package org.apollo.template.View.UI;  
  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.layout.HBox;  
import javafx.scene.layout.Pane;  
import javafx.scene.layout.Priority;  
import javafx.scene.text.Font;  
import javafx.scene.text.FontPosture;  
import javafx.scene.text.FontWeight;  
import org.apollo.template.Model.AvailableRoom;  
  
  
public class AvailableComponent extends HBox {  
  
 private AvailableRoom availableRoom;  
 private Button button;  
  
  
 */\*\*  
 \* Builds the Available component  
 \* @param availableRoom the room object that is available for booking  
 \*/* public AvailableComponent(AvailableRoom availableRoom){  
  
 this.availableRoom = availableRoom;  
  
 *// creates topHbox* topHbox();  
 downHbox();  
   
 *// creates labels using the buildLabel methods* Label label\_roomNo = buildLabel(String.*format*("Lok. %s", availableRoom.getRoomName()),18, FontWeight.*BOLD*);  
 Label label\_floor = buildLabel(String.*format*("%s. Sal", availableRoom.getFloor()),18, FontWeight.*NORMAL*);  
 Label label\_personKap = buildLabel(String.*format*("Person kapacitet: %d", availableRoom.getPersonKapacity()), 18, FontWeight.*NORMAL*);  
 Label label\_roomType = buildLabel(String.*format*("Type: %s", availableRoom.getRoomType()), 16, FontWeight.*NORMAL*);  
   
  
 *// creates button with button text* button = createButton("BOOK");  
  
   
 *// sets up the main hBox* this.setMinHeight(60);  
 this.setPadding(new Insets(0, 20, 0, 0));  
 this.setAlignment(Pos.*CENTER*);  
 this.setStyle("-fx-background-color: #009FE3; -fx-background-radius: 40");  
 this.getChildren().addAll(label\_roomNo, addPane(),label\_floor, addPane(), label\_roomType, addPane(), label\_personKap, addPane(),button);  
   
 }  
  
  
 */\*\*  
 \* This method creates the label to be displayed in the HBox with the specified text, font size, and font weight.  
 \* @param text the text to be displayed in the label  
 \* @param fontSize the fontSize of the text  
 \* @param fontWeight the fontWeight of the text (e.g., FontWeight.NORMAL, FontWeight.BOLD)  
 \* @return a Label with a given text a given fontSize and a given fontWeight  
 \*/* private Label buildLabel(String text, int fontSize, FontWeight fontWeight) {  
  
 Label label = new Label(text);  
 label.setFont(Font.*font*("System", fontWeight, FontPosture.*REGULAR*, fontSize));  
 this.*setHgrow*(label, Priority.*ALWAYS*);  
 this.*setMargin*(label, new Insets(20, 20, 20, 20));  
 return label;  
 }  
  
  
 */\*\*  
 \* This method creates a button with a button text  
 \* @param buttonText the text to be displayed on the button  
 \* @return the created button  
 \*/* private Button createButton(String buttonText){  
 Button button = new Button(buttonText);  
 button.setPrefHeight(33.6);  
 button.setPrefWidth(100);  
 *//button.setFont(Font.font("System", FontWeight.BOLD, FontPosture.REGULAR, 18));* button.setStyle("-fx-background-color: #FBBB2C;");  
 return button;  
 }  
  
  
 */\*\*  
 \* This method creates a pane  
 \* @return the created pane  
 \*/* private Pane addPane (){  
 Pane pane = new Pane();  
 this.*setHgrow*(pane, Priority.*ALWAYS*);  
 return pane;  
 }  
  
  
 public Button getButton() {  
 return button;  
 }  
  
 *// TODO: with IT-udstyr og inventar* private void testInventory(){  
  
 String unicodeCheckMark = "\u2713";  
  
 String projector = "Projector";  
 String loudspeaker = "Højtaler";  
 String airtame = "Airtame";  
 String zoom = "Zoom";  
 String whiteBoaard = "Tavle";  
  
 *// creates labels using the buildLabel methods* Label label\_projector = buildLabel(String.*format*("[ %s %S ] ", projector, unicodeCheckMark),16, FontWeight.*NORMAL*);  
 Label label\_loudspeaker = buildLabel(String.*format*("[ %s %s ]", loudspeaker, unicodeCheckMark),16, FontWeight.*NORMAL*);  
 Label label\_airtame = buildLabel(String.*format*("[ %s %s ]", airtame, unicodeCheckMark), 16, FontWeight.*NORMAL*);  
 Label label\_zoom = buildLabel(String.*format*("[ %s %s ]", zoom, unicodeCheckMark), 16, FontWeight.*NORMAL*);  
 Label label\_whiteBoard = buildLabel(String.*format*("[ %s %s]", whiteBoaard, unicodeCheckMark), 16, FontWeight.*NORMAL*);  
  
  
  
 *// sets up the main hBox* this.setMinHeight(60);  
 this.setPadding(new Insets(0, 20, 0, 0));  
 this.setAlignment(Pos.*CENTER*);  
 this.setStyle("-fx-background-color: #009FE3; -fx-background-radius: 40");  
 this.getChildren().addAll(label\_projector, addPane(),label\_loudspeaker, addPane(), label\_airtame, addPane(), label\_zoom, addPane(),label\_whiteBoard);  
  
 }  
  
}