **software-specific knowledge:**
  - **Exception Handling.**
  - **Implementation Details.**
  - **Functional Requirements.**

  During the process of system comprehension, we found that we need to have more **general** knowledge about the system (i.e., what the system does, system main features, system domain , system type and licenses).

➥ Comprehension Goal:

- Collecting more information about the source code itself to help in refactoring and bugs trouble-shooting**.**

➥ Comprehension Scope:

- **partial** understanding of source code**.**

➥ Comprehension Approach:

- The system comprehension process can work in **three** Approaches**:**
  - **Top-Down:**
    In this approach, the programmer first identifies the goals of the program, followed by possible implementations of those goals such that the implementations match against the code.
  - **Bottom-Up:**
    In this approach, the programmer first identifies program plans from source code, makes annotations, and moves up to the top, goal layer.
  - **Opportunistic (that we used in system comprehension process):**
    - In this approach, we have been able to combine the above two approaches to take best advantage of whatever opportunity is available to make best progress in terms of knowledge gain at any given time.
- We started with top-down to gain an overview of the functions of the program. Then selectively applied bottom-up strategies when nearing “code level” to verify hypotheses resulting from top-down reading.

➢ Static Analysis Report

➥ Uploaded On GitHub.

➢ Change Request

➥ Change Request Type:

- **Bug fix.**

➥ Change Request Description:

- **Overrides hascode in AcademicSessionProvider.java:**
  - This class overrides **“**equals( )**”** and should therefore also override “hashCode”
  - /UniTime/JavaSource/org/unitime/gwt/shared/AcademicSessionProvider.java
- **Fix NullPointerException in AbstractReport.java:**
  - A **“**NullPointerException**”** could be thrown; **“**o**”** is nullable here
  - /UniTime/JavaSource/org/unitime/timetable/reports/AbstractReport.java

- **Unitime.org:**

➥ **https://www.unitime.org/**

- **Apereo Foundation:**

➥ **https://www.apereo.org/**

- **Github Repository:**

➥ **https://github.com/amahdyy/maintenance-lab-assignment-uniTime-system.git**

- **Ticketing System (Jira):**

➥ **https://abdulrahmanmahdy.atlassian.net/jira/core/projects/FIRST**