



Ingegneria del Software e Progettazione Web
Progetto A.A. 2023/2024

MY ITINERARY

0306782

Dennis Mariani

Indice

1. Specifica dei requisiti software	3
1.1 – Introduzione	3
1.1.1 – Obbiettivi del documento.....	3
1.1.2 – Panoramica del sistema definito	3
1.1.3 – Requisiti HW e SW	4
1.1.4 – Sistemi correlati, pro e contro	4
1.2 – User Stories	5
1.2.1 - US1	5
1.2.2 – US2	5
1.2.3 – US3	5
1.3 – Functional Requirements	5
1.3.1 – FR1.....	5
1.3.2 – FR2.....	5
1.3.3 – FR3.....	5
1.4 – Use Cases	6
1.4.1 – Overview Diagram.....	6
1.4.2 – Internal Steps	6
2. Storyboards	8
3. Design.....	11
3.1 – Class Diagram.....	11
3.1.1 – VOPC (View Of Participating Classes).....	11
3.1.2 – Design Level Diagram	12
3.2 – Design Patterns	13
3.3 – Activity Diagram	14
3.4 – Sequence Diagram	15
3.5 – State Diagram.....	16
4. Testing.....	16
5. Exceptions.....	17
6. Persistenza	17
6.1 – Database MySQL.....	18
6.2 – File System.....	18
7. SonarCloud.....	19
8. Discrepanze.....	19

1. Specifica dei requisiti software

1.1 – Introduzione

1.1.1 – Obbiettivi del documento

L'obiettivo del documento presente è quello di descrivere in maniera dettagliata il processo di sviluppo del progetto software "My Itinerary", a partire dai requisiti fino alla fase di testing.

Verrà fornita una panoramica delle funzionalità e dell'architettura del sistema sviluppato, servendosi di strumenti professionali come il linguaggio di modellazione UML e specificando i requisiti hardware e software necessari per eseguire l'applicazione.

Inoltre, il documento confronta l'applicazione con altri sistemi simili, evidenziando vantaggi e svantaggi rispetto ad essi. In tal modo, il lettore è in grado di sviluppare una visione chiara delle aspettative e dei vincoli del progetto.

1.1.2 – Panoramica del sistema definito

"My Itinerary" è un sistema software progettato per organizzare al meglio i propri viaggi grazie alla realizzazione di itinerari personalizzati. L'applicazione aiuta i viaggiatori a scoprire attività e attrazioni nelle varie città del mondo, permettendo di conoscere al meglio l'arte, la storia e la cultura dei luoghi visitati. Inoltre, il sistema si rivolge ai proprietari di ristoranti e alloggi permettendo di pubblicizzare le proprie attività e facilitare il contatto con i viaggiatori.

Le principali funzionalità del sistema includono:

- **Creazione di itinerari personalizzati:** i viaggiatori, in base alla città scelta e ai giorni stabiliti per visitarla, possono pianificare un itinerario, aggiungendo attività e attrazioni da esplorare, come musei, parchi, biblioteche e altro. In particolare, l'applicazione si integra con API di terze parti, come Google Maps, per fornire recensioni e dettagli sui luoghi di interesse turistico.
- **Condivisione degli itinerari:** il sistema permette ad un viaggiatore di condividere il proprio itinerario e di trovare quelli creati da altri viaggiatori, al fine di creare una rete nella quale fornire spunti ed idee sulle varie città.
- **Creazione annunci:** i proprietari di attività di ristorazione o di alloggi possono condividere annunci che verranno suggeriti ai viaggiatori.

"My Itinerary" offre un'interfaccia user-friendly, adatta ad utenti di tutti i tipi e che permette di accedere in maniera semplice a tutte le funzionalità offerte.

1.1.3 – Requisiti HW e SW

Per garantire il corretto funzionamento di "My Itinerary", è necessario rispettare i seguenti requisiti hardware e software.

Requisiti Hardware Minimi:

- Processore: Intel Core i3 o equivalente AMD
- Memoria (RAM): 4 GB
- Schermo: Risoluzione minima 1280 x 800 pixel
- Connessione Internet: Necessaria per l'accesso ai servizi di terze parti.

Requisiti Software:

- Sistema Operativo: Windows 10 o successivo, macOS 10.15 o successivo, distribuzioni Linux con kernel 5.0 o superiore
- Java Runtime Environment (JRE): Versione 17 o successiva

1.1.4 – Sistemi correlati, pro e contro

Nel mercato delle applicazioni di viaggio ci sono sistemi correlati a "My Itinerary", ad esempio Google Travel o Trip Advisor. Vi sono, però, alcune differenze:

Google Travel: offre una piattaforma per la pianificazione di viaggi, basata su mappe e prenotazione di voli e hotel. Sebbene Google Travel sia un ottimo strumento, non è pensato per la personalizzazione dettagliata degli itinerari e sul suggerimento di attività di interesse nelle città desiderate. La nostra applicazione permette agli utenti di pianificare viaggi su misura, anche in base alle tempistiche di viaggio. Nonostante Google Travel presenti importanti differenze con "My Itinerary", non viene esclusa la possibilità di utilizzare entrambi gli strumenti, in quanto riescono a completarsi a vicenda.

TripAdvisor: è uno dei più importanti siti di recensioni e informazioni turistiche, su hotel, ristoranti e attrazioni. I due sistemi sebbene siano correlati, sono profondamente diversi, infatti TripAdvisor è orientato solamente alla consultazione di recensioni e informazioni inerenti alle attività piuttosto che alla pianificazione completa di viaggi.

1.2 – User Stories

1.2.1 - US1

Come viaggiatore, voglio vedere una lista delle attività disponibili nella città che ho selezionato suddivise in categorie*, così da poter scegliere le attività che più si adattano ai miei interessi e preferenze.

*categorie: musei, monumenti, parchi, teatri, vita notturna, arte e biblioteche.

1.2.2 – US2

Come viaggiatore, voglio cercare itinerari in base alla città e alla durata del mio viaggio, così da poter trovare itinerari creati da altri viaggiatori che posso utilizzare come ispirazione.

1.2.3 – US3

Come proprietario di attività, voglio creare annunci pubblicitari per promuovere il mio ristorante o alloggio, così da poter aumentare la mia visibilità tra i viaggiatori che pianificano i loro itinerari.

1.3 – Functional Requirements

1.3.1 – FR1

Il sistema deve interagire con l'API di Google Maps per ottenere e restituire ai viaggiatori una lista delle attività nella città selezionata divise in categorie*.

*categorie: musei, monumenti, parchi, teatri, vita notturna, arte e biblioteche.

1.3.2 – FR2

Il sistema deve fornire al viaggiatore dettagli* completi per ciascuna attività.

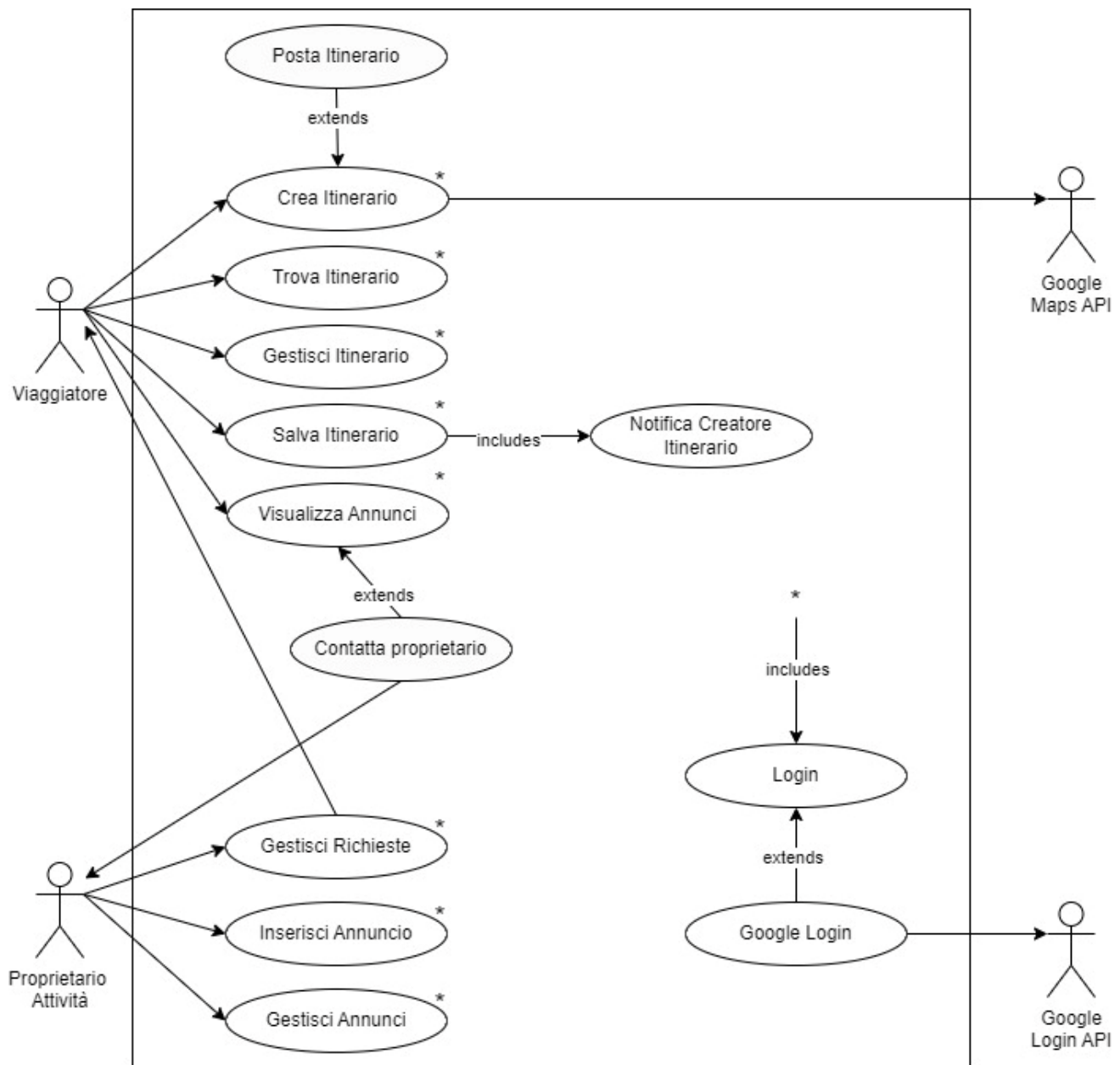
*dettagli: nome, categoria, indirizzo, valutazione e recensioni.

1.3.3 – FR3

Il sistema deve suggerire al viaggiatore una lista di alloggi e ristoranti disponibili nella città scelta una volta completata la creazione dell'itinerario.

1.4 – Use Cases

1.4.1 – Overview Diagram



1.4.2 – Internal Steps

Use Case: Crea Itinerario

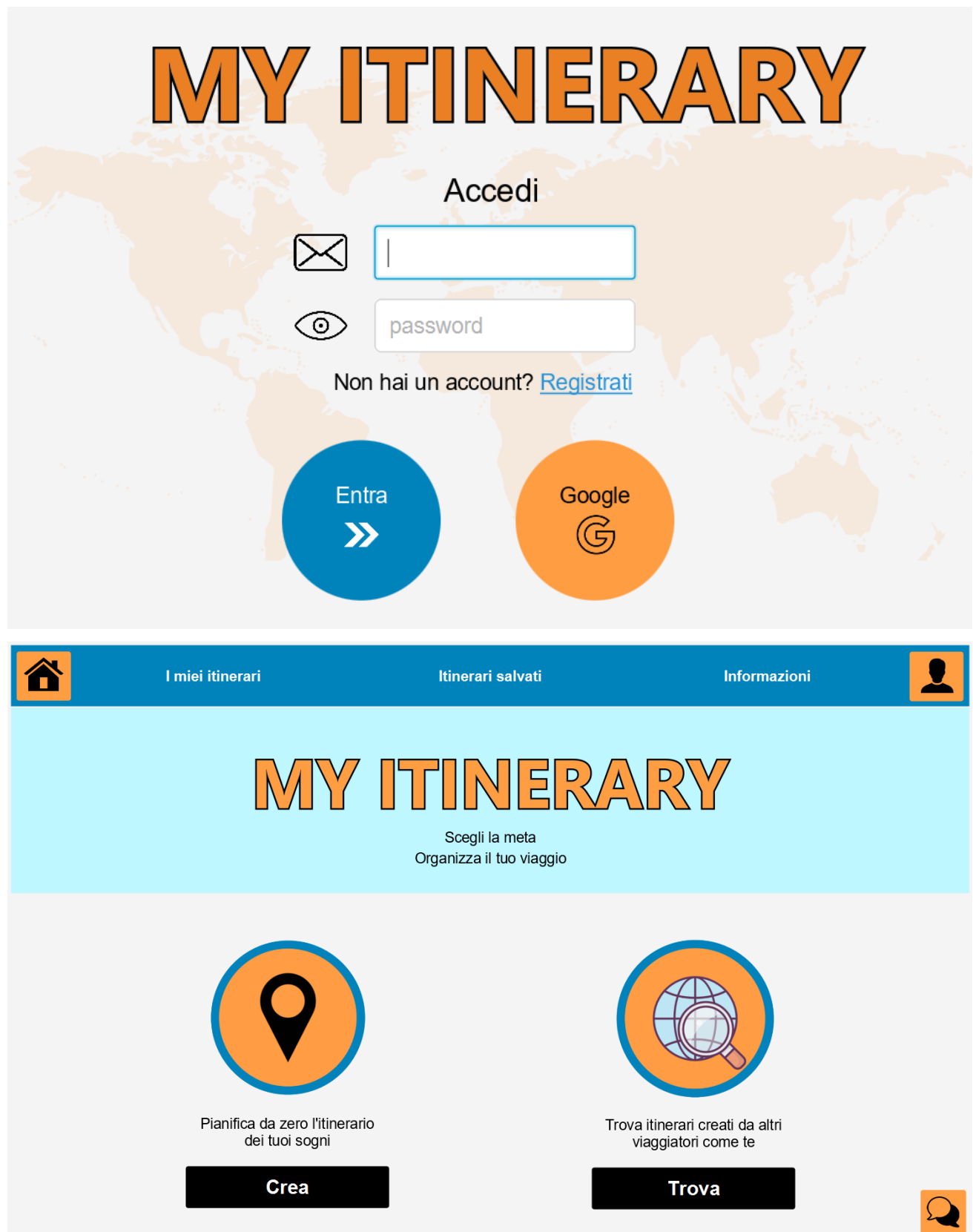
1. L'utente viaggiatore inserisce il nome di una località di suo interesse.
2. Il sistema interagisce con l'API di Google Maps per recuperare una lista di attività e attrazioni disponibili nella località selezionata.
3. L'utente inserisce il numero di giorni di viaggio che desidera pianificare.



4. Il sistema verifica che il numero di giorni inserito sia maggiore di zero.
5. L'utente seleziona le categorie* di attività di suo interesse.
6. Il sistema filtra e mostra le attività corrispondenti alle categorie selezionate dall'utente, restituendo una lista di opzioni pertinenti.
7. L'utente visualizza la lista delle attività e seleziona quelle di maggiore interesse, assegnandole a specifici giorni all'interno dell'itinerario.
8. L'utente termina e conferma la creazione dell'itinerario.
9. Il sistema memorizza l'itinerario.

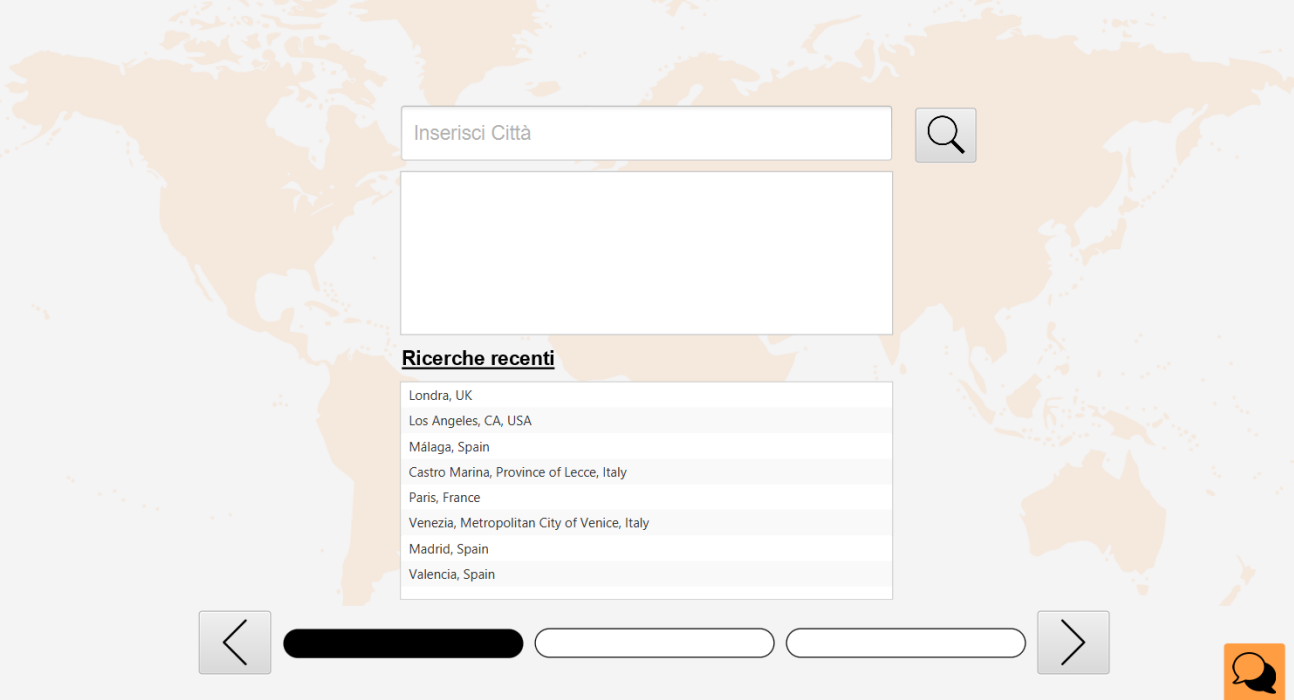
Estensioni:


- 1a. Località inesistente: il sistema segnala l'errore all'utente e richiede l'inserimento di un nome valido.
- 2a. Fallisce l'interazione con Google Maps: il sistema segnala l'errore e chiede di riprovare.
- 2b. Nessuna attività nella località selezionata: il sistema informa l'utente.
- 4a. Numero di giorni non valido: il sistema segnala l'errore e chiede di inserire un numero valido.
- 9a. Il salvataggio fallisce: il sistema segnala l'errore e chiede di riprovare.

2. Storyboards






I miei itinerariItinerari salvatiInformazioni





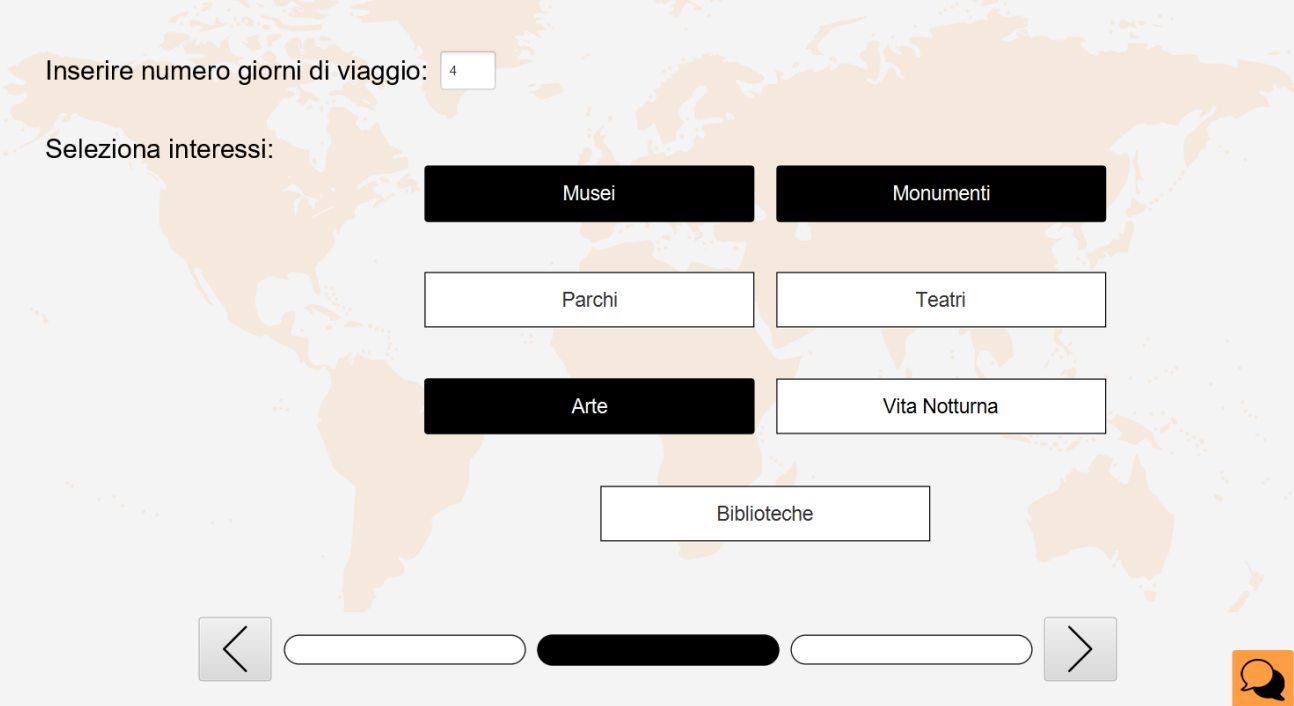


Ricerche recenti

Londra, UK
Los Angeles, CA, USA
Málaga, Spain
Castro Marina, Province of Lecce, Italy
Paris, France
Venezia, Metropolitan City of Venice, Italy
Madrid, Spain
Valencia, Spain



I miei itinerariItinerari salvatiInformazioni



Inserire numero giorni di viaggio:

Seleziona interessi:

Musei

Monumenti




Parchi



Teatri

Arte

Vita Notturna

Biblioteche



[I miei itinerari](#)[Itinerari salvati](#)[Informazioni](#)

Roma, Metropolitan City of Rome Capital, Italy - 1 giorno/i - Creato il: 14/08/2024

Torino - 1 giorno/i - Creato il: 14/08/2024

Londra, UK - 1 giorno/i - Creato il: 14/08/2024

Roma - 1 giorno/i - Creato il: 14/08/2024

Los Angeles, CA, USA - 3 giorno/i - Creato il: 18/08/2024

Los Angeles, CA, USA - 1 giorno/i - Creato il: 18/08/2024


Venezia - 5 giorno/i - Creato il: 18/08/2024

Paris, France - 2 giorno/i - Creato il: 18/08/2024

Venezia, Metropolitan City of Venice, Italy - 1 giorno/i - Creato il: 18/08/2024

Roma - 1 giorno/i - Creato il: 19/08/2024

[Indietro](#)[Vedi Dettagli](#)[Cancella Itinerario](#)[Posta Itinerario](#)



[I miei annunci](#)[Attività](#)[Informazioni](#)

MY ITINERARY

Sei il proprietario di un ristorante, di un pub, di un bar o di qualsiasi altro servizio di ristorazione?
Vuoi affittare il tuo appartamento o le tue camere ai turisti?
Questa applicazione è il posto giusto per te!

Inserisci un annuncio e la tua attività verrà mostrata a tutti i viaggiatori che creano o cercano itinerari nella tua città.
Fai click su uno dei pulsanti sottostanti per procedere con l'inserimento... Buona fortuna!



Inserisci



Inserisci

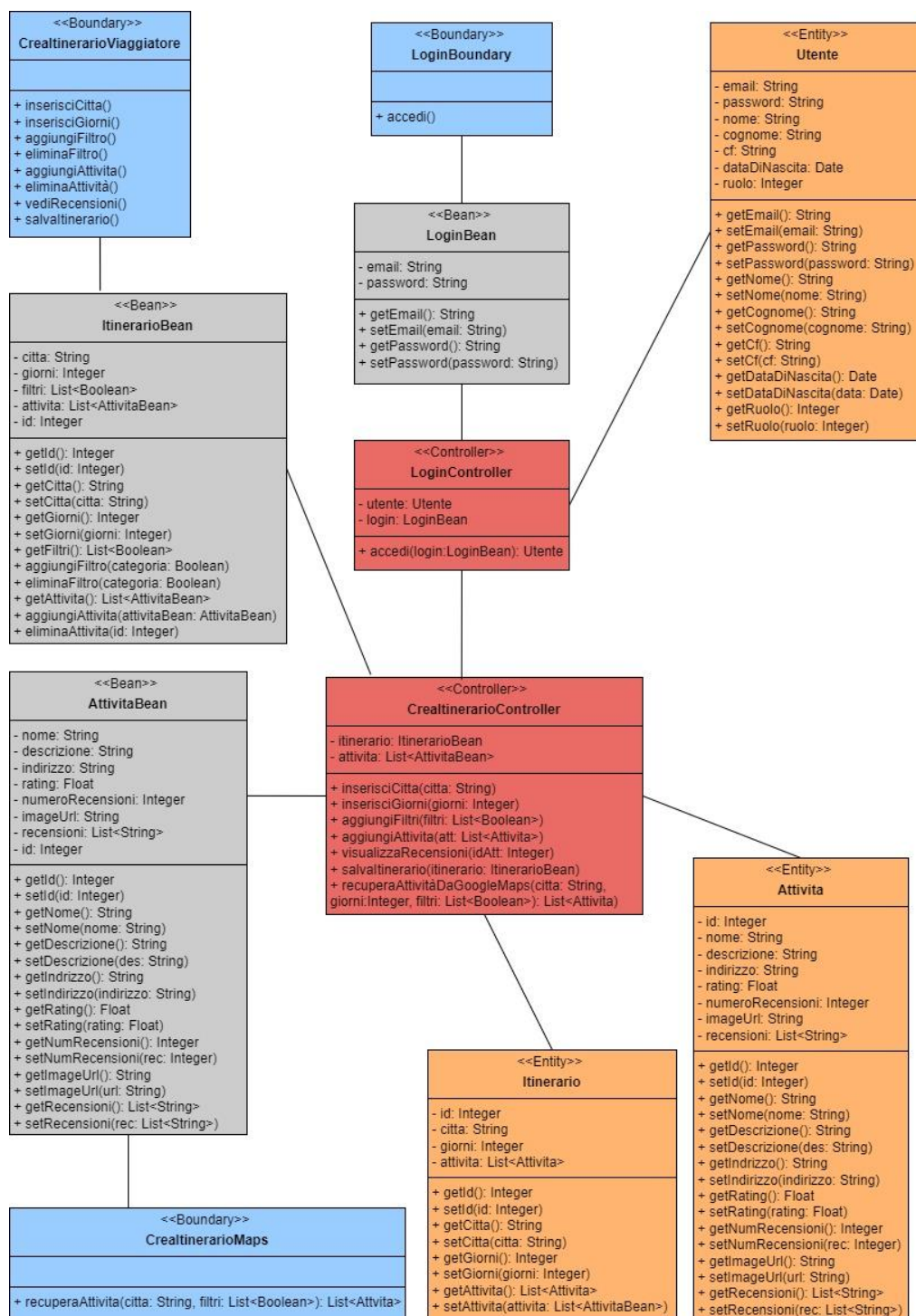


3. Design

3.1 – Class Diagram

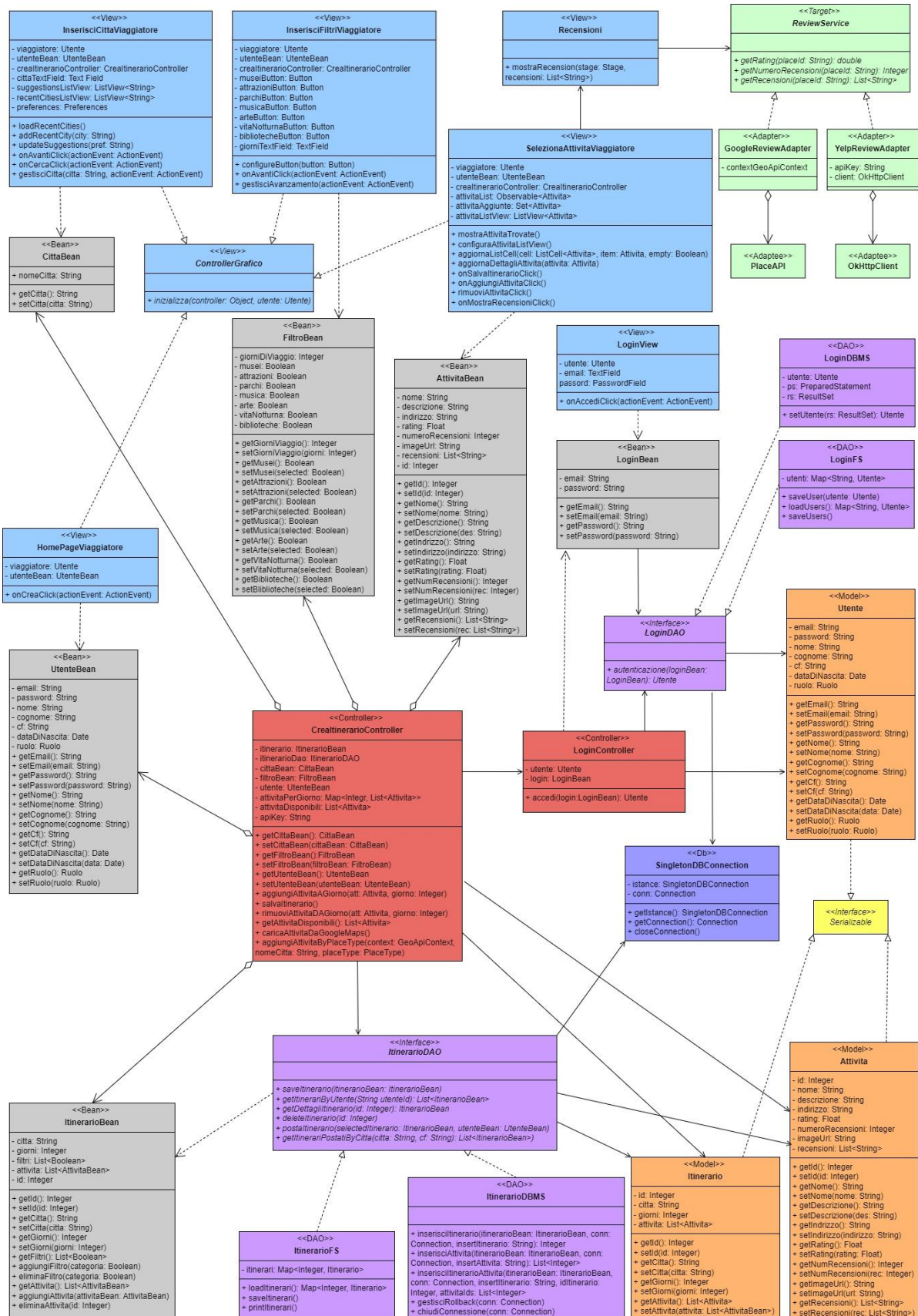
3.1.1 – VOPC (View Of Participating Classes)

Use Case: Crea Itinerario



3.1.2 – Design Level Diagram

Use Case: Crea Itinerario



3.2 – Design Patterns

I pattern GoF utilizzati nel progetto sono: Singleton e Adapter.

Pattern Singleton per Connessione Database:

Questo pattern è stato utilizzato per gestire la connessione al database mediante la classe “SingletonDBConnection”. Questa classe garantisce che ci sia solo una singola connessione al database attiva, migliorando l'efficienza nell'uso delle risorse. L'implementazione del metodo include la sincronizzazione per evitare problemi di concorrenza in ambiente multithread, garantendo che solo un thread possa creare l'istanza iniziale.

Pattern Adapter per Recensioni:

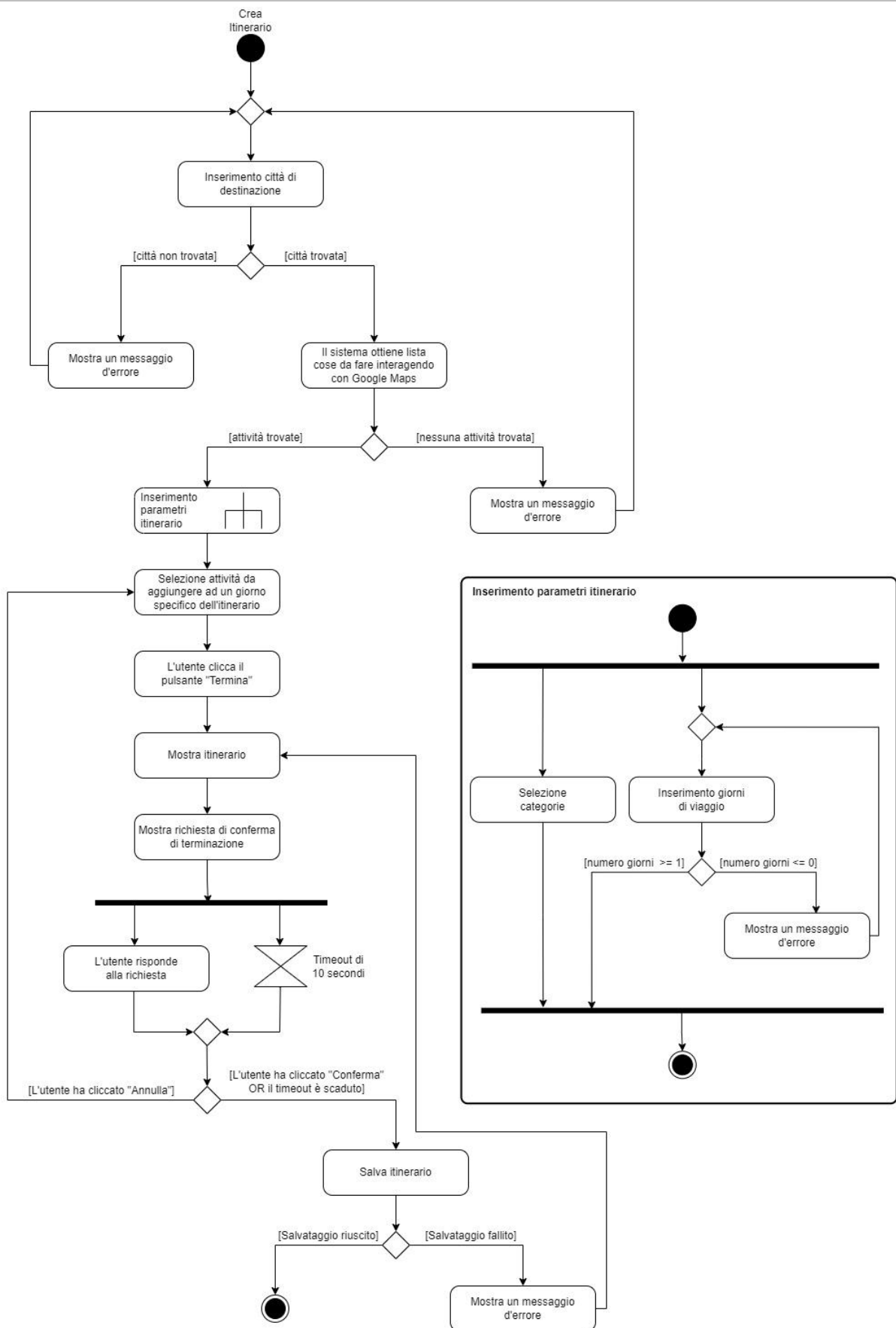
Pattern strutturale che consente di far collaborare tra loro classi con interfacce incompatibili o diverse. Nel progetto, il pattern Adapter è stato implementato per permettere l'interazione tra il sistema e diversi servizi di recensioni esterni, come Google Places e Yelp.

Infatti, questo pattern permette di astrarre l'implementazione del servizio di recensioni, fornendo un'interfaccia comune (ReviewService) che può essere implementata per diversi fornitori di recensioni. Dunque, si può aggiungere o cambiare il fornitore senza dover modificare il codice esistente che utilizza l'interfaccia. Questo rende il sistema più flessibile e aperto all'estensione.

Nel pattern si possono individuare diversi ruoli:

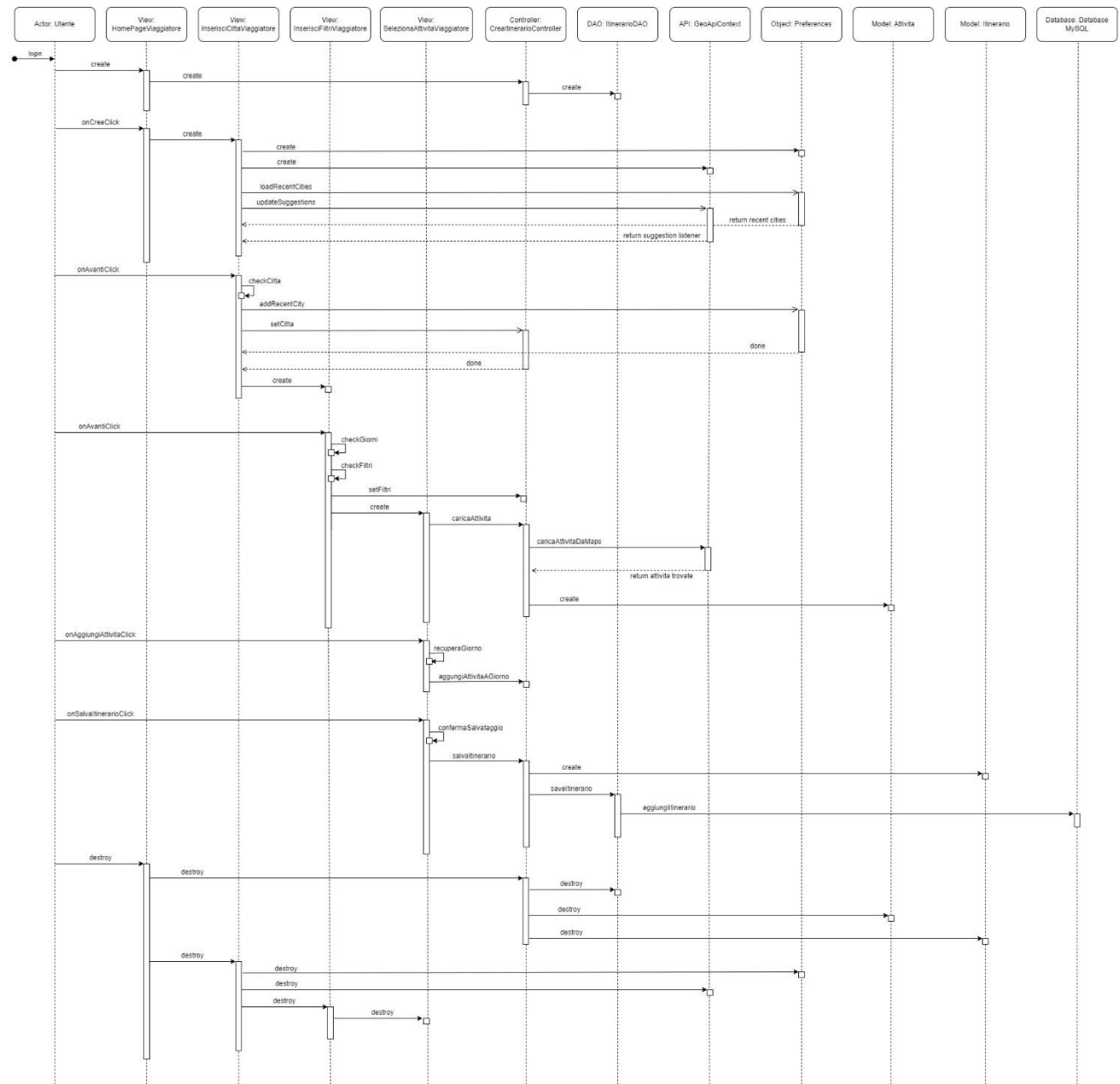
- **Target:** l'interfaccia che definisce i metodi per ottenere le recensioni, nel nostro caso la classe “ReviewService”.
- **Client:** la classe che interagisce con l'interfaccia ReviewService, nel nostro caso la classe “SelezionaAttivitaViaggiatore”.
- **Adapter:** implementa l'interfaccia ReviewService e adatta le chiamate del client ai metodi specifici delle API di Google Places o di Yelp, nel progetto le classi Adapter sono “GoogleReviewAdapter” e “YelpReviewAdapter”.
- **Adaptee:** le classi che forniscono effettivamente le funzionalità richieste ma che hanno un'interfaccia incompatibile con il sistema principale. Nel mio progetto sono le classi PlacesApi di Google e API di Yelp

3.3 – Activity Diagram



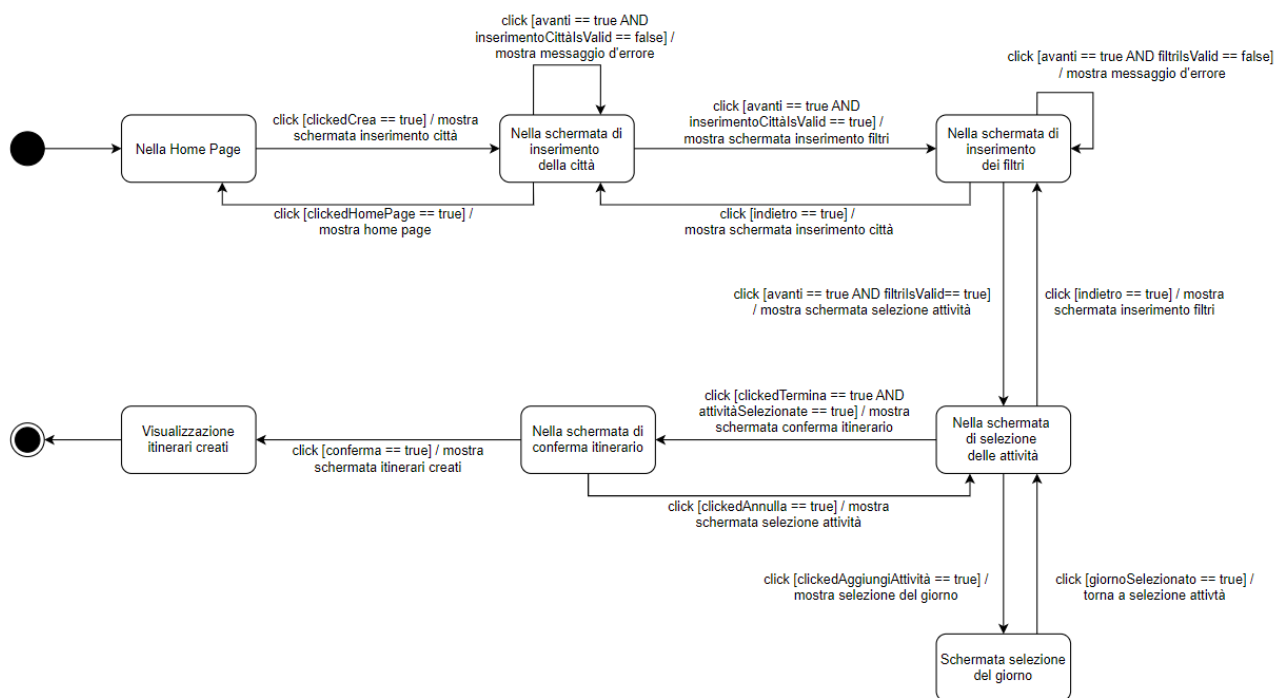
3.4 – Sequence Diagram

Use Case: Crea Itinerario



3.5 – State Diagram

Use Case: Crea Itinerario



4. Testing

Il testing è un componente fondamentale per lo sviluppo del software in quanto garantisce la correttezza del codice, la robustezza del sistema e la sua affidabilità.

Ho sviluppato tre classi di test utilizzando il framework JUnit per coprire le funzionalità principali dell'applicazione:

- **TestItinerario:** contiene il test “testSaveItinerario” che verifica il corretto funzionamento del salvataggio di un itinerario nel database.
- **TestAnnunci:** contiene il test “testInserisciAnnuncio” che verifica l’inserimento di un annuncio di attività (ad esempio, un ristorante) nel database.
- **TestLogin:** contiene il test “testAutenticazione” che verifica il processo di autenticazione degli utenti.

5. Exceptions

Ho implementato diverse classi di eccezioni personalizzate per gestire in modo appropriato gli errori che possono verificarsi durante l'esecuzione e per fornire dei messaggi d'errore specifici. Così facendo aumenta la leggibilità del codice ed è più facile identificare i problemi.

Di seguito una panoramica delle eccezioni implementate:

- **AnnuncioNonTrovatoException:** questa eccezione viene lanciata quando non vengono trovati nel sistema degli annunci che rispettano i requisiti di ricerca richiesti.
- **AutenticazioneFallitaException;** questa eccezione viene lanciata quando fallisce il processo di autenticazione dell'utente, per esempio, se le credenziali inserite sono errate o se c'è un problema di connessione con il sistema di autenticazione.
- **CambioPaginaException:** questa eccezione viene lanciata durante la fase di navigazione tra le diverse schermate dell'applicazione, se si verifica un errore nel cambio di pagina, ad esempio, se il file FXML della pagina non viene trovato.
- **ItemNonTrovatoException:** questa eccezione viene lanciata quando un oggetto generico, non viene trovato nel database o nel file system.
- **PostaItinerarioException:** questa eccezione viene lanciata quando si verifica un errore durante la pubblicazione di un itinerario. Ciò può includere problemi di connessione al database o errori nel salvataggio delle informazioni dell'itinerario.
- **RecensioniNonDisponibiliException:** questa eccezione viene lanciata quando il sistema non è in grado di recuperare le recensioni per una determinata attività o annuncio.
- **SalvataggioItinerarioException:** questa eccezione viene lanciata quando si verifica un errore durante il salvataggio di un itinerario nel sistema. Ciò può essere dovuto a problemi di connessione al database o errori nei dati forniti dall'utente.

6. Persistenza

Nell'applicazione “My Itinerary” ho introdotto due livelli di persistenza: un livello basato su un database relazionale MySQL e l'altro basato su File System.

Sono state adottate delle interfacce DAO (Data Access Object) che definiscono metodi per le operazioni di persistenza, in modo da separare la logica di business dalla logica di accesso ai dati. Ogni DAO gestisce un'entità specifica e presenta due implementazioni concrete:

- **Implementazione DBMS:** utilizza JDBC per comunicare con il database relazionale MySQL, garantendo performance e affidabilità.
- **Implementazione File System:** utilizza la serializzazione Java per leggere e scrivere dati su file, offrendo una soluzione più leggera e semplice.

6.1 – Database MySQL

MySQL è in grado di garantire:

- **Affidabilità:** offre transazioni affidabili e la possibilità di gestire rollback in caso d'errore.
- **Prestazioni:** gestisce efficacemente operazioni di lettura e scrittura su grandi volumi di dati.
- **Scalabilità:** ideale per applicazioni in espansione.

I DAO che implementano la persistenza su MySQL sono:

- **AnnuncioDBMS:** gestisce l'inserimento, l'eliminazione e la ricerca degli annunci creati dai proprietari di attività.
- **ItinerarioDBMS:** responsabile della gestione degli itinerari creati dagli utenti, inclusi l'inserimento, la modifica e l'eliminazione degli itinerari, nonché la pubblicazione degli stessi.
- **LoginDBMS:** gestisce l'autenticazione degli utenti utilizzando query SQL per verificare le credenziali di accesso nel database

6.2 – File System

I vantaggi dell'implementazione con file system sono:

- **Leggerezza:** offre una soluzione semplice e immediata senza configurazioni complesse.
- **Portabilità:** i dati sono facilmente trasferibili tra diverse installazioni dell'applicazione.
- **Facilità di debug:** i file possono essere facilmente ispezionati e modificati manualmente.

I DAO che utilizzano il file system sono:

- **Annuncio FS:** gestisce gli annunci salvandoli in un file annunci.dat. I dati vengono serializzati e memorizzati in formato binario, con un sistema di ID unici per identificare gli annunci.
- **ItinerarioFS:** gestisce la persistenza degli itinerari in un file denominato itinerari.dat. Ogni itinerario è serializzato e include le attività selezionate dall'utente.
- **LoginFS:** gestisce l'autenticazione degli utenti tramite un file chiamato users.dat, che memorizza le informazioni degli utenti registrati.

7. SonarCloud

È stato utilizzato SonarCloud per monitorare e migliorare il codice durante lo sviluppo, grazie all'identificazione di potenziali bug, vulnerabilità di sicurezza e difetti di manutenzione.

Tutti i risultati dell'analisi sono disponibili al seguente link:

https://sonarcloud.io/project/overview?id=Denni-02_MyItinerary2

8. Discrepanze

Durante lo sviluppo dell'applicazione, alcune funzionalità previste nella fase di progettazione non sono state completamente implementate nel codice finale a causa di vincoli di tempo:

- **Recensioni Yelp non implementate:** sebbene fosse prevista l'integrazione con il servizio di recensioni Yelp, questa funzionalità non è stata completata. Tuttavia, la parte relativa alle recensioni Google è stata implementata con successo, utilizzando il pattern Adapter. Questo garantisce che il codice possa essere facilmente esteso in futuro per supportare altri servizi di recensioni, come appunto Yelp.
- **Procedura di registrazione e Google Login:** la procedura di registrazione degli utenti e il login tramite Google non sono stati implementati nel codice finale. È ovviamente disponibile il login normale tramite credenziali.
- **Incompletezza nei gestori di eventi UI:** Alcune interazioni previste nell'interfaccia utente non sono state implementate. In particolare, non è stata completata la gestione dei click relativi alla gestione dell'account e alle informazioni dell'utente.
- **Mancanza di salvataggio degli itinerari creati da altri utenti:** Il sistema consente agli utenti di cercare itinerari creati da altri viaggiatori, ma non supporta ancora il salvataggio di questi itinerari. Questa funzionalità era prevista per permettere agli utenti di memorizzare e riutilizzare gli itinerari pubblicati da altri, ma non è stata completata entro i tempi previsti.

Nonostante queste discrepanze, il codice sviluppato rimane solido e conforme alla progettazione iniziale in termini di funzionamento generale e utilizzo dei pattern. Le funzionalità non implementate rappresentano aree di miglioramento e possono essere integrate in future versioni del progetto.