

Test Plan



Riferimento	
Versione	1.1
Data	10/02/2023
Presentato da:	NC28: Pietro Esposito, Alessandro Nacchia, Lorenzo Castellano
Approvato da:	



Revision History

Data	Versione	Descrizione	Autori
09/12/2023	0.1	Prima stesura.	Pietro Esposito, Alessandro Nacchia, Lorenzo Castellano
10/12/2023	1.0	Definiti tutti i Test Cases	Pietro Esposito
10/02/2024	1.1	Fix Errori	Pietro Esposito



Sommario

Revision History.....	2
Sommario.....	3
1. Introduzione	5
2. Relazione con altri documenti	5
3. Panoramica del sistema	6
3.2 Decomposizione in Sottosistemi.....	6
4. Features da testare / da non testare	7
5. Pass/Fail criteria	7
6. Approccio	8
6.1 Testing di Sistema	8
6.1.1 Functional Testing.....	8
6.1.2 Altri tipi di Testing.....	8
6.2 Testing di Integrazione	8
6.3 Testing di Unità	10
6.4 Ispezione del Codice	10
7. Sospensione e ripristino	10
7.1 Criteri di sospensione	10
7.2 Criteri di ripristino	10
8. Materiale di Testing	10
9. Test Cases	11
9.1 Gestione Utenza	11
9.1.1 Registrazione	11
9.1.2 Login	14
9.1.3 Modifica Dati Personali.....	16
9.2 Gestione Amministratori	19
9.2.1 Invio Notifica Singola	19



<i>9.2.2 Invio Notifica Broadcast</i>	20
9.3 Gestione Ricerca	21
<i>9.2.1 Ricerca</i>	21

1. Introduzione

La piattaforma FYP nasce per offrire supporto e guidare l'utente nella scelta di luoghi adatti alle proprie esigenze. L'utente ha la possibilità di ottenere informazioni in base ai parametri immessi, senza dover necessariamente consultare diverse fonti o cercare informazioni sul web, avendo invece tutto a disposizione in un unico ambiente e ricevendo insight e approfondimenti personalizzati ed elaborati attraverso un modulo di intelligenza artificiale.

Il documento di Test Plan ha l'obiettivo di descrivere ed analizzare le attività di Testing per la piattaforma FYP, garantendo che ogni aspetto funzioni in modo corretto. All'interno del documento sono riportate le strategie di testing adottate, quali funzionalità saranno testate e gli strumenti scelti per la rilevazione degli errori, con lo scopo di presentare al cliente finale una piattaforma priva di malfunzionamenti.

Le attività di testing sono state scelte in base ai seguenti sottosistemi:

- Gestione Utenza
- Gestione Amministratori
- Gestione Ricerca

2. Relazione con altri documenti

I Test Cases sono basati sulle funzionalità del sistema, individuate e raccolte nei seguenti documenti prodotti:

- **Requirements Analysis Document (RAD)**
I Test Case presenti nel Test Plan sono basati sui requisiti funzionali e non funzionali individuati nel RAD.
- **System Design Document (SDD)**
I Test Case presenti nel Test Plan rispettano la suddivisione in sottosistemi individuata nell'SDD.

3. Panoramica del sistema

Il sistema proposto è basato su un'architettura Client-Server, in quanto questa architettura è perfetta per lo sviluppo di web application come FYP, in quanto separa la logica di business dalla logica di presentazione.

Per la logica di business saranno usati:

- Framework SpringBoot (Java)
- MySQL (Database)

Per la logica di presentazione saranno usati:

- HTML5
- CSS3
- JavaScript

3.2 Decomposizione in Sottosistemi

Il sistema viene quindi suddiviso nei seguenti sottosistemi:

- **Gestione utenza:** si occupa di gestire le funzioni di registrazione, login, logout, modifica delle impostazioni utente, l'inserimento e la modifica dei dati personali, ricezioni di notifiche da parte del sistema o amministratore.
- **Gestione amministratori:** si occupa di gestire le funzionalità di visualizzazione utenti, modifica di alcuni parametri degli utenti, invio notifiche.
- **Gestione ricerca:** si occupa di gestire le funzioni di ricerca con o senza filtri applicabili, cronologia di ricerche, salvataggio di luoghi preferiti.
- **Gestione IdQ:** è responsabile del calcolo dell'indice di qualità dei luoghi trovati in una ricerca effettuata tramite un modello di intelligenza artificiale.
- **Persistenza:** Si occupa della gestione della persistenza tramite l'ausilio di un database.

4. Features da testare / da non testare

Di seguito la lista delle features di cui si effettuerà il testing:

- **Gestione Utenza**
 - Registrazione
 - Login
 - Modifica dati personali
- **Gestione Amministratori**
 - Invio Notifica Singola
 - Invio Notifica Broadcast
- **Gestione Ricerca**
 - Ricerca

Le funzionalità di cui non si andrà ad effettuare le attività di testing riguardano quelle che non prevedono input manuale da parte dell'utente, come le attività che riguardano esclusivamente il premere di un tasto.

Il sottosistema “**Gestione IdQ**” rappresenta il modulo di Intelligenza Artificiale, per cui questo non verrà trattato in questo documento.

5. Pass/Fail criteria

Le attività di testing sono mirate ad indentificare la presenza di errori all'interno del sistema. L'esito di un Test Case è valutato in base al risultato atteso della sua esecuzione, basandosi sui requisiti.

Un test ha successo (pass) se, dato un input, l'output ottenuto è diverso dall'output atteso.

Un test fallisce (fail) se, dato un input, l'output ottenuto è uguale all'output atteso.

Il testing deve rispettare i seguenti vincoli:

- Branch Coverage di almeno 75%;
- Testare tutti i requisiti che richiedono input dall'utente;
- Alla modifica o inserimento di caratteristiche nel sistema, effettuare test di regressione.

6. Approccio

Il testing del sistema è suddiviso in tre parti:

- Testing di Sistema;
- Testing di Integrazione;
- Testing di Unità.

6.1 Testing di Sistema

Per il testing di sistema sarà utilizzato il tool Selenium IDE, il quale permette di registrare le azioni che un utente può eseguire sul browser.

Il server verrà deployato in localhost.

6.1.1 Functional Testing

Il Functional Testing consiste nell'individuare i possibili errori generati dagli input degli utenti. Per farlo, viene utilizzata la tecnica del "Category Partition" per individuare i Test Cases.

6.1.2 Altri tipi di Testing

A causa del basso Budget a disposizione, non saranno eseguiti vari tipi di Testing, come quelli riguardanti le performance ed installazione.

6.2 Testing di Integrazione

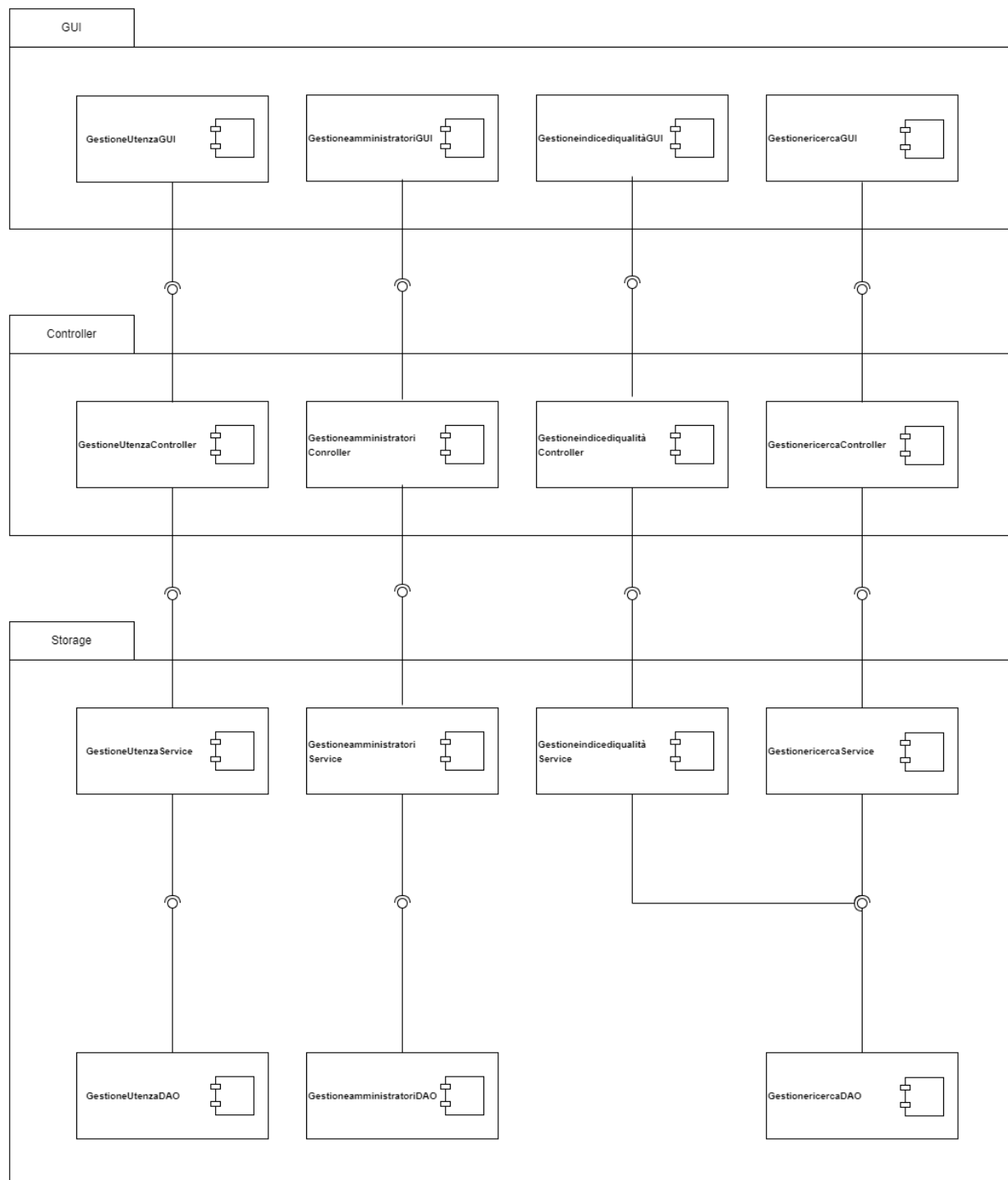
Il software sviluppato è basato sul metodo Object Oriented, per cui conviene utilizzare un approccio bottom-up.

Verrà utilizzato il framework JUnit per i test cases, mentre Mockito per il mocking.

Per la Continuous Integration si farà uso di Github Actions.

Si testeranno prima le classi Service e successivamente le classi Controller. Durante il test dei controller, verrà fatto il mocking tramite Mockito.

Di seguito viene mostrato il diagramma architetturale, esponendo le dipendenze tra i sottosistemi.



6.3 Testing di Unità

Per il Testing di Unità si adotterà la tecnica “Blackbox”, testando ogni metodo delle classi del sistema, escludendo le interfacce ed entity, in quanto queste ultime presentano solo metodi getters and setters.

6.4 Ispezione del Codice

Per aumentare la qualità del codice ci si affiderà principalmente al tool Checkstyle.
Considerato il Budget ridotto assegnato al progetto, non sarà possibile eseguire un’ispezione del codice approfondita e costante.

7. Sospensione e ripristino

In questa sezione vengono specificati i criteri di sospensione del test e le attività di test che dovranno essere ripetute quando si riprende il test.

7.1 Criteri di sospensione

Il testing non verrà sospeso fino alla sua terminazione, anche in caso di rilevazione di un errore. Il testing può essere sospeso solo se vi è un errore nella definizione stessa di uno dei test.

7.2 Criteri di ripristino

Il testing riprenderà solamente dopo aver risolto gli errori individuati.

8. Materiale di Testing

L’esecuzione dei test necessita di un pc non necessariamente avente connessione ad Internet, in quanto il sistema non è ancora stato rilasciato e quindi verrà utilizzato momentaneamente un server locale.

9. Test Cases

9.1 Gestione Utente

9.1.1 Registrazione

Parametro: Username	
Formato:	
$\wedge[A-Za-z][A-Za-z0-9_]\{4,29\}\$$	
Nome Categoria	Scelta per categoria
Formato [FU]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FU_OK]
Disponibilità [DU]	1. Disponibilità = false [ERROR] 2. Disponibilità = true [PROPERTY DU_OK]
Parametro: Email	
Formato:	
$\wedge\wedge+([.]\wedge\wedge+)*@\wedge+([.]\wedge\wedge+)*(\. \wedge\wedge\{2,3\})+\$$	
Nome Categoria	Scelta per categoria
Formato [FE]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FE_OK]
Disponibilità [DE]	1. Disponibilità = false [ERROR] 2. Disponibilità = true [PROPERTY DE_OK]
Parametro: Password	
Formato:	
$\wedge(=?\wedge.[0-9])(=?\wedge.[a-z])(=?\wedge.[A-Z])(=?\wedge.[!@\#\&()\{\}\ \:\;\,\?\wedge\$\wedge\-\+=<>]).\{8,50\}\$$	
Nome Categoria	Scelta per categoria
Formato [FP]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FP_OK]
Parametro: Conferma Password	
Formato:	
$\wedge(=?\wedge.[0-9])(=?\wedge.[a-z])(=?\wedge.[A-Z])(=?\wedge.[!@\#\&()\{\}\ \:\;\,\?\wedge\$\wedge\-\+=<>]).\{8,50\}\$$	
Nome Categoria	Scelta per categoria
Match [MP]	1. Match con Password = false [ERROR] 2. Match con Password = true [PROPERTY MP_OK]

Parametro: Nome	
Formato:	
$^{\wedge}(\{2,50\}\$)[A-Z\alpha-z\grave{A}-\grave{Y}]+([-\., '][A-Z\alpha-z\grave{A}-\grave{Y}]+)^*\$$	
Nome Categoria	Scelta per categoria
Formato [FN]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FN_OK]
Parametro: Cognome	
Formato:	
$^{\wedge}(\{2,50\}\$)[A-Z\alpha-z\grave{A}-\grave{Y}]+([-\., '][A-Z\alpha-z\grave{A}-\grave{Y}]+)^*\$$	
Nome Categoria	Scelta per categoria
Formato [FC]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FC_OK]
Parametro: Data Nascita	
Formato:	
$^{\wedge}(\backslash d\{4\})-(\backslash d\{2\})-(\backslash d\{2\})T(\backslash d\{2\}):(\backslash d\{2\}):(\backslash d\{2\})\$$	
Nome Categoria	Scelta per categoria
Formato [FDN]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FDN_OK]
Validità [VDN]	1. Data >= Data odierna [IF FDN_OK] [ERROR] 2. Data < Data odierna [IF FDN_OK] [PROPERTY VDN_OK]
Parametro: Numero Telefono	
Formato:	
$^{\wedge}([+]?([0-9]{1,3})?)?[-\s]?([0-9]{3})?[-\s]?([0-9]{3,10})[0-9]\$$	
Nome Categoria	Scelta per categoria
Inserimento [INSNT]	1. Inserimento = false [PROPERTY INSNT_NOT_OK] 2. Inserimento = true [PROPERTY INSNT_OK]
Formato [FNT]	1. Formato = false [IF INSNT_OK] [ERROR] 2. Formato = true [IF INSNT_OK] [PROPERTY FNT_OK]

Test Case ID	Test Frame	Esito
TC_1.1.1	FU1	Errato: formato username errato
TC_1.1.2	FU2, FE1	Errato: formato email errato
TC_1.1.3	FU2, FE2, FP1	Errato: formato password errato
TC_1.1.4	FU2, FE2, FP2, MP1	Errato: password non uguali
TC_1.1.5	FU2, FE2, FP2, MP2, FN1	Errato: formato nome errato
TC_1.1.6	FU2, FE2, FP2, MP2, FN2, FC1	Errato: formato cognome errato
TC_1.1.7	FU2, FE2, FP2, MP2, FN2, FC2, FDN1	Errato: formato data nascita
TC_1.1.8	FU2, FE2, FP2, MP2, FN2, FC2, FDN2, VDN1	Errato: Data nascita successiva o uguale alla data odierna
TC_1.1.9	FU2, FE2, FP2, MP2, FN2, FC2, FDN2, VDN2, INSNT1, DU2, DE2	Corretto
TC_1.1.10	FU2, FE2, FP2, MP2, FN2, FC2, FDN2, VDN2, INSNT2, FNT1	Errato: formato numero telefono errato
TC_1.1.11	FU2, FE2, FP2, MP2, FN2, FC2, FDN2, VDN2, INSNT2, FNT2, DU2, DE2	Corretto
TC_1.1.12	FU2, FE2, FP2, MP2, FN2, FC2, FDN2, VDN2, INSNT2, FNT2, DU1	Errore: Username non disponibile
TC_1.1.13	FU2, FE2, FP2, MP2, FN2, FC2, FDN2, VDN2, INSNT2, FNT2, DU2, DE1	Errore: Email non disponibile

9.1.2 Login

Parametro: Username	
Formato:	
$\wedge[A-Za-z][A-Za-z0-9_]\{4,29\}\$$	
Nome Categoria	Scelta per categoria
Match [MU]	1. Match con Username = false [ERROR] 2. Match con Username = true [PROPERTY MU_OK]
Parametro: Email	
Formato:	
$\wedge\wedge+([.-]?\wedge+)*@\wedge+([.-]?\wedge+)*(\.\wedge\{2,3\})+\$$	
Nome Categoria	Scelta per categoria
Match [ME]	1. Match con Email = false [ERROR] 2. Match con Email = true [PROPERTY ME_OK]
Parametro: Password	
Formato:	
$\wedge(?!.*[0-9])(?!.*[a-z])(?!.*[A-Z])(?!.*[!@\#\&()\[\]\{\}\;\:',?*\~\$^_-+=<>]).\{8,50\}\$$	
Nome Categoria	Scelta per categoria
Match [MP]	1. Match con Password = false [ERROR] 2. Match con Password = true [PROPERTY MP_OK]



Test Case ID	Test Frame	Esito
TC_1.2.1	MU1	Errato: match username errato
TC_1.2.2	ME1	Errato: match email errato
TC_1.2.3	MU2, MP1	Errato: match password errato
TC_1.2.4	ME2, MP1	Errato: match password errato
TC_1.2.5	MU2, MP2	Corretto
TC_1.2.6	ME2, MP2	Corretto

9.1.3 Modifica Dati Personali

Parametro: Username	
Formato:	
$\wedge[A-Za-z][A-Za-z0-9_]\{4,29\}\$$	
Nome Categoria	Scelta per categoria
Formato [FU]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FU_OK]
Match [MU]	1. Match con Username esistente = true [IF FU_OK] [ERROR] 2. Match con Username esistente = false [IF FU_OK] [PROPERTY MU_OK]
Parametro: Password	
Formato:	
$\wedge(?\=. *[0-9])(?\=. *[a-z])(?\=. *[A-Z])(?\=. *![@ \# \& () \backslash \{ \} \backslash \cdot ; ' , ? * \sim \$ \wedge \backslash - + = < >])\{8,50\}\$$	
Nome Categoria	Scelta per categoria
Inserimento [INSP]	1. Inserimento = false [PROPERTY INSP_NOT_OK] 2. Inserimento = true [PROPERTY INSP_OK]
Formato [FP]	1. Formato = false [IF INSP_OK] [ERROR] 2. Formato = true [IF INSP_OK] [PROPERTY FP_OK]
Parametro: Conferma Password	
Formato:	
$\wedge(?\=. *[0-9])(?\=. *[a-z])(?\=. *[A-Z])(?\=. *![@ \# \& () \backslash \{ \} \backslash \cdot ; ' , ? * \sim \$ \wedge \backslash - + = < >])\{8,50\}\$$	
Nome Categoria	Scelta per categoria
Match [MP]	1. Match con Password = false [ERROR] 2. Match con Password = true [PROPERTY MP_OK]
Parametro: Nome	
Formato:	
$\wedge(?\=. \{2,50\}\$)[A-Za-zÀ-ÿ]+([-\cdot '][A-Za-zÀ-ÿ]+)*\$$	
Nome Categoria	Scelta per categoria
Formato [FN]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FN_OK]

Parametro: Cognome	
Formato:	
$^{\wedge}(\{2,50\}\$)[A-Z\alpha-z\grave{A}-\ddot{y}]+([-\., '][A-Z\alpha-z\grave{A}-\ddot{y}])^*\$$	
Nome Categoria	Scelta per categoria
Formato [FC]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FC_OK]
Parametro: Data Nascita	
Formato:	
$^{\wedge}(\backslash d\{4\})-(\backslash d\{2\})-(\backslash d\{2\})T(\backslash d\{2\}):(\backslash d\{2\}):(\backslash d\{2\})\$$	
Nome Categoria	Scelta per categoria
Formato [FDN]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FDN_OK]
Validità [VDN]	1. Data >= Data odierna [IF FDN_OK] [ERROR] 2. Data < Data odierna [IF FDN_OK] [PROPERTY VDN_OK]
Parametro: Numero Telefono	
Formato:	
$^{\wedge}([+]?([0-9]{1,3})?[-\s])?([0-9]{3})?[-\s]?([0-9]{3,10})[0-9]\$$	
Nome Categoria	Scelta per categoria
Inserimento [INSNT]	1. Inserimento = false [PROPERTY INSNT_NOT_OK] 2. Inserimento = true [PROPERTY INSNT_OK]
Formato [FNT]	1. Formato = false [IF INSNT_OK] [ERROR] 2. Formato = true [IF INSNT_OK] [PROPERTY FNT_OK]
Parametro: Notifiche	
Nome Categoria	Scelta per categoria
Selezionato [SN]	1. Selezionato = false [ERROR] 2. Selezionato = true [PROPERTY SN_OK]
Parametro: isStudente	
Nome Categoria	Scelta per categoria
Selezionato [SiS]	1. Selezionato = false [PROPERTY SiS_NOT_SEL] 2. Selezionato = true [PROPERTY SiS_SEL]
Parametro: isGenitore	
Nome Categoria	Scelta per categoria
Selezionato [SiG]	1. Selezionato = false [PROPERTY SiG_NOT_SEL] 2. Selezionato = true [PROPERTY SiG_SEL]

Test Case ID	Test Frame	Esito
TC_1.3.1	FU1	Errato: formato username errato
TC_1.3.2	FU2, MU1	Errato: username già usato
TC_1.3.3	FU2, MU2, FN1	Errato: formato nome errato
TC_1.3.4	FU2, MU2, FN2, FC1	Errato: formato cognome errato
TC_1.3.5	FU2, MU2, FN2, FC2, FDN1	Errato: formato data nascita
TC_1.3.6	FU2, MU2, FN2, FC2, FDN2, VDN1	Errato: Data nascita successiva o uguale alla data odierna
TC_1.3.7	FU2, MU2, FN2, FC2, FDN2, VDN2, SN1	Errato: Notifiche non selezionato
TC_1.3.8	FU2, MU2, FN2, FC2, FDN2, VDN2, SN2, INSNT1, INSP1	Corretto
TC_1.3.9	FU2, MU2, FN2, FC2, FDN2, VDN2, INSNT2, FNT1	Errato: formato numero telefono errato
TC_1.3.10	FU2, MU2, FN2, FC2, FDN2, VDN2, INSNT2, FNT2, INSP1	Corretto
TC_1.3.11	FU2, MU2, FN2, FC2, FDN2, VDN2, SN2, INSNT1, INSP2, FP1	Errato: formato password errato
TC_1.3.12	FU2, MU2, FN2, FC2, FDN2, VDN2, SN2, INSNT1, INSP2, FP2, MP1	Errato: password non uguali
TC_1.3.13	FU2, MU2, FN2, FC2, FDN2, VDN2, SN2, INSNT1, INSP2, FP2, MP2	Corretto
TC_1.3.14	FU2, MU2, FN2, FC2, FDN2, VDN2, SN2, INSNT1, INSP2, FP2, MP2, SIS1, SIG1	Corretto
TC_1.3.15	FU2, MU2, FN2, FC2, FDN2, VDN2, SN2, INSNT1, INSP2, FP2, MP2, SIS2, SIG2	Corretto

9.2 Gestione Amministratori

9.2.1 Invio Notifica Singola

Parametro: Username Utente	
Formato:	
$\wedge[A-Za-z][A-Za-z0-9_]\{4,29\}\$$	
Nome Categoria	Scelta per categoria
Match [MU]	1. Match con Username = false [ERROR] 2. Match con Username = true [PROPERTY MU_OK]
Parametro: Testo	
Formato:	
$\wedge[\s\S]\{2,1000\}\$$	
Nome Categoria	Scelta per categoria
Formato [FT]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FT_OK]
Parametro: Data Scadenza	
Formato:	
$\wedge(\d{4})-(\d{2})-(\d{2})T(\d{2}):(\d{2}):(\d{2})\$$	
Nome Categoria	Scelta per categoria
Formato [FDS]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FDS_OK]
Validità [VDS]	1. Data <= Data odierna [IF FDS_OK] [ERROR] 2. Data > Data odierna [IF FDS_OK] [PROPERTY VDS_OK]

Test Case ID	Test Frame	Esito
TC_2.1.1	MU1	Errato: match username errato
TC_2.1.2	MU2, FT1	Errato: formato testo errato
TC_2.1.3	MU2, FT2, FDS1	Errato: formato data scadenza errato
TC_2.1.4	MU2, FT2, FDS2, VDS1	Errato: Data scadenza precedente o uguale alla data odierna
TC_2.1.5	MU2, FT2, FDS2, VDS2	Corretto

9.2.2 Invio Notifica Broadcast

Parametro: Testo	
Formato: $^{\wedge}[\backslash s\backslash S]\{2,1000\}\$$	
Nome Categoria	Scelta per categoria
Formato [FT]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FT_OK]
Parametro: Data Scadenza	
Formato: $^{\wedge}(\backslash d\{4\})-(\backslash d\{2\})-(\backslash d\{2\})T(\backslash d\{2\}):(\backslash d\{2\}):(\backslash d\{2\})\$$	
Nome Categoria	Scelta per categoria
Formato [FDS]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FDS_OK]
Validità [VDS]	1. Data <= Data odierna [IF FDS_OK] [ERROR] 2. Data > Data odierna [IF FDS_OK] [PROPERTY VDS_OK]

Test Case ID	Test Frame	Esito
TC_2.2.1	FT1	Errato: formato testo errato
TC_2.2.2	FT2, FDS1	Errato: formato data scadenza errato
TC_2.2.3	FT2, FDS2, VDS1	Errato: Data scadenza precedente o uguale alla data odierna
TC_2.2.4	FT2, FDS2, VDS2	Corretto

9.3 Gestione Ricerca

9.2.1 Ricerca

Parametro: Coordinate	
Formato: $\wedge((\backslash-? \backslash+?)?\backslash d+(\backslash.\backslash d+)?),\backslash s*((\backslash-? \backslash+?)?\backslash d+(\backslash.\backslash d+)?)\$$	
Nome Categoria	Scelta per categoria
Formato [FXY]	1. Formato = false [ERROR] 2. Formato = true [PROPERTY FXY_OK]
Parametro: Range	
Nome Categoria	Scelta per categoria
Grandezza [GRng]	1. Grandezza < 2 OR > 500 = false [ERROR] 2. Grandezza >=2 AND <=500 = true [PROPERTY GRng_OK]
Parametro: Costo Vita	
Nome Categoria	Scelta per categoria
Selezionato [SCV]	1. Selezionato = false [PROPERTY SCV_NOT_SEL] 2. Selezionato = true [PROPERTY SCV_SEL]
Parametro: Pericolosità Massima	
Nome Categoria	Scelta per categoria
Selezionato [SPMax]	1. Selezionato = false [PROPERTY SPMax_NOT_SEL] 2. Selezionato = true [PROPERTY SPMax_SEL]
Grandezza [GPMax]	1. Grandezza < 2 OR > 100 = false [IF SPMax_SEL] [ERROR] 2. Grandezza >=2 AND <=100 = true [IF SPMax_SEL] [PROPERTY GPMax_OK]
Parametro: Numero Abitanti Minimo	
Nome Categoria	Scelta per categoria
Selezionato [SAMin]	1. Selezionato = false [PROPERTY SAMin_NOT_SEL] 2. Selezionato = true [PROPERTY SAMin_SEL]
Grandezza [GAMin]	1. Grandezza < 0 OR > 1000000 = false [IF SAMin_SEL] [ERROR] 2. Grandezza >=0 AND <=1000000 = true [IF SAMin_SEL] [PROPERTY GAMin_OK]



Parametro: Numero Abitanti Massimo	
Nome Categoria	Scelta per categoria
Selezionato [SAMax]	1. Selezionato = false [PROPERTY SAMax_NOT_SEL] 2. Selezionato = true [PROPERTY SAMax_SEL]
Grandezza [GAMax]	1. Grandezza < 1000 OR > 100000000 = false [IF SAMax_SEL] [ERROR] 2. Grandezza >=1000 AND <=100000000 = true [IF SAMax_SEL] [PROPERTY GAMax_OK]
Parametro: Numero Negozi Minimo	
Nome Categoria	Scelta per categoria
Selezionato [SNMin]	1. Selezionato = false [PROPERTY SNMin_NOT_SEL] 2. Selezionato = true [PROPERTY SNMin_SEL]
Grandezza [GNMin]	1. Grandezza < 0 OR > 10000 = false [IF SNMin_SEL] [ERROR] 2. Grandezza >=0 AND <=10000 = true [IF SNMin_SEL] [PROPERTY GNMin_OK]
Parametro: Numero Scuole Minimo	
Nome Categoria	Scelta per categoria
Selezionato [SSMin]	1. Selezionato = false [PROPERTY SSMIn_NOT_SEL] 2. Selezionato = true [PROPERTY SSMIn_SEL]
Grandezza [GSMin]	1. Grandezza < 0 OR > 100 = false [IF SSMIn_SEL] [ERROR] 2. Grandezza >=0 AND <=100 = true [IF SSMIn_SEL] [PROPERTY GSMin_OK]
Parametro: Numero Ristoranti Minimo	
Nome Categoria	Scelta per categoria
Selezionato [SRMin]	1. Selezionato = false [PROPERTY SRMin_NOT_SEL] 2. Selezionato = true [PROPERTY SRMin_SEL]
Grandezza [GRMin]	1. Grandezza < 0 OR > 1000 = false [IF SRMin_SEL] [ERROR] 2. Grandezza >=0 AND <=1000 = true [IF SRMin_SEL] [PROPERTY GRMin_OK]

Test Case ID	Test Frame	Esito
TC_3.1.1	FXY1	Errato: formato coordinate errato
TC_3.1.2	FXY2, GRng1	Errato: grandezza range errata
TC_3.1.3	FXY2, GRng2, SCV1, SPMaX1, SAMin1, SAMaX1, SNMin1, SSMIn1, SRMin1	Corretto
TC_3.1.4.1	FXY2, GRng2, SCV2, SPMaX2, GPMax1	Errato: Grandezza Pericolosità massima errata
TC_3.1.4.2	FXY2, GRng2, SCV2, SAMin2, GAMin1	Errato: Grandezza Abitanti minimi errata
TC_3.1.4.3	FXY2, GRng2, SCV2, SAMaX2, GAMaX1	Errato: Grandezza Abitanti massimi errata
TC_3.1.4.4	FXY2, GRng2, SCV2, SNMin2, GNMin1	Errato: Grandezza Negozi minimi errata
TC_3.1.4.5	FXY2, GRng2, SCV2, SSMIn2, GSMin1	Errato: Grandezza Scuole minime errata
TC_3.1.4.6	FXY2, GRng2, SCV2 SRMin2, GRMin1	Errato: Grandezza Ristoranti minimi errata
TC_3.1.5	FXY2, GRng2, SCV2, SPMaX2, GPMax2, SAMin2, GAMin2, SAMaX2, GAMaX2, SNMin2, GNMin2, SSMIn2, GSMin2, SRMin2, GRMin2	Corretto