UNIVERSITA' DEGLI STUDI DI
NAPOLI FEDERICO II

Scuola Politecnica e delle Scienze di Base
Corso di Laurea Triennale in Informatica

Corso di Laboratorio di Algoritmi e Strutture Dati

Progettazione e sviluppo di una Agenzia di Viaggi in linguaggio C

Anno Accademico 2021/2022

i docenti
Prof. Aniello Murano
Dott. ssa Silvia Stranieri

gli studenti
Erasmo Prosciutto
Antonio Lanuto
Biagio Scotto

matr. in ordine
N86003546
N86003762
N86003605

Contents

Chapter 1

Introduzione

Si vuole sviluppare un servizio di Booking per viaggi in treno/aereo e Hotel, dove distinguiamo due figure principali: utente e amministratore. Per la fruizione del servizio è fondamentale la registrazione e l'accesso. Stessa cosa vale per gli amministratori per avere la possibilità di gestire il servizio.
Il servizio sarà chiamato "Erampi-Viaggi".

1.1 Sezione Utente

Un utente, dopo essersi registrato, può accedere al sistema e prenotare un viaggio specificando la partenza e la destinazione da una lista di mete.
E' possibile scegliere di viaggiare in treno o in aereo e per ciascuna opzione di viaggio possono essere visualizzate:

- le opzioni più economiche
- le opzioni più veloci

Conclusa la selezione del viaggio, il sistema propone all'utente una lista di Hotel disponibili nella città di destinazione. Dopo questa scelta, il sistema propone il modo più veloce per raggiungere l'Hotel partendo dalla stazione/aeroporto.

1.2 Sezione Admin

Qualora la destinazione di viaggio selezionata dall'utente non fosse raggiungibile in alcun modo, il sistema notificherà il prossimo amministratore al suo accesso per:

- Rendere raggiungibile quella meta
- Eliminare quella meta

1.3 Suddivisione del lavoro

- Erasmo Prosciutto:
  - scelte implementative
  - implementazione di una struttura grafo per la gestione degli Hotel.
  - implementazione delle funzioni di creazione grafo e archi, mappa degli Hotel, algoritmo di Dijsktra per scelta del percorso più veloce.
  - implementazione delle tipologie di viaggio treno/aereo
  - implementazione funzionalità di accesso e registrazione
  - Creazione mappe dei grafi in UML
  - Scrittura del documento Latex
- Antonio Lanuto:
  - implementazione delle funzioni dell'Admin
  - implementazione di una struttura grafo per la gestione dei viaggi, partenze e destinazioni.
  - implementazione dell'algoritmo di Dijsktra per opzione viaggio più veloce e più economico

  - implementazione algoritmi di visita dei grafi
  - implementazione delle grafiche e dei colori
  - implementazione delle funzionalità di ricarica/svuota saldo

- Biagio Scotto:
  - scelte implementative
  - implementazione delle funzioni per i grafi degli Hotel
  - implementazione della funzione acquista relativa agli Hotel
  - implementazione delle grafiche e dei colori
  - implementazione dell'algoritmo di Dijsktra per scelta del percorso più veloce negli Hotel
  - Creazione mappe dei grafi in UML
  - Scrittura del documento Latex

Chapter 2

Scelte implementative

Nel seguente capitolo verranno descritte le principali strutture dati utilizzate e le loro relative implementazioni con le varie funzioni di gestione, oltre all'organizzazione generale dei file.
Inoltre verranno descritti i principali elementi e funzioni caratterizzanti l'applicativo.

2.1 Divisione File

Il lavoro è stato svolto tramite Visual Studio Code con una repository condivisa di GitHub.
Ogni elemento principale è stato suddiviso in più file, rispettivamente .h e .c, al fine di separare al meglio le funzioni di ognuno e di permettere una più rapida e pulita lettura.

Sono presenti i seguenti file:

1. *gestioneFile.c*
2. *gestioneFile.h*
3. *gestioneUtente.c*
4. *gestioneUtente.h*
5. *Main.c*
6. *Grafi_Partenze_Destinazioni.c*
7. *Grafi_Partenze_Destinazioni.h*
8. *GrafiHotel.c*
9. *GrafiHotel.h*
10. *StruttureAccessorie.h*
11. *GestioneAccesso.c*
12. *GestioneAccesso.h*
13. *GestioneGrafica.c*
14. *Gestione Grafica.h*
15. *Città.txt*
16. *Notifiche.txt*
17. *Utenti.txt*

2.2 Strutture dati

Di seguito sono riportate le principali strutture dati utilizzate.

2.2.1 Grafi

Struttura grafo utilizzata per mappare le città disponibili (partenza-destinazione)

- struttura *edge* per rappresentare la lista di adiacenza con pesi per rappresentazione scelta più economica e più veloce per treno/aereo
- struttura *graph* con il numero dei vertici ed il collegamento alla struttura *edge*

2.2.2 GrafiHotel

Struttura grafo utilizzata per mappare i percorsi disponibili da stazione treni / aeroporto ad Hotel

- struttura *edge* per rappresentare la lista di adiacenza con pesi per rappresentazione distanza tra i vari punti, partendo dalla stazione o dall'aeroporto
- struttura *graph* con il numero dei vertici ed il collegamento alla struttura *edge*

2.3 Funzioni cardine

Sono descritte ora le funzioni principali, rappresentanti il nucleo dell'applicativo.

2.3.1 GestioneUtente

1. void acquista(struct Utente *utenteMain,graph *G,graphHotel *G_Hotel,Stringa listacitta[]);

   - gestisce la creazione del viaggio: permette la scelta della città di partenza e di destinazione, la scelta del mezzo di trasporto ,controlla l'effettivo collegamento tra le due e successivamente permette la scelta tra viaggio veloce ed economico; infine permette la scelta di un Hotel, mostrando il percorso più veloce

2. void modifica(graph *G,Stringa arrayCitta[]);

   - nel caso in cui una città di partenza non abbia collegamenti, l'Admin può aggiungere quest'ultimi oppure rimuovere la città

2.3.2 Grafi

1. void DFSClassic/Treno/Aereo(graph* graph, int vertice,int cittaVisitate[]);

   - si occupa di verificare quali sono le città raggiungibili in modalità treno e aereo

2. int dijsktra(graph *G,int source,int target,Stringa Citta[],Stringa path[],int mod);

   - Tramite l'algoritmo di Dijkstra individuiamo i cammini minimi in un grafo, in base al peso relativo alla distanza e/o a quello relativo al prezzo dei viaggi

2.3.3 GrafiHotel

1. void scegliHotel(int partenza, int sceltaModalita);

    - data la città di partenza e il tipo di mezzo di trasporto, la funzione scegliHotel si occupa di mostrare gli Hotel disponibili per la suddetta città e mostra il percorso più veloce per raggiungerlo

2. int dijsktraHotel(graph *G,int source,int target,Stringa Citta[],Stringa path[],int mod);

    - Tramite l'algoritmo di Dijkstra individuiamo i cammini minimi in un grafo, in base al peso relativo alla distanza Stazione/Aeroporto - Hotel

Chapter 3

Funzionamento del sistema

Una volta avviata l'applicazione, l'utente può scegliere se accedere o registrarsi. In caso di registrazione, è possibile eseguirla come Admin o come Utente; il sistema chiederà: nome, cognome, email, password, saldo iniziale.
In caso di accesso come Utente, egli inserirà email e password e potrà scegliere tra varie funzioni: acquista, ricarica, svuota conto, esci.

- **Acquisto**: il sistema mostra a video una lista di città disponibili per la partenza e per la destinazione (in base alla disponibilità dalla città di partenza), selezionabili tramite un identificativo numerico.
  - Nel caso in cui dalla città di partenza selezionata non fosse possibile raggiungere alcuna destinazione, il sistema invierà una notifica all'Admin, il quale potrà scegliere di: aggiungere destinazioni partendo dalla presente città, oppure eliminare la città.
  - Nel caso in cui il processo di scelta partenza-destinazione si sia concluso correttamente, il sistema chiederà all'utente che tipo di mezzo utilizzare per il viaggio (mostrando solamente quelli realmente disponibili in base a partenza-destinazione).

Successivamente, a seconda del mezzo di trasporto scelto, l'utente potrà filtrare i viaggi per: percorso più veloce o percorso più economico. A scelta effettuata, con l'avvenuto acquisto, l'utente potrà decidere se prenotare anche un Hotel.

- **Ricarica**: permette di ricaricare il conto.
- **Svuota**: permette di prelevare il denaro depositato e azzerare il conto.

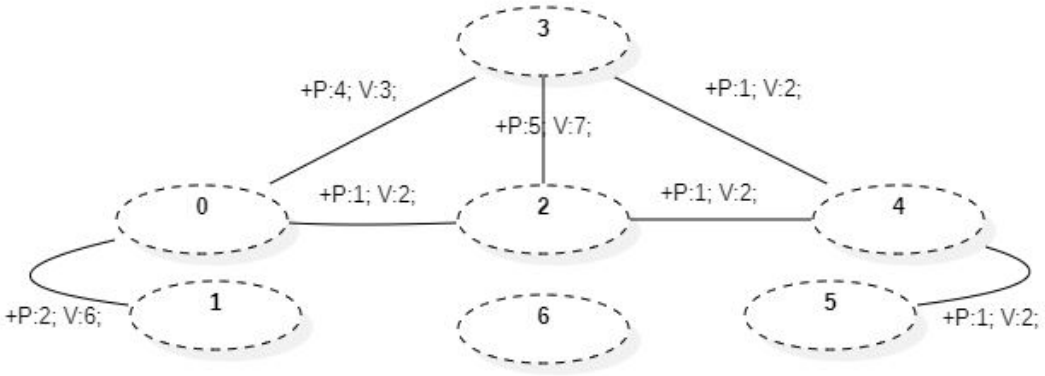
In caso di accesso come Admin, egli inserirà email e password e potrà scegliere di effettuare due tipi di operazioni: Esci Account e Modifica.

- **Modifica**: tramite questa funzione l'Admin potrà eseguire due operazioni:
  - Eliminare le città che non hanno collegamenti con altre
  - Aggiungere collegamenti alle città non collegate

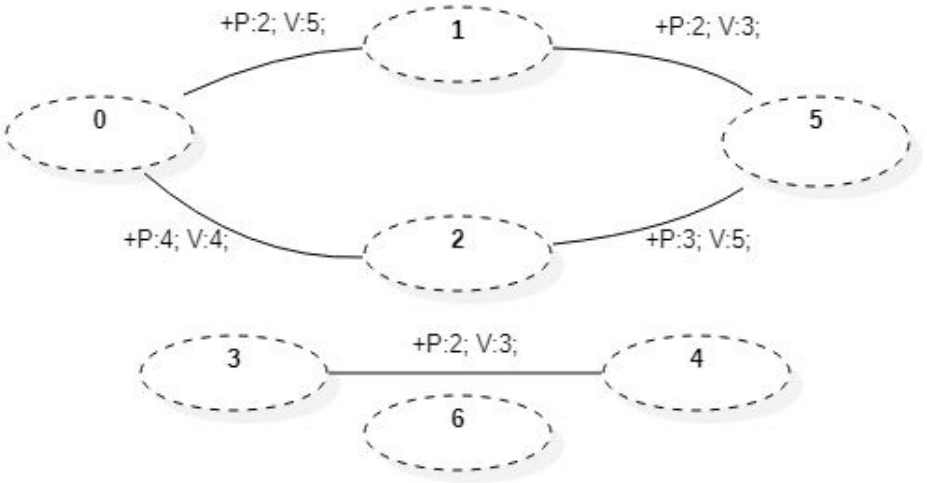
Chapter 4

Mappe dei Grafi

4.1 Percorsi Aerei



4.2 Percorsi Treni



4.3 Percorsi Hotel