

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Gestionale
Classe L8 – Ingegneria dell'Informazione



Sviluppo di un'applicazione per la gestione di agenzie viaggi in una struttura alberghiera

Relatore

Prof. Fulvio Corno

Candidato

Adriana Provenzano

234708

A.A. 2019/2020

Indice

1. Proposta del progetto.....	4
1.1. Descrizione del problema proposto.....	4
1.2. Descrizione della rilevanza gestionale del problema.....	4
1.3. Descrizione del data-set per la valutazione.....	4
1.4. Descrizione preliminare degli algoritmi coinvolti.....	4
1.5. Descrizione preliminare delle funzionalità previste per l'applicazione software	
2. Descrizione dettagliata del problema affrontato.....	6
2.1. Ricerca nel database.....	6
2.2. Proposta degli investimenti.....	7
3. Descrizione del data-set.....	8
3.1. Tabella "Agenzie"	8
3.2. Tabella "Referenti"	9
4. Diagramma delle classi principali.....	11
4.2. Il pacchetto "Agenzie"	11
4.3. Il pacchetto "DB"	13
4.4. Il pacchetto "Model"	13
5. Esempi di utilizzo dell'applicazione.....	17
5.1. Ricerca agenzie.....	17
5.2. Ricerca investimenti.....	19
5.3. Link del video Youtube.....	20
6. Valutazioni dei risultati e conclusioni.....	21

1 Proposta del progetto

1.1. Descrizione del problema proposto

L'obiettivo del progetto è quello di creare un'applicazione che permetta una semplice gestione dei contatti relativi alle agenzie viaggi da parte di una struttura alberghiera. Allo stesso tempo, si vuole anche offrire una proposta di investimento, su una selezione delle agenzie che possono portare i migliori vantaggi alla struttura alberghiera.

1.2. Descrizione della rilevanza gestionale del problema

Dato l'elevato numero di informazioni con cui una struttura alberghiera si deve interfacciare, l'organizzazione di questi dati mi è sembrato un problema di rilevanza gestionale. In particolare, la struttura di riferimento consultava tali informazioni attraverso fogli Excel e bigliettini da visita dei vari agenti. Dunque, ho visto l'ottimizzazione di questa consultazione come un ottimo spunto per mettere in pratica il lavoro svolto a lezione.

L'applicazione che si vuole realizzare ha come primo obiettivo quello di ridurre drasticamente il tempo con il quale si ottengono le informazioni richieste, da qualche minuto a pochi secondi.

Infine, è stato introdotto un ulteriore metodo che si occupa di consigliare alcuni investimenti su una selezione di agenzie su cui sarebbe preferibile puntare per ottenere maggiori guadagni.

1.3. Descrizione dei data-set per la valutazione

Il data-set utilizzato è stato creato sulla base di quello in possesso della struttura alberghiera: le informazioni contenute sono state rese del tutto anonime, con nomi fittizi e valori randomici.

In particolare, nel data-set utilizzato si trovano due tabelle contenenti una le informazioni proprie dell'agenzia e l'altra quelle relative al referente della stessa. Nella prima tabella sono elencate le diverse agenzie viaggi: ogni agenzia è identificata da un codice univoco, è qualificata da un nome e dal nome del suo referente. Le agenzie saranno ulteriormente caratterizzate con informazioni riguardanti la loro effettiva attività, il reale rapporto con la struttura e, infine, con una serie di valori booleani, che delineano i loro principali interessi e quelli dei loro clienti.

Nel data-set dei referenti, invece, saranno elencate le persone, rappresentanti delle agenzie incontrate durante le diverse fiere: oltre ad essere identificati da un codice univoco, i referenti saranno caratterizzati dal nome e cognome, dai loro contatti, dalla lingua parlata e dalla fiera in questione.

1.4. Descrizione preliminare degli algoritmi coinvolti

Nella realizzazione del programma verranno utilizzati sia degli algoritmi di ricerca che di ricorsione. Nel primo caso, i metodi saranno realizzati attraverso il pattern DAO e avranno l'obiettivo di permettere all'utente di ricercare le informazioni necessarie contenute nel data-set, quali le agenzie e le loro caratteristiche. Nel caso della ricorsione, questa sarà utilizzata per un metodo che aiuti l'utente a capire in che modo è meglio investire una parte del suo capitale destinato alle agenzie. In particolare, attraverso la ricorsione, verrà identificata la combinazione migliore delle agenzie conosciute.

1.5. Descrizione preliminare delle funzionalità previste per l'applicazione software

Interfacciandosi con l'applicazione, l'utente si troverà davanti a due scelte e quindi due diverse funzionalità.

Sulla schermata principale saranno disponibili vari metodi di ricerca, che potrà combinare a proprio piacimento per ottenere le informazioni richieste. In particolare, da un lato potrà consultare e ricercare nel data-set una specifica agenzia ricevendo in output tutte le sue informazioni possedute; dall'altro potrà selezionare una o più caratteristiche, ottenendo, in questo modo, come risultato anche diverse agenzie che soddisfano la ricerca.

Attraverso un bottone, si accederà alla seconda schermata, dedicata alla proposta degli investimenti. In questo caso, ricevendo in input dall'utente un valore in euro, il programma restituirà una collezione di agenzie–la cui somma di prezzi rientra nel budget. Tali agenzie risulteranno essere le migliori tra tutte quelle presenti nel database.

La superiorità di alcune agenzie su altre si basa sul confronto di un punteggio calcolato a partire dagli interessi e dall'attività dell'agenzia. In particolare, per evitare che nella selezione siano presenti solo agenzie già ben conosciute (in termini di preventivi e prenotazioni), si offre all'utente la possibilità di selezionare una percentuale della somma di denaro da investire, che sarà dedicata esclusivamente ad agenzie “nuove”, con cui non si sono mai avuti contatti diretti al di fuori delle fiere.

2 Descrizione dettagliata del problema affrontato

Il programma è stato realizzato mettendosi nei panni della struttura alberghiera. In particolare, attraverso l'uso dell'applicazione realizzata si vuole raggiungere una serie di obiettivi, quali:

- velocizzazione della consultazione del database
- ottimizzazione delle ricerche
- visione completa di tutti i possibili risultati della ricerca
- individuazione delle migliori agenzie e proposta di investimenti

Tali obiettivi sono raggiunti attraverso le due funzionalità principali dell'applicazione: la ricerca nel database e la proposta degli investimenti, di seguito riportate nel dettaglio.

2.1. Ricerca nel database

Nella sezione di ricerca, presente nella schermata principale del programma, all'utente è data la possibilità di consultare il database con un semplice click. In particolare, si troverà davanti una serie di bottoni tipici di una finestra JavaFX attraverso cui potrà formulare la sua ricerca.

The screenshot shows a search interface titled "RICERCA AGENZIE". It contains several input fields and controls:

- Seleziona agenzia:** A dropdown menu.
- Seleziona lingua:** A dropdown menu.
- Seleziona gli interessi:** A group of checkboxes including "Cicloturismo", "Enogastronomia", "LGBT", "Pet friendly", "Piccoli gruppi", "Lune di miele", "Arte & Cultura", "Benessere & Bellezza", and "MICE".
- Seleziona fiera:** A dropdown menu.
- Preventivi:** Radio buttons for "SI" and "NO".
- Num. prenotazioni:** A slider ranging from 0 to 4.
- Buttons:** "Reset" and "Ricerca" buttons are located on the right side. A "Ricerca Investimenti" button is located at the bottom right.

Le opzioni a disposizione dell'utente saranno due: selezionare il nome di un'agenzia e ricevere come risultato tutte le sue informazioni presenti nel database, oppure combinare a proprio piacimento diversi metodi di ricerca ottenendo una o più agenzie che rispondono a quei requisiti. Come si può osservare dalla finestra riportata sopra, l'utente avrà a disposizione diversi criteri di selezione quali: la lingua parlata dal referente, gli interessi, la

fiera in cui si è incontrato quell'agenzia, il fatto di aver richiesto o meno preventivi alla struttura alberghiera, oppure il numero di prenotazioni effettuate.

Questa funzionalità di ricerca può essere utilizzata dalla struttura per diversi motivi: ad esempio, per suddividere rapidamente le agenzie in base alla lingua parlata, al fine di poter strutturare delle campagne di email marketing negli idiomi di riferimento. Analogamente, attraverso pochi semplici passaggi, una promozione riservata a coppie in viaggio di nozze potrà essere inviata esclusivamente alle agenzie interessate, e cioè quelle che organizzano spesso lune di miele per i loro clienti.

2.2. Proposta degli investimenti

La seconda funzionalità riguarda la proposta di investimenti. Per quest'opzione verrà richiesto all'utente di inserire in un campo testo una somma di denaro, che rappresenterà la quota budget da destinare alle agenzie. Facendo in modo che la somma dei prezzi delle agenzie rientri nel denaro a disposizione, viene restituita all'utente una selezione di agenzie reputate dall'applicazione le migliori in termini di guadagno. Ciò viene calcolato sulla base di un punteggio che prende come riferimento il numero di prenotazioni, il fatturato, i preventivi e gli interessi caratteristici dell'agenzia. In questo modo però, risulta chiaro che nella selezione finale, saranno presenti solo agenzie "già conosciute", ossia che hanno effettuato prenotazioni generando un fatturato, oppure richiesto preventivi, in quanto caratterizzate da un punteggio più alto. Per evitare ciò, si lascia alla struttura alberghiera di riferimento la possibilità di puntare anche su agenzie "nuove": l'utente potrà selezionare la percentuale di budget da dedicare alle agenzie con cui non si hanno ancora avuto contatti diretti, se non durante le fiere del turismo. Dunque, inserendo il valore di denaro disponibile per questi investimenti, e interagendo con lo slider dedicato alle percentuali, l'utente riceverà il mix di agenzie, tra "vecchie" e "nuove", su cui l'applicazione riterrà conveniente puntare.

RICERCA INVESTIMENTI

Seleziona somma di denaro che si vuole investire: €

Seleziona percentuale che si vuole investire su agenzie "nuove"

3 Descrizione del data-set


Il data-set che viene utilizzato è stato generato in modo del tutto anonimo e randomico, privo di qualsiasi informazione personale, sia relative alle agenzie che ai loro referenti. In particolare, esso contiene lo stesso tipo di informazioni di cui la struttura dispone.

Il data-set è stato prima prodotto sotto forma di file CSV e subito dopo convertito in un file SQL. Nel data-set sono presenti due tabelle:

- la tabella delle *agenzie*, nella quale si trovano elencate tutte le agenzie viaggi conosciute dalla struttura
- la tabella dei *referenti*, incontrati durante le fiere del turismo e rappresentanti delle agenzie

3.1. Tabella “Agenzie”

Nella tabella “agenzie” si trovano le informazioni riguardanti le agenzie incontrate dalla struttura nel corso degli anni di attività. Ogni agenzia è identificata univocamente con un codice intero.

#	Nome	Tipo di dati
 1	Codice	INT
2	Nome_Agenzia	VARCHAR
3	Nome_Referente	VARCHAR
4	Preventivi_01	BIT
5	Num_Prenotazioni	INT
6	Fatturato_netto	INT
7	Cicloturismo	BIT
8	Enogastronomia	BIT
9	LGBT	BIT
10	Pet_friendly	BIT
11	Piccoli_gruppi	BIT
12	Lune_di_miele	BIT
13	ArteCultura	BIT
14	BenessereBellezza	BIT
15	MICE	BIT
16	Prezzo	INT

Oltre al codice, le agenzie sono caratterizzate dai seguenti attributi:

- *Nome_Agenzia*, stringa che indica il nome proprio dell’agenzia.
- *Nome_Referente*, stringa che indica il nome del rappresentante dell’agenzia con cui la struttura alberghiera è venuta a contatto durante una fiera. Questa caratteristica mi permette di collegare le due tabelle nel data-set in quanto attributo in comune.
- *Preventivi_01*, valore booleano che assume valore 0 se l’agenzia non ha mai richiesto un preventivo alla struttura, 1 altrimenti.
- *Num_Prenotazioni*, numero intero che è pari al numero di prenotazioni che l’agenzia ha finora effettuato presso la struttura.
- *Fatturato_Netto*, valore intero che indica il ricavo guadagnato dalla struttura grazie alle prenotazioni effettuate dall’agenzia.


A questi seguono una serie di valori booleani che hanno l'obiettivo di delineare gli interessi dell'agenzia e dei suoi clienti. Questi sono:

- *Cicloturismo*, settato a 1 se i clienti dell'agenzia sono particolarmente interessati ai viaggi su due ruote.
- *Enogastronomia*, settato a 1 se i clienti dell'agenzia sono interessati a scoprire il cibo e i vini del territorio in cui si trova la struttura.
- *LGBT*, settato a 1 se i clienti dell'agenzia fanno parte della comunità LGBT e richiedono una struttura per trascorrere la loro vacanza all'insegna del relax e della tranquillità.
- *Pet_friendly*, settato a 1 se i clienti dell'agenzia richiedono di poter portare nella struttura i loro amici a quattro zampe.
- *Piccoli_gruppi*, settato a 1 se i clienti dell'agenzia sono tipicamente coppie o piccole famiglie.
- *Lune_di_miele*, settato a 1 se i clienti dell'agenzia sono in viaggio di nozze
- *ArteCultura*, settato a 1 se i clienti dell'agenzia sono tipicamente interessati alle bellezze storico-culturali dei luoghi che vanno a visitare
- *BenessereBellezza*, settato a 1 se i clienti dell'agenzia sono alla ricerca del relax e del benessere fisico e psicologico raggiungibile attraverso terme
- *MICE (Meetings, Incentives, Conferences and Exhibitions)*, settato a 1 se i clienti dell'agenzia raggiungerebbero la struttura per riunioni, incentive tour, conferenze ed esposizioni.

L'ultimo campo che caratterizza ogni agenzia è *Prezzo*, un valore intero che indica il prezzo richiesto per una collaborazione annuale con la struttura.

3.2. Tabella "Referenti"

Nella tabella "referenti" si trovano le informazioni riguardanti i referenti, ossia i rappresentanti delle agenzie presenti nella tabella precedente. Anche in questo caso, per una più rapida gestione e per permettere la presenza di agenti omonimi, i referenti sono identificati univocamente con un codice.

#	Nome	Tipo di dati
 1	Codice	INT
2	Nome_e_cognome	VARCHAR
3	telefono	VARCHAR
4	email	VARCHAR
5	lingua	VARCHAR
6	fiera	VARCHAR

Gli agenti sono caratterizzati dai seguenti attributi:

- *Nome_e_cognome*, stringa che indica il nome e il cognome del referente, attraverso cui questa tabella è legata alla tabella delle agenzie.
- *telefono*, stringa che indica il contatto telefonico del referente.
- *email*, stringa che indica il contatto e-mail del referente.
- *lingua*, stringa di tre caratteri che indica la lingua parlata dal referente (es. ITA, ENG, SPA, ecc.).

- *fiera*, stringa in cui è indicato il codice, la città e l'anno in cui si è tenuta la fiera che ha permesso alla struttura e al referente dell'agenzia di conoscersi (es. WTM (Londra) 2018).

4 Diagramma delle classi principali

Per una migliore gestione dell'applicazione, è stato utilizzato il pattern MVC (Model-View-Controller), separando la logica di presentazione con cui si interfaccia l'utente, dalla logica di elaborazione che rende realmente operativa l'applicazione. Inoltre, si è voluto sfruttare anche il pattern DAO (Data-Access-Object) con l'obiettivo di mantenere separate tutte quelle operazioni che richiedono l'accesso alle tabelle del database attraverso delle query SQL.

Per questi motivi, il software è composto da tre pacchetti:

- *Agenzie*, è composto dalla classe *Main* (*Main.java*) che permette di avviare l'applicazione, e dalle due classi *Controller* (*AgenzieController.java*, *InvestimentiController.java*) che gestiscono le interfacce d'utente relative alla ricerca nel database delle agenzie e alla proposta degli investimenti. Infine, il pacchetto contiene i due corrispondenti file FXML (*AgenzieController.fxml*, *InvestimentiController.fxml*) attraverso cui viene definita l'interfaccia grafica dell'applicazione.
- *DB*, è composto dalle due classi che permettono di interagire con il database utilizzando il linguaggio SQL. La prima è la classe *DBConnect.java* in cui è definita la procedura per accedere al database "agenzieDB" e i cui metodi di accesso sono richiamati ogni volta in cui si vuole eseguire una query. L'altra è *AgenzieDAO.java*: essa contiene tutti i metodi, richiamati nel Model, attraverso cui l'applicazione trova le informazioni contenute nel database.
- *Model*, contiene le classi che definiscono la logica vera e propria dell'applicazione. Qui si trovano le classi *Agenzia.java* e *Referente.java* che utilizzano le informazioni contenute nel database per creare degli oggetti corrispondenti di entità rispettivamente Agenzia e Referente. Inoltre, in questo pacchetto è contenuta la classe *Model.java*, i cui metodi sono richiamati dal Controller per fornire le risposte che l'utente chiede all'applicazione.

4.1. Il pacchetto "Agenzie"

Come già anticipato, nel pacchetto Agenzie si trova la classe Main, responsabile del lancio del programma:

```
public class Main extends Application {
    @Override
    public void start(Stage primaryStage) {
        try {
            FXMLLoader loader = new FXMLLoader(getClass().getResource("AgenzieController.fxml"));
            BorderPane root = loader.load();
            AgenzieController controller=loader.getController();
            Model model=new Model();
            controller.setModel(model);
            Scene scene = new Scene(root);
            scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
            primaryStage.setScene(scene);
            primaryStage.show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

Come possiamo vedere, in questa classe viene indicato il file FXML visualizzato al momento dell'avvio del programma da parte dell'utente: la schermata principale sarà quella relativa alla funzionalità di ricerca nel database. Come loader viene settato quello relativo a AgenzieController.fxml; quindi viene collegato il relativo Controller e infine viene aggiunto il Model.

Nel pacchetto sono presenti anche le classi AgenzieController e InvestimentiController.

La prima è quella che gestisce le richieste dell'utente relative alla ricerca nel database: essa conterrà una serie di metodi che ricevono in ingresso i dati inseriti dall'utente, i quali vengono inviati al Model, richiamando i suoi metodi, e da questi ultimi riceverà i risultati che farà visualizzare all'utente.

In questa classe sono definiti diversi tipi di metodi. Una volta che il Model è stato "agganciato" al Controller attraverso il metodo `setModel`, viene chiamato il metodo `setComboItems`, che permette di popolare i menù a tendina dell'applicazione con le informazioni del database. Inoltre, la classe definisce al suo interno due ulteriori metodi: `doReset`, che permette di resettare tutte le scelte prese dall'utente e cancellare tutti i risultati, e `doRicerca`. Questo metodo richiama al suo interno una serie di metodi nel Model che permettono le seguenti funzioni:

- scelta un'agenzia, ottenere tutte le informazioni su quell'agenzia
- scelta una lingua, ottenere tutte le agenzie i cui referenti parlano quella lingua
- selezionati uno o più interessi, ottenere le agenzie che hanno almeno quegli interessi
- scelta una fiera, ottenere tutte le agenzie conosciute attraverso un loro referente durante quella fiera
- selezionato SI o NO, ottenere le agenzie che hanno o meno richiesto un preventivo
- scelto un numero di prenotazioni, ottenere tutte le agenzie che hanno effettuato almeno un numero di prenotazioni pari a quello indicato dall'utente.

Per evitare che l'utente selezioni un'agenzia e allo stesso tempo anche una lingua o un'altra caratteristica, la selezione dell'agenzia disabilita tutti gli altri controlli (ad eccezione di Reset e Ricerca) attraverso il metodo `doDisable`; allo stesso modo, la selezione di ogni caratteristica disabilita il menù a tendina relativo alla scelta dell'agenzia attraverso il metodo `doDisableAgenzia`.

La classe `InvestimentiController` è molto semplice: al suo interno sono stati definiti i metodi `doReset` e `doRicerca` simili a quelli illustrati precedentemente. Diversamente da quanto visto, il metodo `doRicerca` riceve in ingresso un valore in euro e un valore percentuale, invia questi dati al Model ottenendo in output la serie di agenzie su cui l'applicazione consiglia di investire, che verrà visualizzata dall'utente.

Da un'ultima analisi del pacchetto Agenzie possiamo osservare come l'applicazione passa da un'interfaccia all'altra, e cioè dalla funzionalità di ricerca a quella degli investimenti. Questo avviene per mezzo di due metodi che sono stati definiti uno nella classe `AgenzieController` e uno in `Investimenti Controller`.

Interagendo con il bottone "Ricerca Investimenti" in basso a destra nella schermata principale viene richiamato il metodo `doInvestimenti`:

```
void doInvestimenti(ActionEvent event) throws IOException {
    Stage stage = null;
    BorderPane root = null;
    stage = (Stage) btnInvestimenti.getScene().getWindow();
    FXMLLoader loader = new FXMLLoader(getClass().getResource("InvestimentiController.fxml"));
    root = loader.load();
    InvestimentiController controller=loader.getController();
    Model model=new Model();
    controller.setModel(model);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.show();
}
```

Questo metodo associa il file `InvestimentiController.fxml` a `InvestimentiController.java` e a `Model.java`. Di conseguenza, viene visualizzata la nuova interfaccia relativa agli investimenti. Analogamente, anche in quest'ultima ci sarà un bottone in basso a destra, "Ricerca Agenzie", attraverso cui verrà chiamato il metodo `doAgenzie`, che farà passare l'applicazione all'interfaccia relativa alla ricerca nel database.

4.2. Il pacchetto “DB”

Questo pacchetto, come anticipato, contiene i metodi necessari per accedere e interagire con il database attraverso delle query SQL. In particolare, qui si trova la classe DBConnect che permette l’accesso al data-set attraverso delle credenziali e utilizzando la libreria HikariCP. Oltre a questa classe, nel pacchetto DB è contenuta anche la classe AgenzieDAO che definisce tutti i metodi di ricerca richiamati dal Model attraverso delle query. Ad esempio, il metodo che è richiamato dal Model, che è a sua volta richiamato da AgenziaController, per popolare il menù a tendina delle lingue, è il seguente:

```
public LinkedList<String> loadLingue(){
    String sql = "SELECT DISTINCT lingua from referenti";
    LinkedList<String> result = new LinkedList<String>();

    try {
        Connection conn = DBConnect.getConnection();
        PreparedStatement st = conn.prepareStatement(sql);
        ResultSet rs = st.executeQuery();

        while (rs.next()) {
            result.add(rs.getString("lingua"));
        }

        conn.close();
        return result;
    } catch (SQLException e) {
        e.printStackTrace();
        System.out.println("Errore connessione al database");
        throw new RuntimeException("Error Connection Database");
    }
}
```

Come possiamo vedere, in tutti i metodi definiti in questa classe si seguono diversi passaggi:

1. viene definita la query SQL nella forma “SELECT ... FROM ... (WHERE ...)”
2. viene aperta la connessione con il database attraverso il metodo definito in DBConnect
3. viene associata la query alla connessione
4. si ottiene il risultato della query e lo si restituisce al Model sotto forma di lista

4.3. Il pacchetto “Model”

Il pacchetto Model contiene le classi Agenzia e Referente, definite con gli attributi corrispondenti alle informazioni contenute nel database:

```
public class Referente {

    private int codice;
    private String nome_cognome;
    private String telefono;
    private String email;
    private String lingua;
    private String fiera;
```

```

public class Agenzia {

    private int cod;
    private String nome;
    private Referente referente;
    private boolean preventivi;
    private int num_prenotazioni;
    private float fatturato;
    private boolean cicloturismo;
    private boolean enogastronomia;
    private boolean lgbt;
    private boolean pet_friendly;
    private boolean piccoli_gruppi;
    private boolean honey_moon;
    private boolean arte_cultura;
    private boolean benessere_bellezza;
    private boolean mice;
    private int prezzo;
    private float punteggio;
}

```

Le due classi contengono il rispettivo costruttore, opportuni metodi Getters e Setters e un metodo toString per descriversi in formato stringa. L'unica informazione non contenuta nel database è presente nella classe Agenzia ed è il punteggio: quest'informazione è calcolata dall'applicazione ed è utilizzata nella parte relativa agli investimenti per definire un confronto tra due agenzie in termini di guadagno.

Nella classe Model troviamo i seguenti attributi:

```

public class Model {

    AgenzieDAO daoAg;
    TreeMap <Integer, Agenzia> IdMapAg=new TreeMap <Integer, Agenzia> ();
    TreeMap <Integer, Referente> IdMapRe=new TreeMap <Integer, Referente> ();
    List <Agenzia> best1;
    List <Agenzia> best2;
    float best_guadagno1;
    float best_guadagno2;
}

```

Per una più semplice gestione dell'applicazione si sono definite due IdentityMap, una relativa alle agenzie e una relativa ai referenti, che vengono scambiate tra le classi AgenzieDAO e Model.

Tra i vari metodi contenuti nella classe Model, troviamo tutti quelli che fanno da intermediario tra AgenzieController e AgenzieDAO. Ad esempio, uno tra questi è il metodo che, ricevendo dal Controller il nome della fiera scelta dall'utente attraverso l'apposito menù a tendina, restituisce tutte le agenzie che si sono "incontrate" durante quella determinata fiera del turismo:

```

public LinkedList<Agenzia> cercaAgenzieDaFiera (String nomeFiera, LinkedList<Agenzia> risultatoAgenzie) {
    LinkedList <Agenzia> agenzie=new LinkedList <Agenzia> ();
    agenzie=daoAg.getAgenzieDaFiera(IdMapAg, nomeFiera);
    if(!risultatoAgenzie.isEmpty()) {
        return intersezioneListe(risultatoAgenzie, agenzie);
    }
    else
        return agenzie;
}

```

Come possiamo vedere, il Model invoca il metodo getAgenzieDaFiera definito nella classe AgenzieDAO passandogli come parametri il nome della fiera che ha ricevuto dal Controller e l'IdentityMap delle agenzie.

In tutti i metodi i risultati vengono mandati al Controller per essere visualizzati solo una volta dopo aver eseguito l'operazione di intersezione. Infatti, effettuando una ricerca combinata tra più caratteristiche, ad esempio

ricevendo una lista A in base alla lingua e una lista B in base agli interessi, l'utente visualizzerà solo le agenzie appartenenti sia alla lista A che alla lista B.

Nel Model sono definiti anche i metodi che permettono all'applicazione di suggerire all'utente alcuni investimenti. Quest'operazione è effettuata attraverso un algoritmo di ricorsione.

In particolare, il metodo principale richiamato da InvestimentiController è getInvestimenti, definito come segue:

```
public List<Agenzia> getInvestimenti(int spesa, int perc) {
    float p=(float) ((float)perc/100.0);
    float sommaNuove=p*spesa;
    float sommaVecchie=spesa-sommaNuove;

    List <Agenzia> nuove=daoAg.cercaPreventivi(IdMapAg, false);
    List <Agenzia> vecchie=daoAg.cercaPreventivi(IdMapAg, true);

    //calcola punteggio per ogni agenzia
    for(int i=0; i<nuove.size(); i++) {
        int num_interessi=calcolaInteressi(nuove.get(i));
        float punteggio=(float) (num_interessi*0.1);
        nuove.get(i).setPunteggio(punteggio);
    }
    for(int i=0; i<vecchie.size(); i++) {
        float punteggio=0;
        if(vecchie.get(i).isPreventivi())
            punteggio+=0.5;
        punteggio+=3*vecchie.get(i).getNum_prenotazioni()+vecchie.get(i).getFatturato()/100;
        int num_interessi=calcolaInteressi(vecchie.get(i));
        punteggio+=(float) (num_interessi*0.1);
        vecchie.get(i).setPunteggio(punteggio);
    }

    //avvio ricorsione
    best1=new ArrayList <Agenzia>();
    best2=new ArrayList <Agenzia>();
    ArrayList<Agenzia> parziale = new ArrayList<Agenzia>();
    cercaN(parziale, 0, sommaNuove, nuove);
    parziale = new ArrayList<Agenzia>() ;
    cercaV(parziale, 0, sommaVecchie, vecchie);
    List<Agenzia> best=new LinkedList<Agenzia> (best1);
    best.addAll(best2);
    return best;
}
```

La prima operazione svolta da questo metodo consiste nell'utilizzare i parametri inseriti dall'utente per separare la somma di denaro destinata alle agenzie "vecchie", cioè già conosciute al di fuori delle fiere del turismo, da quelle "nuove". Subito dopo, vengono definite le due liste in cui si dividono tutte le agenzie utilizzando il metodo cercaPreventivi della classe AgenzieDAO: il fatto di avere o meno richiesto preventivi alla struttura è usato come discriminante per far appartenere un'agenzia alla categoria delle "nuove" oppure delle "vecchie". Successivamente, per ogni agenzia viene calcolato il punteggio caratteristico che permetterà un confronto imparziale tra due agenzie. In particolare, il punteggio cresce di 0,1 per ogni interesse con cui l'agenzia si caratterizza, di 0,5 se ha richiesto preventivi, di 3 moltiplicato per il numero di prenotazioni effettuate e, infine, di un valore pari al fatturato diviso per 100.

L'ultima operazione, prima di avviare l'algoritmo di ricorsione, consiste nell'inizializzazione delle due liste best1 e best2 che alla fine saranno popolate dalle migliori agenzie, rispettivamente "nuove" e "vecchie", dato il budget indicato dall'utente.

A questo punto vengono richiamati i metodi cercaN e cercaV che eseguono l'algoritmo di ricorsione sulle liste "nuove" e "vecchie". I due metodi sono del tutto analoghi con l'unica differenza di ciclare su agenzie appartenenti a liste diverse e con un budget a disposizione diverso. Tali metodi sono così definiti:

```

private void cercaN(List<Agenzia> parziale, int i, float sommaNuove, List <Agenzia> nuove) {
    int prezzo=calcolaPrezzo(parziale);
    if(prezzo>sommaNuove)
        return;
    if(i==nuove.size())
        return ;

    float guadagno=calcolaGuadagno(parziale);
    if(guadagno>best_guadagno1) {
        best1=new ArrayList<Agenzia> (parziale);
        best_guadagno1=(float) guadagno;
    }

    List <Agenzia> agenzie=daoAg.getPrezzo(IdMapAg, ((int)sommaNuove-calcolaPrezzo(parziale)));
    if(agenzie.contains(nuove.get(i))) {
        parziale.add(nuove.get(i));
        cercaN(parziale, i+1, sommaNuove, nuove);
        parziale.remove(nuove.get(i));
    }
    cercaN(parziale, i+1, sommaNuove, nuove);
}

```

Nel metodo di ricorsione, la prima cosa che viene fatta è considerare i casi limite: infatti, se il prezzo che si spenderebbe con l'attuale parziale è superiore al budget a disposizione oppure se si è arrivati alla fine della lista di agenzie, si esce dal metodo. Dopodiché si calcola il guadagno totale del parziale: se questo è migliore di quello trovato finora, allora salvo la combinazione di agenzie come la migliore (best1) e memorizzo il suo guadagno. Infine mando avanti la ricorsione: questo non avviene con tutte le possibili agenzie presenti nella lista. Infatti, per cercare di ridurre la pesantezza dell'algoritmo si è posta una condizione: si trovano tutte le agenzie il cui prezzo rientra ancora nel budget a disposizione e se un'agenzia non appartiene a questa lista allora non verrà inserita nel parziale.

5 Esempi di utilizzo dell'applicazione

Di seguito è riportato un esempio di utilizzo dell'applicazione.

5.1. Ricerca agenzie

In questo esempio supponiamo che l'utente voglia ricevere le informazioni relative all'agenzia viaggi "Abracadabra" quindi la seleziona dall'apposita tendina. Come possiamo vedere, una volta che l'utente ha selezionato l'agenzia tutti gli altri controlli dell'interfaccia vengono disabilitati.

RICERCA AGENZIE

Seleziona agenzia:

Seleziona lingua:

Seleziona gli interessi

☐ Cicloturismo ☐ Pet friendly ☐ Arte & Cultura

☐ Enogastronomia ☐ Piccoli gruppi ☐ Benessere & Bellezza

☐ LGBT ☐ Lune di miele ☐ MICE

Seleziona fiera:

Preventivi ☐ SI ☐ NO Num. prenotazioni

Cliccando sul bottone "Ricerca" vengono stampate nell'apposita TextArea tutte le informazioni presenti nel database e relative all'agenzia Abracadabra. In particolare, verranno stampati il nome e il cognome del referente, con il suo contatto email, il suo contatto telefonico, la lingua parlata e la fiera del turismo in cui ci si è incontrati. Dopodiché seguirà una serie di interessi che caratterizzano i clienti dell'agenzia Abracadabra e infine è riportato il prezzo richiesto dall'agenzia per la collaborazione annuale con la struttura.

RICERCA AGENZIE

Seleziona agenzia:

Seleziona lingua:

Seleziona gli interessi:

- ☐ Cicloturismo
- ☐ Enogastronomia
- ☐ LGBT
- ☐ Pet friendly
- ☐ Piccoli gruppi
- ☐ Lune di miele
- ☐ Arte & Cultura
- ☐ Benessere & Bellezza
- ☐ MICE

Seleziona fiera:

Preventivi: ☐ SI ☐ NO

Num. prenotazioni:

Agenzia: Abracadabra

Referente: Abi Medrano Montez (email: AbiMedranoMontez@teleworm.us, tel.: 787 779 789, lingua: TED, fiera: ITB (Berlino) 2018)

Interessi [Arte e Cultura, Benessere e Bellezza, Enogastronomia, Luna di Miele, LGBT,]

Prezzo: 1519€

Numero risultati: 1

Supponiamo invece che l'utente voglia eseguire un altro tipo di ricerca interagendo con più controlli dell'interfaccia: ad esempio supponiamo che voglia cercare tutte le agenzie con cicloturismo tra gli interessi e con almeno 2 prenotazioni effettuate:

RICERCA AGENZIE

Seleziona agenzia:

Seleziona lingua:

Seleziona gli interessi:

- ☒ Cicloturismo
- ☐ Enogastronomia
- ☐ LGBT
- ☐ Pet friendly
- ☐ Piccoli gruppi
- ☐ Lune di miele
- ☐ Arte & Cultura
- ☐ Benessere & Bellezza
- ☐ MICE

Seleziona fiera:

Preventivi: ☐ SI ☐ NO

Num. prenotazioni:

Agenzia: Mambo Salentino

Referente: Isabela Ribeiro Souza (email: IsabelaRibeiroSouza@teleworm.us, tel.: 21 222 183 9610, lingua: POR, fiera: PLE (Marrakech) 2018)

L'agenzia ha richiesto preventivi e ha effettuato 4 prenotazioni, facendo ottenere un fatturato netto di 8000.0 €.

Interessi [Arte e Cultura, Benessere e Bellezza, Cicloturismo, Enogastronomia, Luna di Miele, LGBT, MICE,]

Prezzo: 1843€

Agenzia: Nómada Tours

Referente: Júlio Araujo Alves (email: JulioAraujoAlves@dayrep.com, tel.: 21 227 660 3911, lingua: POR, fiera: PLE (Marrakech) 2018)

L'agenzia ha richiesto preventivi e ha effettuato 3 prenotazioni, facendo ottenere un fatturato netto di 6900.0 €.

Interessi [Arte e Cultura, Benessere e Bellezza, Cicloturismo, LGBT, Piccoli gruppi]

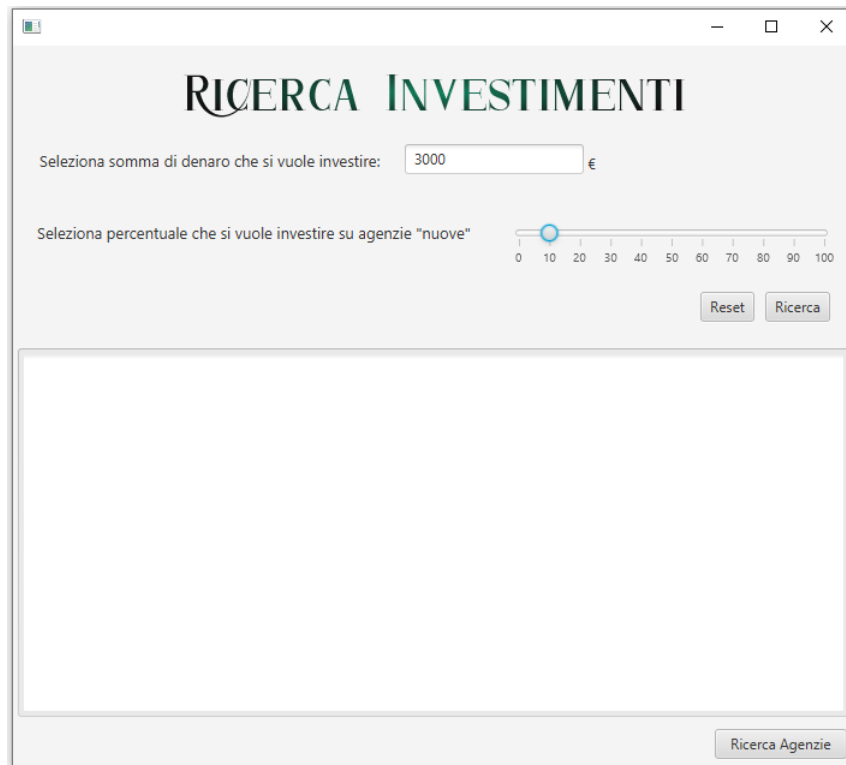
Prezzo: 1221€

Numero risultati: 4

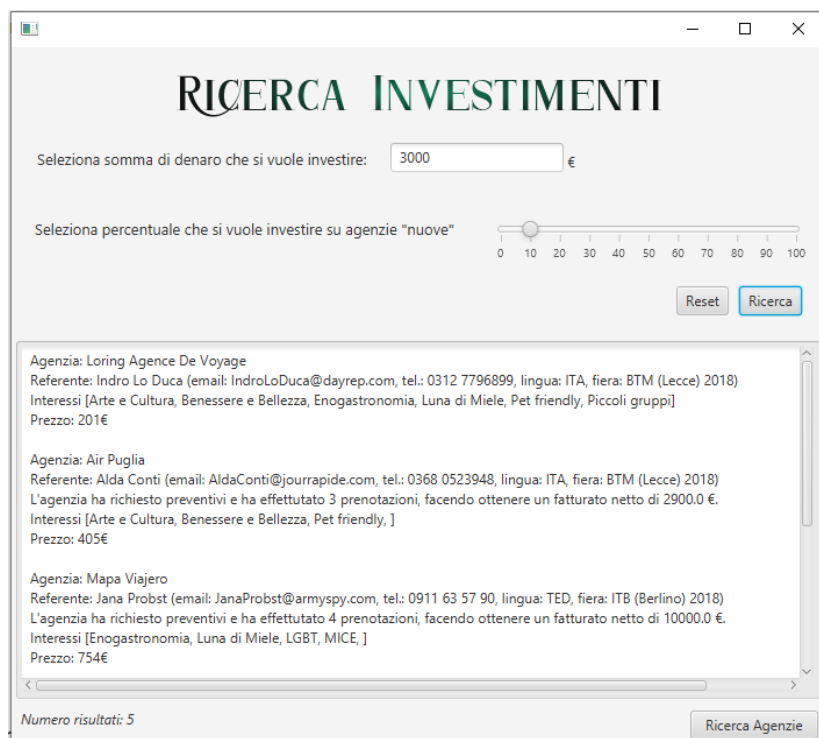
Come indicato dall'etichetta in basso a sinistra la ricerca ha restituito quattro agenzie che rispondono ai requisiti inseriti dall'utente: hanno tutte effettuato almeno 2 prenotazioni e tra gli interessi dei loro clienti c'è il cicloturismo.

5.2. Ricerca investimenti

Attraverso il bottone in basso a destra “Ricerca Investimenti” possiamo passare all’interfaccia relativa alla ricerca degli investimenti. All’utente, in questo caso sarà richiesto di inserire una cifra in euro ed una percentuale:



Se ad esempio dichiara di voler investire 3000 di cui il 10% nelle agenzie nuove, ciò significa che 300€ verranno dedicati alle agenzie con cui non si hanno avuto contatti diretti al di fuori delle fiere del turismo, mentre 2700€ saranno per le agenzie “vecchie”. Se si avvia la ricerca clickando sul bottone “Ricerca” quello che verrà stampato sarà il seguente:



RICERCA INVESTIMENTI

Seleziona somma di denaro che si vuole investire: 3000 €

Seleziona percentuale che si vuole investire su agenzie "nuove" 10

Reset Ricerca

Agenzia: Loring Agence De Voyage
Referente: Indro Lo Duca (email: IndroLoDuca@dayrep.com, tel.: 0312 7796899, lingua: ITA, fiera: BTM (Lecce) 2018)
Interessi [Arte e Cultura, Benessere e Bellezza, Enogastronomia, Luna di Miele, Pet friendly, Piccoli gruppi]
Prezzo: 201€

Agenzia: Air Puglia
Referente: Alda Conti (email: AldaConti@jourrapide.com, tel.: 0368 0523948, lingua: ITA, fiera: BTM (Lecce) 2018)
L'agenzia ha richiesto preventivi e ha effettuato 3 prenotazioni, facendo ottenere un fatturato netto di 2900.0 €.
Interessi [Arte e Cultura, Benessere e Bellezza, Pet friendly,]
Prezzo: 405€

Agenzia: Mapa Viajero
Referente: Jana Probst (email: JanaProbst@armyspy.com, tel.: 0911 63 57 90, lingua: TED, fiera: ITB (Berlino) 2018)
L'agenzia ha richiesto preventivi e ha effettuato 4 prenotazioni, facendo ottenere un fatturato netto di 10000.0 €.
Interessi [Enogastronomia, Luna di Miele, LGBT, MICE,]
Prezzo: 754€

Numero risultati: 5

Ricerca Agenzie

Come possiamo notare l'agenzia Loring Agence De Voyage è l'unica agenzia "nuova" che è rientrata nel mix di agenzie con un prezzo di 201€, mentre le altre sono tutte classificabili come "vecchie".

5.3. Link del video YouTube

Il link al video di esempio di utilizzo dell'applicazione caricato su YouTube è il seguente:

<https://www.youtube.com/watch?v=BoCXf0PKGb0>

6 Valutazione dei risultati e Conclusioni

L'obiettivo dell'applicazione è quello di fornire ad una struttura alberghiera un modo semplice e veloce per gestire le informazioni a disposizione. Quest'approccio ha il vantaggio di essere conveniente sotto molti punti di vista: tra le altre, ha la qualità di dare una visione a 360° su tutti i contatti in modo da poter sfruttare tutte le occasioni possibili e non tralasciare nessuna opportunità.

Per quanto riguarda la proposta degli investimenti, questa offre sicuramente degli ottimi candidati per iniziare o eventualmente continuare una collaborazione con la struttura. Ciò non toglie però che ci possano essere degli altri fattori, aleatori e imprevedibili, che possano pregiudicare la convenienza di un investimento.

Un ulteriore problema riscontrato nello sviluppo dell'applicazione, e in particolar modo la parte degli investimenti, riguarda la pesantezza dell'algoritmo di ricorsione utilizzato in questa fase. Come ho illustrato nella sezione apposita questo è stato risolto inserendo una condizione e andando quindi a diminuire il numero di ricorsioni. Questa soluzione risulta essere però solo parziale e non sufficiente all'aumentare della popolazione del database.

Tenendo conto di tutti questi fattori, l'applicazione risulta comunque molto utile alla struttura di riferimento che potrà sostituire la consultazione di bigliettini da visita dei referenti e di fogli Excel con una consultazione decisamente adeguata alla mole di informazioni, e sicuramente più immediata e completa.