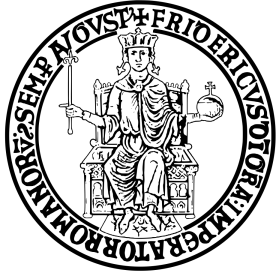
UNIVERSIT**À** DEGLI STUDI DI NAPOLI **FEDERICO II**

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 I

ANNO ACCADEMICO 2021/2022

Specifica, progettazione, implementazione e validazione del Sistema Informativo

“NaTour21”

**Autore:**  **Docenti:**

Gruppo INGSW2122\_S\_03 Prof. Sergio di Martino

Aleks Nikolaev Nikolov Prof. Francesco Cutugno

N86003002

# Indice

[Indice 2](#_Toc99387927)

[Revisioni documentazione 5](#_Toc99387928)

[I - Documento dei Requisiti 6](#_Toc99387929)

[1.1 - Modello funzionale 6](#_Toc99387930)

[1.1.0 - Glossario 6](#_Toc99387931)

[1.1.1 – Presentazione dell’idea progettuale 7](#_Toc99387932)

[1.1.2 – Individuazione del target di utenti 8](#_Toc99387933)

[1.1.3 - Requisiti funzionali 9](#_Toc99387934)

[1.1.4 - Requisiti non funzionali 11](#_Toc99387935)

[1.1.5 – Use Case Diagram 12](#_Toc99387936)

[1.1.6 - Descrizione testuale degli use case 13](#_Toc99387937)

[1.1.7 – Prototipazione visuale 17](#_Toc99387938)

[1.1.8 – Valutazione dell’usabilità a priori 21](#_Toc99387939)

[1.1.9 – Prototipazione funzionale 24](#_Toc99387940)

[1.1.10 – Tabella delle funzioanlità 25](#_Toc99387941)

[1.2 - Modelli di dominio 26](#_Toc99387942)

[1.2.1 – Class diagram di analisi 26](#_Toc99387943)

[Autenticazione (effettua accesso + crea account) 27](#_Toc99387944)

[Recupera account 28](#_Toc99387945)

[Visualizza Home 29](#_Toc99387946)

[Inserisci itinerario 30](#_Toc99387947)

[Visualizza Itinerario Propri 31](#_Toc99387948)

[Visualizza Dettagli Itinerario 32](#_Toc99387949)

[Visualizza Profilo 33](#_Toc99387950)

[1.2.2 – Sequence diagram di analisi 34](#_Toc99387951)

[Inserisci Itinerario Mappa 35](#_Toc99387952)

[Inserisci Itinerario GPX 36](#_Toc99387953)

[1.2.3 – Activity diagram 37](#_Toc99387954)

[Recupero Account 37](#_Toc99387955)

[II - Documento di Design 38](#_Toc99387956)

[2.1 - Archittettura del sistema 38](#_Toc99387957)

[2.1.1 – Architettura del client 38](#_Toc99387958)

[2.1.2 – Architettura del server 40](#_Toc99387959)

[2.1.3 – Gerarchia funzionale 43](#_Toc99387960)

[2.2 – System Design 44](#_Toc99387961)

[2.2.1 – Class diagram di design 44](#_Toc99387962)

[Effettua Accesso 44](#_Toc99387963)

[Effettua Accesso (Google) 45](#_Toc99387964)

[Crea Account 46](#_Toc99387965)

[Recupero Account 47](#_Toc99387966)

[Visualizza Home 48](#_Toc99387967)

[Inserisci Itinerario Mappa 49](#_Toc99387968)

[Inserisci Itinerario GPX 50](#_Toc99387969)

[Visualizza Dettagli Itinerario 51](#_Toc99387970)

[Visualizza Itinerari Propri 52](#_Toc99387971)

[Visualizza Profilo 53](#_Toc99387972)

[2.2.2 – Sequence diagram di design 54](#_Toc99387973)

[Crea Account 54](#_Toc99387974)

[Inserisci Itinerario GPX 55](#_Toc99387975)

[III - Documento di Testing 56](#_Toc99387976)

[3.1 - Unit testing di 3 metodi con JUnit 56](#_Toc99387977)

[3.1.1 – Codice JUnit per il metodo validaCompilazioneCampi 56](#_Toc99387978)

[3.1.2 – Codice JUnit per il metodo generaDurataItinerario 63](#_Toc99387979)

[3.1.3 – Codice JUnit per il metodo generaIntervalloTemporaleInserimento 68](#_Toc99387980)

[3.2 – Valutazione dell’usabilità sul campo 72](#_Toc99387981)

# Revisioni documentazione

|  |  |
| --- | --- |
| Data | Descrizione |
| 13/11/2021 | Stesura iniziale, Strutturazione documento |
| 18/11/2021 | Presentazione dell’idea progettuale, individuazione target utenti, elenco requisiti, use case diagram |
| 21/11/2021 | Descrizione testuale di due use case significativi, mockup, gerarchia funzionale, tabella delle funzionalità |
| 28/11/2021 | Prototipazione funzionale, Valutazione dell’usabilità a priori |
| 10/12/2021 | Modelli di dominio, sequence diagram di analisi e activity diagram |
| 21/12/2021 | Architettura del sistema, client e server |
| 20/01/2022 | Object design |
| 12/03/2022 | Unit testing |
| 24/03/2022 | Valutazione usabilità sul campo |

# I - Documento dei Requisiti

## - Modello funzionale

### 1.1.0 - Glossario

Il seguente è un elenco di termini spesso utilizzati nella sezione corrente della documentazione.

|  |  |
| --- | --- |
| Termine | Descrizione |
| Utente non autenticato | Un utente dell’applicativo che non ha ancora creato un account, oppure non ha effettuato l’autenticazione. |
| Utente autenticato | Un utente che ha effettuato l’autenticazione mediante le credenziali di email e password, oppure utilizzando un account di una piattaforma esterna, come Google. |
| Itinerario | Un tracciato geografico dotato di titolo, durata, un livello di difficoltà, una descrizione (opzionale), e un percorso(opzionale) |
| Percorso | Un percorso/sentiero dotato di un punto d’inizio, un punto di fine e, opzionalmente, da punti di passaggio o tappe. Tali punti vengono individuati dalle coordinate geografiche più comuni: longitudine e latitudine. |
| Livello di difficoltà | E’ una metrica per rappresentare quanto sia difficile da percorrere un itinerario. |
| File GPX | Uno schema XML progettato per il trasferimento di dati GPS tra applicazioni software. |

### 1.1.1 – Presentazione dell’idea progettuale

Si vuole realizzare un sistema informativo, denominato “NaTour21” e finalizzato ad offrire un moderno social network per appassionati di escursioni.

L’utente finale potrà interagire con il sistema mediante un dispositivo mobile il quale, grazie alla presenza di un sensore GPS, risulta essere la tipologia di dispositivo ideale per un’applicativo di natura geografica.

Una volta autenticato, l’utente potrà inserire nuove escursioni sulla piattaforma, dotandole di varie informazioni volte a descrivere le caratteristiche dell’itinerario, come il nome, la durata, il livello di difficoltà, i punti di inizio e fine, e altri. L’itinerario dovrebbe essere rappresentato anche tramite una mappa geografica interattiva.

Gli itinerari possono essere inseriti dall’utente in vari modi, ad esempio interagendo con una mappa, oppure tramite file in formato standard GPX importati dal loro dispositivo.

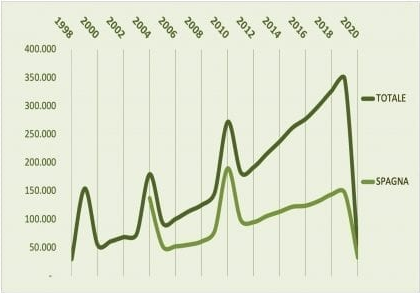
Un itinerario dovrebbe essere visualizzabile dall’utente in una schermata di dettaglio, all’interno della quale si trovano tutti i dettagli sull’itinerario. L’utente dovrebbe inoltre poter visualizzare il percorso dell’itinerario su una mappa, così come eventuali recensioni e fotografie inserite da altri utenti.

L’applicativo va progettato in modo tale da facilitarne la manutenibilità e da agevolare la futura espansione mediante l’integrazione di nuove funzionalità.

### 1.1.2 – Individuazione del target di utenti

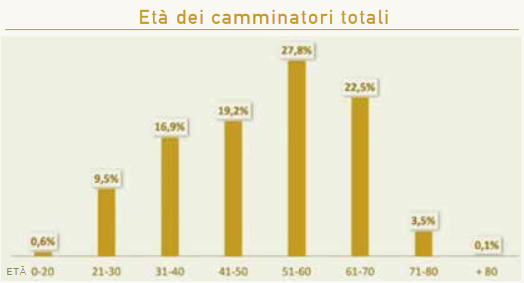
L’utente finale dell’applicativo NaTour21 è un individuo appassionato di attività sportive come l’escursionismo, che è interessato a condividere le proprie esperienze con gli altri. E’ fondamentale quindi offrire a questa categoria di persone un insieme di funzionalità e servizi che possono rendere facile ed immediata la condivisione di itinerari su una piattaforma dedicata. Per individuare questo target di utenti è stata svolta una ricerca bassandosi su dati e statistiche raccolti dai più grandi leader nell’editoria sportiva legata al trekking.

Sorgenti: OutdoorMagazine, Terre di Mezzo

Il grafico a fianco riporta il risultato di una statistica sul numero di persone che si sono recate a percorrere gli itinerari più famosi in Italia. Si può notare come il Covid-19 ha causato un abbassamento drastico del numero di escursionisti.

Tuttavia, con il potenziale approccio della fine della pandemia, si ipotizza che molte persone si dedicheranno di nuovo all’attività di escursionismo, il che risulterebbe in un target di potenziali utenti più ampio.

Successivamente, è stata fatta una raccolta di dati legati all’età dei camminatori. Dal grafico sottostante si può notare che la maggioranza dei camminatori rientra nella fascia d’età tra i 41 e 70 anni. Ciò può dare spunto alla progettazione di un’interfaccia grafica estremamente intuitiva da utilizzare, che si dirige più verso il minimalismo e la semplicità.



### 1.1.3 - Requisiti funzionali

Di seguito sono elencati i requisiti funzionali, ovvero le funzionalità che il sistema deve offrire:

|  |  |
| --- | --- |
| Nome | Crea account |
| Descrizione | Un utente non autenticato deve poter creare un nuovo account con il quale poter accedere all’applicativo. |

|  |  |
| --- | --- |
| Nome | Effettua accesso |
| Descrizione | Un utente non autenticato deve poter effettuare l’accesso, inderendo le proprie credenziali di accesso, oppure tramite un account appartenente ad una piattaforma esterna. |

|  |  |
| --- | --- |
| Nome | Recupera account |
| Descrizione | Il sistema deve consentire ad un utente non autenticato di recuperare il proprio account inserendo la mail collegata all’account e confermando di esserne il proprietario attraverso un codice di verifica. Il recupero consiste nel permettere all’utente di modificare la sua password di accesso. |

|  |  |
| --- | --- |
| Nome | Visualizza home |
| Descrizione | Un utente autenticato deve poter visitare la schermata home dell’applicativo, sulla quale vengono riportati gli itinerari più recenti inseriti da tutti gli utenti. |

|  |  |
| --- | --- |
| Nome | Inserisci itinerario mappa |
| Descrizione | Un utente autenticato deve poter inserire nuovi itinerari tramite una mappa interattiva. |

|  |  |
| --- | --- |
| Nome | Inserisci itinerario GPX |
| Descrizione | Un utente autenticato deve poter inserire nuovi itinerari tramite l’importazione di un file in formato standard GPX. |

|  |  |
| --- | --- |
| Nome | Visualizza dettagli itinerario |
| Descrizione | Un utente autenticato deve poter visualizzare i dettagli di un itinerario, incluse le eventuali recensioni degli utenti e fotografie caricate. |

|  |  |
| --- | --- |
| Nome | Visualizza itinerari propri |
| Descrizione | Un utente autenticato deve poter visualizzare un elenco degli itinerari da esso creati. |

|  |  |
| --- | --- |
| Nome | Rimuovi itinerario |
| Descrizione | Un utente autenticato deve poter rimuovere un itinerario da egli inserito. |

|  |  |
| --- | --- |
| Nome | Visualizza profilo |
| Descrizione | Il sistema deve consentire ad un utente autenticato di visualizzare le informazioni sul suo profilo. |

### 1.1.4 - Requisiti non funzionali

Di seguito sono elencati i requisiti non funzionali, ovvero i vincoli sulle varie qualità del software:

|  |  |
| --- | --- |
| Nome | Performance delle ricerche |
| Descrizione | Le ricerche effettuate dall’utente devono fornire risultati, almeno parziali, entro 3 secondi |

|  |  |
| --- | --- |
| Nome | Apprendibilità |
| Descrizione | L’utente deve essere in grado di utilizzare il 90% delle funzionalità offerte entro un massimo di 2 ore di utilizzo |

|  |  |
| --- | --- |
| Nome | Affidabilità |
| Descrizione | Il sistema è in grado di fornire la funzione richiesta dall’utente il 99.99% delle volte che essa è richiesta |

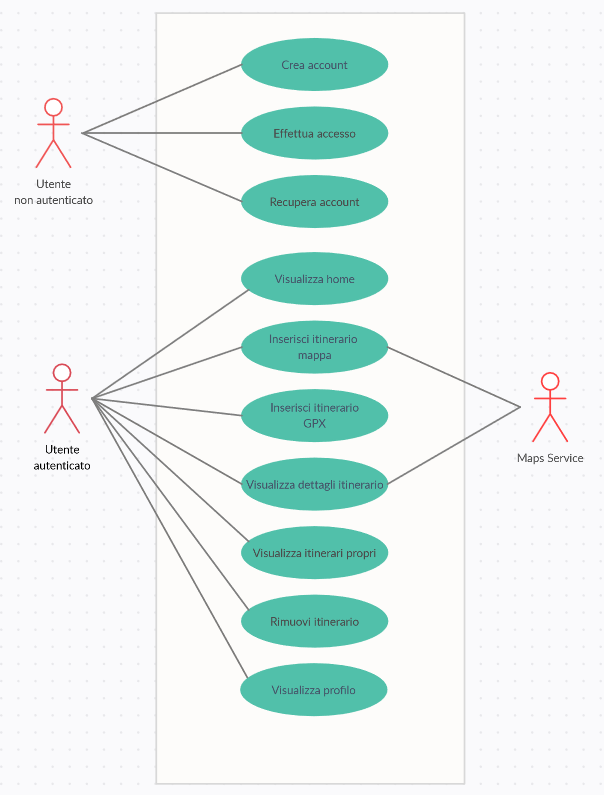
|  |  |
| --- | --- |
| Nome | Manutenibilità |
| Descrizione | Il sistema deve essere realizzato in modo da poter essere facilemente mantenuto. Eventuali modifiche o evolutive dovrebbero essere apportate in maniera facile e in tempi efficienti. |

|  |  |
| --- | --- |
| Nome | Implementazione |
| Descrizione | Il sistema deve consistere di un lato back end realizzato in un linguaggio object-oriented a scelta e un lato client realizzato sulla piattaforma android |

### 1.1.5 – Use Case Diagram

Di seguito viene riportato il diagramma dei casi d’uso, che riporta in modo semplice e schematico gli attori del sistema e le funzionalità associate a questi. Come si può vedere, la principale distinzione è quella tra utenti autenticati e non autenticati.

Inoltre è presente anche l’attore esterno Maps Service, che permetterà di usufruire di vari servizi legati alla geolocalizzazione e alla visualizzazione dei tracciati su una mappa interattiva.



### 1.1.6 - Descrizione testuale degli use case

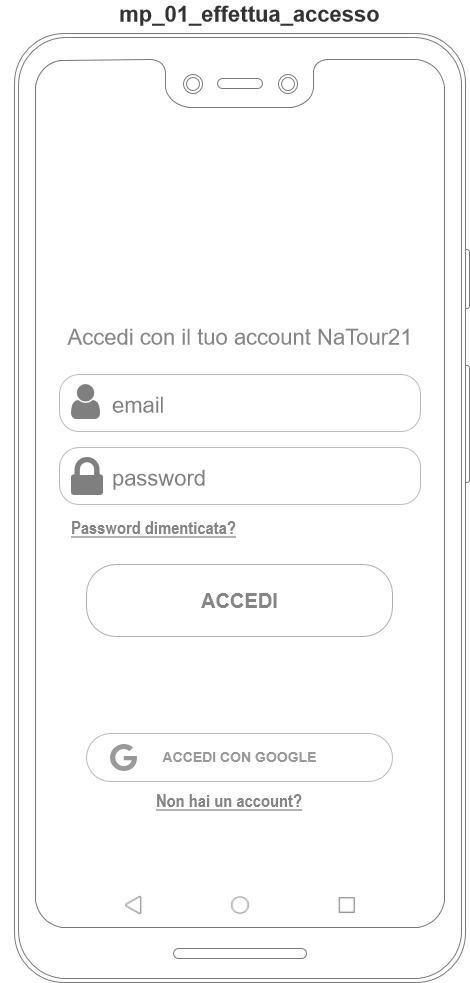
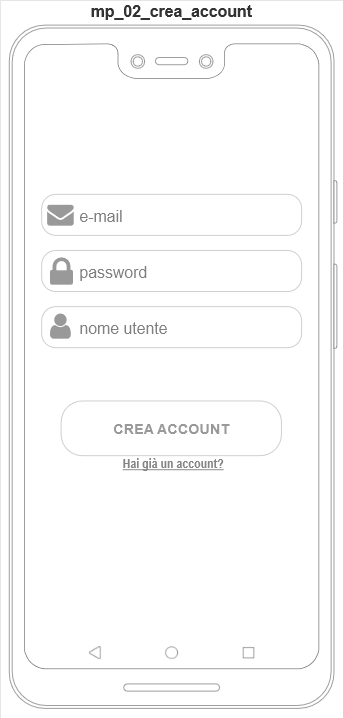
Per la descrizione testuale degli use case è stato utilizzato il template di Alistair Cockburn che permette di offrire, passo per passo, una descrizione del flusso di eventi negli use case.

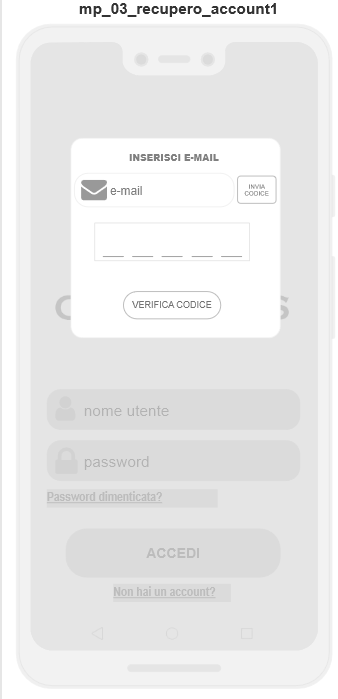
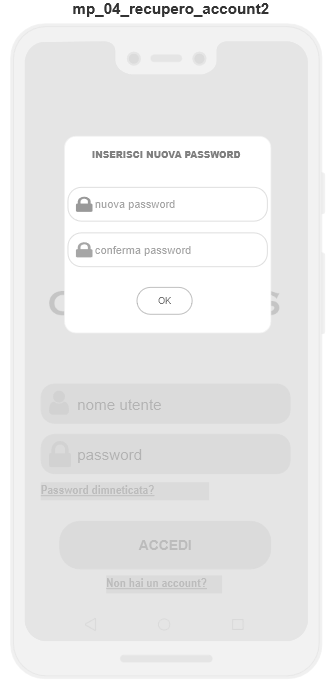
Di seguito si trovano le tabelle per due use case più significativi. I mockup ai quali viene fatto riferimento all’interno delle tabelle mediante il formato ***mp\_xx\_schermata*** si trovano nella sezione [1.1.8 – Prototipazione visuale](#_1.1.8_–_Prototipazione).

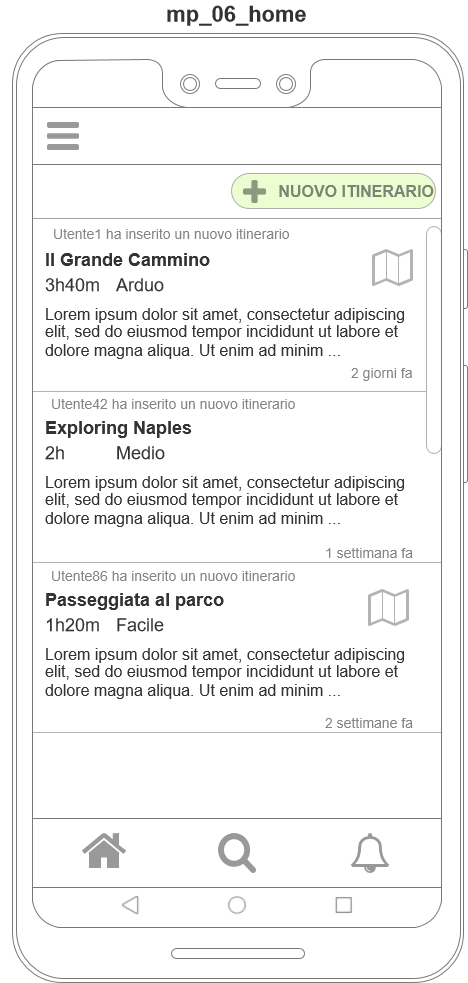
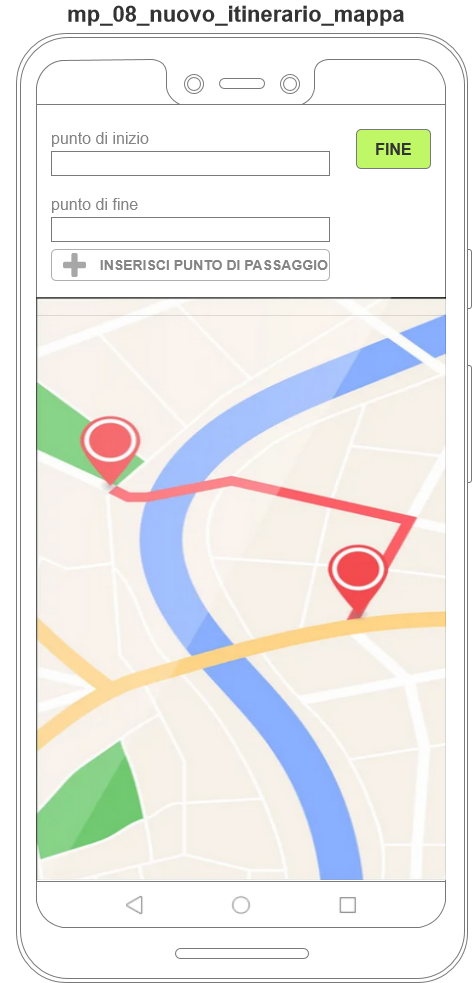
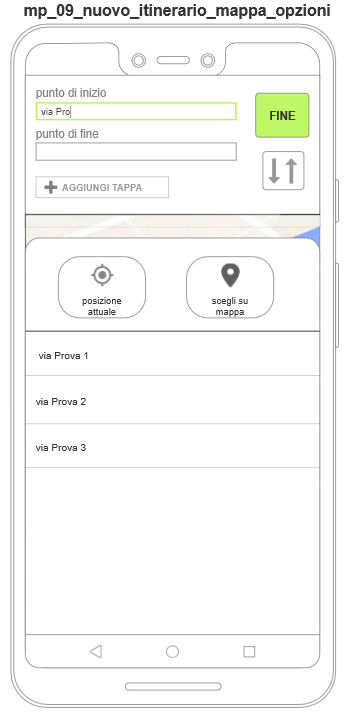
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE** | ***Crea account*** | | |
| Goal in Context | *L’utente non autenticato vuole creare un account* | | |
| Preconditions | *L’utente non autenticato è fornito di mail non collegata ad alcun account* | | |
| Success End Condition | *L’utente non autenticato è adesso in possesso di un account* | | |
| Failed End Condition | *L’utente non autenticato non è in possesso di alcun account* | | |
| Primary Actor | ***Utente non autenticato*** | | |
| Trigger | *Pressione label* ***Non hai un account?*** di ***mp\_01\_effettua\_accesso*** | | |
| MAIN SCENARIO:  Tutto si svolge correttamente, l’operazione va a buon fine | **Step n°** | **Attore: *Utente non autenticato*** | **Sistema** |
| *1* |  | Mostra ***mp\_02\_crea\_account*** |
| *2* | Compila i campi di testo obbligatori email, password, nome utente |  |
| 3 | Preme il pulsante ***CREA ACCOUNT*** di ***mp\_02\_crea\_account*** |  |
| 4 |  | Controlla che i campi di testo obbligatori siano stati compilati correttamente, con esito positivo |
| 5 |  | Controlla che non esista già account associato a quella mail, con esito positivo |
| 6 |  | Crea la nuova utenza |
| 7 |  | Avvisa l’utente dell’avvenuto successo mostrando la dialog ***dl\_01\_account\_creato*** |
| EXTENSION\_1:  L’utente non compila tutti i campi obbligatori | **Step n°** | **Attore: *Utente non autenticato*** | **Sistema** |
| 4.1 |  | Controlla che i campi di testo obbligatori siano stati compilati correttamente, con esito negativo. |
| 5.1 |  | Avvisa l’utente e richiede la compilazione dei campi. |
| 6.1 | Inserisce le informazioni mancanti |  |
| 7.1 | Preme il pulsante ***CREA ACCOUNT*** di ***mp\_02\_crea\_account*** |  |
| 8.1 |  | Torna allo step 4 del MAIN\_SCENARIO |
| EXTENSION\_2:  Esiste già un account corrispondente alla mail inserita dall’utente | **Step n°** | **Attore: *Utente non autenticato*** | **Sistema** |
| 5.2 |  | Controlla che non esista già account associato a quella mail, con esito negativo. |
|  |  | Avvisa l’utente e richiede l’inserimento di una mail diversa |
| 6.2 | Inserisce una nuova mail |  |
| 7.2 | Preme il pulsante ***ISCRIVITI*** di ***mp\_02\_crea\_account*** |  |
| 8.2 |  | Torna allo step 4 del MAIN\_SCENARIO |
| EXTENSION\_3:  L’utente si rende conto che possiede già un account | **Step n°** | **Attore: *Utente non autenticato*** | **Sistema** |
| Fino a step 3 | Preme il pulsante ***hai già un account?*** di ***mp\_02\_crea\_account*** |  |
| Step successivo |  | Mostra ***mp\_01\_effettua\_accesso.*** Parte lo use case ***Effettua accesso*** |

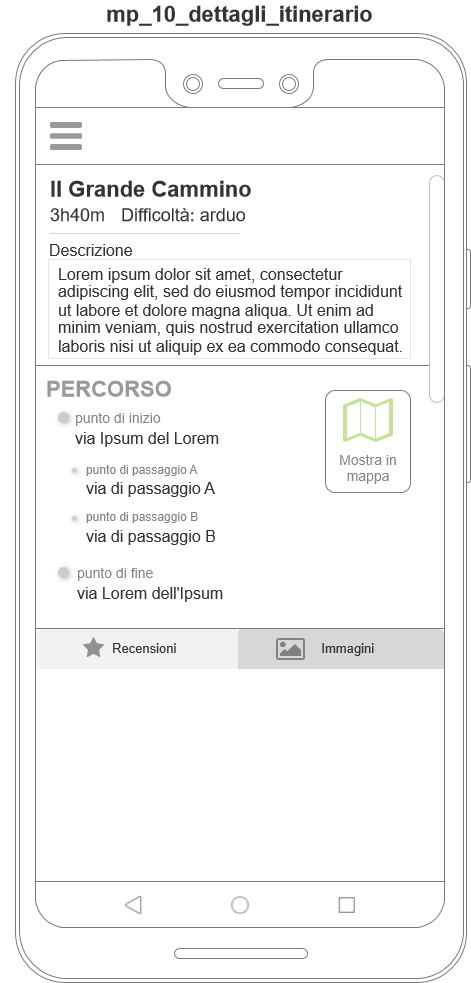
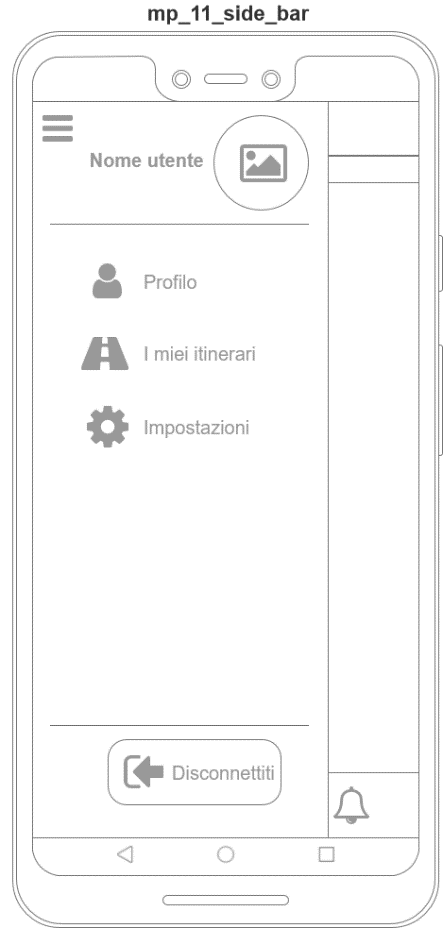
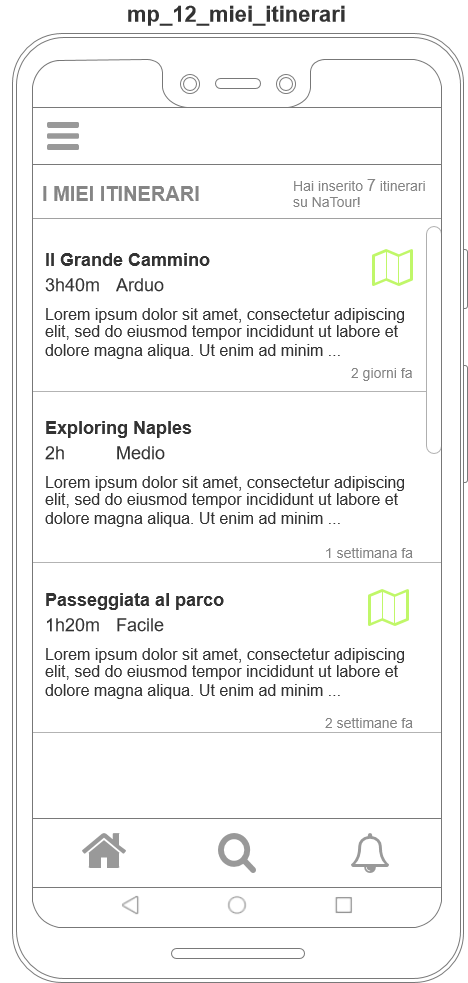
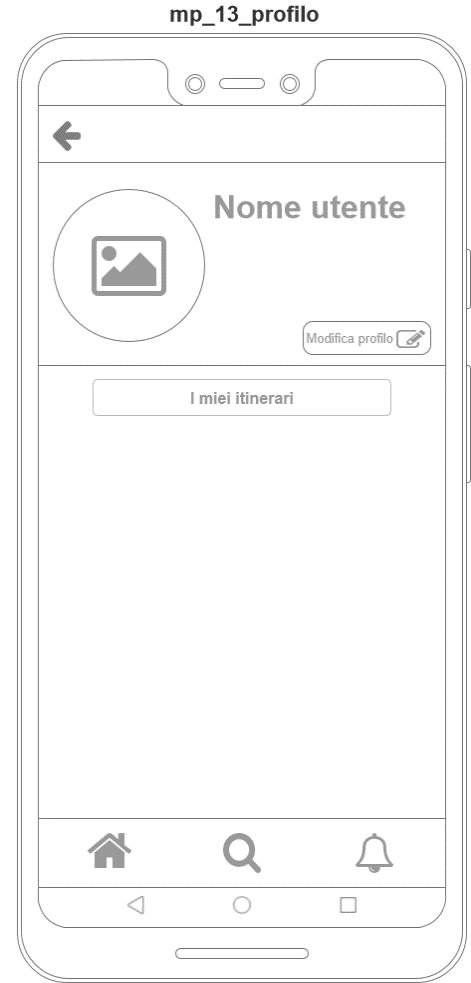
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE** | ***Inserisci itinerario mappa*** | | | |
| Goal in Context | *L’utente vuole inserire un nuovo itinerario* | | | |
| Preconditions | *L’utente è autenticato* | | | |
| Success End Condition | *L’utente inserisce un nuovo itinerario nel sistema* | | | |
| Failed End Condition | *L’utente non ha inserito un nuovo itinerario* | | | |
| Primary Actor | *Utente autenticato* | | | |
| Trigger | *Pressione del pulsante* ***Nuovo itinerario*** *in* ***mp\_06\_home*** | | | |
| MAIN SCENARIO:  Tutto si svolge correttamente e l’utente inserisce un nuovo itinerario con un percorso | **Step n°** | **Attore**  **Utente autenticato** | **Sistema** | **Servizio Mappe** |
| 1 |  | Mostra  ***mp\_07\_nuovo\_itinerario\_dati*** |  |
| 2 | Inserisce il nome, la durata, la difficoltà e, opzionalmente, la descrizione dell’itinerario |  |  |
| 3 | Preme il tasto **Apri mappa** |  |  |
| 4 |  | Mostra ***mp\_08\_nuovo\_itinerario\_mappa*** |  |
| 5 | Inserisce il punto d’inizio e di fine dell’itinerario come indirizzi fisici oppure come punti sulla mappa |  |  |
| 6 |  | Interroga **Servizio Mappe** per ottenere le coordinate geografiche degli indirizzi / punti inseriti |  |
| 7 |  |  | Risponde restituendo le coordinate geografiche richieste |
| 8 |  | Mostra gli indirizzi inseriti dell’utente sulla mappa e disegna un tracciato che li colleghi |  |
| 9 | Preme il tasto **Fine** |  |  |
| 10 |  | Mostra ***mp\_07\_nuovo\_itinerario\_dati***  compilando le informazioni sul percorso con i dati inseriti dall’utente |  |
| 11 | Preme il tasto **Salva** |  |  |
| 12 |  | Salva l’itinerario con percorso |  |
|  |  |  |  |  |
| EXTENSION\_1: Tutto si svolge correttamente e l’utente inserisce un nuovo itinerario senza percorso | **Step n°** | **Attore**  **Utente autenticato** | **Sistema** | **Servizio Mappe** |
| 3.1 | Preme il tasto **Salva** |  |  |
| 4.1 |  | Mostra la dialog ***dl\_02\_avviso\_percorso*** |  |
| 5.1 | Preme il tasto **SI** |  |  |
| 6.1 |  | Salva l’itinerario senza percorso |  |
| EXTENSION\_2:  L’utente inserisce le posizioni geografiche interagendo con la mappa | **Step n°** | **Attore**  **Utente autenticato** | **Sistema** | **Servizio Mappe** |
| 5.2 | Preme su un punto della mappa |  |  |
| 6.2 |  | Interroga il **Servizio Mappe** per ottenere l’indirizzo corrispondente alle coordinate della posizione geografica |  |
| 7.2 |  |  | Risponde restituendo gli indirizzi richiesti |
| 8.2 |  | Riempe i campi punto di inizio e punto di fine con gli indirizzi ricevuti, torna nel MAIN\_SCENARIO |  |
| EXTENSION\_3: L’utente compila parzialmente i campi obbligatori nome, durata e difficoltà in ***mp\_07\_nuovo\_itinerario*** | **Step n°** | **Attore**  **Utente autenticato** | **Sistema** | **Servizio Mappe** |
| 2.3 | Compila solo alcuni tra i campi obbligatori nome, durata e difficoltà |  |  |
| 3.3 – 11.3 | Tutto si svolge come nel MAIN\_SCENARIO | | |
| 12.3 |  | Avvisa l’utente che i campi nome, durata e difficoltà vanno compilati |  |
| 13.3 | Inserisce le informazioni mancanti |  |  |
| 14.3 |  | Salva l’itinerario |  |
| EXTENSION\_4: L’utente inserisce solo un punto di inizio o un punto di fine del percorso | **Step n°** | **Attore**  **Utente autenticato** | **Sistema** | **Servizio Mappe** |
| 5.4 | Compila solamente uno tra i campi **punto di inizio** e **punto di fine** | Di pari passo con l’inserimento degli indirizzi, il sistema interroga **Servizio Mappe**, mostrando una lista di indirizzi suggeriti |  |
| 6.4 – 9.4 | Tutto si svolge come nel MAIN\_SCENARIO | | |
| 10.4 |  | Avvisa l’utente che il percorso inserito non sembra essere completo |  |

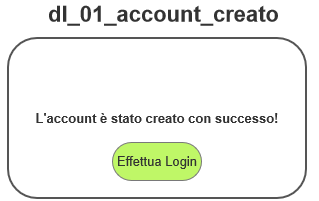
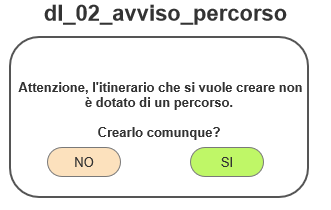
### 1.1.7 – Prototipazione visuale

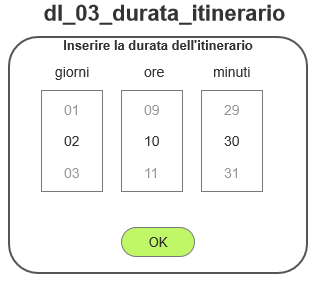












### 1.1.8 – Valutazione dell’usabilità a priori

Per valutare l’usabilità dell’applicativo a priori è stata effettuata una sessione di test di compito, dove sono state generate delle task da effettuare e sono stati affidati ad un numero ristretto di utenti. Gli utenti hanno potuto interagire con un prototipo interattivo, composto dai mockup mostrati nella sezione precedente. Le task che gli utenti hanno dovuto svolgere sono:

* **Task #1**: effettuare la registrazione e accedere con il proprio account
* **Task #2**: visualizzare i dettagli di un itinerario dalla schermata home
* **Task #3**: visualizzare uno degli itinerari che hai inserito precedentemente

Di seguito sono mostrati i risultati dei test, eseguiti su tre utenti:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Task #1 | Task #2 | Task #3 |
| User #1 | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| User #2 | Checkmark with solid fill | Checkmark with solid fill | Close with solid fill |
| User #3 | Checkmark with solid fill | Checkmark with solid fill | Close with solid fill |

Tasso di successo = 7 successi / 9 test totali = 77%

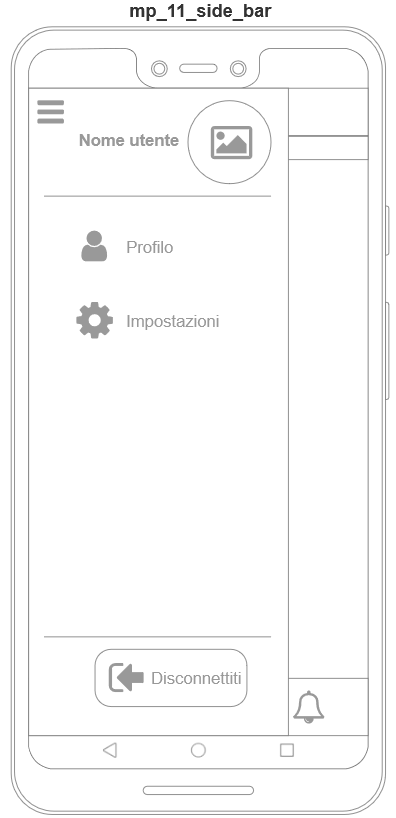
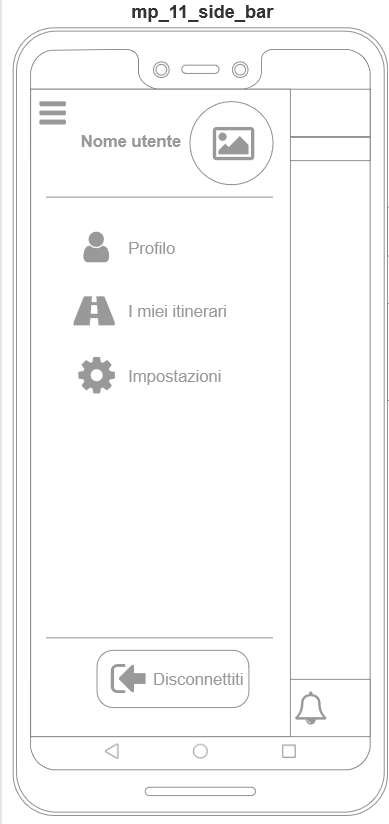
Dall’osservazione degli utenti sono risultati evidenti alcuni problemi, per fortuna non bloccanti, con l’interfaccia grafica dell’applicativo. In particolare, per quanto riguarda la **Task#3**, alcuni utenti hanno avuto difficoltà a trovare la schermata **mp\_12\_miei\_itinerari** dove vengono mostrati gli itinerari inseriti dall’utente

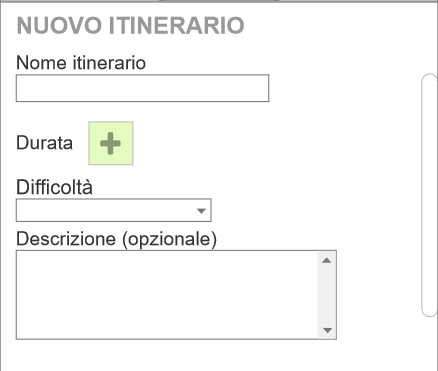
**Soluzione al problema citato:**

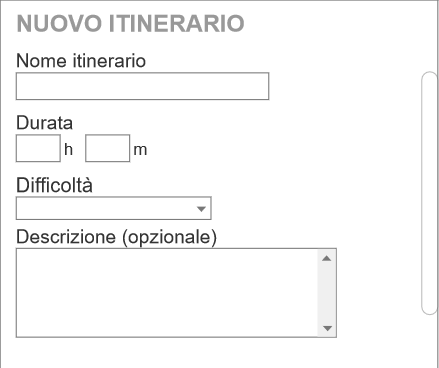
Prima dell’esecuzione dei test di usabilità, il procedimento per arrivare alla schermata **mp\_12\_miei\_itinerari** è stato il seguente:

* 1. dalla schermata **mp\_06\_home** aprire il menu laterale **mp\_11\_side\_bar**
  2. nel menu laterale cliccare sul pulsante profilo
  3. nella schermata **mp\_13\_profilo** cliccare sul pulsante “I miei itinerari”

Il vecchio menu laterale infatti non permetteva l’accesso diretto agli itinerari dell’utente, per cui è stato aggiunto un pulsante che facilita il raggiungimento di quella schermata in maniera più veloce e comoda:

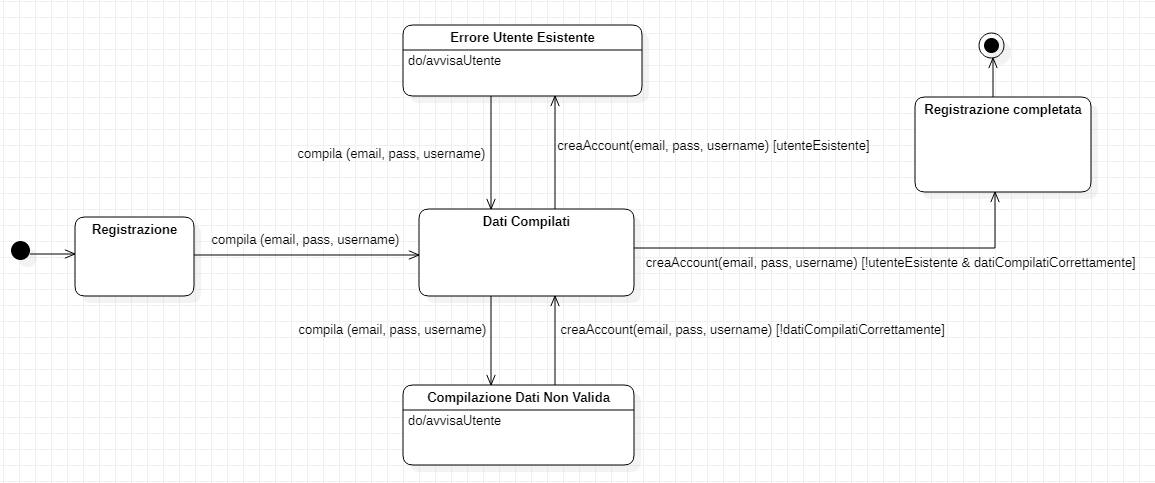


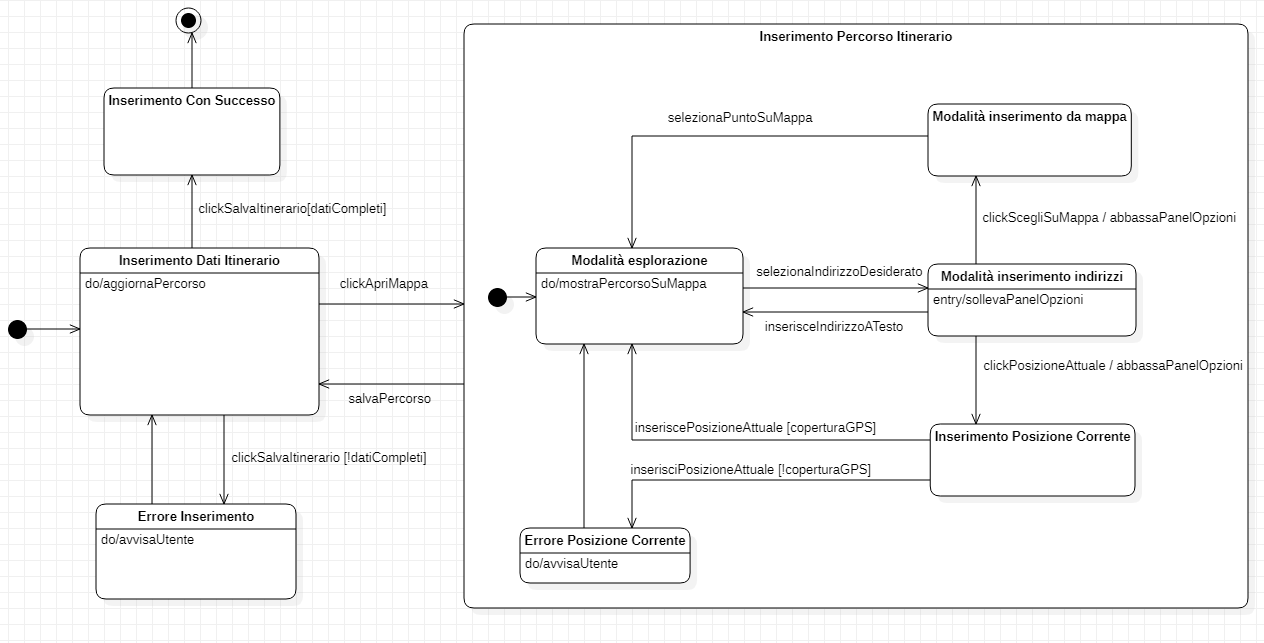
Inoltre, sono state apportate altre modifiche sull’interfaccia grafica, basandosi su alcuni consigli provvenienti dagli utenti. Ad esempio, nella schermata **mp\_07\_nuovo\_itinerario\_dati** è stata modificata la modalità di inserimento della durata dell’itinerario: è stata aggiunta una dialog composta da number picker per un inserimento più comodo e meno prone ad errori, la **dl\_03\_durata\_itinerario**





### – Prototipazione funzionale

**1 – Crea Account**

**2 – Inserisci Itinerario con Mappa**

### 1.1.10 – Tabella delle funzioanlità

Di seguito vengono mostrate le tabelle delle funzionalità dell’applicativo mobile. Le tabelle elencano le funzionalità del sistema, indicando per ognuna di essa il relativo livello di completezza. Per motivi di leggibilità e spazio la tabella verrà suddivisa in sottotabelle.

Sottotabella 1: funzionalità legate all’autenticazione:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Crea account | Effettua accesso | Recupera account |
| Prototipazione visuale (mockup) | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| Descrizione testuale dell’use case | Checkmark with solid fill |  |  |
| Prototipazione funzionale (statechart) | Checkmark with solid fill |  |  |
| Design | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| Implementazione | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| Unit Testing | Checkmark with solid fill |  |  |
| Field testing | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |

Sottotabella 2: funzionalità principali dell’applicativo:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Visualizza home | Inserisci itinerario mappa | Inserisci itinerario GPX | Visualizza dettagli itinerario | Visualizza itinerari propri | Rimuovi itinerario | Visualizza  profilo |
| Prototipazione visuale (mockup) | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| Descrizione testuale dell’use case |  | Checkmark with solid fill |  |  |  |  |  |
| Prototipazione funzionale (statechart) |  | Checkmark with solid fill |  |  |  |  |  |
| Design | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| Implementazione | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| Unit Testing |  | Checkmark with solid fill |  | Checkmark with solid fill |  |  |  |
| Field testing | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |

## - Modelli di dominio

### 1.2.1 – Class diagram di analisi

Di seguito si trova il modello di dominio, ovvero una rappresentazione visuale delle classi relative al dominio. Prima vengono elencati i class diagram di analisi, che individuano gli oggetti in gioco in base all’euristica Three-Object-Type, dove gli oggetti vanno classificati in tre gruppi in base alla loro funzionalità:

* Boundary – modellano l’interfaccia con la quale andranno ad interagire gli utenti
* Control – modellano la business logic
* Entity – modellano i concetti di dominio e l’informazione persistente

Per motivi di leggibilità le classi sono state raggruppate in singole schermate in base allo use case di appartenanza.

#### Autenticazione (effettua accesso + crea account)

#### Recupera account

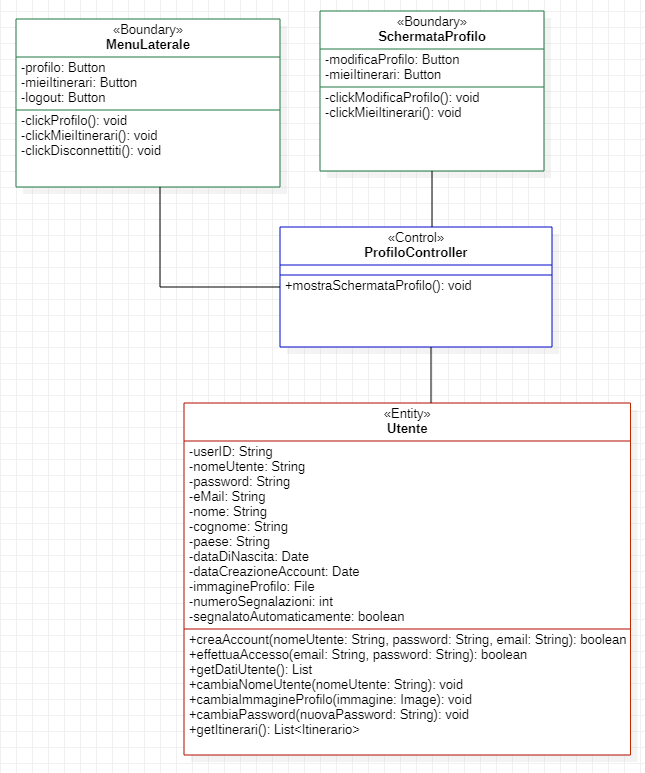
#### Visualizza Home

#### Inserisci itinerario

#### Visualizza Itinerario Propri

#### Visualizza Dettagli Itinerario

#### Visualizza Profilo



### 1.2.2 – Sequence diagram di analisi

Di seguito vengono elencati i diagrammi di sequenza per gli use case di inserimento itinerario: il primo tratta il casi in cui il percorso viene inserito dall’utente utilizzando la mappa interattiva, mentre il secondo nel caso in cui viene importato un file gpx contenente le informazioni geografiche del percorso.

#### Inserisci Itinerario Mappa

#### Inserisci Itinerario GPX

### 1.2.3 – Activity diagram

#### Recupero Account

# II - Documento di Design

## 2.1 - Archittettura del sistema

### 2.1.1 – Architettura del client

Il cliente mobile è stato realizzato su piattaforma Android, con l’ausilio di Android Studio. L’architettura utilizzata per il client mobile è il **MVP (Model-View-Presenter)**. Questo è un tipo di architettura a strati chiusa che promuove la **separation of concerns**, ovvero la distinzione degli oggetti in base al loro ruolo e alle loro responsabilità.

**VIEW**

**PRESENTER**

**MODEL**

In particolare:

* la **View** è composta dall’insieme di elementi di Android che compongono l’interfaccia utente dell’applicativo: Activity, Fragment e Dialog. Sono soggette al ciclo di vita dell’ambiente Android;
* lo strato **Presenter** è un mediatore tra il View e il Model che si occupa della business logic, ed è composto da tutte le classi con suffisso “Presenter”;
* il **Model** è composto dagli oggetti di dominio sui quali si interagisce nell’interfaccia e dalle classi responsabili per l’interfacciamento con il backend e le api esterne

Di seguito vanno elencati alcuni dei componenti e librerie più notevoli utilizzati nella realizzazione dell’applicativo mobile:

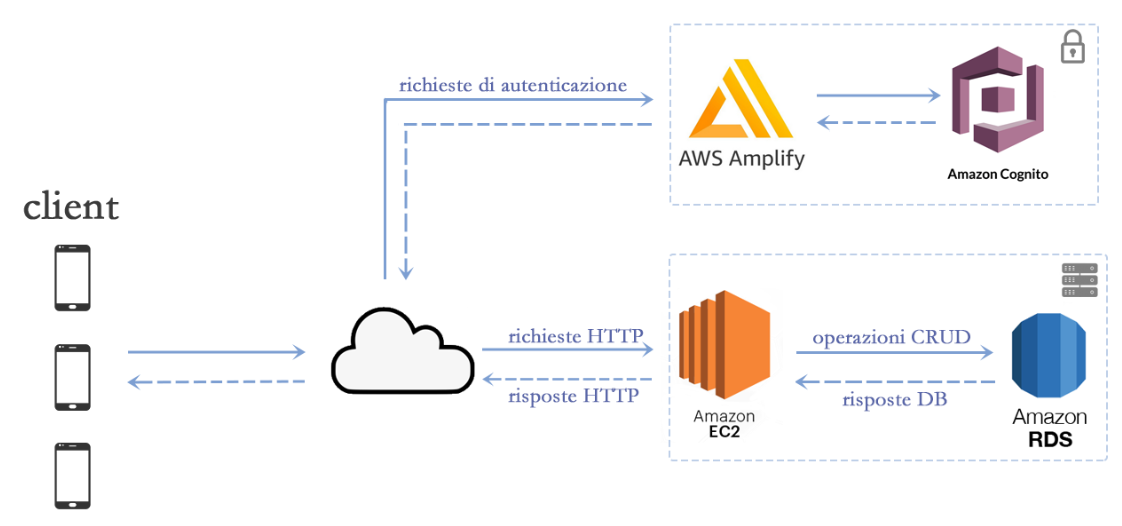
* [**JetpackNavigation**](https://developer.android.com/guide/navigation?gclid=CjwKCAjw07qDBhBxEiwA6pPbHkGbtBUA7IToeRThWwYq9sai6L_0_qWobg6a4kermUhflA-qtfY0sBoC8HQQAvD_BwE&gclsrc=aw.ds): il componente di navigazione si occupa di reagire alle interazioni dell’utente con l’applicativo. Permette di definire un grafo di navigazione che gestisce le transazioni da una schermata all’altra in modo semplice e veloce. In particolare, è stata utilizzata un’architettura Single-Activity, dove ciascuna schermata è rappresentata da un fragment separato.
* [**Volley**](https://developer.android.com/training/volley): una libreria che facilita l’effetuazione di richieste http ad api esterne. Tra le sue funzionalità principali vi è lo scheduling delle richieste http e la possibilità di effettuare richieste in modo concorrente.
* [**osmdroid**](https://github.com/osmdroid/osmdroid): una libreria per l’interazione con mappe geografiche su un dispositivo Android. Contiene una serie di strumenti e utilities per la manipolazione e la rappresentazione dei dati geografici e la creazione di percorsi.
* [**JUnit5**](https://junit.org/junit4/)**:** un framework per realizzare unit test di singoli componenti del sistema.

### 2.1.2 – Architettura del server

L’architettura ad alto livello del sistema è una di tipo **client-server**.

I dispositivi client quindi interagiscono con un back-end, richiedendo dei servizi e risorse mediante una communicazione di tipo **request-response**.

L’architettura del back-end si fonda sui servizi offerti dalla piattaforma **AWS**, che è stata scelta per i numerosi vantaggi che offre:

* **Flessibilità**: AWS offre una vasta gamma di prodotti e servizi di varia natura. Si ha a disposizione un ambiente virtuale dove è possibile caricare il software e i servizi necessari per il funzionamento dell’applicazione.
* **Sicurezza ed affidabilità**: i servizi offerti dagli Amazon Web Services sono altamente sicuri ed affidabili, sia dal punto di vista fisico e operativo, sia a livello di software.
* **Scalabilità e prestazioni elevate**: AWS offre strumenti come Auto Scaling ed Elastic Load Balancing, che permettono di ridimensionare le risorse di calcolo e di storage in base alle esigenze e alle necessità dell’applicazione.

Come si può osservare nel diagramma architetturale, i client possono effettuare richieste verso quelli che sono i due principali gruppi di servizi back-end:

* **servizi di autenticazione**: alla base di questi vi è **Amazon Cognito**, che fornisce strumenti per l’autenticazione e la gestione degli utenti, sia con gli account interni all’applicativo, che mediante account di terze parti, come Google. E’ stata utilizzata la libreria **Amazon Amplify** che facilita l’interfacciamento con Cognito e la configurazione delle user e identity pools;
* **server**: per l’hosting del server è stato scelto il servizio **Amazon EC2**, una piattaforma di calcolo che permette di configurare e lanciare istanze di elaboratori sicure ed affidabili sul cloud. Il principale vantaggio delle istanze EC2 è la possibilità di ridimensionarle in base alle necessità del sistema e al carico che si osserva in un determinato periodo, sia in numero, che come prestazioni dell’elaboratore individuale. I dati invece vanno persistiti all’interno di un database MySQL, che si trova su un’istanza di **Amazon RDS**, un servizio di hosting di database relazionali.

Per la realizzazione del server hostato sulle istanze EC2 è stato utilizzato il framework **Spring**, che presenta numerose features volte alla creazione di back-end in Java. Il framework permette di esportare il programma server come un file jar di dimensioni contenute e di lanciarlo nel giro di pochi secondi sulle istanze EC2.

In particolare, è stato usato il modulo **Spring REST**, che permette di realizzare delle API sicure e facilmente manutenibili, rimuovendo la maggioranza del codice boilerplate necessario a gestire le richieste e risposte HTTP, minimizzando così le probabilità di errori di programmazione.

Il server è stato realizzato con un’architettura chiusa fortemente consigliata dagli sviluppatori di Spring:

**CONTROLLER**

**SERVICE**

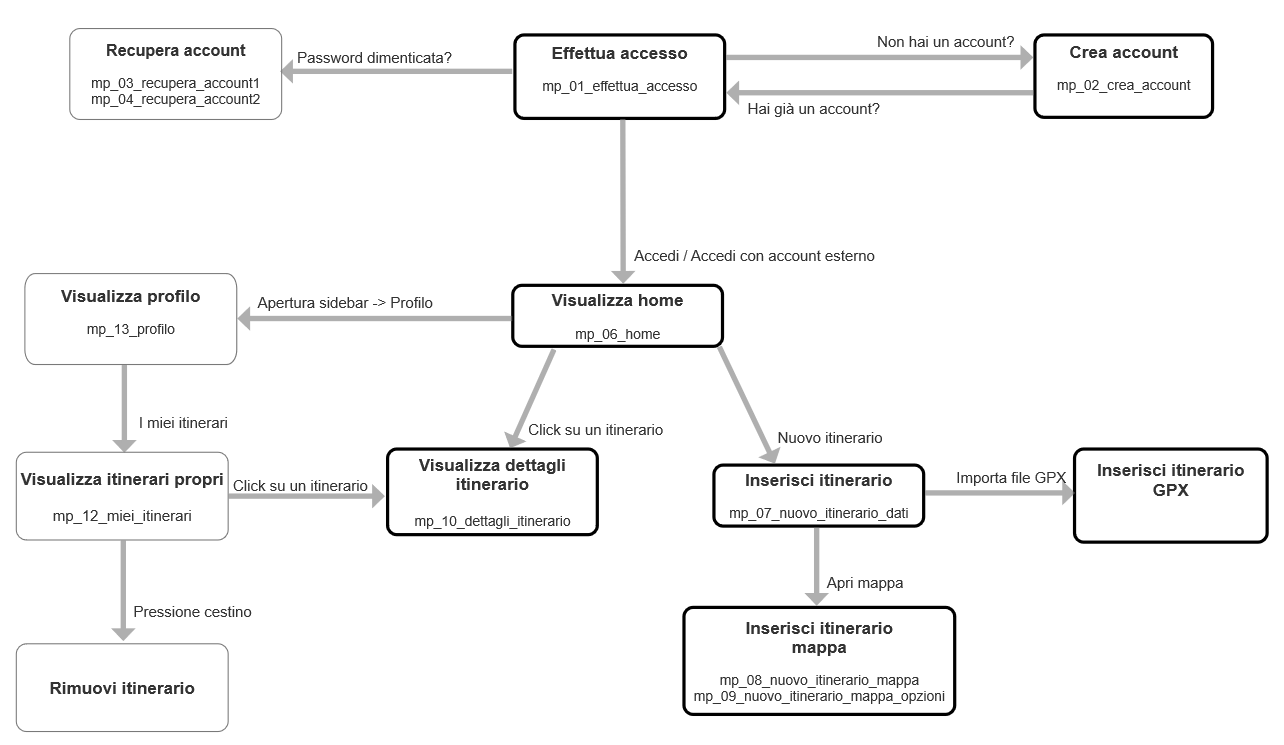
**REPOSITORY**

In particolare:

* lo strato **Controller** è composto da classi che fungono da endpoint: accettano e rispondono alle richieste HTTP del client;
* i **Service** rappresentano la business logic del server ed agiscono in base alle richieste provenienti dal controller;
* lo strato **Repository** è composto da tutti gli oggetti di dominio e dalle classi DAO che comunicano con il database sottostante mediante richieste CRUD

### 2.1.3 – Gerarchia funzionale

Di seguito viene riportata la gerarchia delle funzionalità, che permette di rappresentare la navigazione all’interno dell’applicativo mobile. Ad ogni funzionalità viene associato il mockup ad essa corrispondente. Le frecce rappresentato la transizione da una funzionalità ad un’altra, e sono accompagnate dall’affordance che permette all’utente di interagire con l’applicativo.

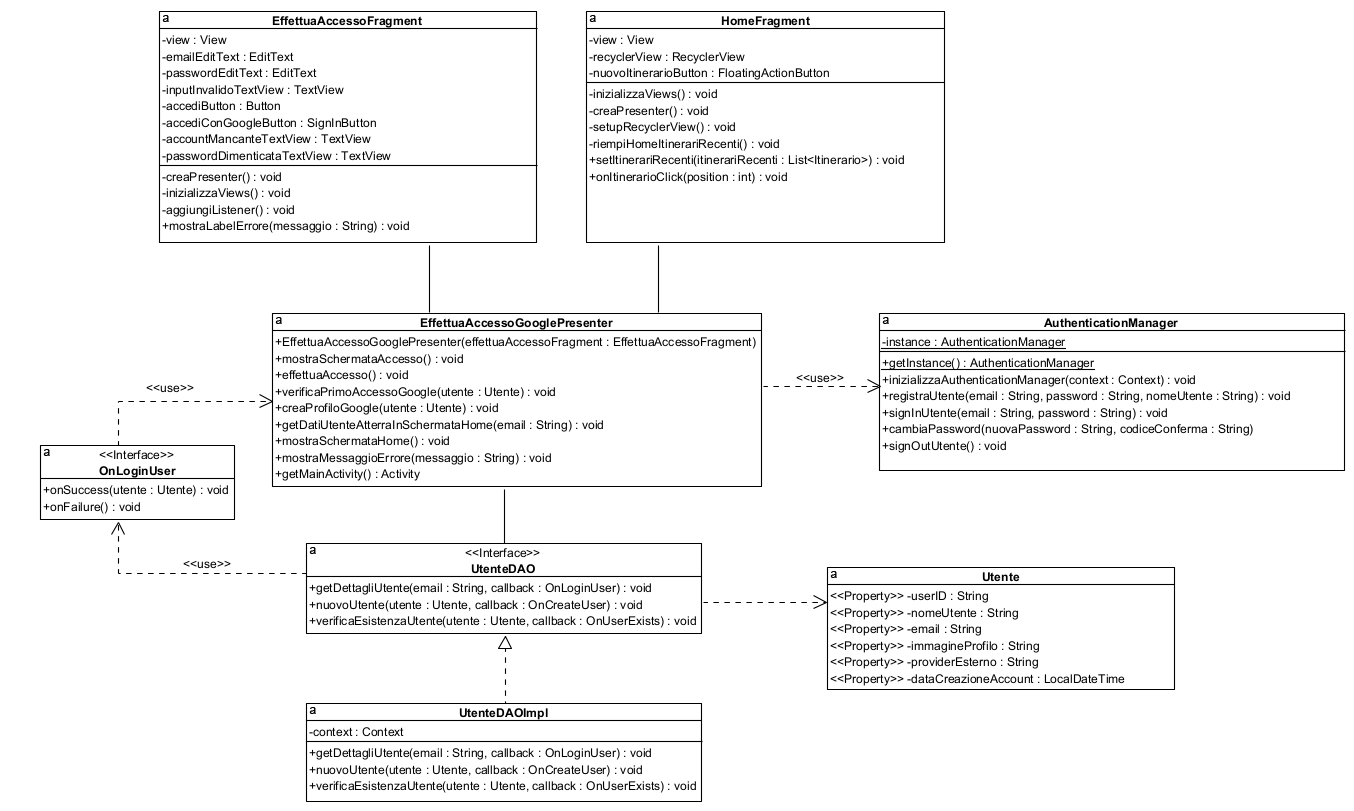
Inoltre, si è effettuata una valutazione sull’importanza delle funzionalità nel contesto dell’applicazione. Prendendo in considerazione le necessità del cliente, ad alcune delle funzionalità è stata data una priorità più alta. Tali funzionalità sono individuabili nel diagramma dal loro bordo più spesso.

## 2.2 – System Design

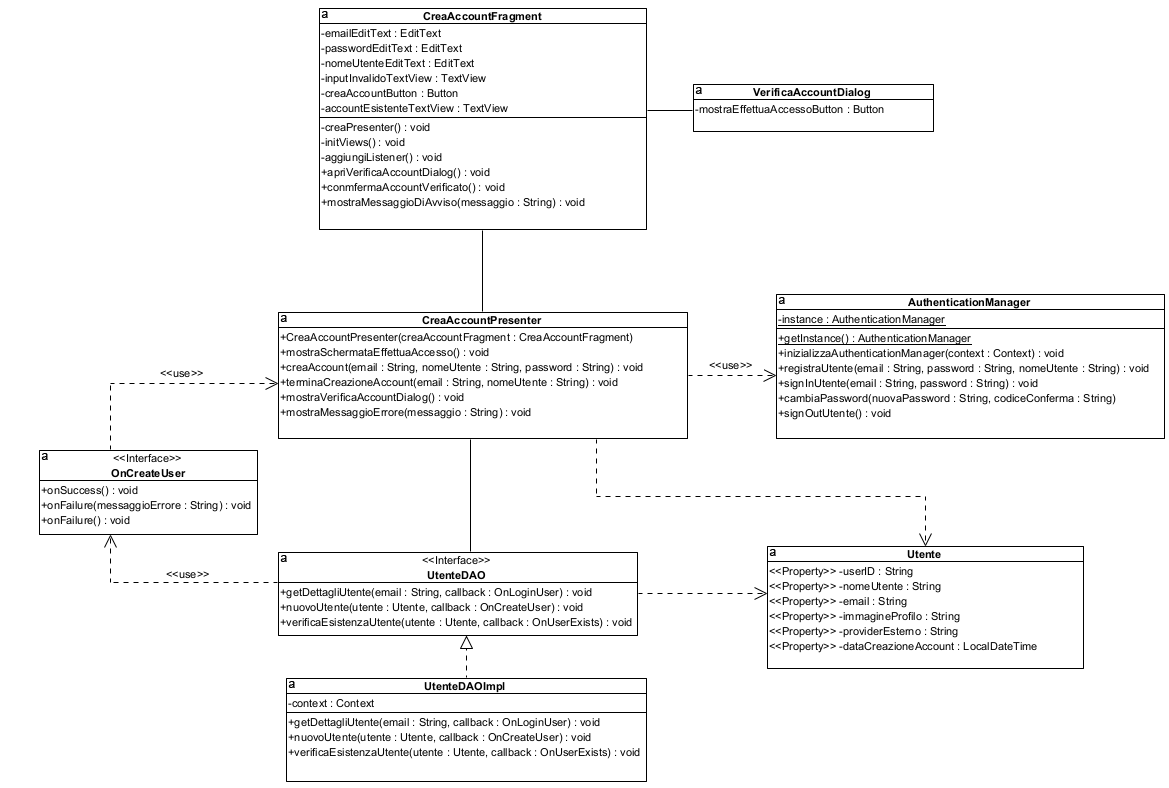
### 2.2.1 – Class diagram di design

#### Effettua Accesso

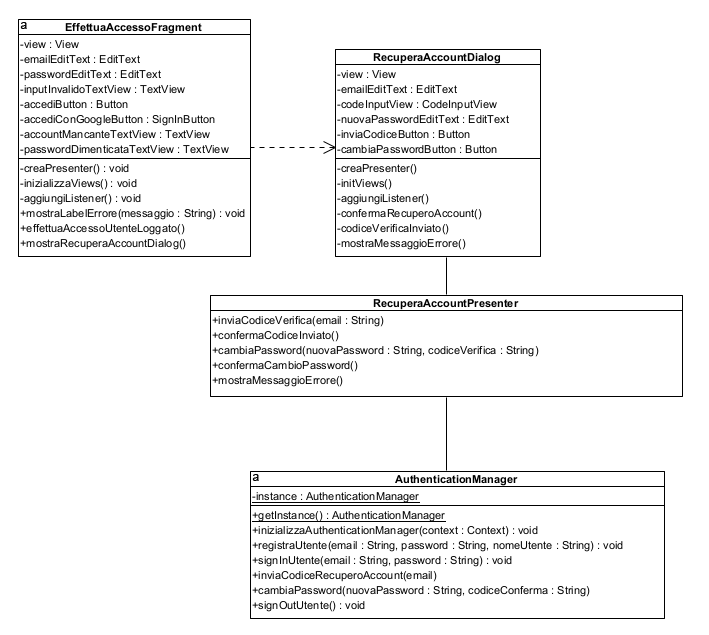
#### Effettua Accesso (Google)



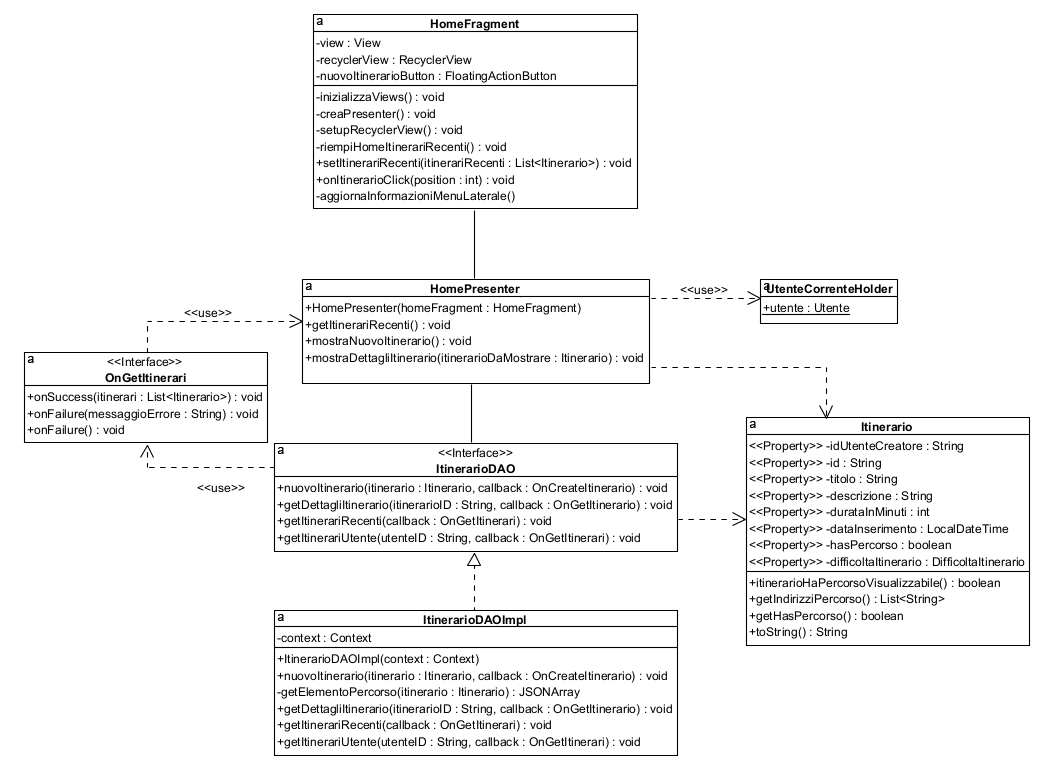
#### Crea Account



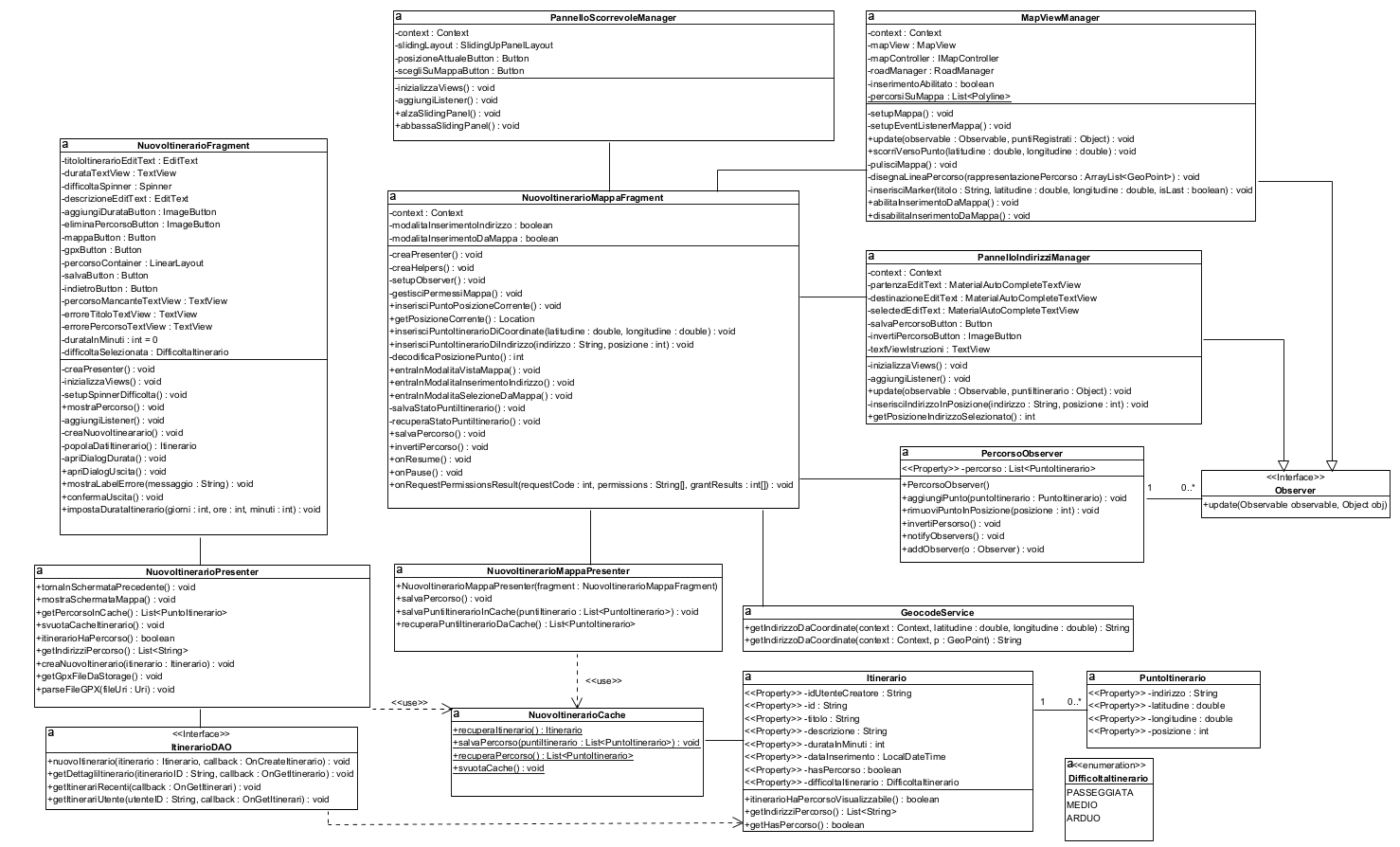
#### Recupero Account



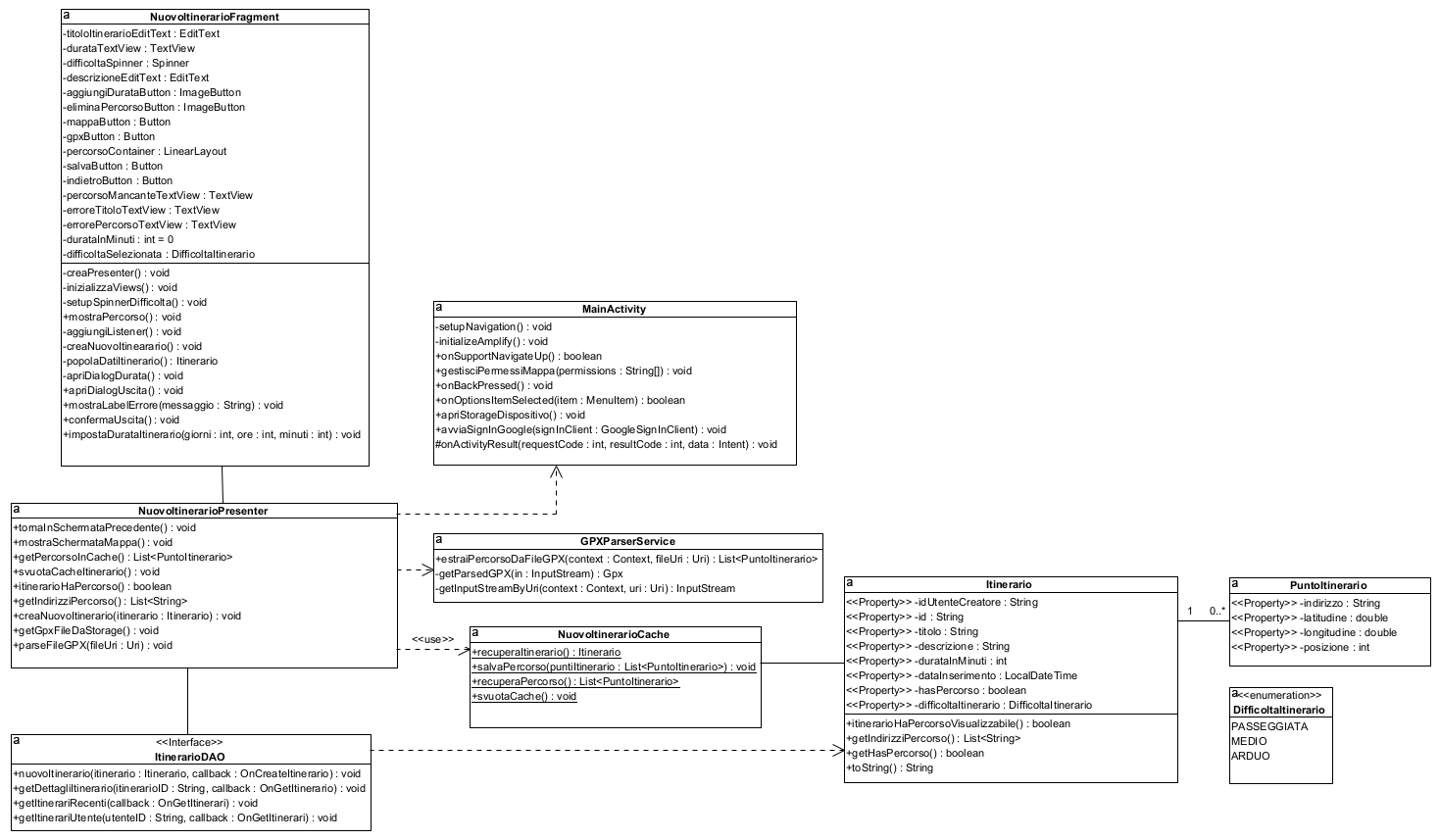
#### Visualizza Home



#### Inserisci Itinerario Mappa



#### Inserisci Itinerario GPX

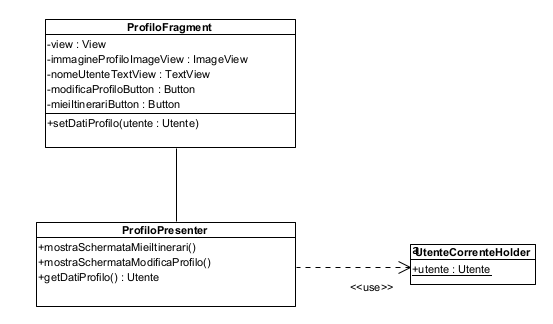


#### Visualizza Dettagli Itinerario

#### Visualizza Itinerari Propri



#### Visualizza Profilo



### 2.2.2 – Sequence diagram di design

#### Crea Account

#### Inserisci Itinerario GPX

# III - Documento di Testing

## 3.1 - Unit testing di 3 metodi con JUnit

### 3.1.1 – Codice JUnit per il metodo validaCompilazioneCampi

Riferimento al metodo, applicativo mobile: **it.ingsw.natour21.control.validators.SignUpInputValidator#validaCompilazioneCampi**

Il metodo **validaCompilazioneCampi** ha il compito di validare la compilazione corretta dei campi di testo durante il processo di registrazione dell’utente. Il metodo accetta come parametri le stringhe da validare, effettua i controlli necessari con il supporto della classe ValidationRules, e restituisce un oggetto di tipo ValidatorResponse avente i seguenti attributi:

* il booleano inputValido, valorizzato a true se i campi sono stati correttamente compilati, false altrimenti
* messaggioErrore: quando inputValido è false, descrive il motivo per cui la validazione è fallita

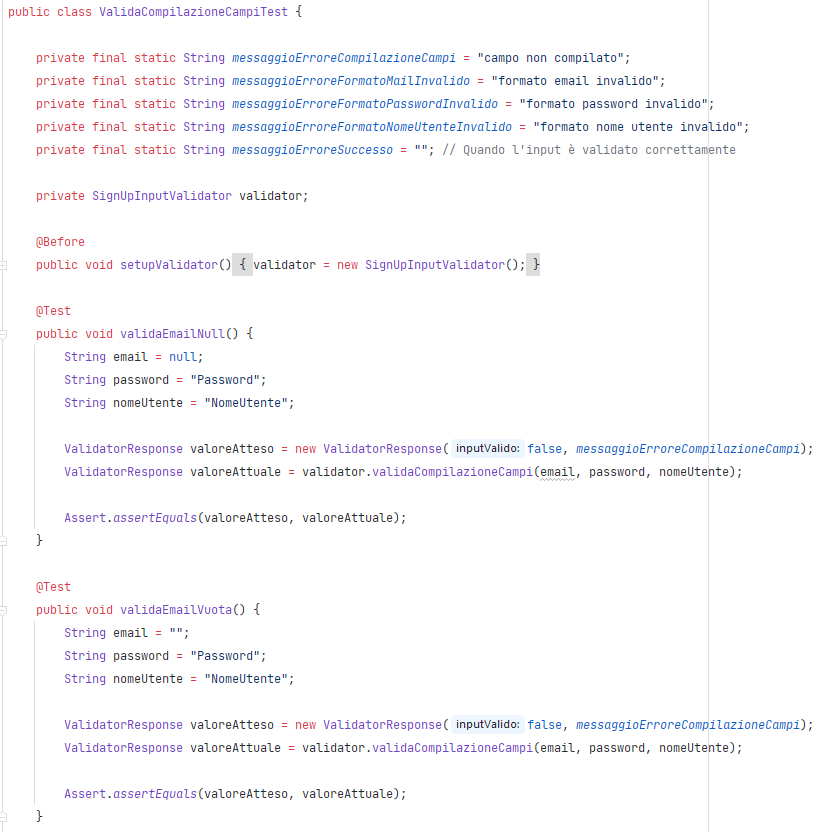
Inoltre, viene di seguito mostrato il contenuto della **classe di supporto ValidationRules**. I metodi di questa classe vanno utilizzati dal metodo da testare e sono parte integrante della logica di validazione, ma sono stati estratti in metodi separati per motivi di leggibilità e pulizia del codice:

**Strategia Black-Box:**

Si effettua l’Input Space Partitioning, ovvero la suddivisione dei parametri del metodo in classi di equivalenza. Poichè le verifiche effettuate su ciascun parametro sono indipendenti da quelle fatte sui parametri rimenenti, si assume che per ogni parametro testato gli altri due siano validi:

|  |  |  |
| --- | --- | --- |
| **Parametro** | **Valore** | **Risultato atteso** |
| email | null | inputValido = false  messaggioErrore = campo non compilato |
| email | stringa vuota: “” | inputValido = false  messaggioErrore = campo non compilato |
| email | stringa contenente solo il carattere “@” | inputValido = false  messaggioErrore = formato mail invalido |
| email | stringa contenente solo il carattere . | inputValido = false  messaggioErrore = formato mail invalido |
| email | stringa contentente il carattere “@” preceduto da “.” | inputValido = false  messaggioErrore = formato mail invalido |
| email | stringa contenente il carattere “@” seguito immediatamente da “.” | inputValido = false  messaggioErrore = formato mail invalido |
| email | stringa contentete il carattere “@” seguito da “.” dopo più di un carattere | inputValido = true  messaggioErrore = “” |
| password | null | inputValido = false  messaggioErrore = campo non compilato |
| password | stringa vuota: “” | inputValido = false  messaggioErrore = campo non compilato |
| password | stringa con lunghezza minore di quella consentita | inputValido = false  messaggioErrore = formato password invalido |
| password | stringa con lunghezza maggiore di quella consentita | inputValido = false  messaggioErrore = formato password invalido |
| password | stringa con lunghezza consentita | inputValido = true  messaggioErrore = “” |
| nomeUtente | null | inputValido = false  messaggioErrore = campo non compilato |
| nomeUtente | stringa vuota: “” | inputValido = false  messaggioErrore = campo non compilato |
| nomeUtente | stringa con lunghezza minore di quella consentita | inputValido = false  messaggioErrore = formato nome utente invalido |
| nomeUtente | stringa con lunghezza maggiore di quella consentita | inputValido = false  messaggioErrore = formato nome utente invalido |
| nomeUtente | stringa contente uno spazio | inputValido = false  messaggioErrore = formato nome utente invalido |
| nomeUtente | stringa che inizia con uno spazio | inputValido = false  messaggioErrore = formato nome utente invalido |
| nomeUtente | stringa che termina con uno spazio | inputValido = false  messaggioErrore = formato nome utente invalido |
| nomeUtente | stringa con lunghezza consentita | inputValido = true  messaggioErrore = “” |

Riferimento alla classe di test:

 **it.ingsw.natour21.control.ValidaCompilazioneCampiBlackBoxTest**



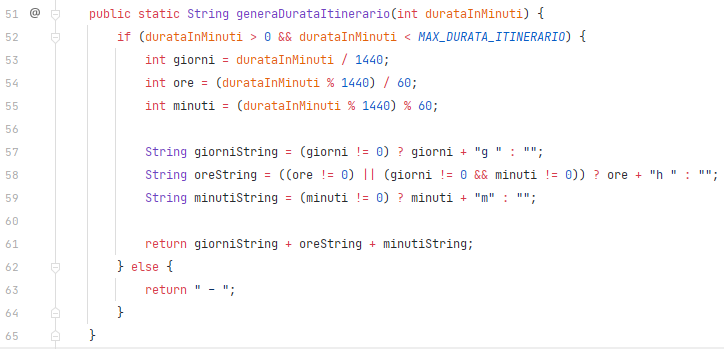






### 3.1.2 – Codice JUnit per il metodo generaDurataItinerario

Riferimento al metodo, applicativo mobile: **it.ingsw.natour21.control.Utils#generaDurataItinerario**

Il metodo generaDurataItinerario ha il compito di generare e restituire una stringa che rappresenti la durata temporale di un itinerario tramite il formato ‘Xg Yh Zm’ (giorni, ore, minuti). Il metodo accetta come parametri la durata di un itinerario espressa in minuti.

**Strategia Black-Box:**

Si effettua l’Input Space Partitioning, ovvero la suddivisione dei valori possibili assunti dal parametro in classi di equivalenza:

* durataInMinuti <= 0 - il metodo restituisce la stringa “ - “
* 0 < durataInMinuti <= MAX\_DURATA\_MINUTI - il metodo genera e restituisce la stringa rappresentante la durata in giorni, ore e minuti
* durataInMinuti > MAX\_DURATA\_MINUTI - il metodo restituisce la stringa “ - ”

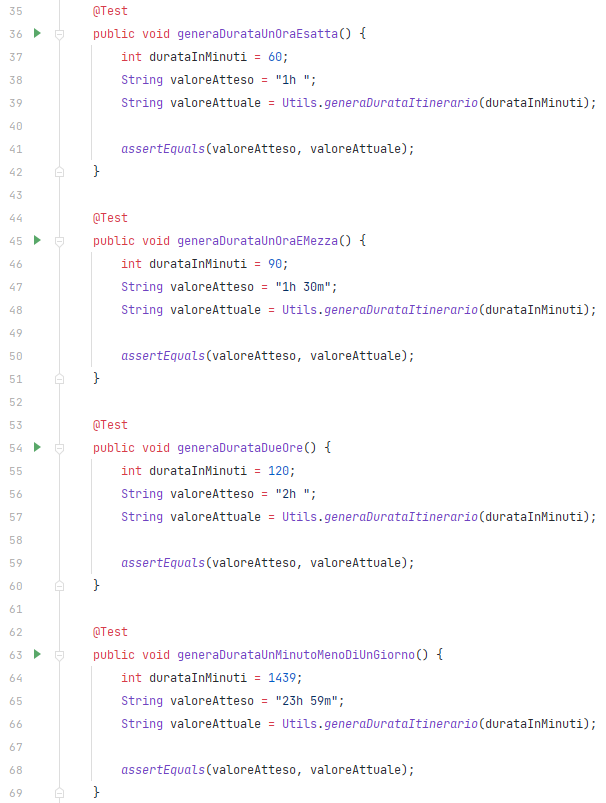
La stringa rappresentante la durata viene generata in maniera tale da omettere i giorni/ore/minuti quando questi sono uguali a 0. Unica eccezione di questa regola sono le ore, che possono essere omesse anche nel caso in cui i giorni e i minuti sono entrambi diversi da 0. Questo controllo viene effettuato dalle righe 57-59 in immagine.

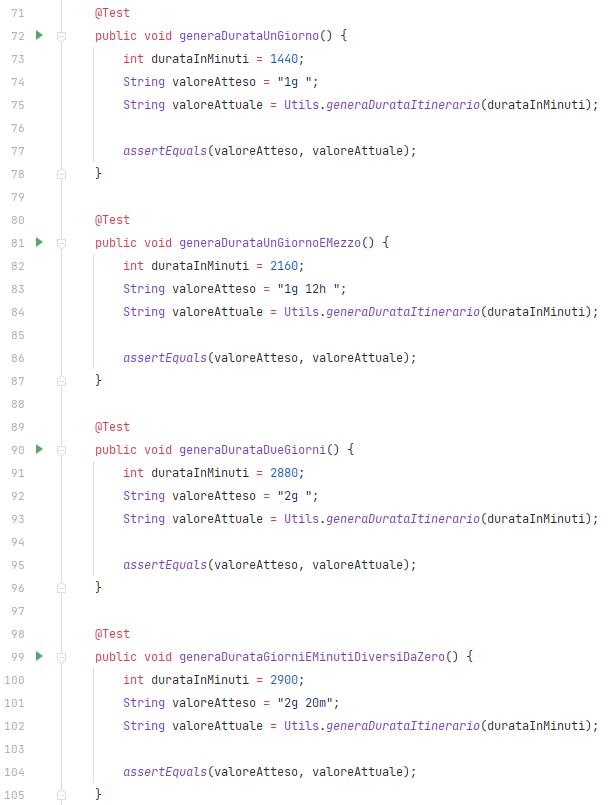
Il valore MAX\_DURATA\_MINUTI rappresenta la massima durata che un itinerario può avere, ed è al momento impostata al valore 14400, che equivale a 10 giorni

|  |  |  |
| --- | --- | --- |
| **Parametro** | **Valore assunto** | **Risultato atteso** |
| durataInMinuti | -100 | - |
| durataInMinuti | 0 | - |
| durataInMinuti | 59 | 59m |
| durataInMinuti | 60 | 60m |
| durataInMinuti | 90 | 1h 30m |
| durataInMinuti | 120 | 2h |
| durataInMinuti | 1439 | 23h 59m |
| durataInMinuti | 1440 | 1g |
| durataInMinuti | 2l60 | 1h 12h |
| durataInMinuti | 2880 | 2g |
| durataInMinuti | 2900 | 2g 20m |
| durataInMinuti | 14399 | 9g 23h 59m |
| durataInMinuti | 14400 | 10g |
| durataInMinuti | 14401 | - |

Riferimento alla classe Test:

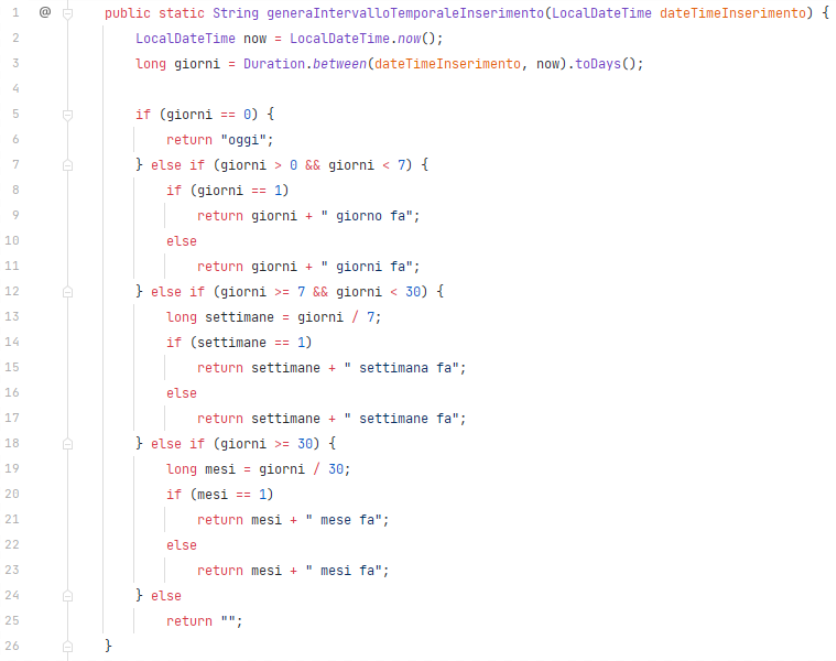
**it.ingsw.natour21.control.GeneraDurataItinerarioBlackBoxTest**





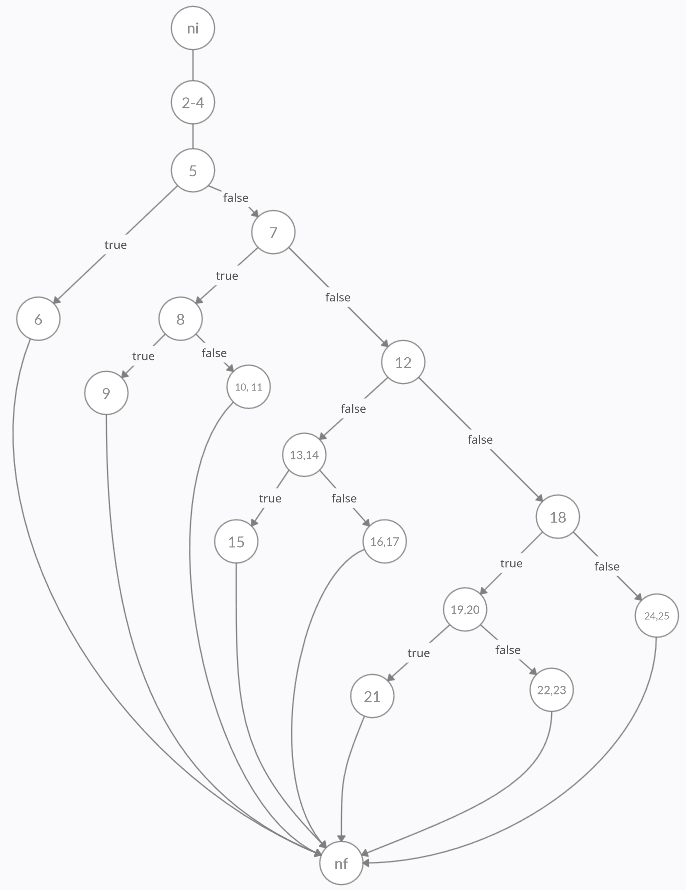
### 3.1.3 – Codice JUnit per il metodo generaIntervalloTemporaleInserimento

Riferimento al metodo, applicativo mobile: **it.ingsw.natour21.control.Utils#generaIntervalloTemporaleInserimento**

Il metodo generaIntervalloTemporaleInserimento ha il compito di generare una stringa che indica quanti giorni/settimane/mesi fa è stato inserito un itinerario in riferimento alla data odierna. Accetta come parametro il DateTime di inserimento dell’itinerario.

**Strategia strutturale (White-Box):**

Data la struttura complessa del metodo, composto da diversi conditional statements anche innestati tra di loro, è stato deciso di utilizzare una strategia di testing strutturale con copertura branch-coverage. Facendo riferimento al seguente grafo del flusso di controllo (GFC) sono stati scritti i seguenti test case:



Riferimento alla classe Test:

**it.ingsw.natour21.control.GeneraIntervalloTemporaleInserimentoBlackBoxTest**





## 3.2 – Valutazione dell’usabilità sul campo

Per l’usabilità dell’applicativo sul campo è stata effettuata una sessione di test sugli stessi utenti che hanno partecipato nella valutazione dell’usabilità a priori. Questa volta gli utenti hanno interagito con la versione finale dell’applicativo, ed è stato pensato un nuovo elenco di task da svolgere:

* **Task #1**: effettuare la registrazione e accedere con il proprio account
* **Task #2**: visualizzare i dettagli di un itinerario dalla schermata home
* **Task #3**: inserire un nuovo itinerario utilizzando la mappa
* **Task #4**: visualizzare uno degli itinerari che hai inserito precedentemente

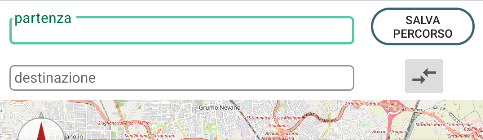
Di seguito sono mostrati i risultati dei test eseguiti sui tre utenti:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Task #1 | Task #2 | Task #3 | Task #4 |
| User #1 | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| User #2 | Checkmark with solid fill | Checkmark with solid fill | Close with solid fill | Checkmark with solid fill |
| User #3 | Close with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |

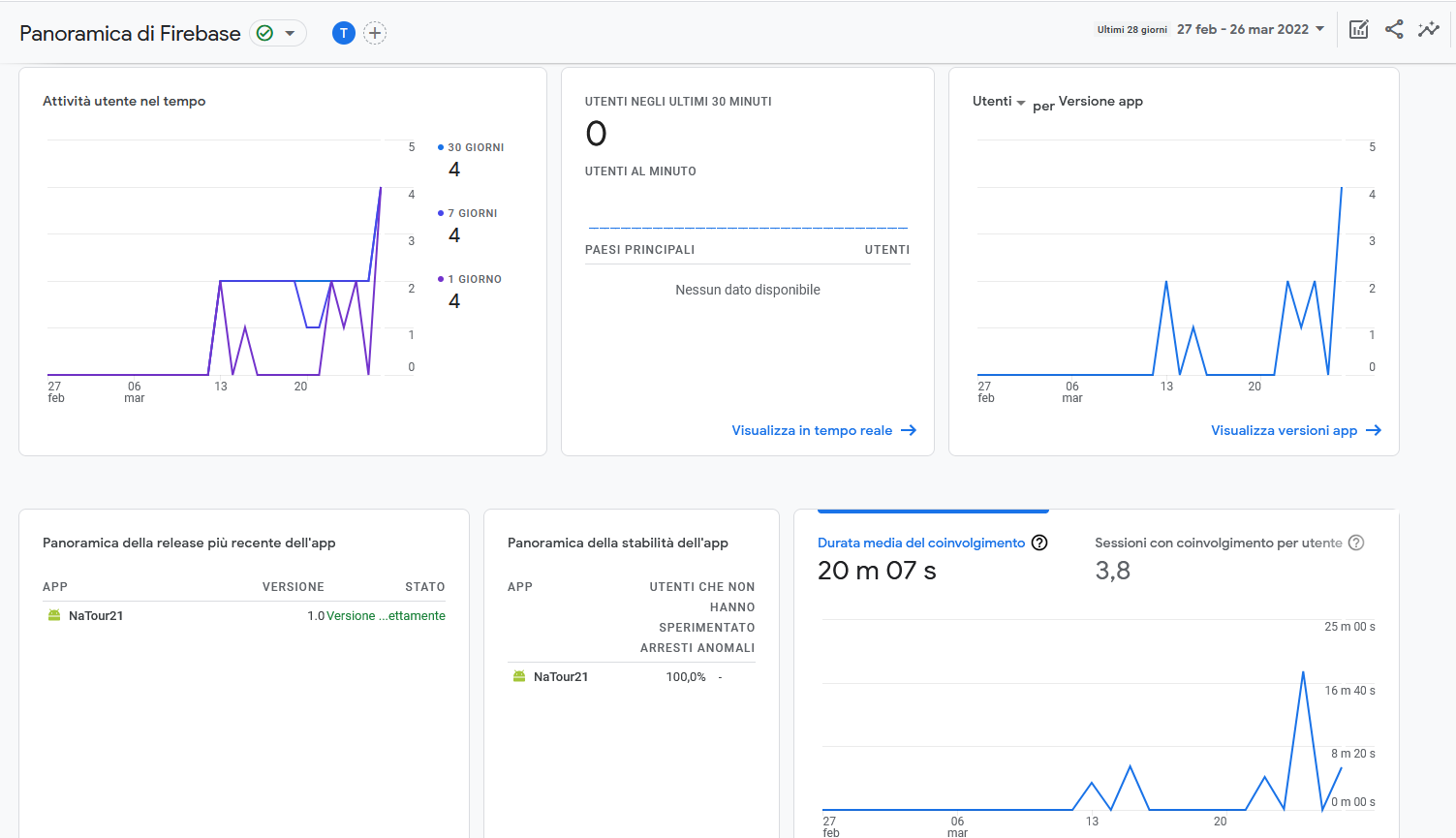
Tasso di successo = 10 successi /12 test totali = 83%

Dall’interazione con l’applicativo da parte degli utenti sono risultati evidenti alcuni problemi sia tecnici che di design:

* uno degli utenti ha riscontrato un problema durante il processo di autenticazione, dove non era capace a confermare la propria mail per poter poi effettuare l’accesso a causa di un bug del sistema.
* un altro utente invece ha avuto difficoltà con l’utilizzo dell’interfaccia di inserimento del percorso geografico di un itinerario **mp\_08\_nuovo\_itinerario\_mappa.** In dettaglio, l’interfaccia non offriva un’indicazione grafica su quale indirizzo si sta inserendo, la partenza o la destinazione. Di conseguenza, l’interfaccia è stata aggiornata appositamente per evidenziare l’indirizzo che l’utente seleziona:



La valutazione sul campo è stata anche una buona opportunità per mettere alla prova i vari strumenti di logging ed analytics che sono stati integrati nell’applicativo. Il sistema ha più modi di generare file di log e registrare gli eventi innescati dall’utente:

* log nella console di sviluppo, quello meno utile in quanto si ha la visibilità solo in ambiente di sviluppo
* file di log generati sul dispositivo di ogni utente
* Firebase Analytics: una piattaforma di analisi sviluppata da Google che mette a disposizione numerosi strumenti tra i quali:
  + una **dashboard** di riepilogo che offre numerose informazioni sugli utenti che hanno interagito con l’applicativo, come il loro numero, la loro regione geografica e il tempo di coinvolgimento medio
  + **Events**: un sistema che permette di registrare le azioni dell’utente, così come eventi di sistema ed errori



* + **DebugView**: una schermata che permette di monitorare gli eventi di uno specifico dispositivo in tempo reale, quando l’applicativo viene avviato nella modalità di debug di Analytics

