CS 5500 Course Project

Phase II (Assigned: February 3, Due: February 21)

Frank Tip and Mike Weintraub

Architecture of the Application

Starting with this phase and going forward, you will be working in a team of four people. Team composition will be decided and announced by the instructors. Phase 2 of the project concerns the creation of a high-level design and involves:

1. Refinement of the use cases you produced in phase 1.
2. Specification of an initial design using UML.
3. Defining interfaces for key APIs and data structures using Java.

System Components

While the exact functionality of the system is to be decided by the team, the envisioned system must consist of three distinct components:

1. A *front end* responsible for analyzing publication data. Initially, the system will only use data from the DBLP publication site (http://dblp.uni-trier.de/db/). Your system should rely on a local copy of the DBLP data (the TAs can assist you with obtaining this data). The front-end component may use an in-memory representation that is suitable for querying. Alternatively, it may rely on a database (e.g. RDBMS or NoSQL).
2. A *query engine* responsible for executing queries against the representation that the front end creates. These queries should have a textual representation that is appropriate for the problem domain and that appeals to the intuition, i.e., the users of your system should *not* be required to know SQL or some other full-featured query language.
3. A *user interface* responsible for providing end users facilities for entering and executing queries and then displaying the results. An end user should be able to constrain queries or filter results, e.g., limit the queries to a selected timeframe, to a subset of conferences, the authors whose names start with a specific sequence of characters, etc.

Specific Tasks

Phase II of the project consists of the following 4 tasks:

Use Cases, Revisited

Your first task is to meet with your team members and decide on the system's functionality: what types of queries it should support, what the user-interface will look like, and so forth. To accomplish this, you should review each other's use case descriptions and agree on a single set of use cases that you plan to implement as a team. This combined set of updated use cases is a deliverable for Phase II.

Forming Sub-Teams

Your second task is to come up with a high-level design for each of your system's components. A sub-team of two people is responsible for each of the three major components of your systems. Since your entire team consists of four people, each team member will be on two sub-teams. Deciding which team members are on which sub-teams is a deliverable for Phase II.

Initial Design

The third task is to create an initial design for each of the components. Each sub-team is responsible for this deliverable. This design should include UML diagrams that provide a high-level overview of the component's structure and for the key classes and objects being manipulated. These UML diagrams should be accompanied by some informal documentation that relates them to your use cases. The UML diagrams should be submitted in a PDF document of **absolutely no more** than 5 pages (for each component).

Interface Definitions

The fourth task is to define Java interfaces that reflect the key APIs for each component. These interfaces should provide sufficient detail to enable the writing of blackbox test cases - someone unfamiliar with the internals of your component should be able to write some test cases. The Java interfaces should be committed into your team repository.

Implementation Requirements

The implementation of your system must meet the following requirements (per the semantics defined in RFC 2119):

- **Implementation language**: your system SHALL BE implemented in the Java programming language.
- **Development environment**: you MUST use the Eclipse Integrated Development Environment for your development.
- **Testing**: tests for your system MUST be automated using the JUnit testing framework.
- **Version control**. The TAs will provide you with an NEU GitHub repository for your project. Please create an CCIS ID for yourself if you don't already have one.
- **Documentation and issue tracking**: you MUST use Atlassian's team collaboration software (JIRA/Confluence) for documentation and issue tracking.
- **Continuous Integration.** You MUST use Jenkins for continuous integration.
- **Usability**. The course TAs should be able to install your system with minimal effort, and run your tests with a press of the button.

Deliverables for Phase II

There are four deliverables for this phase:

1. Updated use cases (≦5 pages PDF)
2. Sub-team composition
3. UML design for each component (≦5 pages PDF for each component)
4. Interface definitions for the components and major data structures.