Università degli Studi di Napoli Federico II Scuola Politecnica e delle Scienze di Base

Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione



Corso di Laurea in Informatica Insegnamento di Ingegneria del Software Anno accademico 2019/2020

Specifica, progettazione, implementazione e validazione del Sistema Informativo "Consiglia Viaggi2019"

Autore:
Ivan Capasso N86002587

Docente: Sergio di Martino

Sommario

| Descrizione del progetto | 3 |
|--|----|
| Introduzione | 3 |
| Documento dei Requisiti Software | 4 |
| Modello Funzionale | 4 |
| Diagramma dei casi d'uso | 8 |
| Mockup | 9 |
| Tabelle di Cockburn | 16 |
| Modello di dominio: | 28 |
| Class diagram | 28 |
| Sequence Diagram | 35 |
| Diagramma di stato di analisi: Ricerca Strutture | 40 |
| Diagramma di attività: Registrazione utente | 41 |
| Pianificazione dettagliata dell'attività | 42 |
| Documento di Design del sistema | 43 |
| Analisi dell'architettura | 43 |
| Diagramma delle classi di design | 47 |
| CRC Cards | 49 |
| Diagrammi di sequenza di design | 58 |
| Documento di Testing del sistema | 60 |
| Test Plan per System Testing | 60 |
| Codice xUnit per unit testing di 2 metodi | 69 |

Capitolo 1

Descrizione del progetto

Introduzione

Il progetto "Consiglia Viaggi 2019" consiste nella realizzazione di un software il cui scopo e quello di permettere agli utenti che utilizzano l'applicativo di visionare e recensire delle strutture.

La ricerca o visualizzazione delle strutture avviene tramite l'utilizzo di una mappa, oppure può avvenire per nome mediante una apposita barra di ricerca oppure ancora selezionando una categoria di strutture dalla home.

Per rendere più veloce la ricerca, c'è la possibilità di aggiungere dei filtri in base alle stelle o alla distanza della struttura e selezionare il tipo di ordinamento dei risultati. La mappa, invece, implementa dei filtri che permettono di decidere la distanza massima rispetto alla posizione attuale dell'utente.

Una volta selezionata una struttura oltre a visualizzarne le informazioni si potranno visualizzare anche le recensioni inserite dagli utenti, anch'esse filtrabili o ordinabili. Si possono scrivere nuove recensioni indicando un numero di stelle comprese tra 1 e 5 e una descrizione di almeno 100 caratteri ma per farlo e necessario essere autenticati. Invece, per tutte le altre funzioni precedentemente descritte, non è richiesta la creazione di un account ma si può accedere all'applicazione come "visitatore".

L'utente per registrarsi dovrà inserire: un nome utente ed una password una mail e il nome reale dell'utente.

Capitolo 2

Documento dei Requisiti Software

Modello Funzionale

Iniziamo identificando i requisiti: essi sono composti da un identificativo, un nome ed una descrizione, ritorneranno particolarmente utili nella fase di testing e nelle tabelle di Cockburn.

Requisiti funzionali

Requisiti utente: ottenuti dopo un brainstorming con gli stakeholders

| ID | RE01 |
|-------------|---|
| Nome | Ricerca e visualizzazione di una struttura |
| Descrizione | L'utente deve avere la possibilità di cercare e visualizzare qualsiasi struttura presente sul software, a prescindere se esso sia autenticato o meno. |
| | Per rendere la ricerca più fluida e veloce, il sistema deve permettere di applicare filtri e ordinamenti sulle strutture. |

| ID | RE02 |
|-------------|--|
| Nome | Visualizzazione recensioni per una struttura |
| Descrizione | Una volta aperta una struttura l'utente deve poter visualizzare |
| | l'elenco delle recensioni scritte da altri utenti riguardo quella struttura. |
| | L'utente può applicare dei filtri e ordinamenti, per esempio in base al numero di stelle, per visionare al meglio le recensioni. |

| ID | RE03 | | |
|-------------|--|--|--|
| Nome | Visualizzazione delle strutture su mappa | | |
| Descrizione | L'utente deve avere la possibilità di visualizzare qualsiasi struttura | | |
| | presente sull'applicativo tramite una mappa; A prescindere | | |
| | se l'utente sia autenticato. | | |
| | | | |
| | L'utente può effettuare dei filtri sulla mappa in base alla | | |
| | distanza di una località in base alla sua posizione attuale. | | |

| ID | RE04 |
|-------------|--|
| Nome | Pubblicazione di recensioni per una struttura |
| Descrizione | Una volta aperta una struttura l'utente deve poter pubblicare una |
| | recensione attraverso un apposito tasto. Per la scrittura della |
| | recensione l'utente deve scegliere una valutazione che va da 1 a 5 |
| | stelle, una descrizione di almeno 100 caratteri. |

| ID | RE05 | | | | |
|-------------|---|--|--|--|--|
| Nome | Accesso autenticato | | | | |
| Descrizione | L'utente deve avere la possibilità di autenticarsi per la pubblicazione di una recensione. | | | | |
| | Una volta autenticato sarà autorizzato a recensire qualsiasi struttura presente nell'applicativo. | | | | |
| | Il sistema dovrà offrire la possibilità di registrarsi o di effettuare un login tramite l'inserimento di credenziali scelte precedentemente. | | | | |
| | Nel caso in cui l'utente voglia registrarsi perché non in possesso di un account, il sistema richiederà all'utente di inserire i seguenti dati: | | | | |
| | Nome | | | | |
| | Cognome | | | | |
| | NicknamePassword | | | | |
| | • E-mail | | | | |
| | Nel caso in cui l'utente sia già registrato può autenticarsi fornendo al sistema i propri dati di accesso. | | | | |

Requisiti non funzionali

Requisiti di sistema: impliciti dai requisiti funzionali.

| ID | RENF01 |
|-------------|---|
| Nome | Visualizzazione della mappa tramite Google Maps |
| Descrizione | La visualizzazione della mappa deve avvenire tramite le API offerte da Google |

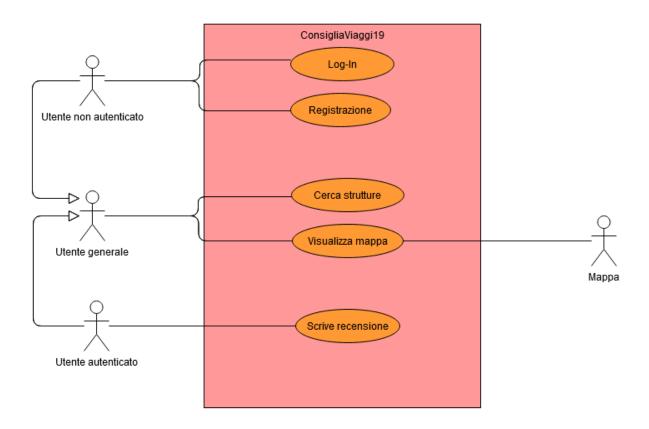
| ID | RENF02 | |
|-------------|--|--|
| Nome | Usabilità mobile | |
| Descrizione | L'applicazione deve funzionare a prescindere dalla grandezza del | |
| | dispositivo. | |

| ID | RENF03 |
|-------------|--|
| Nome | Performance ricerca |
| Descrizione | Una ricerca deve avvenire in meno di 2.5 secondi nel 90% dei casi. |

Requisiti di dominio

| ID | DOM01 |
|-------------|---|
| Nome | Rispetto del GDPR |
| Descrizione | Nella form di registrazione bisogna rispettare il regolamento Generale sulla protezione dei dati GDPR (General Data Protection Regulation). |

Diagramma dei casi d'uso



Mockup

Di seguito sono riportati i mockup delle varie schermate dell'applicativo.

Gli errori e le note riguardanti una schermata sono riportati sulla destra della stessa. Per chiarezza, in alcune schermate **NON** sono riportati tutti gli errori possibili ma solo quelli più significativi.

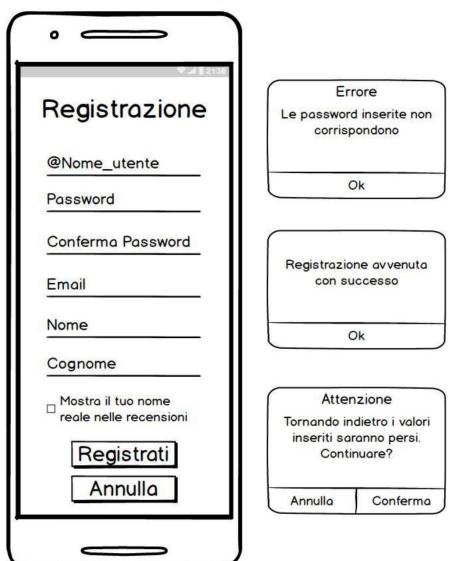
Login utente



Attenzione

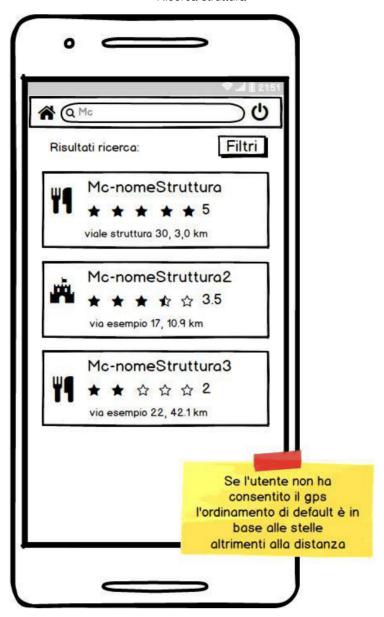
La password o l'username inseriti non sono validi, riprovare

Ok



Марра AQ U CV19 Distanza 5 km 10 km 3 km 1 km 41

Ricerca struttura



Filtri ricerca struttura



Struttura



Se l'utente non ha effettuato l'accesso o ha già scritto una recensione il tasto è disabilitato

Scrittura recensione

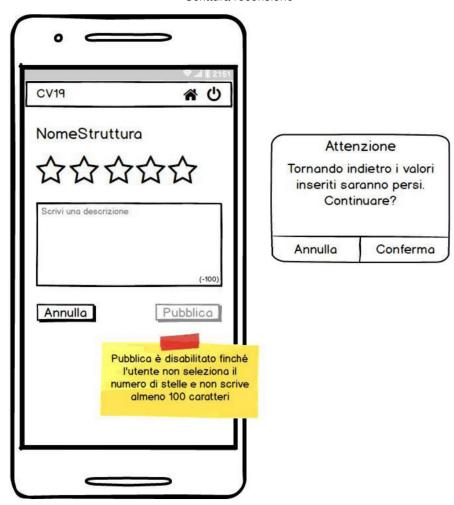


Tabelle di Cockburn

Le seguenti tabelle di Cockburn sono relative ai casi d'uso definiti nello UCD.

Registrazione utente

| Use Case #1 | Si registra | | | |
|--|--|--|---|--|
| Goal in context | L'uten | L'utente vuole creare un account | | |
| Precondition | None | | | |
| Success end condition | Creaz | ione account | | |
| Failed end condition | Accour | nt non creato, chiude la so | chermata di registrazione | |
| Primary actor | Utente | e non autenticato | | |
| Trigger | L'utente preme il tasto "Crea un account" nella schermata "Schermata Iniziale" | | | |
| DESCRIPTION | Step | Utente non autenticato | Sistema | |
| | 1 | Preme il pulsante "Crea un account" nella "Schermata iniziale" | | |
| | 2 | | Crea il nuovo account e | |
| | | | mostra pop-up di suc- | |
| | | | cesso | |
| | 3 | Preme il tasto "Ok" | | |
| | 4 | | Mostra la schermata | |
| | | | "Schermata principale" | |
| EXTENSION #1 | Step | Utente non autenticato | Sistema | |
| L'utente lascia vuoto uno o più campi | 2.1 | | Evidenzia tutti i campi e mostra il messaggio "Questo campo non può essere lasciato vuoto" e infine torna allo step 1 dello scenario principale | |

| EXTENSION #2 | Step | Utente non autenticato | Sistema |
|--|------|------------------------|--------------------------|
| Il nickname scelto | 2.2 | | Evidenzia il campo |
| dall'utente è inferiore | | | nome utente con il |
| i 4 caratteri | | | messaggio "Il nome |
| | | | utente deve essere lungo |
| | | | almeno 4 caratteri" e |
| | | | torna allo step 1 dello |
| | | | scenario principale |
| EXTENSION #3 | Step | Utente non autenticato | Sistema |
| Le due password | 2.3 | | Mostra pop-up di |
| inserite dall'utente non coincidono | | | avvertenza |
| | 3.3 | Preme il tasto "Ok" | |
| | 4.3 | | Torna allo step 1 dello |
| | | | scenario principale |
| EXTENSION #4 | Step | Utente non autenticato | Sistema |
| L'utente inserisce una | 2.4 | | Evidenzia il campo |
| mail non valida | | | email con messaggio" Il |
| | | | valore inserito non |
| | | | corrisponde ad una |
| | | | Email" e torna allo step |
| | | | 1 dello scenario |
| | | | principale |
| EXTENSION #5 | Step | Utente non autenticato | Sistema |
| L'utente inserisce un | 2.5 | | Evidenzia il campo |
| nickname già in | | | nome utente con |
| uso | | | messaggio "Nome |
| | | | utente già in uso" e |
| | | | torna allo step 1 dello |
| | | | scenario principale |

| EXTENSION #6 | Step | Utente non autenticato | Sistema |
|------------------------|------|------------------------|-------------------------|
| L'utente inserisce una | | | Evidenzia il campo |
| email già in uso | | | e-mail con messaggio |
| | | | "Email già in uso" e |
| | | | torna allo step 1 dello |
| | | | scenario principale |
| EXTENSION #7 | Step | Utente non autenticato | Sistema |
| L'utente annulla | 2.7 | Preme il tasto | |
| dopo aver riempito | | "Annulla" | |
| uno o più campi | 3.7 | | Mostra pop-up di |
| | | | avvertenza |
| | 4.7 | Preme il tasto | |
| | | "Conferma" | |
| | 5.7 | | Torna alla schermata |
| | | | "Schermata iniziale" e |
| | | | termina lo use case |
| EXTENSION #8 | Step | Utente non autenticato | Sistema |
| L'utente inserisce una | 2.8 | | Evidenzia il campo |
| password minore di 8 | | | password con messaggio |
| caratteri | | | "La password deve |
| | | | essere lunga almeno 8 |
| | | | Caratteri" e torna allo |
| | | | step 1 dello scenario |
| | | | principale |

Login utente:

| Use Case #2 | | Effettua lo | g-in | |
|-------------------------|---------|-------------------------------------|---------------------------|--|
| Goal in context | L'uten | L'utente vuole effettuare l'accesso | | |
| Precondition | L'uten | L'utente non è autenticato | | |
| Success end condition | Riesce | e ad autenticarsi | | |
| Failed end condition | Non rie | esce ad autenticarsi e ca | mbia schermata | |
| Primary actor | Utente | e non autenticato | | |
| Trigger | Viene | aperto l'applicativo | | |
| DESCRIPTION | Step | Utente non autenticato | Sistema | |
| | 1 | Inserisce Nome Utente | | |
| | | e Password e preme il | | |
| | | tasto "Accedi" | | |
| | 2 | | Effettua il log-in e | |
| | | | visualizza la schermata | |
| | | | "Schermata principale" | |
| SUBVARIATION #1 | Step | Utente non autenticato | Sistema | |
| L'utente accede in un | 1a | Preme il menu in alto a | | |
| secondo momento | | destra ed "Accedi" | | |
| | 2a | | Torna allo step 1 dello | |
| | | | scenario principale | |
| EXTENSION #1 | Step | Utente non autenticato | Sistema | |
| L'utente lascia uno o | 2.1 | | Evidenzia i campi vuoti | |
| entrambi i campi | | | e torna allo step 1 dello | |
| vuoti | | | scenario principale | |
| EXTENSION #2 | Step | Utente non autenticato | Sistema | |
| L'utente inserisce dati | 2.2 | | Mostra pop-up di | |
| non validi per i log-in | | | avvertenza | |
| | 3.2 | Preme il tasto "Ok" | | |
| | 4.2 | | Torna allo step 1 dello | |
| | | | scenario principale | |

Visualizza struttura:

| Use Case #3 | | Visualizza st | ruttura | |
|-----------------------|--------|---|-------------------------|--|
| Goal in context | | L'utente è interessato a visualizzare le informazioni relative ad una struttura | | |
| Precondition | None | | | |
| Success end condition | Trova | la struttura cercata e la a | apre | |
| Failed end condition | Non a | pre la struttura o cambia | schermata | |
| Primary actor | Utente | 9 | | |
| Trigger | L'uten | te preme la barra di ricer | ca | |
| DESCRIPTION | Step | Utente | Sistema | |
| | 1 | Digita il nome della | | |
| | | struttura e preme invio | | |
| | 2 | | Mostra la schermata | |
| | | | "Cerca struttura" con | |
| | | | una lista di strutture | |
| | | | che contengono quel | |
| | | | nome | |
| | 3 | Preme una delle strut- | | |
| | | ture mostrate | | |
| | 4 | | Mostra la schermata | |
| | | | "Visualizza struttura" | |
| | | | relativa a quella | |
| | | | struttura | |
| SUBVARIATION #1 | Step | Utente | Sistema | |
| La ricerca della | 1a | Effettua gli step da 1 a | | |
| struttura avviene | | 4 dello use case "Visu- | | |
| tramite mappa | | alizza mappa" | | |
| | 2a | | Torna allo step 3 dello | |
| | | | scenario principale | |

| SUBVARIATION #2 | Step | Utente | Sistema |
|----------------------|------|-------------------------|---------------------------|
| La ricerca avviene | 1b | Preme una delle tre | |
| per categoria invece | | categorie nella | |
| che per nome | | schermata | |
| | | "Schermata principale" | |
| | 2b | | Filtra la ricerca in base |
| | | | al tasto premuto e |
| | | | torna allo step 3 dello |
| | | | scenario principale |
| EXTENSION #1 | Step | Utente | Sistema |
| L'utente aggiunge | 3.1 | Preme il tasto "Filtri" | |
| filtri alla ricerca | 4.1 | | Mostra la schermata |
| | | | "Filtri ricerca" |
| | 5.1 | Seleziona le sue | |
| | | preferenze e preme il | |
| | | tasto "Conferma" | |
| | 6.1 | | Filtra la ricerca in base |
| | | | ai filtri ed ordinamenti |
| | | | e torna allo step |
| | | | 3 dello scenario |
| | | | principale |
| EXTENSION #2 | Step | Utente | Sistema |
| L'utente torna alla | 3.2 | Preme il tasto "Home" | |
| alla homepage | 4.2 | | Mostra la schermata |
| | | | "Schermata principale" |
| | | | e termina lo use case |

| EXTENSION #3 | Step | Utente | Sistema |
|-------------------|------|--------------------|-------------------------|
| L'utente effettua | 3.3 | Preme la barra di | |
| una nuova ricerca | | ricerca, digita un | |
| | | nuovo nome e preme | |
| | | invio | |
| | 4.3 | | Torna allo step 2 dello |
| | | | scenario principale |
| EXTENSION #4 | Step | Utente | Sistema |
| L'utente preme il | 3.4 | Preme il tasto | |
| tasto "Accedi" | | "Accedi" | |
| | 4.4 | | Mostra la schermata |
| | | | "Schermata iniziale" e |
| | | | termina lo use case |

Visualizza mappa:

| Use Case #4 | | Visu | ıalizza mappa | |
|-----------------------|--|-----------------------|-----------------|-------------|
| Goal in context | L'utente è interessato a visualizzare le strutture nelle sue | | | |
| | vicinar | vicinanze sulla mappa | | |
| Precondition | None | | | |
| Success end condition | La ma | ppa viene visualiz | zzata | |
| Failed end condition | None | | | |
| Primary actor | Utente | 9 | | |
| Trigger | L'uten | te preme la barra | a di ricerca | |
| DESCRIPTION | Step | Utente | Sistema | Марра |
| | 1 | Preme il tasto | | |
| | | "Apri mappa" | | |
| | | nella | | |
| | | schermata | | |
| | | "Schermata | | |
| | | principale" | | |
| | 2 | | Richiede la | |
| | | | mappa al | |
| | | | provider | |
| | | | | |
| | 3 | | | Fornisce la |
| | | | | mappa al |
| | | | | sistema |
| | 4 | | Carica la | |
| | | | posizione | |
| | | | dell'utente | |
| | | | sulla mappa | |
| | 5 | | Carica tutte le | |
| | | | strutture sulla | |
| | | | тарра | |

| | 6 | | Mostra la | |
|-----------------------|------|-----------------|------------------|-------|
| | | | schermata | |
| | | | "Schermata | |
| | | | mappa" | |
| SUBVARIATION #1 | Step | Utente | Sistema | Марра |
| L'utente apre la | 1a | Svolge gli step | | |
| mappa dalla struttura | | 1-4 dello use | | |
| | | case | | |
| | | "Visualizza | | |
| | | struttura" | | |
| | 2a | Preme il tasto | | |
| | | "Apri su | | |
| | | mappa" | | |
| | 3a | | Carica la | |
| | | | posizione | |
| | | | dell'utente e la | |
| | | | singola | |
| | | | struttura | |
| | 4a | | Torna allo step | |
| | | | 6 dello | |
| | | | scenario | |
| | | | principale | |
| EXTENSION #1 | Step | Utente | Sistema | Марра |
| L'utente ha il GPS | 4.1 | | Mostra pop-up | |
| disabilitato e lo | | | di avvertenza | |
| abilita | 5.1 | Preme il tasto | | |
| | | "Conferma" ed | | |
| | | abilita il GPS | | |

| | 6.1 | | Torna allo step | |
|-------------------------|------|----------------|-----------------|-------|
| | | | 4 dello | |
| | | | scenario | |
| | | | principale | |
| EXTENSION #2 | Step | Utente | Sistema | Марра |
| L'utente non abilita il | 4.2 | | Mostra pop-up | |
| GPS | | | di avvertenza | |
| | 5.2 | Preme il tasto | | |
| | | "Annulla" | | |
| | 6.2 | | Torna allo step | |
| | | | 5 dello | |
| | | | scenario | |
| | | | principale | |

Scrittura recensione:

| Use Case #5 | | Scrive recensione | |
|-----------------------|--|-----------------------------|--------------------------|
| Goal in context | L'utente è interessato a pubblicare una sua recensione relativa ad una determinata struttura | | |
| Precondition | None | | |
| Success end condition | L'uten | te scrive e pubblica la red | censione |
| Failed end condition | Annull | a la scrittura della recens | sione o cambia schermata |
| Primary actor | Utente | e Autenticato | |
| Trigger | Preme | e "Scrivi recensione" | |
| DESCRIPTION | Step | Utente Autenticato | Sistema |
| | 1 | Svolge gli step 1-4 | |
| | | dello use case | |
| | | "Visualizza struttura" | |
| | 2 | Preme il tasto "Scrivi | |
| | | una recensione" | |
| | 3 | | Mostra la schermata |
| | | | "Scrivi recensione" |
| | 4 | Riempie il campo | |
| | | descrizione di almeno | |
| | | 100 caratteri e da un | |
| | | voto da 1 a 5 stelle | |
| | 5 | | il tasto "Pubblica" |
| | | | viene attivato |
| | 6 | Preme il tasto | |
| | | "Pubblica" | |
| | 7 | | Mostra pop-up di |
| | | | successo |
| | 8 | Preme il tasto "Ok" | |
| | 9 | | Mostra la schermata |
| | | | "Visualizza struttura" |

| EXTENSION #1 | Step | Utente Autenticato | Sistema |
|------------------------|------|-----------------------|--------------------------|
| L'utente annulla | 4.1 | Preme il tasto "Home" | |
| dopo aver scritto | | o "Indietro" | |
| qualcosa o aver scelto | 5.1 | | Mostra pop-up di |
| una valutazione | | | avvertenza |
| | 6.1 | Preme il tasto | |
| | | "Conferma" | |
| | 7.1 | | Mostra la schermata |
| | | | "Visualizza struttura" e |
| | | | termina lo use case |

Modello di dominio:

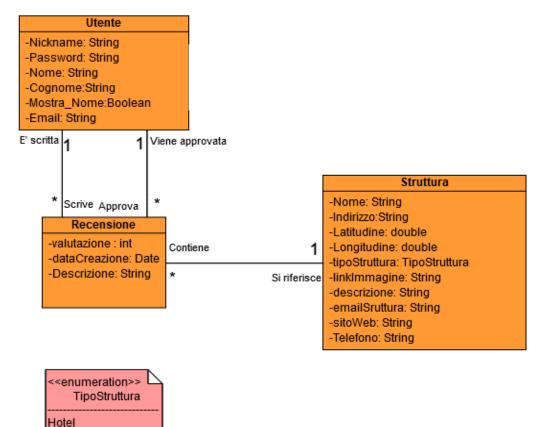
Class diagram

Vengono ora elencati i Class Diagram relativi alle funzionalità che l'applicazione mobile deve offrire. L'individuazione degli oggetti partecipanti è guidata dall'euristica Three-ObjectType, gli oggetti vengono dunque classificati in:

| Object type, gir oggetti verigorio durique ciassineati in. |
|--|
| □ Entity – Modellano l'informazione persistente. |
| □ Boundary – Modellano le interazioni tra gli attori e il sistema. |
| □ Control – Modellano la logica necessaria a svolgere lo use case. |
| Al fine di aumentare la leggibilità dei Class Diagram entity, boundary e control presentano colori differenti: |
| □ Le classi <i>entity</i> sono colorate in verde |
| □ Le classi <i>boundary</i> sono colorate in viola |
| □ Le classi <i>control</i> sono colorate in giallo |
| Inoltre, ove possibile, le classi sono raggruppate a seconda del loro tipo. |

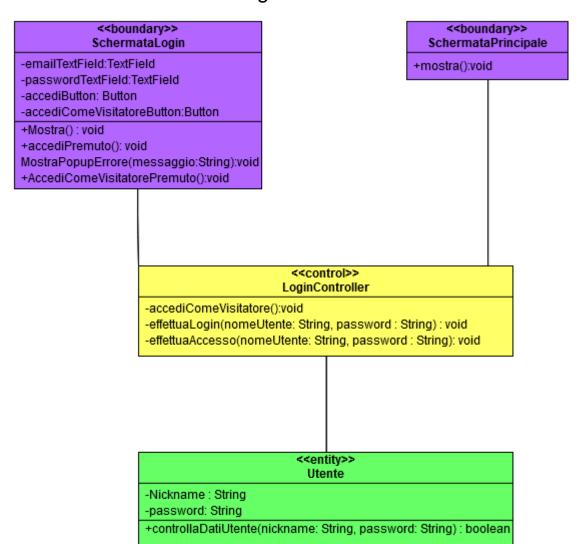
Unica eccezione all'euristica è il Class diagram del database.

Database

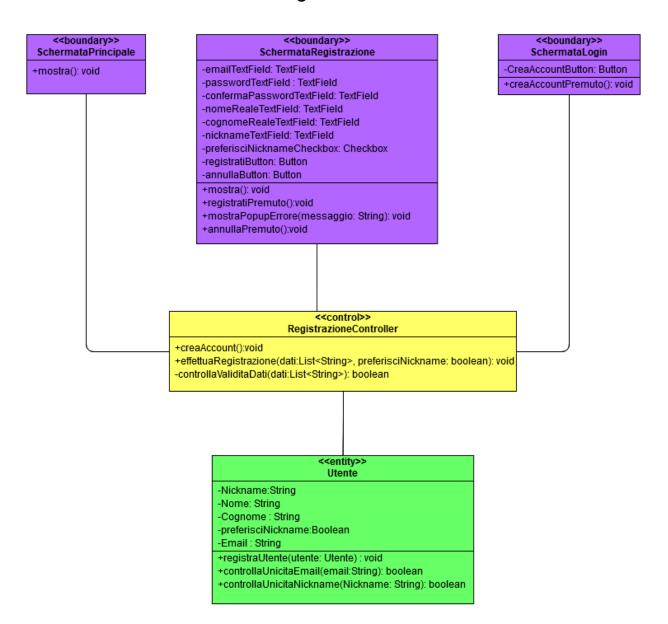


Ristorante AttrazioneTuristica

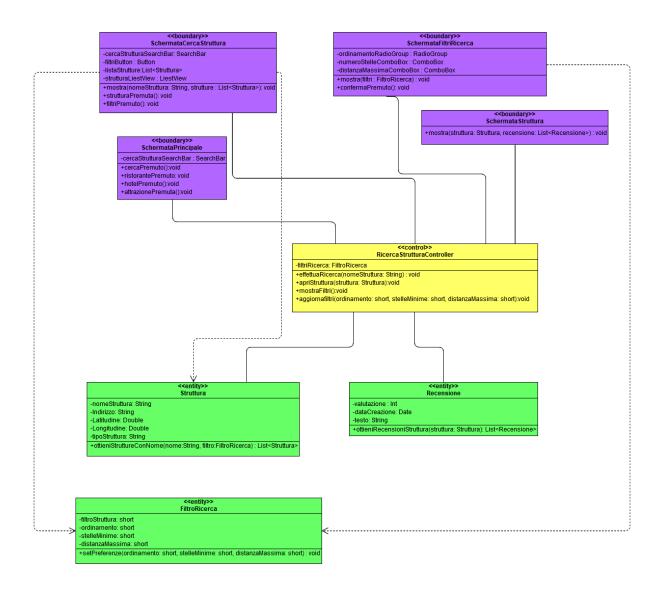
Login utente



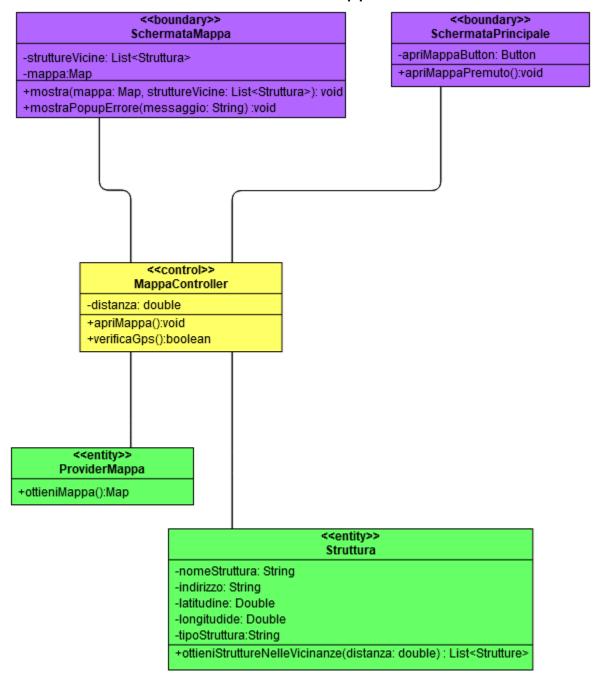
Registrazione



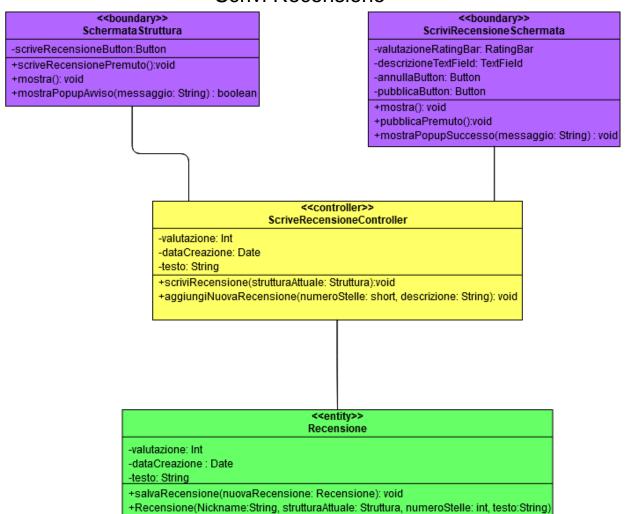
Ricerca Struttura



Visualizza Mappa



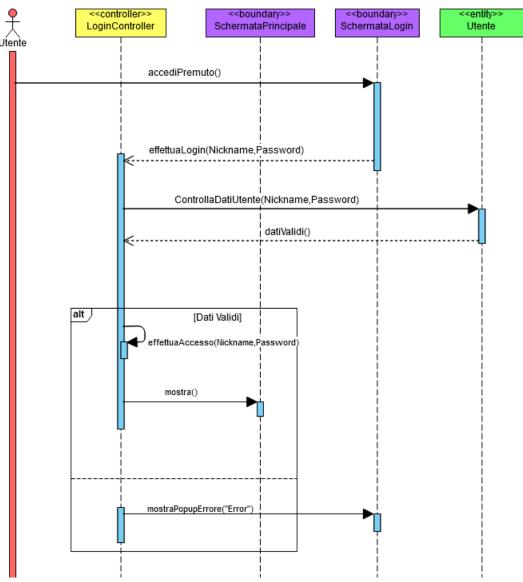
Scrivi Recensione



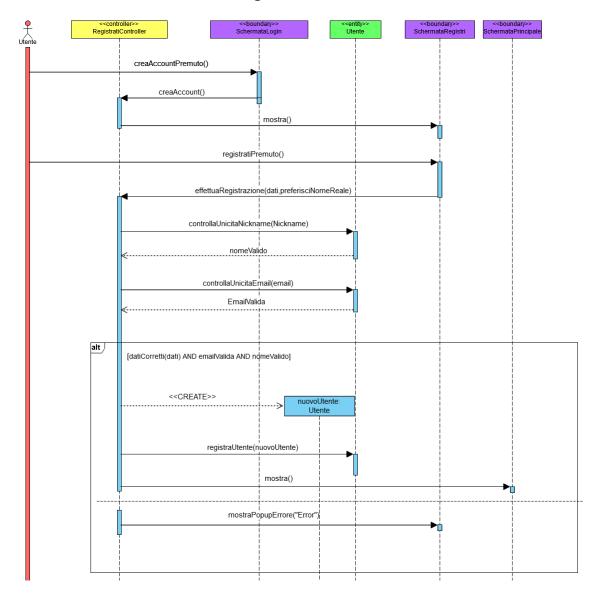
Sequence Diagram

Vengono ora elencati i Sequence Diagram relativi alle funzionalità che l'applicazione deve offrire.

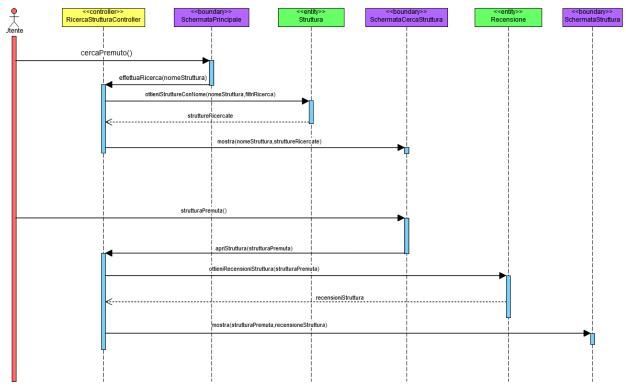
Login Utente



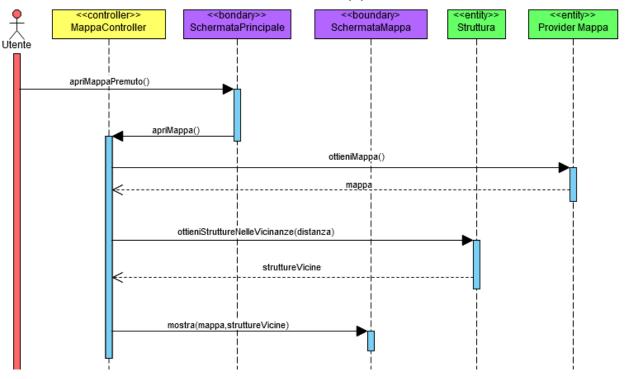
Registrazione



Ricerca Struttura



Visualizza Mappa



Scrivi Recensione

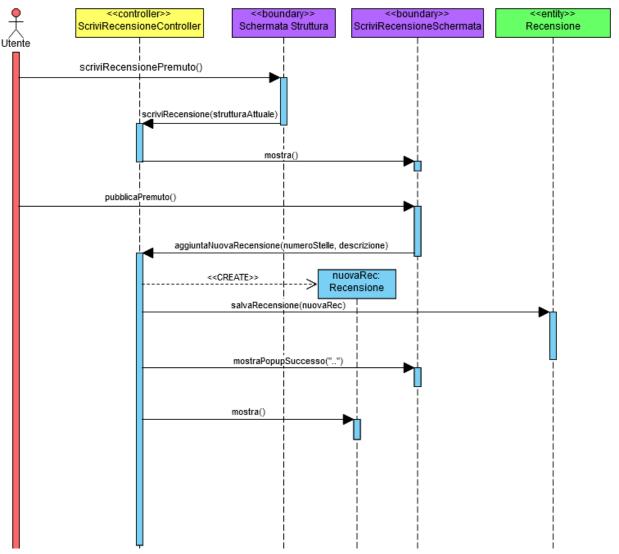


Diagramma di stato di analisi: Ricerca Strutture

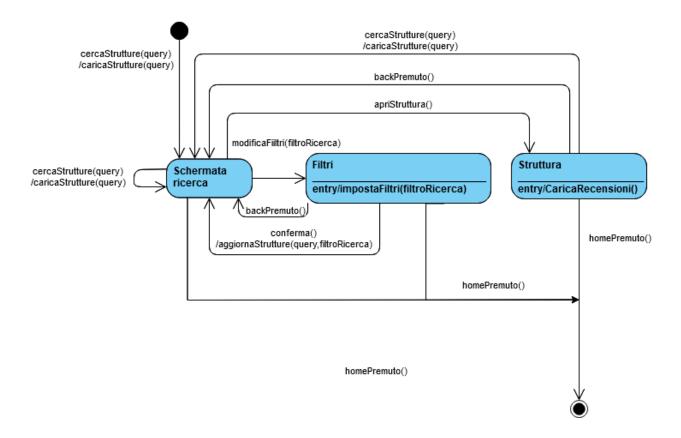
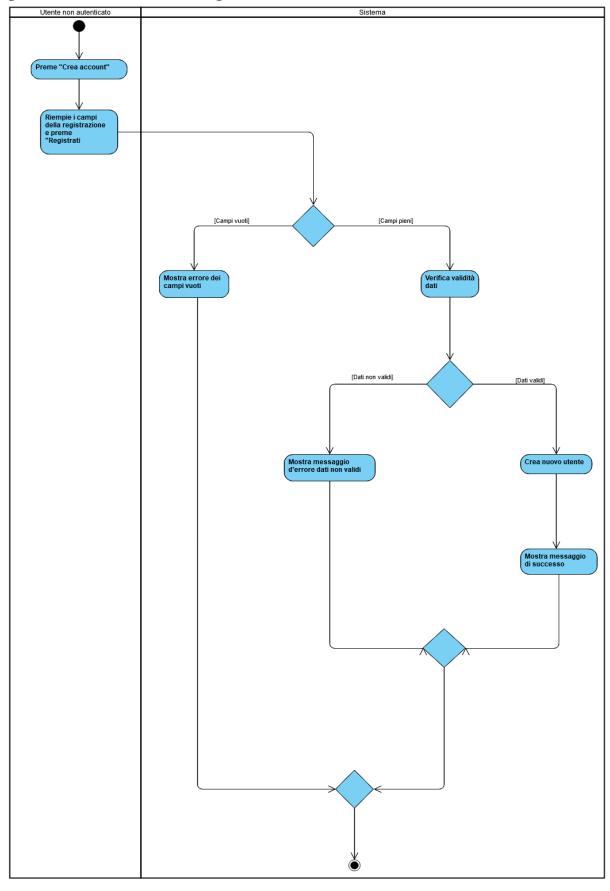


Diagramma di attività: Registrazione utente



Pianificazione dettagliata dell'attività

Viene ora rappresentata una **proposta** di impegno risorse e pianificazione dettagliata delle attività con l'ausilio del seguente diagramma di Gantt:



Come osservabile dal diagramma è stato adottato il modello a cascata essendo questo altamente comprensibile, visibile (distacco quasi netto tra una fase e l'altra), supportabile (non necessità di tool di supporto complessi).

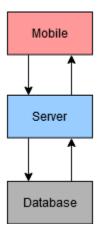
I rischi del modello a cascata sono noti: Dal momento che l'output di ogni fase è di vitale importanza per la fase successiva è necessario minimizzare gli errori al fine di evitare ingenti ritardi dovuti ad un riciclo di tutte le fasi.

Capitolo 3

Documento di Design del sistema Analisi dell'architettura

Architettura esterna:

L'architettura esterna del sistema è suddivisa in tre differenti layers, ognuno dei quali può accedere esclusivamente al layer immediatamente sottostante, rendendo dunque l'architettura chiusa.



Il database e l'app potranno essere interpellati esclusivamente da un server. Quest'ultimo ha lo scopo di far da tramite tra l'applicativo mobile e il Database.

Client e server sono stati scritti in Kotlin. Un linguaggio di programmazione che fa della semplicità il suo asso forte: tipi opzionali, interoperabilità con java garantita al 100%, sintassi chiara e immediata e tante altre features che si ritengono hanno agevolato di molto il lavoro, nonché ridotto il tempo di sviluppo.

L'architettura three-type e le relative sotto architetture client/server che sono state utilizzate, permettono di poter eventualmente aggiungere nuovi client

e metodi di memorizzazione in futuro, aspetto considerato fondamentale nell'implementazione dell'applicativo, poiché la volontà è di rendere il servizio quanto più scalare possibile, considerando l'aggiunta di un nuovo client (ES: Desktop) o di un nuovo metodo di memorizzazione dei dati (ES: DB Oracle).

Le misure atte a far ciò sono descritte nei successivi due paragrafi.

Architettura server:

Per il server è stato utilizzato il framework Spring Boot 2.0. Questo framework è pensato per consentire di sviluppare servizi web, permettendo di lavorare su tutti i livelli di sviluppo: dalla sicurezza al web vero e proprio.

Era chiaro già dal modello funzionale che si doveva fare una distinzione tra un'utente visitatore (che non può scrivere recensioni) e utente registrato (che può scriverle).

Per la gestione dell'autenticazione e per rispettare questo requisito è stato utilizzato anche Spring Security che - oltre a gestire l'autenticazione - gestisce anche le autorizzazioni relative ad un singolo utente.

La scelta di questo framework comporta anche un'ottima scalabilità al nostro sistema distribuito: con l'aggiunta di nuove feature sarà immediato dare maggiori permessi agli utenti registrati, o addirittura creare nuovi ruoli.

Come già detto nel paragrafo precedente, il sistema garantisce una scalabilità tale da rimanere stabile alla modifica del modello di persistenza dei dati. Per far ciò è stato adottato come design pattern il DAO (Data Access Object) e il Factory Method.

Il DAO, mediante l'utilizzo di interfacce, permette di modificare come vengono memorizzati i dati, in modo da poter cambiare il database senza andare ad impattare sul resto del codice.

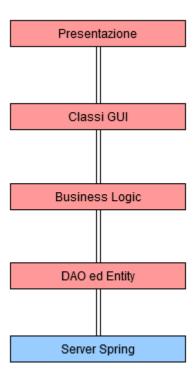
Una problematica di utilizzare una classe DAO come versione concreta e che, alla sua modifica, tutte le classi che dipendono da essa devono essere ricompilate; così come, cambiando il database esso non diventa riciclabile.

Per venire in contro a questo problema si trasforma la classe DAO in un'interfaccia e le sue realizzazioni concrete verranno create mediante la classe Factory. Così facendo, e possibile creare nuove versioni concrete senza impattare il codice preesistente, modificando unicamente la Factory.

Architettura client:

L'applicativo mobile è stato scritto per Android, che per l'interfaccia grafica utilizza XML, fortunatamente grazie ad i vari IDE (Android Studio) è sempre più semplice approcciarsi al design dell'app, rendendo raro lo "sporcarsi le mani" con XML, questo si è tradotto in tempi di sviluppo minori ed efficacia del nostro sistema.

Il client ha un'architettura multilivello illustrata di seguito:



Nel dettaglio:

Livello 0: Server Spring. Al livello più basso risiedono le chiamate al server per ottenere i dati.

Livello 1: DAO ed Entity. Questo livello e formato da interfacce, classi concrete ed una factory. Sono stati utilizzati gli stessi pattern del server in modo da garantire gli stessi vantaggi anche ai client.

Livello 2: Business logic: Si occupa di realizzare le funzionalità richieste.

Livello 3: GUI. Formato da classi che controllano l'interfaccia grafica, si occupano di acquisire dati e inizializzare l'interfaccia grafica

Livello 4: Presentazione. Formato da XML che descrive le pagine a livello visivo.

Servizi cloud utilizzati:

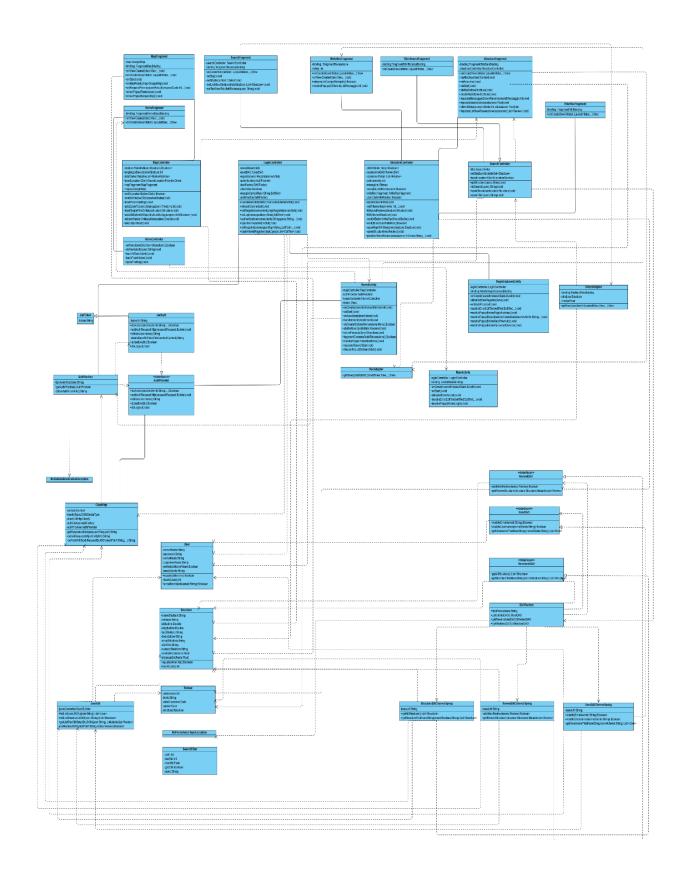
AWS è la piattaforma cloud più completa al momento disponibile, di fondamentale importanza è stato per la gestione -in termini di risorse- del server; questo ci ha permesso di concentrarci sui requisiti del cliente e nell'offrigli le risorse a lui più adeguate, in particolare si è utilizzato:

- Elastic Beanstalk: e un servizio web che gestisce automaticamente implementazione, offre un provisioning di capacità e auto scaling al monitoraggio della salute dell'applicazione. Su questo servizio è stato caricato il server Spring dove in pochi minuti era già pronto per l'uso senza dover effettuare alcuna "mano d'opera" sull'infrastruttura o delle risorse.
- Relational Database Service (RDS): è un servizio che fornisce una capacità ridimensionabile a un costo conveniente, ovvero le prestazioni offerte sono proporzionali al numero di richieste (Ottimo per la scalabilità!). Inoltre, automatizza al tempo stesso le attività di amministrazione del database più dispendiose (diminuita la complessità del sistema). Il motore su cui è stato fatto girare il DB è MySQL.

Diagramma delle classi di design

Segue il class diagram di design Per renderlo più chiaro:

- Le associazioni tra classi troppo distanti tra di loro, in particolare quelle più "tecniche" sono state omesse.
- I metodi privati e i metodi get e set sono stati omessi, salvo casi eccezionali.



CRC Cards

In questo paragrafo sono riportate le CRC cards (Class Responsability Collaborators) utilizzate per dettagliare la collaborazione tra le classi e le loro responsabilità.

Nella sezione "Superclassi" verranno contenute sia interfacce che superclassi, rispettando l'ereditarietà in Kotlin che permette di estendere, allo stesso tempo, una singola classe ma più interfacce.

La superclasse sarà preceduta dalla parola "extends", le interfacce dalla parola "implements".

L'ordine in cui sono rappresentate le classi rispetta la sintassi del Kotlin, quindi: prima la superclasse (se presente) e poi le interfacce (sempre se presenti).

| Nome classe | FilterRecFragment |
|------------------------------|---------------------|
| Superclasse | extends Fragment |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta la schermata dei | HomeActivity |
| filtri delle recensioni | StructureController |

| Nome classe | FilterSearchFragment |
|------------------------------|----------------------|
| Superclasse | extends Fragment |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta la schermata dei | HomeActivity |
| filtri della ricerca | SearchController |
| | FilterSearch |

| Nome classe HomeFragment | Nome classe |
|--------------------------|-------------|
|--------------------------|-------------|

| Superclasse | extends Fragment |
|--------------------------|------------------|
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta la schermata | HomeActivity |
| principale | HomeController |

| Nome classe | MapFragment |
|--------------------------|------------------|
| Superclasse | extends Fragment |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta la schermata | HomeActivity |
| della mappa | MapController |
| | Structure |

| Nome classe | SearchFragment |
|--------------------------|------------------|
| Superclasse | extends Fragment |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta la schermata | StructureAdapter |
| di ricerca | SearchController |
| | Structure |

| Nome classe | WriteRecFragment |
|--------------------------|---------------------|
| Superclasse | extends Fragment |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta la schermata | HomeActivity |
| di scrittura recensioni | StructureController |
| | Structure |

| i | | 1 |
|---|----------------|-------------------|
| | Nome classe | StructureFragment |
| | Superclasse | extends Fragment |
| | Sottoclassi | |
| | Responsabilità | Collaborators |

| Nome classe | HomeActivity |
|-----------------------------------|---------------------------|
| Superclasse | extends AppCompatActivity |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta l'activity principale | AuthFactory |
| dell'applicativo Android | AuthProvider |
| | HomeController |
| | MapController |
| | HomeFragment |
| | WriteRecFragment |

| Nome classe | MainActivity |
|---------------------------------|------------------|
| Superclasse | extends Activity |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta l'activity di Login | LoginController |

| Nome classe | RegistrazioneActivity |
|---------------------------|-----------------------|
| Superclasse | extends Fragment |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta l'activity di | LoginController |
| registrazione | |

| Nome classe | RecensioneAdapter |
|-------------|--|
| Superclasse | extends ArrayAdapter <review></review> |
| Sottoclassi | |
| | |

| Responsabilità | Collaborators |
|---------------------------------|---------------|
| Rappresenta la singola cella di | |
| Recensione una recensione | |

| Nome classe | StructAdapter |
|----------------------------------|--|
| Superclasse | extends ArrayAdapter <structure></structure> |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta la singola cella di | Structure |
| una struttura durante la ricerca | |

| Nome classe | HomeController |
|----------------------------------|----------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Gestisce la actionbar di Android | HomeActivity |
| | MainActivity |
| | AuthProvider |
| | HomeFragment |
| | MapFragment |
| | SearchFragment |

| Nome classe | LoginController |
|-----------------------------------|-----------------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Gestisce la funzionalità relativa | HomeActivity |
| allo use case "Si registra" | MainActivity |
| Gestisce la funzionalità relativa | RegistrazioneActivity |
| allo use case "effettua log-in" | AuthFactory |

| AuthProvider |
|--------------|
| DAOFactory |
| UtenteDAO |
| User |
| ClientHttp |

| Nome classe | MapController |
|-----------------------------------|-------------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Gestisce la funzionalità relativa | HomeActivity |
| allo use case "Visualizza mappa" | |
| | DAOFactory |
| | StrutturaDAO |
| | Structure |
| | MapFragment |
| | StrutturaFragment |

| Nome classe | RicercaController |
|-----------------------------------|-----------------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Gestisce la funzionalità relativa | DAOFactory |
| allo use case "Visualizza | StrutturaDAO |
| Struttura" | Structure |
| | FiltroRicerca |
| | FiltriRicercaFragment |
| | SearchFragment |
| | StrutturaFragment |

| Nome classe | StructureController |
|-----------------------------------|---------------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Gestisce la funzionalità relativa | AuthFactory |
| allo use case "Scrive recensione" | DAOFactory |
| Gestisce i filtri per la | Review |
| visualizzazione delle recensioni | Structure |
| | User |
| | MapFragment |
| | WriteRecFragment |
| | FilterRecFragment |
| | StructureFragment |

| Nome classe | DAOFactory |
|----------------------------------|----------------------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Fabbrica versioni concrete delle | UserDAO |
| interfacce DAO | UserDAOServerSpring |
| | ReviewDAO |
| | ReviewDAOServerSpring |
| | StructureDAO |
| | StructureDAOServerSpring |
| | NoPersistanceTypeException |

| Nome classe | ReviewDAOServerSpring |
|-----------------------------|-----------------------|
| Superclasse | implements ReviewDAO |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Implementa l'interfaccia | Review |
| RecensioneDAO per un server | Structure |
| spring | User |
| | ClientHttp |
| | JsonUtil |

| Nome classe | StructureDAOServerSpring |
|----------------------------|--------------------------|
| Superclasse | implements StructureDAO |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Implementa l'interfaccia | Struttura |
| StrutturaDAO per un server | ClientHttp |
| spring | JsonUtil |

| Nome classe | UserDAOServerSpring |
|--------------------------------|---------------------|
| Superclasse | implements UserDAO |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Implementa l'interfaccia | User |
| UtenteDAO per un server spring | ClientHttp |
| | JsonUtil |

| Nome classe | ClientHttp |
|----------------------------------|---------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Permette di effettuare richieste | AuthFactory |
| HTTP | AuthProvider |

| Nome classe | JsonUtil |
|-------------------------|---------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Converte JSON in entità | Review |
| | Structure |
| | User |

| Nome classe | NoValidAuthenticationException |
|-------------------------------------|--------------------------------|
| Superclasse | extends RuntimeException |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta un'eccezione che | |
| viene lanciata quando non viene | |
| trovato il metodo di autenticazione | |
| Non valido | |

| Nome classe | NoPersistanceTypeException |
|----------------------------------|----------------------------|
| Superclasse | extends RuntimeException |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta un'eccezione che | |
| viene lanciata quando non viene | |
| trovato il metodo di persistenza | |
| nelle di configurazione | |

| Nome classe | Review |
|----------------------------|---------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta una recensione | User |
| | Structure |

| Nome classe | Structure |
|---------------------------|---------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta una struttura | |

| Nome classe | User |
|-----------------------|---------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta un utente | |

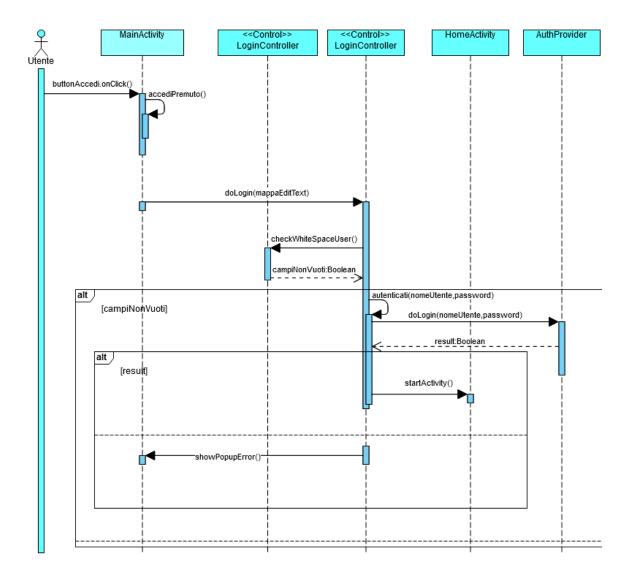
| Nome classe | AuthFactory |
|-------------------------------|--------------------------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Fabbrica versioni concrete di | AuthProvider |
| GestoreAutenticazione | JwtAutenticazione |
| | NoValidAuthenticationException |

| Nome classe | JwtAutenticazione |
|---------------------------|-------------------------|
| Superclasse | implements AuthProvider |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Permette l'autenticazione | JwtToken |
| tramite token JWT | ClientHttp |

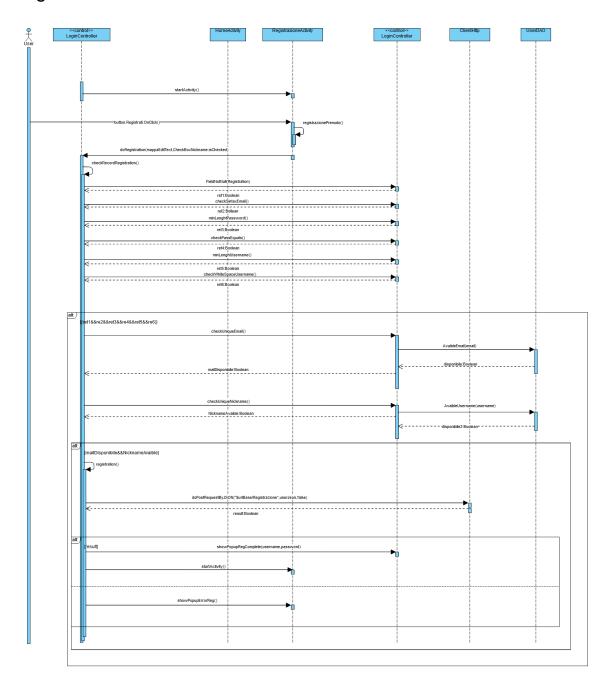
| Nome classe | JwtToken |
|--------------------------|---------------|
| Superclasse | |
| Sottoclassi | |
| Responsabilità | Collaborators |
| Rappresenta un token JWT | |

Diagrammi di sequenza di design

Login utente



Registrazione



Capitolo 4

Documento di Testing del sistema

Test Plan per System Testing

Il test plan che segue è stato scritto per organizzare la fase di testing del sistema al fine di verificare e dimostrare che tutti i requisiti esplicitati precedentemente siano stati correttamente implementati e siano funzionanti.

Test registrazione

| Test ID | 1 | |
|-------------------------------|---|--------------|
| Nome test | Test registrazione | |
| Descrizione test | L'obiettivo di questo test e veri care la | funzionalità |
| | di registrazione | |
| Input | Risultato desiderato | Risultato |
| | | ottenuto |
| Riempie i campi con dati | Registrazione effettuata, visualizza il | Superato |
| validi e preme il tasto | messaggio di successo "Registrazione | |
| "Registrati" | avvenuta con successo" | |
| Preme il tasto "Ok" al | Visualizza la schermata | Superato |
| messaggio di successo | "Schermata Principale" | |
| Preme il tasto "Registrati" | Registrazione fallita, evidenzia tutti i | Superato |
| lasciando tutti i campi vuoti | campi con il messaggio. Il campo non | |
| | può essere vuoto | |

| Preme il tasto "Registrati" | Registrazione fallita, evidenzia il | Superato |
|-------------------------------|-------------------------------------|----------|
| lasciando un qualsiasi | campo vuoto con il messaggio II | |
| singolo campo vuoto | campo non può essere vuoto" | |
| Inserisce "Noctino52 " nel | Registrazione fallita, evidenzia il | Superato |
| campo nome Utente, riempie | campo Nome Utente con il messaggio | |
| gli altri con dati validi e | "Nome utente già in uso" | |
| preme il tasto "Registrati" | | |
| Inserisce "a" nel campo | Registrazione fallita, evidenzia il | Superato |
| nome Utente, riempie gli | campo Nome Utente con il messaggio | |
| altri con dati validi e preme | "Il nome utente deve essere lungo | |
| il tasto "Registrati" | almeno 4 caratteri" | |
| Inserisce "a" nel campo | Registrazione fallita, evidenzia il | Superato |
| email, riempie gli altri con | campo email con il messaggio "Il | |
| dati validi e preme il tasto | valore inserito non corrisponde ad | |
| "Registrati" | una email" | |
| Inserisce | | |
| "NoctinoMisterioso@it.it" nel | Registrazione fallita, evidenzia il | Superato |
| campo email, riempie gli | campo email con il messaggio "Email | |
| altri con dati validi e preme | già in uso" | |
| il tasto "Registrati" | | |
| Inserisce "1234" nel campo | Registrazione fallita, evidenzia il | Superato |
| password, "1234" nel campo | campo password con il messaggio "La | |
| conferma password, riempie | password deve essere lunga almeno 8 | |
| gli altri con dati validi e | caratteri" | |
| preme il tasto "Registrati" | | |
| Inserisce "12345678" nel | Registrazione fallita, evidenzia il | Superato |
| campo password, "1234" nel | campo conferma password con il | |
| campo conferma password, | messaggio "La password deve essere | |
| riempie gli altri con dati | lunga almeno 8 caratteri" | |
| validi e preme il tasto | | |
| "Registrati" | | |
| | 1 | 1 |

| Inserisce "12345678" nel | Registrazione fallita, mostra il | Superato |
|-----------------------------|-------------------------------------|----------|
| campo password, | messaggio d'errore "Le password non | |
| "12345678910" nel campo | Corrispondono" | |
| conferma password, riempie | | |
| gli altri con dati validi e | | |
| preme il tasto "Registrati" | | |
| Preme il tasto "Annulla" | Mostra il messaggio d'errore | Superato |
| con tutti i campi riempiti | Tornando indietro i valori inseriti | |
| | saranno persi. Continuare?" | |
| Preme il tasto "Annulla" | Mostra il messaggio d'errore | Superato |
| con un qualsiasi singolo | Tornando indietro i valori inseriti | |
| campo riempito | saranno persi. Continuare?" | |
| Preme il tasto "Annulla" | Visualizza la schermata | Superato |
| senza inserire niente | "Schermata Iniziale" | |

Dati validi:

Nome utente = "Noctino52" -

Password = "12345678"

Conferma password = "12345678"

Email = "NoctinoMisterioso@it.it"

Nome = "Ivan"

Cognome = "Capasso"

Test login utente

| Test ID | 2 | |
|-----------------------------|--|-----------|
| Nome test | Test login utente | |
| Descrizione test | L'obiettivo di questo test e veri care la funzionalità | |
| | di login per gli utenti | |
| Input | Risultato desiderato | Risultato |
| | | ottenuto |
| Inserisce "Noctino52" nel | Login effettuato, visualizza la | Superato |
| campo Nome Utente, | schermata "SchermataPrincipale" | |
| "ConsigliaViaggi" nel campo | | |
| password e preme il tasto | | |

| "Login" | | |
|--|--|----------|
| Inserisce "Ivan" nel campo Nome Utente, | Login fallito, visualizza messaggio di | Superato |
| "ConsigliaViaggi" | errore "Nome Utente o password | |
| nel campo password e preme | errati. Controlla i valori inseriti e | |
| il tasto "Login" | Riprova" | |
| Inserisce "Noctino52" nel | Login fallito, visualizza messaggio di | Superato |
| campo Nome Utente, | errore "Nome Utente o password | |
| "ingsw" nel campo password | errati. Controlla i valori inseriti e | |
| e preme il tasto "Login" | Riprova" | |
| Inserisce "aaa" nel campo | Login fallito, visualizza messaggio di | Superato |
| Nome Utente, "bbb" nel | errore "Nome Utente o password | |
| campo password e preme il | errati. Controlla i valori inseriti e | |
| tasto "Login" | Riprova" | |
| Preme il tasto "Login" | Login fallito, evidenzia entrambi i | Superato |
| senza inserire niente | campi con il messaggio "Il campo non | |
| | può essere vuoto" | |
| Inserisce "Noctino52" nel | Login fallito, evidenzia il campo | Superato |
| campo Nome Utente e | password con il messaggio "Il campo | |
| preme il tasto "Login" | non può essere vuoto" | |
| | | |

Test filtra strutture

| Test ID | 3 | | |
|---------------------------------|---|-----------|--|
| Nome test | Test filtra strutture | | |
| Descrizione test | L'obiettivo di questo test e veri care che l'utente | | |
| | riesca a filtrare le strutture | | |
| Input | Risultato desiderato | Risultato | |
| | | ottenuto | |
| Ha il GPS abilitato e cerca | Viene mostrato l'elenco delle | Superato | |
| "s" sulla barra di ricerca | strutture ordinate per distanza | | |
| Ha il GPS disabilitato e | Viene mostrato l'elenco delle Supo | | |
| cerca "s" sulla barra di | strutture ordinate per numero | | |
| ricerca | crescente di stelle | | |
| Apre i filtri di ricerca con il | Viene mostrato l'elenco dei filtri con | Superato | |
| GPS abilitato | tutti i filtri abilitati | | |

| Apre i filtri di ricerca con il | Viene mostrato l'elenco dei filtri con i | Superato |
|---------------------------------|---|----------|
| GPS disabilitato | filtri "Distanza massima per le | |
| | Strutture" e "ordina per distanza" | |
| | disabilitati | |
| Seleziona l'ordinamento | Viene mostrato l'elenco delle | Superato |
| "Ordina per stelle | strutture ordinate per numero | |
| Crescenti", nessun filtro e | crescente di stelle | |
| preme il tasto "Conferma" | | |
| Seleziona l'ordinamento | Viene mostrato l'elenco delle | Superato |
| "Ordina per stelle | strutture ordinate per numero | |
| Decrescenti", nessun filtro e | decrescente di stelle | |
| preme il tasto "Conferma" | | |
| | | |
| Note | Il database deve contenere le strutture elencate in | |
| | basso. | |

Test visualizza mappa

| Test ID | 4 | | |
|------------------------------|--|-----------|--|
| Nome test | Test visualizza mappa | | |
| Descrizione test | L'obiettivo di questo test e verificare che l'utente | | |
| | riesca a visualizzare la mappa | | |
| Input | Risultato desiderato | Risultato | |
| | | ottenuto | |
| Preme il tasto "Apri | Viene aperta la mappa, sono visibili | Superato | |
| Mappa" dalla schermata | tutte le strutture e la posizione | | |
| principale con GPS abilitato | dell'utente | | |
| Preme il tasto "Apri | Viene mostrato il messaggio di | Superato | |
| Mappa" dalla schermata | avvertenza "II GPS sembra essere | | |
| principale con GPS | disabilitato, alcune funzionalità | | |
| disabilitato | saranno limitate. Vuoi attivarlo?" | | |
| Preme il tasto "Conferma" | Viene reindirizzato alle impostazioni | Superato | |
| sul pop-up del GPS | del dispositivo per abilitarlo | | |
| disabilitato | | | |
| Preme il tasto "Annulla" sul | Viene aperta la mappa, sono visibili | Superato | |
| pop-up del GPS disabilitato | tutte le strutture ma non la posizione dell'utente | | |
| Preme il tasto "Visualizza | Viene aperta la mappa con solo la | Superato | |
| su mappa" dalla schermata | struttura e la posizione dell'utente | | |
| di una struttura con GPS | | | |
| abilitato | | | |
| Preme il tasto "Visualizza | Viene mostrato il messaggio di | Superato | |
| su mappa" dalla schermata | avvertenza "II GPS sembra essere | | |
| di una struttura con GPS | disabilitato, alcune funzionalità | | |
| disabilitato | saranno limitate. Vuoi attivarlo?" | | |
| Preme il tasto "Visualizza | Viene aperta la mappa con solo la | Superato | |
| su mappa" e "Annulla" sul | struttura e senza la posizione | | |
| pop-up del GPS disabilitato | dell'utente Superato | | |

Test scrittura recensione

| Test ID | 5 | | |
|-----------------------------|---|-----------|--|
| Test name | Test scrittura recensione | | |
| Test description | L'obiettivo di questo test e verificare la funzionalità | | |
| | di scrittura recensioni | | |
| Input | Risultato desiderato | Risultato | |
| | | ottenuto | |
| Scrive una recensione | Il tasto "Pubblica" è abilitato | Superato | |
| maggiore di 100 caratteri e | | | |
| preme una delle 5 stelle | | | |
| Scrive una recensione | Viene mostrato il messaggio di | Superato | |
| maggiore di 100 caratteri, | successo "Pubblicazione Avvenuta con successo!" | | |
| preme una delle 5 stelle e | | | |
| preme il tasto "Pubblica" | | | |
| Preme una delle 5 stelle e | Viene mostrato il messaggio di | Superato | |
| preme il tasto "Home" | avvertenza "Tornando alla home | | |
| | perderai i dati inseriti. Continuare?" | | |
| Scrive "test" come | Viene mostrato il messaggio di | Superato | |
| recensione e preme il tasto | avvertenza "Tornando alla home | | |
| "Home" | perderai i dati inseriti. Continuare?" | | |
| Preme una delle 5 stelle e | Viene mostrato il messaggio di | Superato | |
| preme il tasto "Annulla" | avvertenza "Tornando indietro i | | |
| | valori inseriti saranno persi. | | |
| | Continuare?" | | |
| Scrive "test" come | Viene mostrato il messaggio di | Superato | |
| recensione e preme il tasto | avvertenza "Tornando indietro i | | |
| "Annulla" | valori inseriti saranno persi. | | |
| | Continuare?" | | |
| Preme una delle 5 stelle e | Il tasto "Pubblica" è disabilitato | Superato | |
| non scrive nulla | | | |

| Preme una delle 5 stelle e | Il tasto "Pubblica" è disabilitato | Superato |
|----------------------------|--|----------|
| scrive una descrizione | | |
| minore di 100 caratteri | | |
| Scrive una recensione | Il tasto "Pubblica" è disabilitato | Superato |
| maggiore di 100 caratteri | | |
| ma non preme nessuna delle | | |
| 5 stelle | | |
| Note | L'utente è autenticato e non ha già recensito la | |
| | struttura | |

Codice xUnit per unit testing di 2 metodi

I test di unita che seguono sono stati scritti ed eseguiti utilizzando il framework JUnit.

Unit testing del metodo ottieniStella

Il metodo ottieniStella si occupa di valutare se una stella debba essere rappresentata come una stella vuota, mezza o piena.

Il metodo ha due parametri: il primo rappresenta la stella da esaminare, il secondo la valutazione media della struttura. Restituisce:

Una stella piena se la differenza tra la stella e la valutazione e minore di 0.3;

Una stella a meta se la differenza tra la stella e la valutazione e maggiore di 0.3 ma minore di 0.75;

Una stella vuota altrimenti.

Una stella per essere valida deve avere un valore compreso tra 1 a 5 nel dominio degli interi, mentre la valutazione di una struttura nell'intervallo [0; 5] nel dominio dei float.

```
1.
      fun ottieniStella(numeroStella: Int, valutazione: Float): Int {
       if(numeroStella < 1 || numeroStella > 5 || valutazione < 0 ||</pre>
  valutazione > 5) {
3.
               throw IllegalArgumentException()
4.
5.
         return when {
               (numeroStella-valutazione < 0.3) ->
  R.drawable.ic star black 24dp
              (numeroStella-valutazione <= 0.75) ->
8.
  R.drawable.ic star half black 24dp
9.
             else -> R.drawable.ic star border black 24dp
10.
11.
```

Questo metodo è stato testato sia tramite strategia Black Box che White Box, il primo per verificare il rispetto dei requisiti, il secondo per verificare che non ci fossero comportamenti anomali in alcune parti del metodi.

Per testare il metodo in black box sono state identificate le seguenti classi di equivalenza:

Parametro numeroStella:

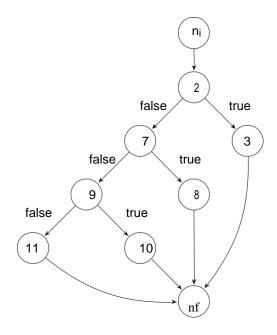
```
{ CE1: [Int:MIN VALUE; 0] (Non valido)
 { CE2: [1; 5] (Valido)
 { CE3: [6; Int:MAX VALUE] (Non valido)
 Parametro valutazione
 { CE4: [Float:MIN VALUE; 0) (Non valido)
 { CE5: [0; 5] (Valido)
 { CE6: (5; Float:MAX VALUE] (Non valido)
```

Seguendo la strategia Weak Equivalence Class Testing sono stati scritti 5 test, uno per ogni classe non valida ed uno massimizzando quelle valide:

- 1. ottieniStellaConNumeroStellaMinoreDiUno()
- 2. ottieniStellaConNumeroStellaMaggioreDiCinque()
- 3. ottieniStellaConValutazioneMinoreDiUno()
- 4. ottieniStellaConValutazioneMaggioreDiCinque()
- 5. ottieniStellaConParametriValidi()

```
1. class StructureFragmentTest {
2.
       private val stellaVuota = R.drawable.ic star border black 24dp
3.
       private val stellaMedia = R.drawable.ic star half black 24dp
4.
       private val stellaPiena = R.drawable.ic star black 24dp
5.
6.
       private lateinit var strutturaFragment: StructureFragment
7.
8.
       @Before
9.
       fun istanziaFragment() {
10.
            val strutturaStub = Struttura("", 2.0, 3.0)
11.
            strutturaFragment = StrutturaFragment(strutturaStub)
12.
13.
14.
        @Test (expected = IllegalArgumentException::class)
15.
        fun ottieniStellaConNumeroStellaMinoreDiUno() {
16.
            strutturaFragment.ottieniStella(-1, 2f)
17.
18.
19.
        @Test (expected = IllegalArgumentException::class)
20.
        fun ottieniStellaConNumeroStellaMaggioreDiCinque() {
21.
             strutturaFragment.ottieniStella(42, 3.5f)
22.
23.
24.
        @Test (expected = IllegalArgumentException::class)
25.
        fun ottieniStellaConValutazioneMinoreDiUno() {
            strutturaFragment.ottieniStella(2, -5f)
26.
27.
28.
29.
        @Test (expected = IllegalArgumentException::class)
30.
        fun ottieniStellaConValutazioneMaggioreDiCinque() {
31.
            strutturaFragment.ottieniStella(4, 33f)
32.
33.
34.
        @Test
35.
        fun ottieniStellaConParametriValidi() {
36.
            val stella = strutturaFragment.ottieniStella(3, 4.5f)
37.
            assertEquals(stellaPiena, stella)
38.
```

Questo invece è il grafo che rappresenta il flusso di controllo del whiteboxing:



Da questo GFC sono stati rilevati i seguenti test per ottenere una Brench Coverage:

```
1. // Test White box
2.
       @Test (expected = IllegalArgumentException::class)
       fun whiteboxPath_2_3() {
3.
4.
           strutturaFragment.ottieniStella(-6, 2f)
5.
       }
6.
7.
       @Test
8.
       fun whiteboxPath 2 7 8() {
           val stella = strutturaFragment.ottieniStella(1, 4.5f)
9.
10.
             assertEquals(stellaPiena, stella)
11.
        }
12.
13.
        @Test
        fun whiteboxPath 2 9 10() {
14.
             val stella = strutturaFragment.ottieniStella(3, 2.5f)
15.
16.
             assertEquals(stellaMedia, stella)
17.
18.
19.
        @Test
20.
        fun whiteboxPath 2 9 11() {
21.
            val stella = strutturaFragment.ottieniStella(5, 4f)
22.
             assertEquals(stellaVuota, stella)
         }
23.
24.
```

Unit testing del metodo verificaNomeUtenteDisponibile

Il metodo verificaNomeUtenteDisponibile è un metodo del server utilizzato nella creazione di un nuovo utente che si occupa di valutare se un nome utente è disponibile oppure già utilizzato da un altro utente: il suo parametro è il nome utente da verificare.

Il valore di ritorno è True se il nome utente e libero ed utilizzabile; False se il nome utente è già utilizzato.

Un nome utente può contenere qualsiasi carattere ad esclusione degli spazi; inoltre, se la sua lunghezza non è compresa tra 4 e 16 caratteri, ritorna false. Questo per coerenza con la definizione del dominio del nickname presente nel DB.

```
1. override fun verificaNomeUtenteDisponibile(nomeUtente: String):
 Boolean{
         if(nomeUtente.length < 4 || nomeUtente.length > 16 ||
 nomeUtente.contains(" ")) {
3.
             return false
4.
5.
         val querySql = "SELECT COUNT(id) as occupato FROM utente WHERE
 nomeUtente = ?"
7.
        val preparedStatement = MySQLConnection.preparaQuery(querySql)
8.
         preparedStatement.setString(1, nomeUtente)
9.
         val resultSet = preparedStatement.executeQuery()
10.
11.
          resultSet.next()
12.
           return !resultSet.getBoolean("occupato")
13.
```

Per testare il metodo in black box sono state identificate le seguenti classi di equivalenza per il parametro nomeUtente:

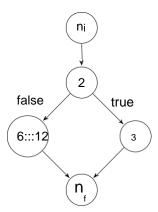
```
CE1: nomeUtente.length 2 [Int:MIN VALUE; 4) (Non disponibile)
CE2: nomeUtente.length 2 (16; Int:MAX VALUE] (Non disponibile)
CE3: nomeUtente.length 2 [4; 16] AND nomeUtente.lowercased (Disponibile)
CE3: nomeUtente.length 2 [4; 16] AND nomeUtente.uppercased (Disponibile)
CE4: nomeUtente.contains("") (Non disponibile)
CE5: @nomeUtente2 Database (Non disponibile)
CE6: @nomeUtente2 Database (Disponibile)
```

Seguendo la strategia Weak Equivalence Class Testing sono stati scritti 6 test, uno per ogni classe "non disponibile" ed uno massimizzando quelle "disponibile":

- nomeUtenteNonDisponibileConLunghezzaMinoreDiQuattro()
- 2. nomeUtenteNonDisponibileConLunghezzaMaggioreDiSedici()
- 3. nomeUtenteNonDisponibileConSpazi()

- 4. nomeUtenteNonDisponibileInMinuscoloNelDatabase()
- 5. nomeUtenteNonDisponibileInMaiuscoloNelDatabase()
- 6. nomeUtenteDisponibileNelDatabase()

Questo invece è il grafo che rappresenta il flusso di controllo del whiteboxing:



Da questo GFC è stata ottenuta una Node Coverage e Brench Coverage attraverso i metodi black box:

- 1. nomeUtenteNonDisponibileConSpazi()
- 2. nomeUtenteNonDisponibileInMinuscoloNelDatabase()

```
1. class UtenteDAOMySQLTest {
      private lateinit var utenteDAO: UtenteDAOMySQL
2.
3.
4.
      @Before
      fun istanziaUtente() {
5.
          utenteDAO = UtenteDAOMySQL()
6.
7.
8.
9.
     @Test
      fun nomeUtenteNonDisponibileConLunghezzaMinoreDiQuattro() {
10.
11.
            assertFalse(utenteDAO.verificaNomeUtenteDisponibile("abc"))
12.
13.
14.
       @Test
15.
       fun nomeUtenteNonDisponibileConLunghezzaMaggioreDiSedici() {
16.
  assertFalse(utenteDAO.verificaNomeUtenteDisponibile("0123456789abcdefh"
  ) )
17.
        }
18.
19.
       @Test
20.
       fun nomeUtenteNonDisponibileConSpazi() {
            assertFalse(utenteDAO.verificaNomeUtenteDisponibile("ciao
 mondo"))
22.
   }
23.
24.
       @Test
25.
       fun nomeUtenteNonDisponibileInMinuscoloNelDatabase() {
26.
 assertFalse(utenteDAO.verificaNomeUtenteDisponibile("Noctino52"))
27.
28.
29.
        @Test
30.
        fun nomeUtenteNonDisponibileInMaiuscoloNelDatabase() {
  assertFalse(utenteDAO.verificaNomeUtenteDisponibile("NOCTINO52"))
32.
33.
34.
35.
        fun nomeUtenteDisponibileNelDatabase() {
  assertTrue(utenteDAO.verificaNomeUtenteDisponibile("test4321"))
37.
       }
38. }
```