Sistema di tracciamento contatti per Ristoranti

Documentazione del codice sorgente dell'applicativo

Traccia 1 – Gruppo OOBD2122_22 – Borrelli Jonathan

L'applicativo permette di tenere traccia di tutti gli avventori di uno o più ristoranti, mostrando statistiche giornaliere, mensili e annuali sulla quantità totale di avventori di ogni ristorante.

1 INDICE

2 C	RC Cards	2
2.1	GUI	2
2.2	DAO	7
3 C	Convenzioni	13
3.1	Come impostare la foto di un ristorante	13
4 S	equence Diagrams	14
4.1	createHistogram()	14
4.2	inserisciTavolata()	15
5 C	Class diagram	17
5.1	class diagram senza dipendenze	17
5.2	Class diagram con dipendenze	19

2 CRC CARDS

2.1 GUI

JFrame
 DBConnector RistoranteDAO SalaDAO AvventoreDAO CameriereDAO ClientelaDAO ServizioDAO TavoloDAO ErrorMessage Home

	Home		JFrame
 Mostra l'elenco di ristoranti del database Permette di selezionare uno dei ristoranti Apre la finestra del ristorante scelto 		RistoranteFrameRistoranteRistoranteDAO	

RistoranteFrame		
 Avventore Ristorante SalaDAO Sala CameriereDAO Cameriere DBConnector ErrorMessage HistogramPanel ClientelaFrame 		

	HistogramPanel	JPanel
 Mostra un istogramma Può aggiungere un dato all'istogramma 		

JFrame

ClientelaFrame

- Può tornare indietro alla finestra RistoranteFrame
- Può aprire la finestra di inserimento di una nuova tavolata
- Sa il ristorante scelto
- Mostra un elenco selezionabile di tutte le date in cui il ristorante ha avuto clienti
- Mostra l'elenco di tavolate corrispondenti ad una certa data scelta dall'utente
- Per ogni Tavolata mostra cameriere che l'ha servita e avventori
- Può aprire una finestra di errore
- Può aggiornarsi nel caso di modi che nel database

- Clientela
- Ristorante
- NuovaTavolata
- ClientelaDAO
- ErrorMessage
- Servizio
- ServizioDAO
- Cameriere
- CameriereDAO
- Avventore
 - AvventoreDAO

	JFrame
NuovaTavolata	
 Può tornare indietro alla finestra ClientelaFrame Sa il ristorante scelto Può inserire una tavolata nel database Può aprire una finestra di errore Può aprire una finestra di inserimento avvenuto con successo Aggiorna ClientelaFrame con i nuovi avventori Aggiorna RistoranteFrame con i nuovi avventori 	 ClientelaFrame Ristorante ListaAvventoriPanel Tavolo TavoloDAO Cameriere CameriereDAO Avventore AvventoreDAO Servizio ServizioDAO InformazioniScorretteException ErrorMessage SuccessMessage RistoranteFrame

	JPanel
ListaAvvent	oriPanel
	NuovaTavolata
Mostra una lista di avventori	Avventore
Può aggiungere un avventore dalla lista	 InformazioniScorretteException
Può eliminare un avventore dalla lista	
Può svuotare tutta la lista	
Sa il numero di avventori presenti nella lista	
Sa gli avventori della lista	

	JDialog
ErrorMessage	
 Mostra un messaggio di errore Può tornare indietro alla finestra da cui si è generato l'errore 	
	JDialog
SuccessMessage	
 Mostra un messaggio di successo di un'operazione Può tornare indietro alla finestra da cui proviene l'operazione 	
	Exception
InformazioniScorretteException	
Sa il messaggio di errore	

2.2 DAO

DBConnector	
Crea la connessione con il database usando URL, username e password forniti	
dall'esterno	
Sa la connessione con il database	
Esiste massimo una sola istanza di questa classe	
luta of a co	
Interface DAO <t></t>	
DAONI	
Deve poter estrarre tutta la tabella di un Database	
Dato un certo ID deve poter estrarre la corrispondente riga dal database	
 Deve poter inserire una nuova riga nel database Deve poter eliminare una riga dal database 	
Deve poter eliminare una riga dai database	
	DAO <avventore></avventore>
AvventoreDAO	
Avventoredad	
	Avventore
Estrae tutti gli avventori dal Database	- Avenue
Può estrarre un singolo avventore	
Può inserire un avventore nel database	
Può rimuovere uno specifico avventore dal database	
Può controllare se un avventore esiste o no nel database	
Può inserire un gruppo di avventori	

Avventore	
 Sa il codice della carta d'identità Sa il nome Sa il cognome 	
Sa il numero di telefono	

		DAO <cameriere></cameriere>
CameriereDAO		
 Estrae tutti i camerieri dal Database Può estrarre un singolo cameriere Può inserire un cameriere nel database 	Cameriere Ristorante	
 Può rimuovere uno specifico cameriere dal database Può controllare se un cameriere esiste o no nel database Può ottenere tutti i camerieri di un dato ristorante 		

Camariara			
Cameriere	Cameriere		
Sa il codice della carta d'identità			
Sa il nome			
Sa il cognome			
Sa il numero di telefono			
Sa la data di nascita			
Sa la città			
Sa l'indirizzo			
Sa il numero civico			
• Sa il sesso			
Sa l'ID del ristorante in cui lavora			

		DAO <clientela></clientela>
ClientelaDAO		
 Estrae tutte le clientele dal Database Può estrarre una singola clientela Può inserire una clientela nel database Può rimuovere una specifica clientela dal database Può estrarre tutte le date in cui un ristorante ha avuto almeno un cliente Può estrarre tutti gli avventori di una specifica tavolata Può estrarre tutti i tavoli occupati in un certo giorno di uno specifico ristorante Può inserire una tavolata 	• Clientela • Ristorante	

Cliantala				
Clientela				
Sa il codice della carta d'identità del cliente				
Sa la data in cui è arrivato				
Sa il tavolo che ha occupato				
sa il tavolo che lla occupato				
	DAO <ristorante></ristorante>			
	DAONISIOI airte			
RistoranteDAO				
	Ristorante			
• Estrae tutti i ristoranti dal Database				
• Può estrarre un singolo ristorante				
Può inserire un ristorante nel database				
Può rimuovere uno specifico ristorante dal database				
	I			
Ristorante				
• Sa l'ID				
• Sa il nome				
• Sa la città				
Sa l'indirizzo				
Sa il numero civico				
Sa il numero di sale che contiene				
Sa l'orario di apertura				
Sa l'orario di chiusura				
Ha una foto e sa il nome del le di questa foto				
• The same roots of the months and the an question roots				

	DAO <sala></sala>
SalaDAO	
 Estrae tutte le sale dal Database Può estrarre una singola sala Può inserire una sala nel database Può rimuovere una specifica sala dal database Può estrarre tutte le sale di uno specifico ristorante 	SalaRistorante
Sala	
 Sa l'ID Sa il numero di tavoli che contiene Sa l'ID del ristorante 	
	DAO <servizio></servizio>
ServizioDAO	
 Estrae tutte le istanze di servizio dal Database Può estrarre una singola istanza di servizio Può inserire un'stanza di servizio nel database Può rimuovere una specifica istanza di servizio dal database 	• Servizio

<u></u>	
Servizio	
Sa il codice della carta d'identità del cameriere che ha servito	
Sa l'ID del tavolo che è stato servito	
Sa la data in cui è avvenuto il servizio	
	DAO <tavolo></tavolo>
TavoloDAO	
•	• Tavolo
• Estrae tutti i tavoli dal Database	Ristorante
Può estrarre un singolo tavolo Duò incomina un tavala mal database	
Può inserire un tavolo nel database Duò rimunyore une specifica tavale del database	
Può rimuovere uno specifico tavolo dal database Può estrarro tutti i tavoli di uno specifico ristoranto	
Può estrarre tutti i tavoli di uno specifico ristorante	
Tavolo	
Sa l'ID	
Sa il numero di posti disponibili massimi	
Sa l'ID della sala in cui appartiene	

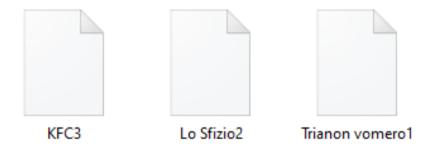
3 CONVENZIONI

3.1 COME IMPOSTARE LA FOTO DI UN RISTORANTE

Rinominare il file dell'immagine nel seguente modo:

<nome ristorante><ID ristorante>

Senza la desinenza del nome del file, quindi senza ".png" o ".jpeg" o altro Esempi:

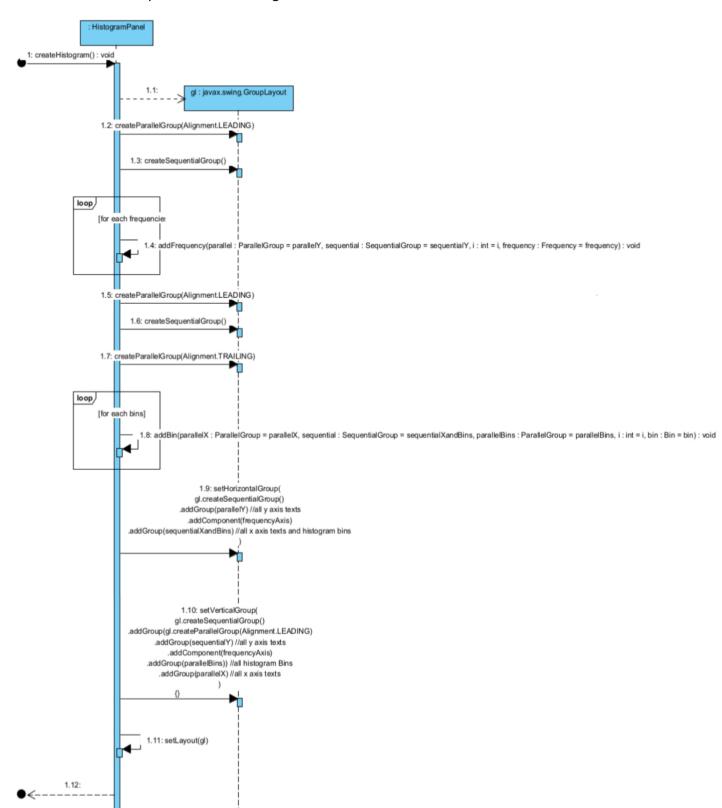


Dopodiché basterà inserirli all'interno della cartella "src/images/" e l'applicativo mostrerà quella foto in automatico.

4 SEQUENCE DIAGRAMS

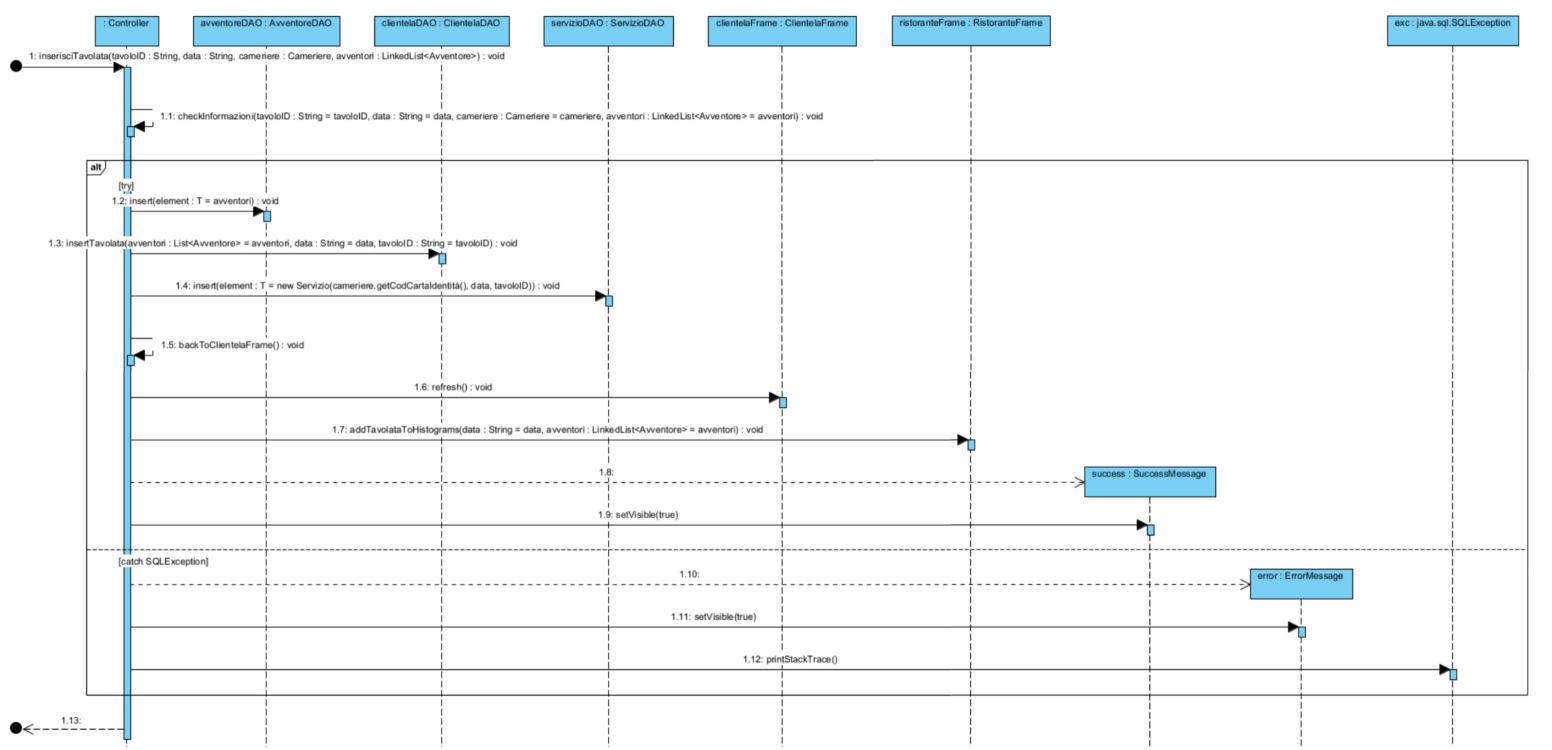
4.1 CREATEHISTOGRAM()

Costruisce bin per bin l'intero istogramma.



4.2 INSERISCITAVOLATA()

Data un'intera tavolata, Inserisce nel database le corrispondenti istanze di avventori, cameriere, clientela e servizio



5 CLASS DIAGRAM

5.1 CLASS DIAGRAM SENZA DIPENDENZE

DBConnector(String,String,String) SgetIstance(String,String,String):DBConnect SgetConnection():Connection screenSize: Dimension = Toolkit.getDefaultToolkit().getScreenSize() ⁵screenWidth: int = (int) screenSize.getWidth(screenHeight: int = (int) screenSize.getHeight() queryStatisticheAvventoriGiornalieri: String = "SELECT codR, EXTRACT(DAY FROM dataDiArrivo) AS giorno, EXTRACT(MONTH FROM dataDiArrivo) AS mese, EXTRACT(YEAR FROM dataDiArrivo) AS anno, dataDiArrivo, SUM(n°avventori) AS avventori_tot
+ "FROM Ristorante NATURAL JOIN Sala NATURAL JOIN Tavolo NATURAL JOIN Tavolata "
+ "GROUP BY codR, dataDiArrivo" queryStatisticheAvventoriMensili: String = "SBLECT codR. EXTRACT(YEAR FROM dataDiArrivo) AS anno, EXTRACT(MONTH FROM dataDiArrivo) AS mese, SUM(n°avventori) AS avventori_tot "
+ "FROM Ristorante NATURAL JOIN Sala NATURAL JOIN Tavolo NATURAL JOIN Tavolata "
+ "GROUP BY codR. EXTRACT(YEAR FROM dataDiArrivo), EXTRACT(MONTH FROM dataDiArrivo) "
queryStatisticheAvventoriAnnuali: String = "SBLECT codR. EXTRACT(YEAR FROM dataDiArrivo) AS anno, SUM(n°avventori) AS avventori_tot "
+ "FROM Ristorante NATURAL JOIN Sala NATURAL JOIN Tavolo NATURAL JOIN Tavolata "
+ "GROUP BY codR. EXTRACT(YEAR FROM dataDiArrivo) " o^SdayHistogramBins: ArrayList<String> oSdayHistogramFrequencies: ArrayList<Integer> oSmonthHistogramFrequencies: ArrayList<Integer> o^SyearHistogramBins: ArrayList<String> o^SyearHistogramFrequencies: ArrayList<Integer Controller() ScaleImage(JLabel,String):void ScaleImage(JButton,String):void SaveAllIntoFile():void getUrlFromFile():String SgetUsernameFromFile():String SgetPassw ordFromFile():String ScheckDataBase():void oSgoToClientelaFrame(Ristorante):void getInfoOf(Ristorante):String ScalculateHistograms(Ristorante):void SgetInfoOf(Clientela):String getInfoOfCameriereOf(String,String):String SbackToRistoranteFrame():void goToNuovaTavolata(Ristorante):void SgetInfoCamerieriOf(Ristorante):String[] osinserisciTavolata(String, String, Cameriere, LinkedList<Avventore>):void ScheckInformazioni(String,String,Cameriere,LinkedList<Avventore>):voice codCartaldentità: String **⊕** CameriereDAO cognome: String numeroDiTelefono: String -DBlogin 0.1 query: String dataDiNascita: String statement: Statement città: String □ table: ResultSet indirizzo: String OBLogin connection: Connection numeroCivico: String CameriereDAO(Connection) w idth: int = 650 exists(Cameriere):boolean height: int = 425 onotExists(Cameriere):boolean Cameriere(String, String, Stri x: int = Controller.screenWidth/2 - w idth/2 get(String):Cameriere ©Cameriere(String,String,String,String,String) y: int = Controller.screenHeight/2 - height/2 getCamerieriOf(Ristorante):List<Camerier</p> getCodCartaldentità():String contentPane: JPanel getAll():List<Cameriere> getNome():String url: JTextField insert(Cameriere):void getCognome():String username: JTextField delete(Cameriere):void getNumeroDiTelefono():String passw ord: JPassw ordField getDataDiNascita():String ricordaPassw ord: JCheckBox getCittà():String SoFDBinfoFilePath: String = System.getProperty("user.dir") + File.separator + "src\\saves\\DBinfo.txl getIndirizzo():String ©DBLogin() getNumeroCivico():String getUrl():String getSesso():String getUsername():String getRistorantelD():String getPassw ord():String ricordaPassw ordlsSelected():boolea +clientelaDAQ ○ Clientela DAO <u>+home</u> Ø..1 dateList: LinkedList<String> <<Java Class>> a query: String Clientela DAO u table: ResultSet w idth: int = 700 codCartaldentità: String connection: Connection -clientelaList height: int = 425 dataDiArrivo: String ClientelaDAO(Connection) x: int = Controller.screenWidth/2 - w idth/2 tavololD: String getAvventoriOf(String,String):List<Clientela> y: int = Controller.screenHeight/2 - height/2 Clientela(String,String,String) getAllTavoliOf(String):List<String> contentPane: JPanel getCodCartaldentità():String getAllTavoliOf(Ristorante,String):List<String> panel: JPanel = new JPanel() getDataDiArrivo():String getAllDistinctDates():List<String> getTavoloID():String getAllDistinctDatesOf(Ristorante):List<String> setScrollHeight(int):void getAll():List<Clientela> <<Java Class>> HistogramPanel insertTavolata(List<Avventore>,String,String):void delete(Clientela):void SoF axis Width: int = 15 axisHeight: int = 13 numberOfFrequencies: int = 20 numberOfBins: int = 29 +ristoranteFrame / SoFimageFrequencyAxisWidth: int = 29 imageBinAxisHeight: int = 26 imageTopBorderHeight: int = 26 imageBinWidth: int = 63 imageWidth: int = 1919 +salaDAO 0..1 $\frac{d}{dt} = \frac{dt}{dt}$ w idth: int = 700 imageHeight: int = 1116 **⊙** SalaDAO height: int = 425 frequencyAxisWidth: int = imageFrequencyAxisWidth x: int = Controller.screenWidth/2 - w idth/2 **⊙ Sala** DAO binAxisHeight: int = imageBinAxisHeight Fy: int = Controller.screenHeight/2 - height/2 query: String binWidth: int = imageBinWidth statement: Statement contentPane: JPanel width: int = imageWidth ID: String northPanel: JPanel table: ResultSet n height: int = imageHeight numeroDiTavoli: int southPanel: JPanel connection: Connection dataList: LinkedList<String> ristorantelD: String w estPanel: JPanel n frequencyAxis: JLabel SalaDAO(Connection) Sala(String,int,String) centerPanel: JTabbedPane -yearHistogram maxFrequency: int = numberOfFrequencies get(String):Sala getID():String indietro: JButton gl: GroupLayout getAll():List<Sala> getNumeroDiTavoli():in cronologiaClienti: JButton getAllOf(Ristorante):List<Sala:</p> getRistorantelD():String ©RistoranteFrame(Ristorante) insert(Sala):void HistogramPanel(int,int,int) ■ NorthPanel():void delete(Sala):void CHistogramPanel(int,int,int,String[]) ■ WestPanel():void CHistogramPanel(int,int,int,String[],int[]) CenterPanel():void CHistogramPanel(int,String[]) SouthPanel():void CHistogramPanel(int,String[],int[]) setButtonsEnabled(boolean):void GHistogramPanel(int,int,ArrayList<String>)
GHistogramPanel(int,int,ArrayList<String>,ArrayList<Integer>) getRistoranteScelto():Ristorante addTavolataToHistograms(String,LinkedList<Avventore>): getDayOf(Date):String FHistogramPanel(ArrayList<String>,ArrayList<Integer>) getMonthOf(Date):String add(String):void getYearOf(Date):String ■ Init(int):void Resize(int,int):void maxBinTitleWidth():int addResizeListener():void createHistogram():void □ ID: String addFrequency(ParallelGroup,SequentialGroup,int,Frequency):voice nome: String addBin(ParallelGroup,SequentialGroup,ParallelGroup,int,Bin):void città: String +ristoranteDAO 0..1 update():void ndirizzo: String <<Java Interface: maxFrequency():int numeroCivico: String ● DAO<T> generateNextMultipleOf_NumberOfFrequencies_Of(int):int <<Java Class> Ristorante DAO numeroDiSale: byte scalelmage(JLabel,String):void oraApertura: byte = 20 frequencies 0.*

«Java Class»

Frequency
Gui getAll():List<T> -bins 0..* query: String oraChiusura: byte = 22 insert(T):void Fw idth: int = 800 statement: Statement foto: String delete(T):void height: int = 600 u table: ResultSet CRistorante(String,String,String,String,String,byte) connection: Connection 11 **⊕ Bin** GUI -ristorante y: int = Controller.screenHeight/2 - height/2 ©RistoranteDAO(Connection getID():String contentPane: JPanel o title: JLabel get(String):Ristorante o title: JLabel getNome():String w estPanel: JScrollPane getAll():List<Ristorante> setNome(String):void o background: JLabel centerPanel: JScrollPane insert(Ristorante):void rectangle: JLabel getCittà():String addButton: JButton delete(Ristorante):void frequency: int setCittà(String):void closeButton: JButton getNumeroRistoranti():int getIndirizzo():String ^CBin() ClientelaFrame(Ristorante) setIndirizzo(String):void Bin(int) FillWestPanel():JPanel getNumeroCivico():String FillCenterPanel(String):JPanel setNumeroCivico(String):void setScrollRectangle(JPanel,int,int):void getNumeroDiSale():byte setButtonsEnabled(boolean):void getOraapertura():byte refresh():void getOrachiusura():byte getFoto():String +nuovaTavolata <<Java Class>> ● Nuova Tavolata ServizioDAO GUI oFw idth: int = 600 query: String statement: Statement codCameriere: String table: ResultSet x: int = Controller.screenWidth/2 - w idth/2 dataDiArrivo: String -servizioList connection: Connection y: int = Controller.screenHeight/2 - height/2 tavololD: String tavololD: String ServizioDAO(Connection Servizio(String,String,String) a camerierelD: String get(String,String):Servizio getCodCameriere():String numeroPostiDisponibili: int = 0 getAll():List<Servizio> getDataDiArrivo():String contentPane: JPanel insert(Servizio):void <<Java Class>> getTavololD():String numeroMaxAvventori: JLabel ■ Informazioni Scorrette Exception delete(Servizio):void data: JTextField nome: JTextField message: String = "La casella di testo non può essere vuota. Inserisci qualcosa e riprova" a cognome: JTextField of InformazioniScorretteException() numeroDiTelefono: JTextField codCartaldentità: JTextField getMessage():String aggiungi: JButton listaAvventoriScrollPane: JScrollPane o^CNuovaTavolata(Ristorante) ■ InserisciAvventoriPanel():void **⊙** SuccessMessage ■ aggiungiAvventore():void ⊕ ErrorMessage resetListaAvventori():void ■ inserisci():void w idth: int = 360 w idth: int = 360 getAggiungiButton():JButton height: int = 150 height: int = 150 contentPanel: JPanel = new JPanel -nuovaTavolata 0..1 contentPanel: JPanel = new JPanel() frame: JFrame frame: JFrame **⊙ AvventoreDAO**DAO SuccessMessage(JFrame,String) <<Java Class>> close():void Avventore a query: String statement: Statement -listaAvventoriPanel 0..1 n table: ResultSet codCartaldentità: String connection: Connection <<Java Class>> nome: String **⊜**ListaAvventoriPanel -avventori ©AvventoreDAO(Connection) a cognome: String numeroDiTelefono: String exists(Avventore):boolean scrollHeight: int = 10 Avventore(String, String, String, String) notExists(Avventore):boolea ©ListaAvventoriPanel(NuovaTavolata) getCodCartaldentità():String get(String):Avventore getAll():List<Avventore> addAvventore(String,String,String): getNome():String insert(Avventore):void getAll():LinkedList<Avventore> getCognome():String insert(List<Avventore>):void getNumeroAvventori():int getNumeroDiTelefono():String delete(Avventore):void clear():void +tavoloDAO 0..1 <<Java Class>> **⊙**TavoloDAO a query: String statement: Statement ⊕ Tavolo table: ResultSet connection: Connection tavololD: String -tavoli TavoloDAO(Connection) numeroMassimoDiAvventori: int salaID: String get(String):Tavolo

getNumeroMassimoDiAvventori():i

getID():String

getSalaID():String

getTavoliOf(Ristorante):List<Tavol</p>

getAll():List<Tavolo>

insert(Tavolo):void

delete(Tavolo):void

OBConnector
database

Sconnection: Connection = null

5.2 CLASS DIAGRAM CON DIPENDE	NZE
-------------------------------	-----

