

Design Document

Jens Henninger Daniel Maier Paul Fink
 Florian Jennewein

January 7, 2016

Contents

I	Low-Level-Design	1
1	Introduction	3
2	The Client-side Application	5
3	The Server Application	7
3.1	Complete View	7
3.2	Detail View	8
3.2.1	The Main Servlet	8
3.2.2	The User Section	9
3.2.3	The Data Section	13
3.3	Used Patterns	15
3.4	Resulting Source Code	16
3.4.1	AlgorithmHandler	16
3.4.2	Converter	17
3.4.3	DataAccessor	19
3.4.4	DataHandler	21
3.4.5	DataManager	21
3.4.6	DataSet	24
3.4.7	DeleteHandler	24
3.4.8	DeleteUserHandler	25
3.4.9	DownloadHandler	27
3.4.10	HttpServlet	28
3.4.11	IAlgorithm	29
3.4.12	IFormat	29
3.4.13	LoginHandler	30
3.4.14	LogoutHandler	31
3.4.15	MainServlet	32
3.4.16	Model	34
3.4.17	NewAlgorithm	35
3.4.18	NewFormat	36
3.4.19	Packet	37
3.4.20	RegistrationHandler	37
3.4.21	ReqHandler	38

3.4.22	RestHandler	39
3.4.23	SearchUserHandler	41
3.4.24	UploadHandler	42
3.4.25	User	43
3.4.26	UserAreaHandler	43
3.4.27	UserGroup	45
3.4.28	UserHandler	46
3.4.29	UserManager	47

Part I

Low-Level-Design

Chapter 1

Introduction

This design document describes the low-level-Design of the desired system "Sojabohne" through class- and sequence- diagrams and also includes the basic source code of the application. The whole document is based on the architectural design document: "https://moodle.uni-mainz.de/pluginfile.php/33758/mod_resource/content/1/architectural_design.pdf"

Chapter 2

The Client-side Application

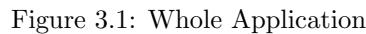
Because the Client server isn't implemented in java, there's no need for us to implement the client side of our application. So we focus on the server application.

Chapter 3

The Server Application

3.1 Complete View

The main idea regarding our application is that all required tasks are handled with their own handler class, which is called through a specific request and handles it with the appropriate method. As you can see in the figure we grouped the handlers in data- and user-handlers and each handler is connected with the user-/datamanager which instances allow each handler to access the database(always via manager and via SPARQL). For example, the user section has classes and methods for user administration and the right system and the data section has classes for package management and. At the top of our application stands the required Rest interface and the HttpServlet.



3.2.1 The Main Servlet

The head section contains four components. The "MainServlet" is the main part and the brain of the operation. Any request goes through here and is then send to the correct processing station. The three methods "doGet", "doPost" and "doDelete" initiate/handle a Http get-, post- and delete-request, which are not processed here but either by the "HttpServlet" or the "RestHandler". The "HttpServlet" handles regular Http-request, while the "RestHandler" handles request initiated through the REST interface. The three remaining methods of the "MainServlet" work with our "ReqHandler" Interface. These handler handle the actual request, for example logging out a user. For each interaction with our application exists one of these handlers, which have one main function, which is to "handle" their specific request. The MainServlet has three methods to work with that interface. First "addReqHandler" which adds a new "ReqHandler", secondly "removeRegHandler", which removes an existing "ReqHandler" and lastly removeAll(), which removes all existing "ReqHandlers".

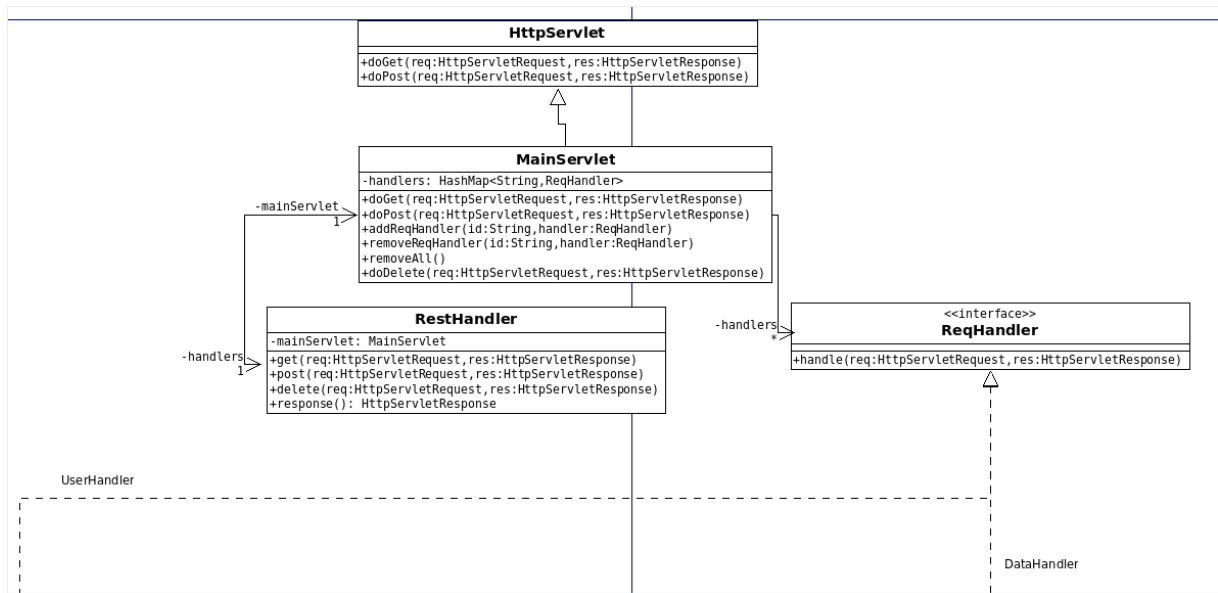


Figure 3.2: Application Head

3.2.2 The User Section

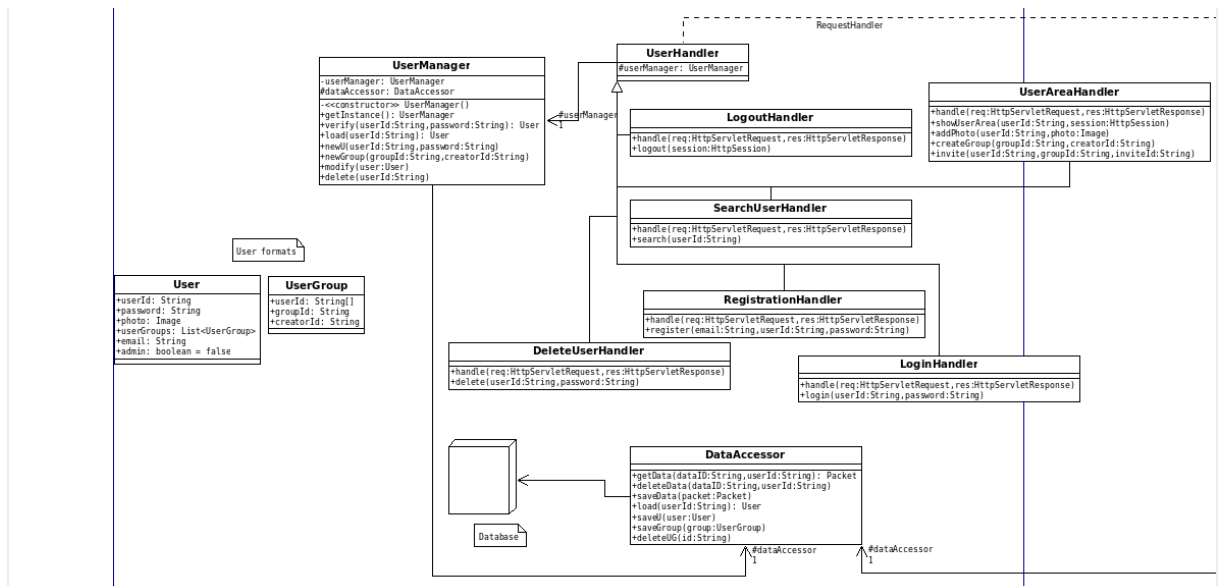


Figure 3.3: User Section

In this section we want to zoom in the user part. As already mentioned there are specific classes for every user request. In the diagram you can see that we offer classes for registration, login, logout, for delete or search users and a user area. All the specific handler classes are derived from the class `UserHandler`.

The class `RegistrationHandler` offers in addition to the `handle` method for communicating with the `HttpServlet` the method `register`, which takes an E-Mail adress, a `userId` and a password. Later we discuss the user registration within a sequence diagram in detail. The login will be handled by the `LoginHandler` class. The method `login` checks via the `UserManager` and the `DataAccessor` whether the transferred parameters `userId` and password are correct. The `LogoutHandler` class ensures, that after the logout request all data of the actual session are stored in the database.

In comparison with the `RegistrationHandler` the `DeleteUserHandler` deals with the opposite action. The cancellation will be initiated by the method call `delete`. After verifying the correctness of the parameters `userId` and password the user will be deleted via the method calls `delete` of the `UserManager` and `deleteU` of the `DataAccessor`.

Another usefull class is the `UserSearchClass` which make sure that User can be searched by their `UserId` to load specific data. The last handler class is the `UserAreaHandler`. Within this class the user can change their user profile after calling the `showUserArea` method or can upload a photo by the `addPhoto` method. In addition to that the class offers the function `createGroup` and `invite` to manage the groups of an user.

For more modularity, seperation and reusability all handler classes uses the methods of the `UserManager` class and to handle the direct accesses to the database the `DataAccessor` class is used before, which is the only class who can interact with the database directly.

As we can see most of the handler classes are nearly used in the same way. Now we want to have a closer look on the user registration via the following sequence diagram as an example how the classes are used and interact together.

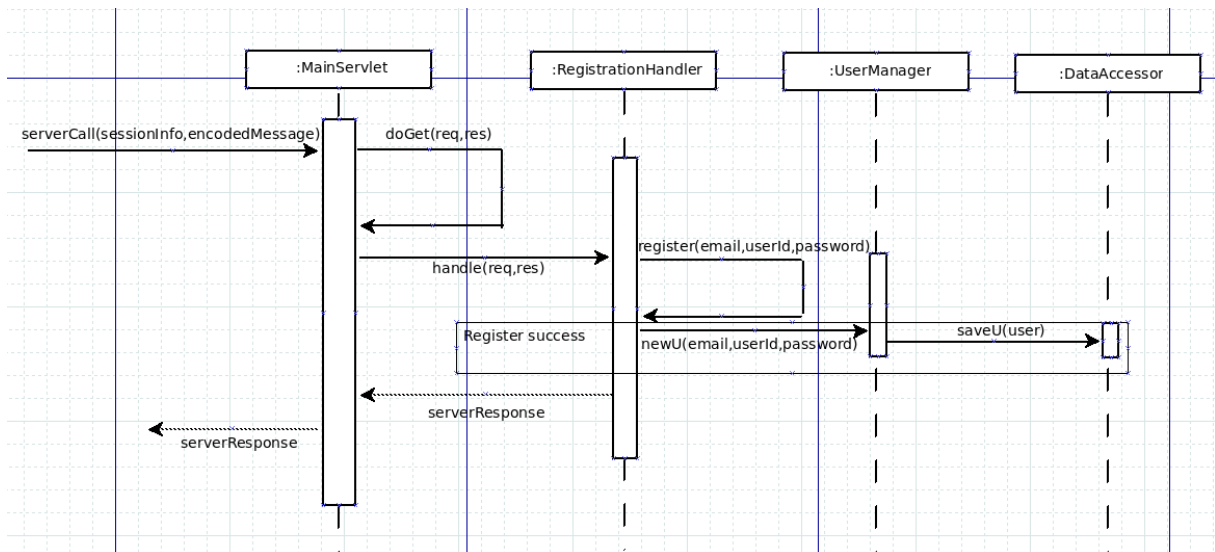


Figure 3.4: Use of an Algorithm

When the user sends a request via browser the client communicates within a `serverCall` with the server. The class `MainServlet` process the request of the client and call the `handle` method from the `RegistrationHandler` class within the `HttpServletRequest` and `HttpServletResponse` parameters. To register the user the parameters `email`, `userId` and `password` are needed. For a successful registration the method `newU` from the `UserManager` is called to create a new User. The `DataAccessor` is the only one who can interact directly with the database, so as a proxy this class saves the new User data by calling the method `saveU`. After a successful registration a `serverResponse` is given back to the client.

As another example we show the creation of a group.

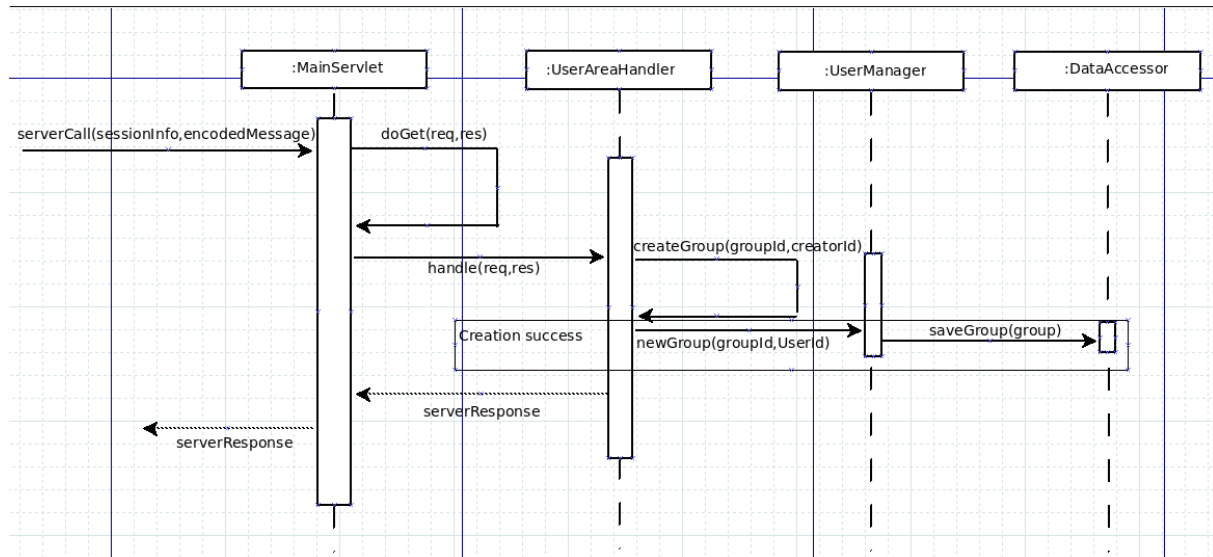


Figure 3.5: Group Creation

The process to create a group uses the same main principle as in the example before. The difference is that we have to use the method `createGroup` of the `UserAreaHandler` class and the methods `newGroup` of the `UserManager` class and `saveGroup` of the `DataAccessor`. These examples emphasize the benefits of the used whole part and proxy patterns.

3.2.3 The Data Section

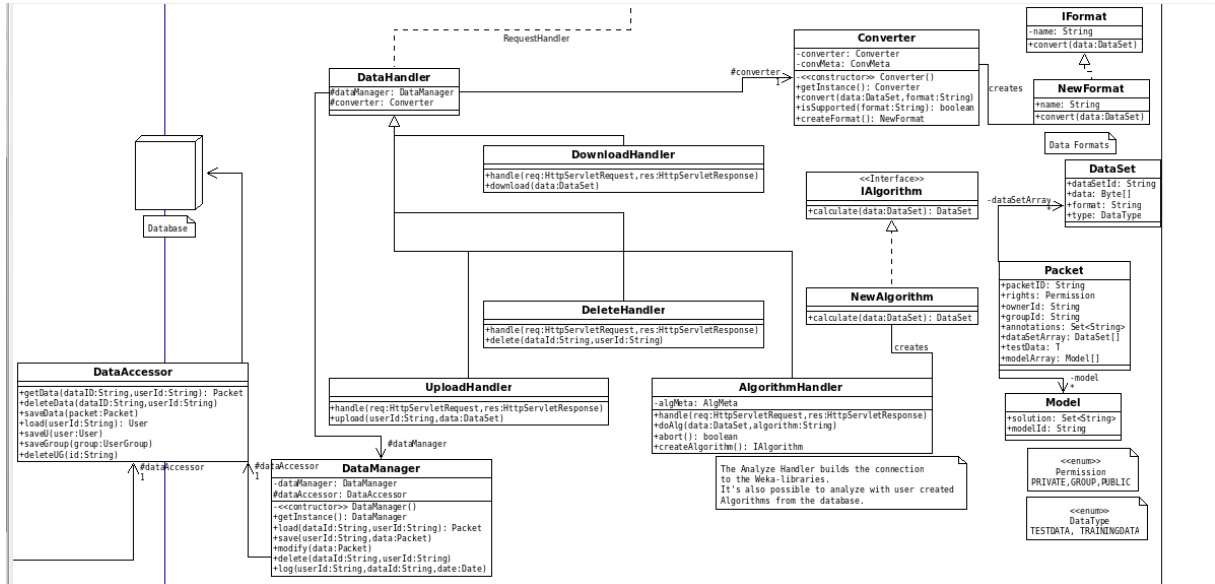


Figure 3.6: Data Section

In the Data section we have different handler classes like download, delete, upload (and that's exactly what they're used for) and algorithm which all extends the DataHandler class, and which itself implements the RequestHandler interface. When MainServlet calls a specific handler with a request, the handler processes the request via the DataManager instance. The DataManager itself communicates with the DataAccessor, which is the only class with the access to the database. The DataAccessor is able to get packets (see below) from the specific user and data ID's, delete and save packets. It is also able to save a user or a group, and also to delete a group, or get a user name from a specific userID. An other important part is the AlgorithmHandler class. It provides methods for execute an chosen algorithm on a specific dataset, also it is able to create a NewAlgorithm class when user wants to via Factory Method.

A packet class provides methods and attributes to create new packets, which hold strings with owner and group ID's, and also arrays with models, datasets and testdata. They also hold information about the permission of a package. Packets will be stored in the database and will be loaded depends on a called handler. The last part is a converter class. Its main function is to convert the result to a useful format for downloading. It's also possible to create new possible formats later through Factory Method.

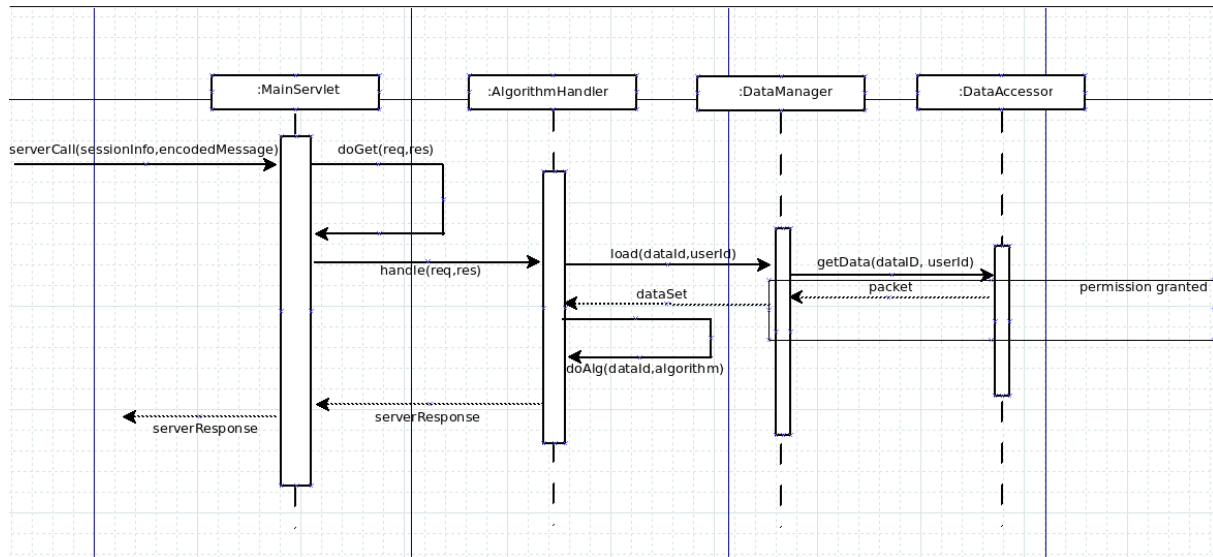


Figure 3.7: Use of Algorithm

In the first sequence the Mainservlet receives a request from the client to use an algorithm on specific data, so the MainServlet calls the required AlgorithmHandler. To get the Data the AlgorithmHandler uses the "load" method of the Datamanager, which calls the DataAccessor, then DataAccessor method gets the packet from the database. The packet takes the opposite way and goes through DataManager, wich returns the required dataSet to the AlgorithmHandler. Then AlgorithmHandler class calls "do algorithm" method on the dataSet, and sends the Response back to the MainServlet.

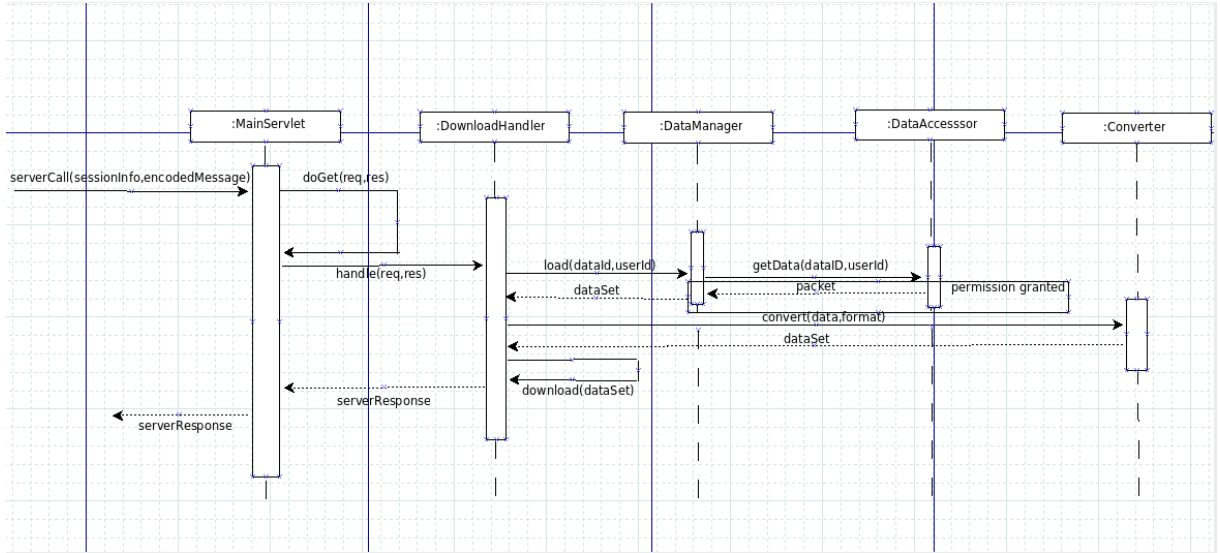


Figure 3.8: Download Data

In this sequence diagram, a download sequence is shown. The MainServlet gets a request from the client and calls the required DownloadHandler. For loading the packet to download the DownloadHandler calls the DataManager, which receives a packet via DataAccessor, and returns the data to the DownloadHandler. Now DownloadHandler has possibilities to give the data to the converter for converting or to provide it directly for downloading via backtransfer to the MainServlet. For conversion a Converter class is being called, which also returns a converted data.

3.3 Used Patterns

For our application, we used the following patterns:

Whole-Part: This pattern is very important to guarantee the required modularity, reusability and a clear separation through the use of classes, which aggregate particular objects/services. The most obvious example for this pattern is the relationship between the MainServlet and handler classes.

Proxy: This pattern is used for more separation and reusability regarding the database, through our DataAccessor class, our proxy class which is the only class which is able to interact with the database.

Singleton: This pattern is for some kind of stability and safety because it prevents that there's more than one instance of a specific class, which is only needed once for example are the manager classes.

Factory Method: To make the plug-in of new result formats and algorithms possible, we use this pattern in the algorithm handler and the converter.

3.4 Resulting Source Code

3.4.1 AlgorithmHandler

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import DataSet;
import IAlgorithm;
import DataHandler;
import NewAlgorithm;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class AlgorithmHandler extends DataHandler
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */
    private AlgMeta algMeta;
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    /**
     * Operation
     *
     * @param req
     * @param res
     * @return
     */
    public handle ( HttpServletRequest req, HttpServletResponse res )
    {
        // ## Implementation preserve start class method.handle@@@@HttpSer
        // ## Implementation preserve end class method.handle@@@@HttpServle
    }
    /**
     * Operation
     *
     * @param data
     * @param algorithm
     * @return
     */
    public doAlg ( DataSet data, String algorithm )
    {
        // ## Implementation preserve start class method.doAlg@@@@DataSet@S
        // ## Implementation preserve end class method.doAlg@@@@DataSet@Str
    }
}

```

```

    /**
     * Operation
     *
     * @return boolean
     */
    public boolean abort ( )
    {
        // ## Implementation preserve start class method.abort@boolean@@
        // ## Implementation preserve end class method.abort@boolean@@
    }
    /**
     * Operation
     *
     * @return IAlgorithm
     */
    public IAlgorithm createAlgorithm ( )
    {
        // ## Implementation preserve start class method.createAlgorithm@IAlgorithm@@
        // ## Implementation preserve end class method.createAlgorithm@IAlgorithm@@
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.2 Converter

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import Converter;
import DataSet;
import NewFormat;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class Converter
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.

// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */

```

```

private Converter converter;
private ConvMeta convMeta;
// ## Implementation preserve start class attributes.
// ## Implementation preserve end class attributes.
/**
 * Operation
 *
 * @return Converter
 */
public Converter getInstance ( )
{
    // ## Implementation preserve start class method.getInstance@Converter
    // ## Implementation preserve end class method.getInstance@Converter
}
/**
 * Operation
 *
 * @param data
 * @param format
 * @return
 */
public convert ( DataSet data, String format )
{
    // ## Implementation preserve start class method.convert@@@@DataSet
    // ## Implementation preserve end class method.convert@@@@DataSet
}
/**
 * Operation
 *
 * @param format
 * @return boolean
 */
public boolean isSupported ( String format )
{
    // ## Implementation preserve start class method.isSupported@boolean
    // ## Implementation preserve end class method.isSupported@boolean
}
/**
 * Operation
 *
 * @return NewFormat
 */
public NewFormat createFormat ( )
{
    // ## Implementation preserve start class method.createFormat@NewForm
    // ## Implementation preserve end class method.createFormat@NewForm
}

```

```

    }
    /**
     * Operation
     *
     * @return
     */
    private Converter ( )
    {
        // ## Implementation preserve start class method.Converter@@@
        // ## Implementation preserve end class method.Converter@@@
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.3 DataAccessor

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import Packet;
import UserManager;
import DataManager;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class DataAccessor
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.

// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    /** Associations */
    private UserManager userManager;
    private DataManager dataManager;
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *

```

```

    * @param dataID
    * @return
    */
    public getData ( String dataID , String userId )
    {
        // ## Implementation preserve start class method.getData@@@@String
        // ## Implementation preserve end class method.getData@@@@String
    }
    /**
    * Operation
    *
    * @param dataID
    * @return
    */
    public Packet deleteData ( String dataID , String userId )
    {
        // ## Implementation preserve start class method.deleteData@@@@String
        // ## Implementation preserve end class method.deleteData@@@@String
    }
    /**
    * Operation
    *
    * @param packet
    * @return
    */
    public saveData ( Packet packet )
    {
        // ## Implementation preserve start class method.saveData@@@@Packet
        // ## Implementation preserve end class method.saveData@@@@Packet
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.

    public User load(String userId){

    }

    public saveU(User user){

    }

    public saveGroup(UserGroup group){

    }

```



```

    public deleteUG(String id){

    }
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.4 DataHandler

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import DataManager;
import Converter;
import ReqHandler;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class DataHandler
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.
    implements ReqHandler
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */
    protected DataManager dataManager;
    protected Converter converter;
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.5 DataManager

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import DataManager;
import DataAccessor;
import Packet;
import DataHandler;

```

```

// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class DataManager
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.

// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */
    private DataManager dataManager;
    protected DataAccessor dataAccessor;
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    /**
     * Operation
     *
     * @return DataManager
     */
    public DataManager getInstance ( )
    {
        // ## Implementation preserve start class method.getInstance@DataMan
        // ## Implementation preserve end class method.getInstance@DataManag
    }
    /**
     * Operation
     *
     * @param dataId
     * @param userId
     * @return Packet
     */
    public Packet load ( String dataId, String userId )
    {
        // ## Implementation preserve start class method.load@Packet@@@Stri
        // ## Implementation preserve end class method.load@Packet@@@String
    }
    /**
     * Operation
     *
     * @param userId
     * @param data
     * @return
     */
    public save ( String userId, Packet data )

```

```

{
    // ## Implementation preserve start class method.save@@@String@Packet
    // ## Implementation preserve end class method.save@@@String@Packet
}
/**
 * Operation
 *
 * @param data
 * @return
 */
public modify ( Packet data )
{
    // ## Implementation preserve start class method.modify@@@@Packet
    // ## Implementation preserve end class method.modify@@@@Packet
}
/**
 * Operation
 *
 * @param dataId
 * @param userId
 * @return
 */
public delete ( String dataId , String userId )
{
    // ## Implementation preserve start class method.delete@@@@String@String
    // ## Implementation preserve end class method.delete@@@@String@String
}
/**
 * Operation
 *
 * @param userId
 * @param dataId
 * @param date
 * @return
 */
public log ( String userId , String dataId , Date date )
{
    // ## Implementation preserve start class method.log@@@@String@String@Date
    // ## Implementation preserve end class method.log@@@@String@String@Date
}
/**
 * Operation
 *
 * @return
 */
private DataManager ( )

```

```

    {
        // ## Implementation preserve start class method.DataManager@@@
        // ## Implementation preserve end class method.DataManager@@@
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.6 DataSet

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import Packet;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class DataSet
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.

// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */
    public String dataId;
    public Byte[] data;
    public String format;
    public DataType type;
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.7 DeleteHandler

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import DataHandler;

```

```

// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class DeleteHandler extends DataHandler
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *
     * @param req
     * @param res
     * @return
     */
    public handle ( HttpServletRequest req, HttpServletResponse res )
    {
        // ## Implementation preserve start class method.handle@@@HttpServletRequest
        // ## Implementation preserve end class method.handle@@@HttpServletRequest
    }
    /**
     * Operation
     *
     * @param dataId
     * @param userId
     * @return
     */
    public delete ( String dataId, String userId )
    {
        // ## Implementation preserve start class method.delete@@@@String@String
        // ## Implementation preserve end class method.delete@@@@String@String
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.8 DeleteUserHandler

```

// ## Implementation preserve start class opening.

```

```

// ## Implementation preserve end class opening.
import UserHandler;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class DeleteUserHandler extends UserHandler
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *
     * @param req
     * @param res
     * @return
     */
    public handle ( HttpServletRequest req, HttpServletResponse res )
    {
        // ## Implementation preserve start class method.handle@@@@HttpSer
        // ## Implementation preserve end class method.handle@@@@HttpServe
    }
    /**
     * Operation
     *
     * @param userId
     * @param password
     * @return
     */
    public delete ( String userId, String password )
    {
        // ## Implementation preserve start class method.delete@@@@String@S
        // ## Implementation preserve end class method.delete@@@@String@Str
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.9 DownloadHandler

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import DataSet;
import DataHandler;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class DownloadHandler extends DataHandler
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *
     * @param req
     * @param res
     * @return
     */
    public handle ( HttpServletRequest req, HttpServletResponse res )
    {
        // ## Implementation preserve start class method.handle@@@HttpServletRequest
        // ## Implementation preserve end class method.handle@@@HttpServletRequest
    }
    /**
     * Operation
     *
     * @param data
     * @return
     */
    public download ( DataSet data )
    {
        // ## Implementation preserve start class method.download@@@@DataSet
        // ## Implementation preserve end class method.download@@@@DataSet
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.

```

```
// ## Implementation preserve end class closing.
```

3.4.10 HttpServlet

```
// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.
```

```
public class HttpServlet
```

```
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.
```

```
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.
```

```
{
```

```
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
```

```
/**
```

```
 * Operation
```

```
 *
```

```
 * @param req
```

```
 * @param res
```

```
 * @return
```

```
 */
```

```
public doGet ( HttpServletRequest req, HttpServletResponse res )
```

```
{
```

```
    // ## Implementation preserve start class method.doGet@@@HttpServlet
```

```
    // ## Implementation preserve end class method.doGet@@@HttpServlet
```

```
}
```

```
/**
```

```
 * Operation
```

```
 *
```

```
 * @param req
```

```
 * @param res
```

```
 * @return
```

```
 */
```

```
public doPost ( HttpServletRequest req, HttpServletResponse res )
```

```
{
```

```
    // ## Implementation preserve start class method.doPost@@@HttpServlet
```

```
    // ## Implementation preserve end class method.doPost@@@HttpServlet
```

```
}
```

```
// ## Implementation preserve start class other.operations.
```



```

    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.11 IAlgorithm

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import DataSet;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public interface IAlgorithm
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.

// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *
     * @param data
     * @return DataSet
     */
    public DataSet calculate ( DataSet data );
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.12 IFormat

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import DataSet;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class IFormat
// ## Implementation preserve start class extends.

```

```

// ## Implementation preserve end class extends.

// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */
    private String name;
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *
     * @param data
     * @return
     */
    public convert ( DataSet data )
    {
        // ## Implementation preserve start class method.convert@@@DataSet
        // ## Implementation preserve end class method.convert@@@DataSet
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.13 LoginHandler

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import UserHandler;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class LoginHandler extends UserHandler
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.

```

```

// ## Implementation preserve end class associations.
/**
 * Operation
 *
 * @param req
 * @param res
 * @return
 */
public handle ( HttpServletRequest req, HttpServletResponse res )
{
    // ## Implementation preserve start class method.handle@@@HttpServletRequest
    // ## Implementation preserve end class method.handle@@@HttpServletRequest
}
/**
 * Operation
 *
 * @param userId
 * @param password
 * @return
 */
public login ( String userId, String password )
{
    // ## Implementation preserve start class method.login@@@String@String
    // ## Implementation preserve end class method.login@@@String@String
}
// ## Implementation preserve start class other.operations.
// ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.14 LogoutHandler

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import UserHandler;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class LogoutHandler extends UserHandler
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.
{
    // ## Implementation preserve start class attributes.

```

```

// ## Implementation preserve end class attributes.
// ## Implementation preserve start class associations.
// ## Implementation preserve end class associations.
/**
 * Operation
 *
 * @param req
 * @param res
 * @return
 */
public handle ( HttpServletRequest req , HttpServletResponse res )
{
    // ## Implementation preserve start class method.handle@@@HttpSer
    // ## Implementation preserve end class method.handle@@@HttpServle
}
/**
 * Operation
 *
 * @param session
 * @return
 */
public logout ( HttpSession session )
{
    // ## Implementation preserve start class method.logout@@@HttpSess
    // ## Implementation preserve end class method.logout@@@HttpSession
}
// ## Implementation preserve start class other.operations.
// ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.15 MainServlet

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import ReqHandler;
import HttpServlet;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class MainServlet extends HttpServlet
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

```

```

{
    /** Attributes */
    private HashMap<String,ReqHandler> handlers;
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *
     * @param req
     * @param res
     * @return
     */
    public doGet ( HttpServletRequest req, HttpServletResponse res )
    {
        // ## Implementation preserve start class method.doGet@@@@HttpServletRequest
        // ## Implementation preserve end class method.doGet@@@@HttpServletRequest
    }
    /**
     * Operation
     *
     * @param req
     * @param res
     * @return
     */
    public doPost ( HttpServletRequest req, HttpServletResponse res )
    {
        // ## Implementation preserve start class method.doPost@@@@HttpServletRequest
        // ## Implementation preserve end class method.doPost@@@@HttpServletRequest
    }
    /**
     * Operation
     *
     * @param id
     * @param handler
     * @return
     */
    public addReqHandler ( String id, ReqHandler handler )
    {
        // ## Implementation preserve start class method.addReqHandler@@@@String@ReqHa
        // ## Implementation preserve end class method.addReqHandler@@@@String@ReqHa
    }
    /**
     * Operation
     *

```

```

    * @param id
    * @param handler
    * @return
    */
    public removeReqHandler ( String id , ReqHandler handler )
    {
        // ## Implementation preserve start class method.removeReqHandler@@@
        // ## Implementation preserve end class method.removeReqHandler@@@
    }
    /**
    * Operation
    *
    * @return
    */
    public removeAll ( )
    {
        // ## Implementation preserve start class method.removeAll@@@
        // ## Implementation preserve end class method.removeAll@@@
    }
    /**
    * Operation
    *
    * @param req
    * @param res
    * @return
    */
    public doDelete ( HttpServletRequest req , HttpServletResponse res )
    {
        // ## Implementation preserve start class method.doDelete@@@@HttpS
        // ## Implementation preserve end class method.doDelete@@@@HttpSer
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.16 Model

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class Model

```

```

// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.

// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */
    public Set<String> solution;
    public String modelId;
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.17 NewAlgorithm

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import DataSet;
import IAlgorithm;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class NewAlgorithm
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.
    implements IAlgorithm
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *
     * @param data

```

```

    * @return DataSet
    */
    public DataSet calculate ( DataSet data )
    {
        // ## Implementation preserve start class method.calculate@DataSet@@@
        // ## Implementation preserve end class method.calculate@DataSet@@@
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.18 NewFormat

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import DataSet;
import IFormat;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class NewFormat extends IFormat
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */
    public String name;
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *
     * @param data
     * @return
     */
    public convert ( DataSet data )
    {
        // ## Implementation preserve start class method.convert@@@@DataSet
        // ## Implementation preserve end class method.convert@@@@DataSet
    }
    // ## Implementation preserve start class other.operations.

```



```

    // ## Implementation preserve end class other.operations.
}

```

```

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.19 Packet

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import Model;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class Packet
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.

// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */
    public String packetID;
    public Permission rights;
    public String ownerId;
    public String groupId;
    public Set<String> annotations;
    public DataSet[] dataSetArray;
    public T testData;
    public Model[] modelArray;
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.20 RegistrationHandler

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import UserHandler;
// ## Implementation preserve start class import.

```

```

// ## Implementation preserve end class import.

public class RegistrationHandler extends UserHandler
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *
     * @param req
     * @param res
     * @return
     */
    public handle ( HttpServletRequest req, HttpServletResponse res )
    {
        // ## Implementation preserve start class method.handle@@@HttpSer
        // ## Implementation preserve end class method.handle@@@HttpServe
    }
    /**
     * Operation
     *
     * @param email
     * @param userId
     * @param password
     * @return
     */
    public register ( String email, String userId, String password )
    {
        // ## Implementation preserve start class method.register@@@String
        // ## Implementation preserve end class method.register@@@String@S
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.21 ReqHandler

```

// ## Implementation preserve start class opening.

```

```

// ## Implementation preserve end class opening.
import MainServlet;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public interface ReqHandler
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.

// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /**
     * Operation
     *
     * @param req
     * @param res
     * @return
     */
    public handle ( HttpServletRequest req, HttpServletResponse res );
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.22 RestHandler

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import MainServlet;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class RestHandler
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.

// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */
    private MainServlet mainServlet;
// ## Implementation preserve start class attributes.
// ## Implementation preserve end class attributes.

```

```

/**
 * Operation
 *
 * @param req
 * @param res
 * @return
 */
public get ( HttpServletRequest req, HttpServletResponse res )
{
    // ## Implementation preserve start class method.get@@@HttpServlet
    // ## Implementation preserve end class method.get@@@HttpServlet
}

/**
 * Operation
 *
 * @param req
 * @param res
 * @return
 */
public post ( HttpServletRequest req, HttpServletResponse res )
{
    // ## Implementation preserve start class method.post@@@HttpServlet
    // ## Implementation preserve end class method.post@@@HttpServlet
}

/**
 * Operation
 *
 * @param req
 * @param res
 * @return
 */
public delete ( HttpServletRequest req, HttpServletResponse res )
{
    // ## Implementation preserve start class method.delete@@@HttpServlet
    // ## Implementation preserve end class method.delete@@@HttpServlet
}

/**
 * Operation
 *
 * @return HttpServletResponse
 */
public HttpServletResponse response ( )
{
    // ## Implementation preserve start class method.response@HttpServlet
    // ## Implementation preserve end class method.response@HttpServlet
}

```

```

    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

```

```

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.23 SearchUserHandler

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import UserHandler;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

```

```

public class SearchUserHandler extends UserHandler
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

```

```

{
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *
     * @param req
     * @param res
     * @return
     */
    public handle ( HttpServletRequest req, HttpServletResponse res )
    {
        // ## Implementation preserve start class method.handle@@@HttpServletRequest
        // ## Implementation preserve end class method.handle@@@HttpServletRequest
    }
    /**
     * Operation
     *
     * @param userId
     * @return
     */
    public search ( String userId )
    {
        // ## Implementation preserve start class method.search@@@String
        // ## Implementation preserve end class method.search@@@String
    }
}

```

```

    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.24 UploadHandler

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import DataSet;
import DataHandler;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class UploadHandler extends DataHandler
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *
     * @param req
     * @param res
     * @return
     */
    public handle ( HttpServletRequest req, HttpServletResponse res )
    {
        // ## Implementation preserve start class method.handle@@@HttpSer
        // ## Implementation preserve end class method.handle@@@HttpServle
    }
    /**
     * Operation
     *
     * @param userId
     * @param data
     * @return
     */
    public upload ( String userId, DataSet data )

```

```

    {
        // ## Implementation preserve start class method.upload@@@@String@DataSet
        // ## Implementation preserve end class method.upload@@@@String@DataSet
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.25 User

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class User
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.

// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */
    public String userId;
    public String password;
    public Image photo;
    public List<UserGroup> userGroups;
    public String email;
    public boolean admin = false;
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.26 UserAreaHandler

```

// ## Implementation preserve start class opening.

```

```

// ## Implementation preserve end class opening.
import UserHandler;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class UserAreaHandler extends UserHandler
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    /**
     * Operation
     *
     * @param req
     * @param res
     * @return
     */
    public handle ( HttpServletRequest req, HttpServletResponse res )
    {
        // ## Implementation preserve start class method.handle@@@HttpSer
        // ## Implementation preserve end class method.handle@@@HttpSer
    }
    /**
     * Operation
     *
     * @param userId
     * @param session
     * @return
     */
    public showUserArea ( String userId, HttpSession session )
    {
        // ## Implementation preserve start class method.showUserArea@@@S
        // ## Implementation preserve end class method.showUserArea@@@Str
    }
    /**
     * Operation
     *
     * @param userId
     * @param photo
     * @return
     */
    public addPhoto ( String userId, Image photo )

```



```

{
    // ## Implementation preserve start class method.addPhoto@@@String@Image
    // ## Implementation preserve end class method.addPhoto@@@String@Image
}
/**
 * Operation
 *
 * @param groupId
 * @param creatorId
 * @return
 */
public createGroup ( String groupId, String creatorId )
{
    // ## Implementation preserve start class method.createGroup@@@String@String
    // ## Implementation preserve end class method.createGroup@@@String@String
}
/**
 * Operation
 *
 * @param userId
 * @param groupId
 * @param inviteId
 * @return
 */
public invite ( String userId, String groupId, String inviteId )
{
    // ## Implementation preserve start class method.invite@@@String@String@String
    // ## Implementation preserve end class method.invite@@@String@String@String
}
// ## Implementation preserve start class other.operations.
// ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.27 UserGroup

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class UserGroup
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.

```

```

// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */
    public String [] userId;
    public String groupId;
    public String creatorId;
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class associations.
    // ## Implementation preserve end class associations.
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```

3.4.28 UserHandler

```

// ## Implementation preserve start class opening.
// ## Implementation preserve end class opening.
import UserManager;
import ReqHandler;
// ## Implementation preserve start class import.
// ## Implementation preserve end class import.

public class UserHandler
// ## Implementation preserve start class extends.
// ## Implementation preserve end class extends.
    implements ReqHandler
// ## Implementation preserve start class inheritance.
// ## Implementation preserve end class inheritance.

{
    /** Attributes */
    protected UserManager userManager;
    // ## Implementation preserve start class attributes.
    // ## Implementation preserve end class attributes.
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.

```

```
// ## Implementation preserve end class closing.
```

3.4.29 UserManager

```
// ## Implementation preserve start class opening.
```

```
// ## Implementation preserve end class opening.
```

```
import UserManager;
```

```
import DataAccessor;
```

```
import User;
```

```
// ## Implementation preserve start class import.
```

```
// ## Implementation preserve end class import.
```

```
public class UserManager
```

```
// ## Implementation preserve start class extends.
```

```
// ## Implementation preserve end class extends.
```

```
// ## Implementation preserve start class inheritance.
```

```
// ## Implementation preserve end class inheritance.
```

```
{
```

```
    /** Attributes */
```

```
    private UserManager userManager;
```

```
    protected DataAccessor dataAccessor;
```

```
    // ## Implementation preserve start class attributes.
```

```
    // ## Implementation preserve end class attributes.
```

```
    // ## Implementation preserve start class associations.
```

```
    // ## Implementation preserve end class associations.
```

```
    /**
```

```
     * Operation
```

```
     *
```

```
     * @return UserManager
```

```
    */
```

```
    public UserManager getInstance ( )
```

```
    {
```

```
        // ## Implementation preserve start class method.getInstance@UserManager@@
```

```
        // ## Implementation preserve end class method.getInstance@UserManager@@
```

```
    }
```

```
    /**
```

```
     * Operation
```

```
     *
```

```
     * @param userId
```

```
     * @param password
```

```
     * @return User
```

```
    */
```

```
    public User verify ( String userId , String password )
```

```
    {
```

```

        // ## Implementation preserve start class method.verify@User@@@String
        // ## Implementation preserve end class method.verify@User@@@String
    }
    /**
     * Operation
     *
     * @param userId
     * @return User
     */
    public User load ( String userId )
    {
        // ## Implementation preserve start class method.load@User@@@String
        // ## Implementation preserve end class method.load@User@@@String
    }
    /**
     * Operation
     *
     * @param userId
     * @param password
     * @return
     */
    public newU ( String userId , String password )
    {
        // ## Implementation preserve start class method.newU@@@@String@String
        // ## Implementation preserve end class method.newU@@@@String@String
    }
    /**
     * Operation
     *
     * @param groupId
     * @param creatorId
     * @return
     */
    public newGroup ( String groupId , String creatorId )
    {
        // ## Implementation preserve start class method.newGroup@@@@String@String
        // ## Implementation preserve end class method.newGroup@@@@String@String
    }
    /**
     * Operation
     *
     * @param user
     * @return
     */
    public modify ( User user )
    {

```

```

        // ## Implementation preserve start class method.modify@@@@User
        // ## Implementation preserve end class method.modify@@@@User
    }
    /**
     * Operation
     *
     * @param userId
     * @return
     */
    public delete ( String userId )
    {
        // ## Implementation preserve start class method.delete@@@@String
        // ## Implementation preserve end class method.delete@@@@String
    }
    /** Constructor *
     * Operation
     *
     * @param
     * @return
     */
    private UserManager (    )
    {
        // ## Implementation preserve start class method.UserManager@@@@
        // ## Implementation preserve end class method.UserManager@@@@
    }
    // ## Implementation preserve start class other.operations.
    // ## Implementation preserve end class other.operations.
}

// ## Implementation preserve start class closing.
// ## Implementation preserve end class closing.

```