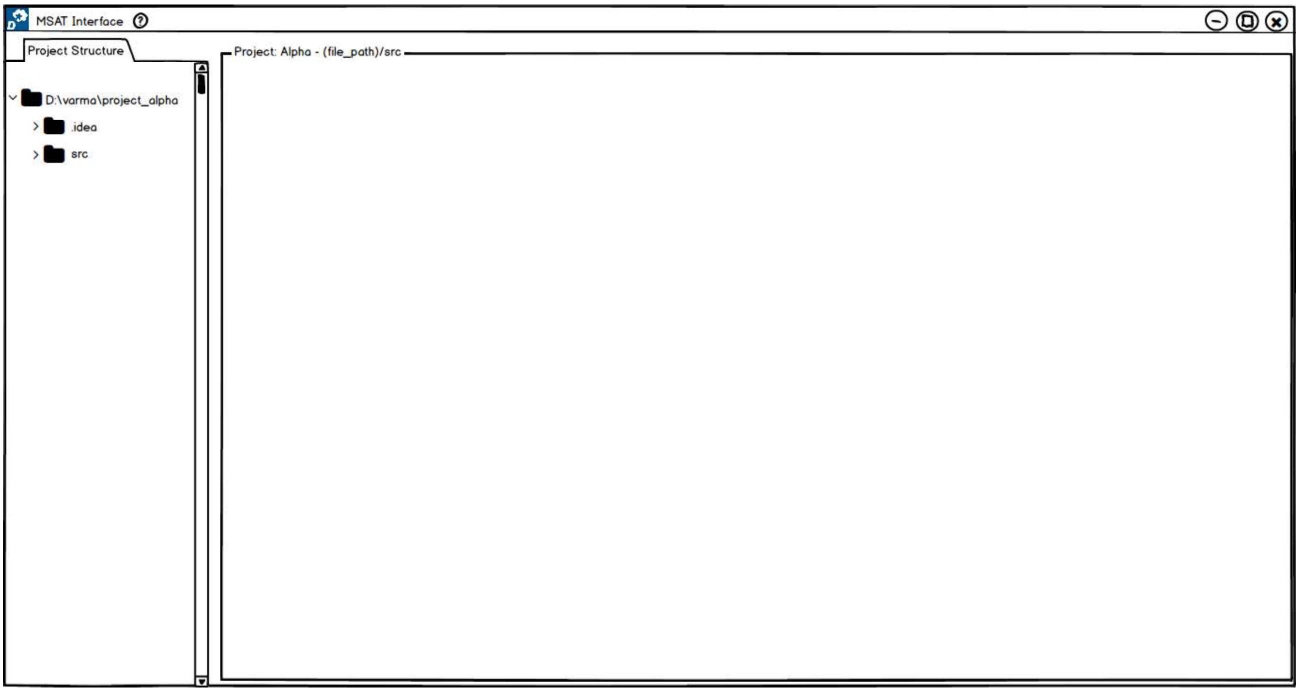
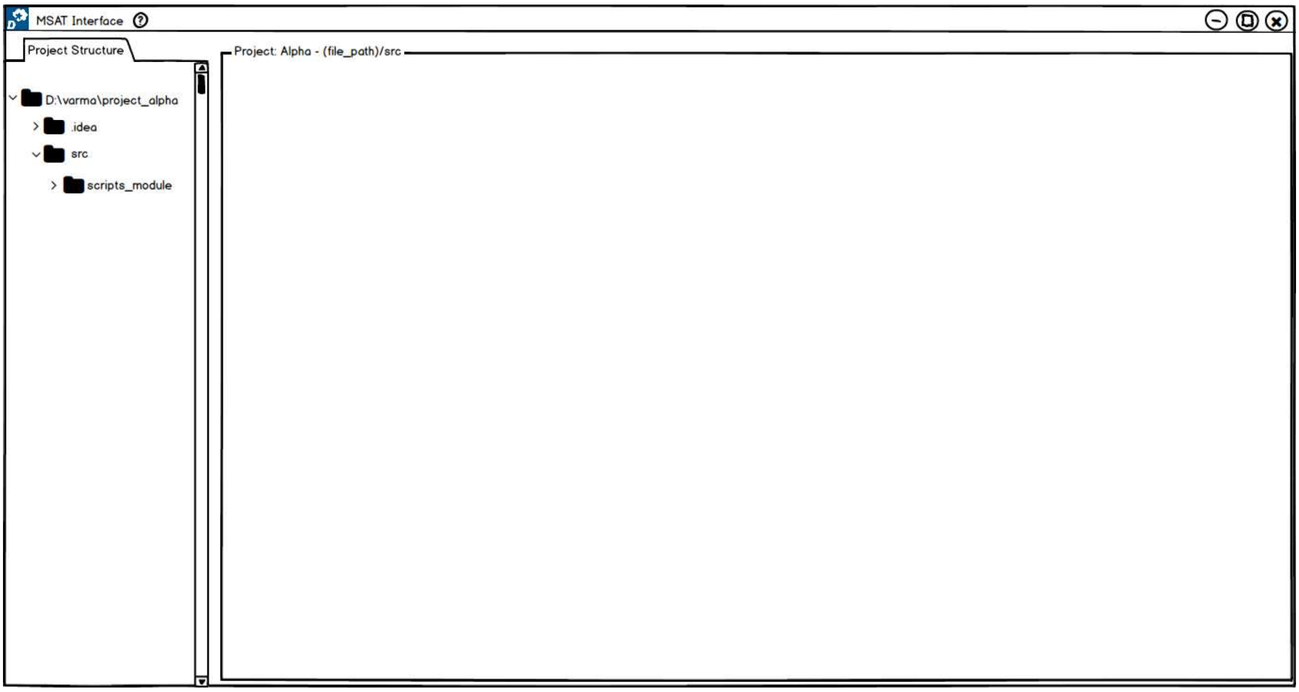
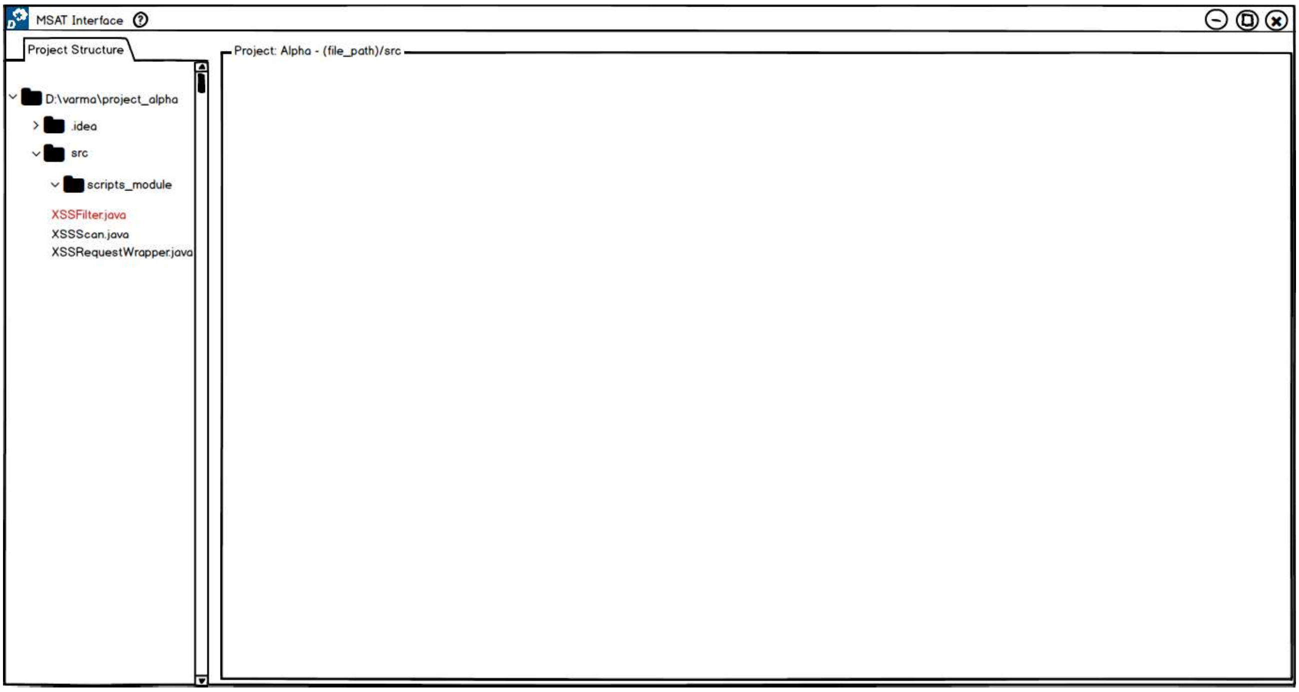
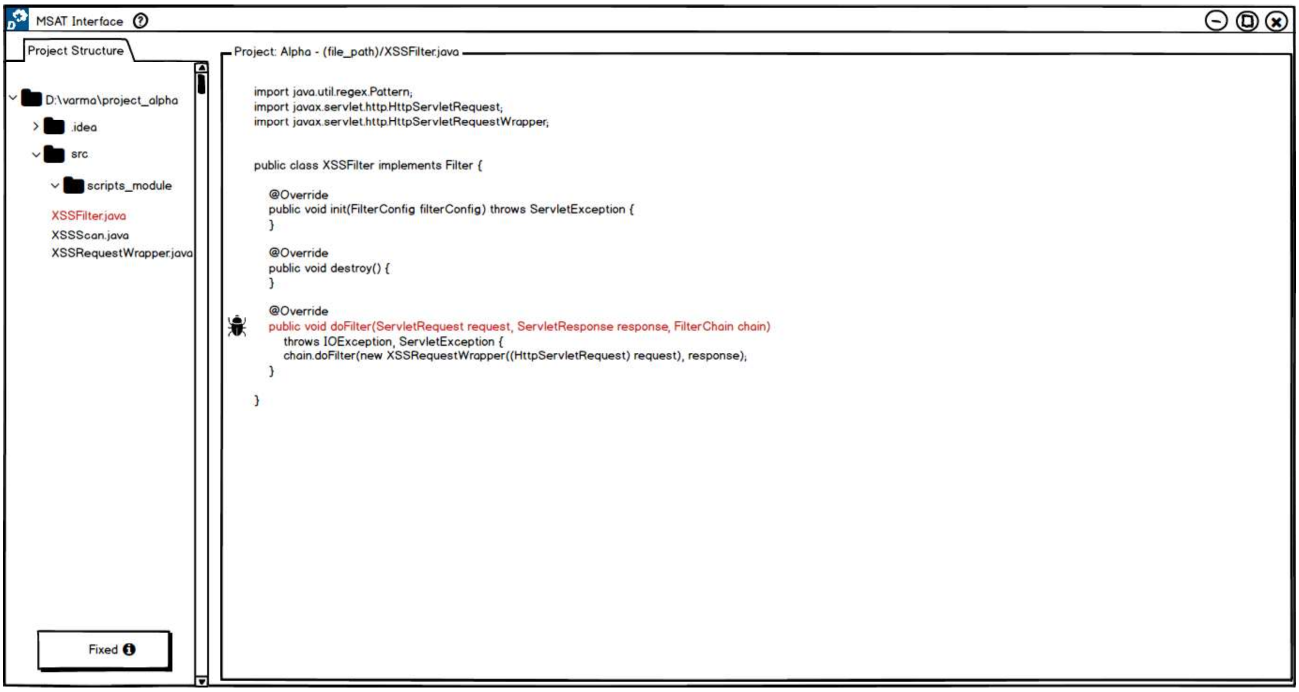
UXD Cycle 3 - RQ 1 - Bug Icon | Next



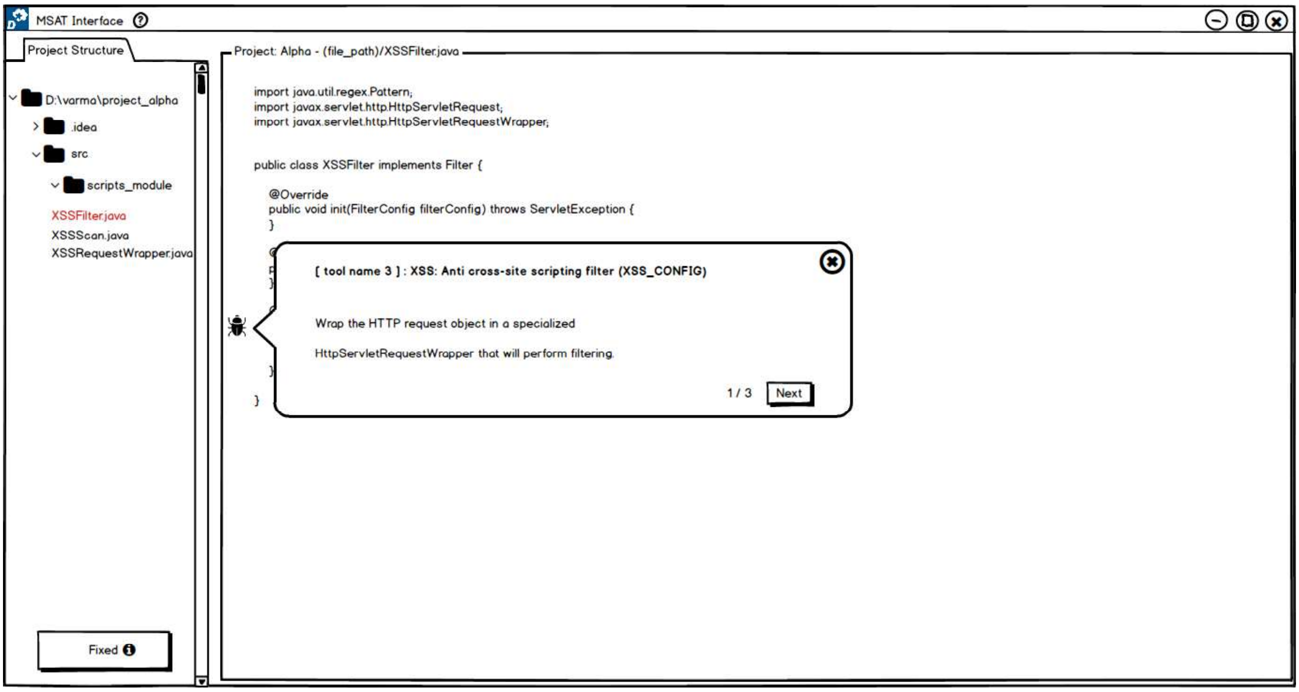






Project: Alpha - (file_path)/XSSFilter.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {
   }

   @Override
   public void destroy() {
   }

   @Override
   public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
      throws IOException, ServletException {
      chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
   }

}
```



Project: Alpha - (file_path)/XSSFilter.java

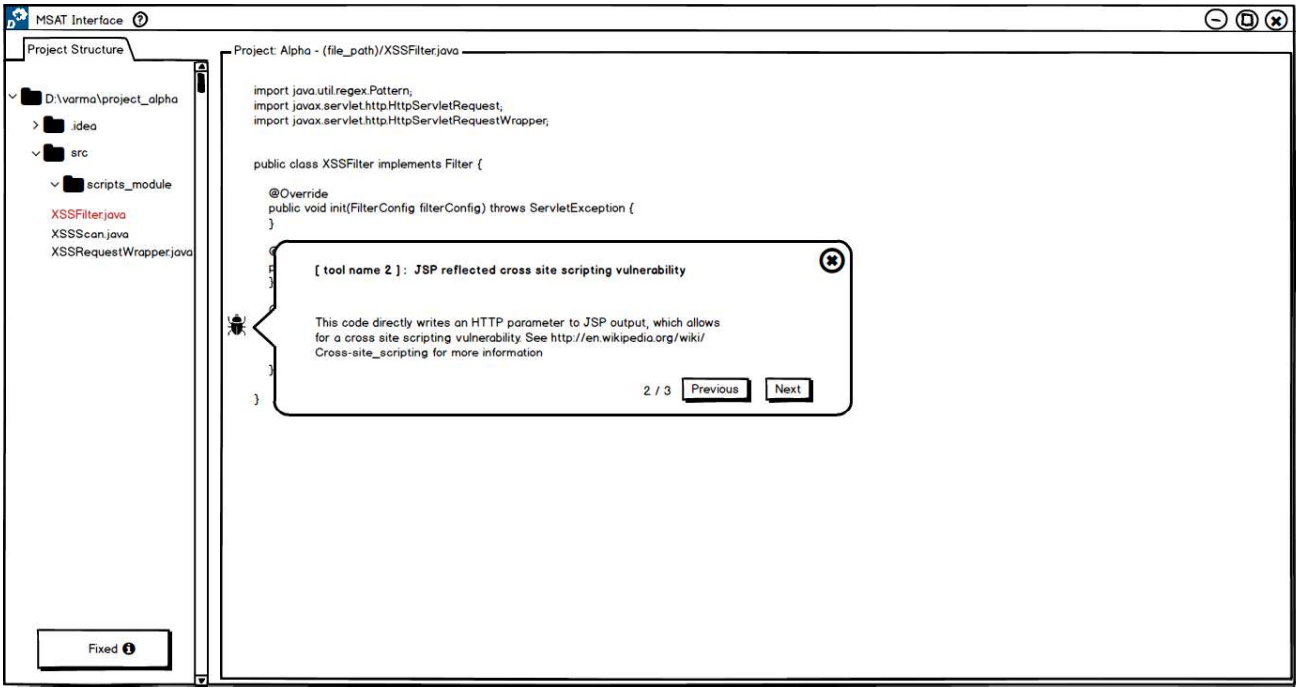
```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {
   }
```

[tool name 3] : XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized

HttpServletRequestWrapper that will perform filtering.

1 / 3



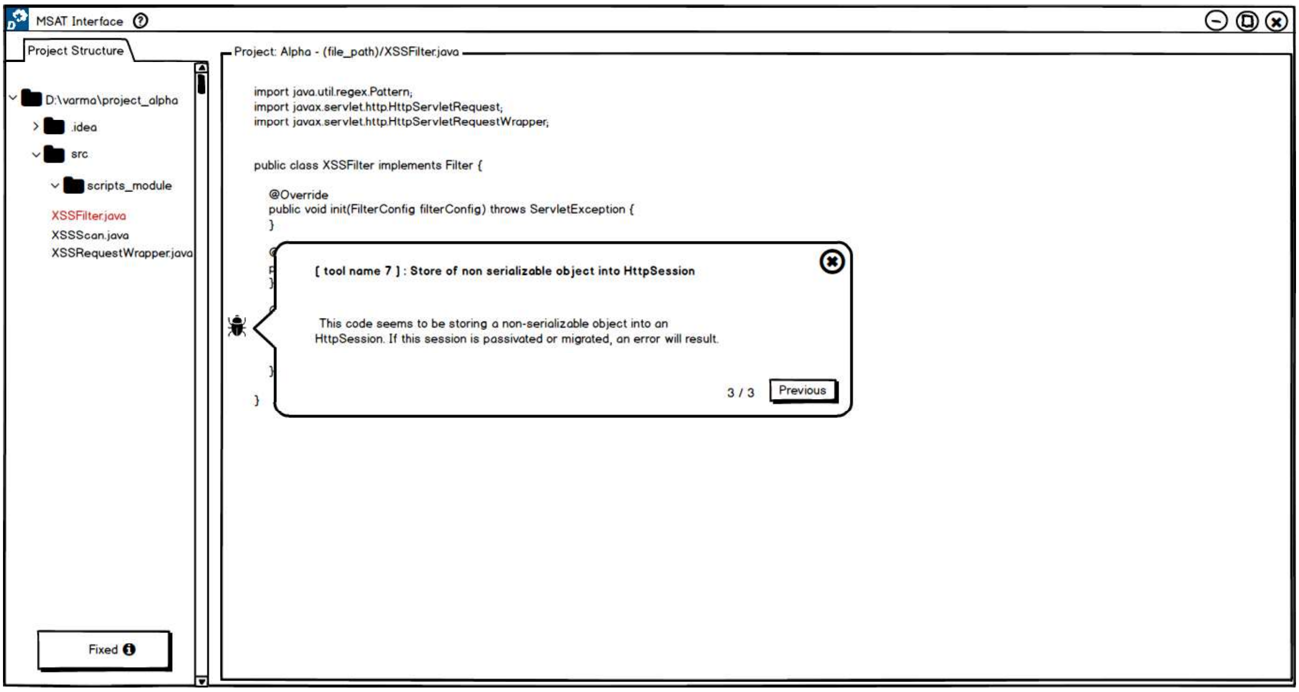
Project: Alpha - (file_path)/XSSFilter.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {
   }
```

[tool name 2] : JSP reflected cross site scripting vulnerability

This code directly writes an HTTP parameter to JSP output, which allows for a cross site scripting vulnerability. See http://en.wikipedia.org/wiki/Cross-site_scripting for more information

2 / 3

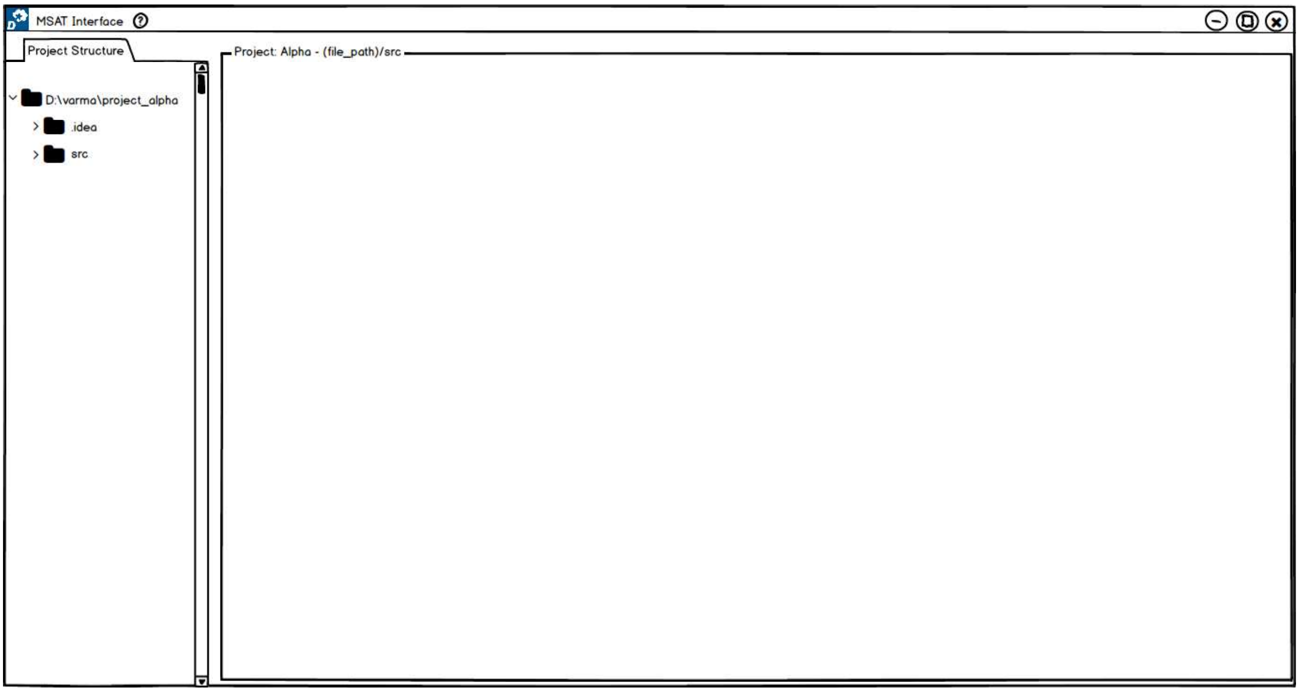


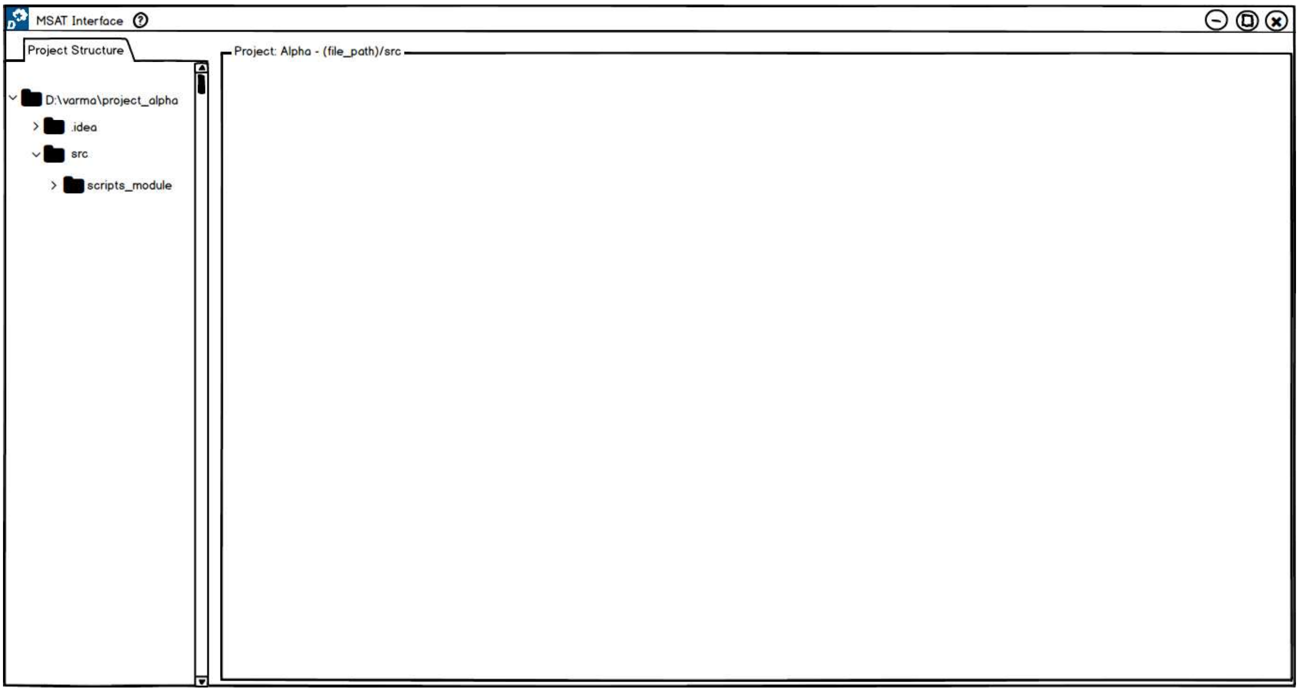
```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {
   }

   @
   p
   }

   @



   }
}
```

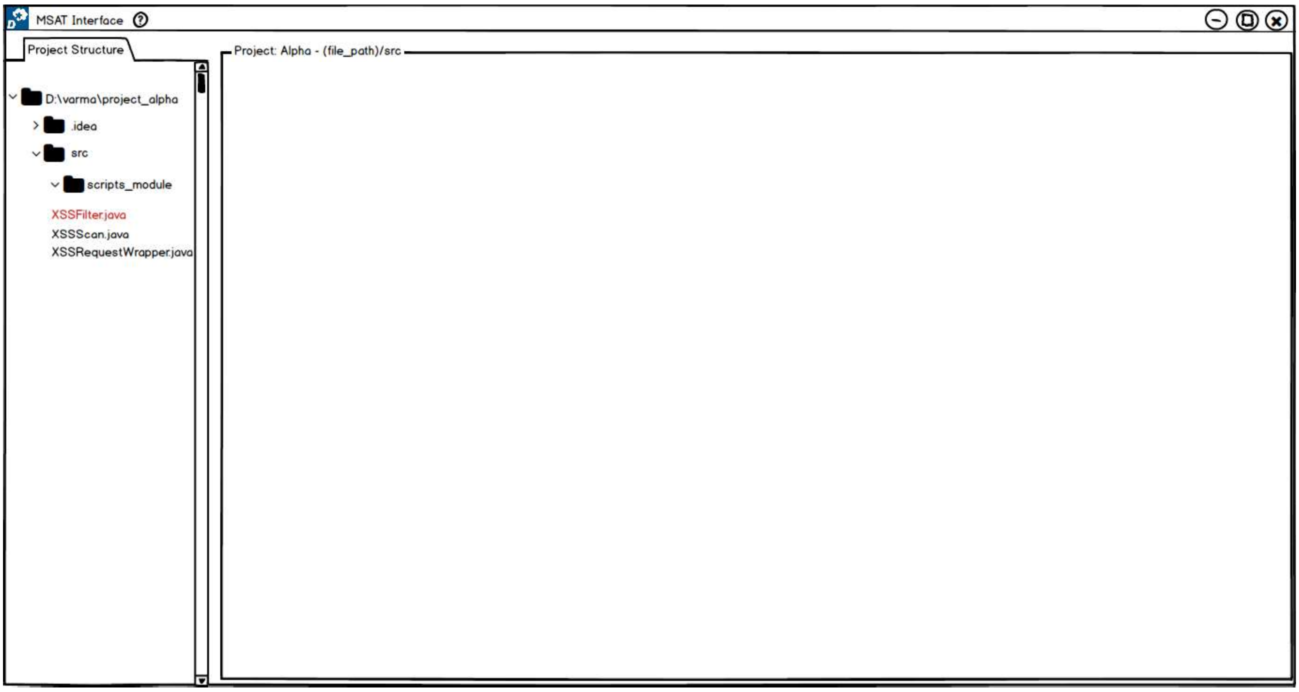
[tool name 7] : Store of non serializable object into HttpSession

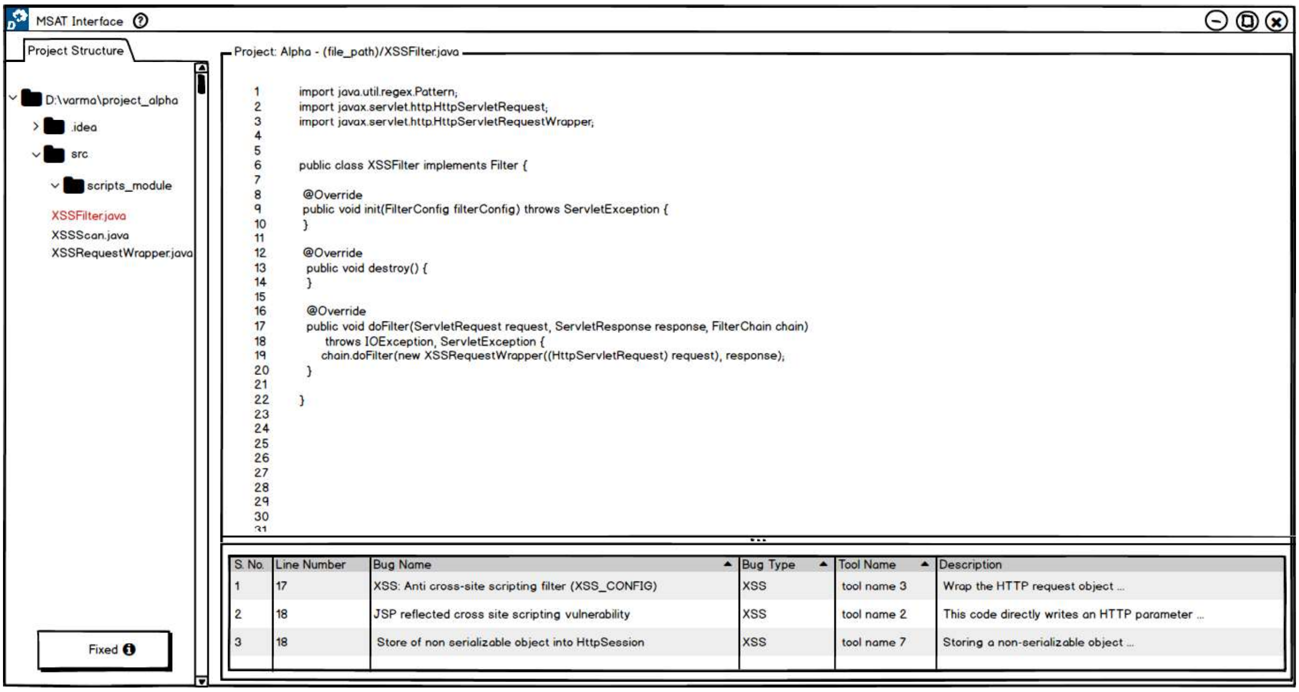
This code seems to be storing a non-serializable object into an HttpSession. If this session is passivated or migrated, an error will result.

UXD Cycle 3 - RQ 1 - List







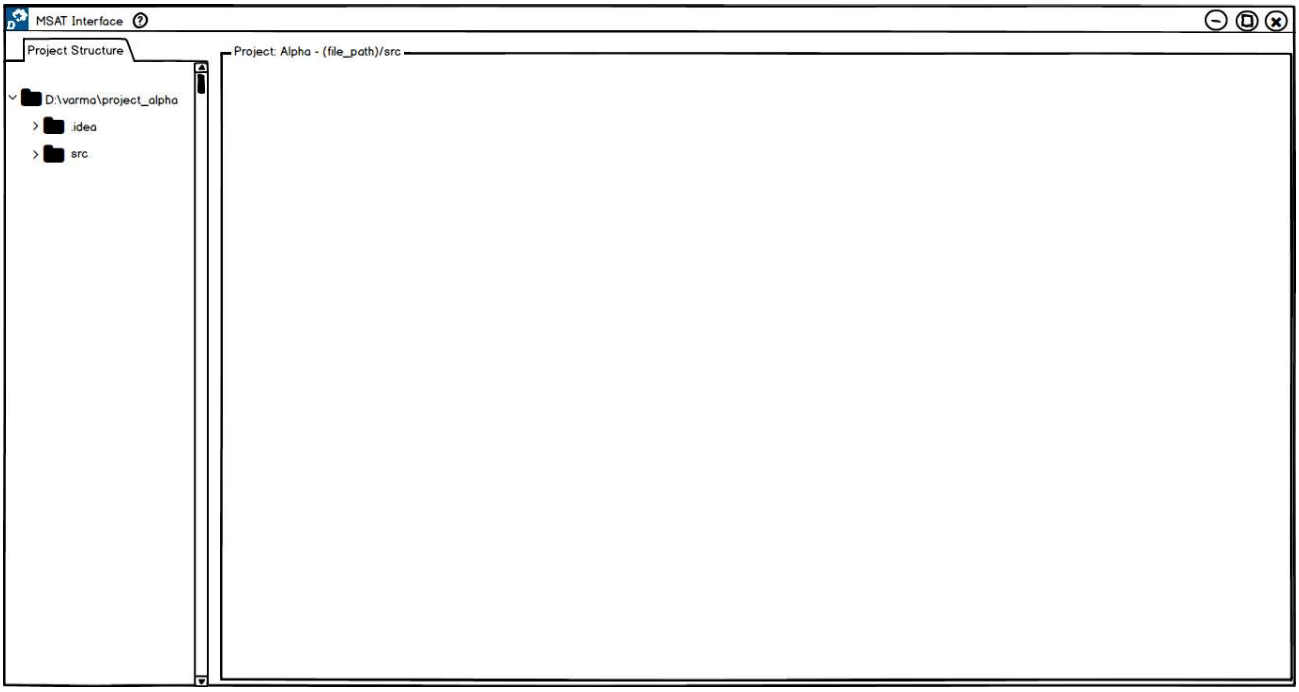


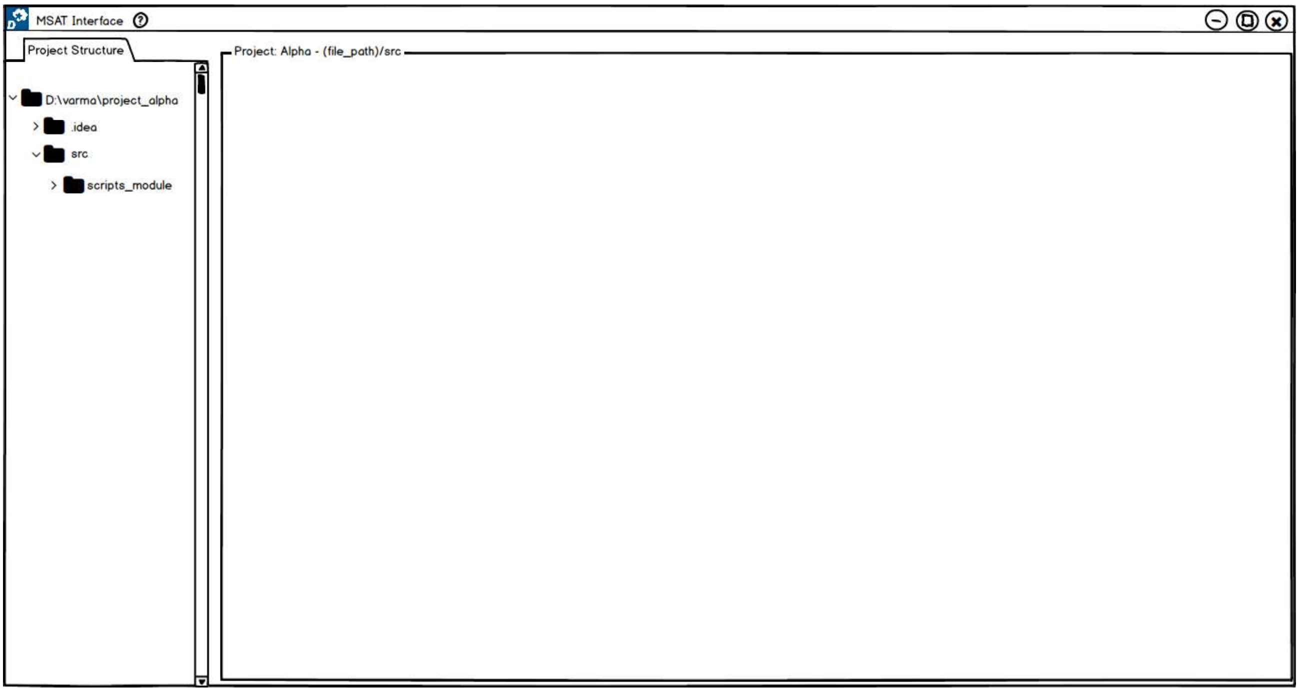
Project: Alpha - (file_path)/XSSFilter.java

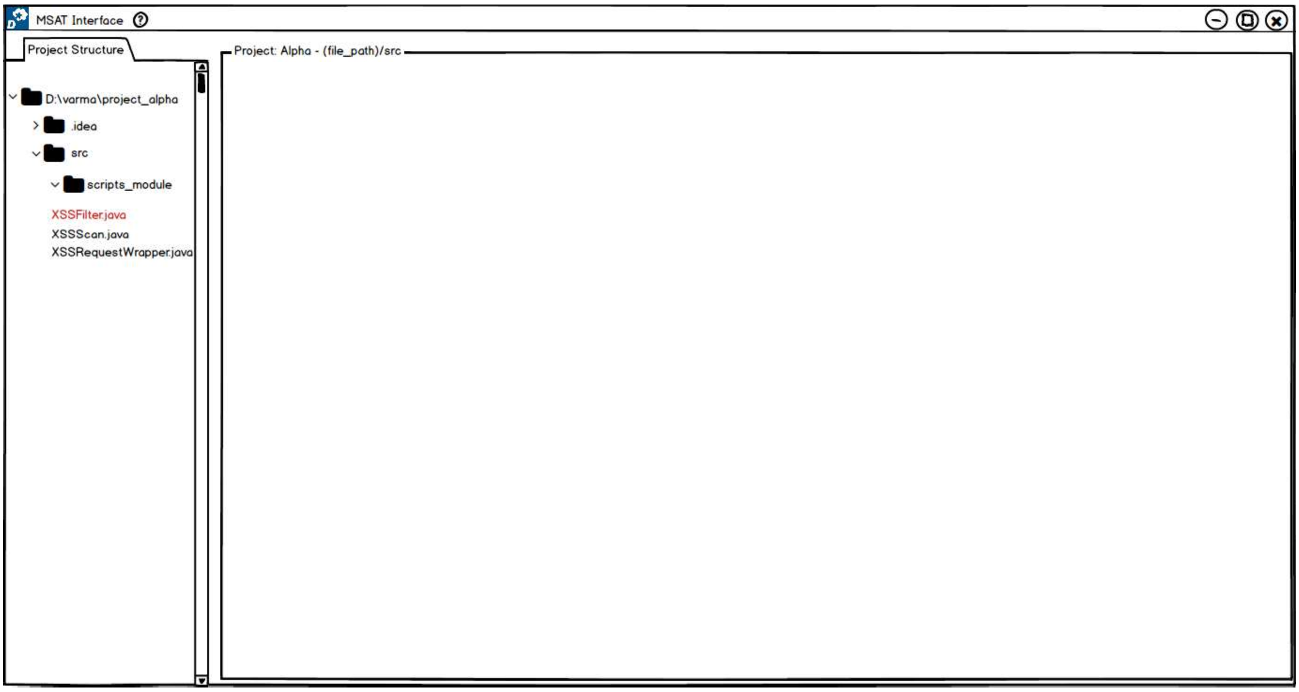
```
1         import java.util.regex.Pattern;
2         import javax.servlet.http.HttpServletRequest;
3         import javax.servlet.http.HttpServletRequestWrapper;
4
5
6         public class XSSFilter implements Filter {
7
8          @Override
9          public void init(FilterConfig filterConfig) throws ServletException {
10         }
11
12         @Override
13          public void destroy() {
14          }
15
16          @Override
17          public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
18               throws IOException, ServletException {
19              chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
20          }
21
22        }
23
24
25
26
27
28
29
30
31
```

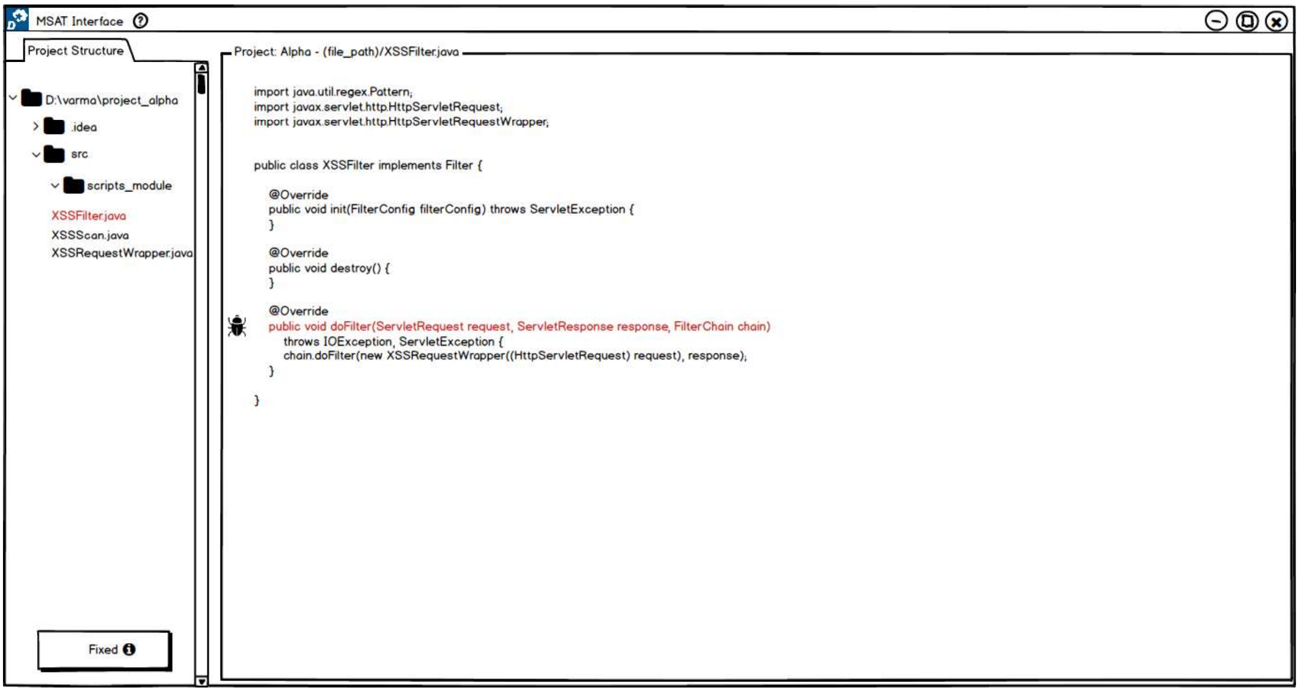
S. No.	Line Number	Bug Name	Bug Type	Tool Name	Description
1	17	XSS: Anti cross-site scripting filter (XSS_CONFIG)	XSS	tool name 3	Wrap the HTTP request object ...
2	18	JSP reflected cross site scripting vulnerability	XSS	tool name 2	This code directly writes an HTTP parameter ...
3	18	Store of non serializable object into HttpSession	XSS	tool name 7	Storing a non-serializable object ...

UXD Cycle 3 - RQ 1 - Horizontal



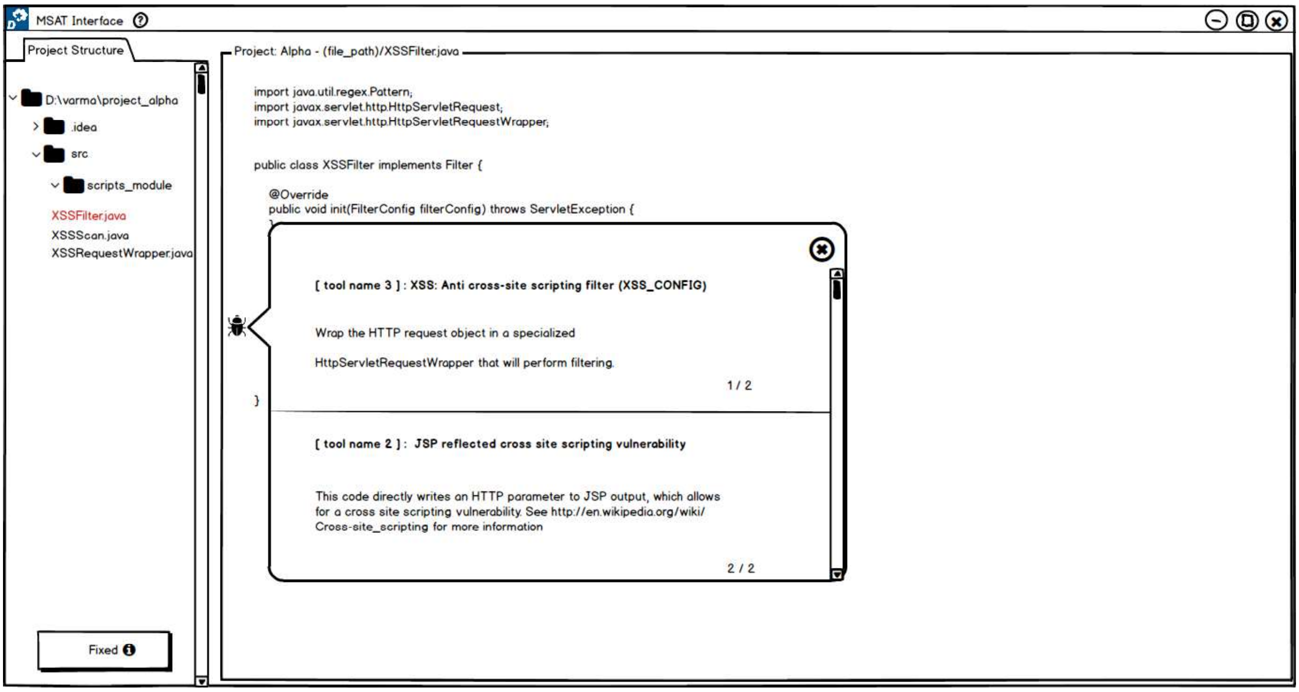






Project: Alpha - (file_path)/XSSFilter.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {
   }

   @Override
   public void destroy() {
   }

   @Override
   public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
      throws IOException, ServletException {
      chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
   }

}
```



Project: Alpha - (file_path)/XSSFilter.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {

}
```

[tool name 3] : XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized

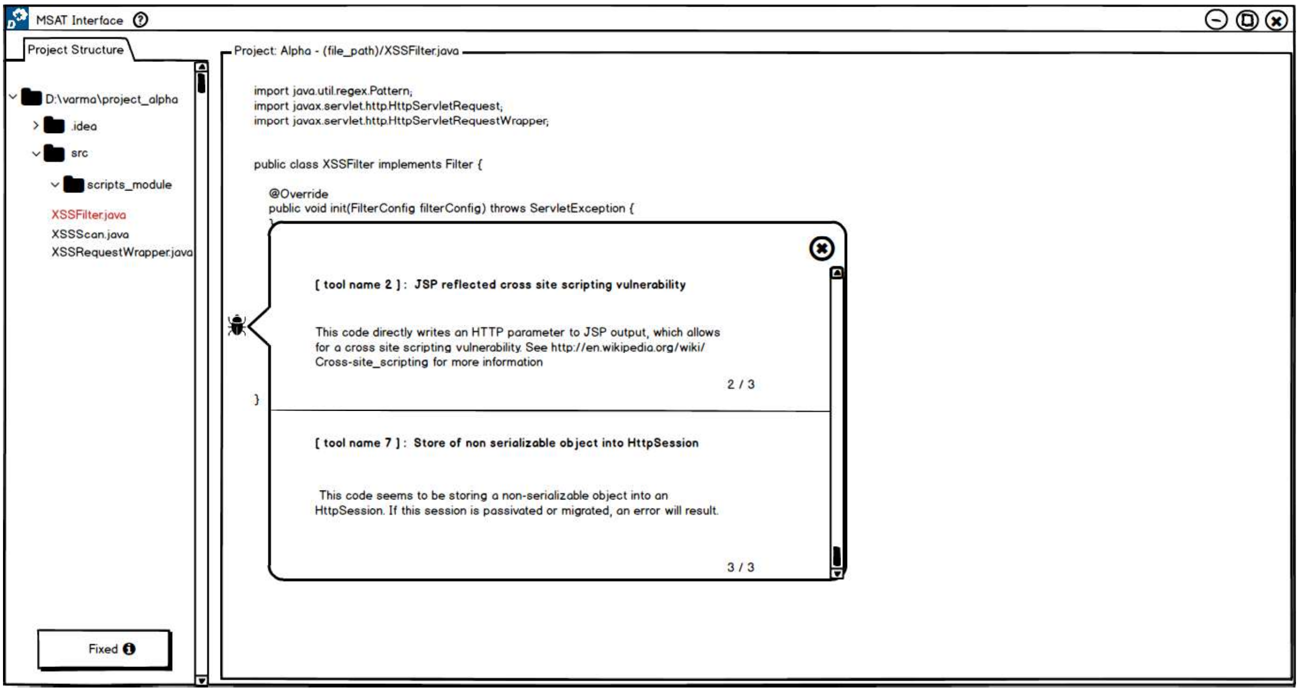
HttpServletRequestWrapper that will perform filtering.

1 / 2

[tool name 2] : JSP reflected cross site scripting vulnerability

This code directly writes an HTTP parameter to JSP output, which allows for a cross site scripting vulnerability. See http://en.wikipedia.org/wiki/Cross-site_scripting for more information

2 / 2



Project: Alpha - (file_path)/XSSFilter.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {

}
```

[tool name 2] : JSP reflected cross site scripting vulnerability

This code directly writes an HTTP parameter to JSP output, which allows for a cross site scripting vulnerability. See http://en.wikipedia.org/wiki/Cross-site_scripting for more information

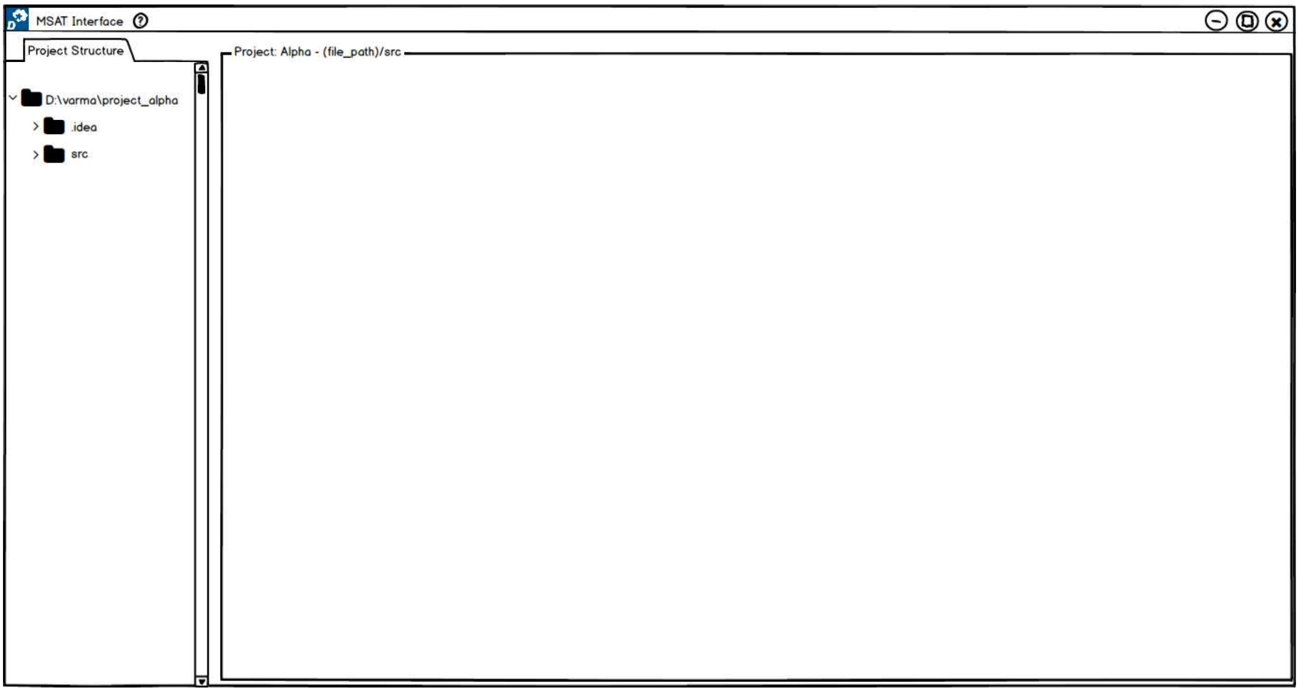
2 / 3

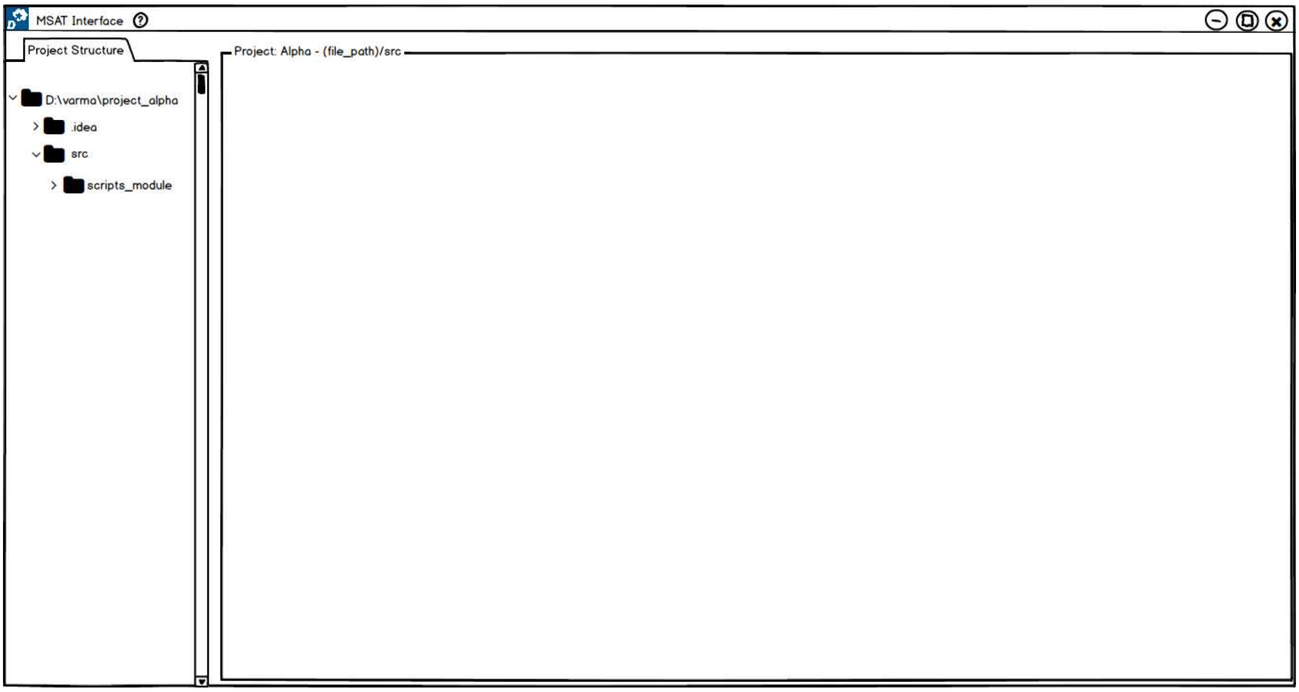
[tool name 7] : Store of non serializable object into HttpSession

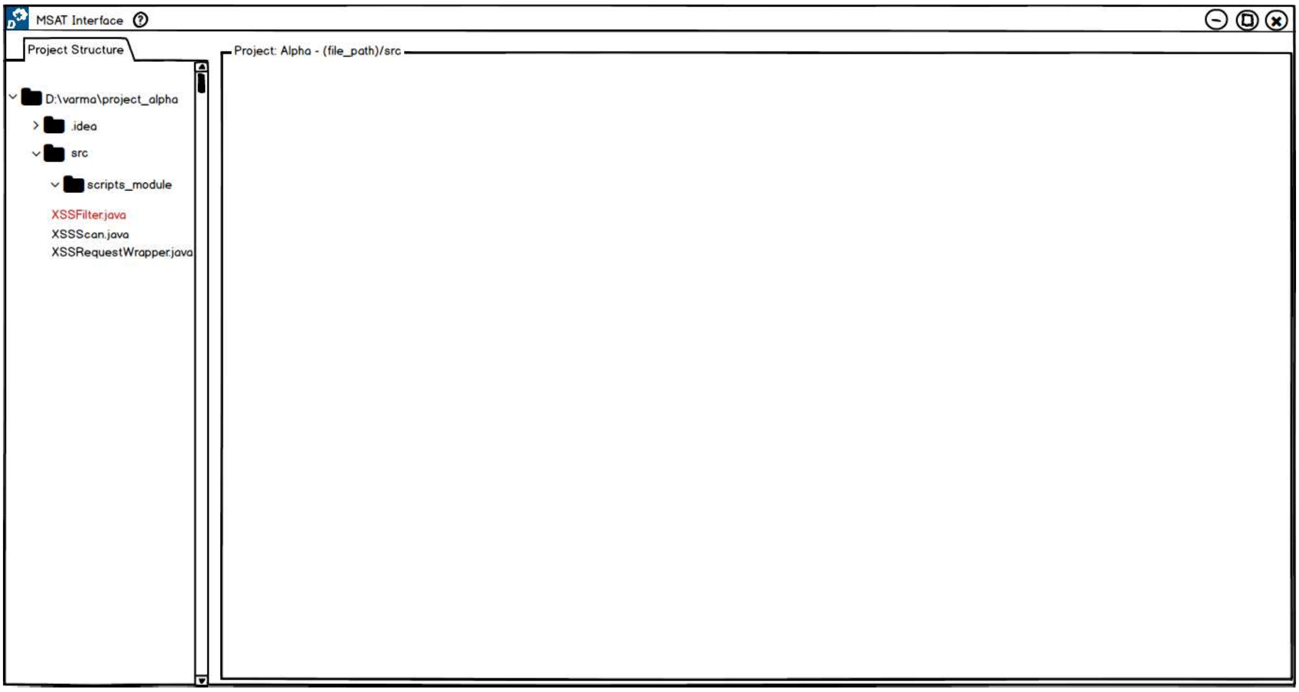
This code seems to be storing a non-serializable object into an HttpSession. If this session is passivated or migrated, an error will result.

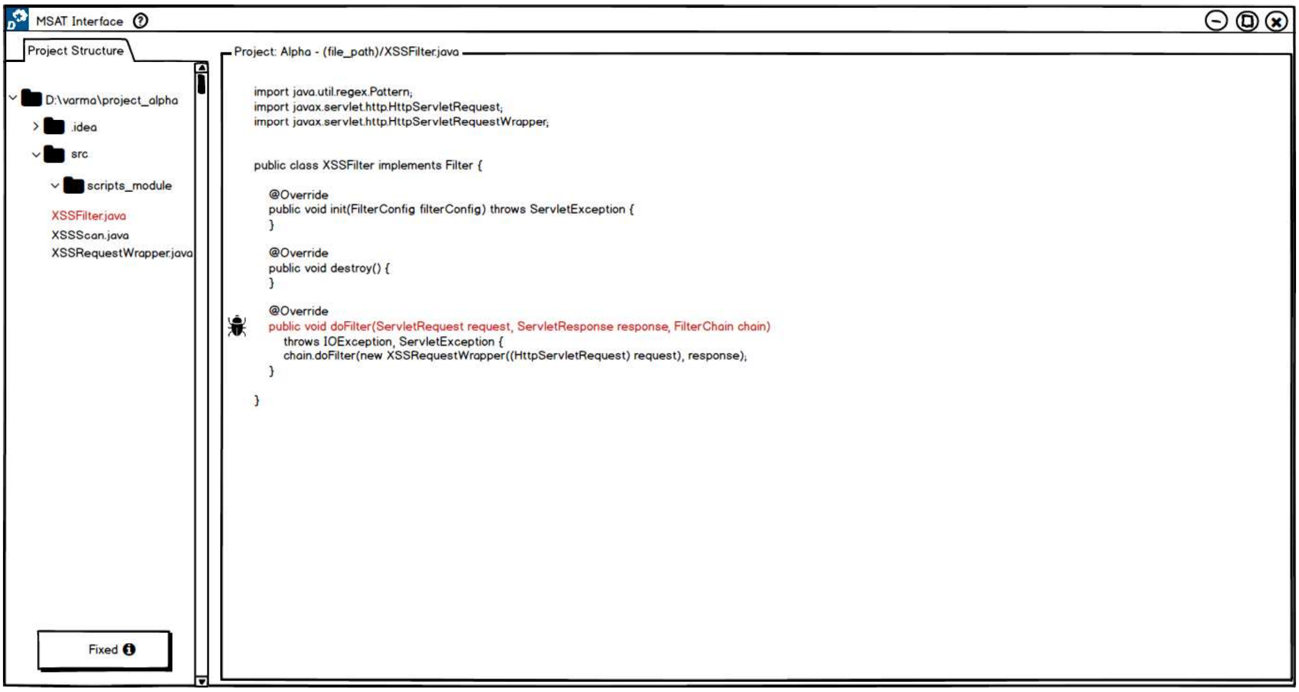
3 / 3

UXD Cycle 3 - RQ 1 - Vertical



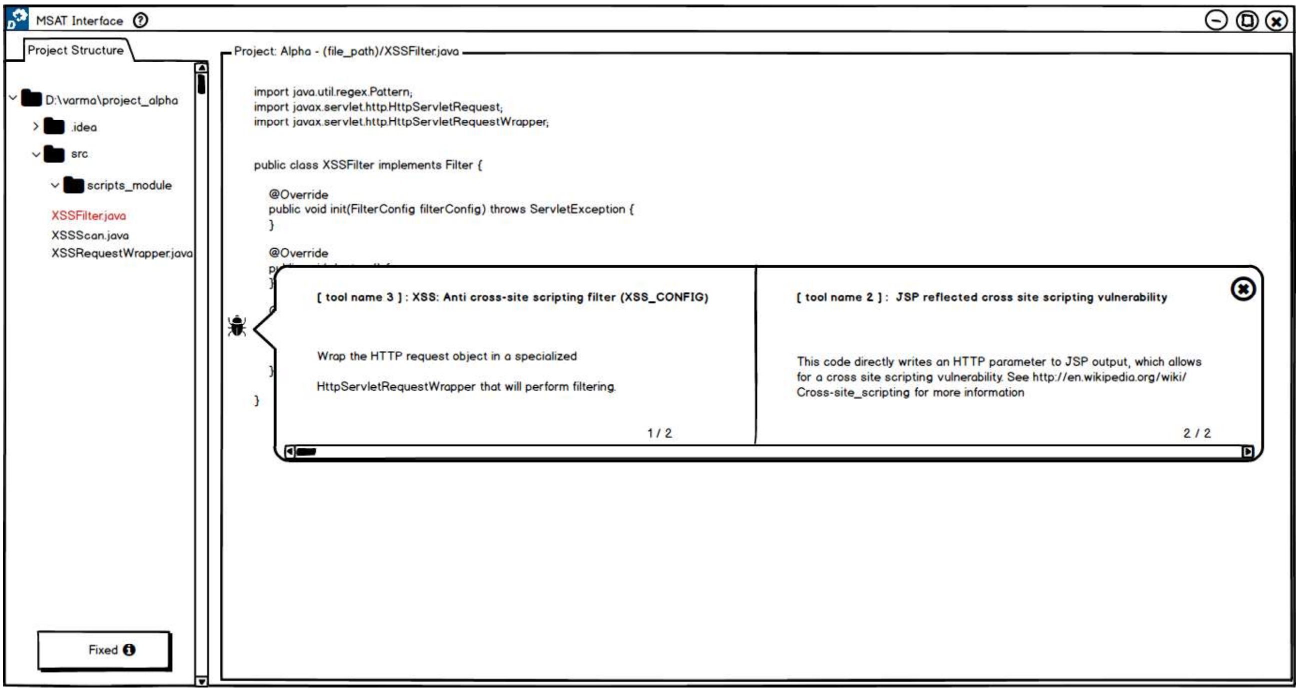






Project: Alpha - (file_path)/XSSFilter.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

  @Override
  public void init(FilterConfig filterConfig) throws ServletException {
  }

  @Override
  public void destroy() {
  }

  @Override
  public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
    throws IOException, ServletException {
    chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
  }

}
```



Project: Alpha - (file_path)/XSSFilter.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

  @Override
  public void init(FilterConfig filterConfig) throws ServletException {
  }

  @Override
```

[tool name 3] : XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized

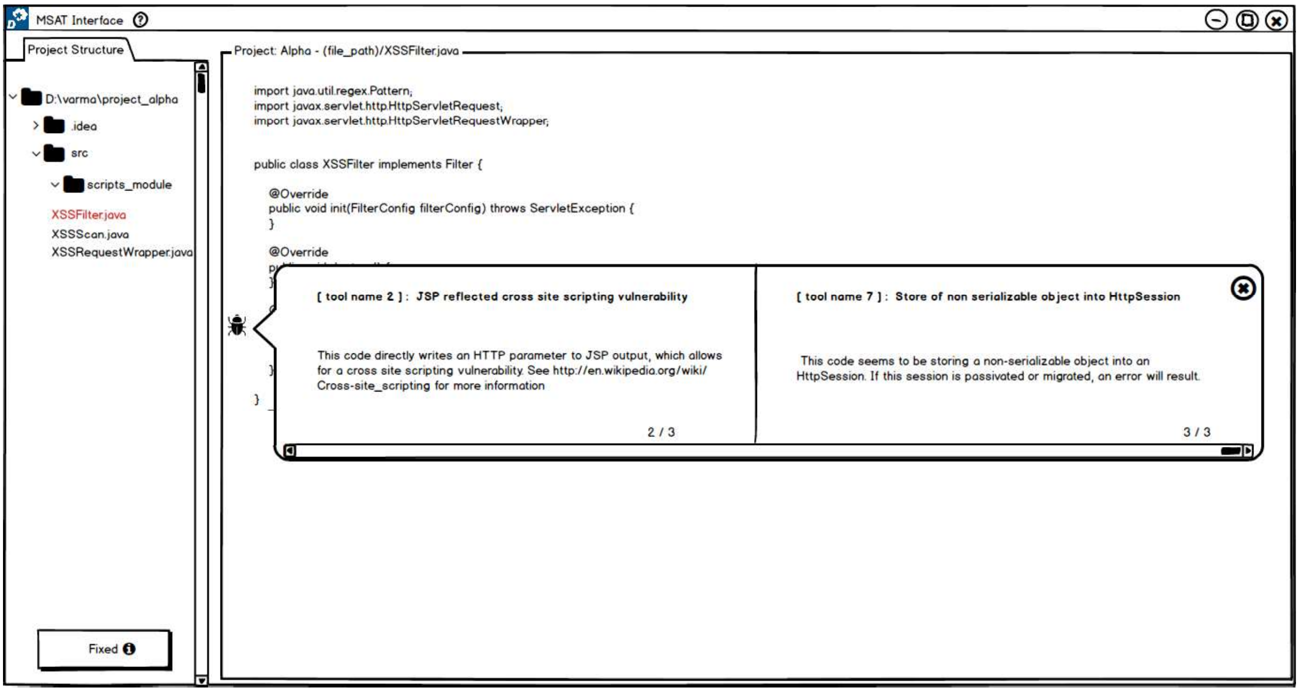
HttpServletRequestWrapper that will perform filtering.

1 / 2

[tool name 2] : JSP reflected cross site scripting vulnerability

This code directly writes an HTTP parameter to JSP output, which allows for a cross site scripting vulnerability. See http://en.wikipedia.org/wiki/Cross-site_scripting for more information

2 / 2



Project: Alpha - (file_path)/XSSFilter.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

  @Override
  public void init(FilterConfig filterConfig) throws ServletException {
  }

  @Override
```

[tool name 2] : JSP reflected cross site scripting vulnerability

This code directly writes an HTTP parameter to JSP output, which allows for a cross site scripting vulnerability. See http://en.wikipedia.org/wiki/Cross-site_scripting for more information

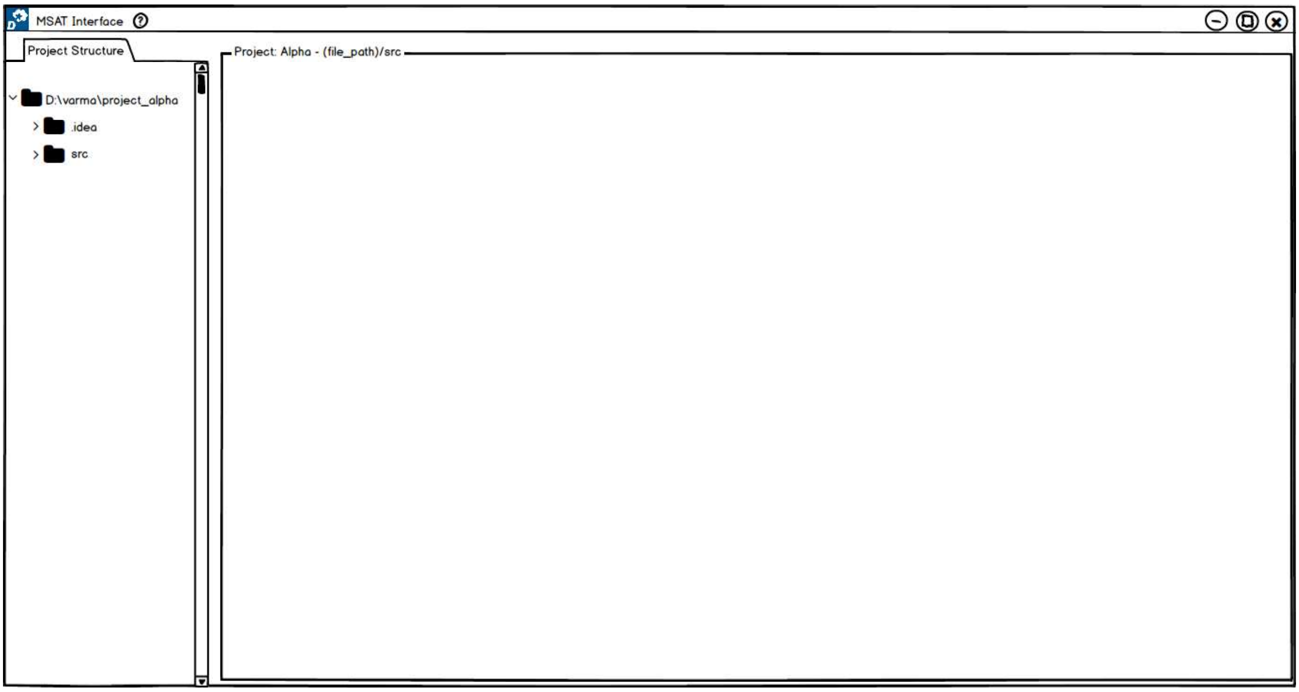
2 / 3

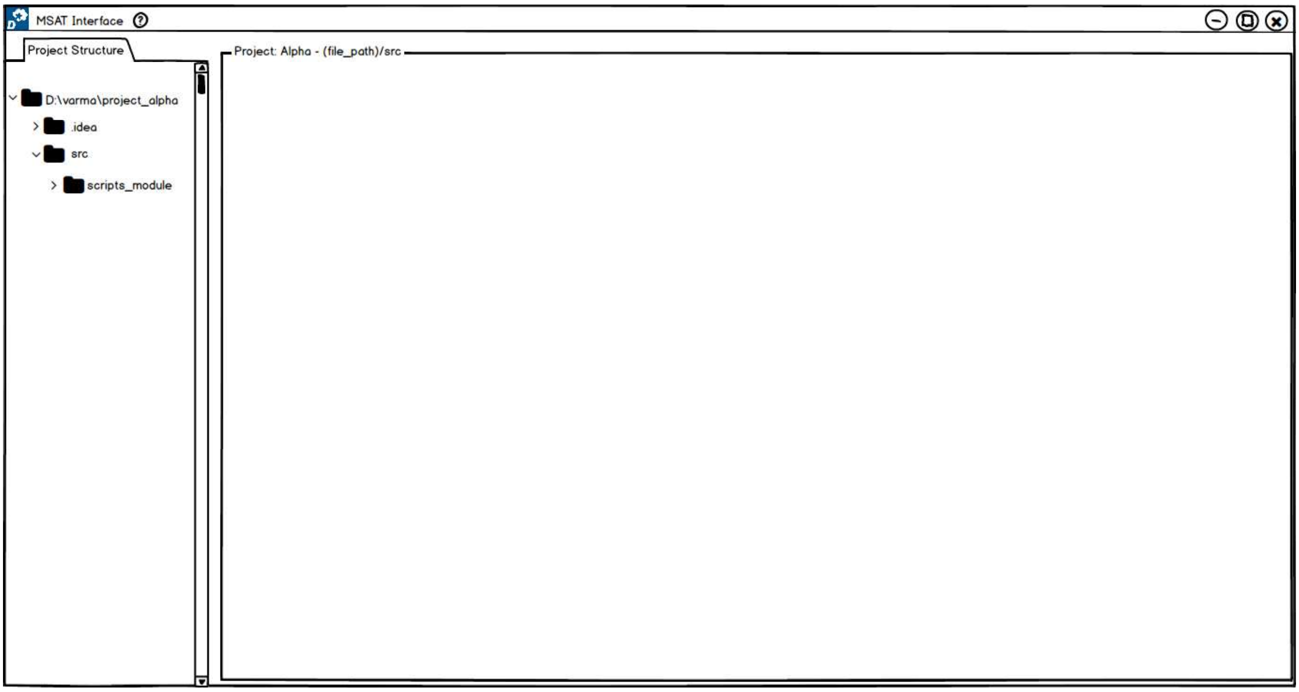
[tool name 7] : Store of non serializable object into HttpSession

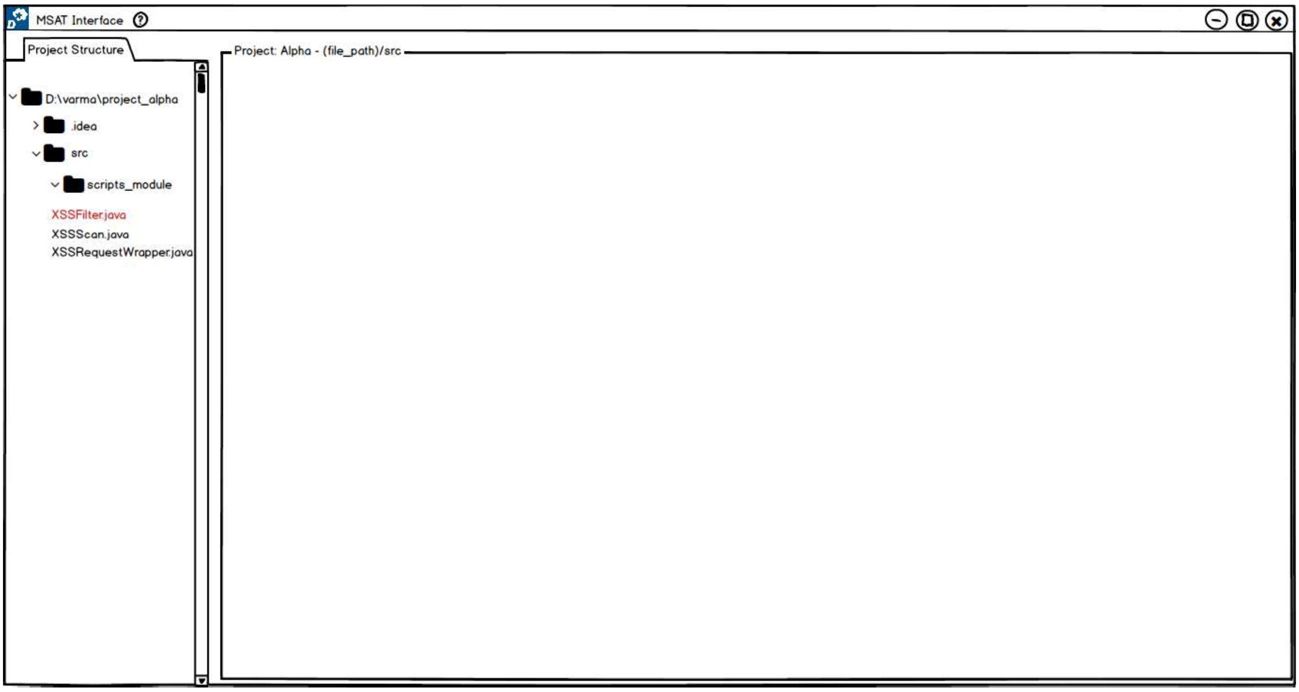
This code seems to be storing a non-serializable object into an HttpSession. If this session is passivated or migrated, an error will result.

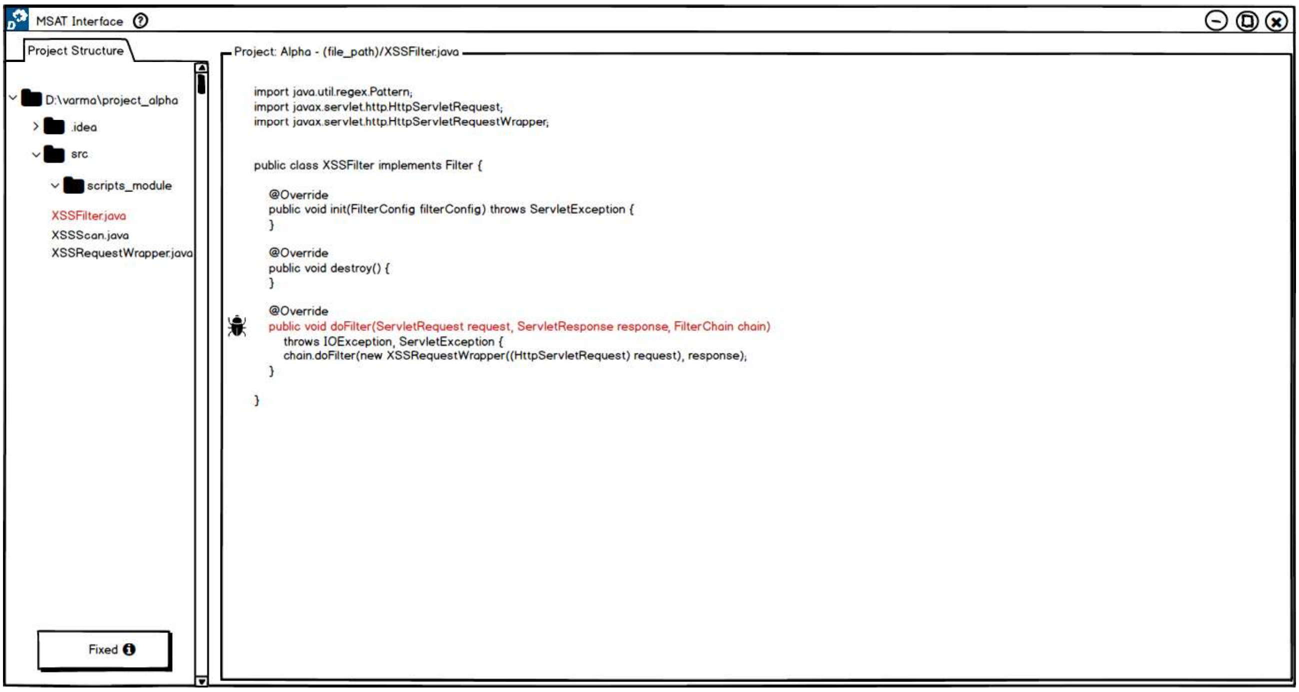
3 / 3

UXD Cycle 3 - RQ 1 - Table



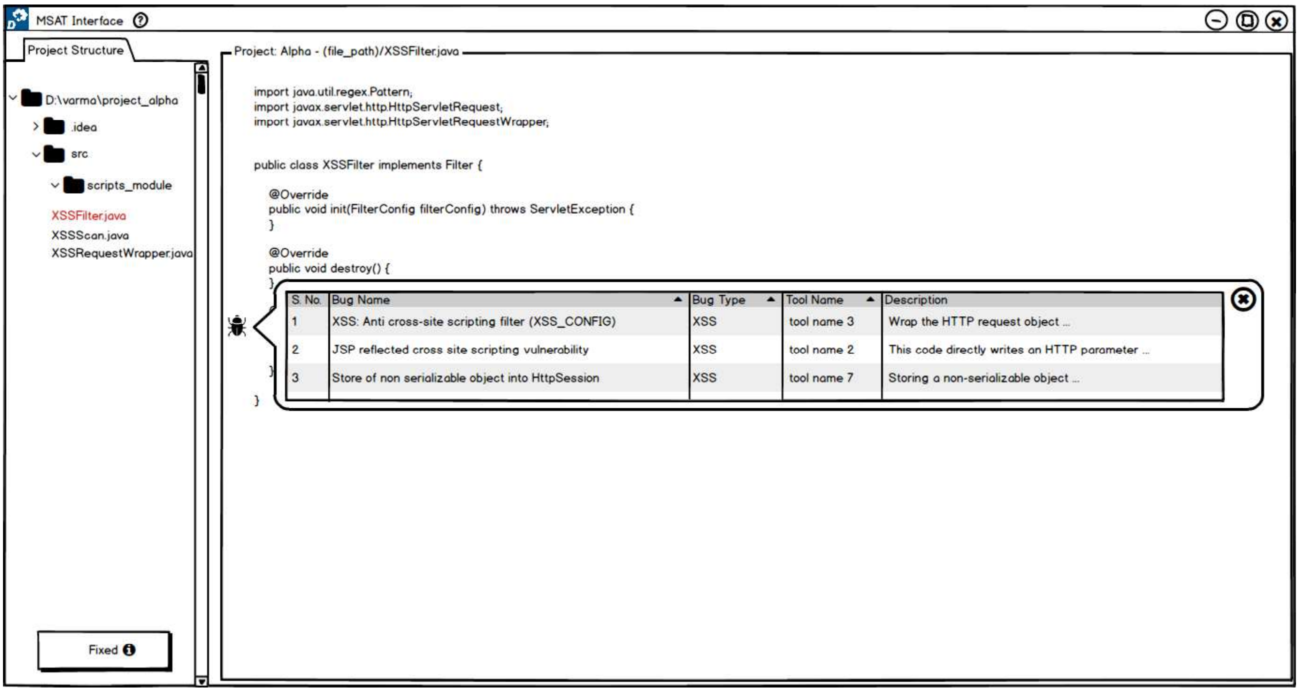






Project: Alpha - (file_path)/XSSFilter.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {
   }

   @Override
   public void destroy() {
   }

   @Override
   public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
      throws IOException, ServletException {
      chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
   }

}
```

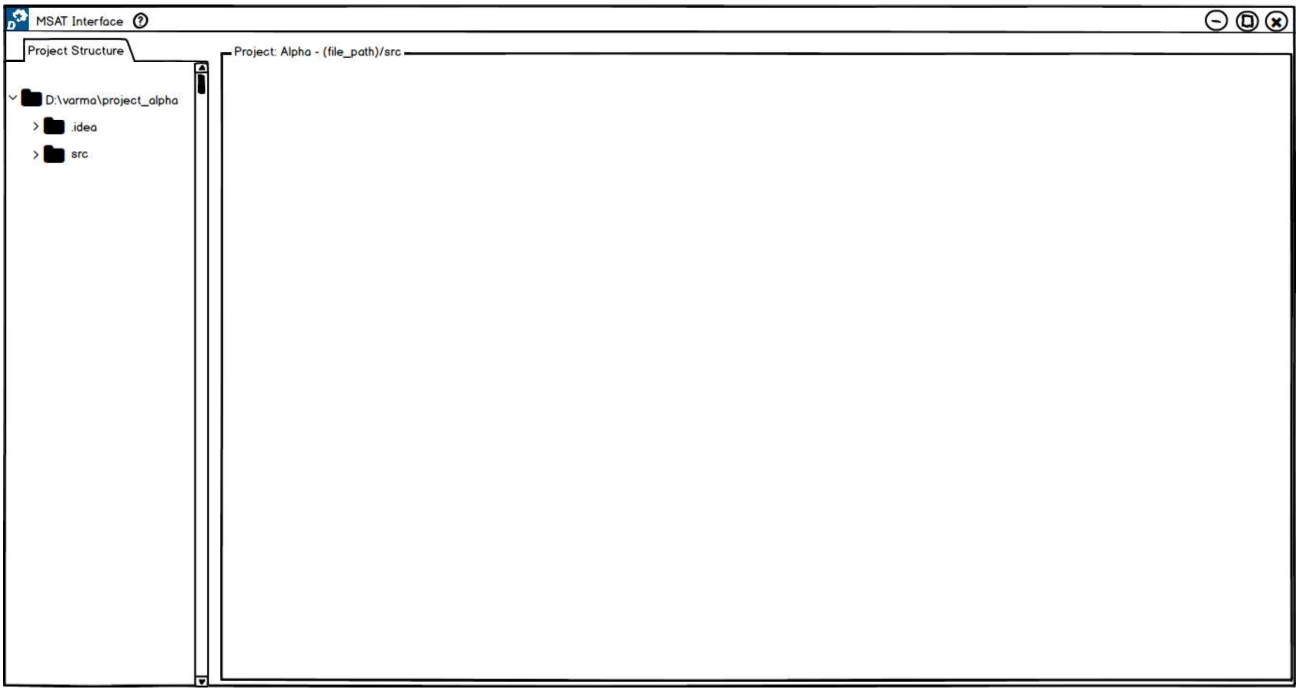


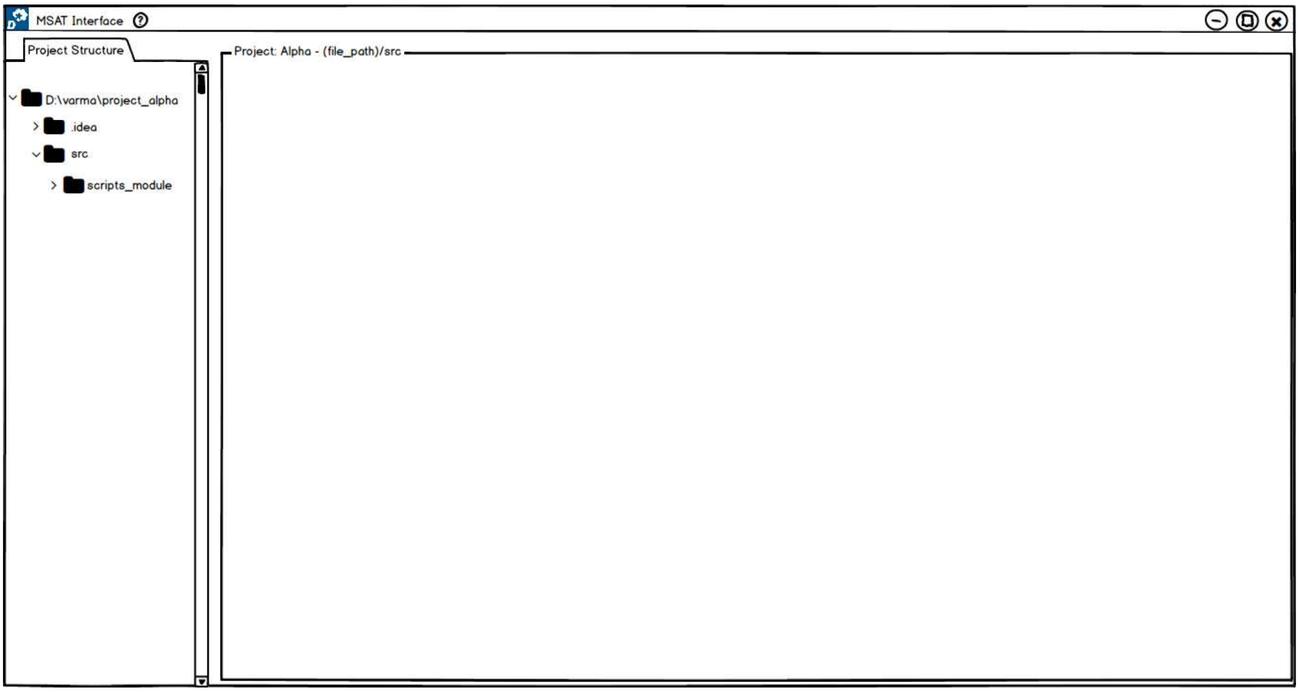
Project: Alpha - (file_path)/XSSFilter.java

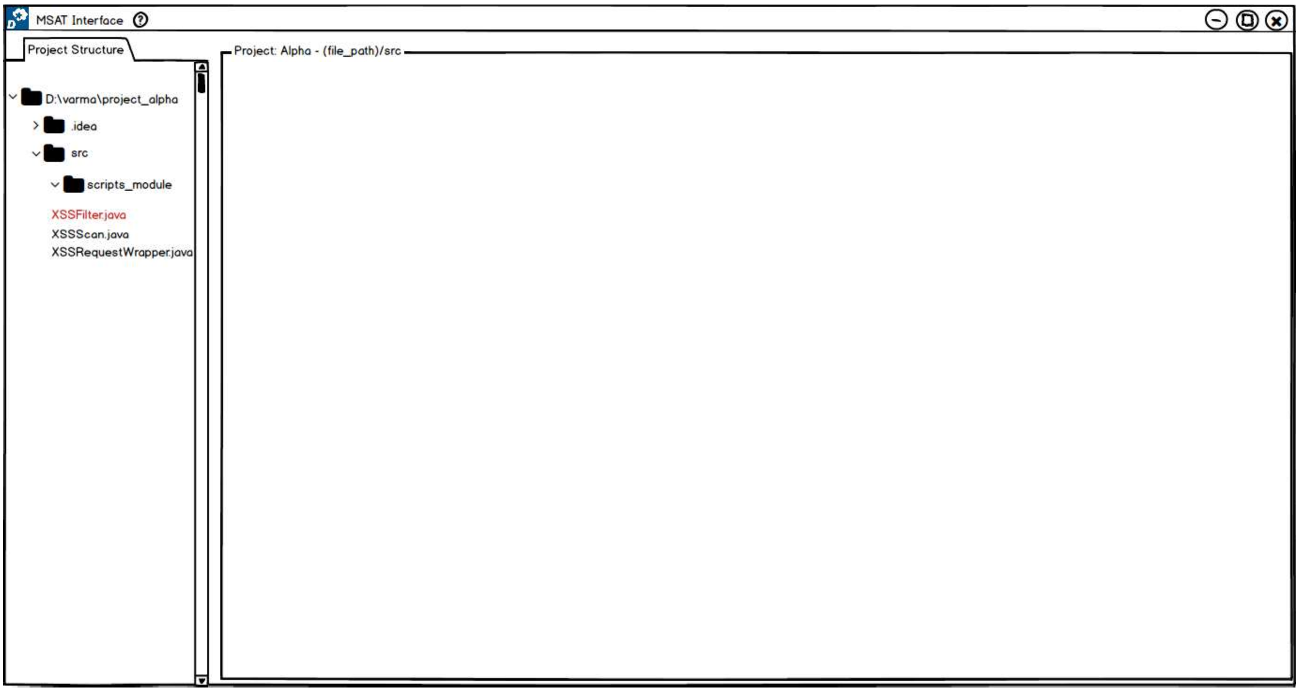
```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {
   }

   @Override
   public void destroy() {
   }

   }

}
```

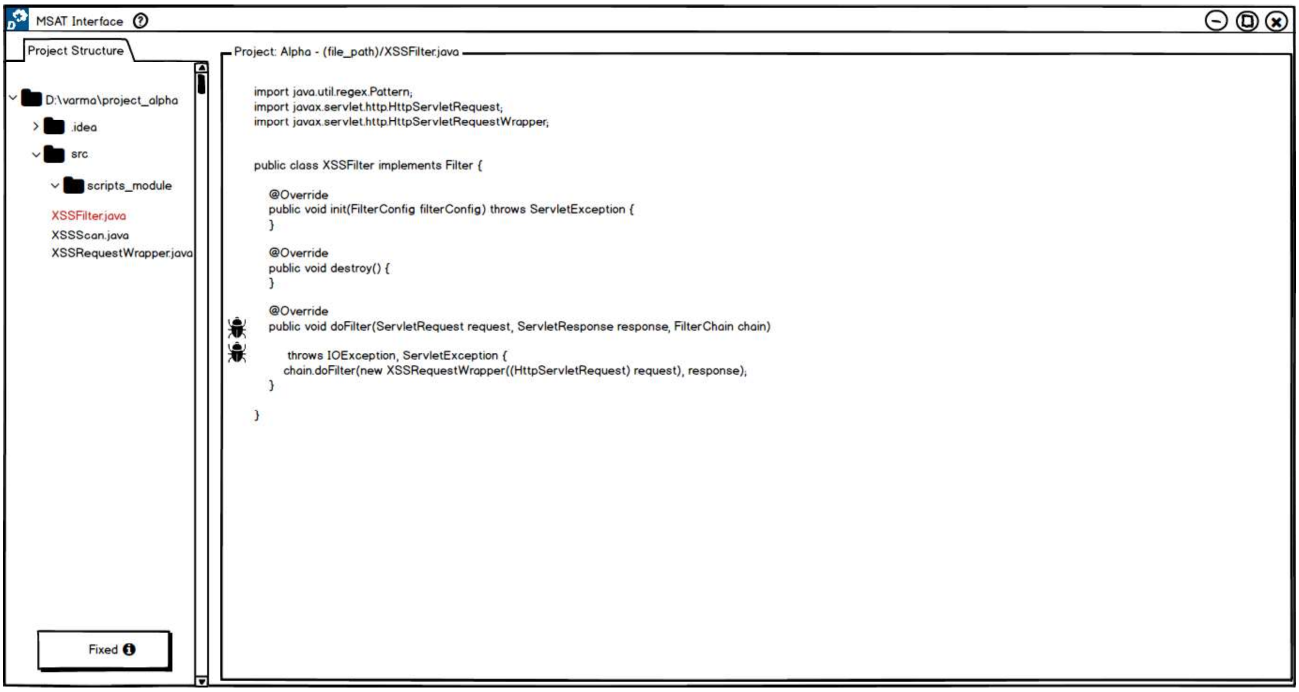
S. No.	Bug Name	Bug Type	Tool Name	Description
1	XSS: Anti cross-site scripting filter (XSS_CONFIG)	XSS	tool name 3	Wrap the HTTP request object ...
2	JSP reflected cross site scripting vulnerability	XSS	tool name 2	This code directly writes an HTTP parameter ...
3	Store of non serializable object into HttpSession	XSS	tool name 7	Storing a non-serializable object ...

UXD Cycle 3 - RQ 1 - Similar Boxes

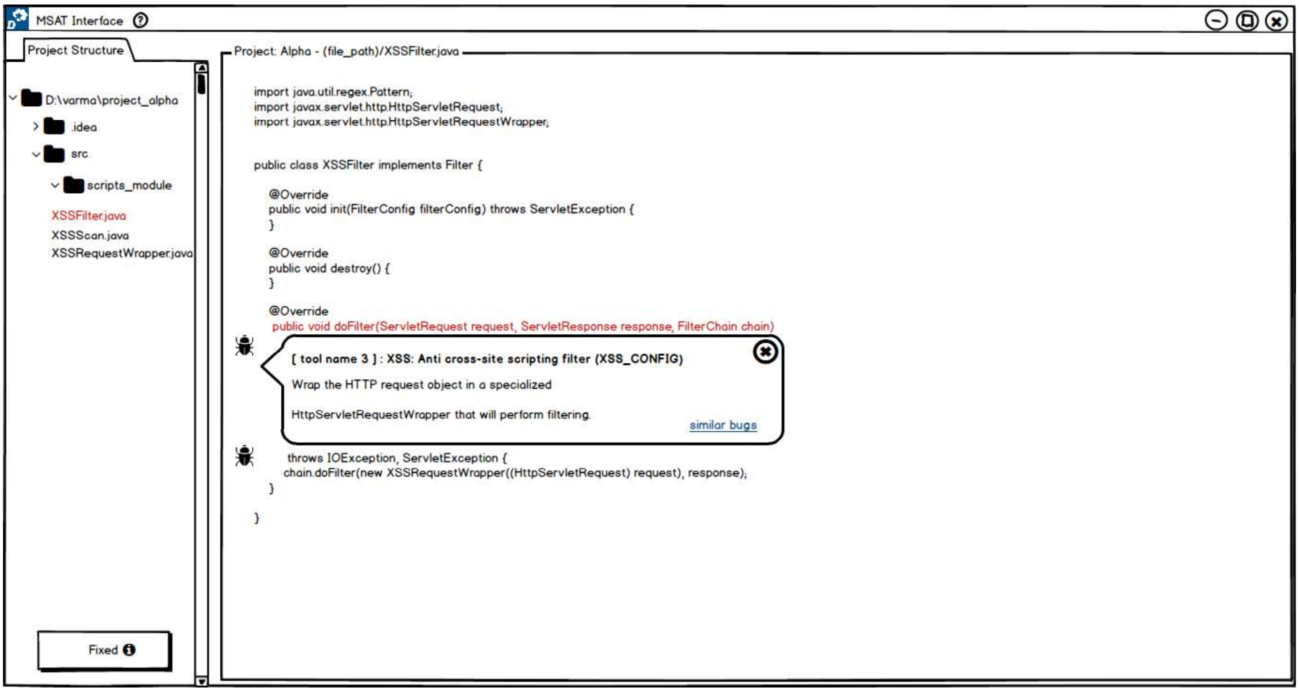








```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSFilter implements Filter {

  @Override
  public void init(FilterConfig filterConfig) throws ServletException {
  }

  @Override
  public void destroy() {
  }

  @Override
  public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)

     throws IOException, ServletException {
    chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
  }

}
```



```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSFilter implements Filter {

  @Override
  public void init(FilterConfig filterConfig) throws ServletException {
  }

  @Override
  public void destroy() {
  }

  @Override
  public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
```

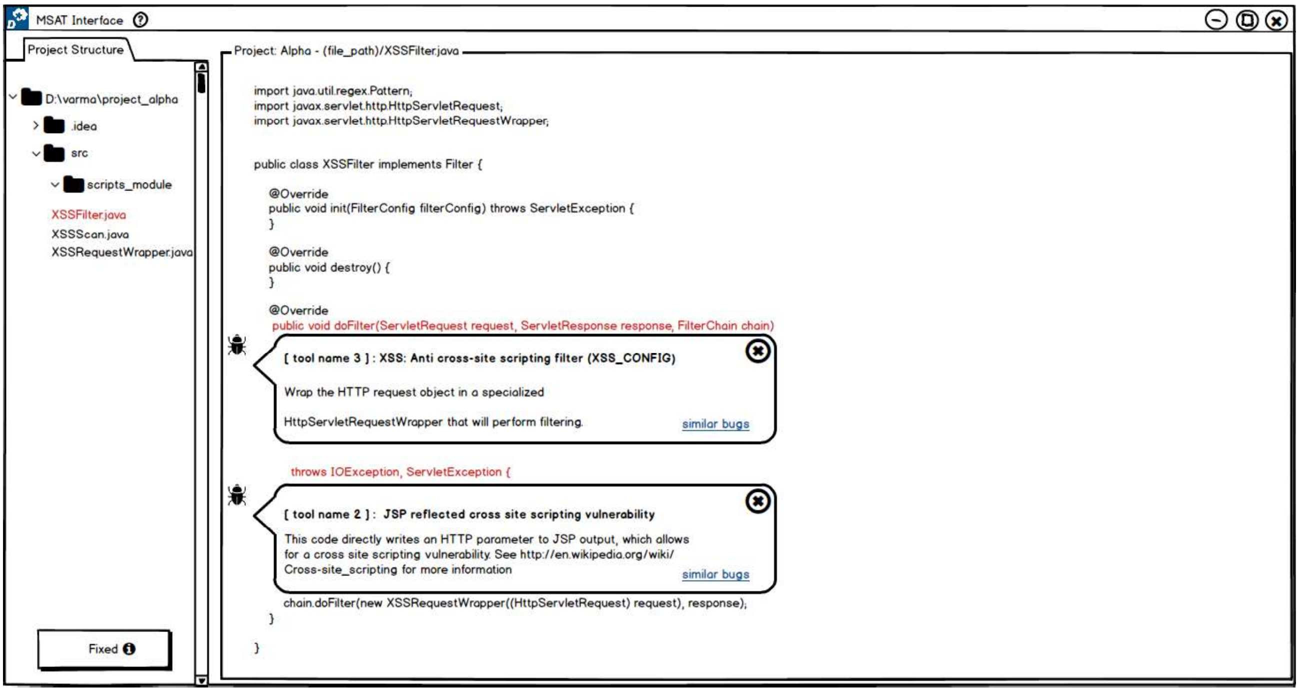
[tool name 3] : XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized

HttpServletRequestWrapper that will perform filtering.

similar bugs

```
     throws IOException, ServletException {
    chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
  }

}
```



```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSFilter implements Filter {

  @Override
  public void init(FilterConfig filterConfig) throws ServletException {
  }

  @Override
  public void destroy() {
  }

  @Override
  public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
```

[tool name 3] : XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized

HttpServletRequestWrapper that will perform filtering.

similar bugs

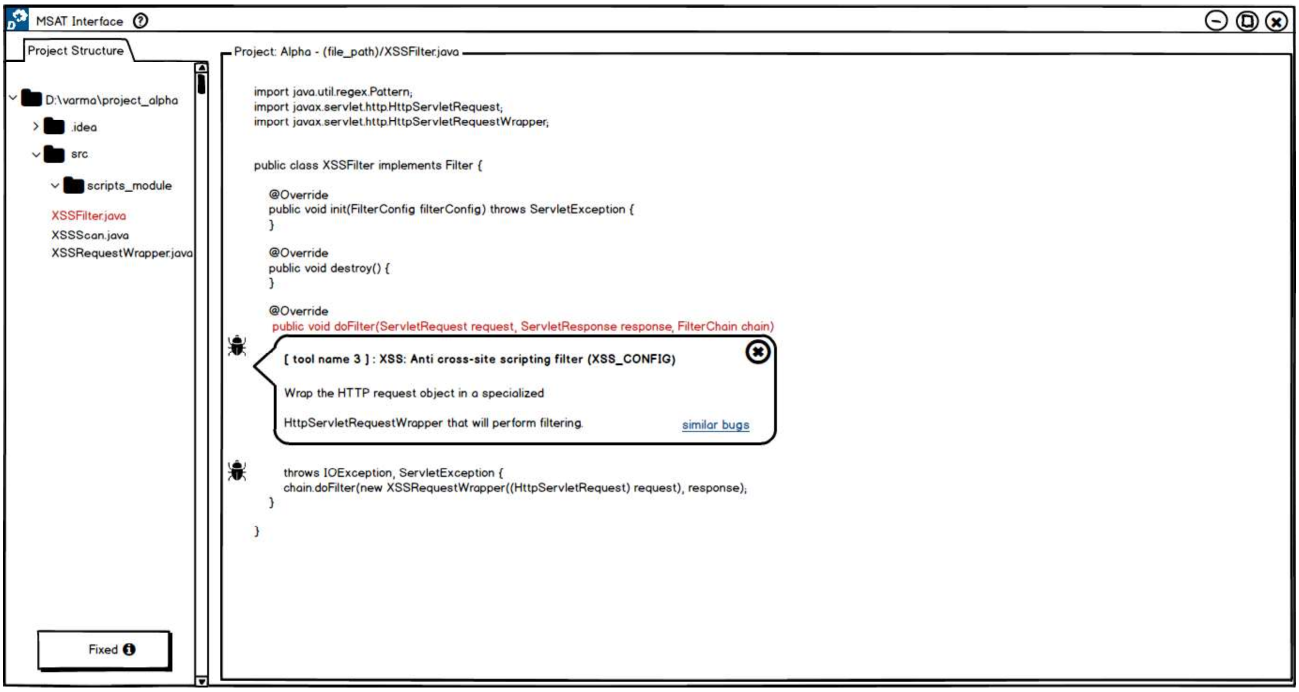
```
     throws IOException, ServletException {
```

[tool name 2] : JSP reflected cross site scripting vulnerability

This code directly writes an HTTP parameter to JSP output, which allows
for a cross site scripting vulnerability. See http://en.wikipedia.org/wiki/
Cross-site_scripting for more information

similar bugs

```
    chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
  }

}
```



```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
    }

    @Override
    public void destroy() {
    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
```

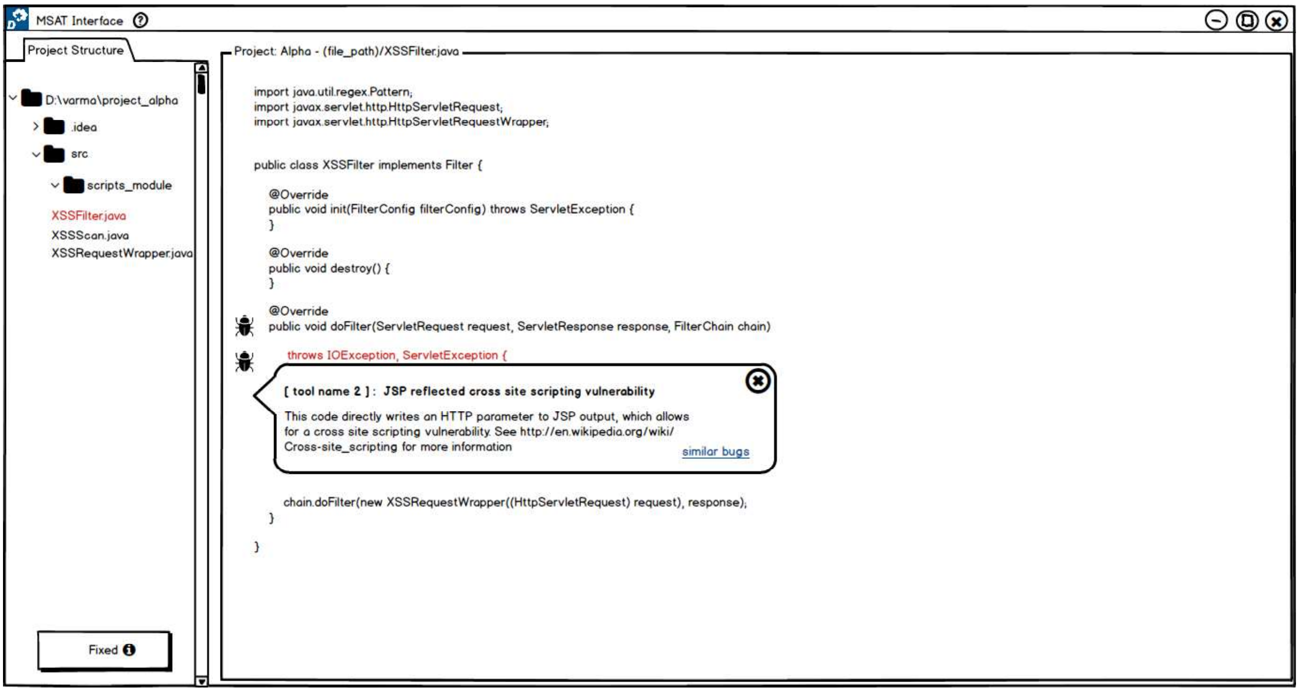
[tool name 3] : XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized

HttpServletRequestWrapper that will perform filtering.

similar bugs

```
        throws IOException, ServletException {
        chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
    }

}
```



```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
    }

    @Override
    public void destroy() {
    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)

        throws IOException, ServletException {
```

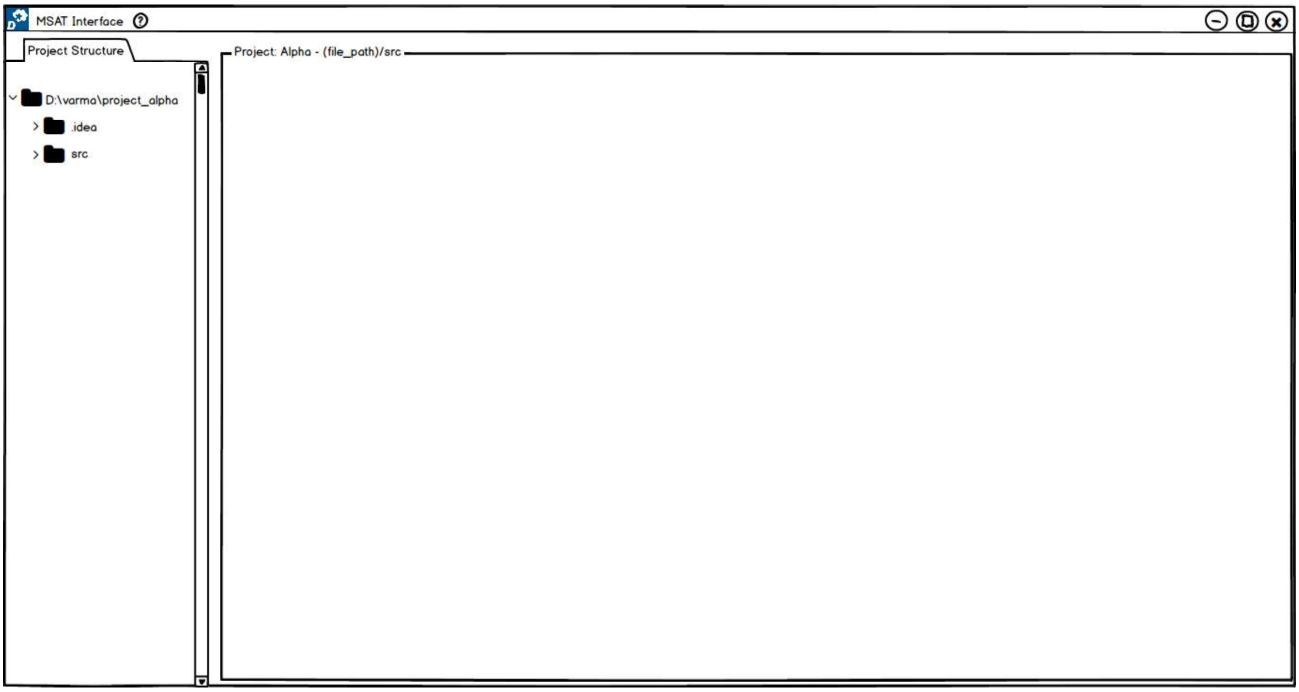
[tool name 2] : JSP reflected cross site scripting vulnerability

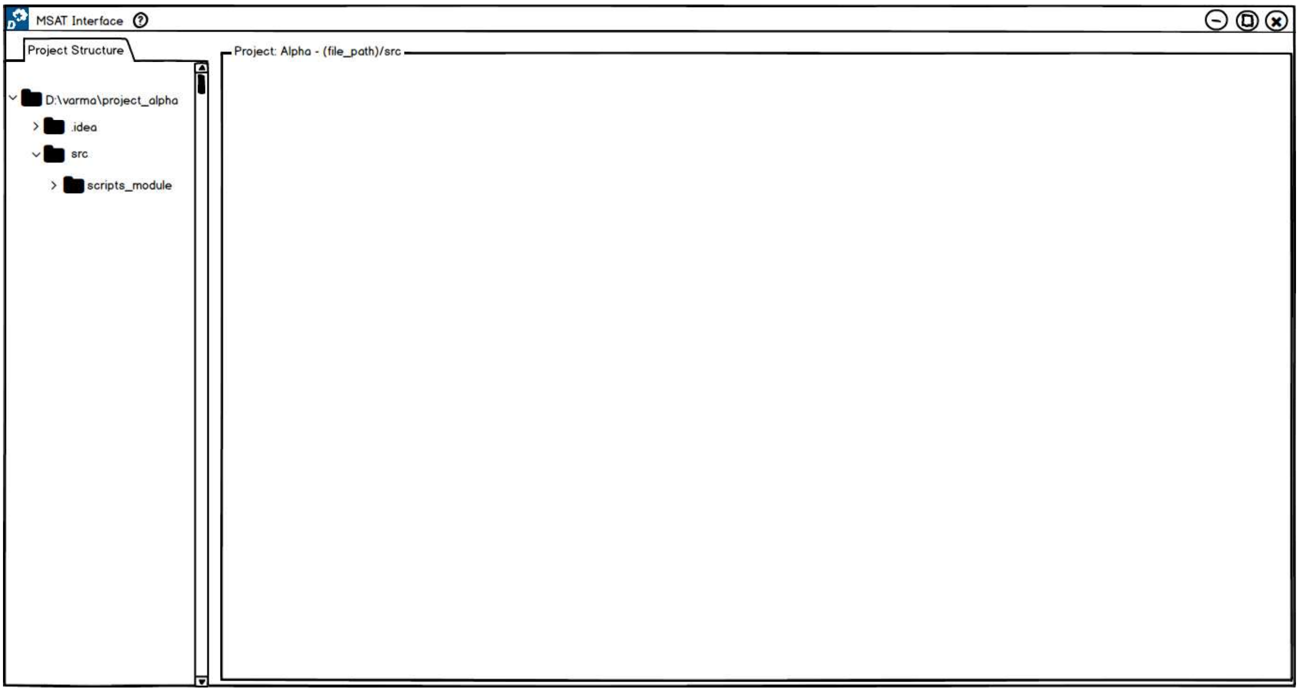
This code directly writes an HTTP parameter to JSP output, which allows for a cross site scripting vulnerability. See http://en.wikipedia.org/wiki/Cross-site_scripting for more information

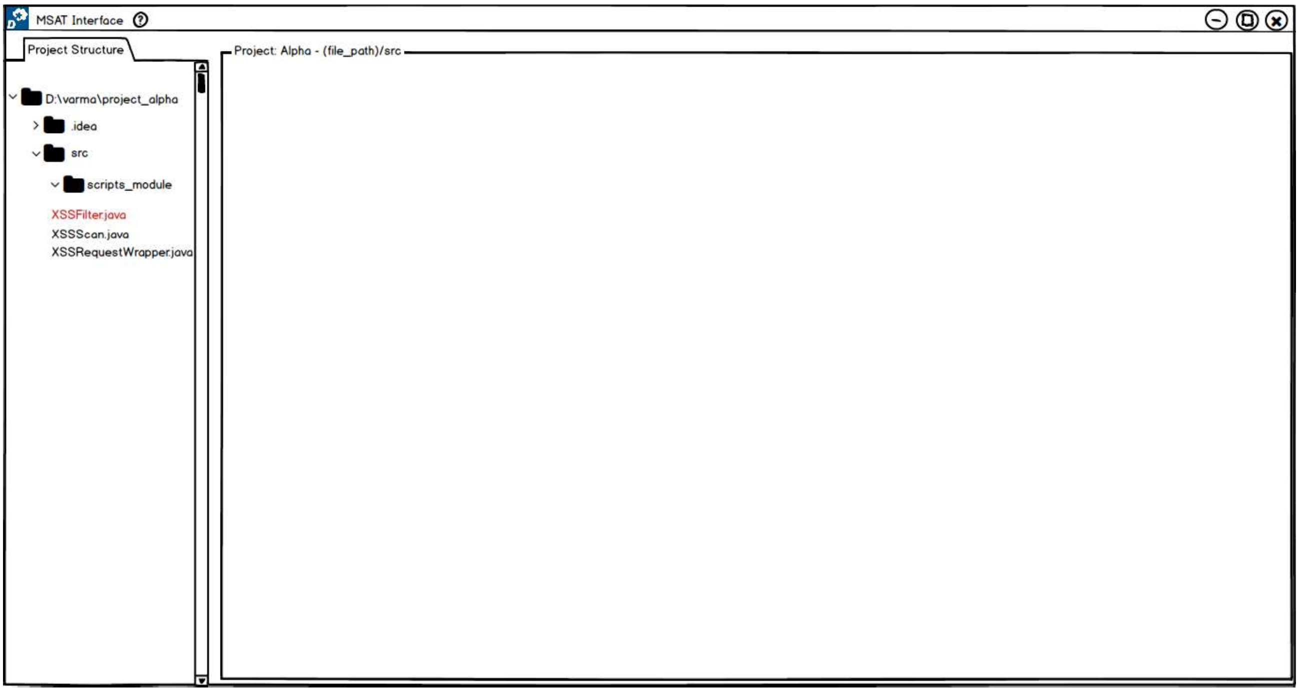
similar bugs

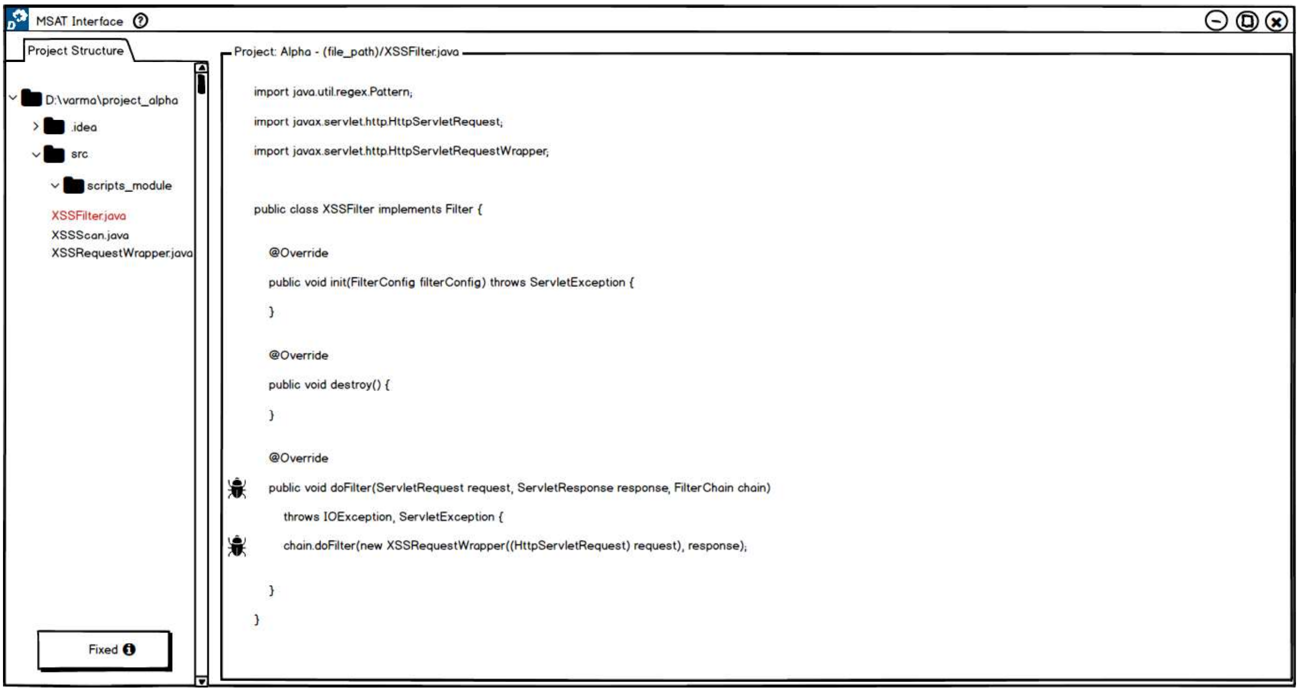
```
        chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
    }

}
```

UXD Cycle 3 - RQ 1 - Similar List



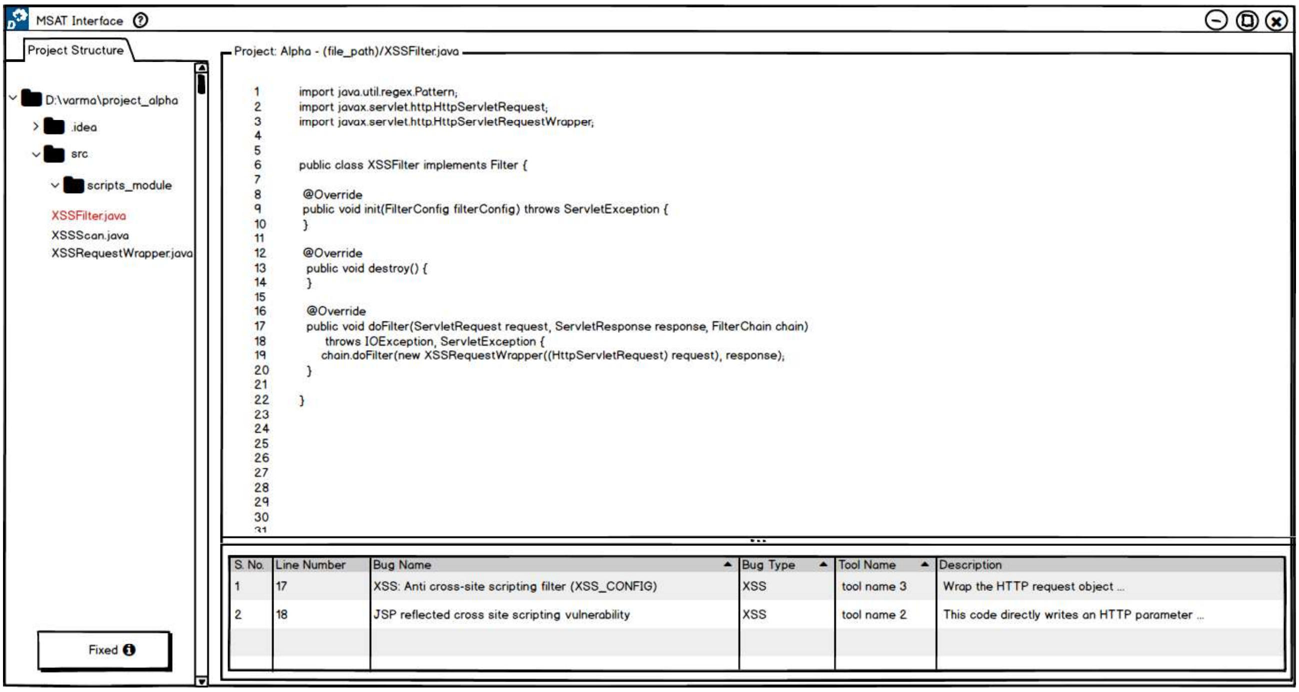






Project: Alpha - (file_path)/XSSFilter.java

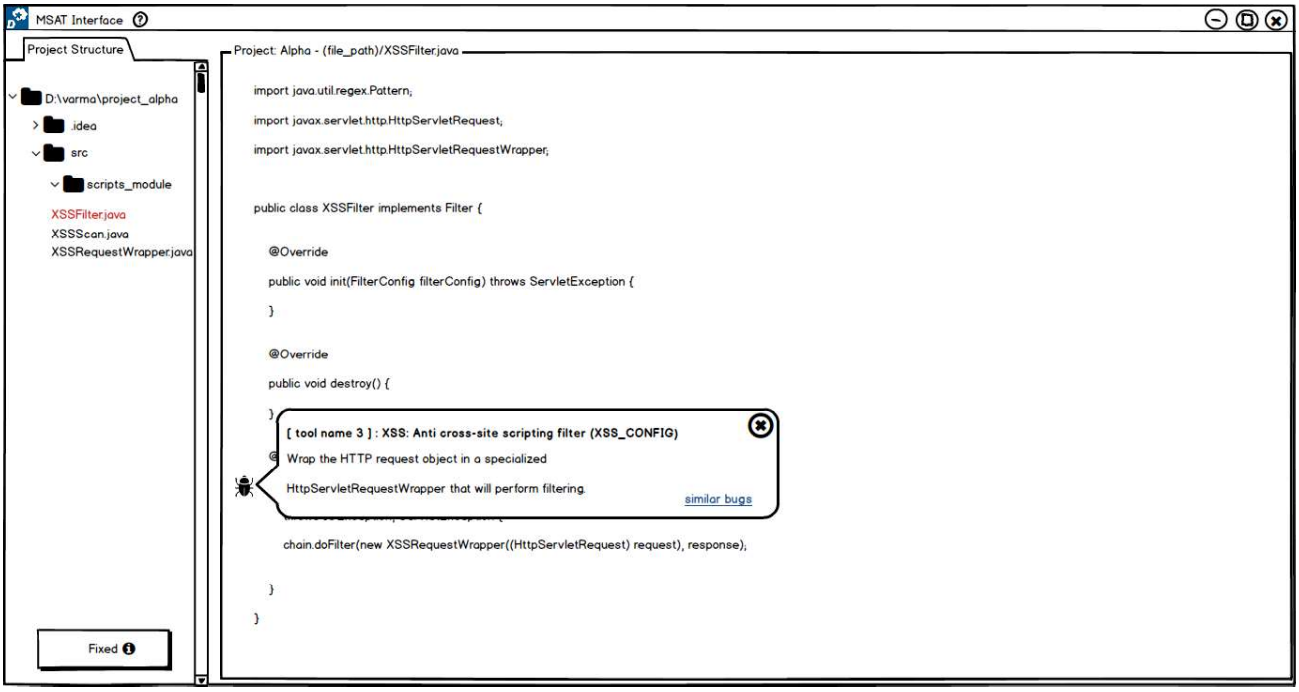
```
import java.util.regex.Pattern;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

    @Override

    public void init(FilterConfig filterConfig) throws ServletException {

    }

    @Override

    public void destroy() {

    }

    @Override

    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)

        throws IOException, ServletException {

        chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);


    }

}
```



Project: Alpha - (file_path)/XSSFilter.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

 @Override
 public void init(FilterConfig filterConfig) throws ServletException {
 }

 @Override
  public void destroy() {
  }

  @Override
  public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
       throws IOException, ServletException {
     chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
  }

}
```

S. No.	Line Number	Bug Name	Bug Type	Tool Name	Description
1	17	XSS: Anti cross-site scripting filter (XSS_CONFIG)	XSS	tool name 3	Wrap the HTTP request object ...
2	18	JSP reflected cross site scripting vulnerability	XSS	tool name 2	This code directly writes an HTTP parameter ...



Project: Alpha - (file_path)/XSSFilter.java

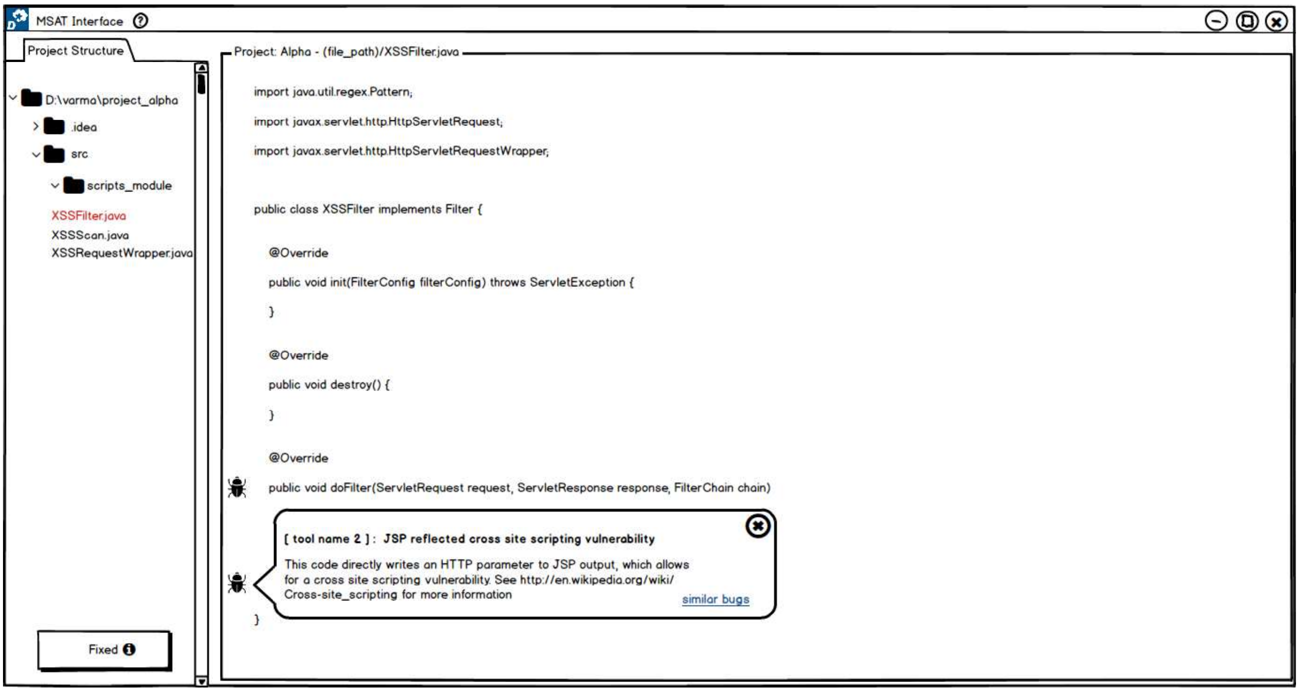
```
import java.util.regex.Pattern;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

    @Override

    public void init(FilterConfig filterConfig) throws ServletException {

    }

    @Override

    public void destroy() {

    }

        chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);


    }

}
```

[tool name 3] : XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized

HttpServletRequestWrapper that will perform filtering.

similar bugs

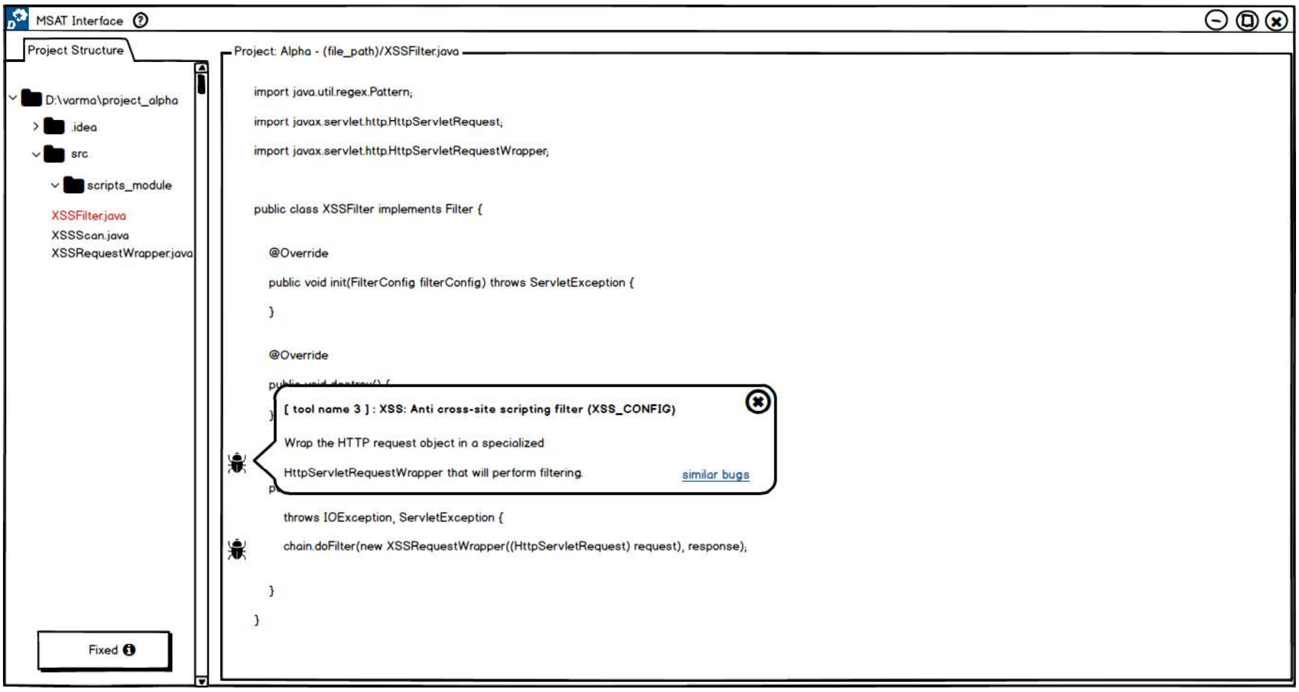


```
import java.util.regex.Pattern;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override

   public void init(FilterConfig filterConfig) throws ServletException {

   }

   @Override

   public void destroy() {

   }

   @Override

   public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)

}
```

[tool name 2] : JSP reflected cross site scripting vulnerability

This code directly writes an HTTP parameter to JSP output, which allows for a cross site scripting vulnerability. See http://en.wikipedia.org/wiki/Cross-site_scripting for more information

similar bugs



```
import java.util.regex.Pattern;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override

   public void init(FilterConfig filterConfig) throws ServletException {

   }

   @Override

   public void destroy() {

   }

   p

      throws IOException, ServletException {

      chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);

   }

}
```

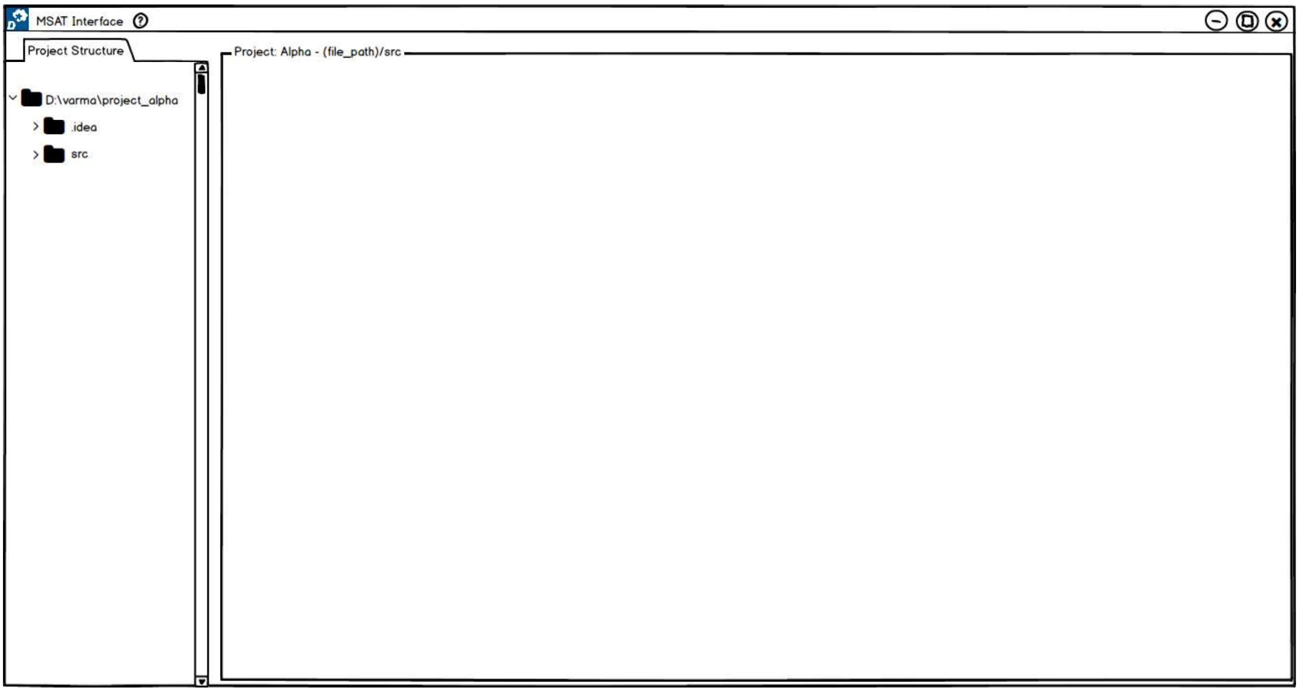
[tool name 3] : XSS: Anti cross-site scripting filter (XSS_CONFIG)

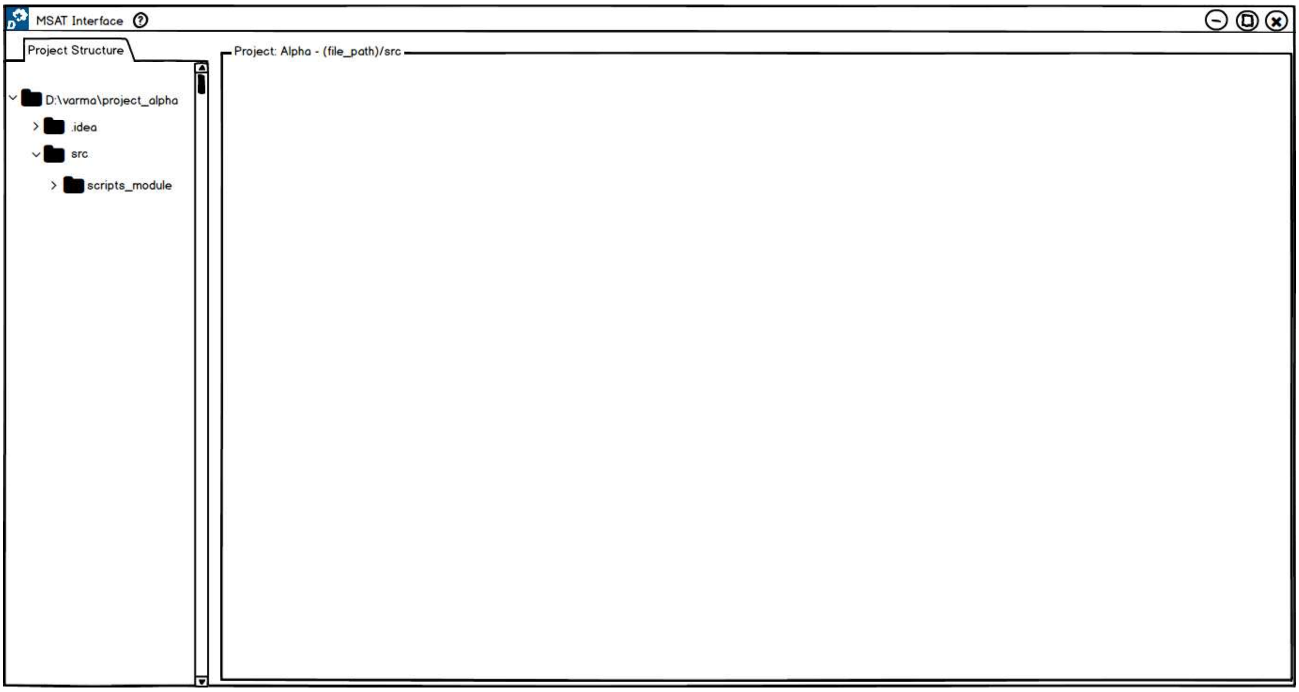
Wrap the HTTP request object in a specialized

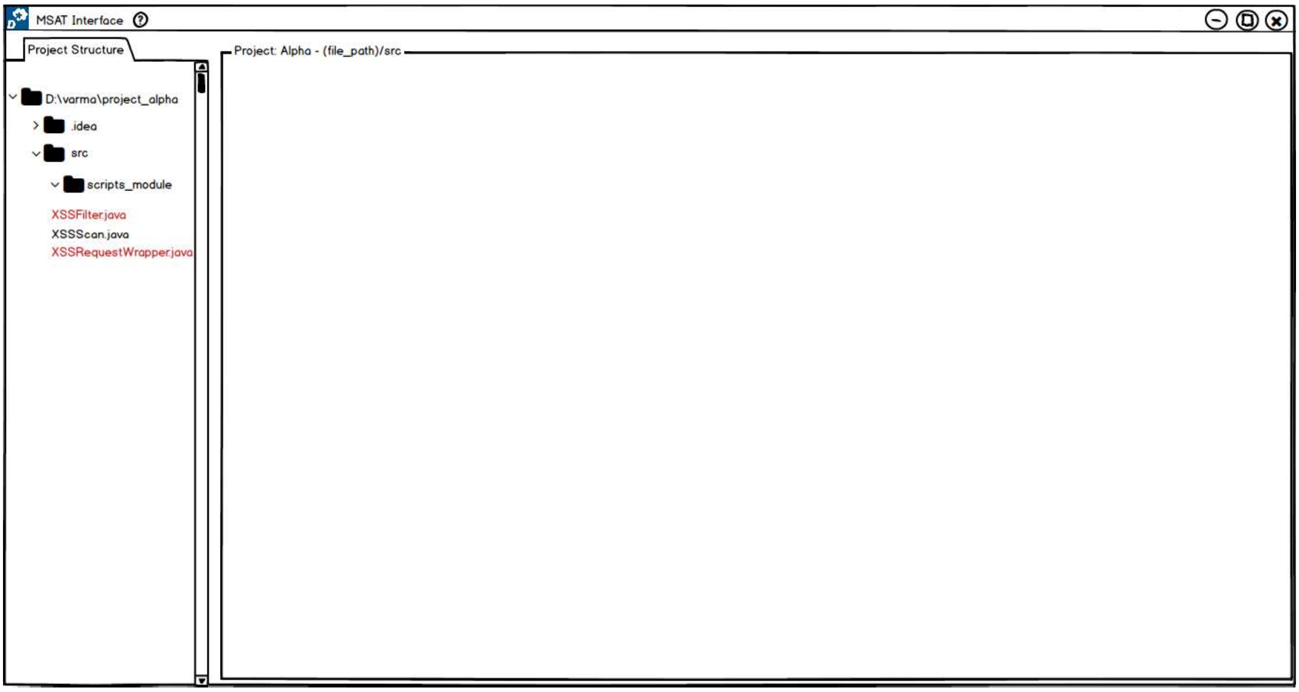
HttpServletRequestWrapper that will perform filtering.

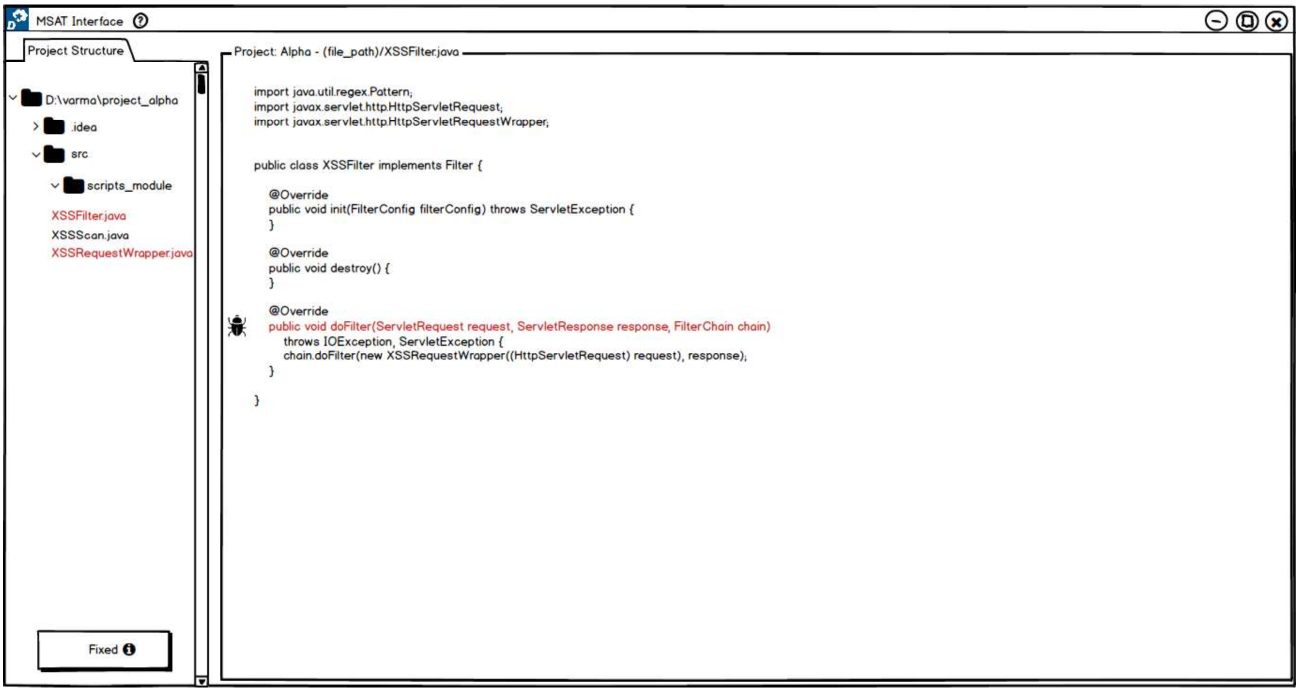
similar bugs

UXD Cycle 3 - RQ 3 - Multiple Window

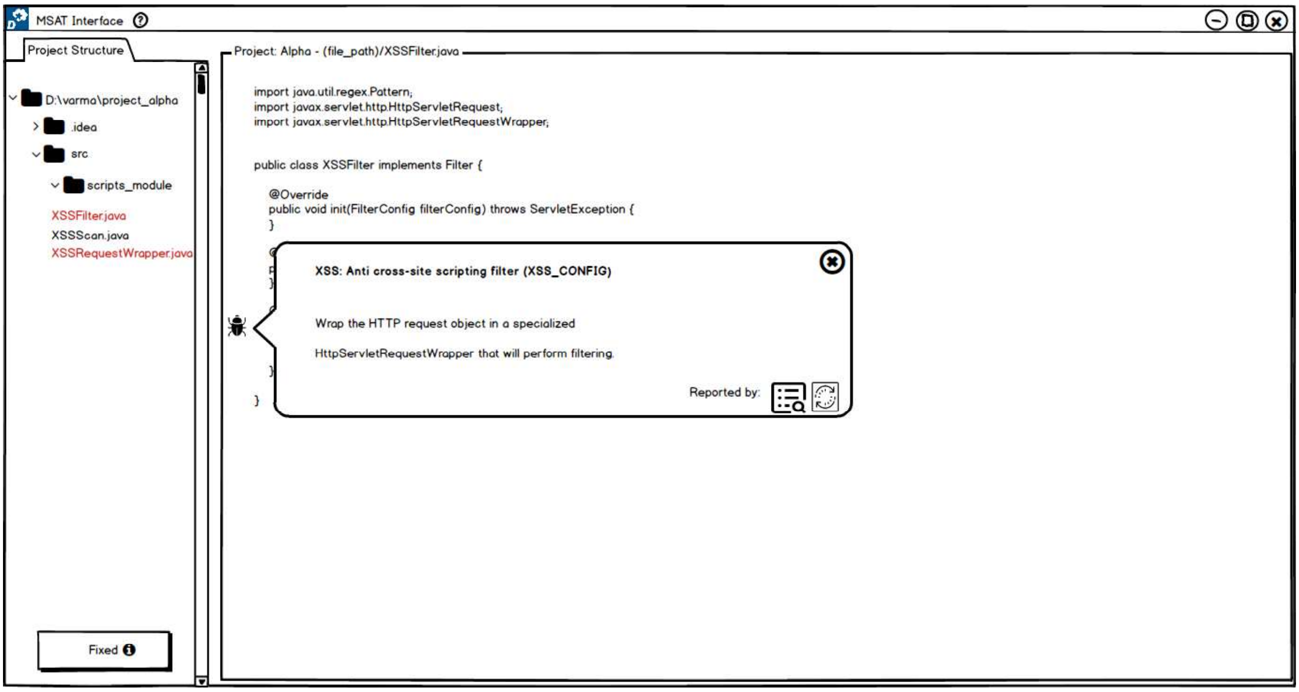








```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {
   }

   @Override
   public void destroy() {
   }

   @Override
   public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
      throws IOException, ServletException {
      chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
   }

}
```

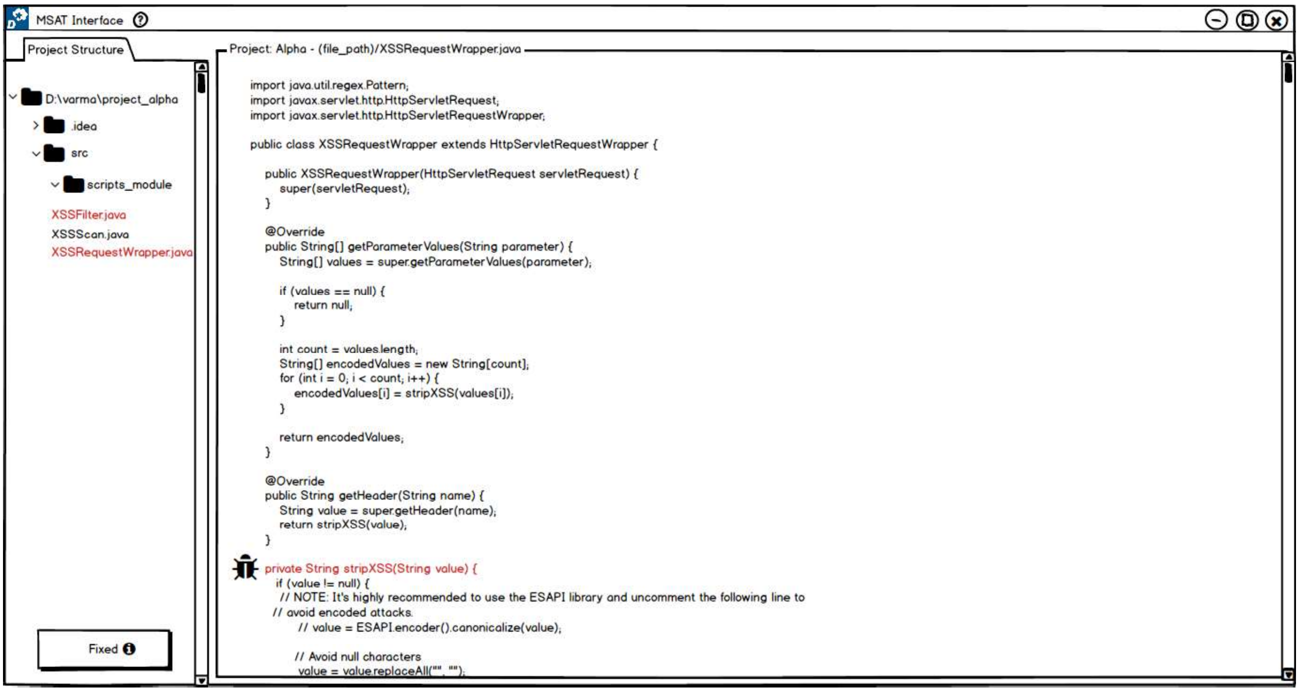


```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {
   }
```

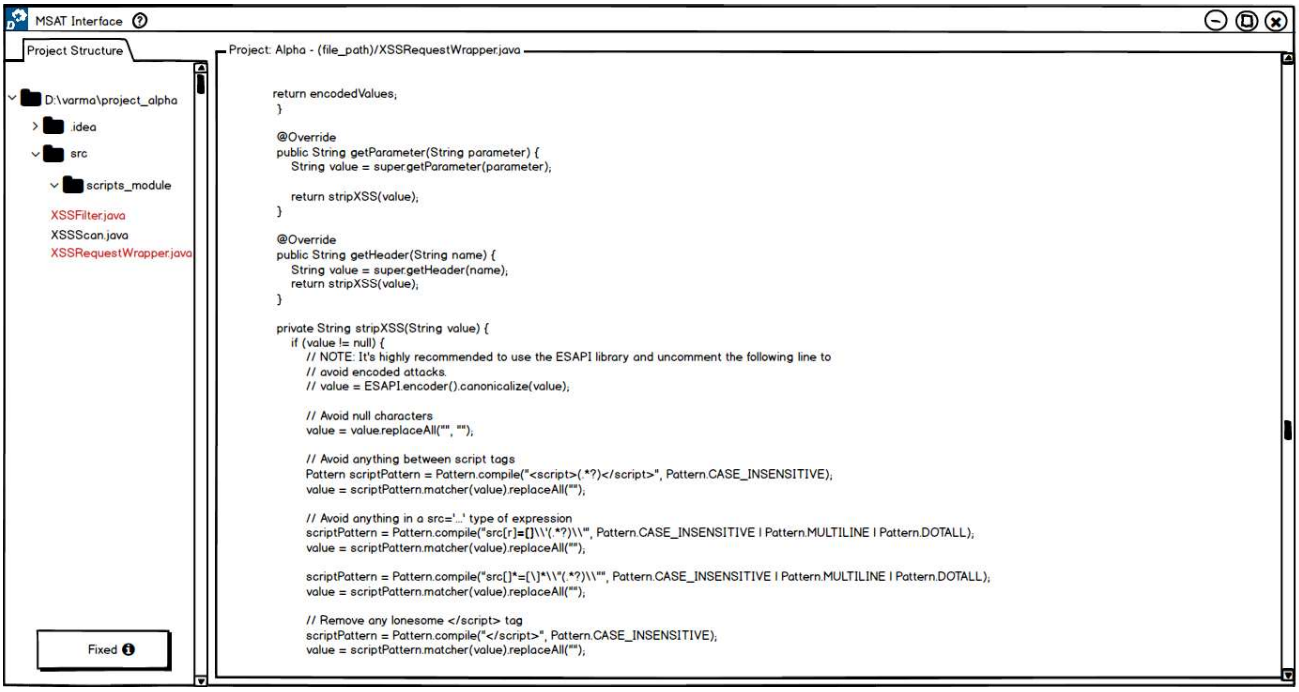
XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized

HttpServletRequestWrapper that will perform filtering.

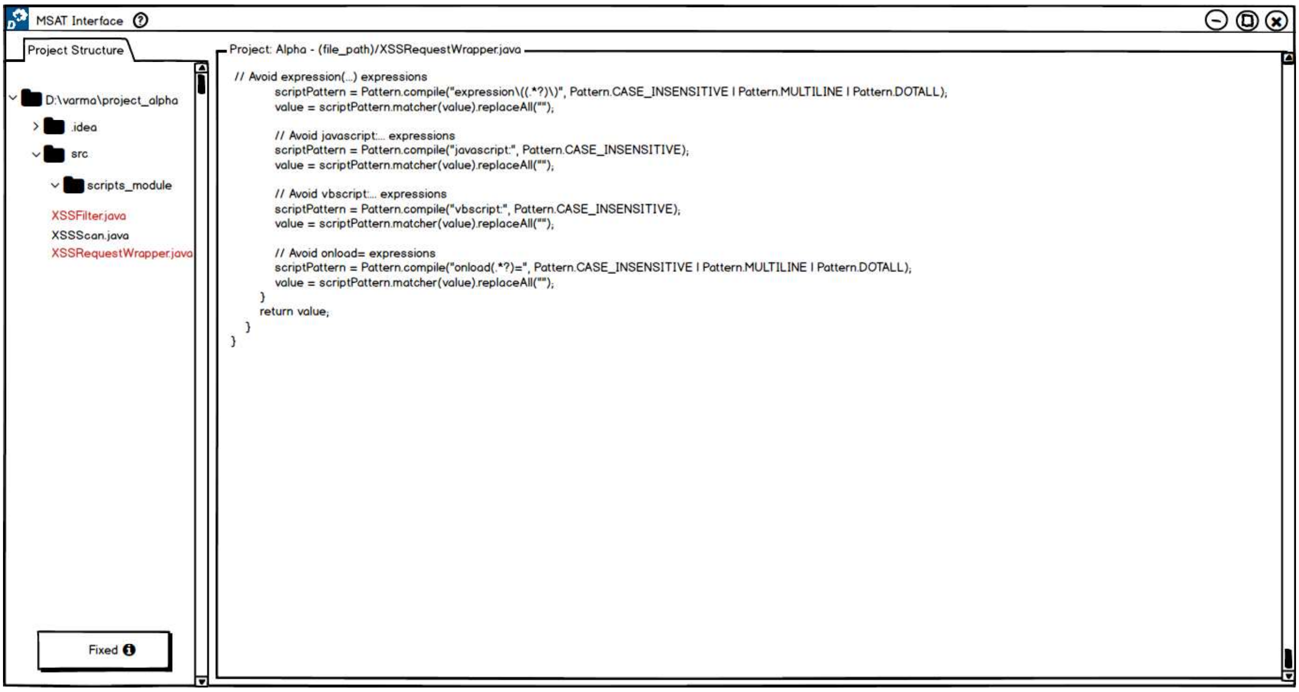


```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

   public XSSRequestWrapper(HttpServletRequest servletRequest) {
      super(servletRequest);
   }

   @Override
   public String[] getParameterValues(String parameter) {
      String[] values = super.getParameterValues(parameter);

      if (values == null) {
         return null;
      }

      int count = values.length;
      String[] encodedValues = new String[count];
      for (int i = 0; i < count; i++) {
         encodedValues[i] = stripXSS(values[i]);
      }

      return encodedValues;
   }

   @Override
   public String getHeader(String name) {
      String value = super.getHeader(name);
      return stripXSS(value);
   }

   private String stripXSS(String value) {
     if (value != null) {
      // NOTE: It's highly recommended to use the ESAPI library and uncomment the following line to
     // avoid encoded attacks.
         // value = ESAPI.encoder().canonicalize(value);

        // Avoid null characters
         value = value.replaceAll("", "");
```



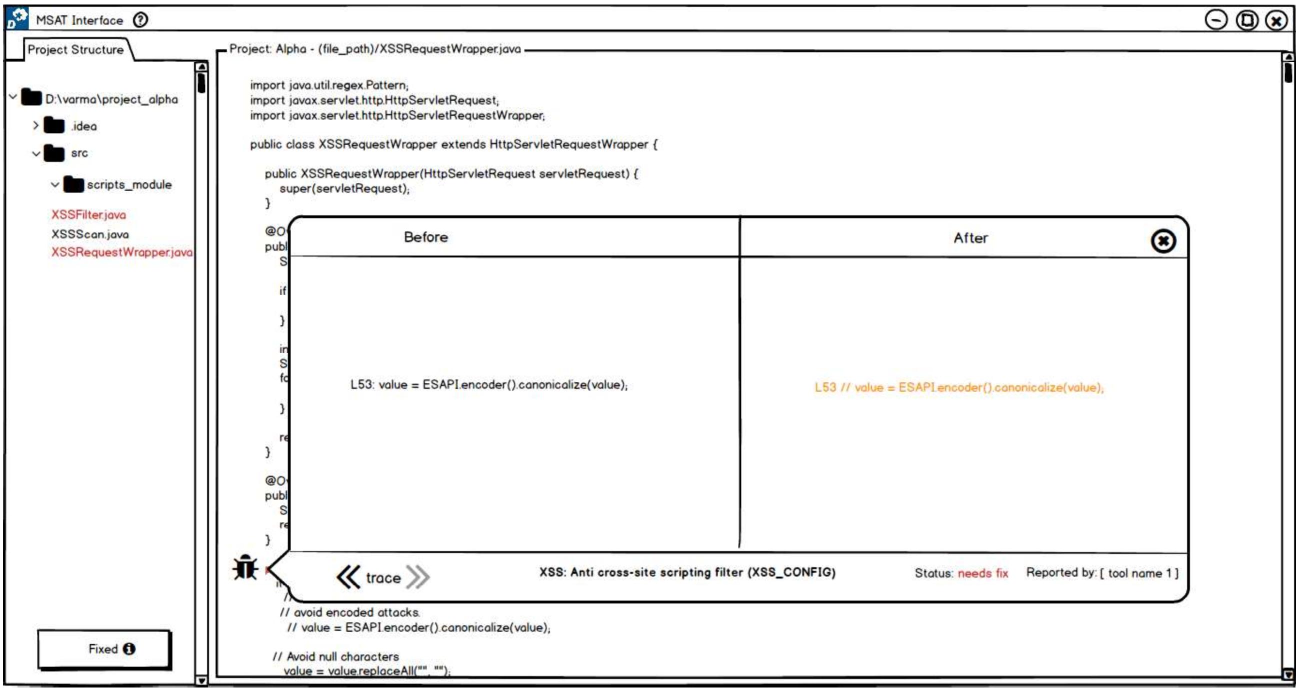
Project: Alpha - (file_path)/XSSRequestWrapper.java

```
return encodedValues;
 }

 @Override
 public String getParameter(String parameter) {
    String value = super.getParameter(parameter);

    return stripXSS(value);
 }

 @Override
 public String getHeader(String name) {
    String value = super.getHeader(name);
    return stripXSS(value);
 }

 private String stripXSS(String value) {
    if (value != null) {
       // NOTE: It's highly recommended to use the ESAPI library and uncomment the following line to
       // avoid encoded attacks.
       // value = ESAPI.encoder().canonicalize(value);

       // Avoid null characters
       value = value.replaceAll("", "");

       // Avoid anything between script tags
       Pattern scriptPattern = Pattern.compile("<script>(.*?)</script>", Pattern.CASE_INSENSITIVE);
       value = scriptPattern.matcher(value).replaceAll("");

       // Avoid anything in a src='...' type of expression
       scriptPattern = Pattern.compile("src[r]=[]\\'(.*?)\\'", Pattern.CASE_INSENSITIVE | Pattern.MULTILINE | Pattern.DOTALL);
       value = scriptPattern.matcher(value).replaceAll("");

       scriptPattern = Pattern.compile("src[]*=[\]*\\"(.*?)\\"", Pattern.CASE_INSENSITIVE | Pattern.MULTILINE | Pattern.DOTALL);
       value = scriptPattern.matcher(value).replaceAll("");

       // Remove any lonesome </script> tag
       scriptPattern = Pattern.compile("</script>", Pattern.CASE_INSENSITIVE);
       value = scriptPattern.matcher(value).replaceAll("");
```



Project: Alpha - (file_path)/XSSRequestWrapper.java

```
// Avoid expression(...) expressions
         scriptPattern = Pattern.compile("expression\((.*?)\)", Pattern.CASE_INSENSITIVE | Pattern.MULTILINE | Pattern.DOTALL);
         value = scriptPattern.matcher(value).replaceAll("");

         // Avoid javascript:... expressions
         scriptPattern = Pattern.compile("javascript:", Pattern.CASE_INSENSITIVE);
         value = scriptPattern.matcher(value).replaceAll("");

         // Avoid vbscript:... expressions
         scriptPattern = Pattern.compile("vbscript:", Pattern.CASE_INSENSITIVE);
         value = scriptPattern.matcher(value).replaceAll("");

         // Avoid onload= expressions
         scriptPattern = Pattern.compile("onload(.*?)=", Pattern.CASE_INSENSITIVE | Pattern.MULTILINE | Pattern.DOTALL);
         value = scriptPattern.matcher(value).replaceAll("");
      }
      return value;
   }
}
```

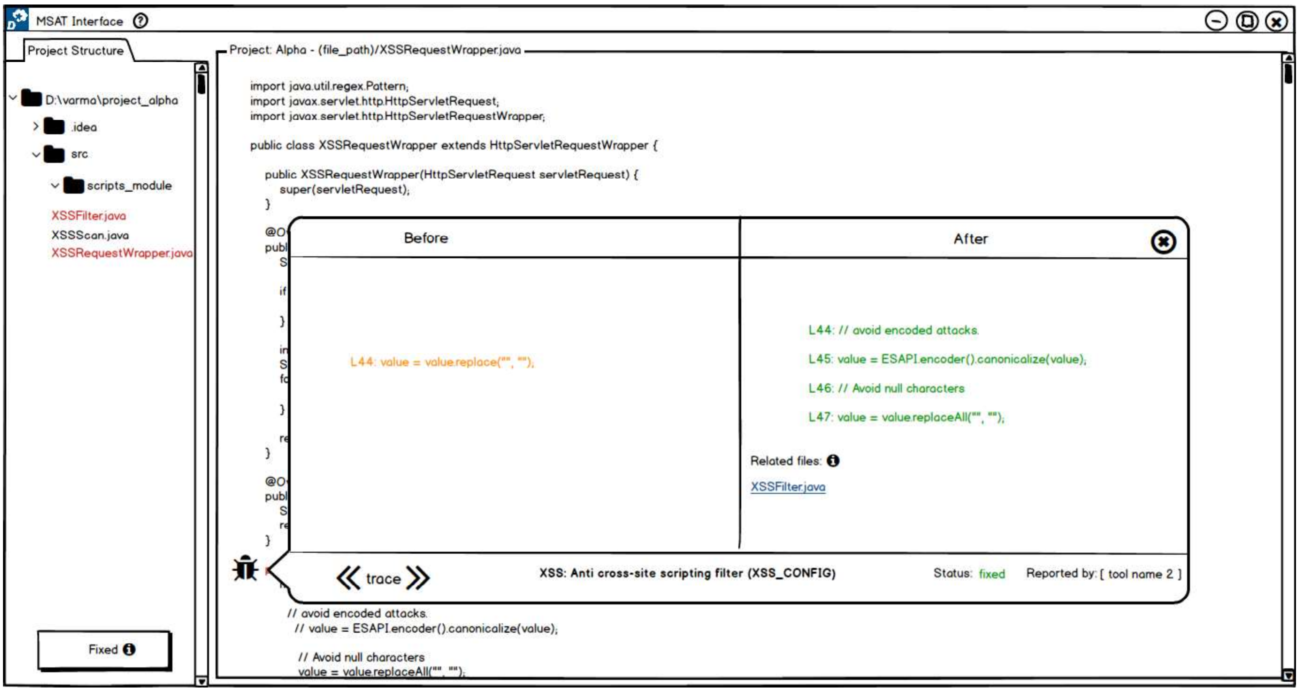


Project: Alpha - (file_path)/XSSRequestWrapper.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

   public XSSRequestWrapper(HttpServletRequest servletRequest) {
      super(servletRequest);
   }

   @O
   publ
      S

      if

      }

      in
      S
      fo

      }

      re
   }

   @O
   publ
      S
      re
   }

      // avoid encoded attacks.
       // value = ESAPI.encoder().canonicalize(value);

     // Avoid null characters
       value = value.replaceAll("", "");
```

Before	After
L53: value = ESAPI.encoder().canonicalize(value);	L53 // value = ESAPI.encoder().canonicalize(value);

XSS: Anti cross-site scripting filter (XSS_CONFIG)   Status: needs fix   Reported by: [tool name 1]



Project: Alpha - (file_path)/XSSRequestWrapper.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

   public XSSRequestWrapper(HttpServletRequest servletRequest) {
      super(servletRequest);
   }
```

Before	After

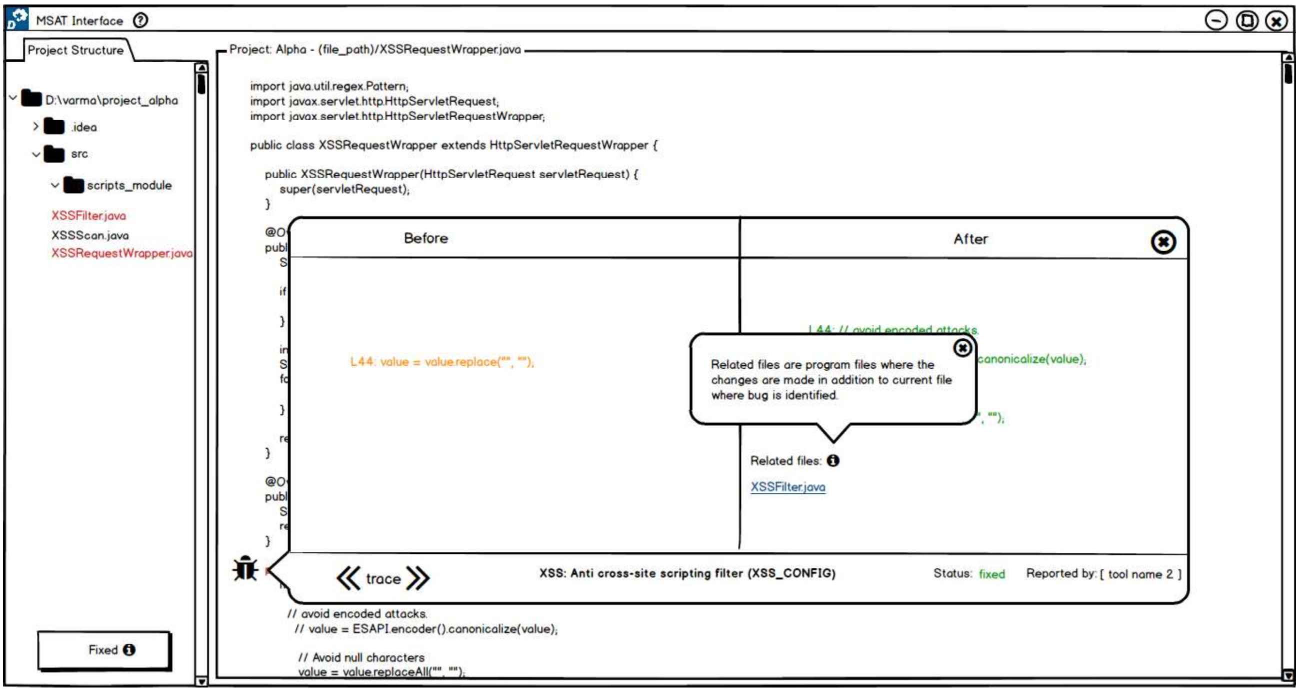
L44: value = value.replace("", "");	L44: // avoid encoded attacks.
	L45: value = ESAPI.encoder().canonicalize(value);
	L46: // Avoid null characters
	L47: value = value.replaceAll("", "");

Related files:

XSSFilter.java

XSS: Anti cross-site scripting filter (XSS_CONFIG) Status: fixed Reported by: [tool name 2]

```
// avoid encoded attacks.
  // value = ESAPI.encoder().canonicalize(value);

  // Avoid null characters
  value = value.replaceAll("", "");
```



Project: Alpha - (file_path)/XSSRequestWrapper.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

   public XSSRequestWrapper(HttpServletRequest servletRequest) {
      super(servletRequest);
   }
```

Before	After

L44: value = value.replace("", "");	L44: // avoid encoded attacks.
	canonicalize(value);
	", "");

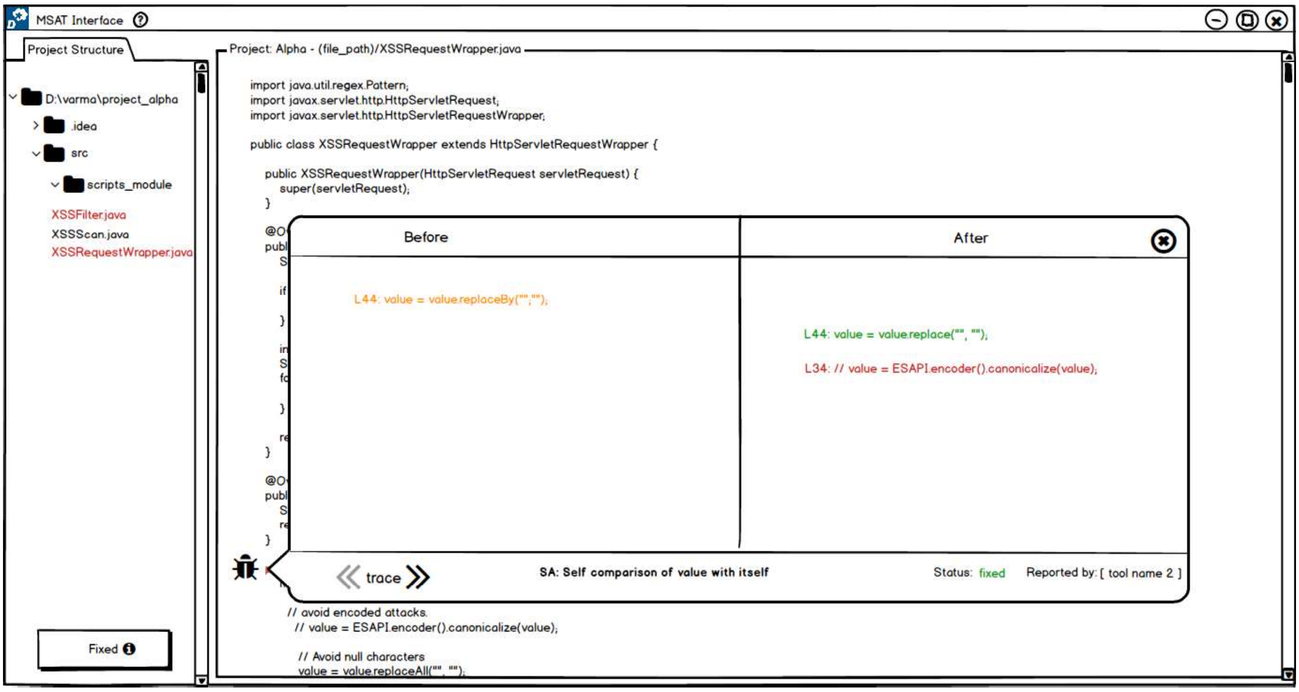
Related files are program files where the changes are made in addition to current file where bug is identified.

Related files:

XSSFilter.java

XSS: Anti cross-site scripting filter (XSS_CONFIG) Status: fixed Reported by: [tool name 2]

```
// avoid encoded attacks.
  // value = ESAPI.encoder().canonicalize(value);

  // Avoid null characters
  value = value.replaceAll("", "");
```



Project: Alpha - (file_path)/XSSRequestWrapper.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

   public XSSRequestWrapper(HttpServletRequest servletRequest) {
      super(servletRequest);
   }
```

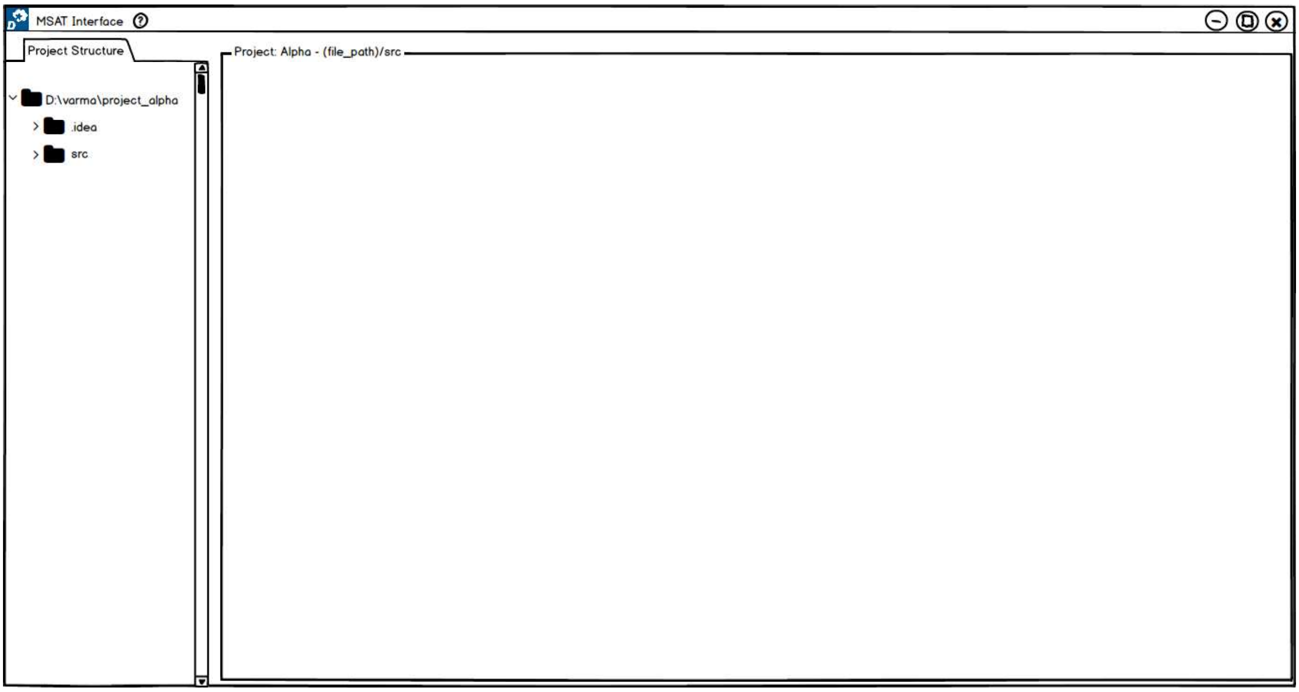
Before	After

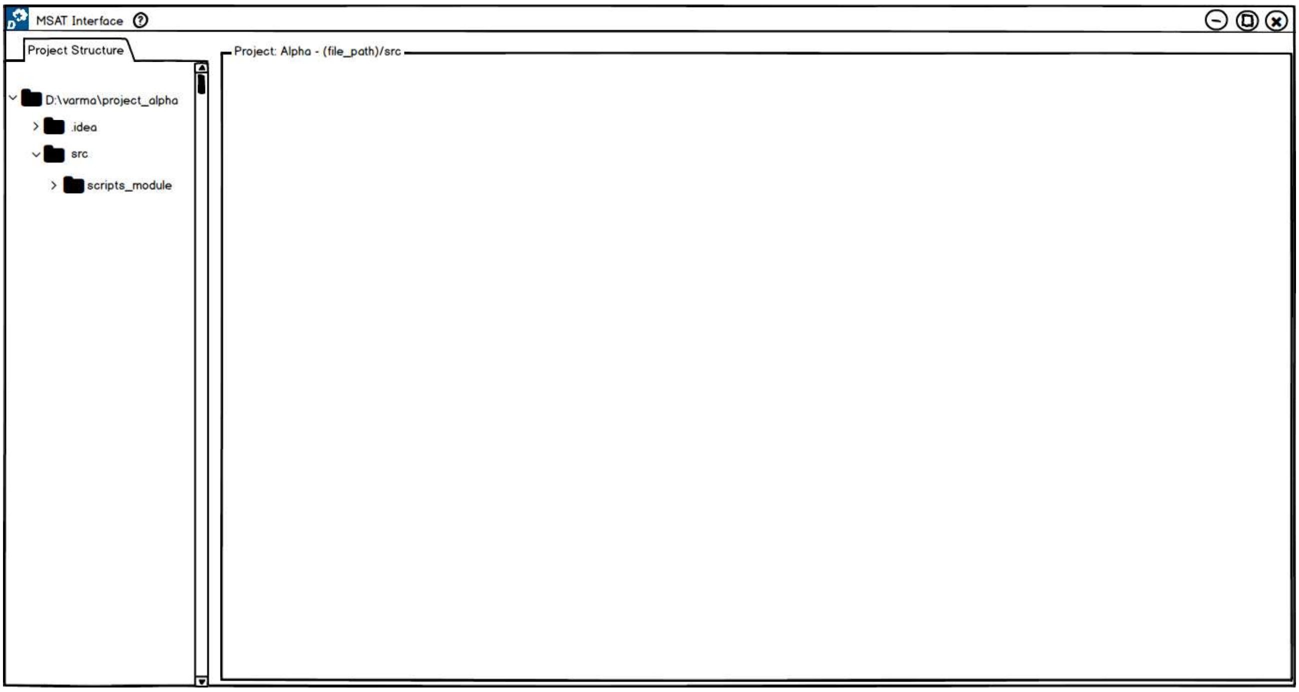
L44: value = value.replaceBy("","");	L44: value = value.replace("", "");
	L34: // value = ESAPI.encoder().canonicalize(value);

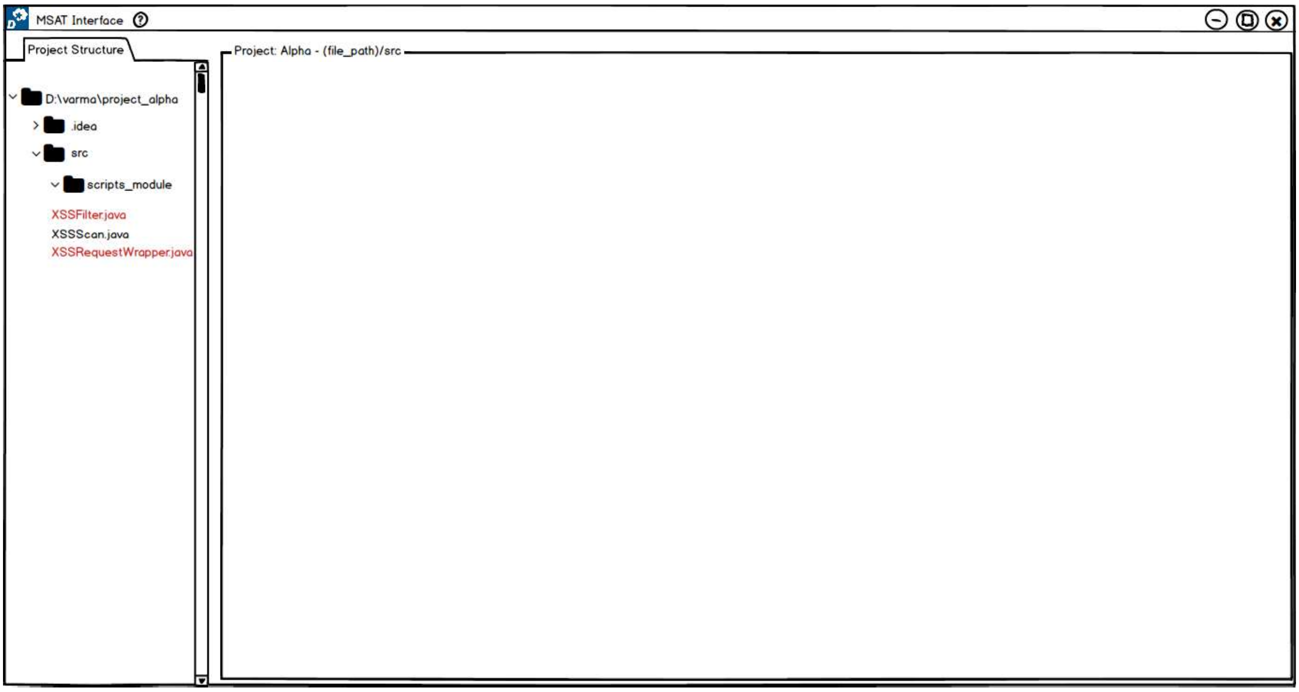
SA: Self comparison of value with itself Status: fixed Reported by: [tool name 2]

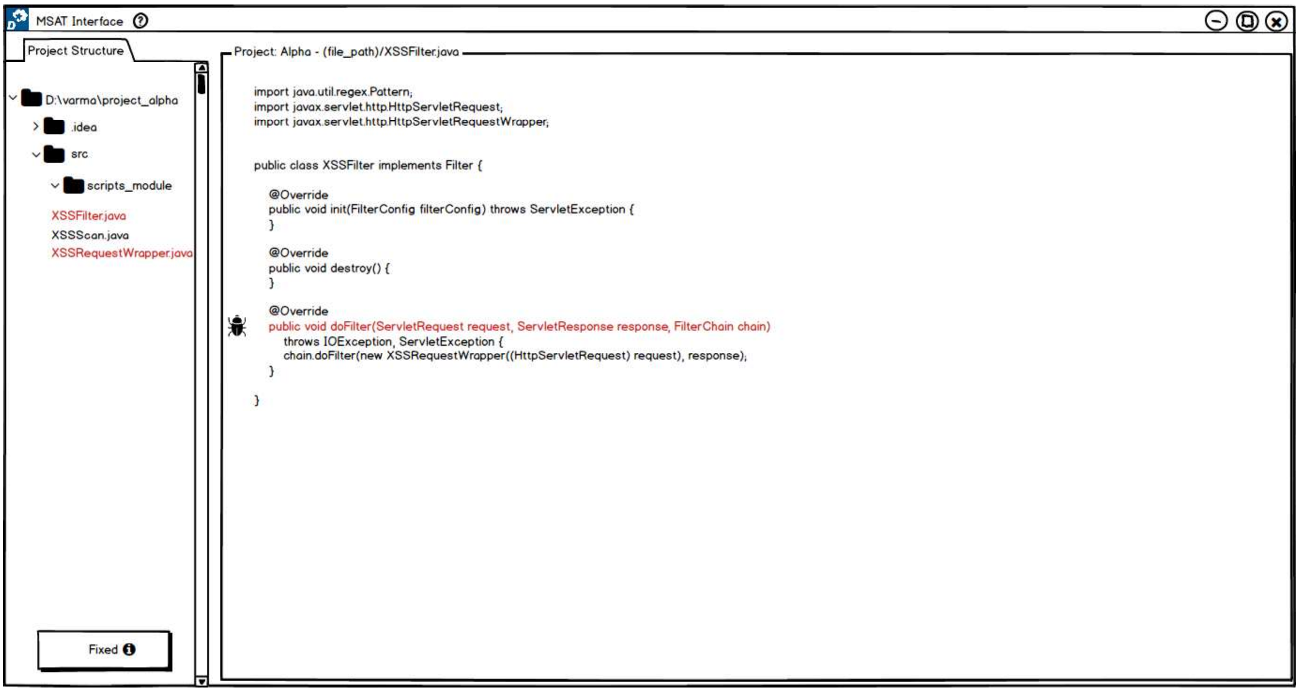
```
// avoid encoded attacks.
  // value = ESAPI.encoder().canonicalize(value);

  // Avoid null characters
  value = value.replaceAll("", "");
```

UXD Cycle 3 - RQ 3 - Table View

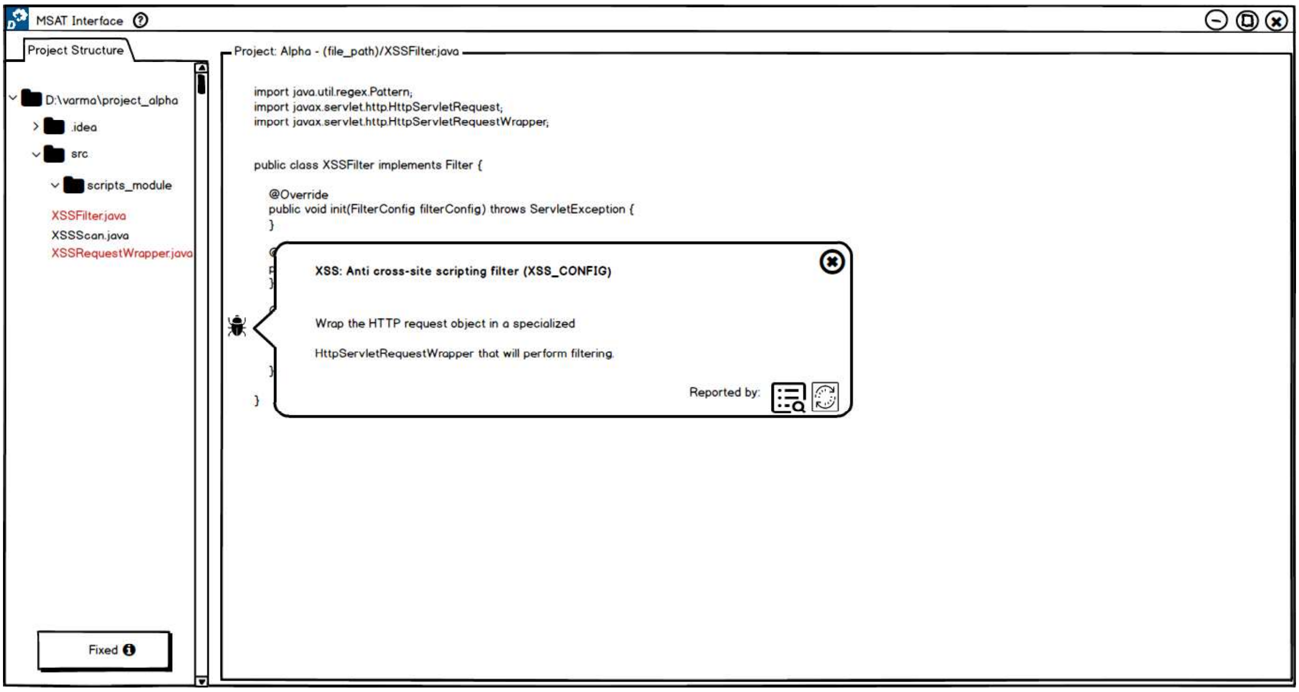








```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {
   }

   @Override
   public void destroy() {
   }

   @Override
   public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
      throws IOException, ServletException {
      chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
   }

}
```



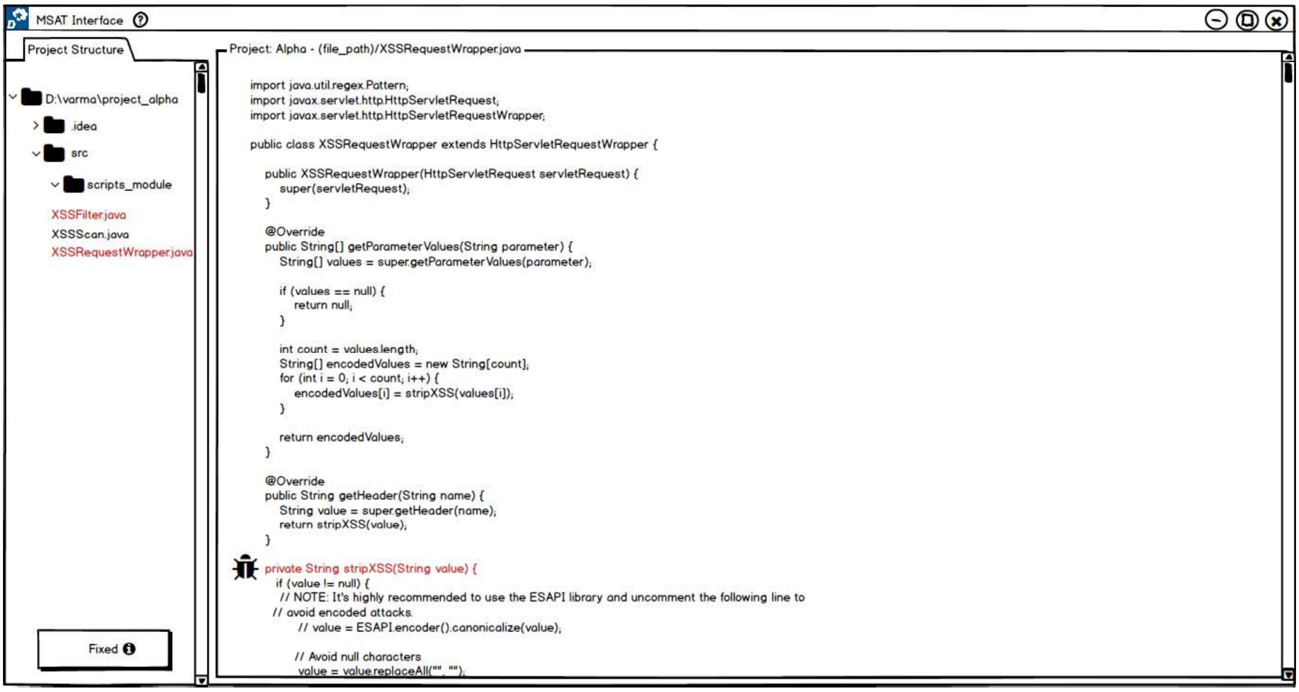
```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

   @Override
   public void init(FilterConfig filterConfig) throws ServletException {
   }

}
```

XSS: Anti cross-site scripting filter (XSS_CONFIG)

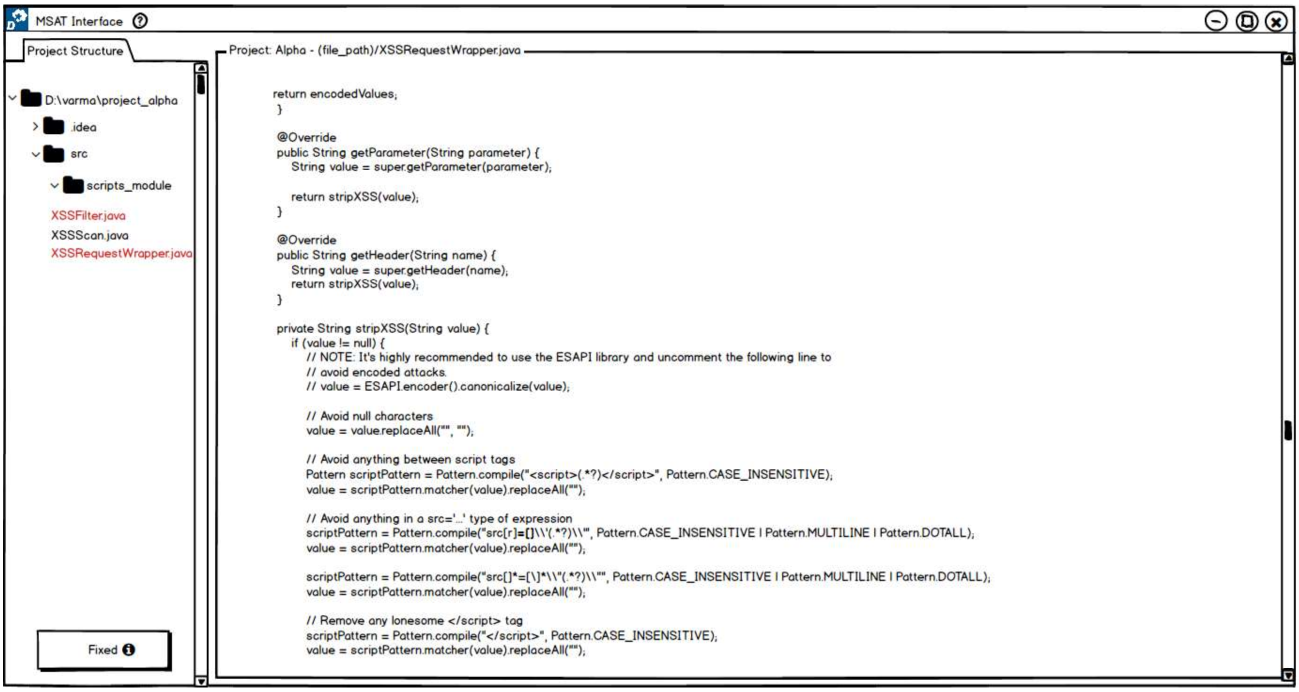
Wrap the HTTP request object in a specialized

HttpServletRequestWrapper that will perform filtering.

Reported by:

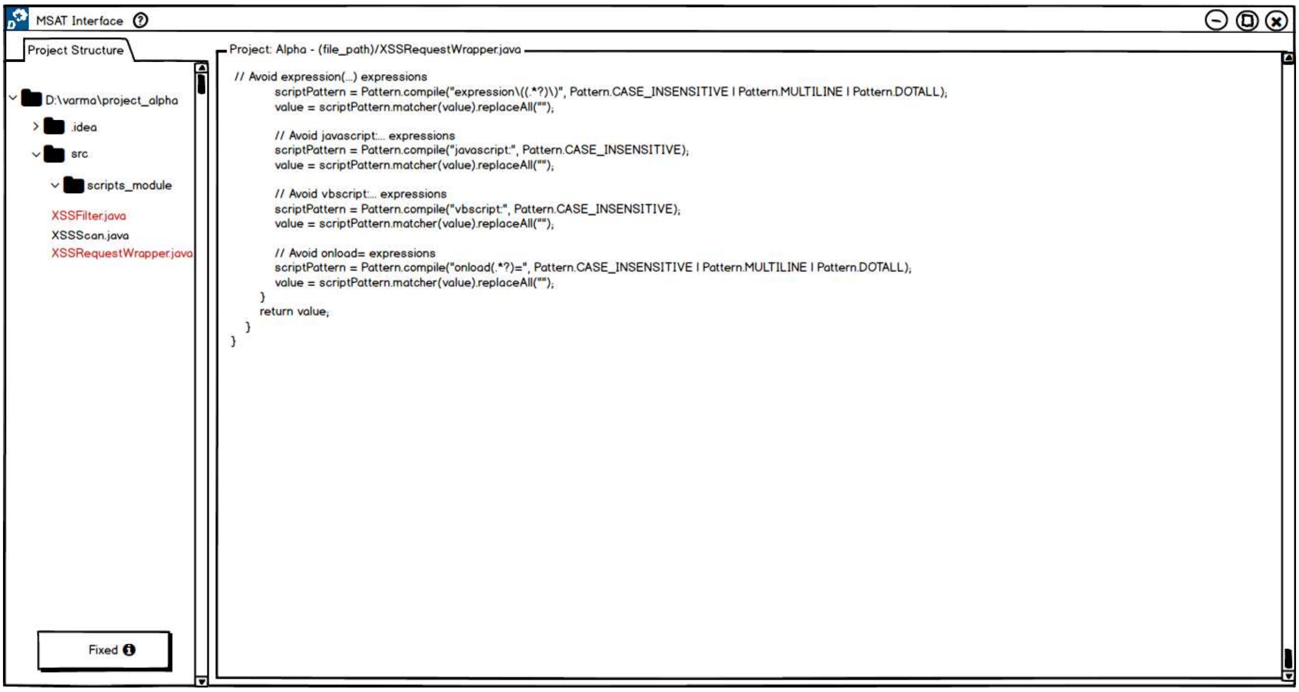


```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

   public XSSRequestWrapper(HttpServletRequest servletRequest) {
      super(servletRequest);
   }

   @Override
   public String[] getParameterValues(String parameter) {
      String[] values = super.getParameterValues(parameter);

      if (values == null) {
         return null;
      }

      int count = values.length;
      String[] encodedValues = new String[count];
      for (int i = 0; i < count; i++) {
         encodedValues[i] = stripXSS(values[i]);
      }

      return encodedValues;
   }

   @Override
   public String getHeader(String name) {
      String value = super.getHeader(name);
      return stripXSS(value);
   }

   private String stripXSS(String value) {
     if (value != null) {
      // NOTE: It's highly recommended to use the ESAPI library and uncomment the following line to
     // avoid encoded attacks.
         // value = ESAPI.encoder().canonicalize(value);

         // Avoid null characters
         value = value.replaceAll("", "");
```



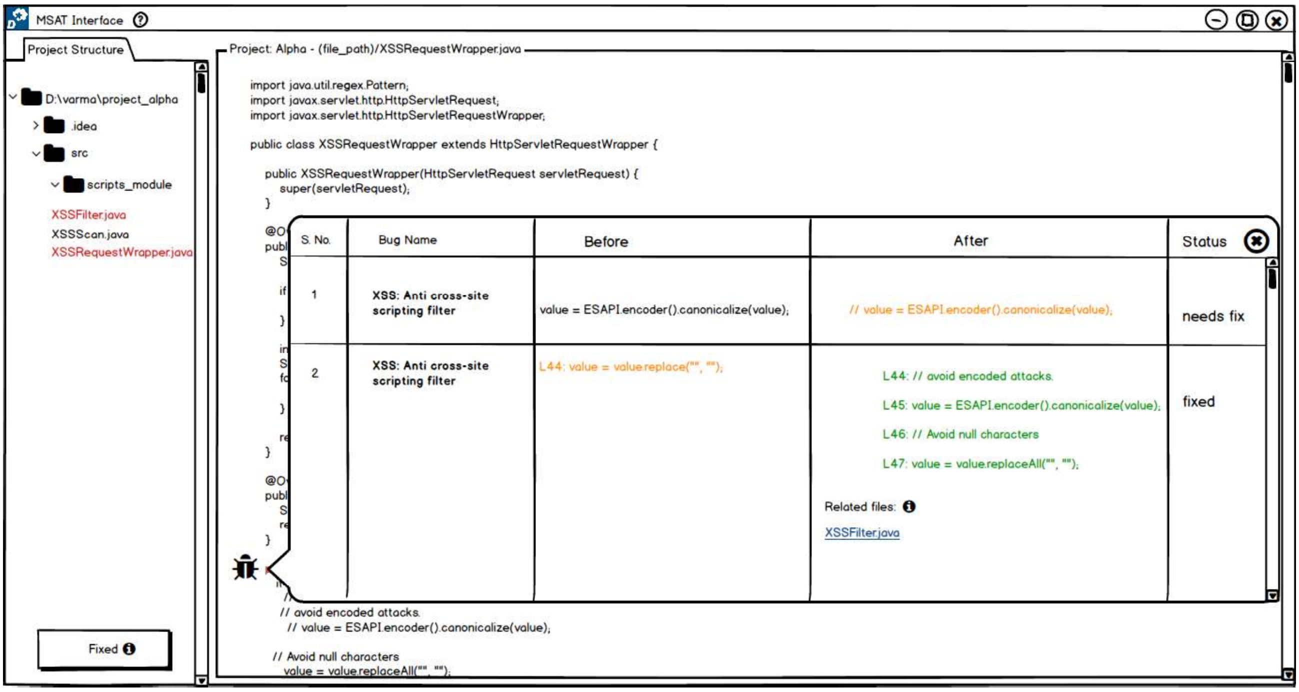
Project: Alpha - (file_path)/XSSRequestWrapper.java

```
return encodedValues;
 }

 @Override
 public String getParameter(String parameter) {
    String value = super.getParameter(parameter);

    return stripXSS(value);
 }

 @Override
 public String getHeader(String name) {
    String value = super.getHeader(name);
    return stripXSS(value);
 }

 private String stripXSS(String value) {
    if (value != null) {
        // NOTE: It's highly recommended to use the ESAPI library and uncomment the following line to
        // avoid encoded attacks.
        // value = ESAPI.encoder().canonicalize(value);

        // Avoid null characters
        value = value.replaceAll("", "");

        // Avoid anything between script tags
        Pattern scriptPattern = Pattern.compile("<script>(.*?)</script>", Pattern.CASE_INSENSITIVE);
        value = scriptPattern.matcher(value).replaceAll("");

        // Avoid anything in a src='...' type of expression
        scriptPattern = Pattern.compile("src[r]=[]\\'(.*?)\\'", Pattern.CASE_INSENSITIVE I Pattern.MULTILINE I Pattern.DOTALL);
        value = scriptPattern.matcher(value).replaceAll("");

        scriptPattern = Pattern.compile("src[]*=[\]*\\"(.*?)\\"", Pattern.CASE_INSENSITIVE I Pattern.MULTILINE I Pattern.DOTALL);
        value = scriptPattern.matcher(value).replaceAll("");

        // Remove any lonesome </script> tag
        scriptPattern = Pattern.compile("</script>", Pattern.CASE_INSENSITIVE);
        value = scriptPattern.matcher(value).replaceAll("");
```



Project: Alpha - (file_path)/XSSRequestWrapper.java

```
// Avoid expression(...) expressions
        scriptPattern = Pattern.compile("expression\((.*?)\)", Pattern.CASE_INSENSITIVE I Pattern.MULTILINE I Pattern.DOTALL);
        value = scriptPattern.matcher(value).replaceAll("");

        // Avoid javascript:... expressions
        scriptPattern = Pattern.compile("javascript:", Pattern.CASE_INSENSITIVE);
        value = scriptPattern.matcher(value).replaceAll("");

        // Avoid vbscript:... expressions
        scriptPattern = Pattern.compile("vbscript:", Pattern.CASE_INSENSITIVE);
        value = scriptPattern.matcher(value).replaceAll("");

        // Avoid onload= expressions
        scriptPattern = Pattern.compile("onload(.*?)=", Pattern.CASE_INSENSITIVE I Pattern.MULTILINE I Pattern.DOTALL);
        value = scriptPattern.matcher(value).replaceAll("");
    }
    return value;
  }
}
```

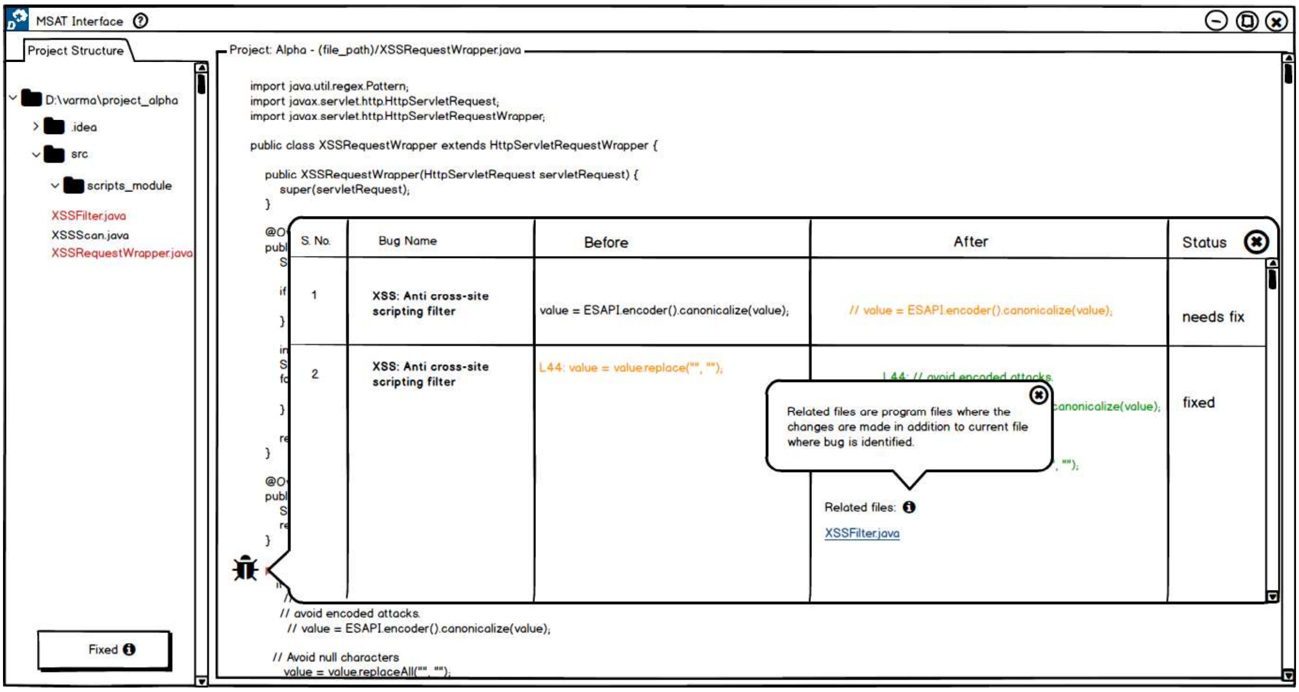


Project: Alpha - (file_path)/XSSRequestWrapper.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

   public XSSRequestWrapper(HttpServletRequest servletRequest) {
      super(servletRequest);
   }
```

S. No.	Bug Name	Before	After	Status
1	**XSS: Anti cross-site scripting filter**	value = ESAPI.encoder().canonicalize(value);	// value = ESAPI.encoder().canonicalize(value);	needs fix
2	**XSS: Anti cross-site scripting filter**	L44: value = value.replace("", "");	L44: // avoid encoded attacks. L45: value = ESAPI.encoder().canonicalize(value); L46: // Avoid null characters L47: value = value.replaceAll("", ""); Related files: ℹ XSSFilter.java	fixed

```
      // avoid encoded attacks.
       // value = ESAPI.encoder().canonicalize(value);

     // Avoid null characters
       value = value.replaceAll("", "");
```



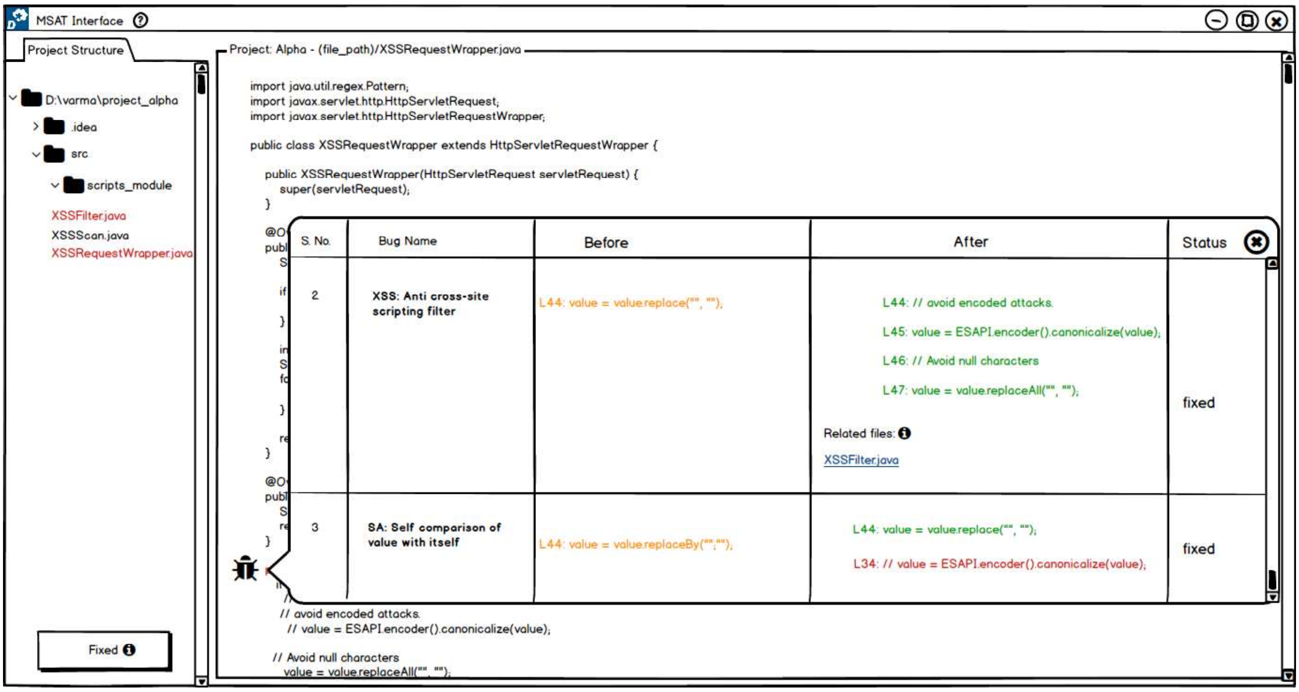
Project: Alpha - (file_path)/XSSRequestWrapper.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

    public XSSRequestWrapper(HttpServletRequest servletRequest) {
        super(servletRequest);
    }

    @O
    publ
        S

        if

        }

        in
        S
        fo

        }

        re
    }

    @O
    publ
        S
        re
    }

        //
      // avoid encoded attacks.
        // value = ESAPI.encoder().canonicalize(value);

    // Avoid null characters
        value = value.replaceAll("", "");
```

S. No.	Bug Name	Before	After	Status
1	XSS: Anti cross-site scripting filter	value = ESAPI.encoder().canonicalize(value);	// value = ESAPI.encoder().canonicalize(value);	needs fix

2	XSS: Anti cross-site scripting filter	L44: value = value.replace("", "");	L44: // avoid encoded attacks.
			canonicalize(value);
			, "");
			Related files:
			XSSFilter.java
				fixed

Related files are program files where the changes are made in addition to current file where bug is identified.

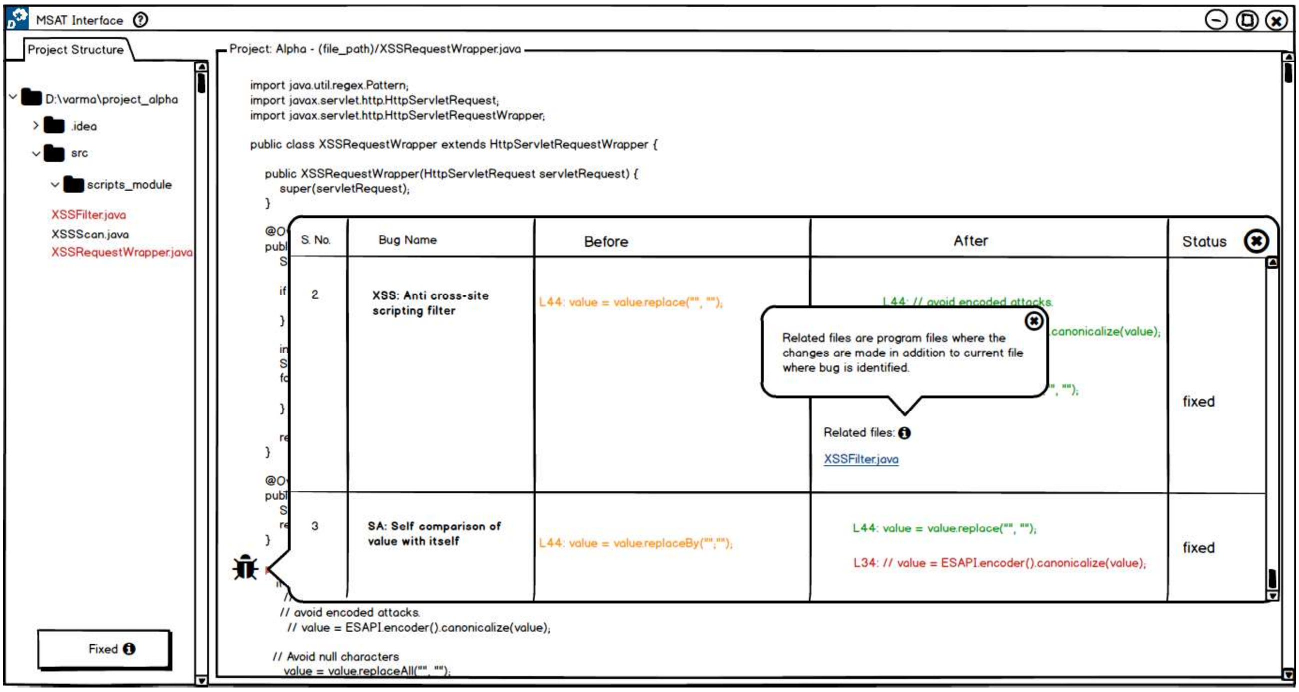


Project: Alpha - (file_path)/XSSRequestWrapper.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

    public XSSRequestWrapper(HttpServletRequest servletRequest) {
        super(servletRequest);
    }

    @O
    publ
        S

        if

        }

        in
        S
        fo

        }

        re
    }

    @O
    publ
        S
        re
    }

        //
      // avoid encoded attacks.
        // value = ESAPI.encoder().canonicalize(value);

    // Avoid null characters
        value = value.replaceAll("", "");
```

S. No.	Bug Name	Before	After	Status
2	XSS: Anti cross-site scripting filter	L44: value = value.replace("", "");	L44: // avoid encoded attacks. L45: value = ESAPI.encoder().canonicalize(value); L46: // Avoid null characters L47: value = value.replaceAll("", ""); Related files: XSSFilter.java	fixed

3	SA: Self comparison of value with itself	L44: value = value.replaceBy("","");	L44: value = value.replace("", "");
			L34: // value = ESAPI.encoder().canonicalize(value);
				fixed



Project: Alpha - (file_path)/XSSRequestWrapper.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

    public XSSRequestWrapper(HttpServletRequest servletRequest) {
        super(servletRequest);
    }

    @O
    publ
        S

        if

        }

        in
        S
        fo

        }

        re
    }

    @O
    publ
        S
        re
    }

        //
      // avoid encoded attacks.
        // value = ESAPI.encoder().canonicalize(value);

    // Avoid null characters
        value = value.replaceAll("", "");
```

S. No.	Bug Name	Before	After	Status

2	XSS: Anti cross-site scripting filter	L44: value = value.replace("", "");	L44: // avoid encoded attacks.
			canonicalize(value);
			, "");
			Related files:
			XSSFilter.java
				fixed

3	SA: Self comparison of value with itself	L44: value = value.replaceBy("","");	L44: value = value.replace("", "");
			L34: // value = ESAPI.encoder().canonicalize(value);
				fixed

Related files are program files where the changes are made in addition to current file where bug is identified.