# 1. Login and registration function

#### **Step 1: Create the login page**

1. Create a named login. The login page of the jsp.
2. Logession forms were constructed using the HTML and JSP tags.
3. Include the form fields for entering the user name and password.

#### **Step 2: Create the registration page**

1. Create a project called register. The registration page for the jsp.
2. The registry form was constructed using the HTML and JSP tags.
3. Include the form fields for entering the user name and password.

#### **Step 3: Implement the user authentication**

1. Create a Java Servlet called LoginServlet to process a user login.
2. In LoginServlet, verify that the user-provided user name and password match the user in the database.
3. If matched, create a user session.
4. If not matched, to return an error message.

#### **Step 4: Implement user registration**

1. Create a Java Servlet called RegisterServlet to process user registration.
2. In RegisterServlet, save the user-provided user name and password to the database.
3. An error message was returned if the user name already exists.

# 2. Robot addition, deletion, change and check function

#### **Step 1: Create the robot list page**

1. Create a project called robotList. The page of jsp to display the bot list.
2. Use the JSP to iterate and displays the robot record in the robot list.
3. Provide "Detail", "Edit" and "Delete" buttons for each robot record.

#### **Step 2: Create the Add robot page**

1. Create a project called addRobot. Page of jsp, allowing users to add new robots.
2. Use the HTML and JSP tags to create form fields to enter the robot's name, picture, and video URL for the robot.
3. Realize the image upload function:
   1. Create a Java Servlet called ImageServlet to process the file upload.
   2. configure web. The multipart-config element in xml to support file uploading.
   3. In Servlet, get the uploaded file and save it to the specified directory on the server.
   4. Update the database record of the robot to contain the uploaded file path.

#### **Step 3:3. Create the edit robot page**

1. Create a project called editRobot. The page of jsp, which allows the user to edit the existing bot information.
2. Use the HTML and JSP tags to create form fields to edit the robot's name, picture, and video URL.
3. Realize the image upload function (synchronization step 2).

#### **Step 4: Create the robot details page**

1. Create a project called robotDetail. The page of jsp, which displays the details of the robot, including names, pictures, and videos.
2. Display the robot using the JSP tag.

#### **Step 5: Realize the robot's addition, deletion, modification and check**

1. Create a Java Servlets (such as a RobotServlet) to handle the addition, deletion, modification and checking operation of the robot.
2. Write Servlet code to handle requests to add, edit, and delete the bot.

# 4. User's adding, deleting, changing and checking function

#### **Step 1: Create a user list page**

1. Create a project called userList. Page of jsp, showing a list of users.
2. Use the JSP to iterate and display the user records in the user list.
3. Provide the "Edit" and "Delete" buttons for each user record.

#### **Step 2: Create an Add user page**

1. Create a project called addUser. Page of jsp, which allows users to add new users.
2. Use the HTML and JSP tags to create form fields to enter the user name and password.

#### **Step 3: Create an edit user page**

1. Create a project called editUser. Page of jsp that allows users to edit existing user information.
2. Create form fields using HTML and JSP tags to edit user names and passwords.

#### **Step 4: Realize the user's addition, deleting, changing and checking**

1. Create Java Servlets (such as UserServlet) to handle user addition, deletion, and change operations.
2. Write Servlet code to handle add, edit, and delete user requests.

# 5. Explore the recording visualization function

#### **Step 1: Create the Discovery record list page**

1. Create a project called explorationList. The page of jsp for displaying a list of exploration records.
2. Using JSP iterates and displays the exploration record in the list of exploration records.
3. Provide the "Edit" and "Delete" buttons for each exploration record.

#### **Step 2: Create the Add exploration record page**

1. Create a project called addExploration. The page of jsp, which allows the user to add a new exploration record.
2. Form fields were created using HTML and JSP tags for input of robot ID, exploration time, and treasure counting.

#### **Step 3: Create an edit exploration record page**

1. Create a project called editExploration. The page of jsp, which allows the user to edit the existing exploration record.
2. Create form fields using HTML and JSP tags for editing exploration time and treasure counts.

#### **Step 4: Realize the addition, deletion, modification and checking of exploration records**

1. Create a Java Servlets (such as ExplorationServlet) to handle the addition, deletion, and modification of exploration records.
2. Write Servlet code to handle requests to add, edit, and delete exploration records.

# 6. Implementation of the menu bar

#### **Step 1: Create a menu bar page**

1. Create a name for the top. The page for jsp, used to display the top menu bar.
2. Create menu items using JSP tags, linking to list pages of users, robots, and exploration records.

#### **Step 2: Create the top menu bar**

1. Create a name for the top. Ppage for jsp to display the menu at the top of the page.
2. Use the JSP tags to create list pages linked to the user, bot, and exploration records.

#### **Step 3: Use the menu bar**

1. Introduce top at the top of each page. To jsp to display the menu bar.
2. Optionally display the login / logout link according to the user's login status.