package total.store.servlet;  
  
import org.apache.commons.fileupload.FileItem;  
import org.apache.commons.fileupload.disk.DiskFileItemFactory;  
import org.apache.commons.fileupload.servlet.ServletFileUpload;  
import total.store.model.BookRecovery;  
import total.store.model.RecoveryOrder;  
import total.store.service.BookRecoveryService;  
import total.store.service.RecoveryOrderService;  
import total.store.service.UserService;  
import total.store.service.impl.BookRecoveryServiceImpl;  
import total.store.service.impl.RecoveryOrderServiceImpl;  
import total.store.service.impl.UserSeviceImpl;  
import total.store.util.Character\_generation;  
  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.io.InputStream;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.List;  
  
@WebServlet(name = "recoveryBookServlet",urlPatterns = "/recoveryBookServlet")  
public class RecoveryBookServlet extends HttpServlet {  
 @Override  
 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 String action = req.getParameter("action");  
 resp.setContentType("text/html;charset=utf-8");  
 BookRecoveryService bookRecoveryService = new BookRecoveryServiceImpl();  
 if (action.equals("re\_book\_list")){  
 List<BookRecovery> bookRecoveryList = bookRecoveryService.bookre\_list\_audito();  
 req.getSession().setAttribute("bookRecoveryList",bookRecoveryList);  
 resp.sendRedirect("auditor-book-recovery.jsp");  
 }else if (action.equals("re\_hi\_book\_list")){  
 List<BookRecovery> bookreHiListAudito = bookRecoveryService.bookre\_hi\_list\_audito();  
 req.getSession().setAttribute("bookreHiListAudito",bookreHiListAudito);  
 resp.sendRedirect("auditor-book-history-recovery.jsp");  
 }else if (action.equals("user\_rebook")){  
 String userid = req.getParameter("userid");  
 List<BookRecovery> bookreListUser = bookRecoveryService.bookre\_list\_user(Integer.parseInt(userid));  
 req.getSession().setAttribute("bookreListUser",bookreListUser);  
 List<BookRecovery> listHistoryUser = bookRecoveryService.bookre\_list\_history\_user(Integer.parseInt(userid));  
 req.getSession().setAttribute("listHistoryUser",listHistoryUser);  
 resp.sendRedirect("user-recovery-information.jsp");  
 }else if (action.equals("admin\_re")){  
 List<BookRecovery> bookreListAdmin = bookRecoveryService.bookre\_list\_admin();  
 req.getSession().setAttribute("bookreListAdmin",bookreListAdmin);  
 resp.sendRedirect("admin-book-recovery.jsp");  
 }else if (action.equals("admin\_re\_his")){  
 List<BookRecovery> bookreListHistoryAdmin = bookRecoveryService.bookre\_list\_history\_admin();  
 req.getSession().setAttribute("bookreListHistoryAdmin",bookreListHistoryAdmin);  
 resp.sendRedirect("admin-book-history-recovery.jsp");  
 }else if (action.equals("user\_buybook")){  
 List<BookRecovery> bookreListHistoryuser = bookRecoveryService.book\_list\_buy\_history();  
 req.getSession().setAttribute("bookreListHistoryuser",bookreListHistoryuser);  
 resp.sendRedirect("shop-no-sidebar.jsp");  
 }else if (action.equals("single\_rebook")){  
 String bookRecoveryid = req.getParameter("bookRecoveryid");  
 BookRecovery bookre\_one = bookRecoveryService.bookre\_one(bookRecoveryid);  
 req.getSession().setAttribute("bookre\_one",bookre\_one);  
 resp.sendRedirect("single-reproduct.jsp");  
 }  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
 String action = req.getParameter("action");  
 resp.setContentType("text/html;charset=utf-8");  
 BookRecoveryService bookRecoveryService = new BookRecoveryServiceImpl();  
 if (action.equals("recoverybook\_add")){  
 String uploadFilePath = this.getServletContext().getRealPath("/recovery");  
 File file = new File(uploadFilePath);  
 if (!file.exists() && !file.isDirectory()){  
 file.mkdir();  
 }  
 //创建工厂  
 DiskFileItemFactory factory = new DiskFileItemFactory();  
 //用于解析上传数据的工具类ServletFileUpload  
 ServletFileUpload upload = new ServletFileUpload(factory);  
 //设置缓存文件最大值  
 upload.setFileSizeMax(10\*1024\*1024);  
 //设置上传文件最大值  
 upload.setSizeMax(5\*1024\*1024);  
 upload.setHeaderEncoding("UTF-8");  
 if (!ServletFileUpload.isMultipartContent(req)){  
 return;  
 }  
 //初始化为空  
 String bookname = null; // 书籍名  
 String bookautorname = null; // 作者名  
 String bookpublishinghouse = null; // 出版社信息  
 String userid = req.getParameter("userid"); //用户信息  
 String img\_1 = null;// 书籍封面  
 String img\_2 = null;// 内页一  
 String img\_3 = null;// 内页二  
 String img\_4 = null;// 内页三  
 String img\_5 = null;// 内页四  
 String img\_6 = null;// 内页五  
 String img\_7 = null;// 书籍底面  
 //调用解析方法  
 try {  
 //定义一个集合 接受上传的所有请求参数  
 List<FileItem> list = upload.parseRequest(req);  
 //循环判断，每一个表单项，是普通类型，还是上传的文件  
 for (FileItem fileItem : list) {  
 if (fileItem.isFormField()) {  
 //普通表单项  
 String filedName = fileItem.getFieldName();  
 if (filedName.equals("bookname")) {  
 bookname = fileItem.getString("UTF-8");  
 } else if (filedName.equals("bookautorname")) {  
 bookautorname = fileItem.getString("UTF-8");  
 } else if (filedName.equals("bookpublishinghouse")) {  
 bookpublishinghouse = fileItem.getString("UTF-8");  
 }  
 } else {  
 String filedName = fileItem.getFieldName();  
 if (filedName.equals("img1")){  
 String filename = fileItem.getName();  
 if (filename == null || filename.trim().equals("")){  
 img\_1 = "/recovery/555.png";  
 continue;  
 }  
 filename = filename.substring(filename.lastIndexOf("\\")+1);  
 InputStream in = fileItem.getInputStream();  
 FileOutputStream out = new FileOutputStream(uploadFilePath+"\\"+filename);  
 byte buffer[] = new byte[1024];  
 int len = 0;  
 while ((len = in.read(buffer))>0){  
 out.write(buffer,0,len);  
 }  
 img\_1 = "/recovery/"+filename;//书籍封面  
 in.close();  
 out.close();  
 fileItem.delete();  
 } else if (filedName.equals("img3")){  
 String filename = fileItem.getName();  
 if (filename == null || filename.trim().equals("")){  
 img\_2 = "/recovery/555.png";  
 continue;  
 }  
 filename = filename.substring(filename.lastIndexOf("\\")+1);  
 InputStream in = fileItem.getInputStream();  
 FileOutputStream out = new FileOutputStream(uploadFilePath+"\\"+filename);  
 byte buffer[] = new byte[1024];  
 int len = 0;  
 while ((len = in.read(buffer))>0){  
 out.write(buffer,0,len);  
 }  
 img\_2 = "/recovery/"+filename;;  
 in.close();  
 out.close();  
 fileItem.delete();  
 }else if (filedName.equals("img4")){  
 String filename = fileItem.getName();  
 if (filename == null || filename.trim().equals("")){  
 img\_3 = "/recovery/555.png";  
 continue;  
 }  
 filename = filename.substring(filename.lastIndexOf("\\")+1);  
 InputStream in = fileItem.getInputStream();  
 FileOutputStream out = new FileOutputStream(uploadFilePath+"\\"+filename);  
 byte buffer[] = new byte[1024];  
 int len = 0;  
 while ((len = in.read(buffer))>0){  
 out.write(buffer,0,len);  
 }  
 img\_3 = "/recovery/"+filename;;  
 in.close();  
 out.close();  
 fileItem.delete();  
 }else if (filedName.equals("img5")){  
 String filename = fileItem.getName();  
 if (filename == null || filename.trim().equals("")){  
 img\_4 = "/recovery/555.png";  
 continue;  
 }  
 filename = filename.substring(filename.lastIndexOf("\\")+1);  
 InputStream in = fileItem.getInputStream();  
 FileOutputStream out = new FileOutputStream(uploadFilePath+"\\"+filename);  
 byte buffer[] = new byte[1024];  
 int len = 0;  
 while ((len = in.read(buffer))>0){  
 out.write(buffer,0,len);  
 }  
 img\_4 = "/recovery/"+filename;;  
 in.close();  
 out.close();  
 fileItem.delete();  
 }else if (filedName.equals("img6")){  
 String filename = fileItem.getName();  
 if (filename == null || filename.trim().equals("")){  
 img\_5 = "/recovery/555.png";  
 continue;  
 }  
 filename = filename.substring(filename.lastIndexOf("\\")+1);  
 InputStream in = fileItem.getInputStream();  
 FileOutputStream out = new FileOutputStream(uploadFilePath+"\\"+filename);  
 byte buffer[] = new byte[1024];  
 int len = 0;  
 while ((len = in.read(buffer))>0){  
 out.write(buffer,0,len);  
 }  
 img\_5 = "/recovery/"+filename;  
 in.close();  
 out.close();  
 fileItem.delete();  
 }else if (filedName.equals("img7")){  
 String filename = fileItem.getName();  
 if (filename == null || filename.trim().equals("")){  
 img\_6 = "/recovery/555.png";  
 continue;  
 }  
 filename = filename.substring(filename.lastIndexOf("\\")+1);  
 InputStream in = fileItem.getInputStream();  
 FileOutputStream out = new FileOutputStream(uploadFilePath+"\\"+filename);  
 byte buffer[] = new byte[1024];  
 int len = 0;  
 while ((len = in.read(buffer))>0){  
 out.write(buffer,0,len);  
 }  
 img\_6 = "/recovery/"+filename;;  
 in.close();  
 out.close();  
 fileItem.delete();  
 }else if (filedName.equals("img2")){  
 String filename = fileItem.getName();  
 if (filename == null || filename.trim().equals("")){  
 img\_7 = "/recovery/555.png";  
 continue;  
 }  
 filename = filename.substring(filename.lastIndexOf("\\")+1);  
 InputStream in = fileItem.getInputStream();  
 FileOutputStream out = new FileOutputStream(uploadFilePath+"\\"+filename);  
 byte buffer[] = new byte[1024];  
 int len = 0;  
 while ((len = in.read(buffer))>0){  
 out.write(buffer,0,len);  
 }  
 img\_7 = "/recovery/"+filename;;  
 in.close();  
 out.close();  
 fileItem.delete();  
 }  
 }  
 }  
 Character\_generation character\_generation = new Character\_generation();  
 String re = character\_generation.getRandomString\_re();  
 BookRecovery bookRecovery = new BookRecovery(re,bookname,bookautorname,bookpublishinghouse,Integer.parseInt(userid),img\_1,img\_2,img\_3,img\_4,img\_5,img\_6,img\_7);  
 Boolean bookreAddAudito = bookRecoveryService.bookre\_add\_audito(bookRecovery);  
 if (bookreAddAudito){  
 resp.sendRedirect("shop-list.jsp");  
 }else {  
 System.out.println("失败");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }else if(action.equals("book\_yre")){  
 String recoverybookid = req.getParameter("recoverybookid");  
 String number = req.getParameter("number");  
 BookRecovery bookRecovery = new BookRecovery(recoverybookid,Double.parseDouble(number));  
 Boolean result = bookRecoveryService.bookre\_mosta\_audito(bookRecovery);  
 if (result){  
 req.setAttribute("rebook\_message","修改成功");  
 }else {  
 req.setAttribute("rebook\_message","修改失败");  
 }  
 }else if (action.equals("rebook\_not")){  
 String recoverybookid = req.getParameter("recoverybookid");  
 Boolean result = bookRecoveryService.bookre\_monsta\_audito(recoverybookid);  
 if (result){  
 req.setAttribute("rebook\_message","修改成功");  
 }else {  
 req.setAttribute("rebook\_message","修改失败");  
 }  
 }else if (action.equals("user\_re\_go")){  
 String bookRecoveryid = req.getParameter("bookRecoveryid");  
 int userid = bookRecoveryService.user\_infor\_search(bookRecoveryid);  
 double price\_infor = bookRecoveryService.price\_infor(bookRecoveryid);  
 if (price\_infor>0){  
 UserService userService = new UserSeviceImpl();  
 double user\_sear = userService.user\_sear(userid);  
 double user\_sear\_to = user\_sear + price\_infor;  
 Boolean result = userService.user\_modify\_pri(userid, user\_sear\_to);  
 if (result){  
 req.setAttribute("rebook\_message","修改成功");  
 Boolean monsta\_user = bookRecoveryService.bookre\_monsta\_user(bookRecoveryid);  
 if (monsta\_user){  
 req.setAttribute("rebook\_message","回收成功");  
 }else {  
 req.setAttribute("rebook\_message","回收失败");  
 }  
 }else {  
 req.setAttribute("rebook\_message","修改失败");  
 }  
 }  
 }else if (action.equals("recovery\_delete")){  
 String bookRecoveryid = req.getParameter("bookRecoveryid");  
 Boolean denstaUser = bookRecoveryService.bookre\_densta\_user(bookRecoveryid);  
 if (denstaUser){  
 req.setAttribute("rebook\_message","修改成功");  
 }else {  
 req.setAttribute("rebook\_message","修改失败");  
 }  
 }else if (action.equals("re\_book\_admin")){  
 String bookRecoveryid = req.getParameter("recoverybookid");  
 Boolean admin = bookRecoveryService.bookre\_mosta\_admin(bookRecoveryid);  
 if (admin){  
 req.setAttribute("rebook\_message","修改成功");  
 }else {  
 req.setAttribute("rebook\_message","修改失败");  
 }  
 }else if (action.equals("buy\_book")){  
 String bookRecoveryid = req.getParameter("bookRecoveryid");  
 String price = req.getParameter("price");  
 String userid = req.getParameter("userid");  
 String address = req.getParameter("address");  
 UserService userService = new UserSeviceImpl();  
 double user\_sear = userService.user\_sear(Integer.parseInt(userid));  
 if (user\_sear<Double.parseDouble(price)){  
 req.getRequestDispatcher("index.jsp").forward(req,resp);  
 }else {  
 double user\_price = user\_sear - Double.parseDouble(price);  
 Boolean modify\_pri = userService.user\_modify\_pri(Integer.parseInt(userid), user\_price);  
 if (modify\_pri){  
 req.setAttribute("rebook\_message","购买成功");  
 Boolean mostaUser = bookRecoveryService.bookre\_mosta\_user(bookRecoveryid);  
 if (mostaUser){  
 req.setAttribute("rebook\_message","修改成功");  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  
 String dateNow = sdf.format(new Date());  
 Character\_generation character\_generation = new Character\_generation();  
 String id = character\_generation.getRandomString\_re();  
 RecoveryOrder recoveryOrder = new RecoveryOrder(id,bookRecoveryid,Integer.parseInt(address),Double.parseDouble(price));  
 RecoveryOrderService recoveryOrderService = new RecoveryOrderServiceImpl();  
 Boolean aBoolean = recoveryOrderService.recoveryOrder\_add(recoveryOrder);  
 if (aBoolean){  
 req.setAttribute("recoveryOrder\_add","添加成功");  
 }else {  
 req.setAttribute("recoveryOrder\_add","添加失败");  
 }  
 }else {  
 req.setAttribute("rebook\_message","修改失败");  
 }  
 }else {  
 req.setAttribute("rebook\_message","购买失败");  
 }  
 }  
 }  
 }  
}