Exercise

The Internet Archive provides free access to collections of digitalised materials, software applications, etc. The Internet Archive collects this material automatically, using a web crawler. A web crawler is a bot that browses the World Wide Web. A Web crawler starts with a list of URLs to visit, called the seeds. As the crawler visits these URLs, it identifies all the hyperlinks in the pages and adds them to the list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies. If the crawler is performing archiving of websites it copies and saves the information as it goes. The archives are usually stored in such a way they can be viewed, read and navigated as they were on the live web, but are preserved as 'snapshots'.[1]). Heritrix is a web crawler designed for the Internet Archive. The Internet Archive stores the web resources it crawls in an Arc file.

Here we are going to take a look at the class `FetchHTTP`. Draw its corresponding class and sequence diagram.

```
/**
 * HTTP fetcher that uses <a href="http://hc.apache.org/">Apache HttpComponents</a>.
 * @author nlevitt
 * https://github.com/internetarchive/heritrix3/blob/05811705ed996122bea1f4e034c1ed5f7a07
 * modules/src/main/java/org/archive/modules/fetcher/FetchHTTP.java
 */
public class FetchHTTP extends Processor implements Lifecycle
  protected void innerProcess(final CrawlURI curi) throws InterruptedException {
    // Get a reference to the HttpRecorder that is set into this ToeThread.
    final Recorder rec = curi.getRecorder();

    // Shall we get a digest on the content downloaded?
    boolean digestContent = getDigestContent();
    String algorithm = null;
    if (digestContent) {
       algorithm = getDigestAlgorithm();
       rec.getRecordedInput().setDigest(algorithm);
    } else {
       // clear
       rec.getRecordedInput().setDigest((MessageDigest)null);
    }

    FetchHTTPRequest req = new FetchHTTPRequest(this, curi);


    rec.getRecordedInput().setLimits(getMaxLengthBytes(),
```

[1]This is copied from Wikipedia from Internet Archive and [Web Crawler](https://en.wikipedia.org/wiki/Web_crawler

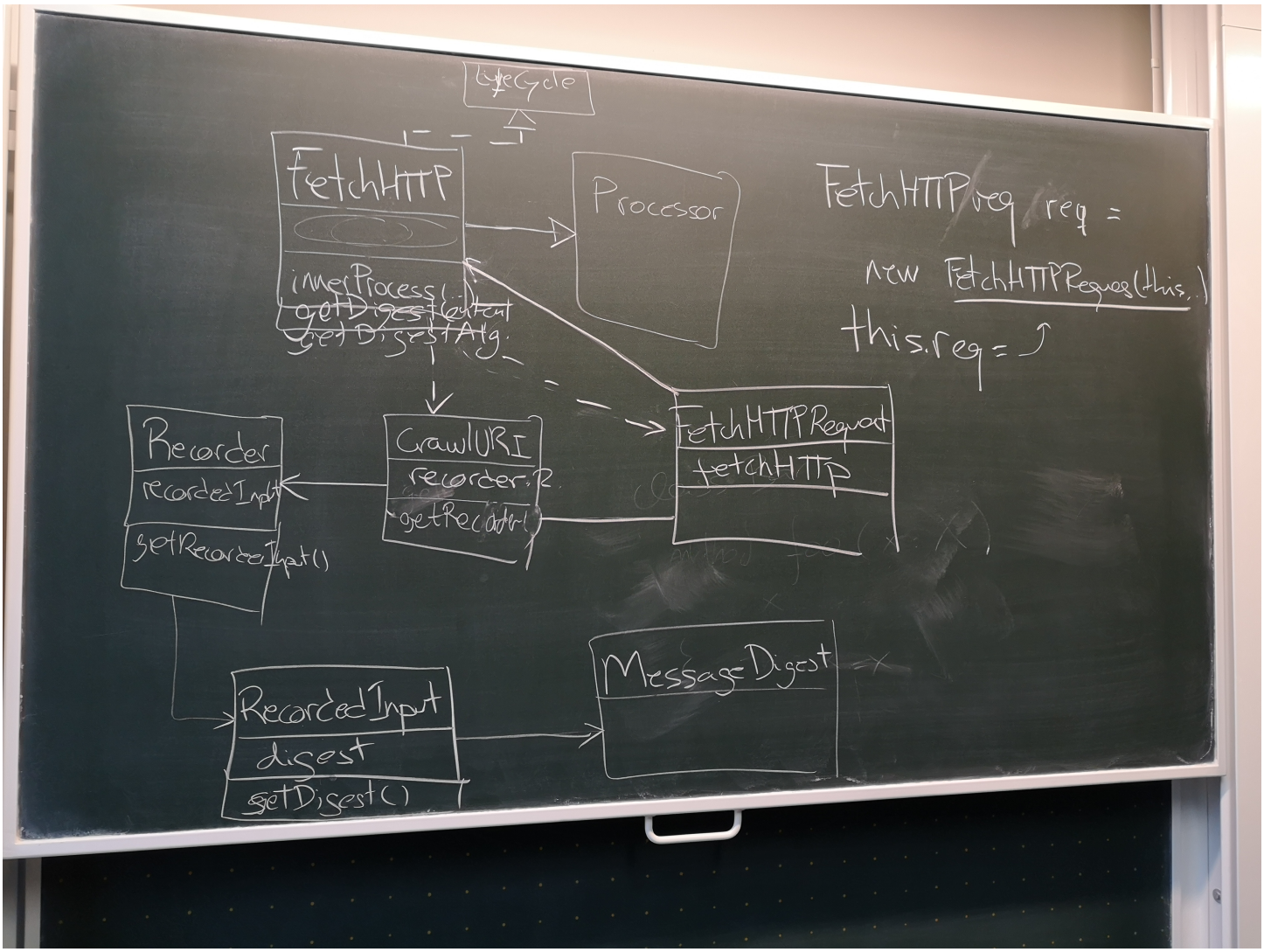
```
                10001 * (long) getTimeoutSeconds(), (long) getMaxFetchKBSec());

    HttpResponse response = null;
    try {
        response = req.execute();
        addResponseContent(response, curi);
    } catch (ClientProtocolException e) {
        failedExecuteCleanup(curi, e);
        return;
    } catch (IOException e) {
      if ("handshake alert:  unrecognized_name".equals(e.getMessage())) {
            req.setDisableSNI(true);
         try {
           response = req.execute();
           addResponseContent(response, curi);
         } catch (ClientProtocolException ee) {
           failedExecuteCleanup(curi, e);
           return;
         } catch (IOException ee) {
           failedExecuteCleanup(curi, e);
           return;
         }
      } else {
        failedExecuteCleanup(curi, e);
        return;
      }
    }
    ...
  }
}
```

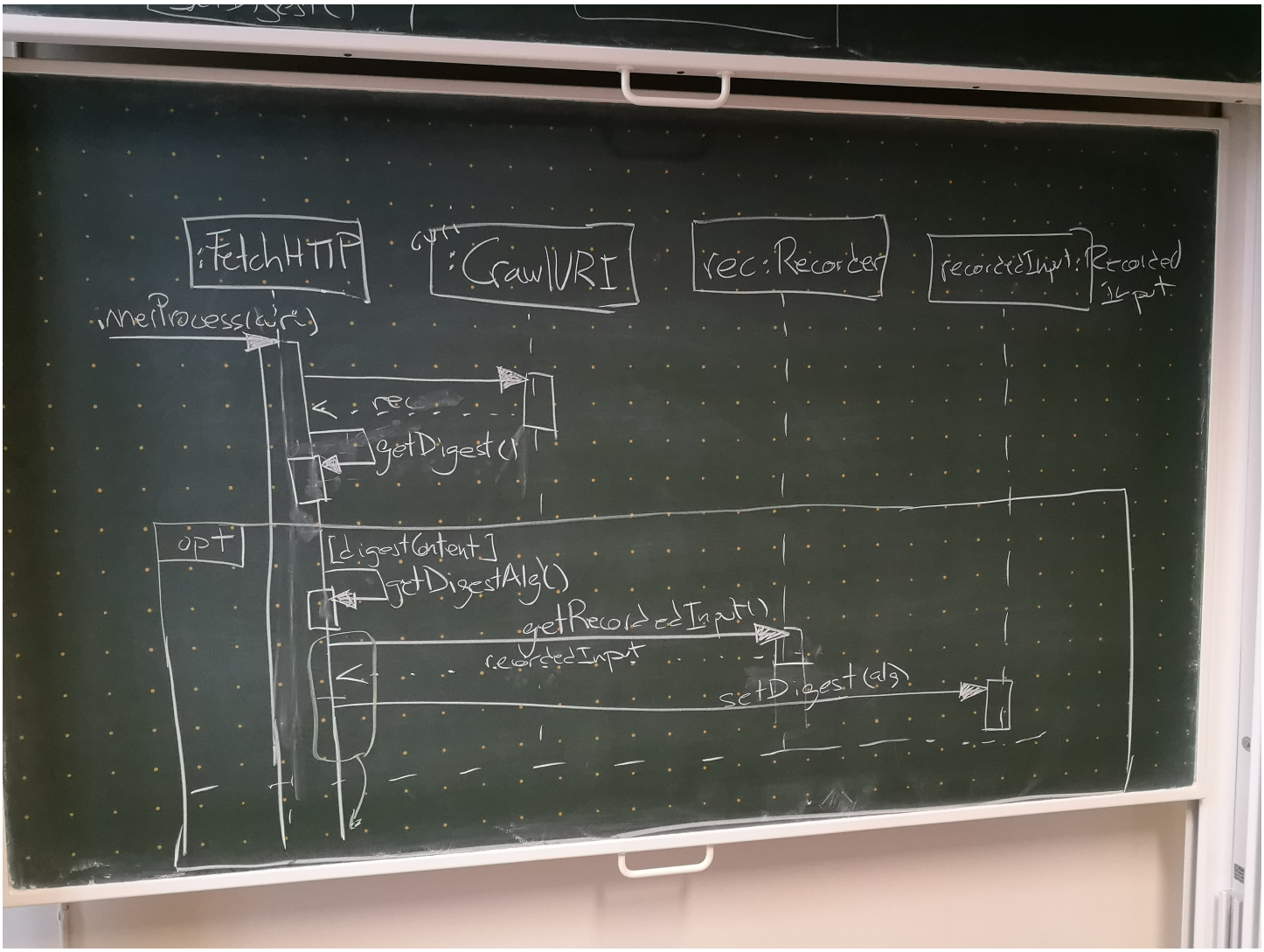
Solution

As part of the solution, we did some assumptions:

1. We may have omitted getters and setters as they occupy too much space
2. We omitted the visibility (public, private, etc). This is because the example shown does not include all the sources and we could not know.
3. We assume that constructors are given arguments and they just set the attributes of the class. This is again because showing all the constructors involves more code for the exercise at hand.

The exercise shows only the relationships that exist in a single method, `innerProcess`. Had we taken the whole source code, our assumptions probably would not hold. Take this into consideration when studying the solution. If there is any mistake, you are more than welcome to report it and I will fix it. If there is anything unclear, please ask me :)


Lifecycle
FetchHTTP
innerProcess()
getDigestOutput
getDigestAlg
Processor
FetchHTTPReq req =
new FetchHTTPRequest(this,)
this.req = ↗
Recorder
recordedInput
setRecordedInput()
CrawlURI
recorder, R
getRecorder()
FetchHTTPRequest
fetchHTTP
RecordedInput
digest
getDigest()
MessageDigest


:FetchHTTP
:CrawlURI
rec:Recorder
recordedInput:RecordedInput
innerProcess(curi)
rec
getDigest()
opt
[digestContent]
getDigestAlg()
getRecordedInput()
recordedInput
setDigest(alg)