City University of Hong Kong

2023-2024 Semester A

CS3343 Software Engineering Practice

Bug Report

Group 10

Study room scheduler

Project Manager: Dilnaz AUSHAKHIMOVA

Testing Team Lead: Nurdaulet TAUMERGENOV

Developer’s Team Lead: Nur ALDEKEN

Programmers and Testers: Aliya OSPANOVA, Alibi ZHENIS, Anton SHATOKHIN, Beket YERMEKOV

Bug-00001

**Title**: **Program crashes when we click on ‘Delete booking’or ‘Change booking’ after we launch**

**Status**: Closed

**Product**: Study room scheduler

**Type**: Crash

**Component**: Change Booking & Delete Booking

**Version**: 1.2

**Platform**: PC Window

**Importance**: Critical

**Reported**: 2023-10-26 18:12:13 HKT by Beket Yermekov

**Modified**: 2023-10-27 22:38:55 HKT

**Assignee**: Nurdaulet Taumergenov

Beket Yermekov 2023-10-26 18:12:13 HKT

**Problem:**

I tried to test the ‘Delete booking’ and ‘Change booking’ option when I run the program for the first time. The program crashed every time I click on the ‘Delete’ or ‘Change’ buttons.

Steps to reproduce:

1. Run Main.java
2. Click ‘Delete booking’ button.

Alternative step:

2\*) Click ‘Change booking’ button.

**Expected result:**

In case when we launch the program from the very start (without launching existing system, since there is no booking made,‘Delete’ and ‘Change’ buttons should not be functional.

**Actual result:**

A black rectangular frame with colorful text

Description automatically generated

Nurdaulet Taumergenov 2023-10-27 22:38:55 HKT

Exception handled: When the program runs without loading an existing system, ‘Delete booking’ and ‘Change booking’ will display ‘No bookings available’.

###

private void addActionButtons(JPanel btnPanel) {  
 if (allRoomsList == null) {  
 JLabel label = new JLabel("No bookings available");  
 label.setHorizontalAlignment(SwingConstants.*CENTER*);  
 btnPanel.add(label);  
 } else {  
 for (StudyRoom sr : allRoomsList.getRooms()) {  
 addButton(btnPanel, sr.getName(), sr.getName());  
 }  
 }  
}

A screenshot of a computer

Description automatically generated

Bug-00002

**Title**: **Change Booking erases Booker’s Name**

**Status**: Closed

**Product**: Study room scheduler

**Type**: Functional error

**Component**: Change Booking

**Version**: 1.2

**Platform**: PC Window

**Importance**: Minor

**Reported**: 2023-11-05 15:11:17 HKT by Beket Yermekov

**Modified**: 2023-11-07 21:25:55 HKT

**Assignee**: Nur Aldeken

Beket Yermekov 2023-10-26 18:12:13 HKT

**Problem:**

After Change Booking, Show Schedule does not show the name.

Steps to reproduce:

1. Run Main.java
2. Click ‘Book Study Space’ button.
3. Choose a room and indicate your Name.
4. Click ‘Change Booking’ button.
5. Change the time slot to another.
6. Click ‘Show Schedule’ button.
7. Enter the room you have chosen

**Expected result: Actual result:**

A white paper with black text

Description automatically generated

The changed timeslot must indicate the original Name of the booker.

Nur Aldeken 2023-11-07 21:25:55 HKT

Bug:

BookTimeSlot function invokes .getTimeSlotUser() after deleting the time slot, hence the user name is lost.

Solution:

Store user name before deleting the time slot, and use the name for BookTimeSlot to invoke .getTimeSlotUser():

private void addPopUpItems(JFrame popUpFrame, Container panel, int time, int roomIndex) {  
 for (int i = 0; i < schedule.size(); i++) {  
 if (schedule.get(i).equals("Free")) {  
 optionsList.add(Integer.*toString*(i + 9));  
 }  
 }  
 String[] options = optionsList.toArray(new String[0]);  
 JLabel nameLabel = new JLabel("Change old booking to:");  
 JComboBox<String> comboBox = new JComboBox<>(options);  
 JButton submit = new JButton("Submit");  
 submit.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent actionEvent) {  
 String name = allRoomsList.get(roomIndex).getTimeSlotUser(time);  
 allRoomsList.get(roomIndex).deleteTimeSlot(time);  
 int selected = Integer.*valueOf*(comboBox.getItemAt(comboBox.getSelectedIndex()));  
 allRoomsList.get(roomIndex).bookTimeSlot(selected, name);  
 popUpFrame.dispose();  
 changeFrame.dispose();  
 new ChangeGUI(allRoomsList);  
 }  
 });  
 panel.add(nameLabel);  
 panel.add(comboBox);  
 panel.add(submit);  
}

Bug-00003

**Title**: **Show schedule does not show Booking data**

**Status**: Closed

**Product**: Study room scheduler

**Type**: Functional error

**Component**: Show Schedule

**Version**: 1.0

**Platform**: PC Window

**Importance**: Minor

**Reported**: 2023-09-06 12:12:17 HKT by Beket Yermekov

**Modified**: 2023-09-07 15:35:35 HKT

**Assignee**: Nurdaulet Taumergenov

Beket Yermekov 2023-09-06 12:12:17 HKT

**Problem:**

After Booking study space, ‘Show schedule’ does not show any changes.

Steps to reproduce:

1. Run Main.java
2. Click ‘Book Study Space’ button.
3. Choose a room and indicate your Name.
4. Click ‘Show Schedule’ button.

***After Booking study space:***

**Expected result: Actual result:**

A screen shot of a sign

Description automatically generated

‘Show schedule’ does not indicate booking data.

Nurdaulet Taumergenov 2023-09-07 15:35:35 HKT

At current state of project development, we have implemented a TimeSlot class that makes easy handling adding/deleting booking:

// class that holds data about each timeslot.  
public class TimeSlot implements Writable {  
 private String userName;  
 private boolean status;  
  
 // Modifies: this  
 // Effects: sets default values for the local variables  
 public TimeSlot() {  
 this.userName = "";  
 this.status = true;  
 }  
  
 public TimeSlot(String username, boolean status) {  
 this.userName = username;  
 this.status = status;  
 }  
  
 // Effects: returns a status of the timeslot.  
 public boolean getStatus() {  
 return status;  
 }  
  
 // Modifies: this  
 // Effects: changes the status of the timeslot to booked and assigns a new userName.  
 public void book(String name) {  
 this.status = false;  
 this.userName = name;  
 EventLog.*getInstance*().logEvent(new Event("Timeslot successfully added!"));  
 }  
  
 // Modifies: this  
 // Effects: changes the status of the timeslot to free and deletes the userName.  
 public void delete() {  
 this.userName = "";  
 this.status = true;  
 EventLog.*getInstance*().logEvent(new Event("Deleted an old booking"));  
 }  
  
  
}

StudyRoom.java fragment:

public void bookTimeSlot(int time, String name) {  
 TimeSlot current = allTimeSlots.get(time - 9);  
 current.book(name);  
 EventLog.*getInstance*().logEvent(new Event(String.*format*("Booked a timeslot for %s at %d", name, time)));  
}

public void deleteTimeSlot(int time) {  
 TimeSlot current = allTimeSlots.get(time - 9);  
 current.delete();  
 EventLog.*getInstance*().logEvent(new Event(String.*format*("Deleted timeslot at %d", time)));  
}

Bug-00004

**Title**: **Booking study space books a time slot for all rooms**

**Status**: Closed

**Product**: Study room scheduler

**Type**: Functional error

**Component**: Book study space

**Version**: 1.0

**Platform**: PC Window

**Importance**: Minor

**Reported**: 2023-09-10 12:00:17 HKT by Beket Yermekov

**Modified**: 2023-09-11 12:12:32 HKT

**Assignee**: Nurdaulet Taumergenov

Beket Yermekov 2023-09-10 12:00:17 HKT

**Problem:**

When I try to book a certain time slot for a particular room, it books the time slot for all other rooms as well.

Steps to reproduce:

1. Run Main.java
2. Click ‘Book Study Space’ button.
3. Choose a room and indicate your Name.
4. Click ‘Show Schedule’ button.
5. Browse each room to see the booked timeslots

***After Booking study space:***

**Expected result:**

Time slot booked for 1 room only.

**Actual result:**

Time slot is booked for every room.

Nurdaulet Taumergenov 2023-09-11 12:12:32 HKT

Issue solved:

(BookGUI.java) actionPerformed function implements booking a timeslot for an indicated room.

submit.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent actionEvent) {  
 allRoomsList.get(roomIndex).bookTimeSlot(time, textField.getText());  
 popUpFrame.dispose();  
 bookFrame.dispose();  
 new BookGUI(allRoomsList);  
 }  
});

Bug-00005

**Title**: **Delete booking sets default value to NILL**

**Status**: Closed

**Product**: Study room scheduler

**Type**: Functional error

**Component**: Delete Booking

**Version**: 1.0

**Platform**: PC Window

**Importance**: Minor

**Reported**: 2023-09-15 13:04:11 HKT by Beket Yermekov

**Modified**: 2023-09-18 18:17:25 HKT

**Assignee**: Nurdaulet Taumergenov

Beket Yermekov 2023-09-15 13:04:11 HKT

**Problem:**

The default field name for non-booked rooms should be ‘Free’ initially. However, when I delete my existing booking, its field name does not change back to ‘Free’, rather it stays as ‘’(empty string).

Steps to reproduce:

1)Run Main.java

1. Click ‘Book Study Space’ button.
2. Choose a room and indicate your Name.
3. Click ‘Delete booking’ and delete the time slot.
4. Click ‘Show Schedule’ button.
5. The freed timeslot shows empty string (not ‘Free’)

**Expected result: Actual result:**

A screen shot of a sign

Description automatically generated A screen shot of a sign

Description automatically generated

Nurdaulet Taumergenov 2023-09-18 18:17:25 HKT

Issue solved:

(StudyRoom.java) deleteTimeSlot function ensures TimeSlot to set status to **True** indicating that it is a free time slot. Consequently, getSchedule()can list free time slots as ‘Free’.

public void deleteTimeSlot(int time) {  
 TimeSlot current = allTimeSlots.get(time - 9);  
 current.delete();  
 EventLog.*getInstance*().logEvent(new Event(String.*format*("Deleted timeslot at %d", time)));  
}

public List<String> getSchedule() {  
 List<String> output = new ArrayList<>();  
 for (int i = 9; i < 18; i++) {  
 if (allTimeSlots.get(i - 9).getStatus()) {  
 output.add("Free");  
 } else {  
 output.add(allTimeSlots.get(i - 9).getUserName());  
 }  
 }  
 return output;  
}

(TimeSlot.java)

public void delete() {  
 this.userName = "";  
 this.status = true;  
 EventLog.*getInstance*().logEvent(new Event("Deleted an old booking"));  
}

Bug-00006

**Title**: **Unable to save current state**

**Status**: Closed

**Product**: Study room scheduler

**Type**: Functional error

**Component**: Save data

**Version**: 1.0

**Platform**: PC Window

**Importance**: Minor

**Reported**: 2023-10-12 15:20:41 HKT by Beket Yermekov

**Modified**: 2023-10-15 11:12:51 HKT

**Assignee**: Nur Aldeken

Beket Yermekov 2023-10-12 15:20:41 HKT

**Problem:**

After performing several Booking/Deleting/Changing operations, I wanted to save the state, but it saves empty JSON file.

Steps to reproduce:

1)Run Main.java

2)Close the App

3)Save the file

4)Load the file

**Expected result:**

After saving the current state, the JSON file must have booking data.

**Actual result:**

After saving the current state, the JSON file is empty.

Nur Aldeken 2023-10-15 11:12:51 HKT

Issue solved:

New persistence implemented:

(JsonWriter.java) : Ensures safe saving of current project state.

package persistence;  
  
import model.ListRooms;  
import org.json.JSONObject;  
  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.PrintWriter;  
  
// Taken from the project JsonSerializationDemo  
// Represents a writer that writes JSON representation of workroom to file  
public class JsonWriter {  
 private static final int *TAB* = 4;  
 private PrintWriter writer;  
 private final String destination;  
  
 // EFFECTS: constructs writer to write to destination file  
 public JsonWriter(String destination) {  
 this.destination = destination;  
 }  
  
 public JsonWriter(File file) {  
 this.destination =  
 (file.getAbsolutePath().contains(".json")) ? file.getAbsolutePath() : file.getAbsolutePath() + ".json";  
 }  
  
 // MODIFIES: this  
 // EFFECTS: opens writer; throws FileNotFoundException if destination file cannot  
 // be opened for writing  
 public void open() throws FileNotFoundException {  
 writer = new PrintWriter(new File(destination));  
 }  
  
 // MODIFIES: this  
 // EFFECTS: writes JSON representation of workroom to file  
 public void write(ListRooms lr) {  
 JSONObject json = lr.toJson();  
 saveToFile(json.toString(*TAB*));  
 }  
  
 // MODIFIES: this  
 // EFFECTS: closes writer  
 public void close() {  
 writer.close();  
 }  
  
 // MODIFIES: this  
 // EFFECTS: writes string to file  
 private void saveToFile(String json) {  
 writer.flush();  
 writer.print(json);  
 }  
}

Bug-00007

**Title**: **Unable to load existing state**

**Status**: Closed

**Product**: Study room scheduler

**Type**: Functional error

**Component**: Load data

**Version**: 1.0

**Platform**: PC Window

**Importance**: Minor

**Reported**: 2023-10-12 15:25:20 HKT by Beket Yermekov

**Modified**: 2023-10-15 13:45:51 HKT

**Assignee**: Nur Aldeken

Beket Yermekov 2023-10-12 15:25:20 HKT

**Problem:**

After performing several Booking/Deleting/Changing operations, I wanted to save the state, but it saves empty JSON file.

Steps to reproduce:

1)Run Main.java

2)Click ‘File’ button on left top corner

3)Click ‘Load workspace’ button

4)Load your existing state (JSON file)

**Expected result:**

After loading the existing state, the app should contain several booking data.

**Actual result:**

After loading the existing state, the app does not contain past booking data

Nur Aldeken 2023-10-15 13:45:51 HKT

Issue solved:

New persistence implemented:

(JsonWriter.java): Ensures safe loading of existing project state.

private final String source;  
  
 // EFFECTS: constructs reader to read from source file  
 public JsonReader(String source) {  
 this.source = source;  
 }  
  
 // EFFECTS: constructs reader to read from source file  
 public JsonReader(File file) {  
 this.source = file.getAbsolutePath();  
 }  
  
  
 // EFFECTS: reads ListRooms from file and returns it;  
 // throws IOException if an error occurs reading data from file  
 public ListRooms read() throws IOException {  
 String jsonData = readFile(source);  
 JSONObject jsonObject = new JSONObject(jsonData);  
 return parseStudyRooms(jsonObject);  
 }  
  
 // EFFECTS: reads source file as string and returns it  
 private String readFile(String source) throws IOException {  
 StringBuilder contentBuilder = new StringBuilder();  
  
 try (Stream<String> stream = Files.*lines*(Paths.*get*(source), StandardCharsets.*UTF\_8*)) {  
 stream.forEach(s -> contentBuilder.append(s));  
 }  
  
 return contentBuilder.toString();  
 }  
  
  
 // EFFECTS: parses workroom from JSON object and returns it  
 private ListRooms parseStudyRooms(JSONObject jsonObject) {  
 ListRooms lr = new ListRooms("from Reader");  
 addRooms(lr, jsonObject);  
 return lr;  
 }  
  
 // MODIFIES: lr  
 // EFFECTS: parses StudyRooms from JSON object and adds them to ListRooms  
 private void addRooms(ListRooms lr, JSONObject jsonObject) {  
 JSONArray jsonArray = jsonObject.getJSONArray("data");  
 // String name = jsonObject.getString("name");  
 int i = 0;  
 for (Object json : jsonArray) {  
 JSONObject nextRoom = (JSONObject) json;  
 String nameR = nextRoom.getString("name");  
 lr.add(new StudyRoom(nameR));  
 addRoom(lr.get(i), nextRoom);  
 i++;  
 }  
 }

// MODIFIES: lr  
 // EFFECTS: parses StudyRoom's from JSON object and adds it to ListRooms  
 private void addRoom(StudyRoom sr, JSONObject jsonObject) {  
 JSONArray jsonArray = jsonObject.getJSONArray("TimeSlots");  
 int i = 0;  
 for (Object json : jsonArray) {  
 JSONObject nextTimeSlot = (JSONObject) json;  
 addTimeSlot(sr, i, nextTimeSlot);  
 i++;  
 }  
 }  
  
 // MODIFIES: lr  
 // Effects: parses TimeSlot's from JSON object and adds it to StudyRooms  
 private void addTimeSlot(StudyRoom sr, int i, JSONObject jsonObject) {  
 String username = jsonObject.getString("Username");  
 boolean status = jsonObject.getBoolean("status");  
 TimeSlot timeslot = new TimeSlot(username, status);  
 sr.changeTimeSlot(i, timeslot);  
 }  
}

Bug-00008

**Title**: **Deleting a certain time slot deletes all other time slots**

**Status**: Closed

**Product**: Study room scheduler

**Type**: Functional error

**Component**: Delete booking

**Version**: 1.0

**Platform**: PC Window

**Importance**: Minor

**Reported**: 2023-09-21 11:35:54 HKT by Beket Yermekov

**Modified**: 2023-09-21 18:15:14 HKT

**Assignee**: Nurdaulet Taumergenov

Beket Yermekov 2023-10-12 15:25:20 HKT

**Problem:**

After deleting a particular timeslot, bookings at all other timeslots disappear as well.

Steps to reproduce:

1)Run Main.java

2)Make multiple bookings by ‘Book study space’.

3)Delete one booking by ‘Delete booking’.

4)Display time slots by ‘Show Schedule’.

**Expected result:**

‘Delete booking’ button must ensure safe deletion of a timeslot, not affecting other booking data.

**Actual result:**

‘Delete booking’ button deletes all booking data.

Nurdaulet Taumergenov 2023-09-21 18:15:14 HKT

Issue solved: Interlinking TimeSlot and StudyRoom such that study room has its own set of time slot instances and TimeSlot implements delete function to ensure isolated removal of booking.

(StudyRoom.java)

public void deleteTimeSlot(int time) {  
 TimeSlot current = allTimeSlots.get(time - 9);  
 current.delete();  
 EventLog.*getInstance*().logEvent(new Event(String.*format*("Deleted timeslot at %d", time)));  
}

(TimeSlot.java)

public void delete() {  
 this.userName = "";  
 this.status = true;  
 EventLog.*getInstance*().logEvent(new Event("Deleted an old booking"));  
}

Bug-00009

**Title**: **Booking a certain room leads to booking of all other time slots**

**Status**: Closed

**Product**: Study room scheduler

**Type**: Functional error

**Component**: Book study space

**Version**: 1.0

**Platform**: PC Window

**Importance**: Minor

**Reported**: 2023-09-23 11:40:01 HKT by Beket Yermekov

**Modified**: 2023-09-23 20:18:04 HKT

**Assignee**: Nurdaulet Taumergenov

Beket Yermekov 2023-09-23 11:40:01

**Problem:**

After booking a particular timeslot, all other time slots get booked.

Steps to reproduce:

1)Run Main.java

2)Make **one** booking by ‘Book study space’.

3)Display time slots by ‘Show Schedule’.

**Expected result:**

‘Book study space’ button must ensure booking of one timeslot only at a time, not affecting other time slots.

A screen shot of a sign

Description automatically generated

**Actual result:**

Applying ‘Book study space’ to one time slot makes a booking to all other time slots.A screenshot of a computer screen

Description automatically generated

Nurdaulet Taumergenov 2023-09-21 18:15:14 HKT

Issue solved: Interlinking TimeSlot and StudyRoom such that study room has its own set of time slot instances and TimeSlot implements book function to ensure isolated booking of time slots.

(StudyRoom.java)

public void bookTimeSlot(int time, String name) {  
 TimeSlot current = allTimeSlots.get(time - 9);  
 current.book(name);  
 EventLog.*getInstance*().logEvent(new Event(String.*format*("Booked a timeslot for %s at %d", name, time)));  
}

(TimeSlot.java)

public void book(String name) {  
 this.status = false;  
 this.userName = name;  
 EventLog.*getInstance*().logEvent(new Event("Timeslot successfully added!"));  
}

Bug-00010

**Title**: [**CMD mode] Event logs are not displayed.**

**Status**: Closed

**Product**: Study room scheduler (CMD)

**Type**: Functional error

**Component**: EventLog

**Version**: 1.1

**Platform**: PC Window

**Importance**: Critical

**Reported**: 2023-10-14 18:12:41 HKT by Beket Yermekov

**Modified**: 2023-10-15 19:28:57 HKT

**Assignee**: Nurdaulet Taumergenov

Beket Yermekov 2023-10-14 18:12:41

**Problem:**

When we launch the app in cmd mode, events logs are not displayed.

Steps to reproduce:

1)Run Main.java in CMD mode

2)Make a following sequence of input numbers:

2.2) ‘2’ -> ‘Book a Study Room’

2.3) ‘1’ -> ‘Choose room “X100”’

2.4) ‘10’ -> ‘Choose a Time Slot’

2.5) ‘Name1’ -> ‘Type a name’

3)Enter ‘q’ as a quit command to terminate the system

**Expected result:**

After action is done, the system must display event log:

A screen shot of a computer

Description automatically generated

**Actual result:**

A screen shot of a computer

Description automatically generated

Nurdaulet Taumergenov 2023-10-15 19:28:57 HKT

Issue solved: Implemented a Singleton class EventLog that manages printing out event logs as needed:

package model;  
  
import java.util.ArrayList;  
import java.util.Collection;  
import java.util.Iterator;  
  
*/\*\*  
 \* Represents a log of alarm system events.  
 \* We use the Singleton Design Pattern to ensure that there is only  
 \* one EventLog in the system and that the system has global access  
 \* to the single instance of the EventLog.  
 \*/*public class EventLog implements Iterable<Event> {  
 */\*\* the only EventLog in the system (Singleton Design Pattern) \*/* private static EventLog *theLog*;  
 private Collection<Event> events;  
  
 */\*\*  
 \* Prevent external construction.  
 \* (Singleton Design Pattern).  
 \*/* private EventLog() {  
 events = new ArrayList<Event>();  
 }  
  
 */\*\*  
 \* Gets instance of EventLog - creates it  
 \* if it doesn't already exist.  
 \* (Singleton Design Pattern)  
 \** ***@return*** *instance of EventLog  
 \*/* public static EventLog getInstance() {  
 if (*theLog* == null) {  
 *theLog* = new EventLog();  
 }  
 return *theLog*;  
 }  
  
 */\*\*  
 \* Adds an event to the event log.  
 \** ***@param*** *e the event to be added  
 \*/* public void logEvent(Event e) {  
// events.add(e);  
 }  
  
 */\*\*  
 \* Clears the event log and logs the event.  
 \*/* public void clear() {  
 events.clear();  
 logEvent(new Event("Event log cleared."));  
 }  
  
 @Override  
 public Iterator<Event> iterator() {  
 return events.iterator();  
 }  
}

Bug-00011

**Title**: [**CMD mode] Change booking prints out log with empty username**

**Status**: Closed

**Product**: Study room scheduler (CMD)

**Type**: Functional error

**Component**: EventLog & Change Booking

**Version**: 1.1

**Platform**: PC Window

**Importance**: Critical

**Reported**: 2023-10-17 12:24:11 HKT by Beket Yermekov

**Modified**: 2023-10-18 15:23:45 HKT

**Assignee**: Nurdaulet Taumergenov

Beket Yermekov 2023-10-17 12:24:11 HKT

**Problem:**

When we launch the app in cmd mode, events logs are not displayed.

Steps to reproduce:

1)Run Main.java in CMD mode

2)Make a following sequence of input numbers:

2.2) ‘2’ -> ‘Book a Study Room’

2.3) ‘1’ -> ‘Choose room “X100”’

2.4) ‘10’ -> ‘Choose a Time Slot’

2.5) ‘Name1’ -> ‘Type a name’

3)Enter ‘q’ as a quit command to terminate the system

**Expected result:**

After action is done, the system must display event log:

A screen shot of a computer

Description automatically generated

**Actual result:**

A screen shot of a computer

Description automatically generated

Nurdaulet Taumergenov 2023-10-18 15:23:45 HKT

Issue solved: Singeton EventLog provides logEvent method that comprehensively registers new log for Booking.

public void bookTimeSlot(int time, String name) {  
 TimeSlot current = allTimeSlots.get(time - 9);  
 current.book(name);  
 EventLog.*getInstance*().logEvent(new Event(String.*format*("Booked a timeslot for %s at %d", name, time)));  
}