In this assignment, you will analyze a different event-based application from the one you analyzed in the previous assignment. You are still asked to determine how to perform **three maintenance tasks** in that event-based (alternatively referred to as “message-based”) application. While performing these tasks, you will be expected to report the **message-based dependencies** either within a component or between components that you have examined in order to determine how to complete the task. Note that you are not required to actually change the application’s code to complete this assignment; you are only required to note **what parts of the application need to be changed** to complete each maintenance task.

You will also need to explain **why** you ended up examining the dependencies you highlight. Given that you need to familiarize yourself sufficiently with the event-based application you are analyzing, you may find that you have also examined dependencies that, in the end, you determined to be irrelevant for the particular maintenance task you are performing. You should report these dependencies as well.

You will be assigned the maintenance tasks and the particular event-based application individually. Your application and/or tasks may be similar to some of your classmates’, but they will not be identical, so it is critical that you work on this assignment on your own. To obtain the maintenance tasks and the application you will be working on, please send an email to the TA with the subject **“CSCI 578 Spring 2013 HW3 – Task Request”**. You can expect a response from the TA within a few hours.

This time around we ask you to first try to use static analysis tools to determine dependencies relevant to conducting these maintenance tasks. In particular, we will provide you with the IBM Wala slicer and a driver that allows you to employ the slicer. IBM Wala is a static analysis framework for Java and related languages. The slicer performs program slicing, which, given a statement \( s \) and a set of variables at that statement, provides the program statements, called a **slice**, that may affect or be affected by those variables at \( s \). The driver will visualize the slice as a graph and output the statements along with their line numbers. Note that the statements will be represented using Wala’s Intermediate Representation (IR). If you find Wala’s slicer to be insufficient, we ask you to then try using the reference finding feature of Eclipse. If all else fails for you, feel free to try other tools or simply revert to whatever methodology worked for you in the previous assignment.

You are also required to keep track of how much **time and effort** you placed into determining how to perform each maintenance task. Below we provide guidelines for tracking this information. You need to record your time and effort accurately. You will not get credit if you try to decide which metric we will use when grading (e.g., minimizing or maximizing the time spent on the task) and then try to optimize your reported results to that metric; you will only get credit for your work if your results are accurate. We will correlate your effort data for homework assignments 2 with the related assignments 3 and 4 before providing a grade pertaining to the effort; this part of the grade will be counted into the last homework assignment.
Structure of Your Report
System Assigned: <Name of system>

For each maintenance task, specify:

1) Task No:
2) Description of Task:
3) Components you examined:
   a) Names of the components you examined
   b) The reason you examined them
4) Message-based dependencies that are relevant to the maintenance task:
   a) For each component you examined, show the following information
      i) For each dependency within a component
         (1) Name of Class which contains the dependency:
            (a) Name and type of Consumed message
               (i) Names of consumed message’s attributes
               (ii) Name of the method or interface in which the message is consumed
            (b) Name and type of Published Message
               (i) Names of published message’s attributes
               (ii) Name of the method or interface in which the message is published
         (2) Why is this dependency relevant?
         (3) How did you find this dependency? Did you use the slicer, Eclipse, or another methodology? Providing example output (e.g. program statements or graph diagrams) from your tools may help.
      ii) For each dependency between components
         (1) Name of the component which published the message(source)
            (i) Name of the message or event
            (ii) Names of published message’s attributes
            (iii) Name of the method of the source component in which the message is published
         (2) Name of the component which consumed the message(sink)
            (i) Name of the message or event
            (ii) Names of consumed message’s attributes
            (iii) Name of the method of the sink component in which the message is consumed
         (3) Why is this dependency relevant?
         (4) How did you find this dependency? Did you use the slicer, Eclipse, or another methodology? Providing example output (e.g. program statements or graph diagrams) from your tools may help.
4) Message-based dependencies that you examined in addition to the relevant dependencies:
   a) Use the same format as in 4)
6) Explanation of how the code would need to be modified in order to complete the maintenance task
   a) Questions that may need to be answered
      i) What message-based dependencies may need to be modified, added, or removed?
      ii) What methods need to be modified, added, or removed?
      iii) What variables or objects may need to be modified, added, or removed?
   b) Please explain why any methods, variables, objects, etc. are relevant to the maintenance task at hand.
7) Description of the time and effort required to perform the task
   a) How much time and effort did it take to learn the tools that you used?
      i) For each tool
         (1) Name of the tool you used
         (2) Time and effort in hours to learn – for accuracy, please keep track of and report fractions of an hour
   b) What made learning the tools you used particularly challenging? For example, were you having a hard time reading Wala’s IR? Was the driver for the slicer hard to modify or understand?
   c) How difficult was it to find the message-based dependencies that are relevant to the task? For example, a component may be relatively small allowing you to quickly determine dependencies simply through code reading.
      (1) More specific issues that you can discuss
         (a) How long did it take to find the message-based dependencies for this task?
         (b) How long did it take to determine if the message-based dependencies that you found were relevant?
         (c) How long did it take to determine the parts of the task that did NOT involve message-based dependencies?
d) How important were the message-based dependencies that are relevant to the task? For example, determining how to modify a particular object in a component may be more challenging than actually determining the appropriate dependency that may need to be altered.

i) Issues to consider regarding this question
   (1) Did the message-based dependency actually inform you exactly which parts of a component need to be modified? In other words, did tracking the dependency actually help you identify which specific statements, methods, variables, or objects needed to be modified?
   (2) Did the task actually require modifying, adding, or removing one or more message-based dependencies?