



AlphaSix

AlphaSix

Analisi dei Requisiti

Informazioni sul documento

Nome Documento	AnalisiDeiRequisiti v3.0.0.pdf
Versione	3.0.0
Data di Creazione	27 novembre 2018
Data ultima modifica	08 aprile 2019
Stato	Approvato
Redazione	Ciprian Voinea Nicola Carlesso
Verifica	Matteo Marchiori Samuele Gardin
Approvazione	Nicola Carlesso
Uso	Esterno
Distribuzione	AlphaSix
Destinato a	Prof. Tullio Vardanega, Prof. Riccardo Cardin, Imola Informatica
Email di riferimento	alpha.six.unipd@gmail.com

Descrizione

L'Analisi dei Requisiti descrive le funzionalità di Butterfly, determinandone i requisiti che analizziamo e valutiamo.

Registro delle modifiche

Versione	Descrizione	Ruolo	Nominativo	Data
3.0.0	Approvazione per il rilascio	Responsabile	Nicola Carlesso	2019-04-09
2.3.0	Verifica finale	Verificatore	Matteo Marchiori	2019-04-08
2.2.1	Aggiunta immagine container Rancher in §A.9	Analista	Ciprian Voinea	2019-04-06
2.2.0	Verifica	Verificatore	Ciprian Voinea	2019-03-29
2.1.2	Rimosso requisito non verificabile	Analista	Laura Cameran	2019-03-27
2.1.1	Aggiunti casi d'uso riguardanti commenti di issue e commit	Analista	Samuele Gardin	2019-03-26
2.2.1	Aggiunto §2.3.4.1 e ampliato §2.3.4 per il Gestore Personale e l'inoltro delle notifiche	Analista	Ciprian Voinea	2019-03-26
2.1.0	Verifica	Verificatore	Nicola Carlesso	2019-03-25
2.0.2	Modificata introduzione casi d'uso in §3	Verificatore	Matteo Marchiori	2019-03-24
2.0.2	Verificati i nomi dei casi d'uso per correzione RP	Verificatore	Matteo Marchiori	2019-03-22
2.0.1	Aggiunta di Rancher in §A.8.1 per gestione container e MongoDB in §A.5	Responsabile	Ciprian Voinea	2019-03-22
2.0.0	Approvazione per il rilascio	Responsabile	Samuele Gardin	2019-03-04
1.3.0	Verifica finale	Verificatore	Timoty Granziero	2019-03-01
1.2.3	Migliorato §4.1	Analista	Ciprian Voinea	2019-02-27
1.2.2	Verifica §3	Verificatore	Matteo Marchiori	2019-02-26
1.2.1	Aggiornati diagrammi UML in §3	Analista	Nicola Carlesso	2019-02-20
1.2.0	Verifica	Verificatore	Samuele Gardin	2019-02-18
1.1.3	Riviste sezioni §3	Analista	Nicola Carlesso	2019-02-16
1.1.2	Verifica §3	Verificatore	Matteo Marchiori	2019-02-09
1.1.1	Approfondimento §3	Analista	Ciprian Voinea	2019-02-06
1.1.0	Verifica	Verificatore	Matteo Marchiori	2019-02-04
1.0.3	Inserimento §3	Analista	Ciprian Voinea	2019-01-28
1.0.2	Approfondimento §2	Responsabile	Laura Cameran	2019-01-26
1.0.1	Migrazione tecnologie in appendice §A	Analista	Ciprian Voinea	2019-01-24
1.0.0	Approvazione per il rilascio	Responsabile	Nicola Carlesso	2019-01-12
0.3.0	Verifica finale	Verificatore	Matteo Marchiori	2019-01-10
0.2.6	Aggiunto §4.5	Analista	Laura Cameran	2018-12-28

Versione	Descrizione	Ruolo	Nominativo	Data
0.2.5	Aggiunto §4.3	Analista	Laura Cameran	2018-12-27
0.2.4	Aggiunto §4.4	Analista	Nicola Carlesso	2018-12-23
0.2.3	Inserito §4.2	Analista	Nicola Carlesso	2018-12-20
0.2.2	Aggiunto §4.1	Analista	Laura Cameran	2018-12-19
0.2.1	Iniziati requisiti	Analista	Laura Cameran	2018-12-15
0.2.0	Verifica del documento	Verificatore	Timoty Granziero	2018-12-15
0.1.9	Inserito UC4	Analista	Timoty Granziero	2018-12-10
0.1.8	Inserito UC1	Analista	Matteo Marchiori	2018-12-10
0.1.7	Inserito UC2	Analista	Laura Cameran	2018-12-09
0.1.6	Inserito UC3	Analista	Ciprian Voinea	2018-12-08
0.1.5	Inserito UC5	Analista	Samuele Gardin	2018-12-07
0.1.4	Aggiunto UC6	Analista	Nicola Carlesso	2018-12-07
0.1.3	Aggiunto §A.8.1	Analista	Ciprian Voinea	2018-12-06
0.1.2	Aggiunto §2.3.4	Analista	Ciprian Voinea	2018-12-05
0.1.1	Aggiunto §3	Analista	Laura Cameran	2018-12-05
0.1.0	Verifica documento	Verificatore	Nicola Carlesso	2018-12-05
0.0.6	Aggiunto §2.3.3	Analista	Ciprian Voinea	2018-12-04
0.0.5	Aggiunti §2.3.2 e §2.3.1	Analista	Ciprian Voinea	2018-12-03
0.0.4	Aggiunta §2	Analista	Matteo Marchiori	2018-12-01
0.0.3	Aggiunto §1	Analista	Nicola Carlesso	2018-11-30
0.0.2	Creazione struttura documento	Analista	Nicola Carlesso	2018-11-29
0.0.1	Creazione template	Redattore	Timoty Granziero	2018-11-27

Indice

1	Introduzione	1
1.1	Glossario e documenti esterni	1
1.2	Scopo documento	1
1.3	Scopo del prodotto	1
1.4	Riferimenti	1
1.4.1	Normativi	1
1.4.2	Informativi	1
2	Descrizione del prodotto	3
2.1	Funzionalità	3
2.2	Caratteristiche degli utenti	3
2.3	Tecnologie e scelte relative al progetto	3
2.3.1	Producer	4
2.3.1.1	Producer GitLab	4
2.3.1.2	Producer Redmine	4
2.3.2	Apache Kafka	5
2.3.3	Consumer	5
2.3.3.1	Consumer Telegram	5
2.3.3.2	Consumer Email	5
2.3.4	Gestore Personale	5
2.3.4.1	Inoltro notifiche	6
2.3.4.2	Informazioni aggiuntive	6
3	Casi d'uso	7
3.1	Attori	7
3.2	Sistemi e sottosistemi	7
3.3	Elenco casi d'uso	9
3.3.1	UC1-PR - Redmine invia segnalazione al Producer Redmine	9
3.3.2	UC1.1-PR - Redmine segnala apertura issue al Producer Redmine	9
3.3.3	UC1.2-PR - Redmine segnala la modifica di una issue al Producer Redmine	10
3.3.4	UC1.3-PR - Redmine segnala il commento di una issue al Producer Redmine	11
3.3.5	UC2-PG - GitLab invia segnalazione al Producer GitLab	12
3.3.6	UC2.1-PG - GitLab segnala apertura issue al Producer GitLab	13
3.3.7	UC2.2-PG - Gitlab segnala la modifica di una issue al Producer Gitlab	13
3.3.8	UC2.3-PG - Gitlab segnala il commento di una issue al Producer Gitlab	14
3.3.9	UC2.4-PG - GitLab segnala un evento di push a Producer GitLab	14
3.3.10	UC2.5-PG - Gitlab segnala il commento di un commit al Producer Gitlab	15
3.3.11	UC3-GP - Producer Redmine invia messaggio al Gestore Personale	16
3.3.12	UC3.1-GP - Producer Redmine invia messaggio di apertura issue al Gestore Personale	17
3.3.13	UC3.2-GP - Producer Redmine invia messaggio di modifica issue al Gestore Personale	17
3.3.14	UC3.3-GP - Producer Redmine invia messaggio di commento issue al Gestore Personale	18
3.3.15	UC4-GP - Producer GitLab invia messaggio al Gestore Personale	19
3.3.16	UC4.1-GP - Producer GitLab invia messaggio di push al Gestore Personale	20
3.3.17	UC4.2-GP - Producer GitLab invia messaggio di una nuova issue al Gestore Personale	21

3.3.18	UC4.3-GP - Producer GitLab invia messaggio di modifica di una issue al Gestore Personale	21
3.3.19	UC4.4-GP - Producer GitLab invia messaggio di commento di una issue al Gestore Personale	22
3.3.20	UC4.5-GP - Producer GitLab invia messaggio di commento di un commit al Gestore Personale	22
3.3.21	UC5-CT - Gestore Personale invia il messaggio finale al Consumer Telegram	23
3.3.22	UC6-CE - Gestore Personale invia il messaggio finale al Consumer Email .	24
3.3.23	UC7-BT - Consumer Telegram inoltra il messaggio finale al bot Telegram	25
3.3.24	UC8-SE - Consumer Email inoltra il messaggio finale al server Email . . .	26
3.3.25	UC9-GP - Accesso	26
3.3.26	UC9.1-GP - Accesso dell'utente nel sistema	27
3.3.27	UC9.1.1-GP - Inserimento identificativo	28
3.3.28	UC9.2-GP - Errore identificativo inesistente	28
3.3.29	UC10-GP - Uscita dell'utente dal sistema	28
3.3.30	UC11-GP - Inserimento utente	29
3.3.31	UC11.1-GP - Aggiunta nuovo utente	30
3.3.32	UC11.1.1-GP - Inserimento nome utente	31
3.3.33	UC11.1.2-GP - Inserimento cognome utente	31
3.3.34	UC11.1.3-GP - Inserimento contatto Email	31
3.3.35	UC11.1.4-GP - Inserimento contatto Telegram	32
3.3.36	UC11.2-GP - Errore utente già presente nel sistema	32
3.3.37	UC12-GP - Rimozione utente	33
3.3.38	UC12.1-GP - Rimozione utente dal sistema	34
3.3.39	UC12.1.1-GP - Inserimento contatto Email	35
3.3.40	UC12.1.2-GP - Inserimento contatto Telegram	35
3.3.41	UC12.2-GP - Errore contatto non presente nel sistema	35
3.3.42	UC13-GP - Modifica utente	36
3.3.43	UC13.1-GP - Modifica di un utente del sistema	37
3.3.44	UC13.1.1-GP - Inserimento del nuovo nome	38
3.3.45	UC13.1.2-GP - Inserimento del nuovo cognome	38
3.3.46	UC13.1.3-GP - Inserimento del nuovo contatto Email	38
3.3.47	UC13.1.4-GP - Inserimento del nuovo contatto Telegram	39
3.3.48	UC13.2-GP - Errore, nuovi dati dell'utente già esistenti	39
3.3.49	UC14-GP - Aggiunta preferenze	40
3.3.50	UC14.1-GP - Iscrizione Topic	41
3.3.51	UC14.2-GP - Aggiunta dei giorni di indisponibilità nel calendario	41
3.3.52	UC14.3-GP - Aggiunta della piattaforma di messaggistica preferita	42
3.3.53	UC14.4-GP - Aggiunta keyword per i push di GitLab	42
3.3.54	UC14.5-GP - Errore keyword già esistente	42
3.3.55	UC14.6-GP - Aggiunta progetti di interesse	43
3.3.56	UC14.7-GP - Aggiunta priorità progetti	43
3.3.57	UC15-GP - Rimozione preferenze	44
3.3.58	UC15.1-GP - Disiscrizione Topic	45
3.3.59	UC15.2-GP - Rimozione di uno o più giorni di irreperibilità nel calendario	45
3.3.60	UC15.3-GP - Rimozione preferenza piattaforma di messaggistica	46
3.3.61	UC15.4-GP - Rimozione di keyword per i push di GitLab	46
3.3.62	UC15.5-GP - Errore keyword da rimuovere non presente	46
3.3.63	UC15.6-GP - Rimozione progetti di interesse	47
3.3.64	UC15.7-GP - Rimozione delle priorità sui progetti	47

4	Requisiti	48
4.1	Requisiti di funzionalità	49
4.2	Requisiti di qualità	53
4.3	Requisiti di vincolo	55
4.4	Tracciamento	56
4.4.1	Tracciamento fonte-requisito	56
4.4.2	Tracciamento requisito-fonte	64
4.5	Riepilogo	71
A	Approfondimenti tecnologici	72
A.1	Webhook	72
A.2	Redmine	72
A.3	GitLab	73
A.4	Apache Kafka	73
A.5	MongoDB	73
A.6	Telegram	73
A.6.1	BotFather	73
A.7	Email	73
A.8	Docker	73
A.8.1	Container software	74
A.8.1.1	Differenza tra container e macchina virtuale	74
A.8.2	DockStation	74
A.9	Rancher	74

Elenco delle tabelle

1	Elenco dei requisiti di funzionalità (1)	49
2	Elenco dei requisiti di funzionalità (2)	50
3	Elenco dei requisiti di funzionalità (3)	51
4	Elenco dei requisiti di funzionalità (4)	52
5	Elenco dei requisiti di qualità (1)	53
6	Elenco dei requisiti di qualità (2)	54
7	Elenco dei requisiti di vincolo	55
8	Elenco dei requisiti del capitolato (1)	56
9	Elenco dei requisiti del capitolato (2)	57
10	Elenco dei requisiti per i casi d'uso (1)	58
11	Elenco dei requisiti per i casi d'uso (2)	59
12	Elenco dei requisiti per i casi d'uso (3)	60
13	Elenco dei requisiti per gli obiettivi di qualità e verbali	60
14	Elenco dei requisiti da fonte interna (1)	61
15	Elenco dei requisiti da fonte interna (2)	62
16	Elenco dei requisiti da fonte interna (3)	63
17	Elenco dei requisiti funzionali in rapporto alle fonti (1)	64
18	Elenco dei requisiti funzionali in rapporto alle fonti (2)	65
19	Elenco dei requisiti funzionali in rapporto alle fonti (3)	66
20	Elenco dei requisiti funzionali in rapporto alle fonti (4)	67
21	Elenco dei requisiti funzionali in rapporto alle fonti (5)	68
22	Elenco dei requisiti di qualità in rapporto alle fonti (1)	69
23	Elenco dei requisiti di qualità in rapporto alle fonti (2)	70
24	Elenco dei requisiti di vincolo in rapporto alle fonti	71
25	Riepilogo dei requisiti	71

Elenco delle figure

1	Visione generale del sistema Butterfly	3
2	UC1-PR - Redmine invia segnalazione al Producer Redmine	9
3	UC2-PG - GitLab invia segnalazione al Producer GitLab	12
4	UC3-GP - Producer Redmine invia messaggio al Gestore Personale	16
5	UC4-GP - Producer GitLab invia messaggio al Gestore Personale	19
6	UC5-CT - Gestore Personale invia il messaggio finale al Consumer Telegram	23
7	UC6-CE - Gestore Personale invia il messaggio finale al Consumer Email	24
8	UC7-BT - Consumer Telegram inoltra il messaggio finale al bot Telegram	25
9	UC8-SE - Consumer Email inoltra il messaggio finale al server Email	26
10	UC9-GP - Accesso	26
11	UC9.1-GP - Accesso dell'utente nel sistema	27
12	UC10-GP - Uscita dell'utente dal sistema	28
13	UC11-GP - Inserimento utente	29
14	UC11.1-GP - Aggiunta nuovo utente	30
15	UC12-GP - Rimozione utente	33
16	UC12.1-GP - Rimozione utente dal sistema	34
17	UC13-GP - Modifica utente	36
18	UC13.1-GP - Modifica di un utente del sistema	37
19	UC14-GP - Aggiunta preferenze	40
20	UC15-GP - Rimozione preferenze	44
21	Alcuni servizi attivi su Rancher	75

1 Introduzione

1.1 Glossario e documenti esterni

Al fine di rendere il documento più chiaro possibile, i termini che possono assumere un significato ambiguo o i riferimenti a documenti esterni avranno delle diciture convenzionali:

- **D**: indica che il termine si riferisce al nome di un particolare documento (ad esempio *PianoDiProgetto v4.0.0_D*).
- **G**: indica che il termine si riferisce ad una voce riportata nel *Glossario v3.0.0_D* (ad esempio *REDMINE_G*).

1.2 Scopo documento

Tale DOCUMENTO_G ha l'obiettivo di esporre e analizzare i REQUISITI_G espliciti e impliciti per la realizzazione del PROGETTO_G Butterfly (C1) proposto dall'azienda Imola Informatica.

Il documento vuole fungere da base per la fase di progettazione del software in modo che essa sia conforme alle richieste fatte dall'azienda proponente.

1.3 Scopo del prodotto

Lo scopo del PRODOTTO_G è creare un APPLICATIVO_G per poter gestire i messaggi o le segnalazioni provenienti da diversi prodotti per la realizzazione di software, come REDMINE_G, GITLAB_G e opzionalmente SONARQUBE_G, attraverso un BROKER_G che possa incanalare questi messaggi e distribuirli a strumenti come TELEGRAM_G, e-mail e opzionalmente SLACK_G.

Per ciascuna di queste tecnologie viene creato un PRODUCER_G e un CONSUMER_G associato per l'effettiva ricezione e invio delle varie segnalazioni. Il software dovrà inoltre essere in grado di riconoscere il TOPIC_G dei messaggi in input per poterli inviare in determinati canali a cui i destinatari dovranno iscriversi.

Dovrà essere in grado, attraverso la lettura di particolari METADATI_G, di reindirizzare i messaggi ricevuti al destinatario più appropriato. È anche richiesto di creare un canale specifico per gestire le particolari esigenze dell'azienda. Questo viene chiamato Gestore Personale e permette agli utenti di poter personalizzare su quale piattaforma di messaggistica e da quali Topic specifici ricevere i messaggi.

1.4 Riferimenti

1.4.1 Normativi

1. *NormeDiProgetto v4.0.0_D*
2. CAPITOLATO_G d'appalto C1
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C1.pdf>

1.4.2 Informativi

1. Presentazione capitolato C1
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C1p.pdf>
2. SLIDE_G del corso di Ingegneria del Software
 - Analisi dei requisiti
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L08.pdf>



- Diagrammi dei CASI D'USO_G
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/E05b.pdf>
- 3. WEBHOOK_G di Redmine
<http://www.redmine.org/projects/redmine/wiki/Hooks>
- 4. Webhook di GitLab
<https://docs.gitlab.com/ee/user/project/integrations/Webhooks.html>
- 5. Bot di Telegram
<https://core.telegram.org/bots>
- 6. RANCHER_G:
<https://rancher.com/docs/>

2 Descrizione del prodotto

2.1 Funzionalità

Le funzionalità offerte dal prodotto sono:

- Ricezione delle segnalazioni provenienti da Redmine e GitLab
- Gestione dei Topic attraverso l'utilizzo del Broker Apache Kafka
- Inoltro dei messaggi verso Telegram ed Email
- Personalizzazione da parte dell'utente finale di impostazioni riguardanti le segnalazioni interessate

2.2 Caratteristiche degli utenti

Gli utenti che useranno il prodotto saranno team di sviluppatori software che lavorano abitualmente usando gli strumenti per realizzare CI/CD_G. Tramite Butterfly, essi potranno configurare la PIPELINE_G per lo sviluppo del proprio progetto, in modo da ottenere le notifiche in tempo reale direttamente sulle piattaforme di messaggistica selezionate. Nel caso in cui la persona non sia reperibile, verrà effettuata una segnalazione in un apposito calendario sul quale il sistema farà riferimento per l'inoltro alla seconda persona più appropriata a riceverla all'interno del contesto lavorativo.

2.3 Tecnologie e scelte relative al progetto

L'obiettivo di questo paragrafo è descrivere in maniera specifica come Butterfly andrà a interfacciarsi con le tecnologie.

Per ciascuna di queste è previsto un MICROSERVIZIO_G, il quale avrà come scopo quello di fare da tramite tra lo strumento che genera il messaggio e quello che lo riceve.

Questo pattern si chiama PUBLISHER / SUBSCRIBER_G e utilizza uno strumento intermedio detto Broker per lo smistamento dei messaggi e la gestione dei flussi.

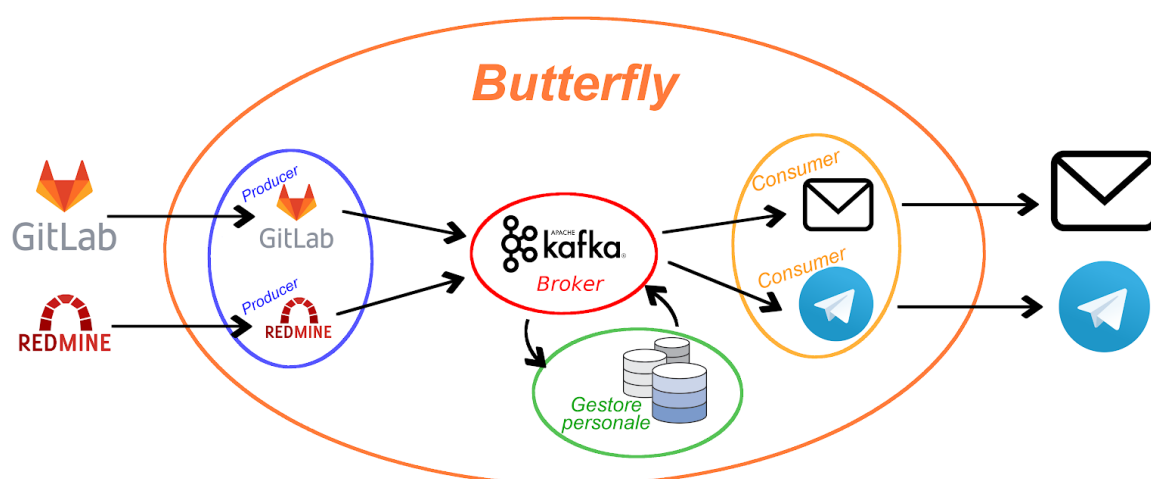


Figura 1: Visione generale del sistema Butterfly

L'immagine precedente rappresenta una suddivisione del sistema in quattro sezioni principali:

- **Arancione:** Butterfly, il sistema nella sua interezza.
- **Blu:** I Producer di GitLab e Redmine.
- **Rosso:** Il Broker. Per Butterfly useremo Apache Kafka.
- **Verde:** Il Gestore Personale.
- **Giallo:** I Consumer di Telegram ed Email

Questa scomposizione ci ha permesso di analizzare più approfonditamente i vari sottosistemi del prodotto in rapporto alle tecnologie esterne a Butterfly (GitLab, Redmine, Telegram ed Email), ai microservizi interni come i Producer / Consumer associati e il Broker.

2.3.1 Producer

Per ciascuno degli strumenti che invia messaggi verso il sistema è necessario creare una componente applicativa di tipo Producer che li riceva e li elabori inoltrandoli successivamente verso il Broker. Le tecnologie dalle quali vengono ricevuti i messaggi sono elencate di seguito in ordine di priorità in base a quanto richiesto dalla PROPONENTE_G.

1. Redmine
2. GitLab
3. SonarQube

Abbiamo scelto di sviluppare i Producer per le prime due applicazioni in base alle priorità suggerite da Imola Informatica nel capitolato, lasciando l'applicativo relativo a SonarQube come opzionale. Ciascun messaggio ricevuto da queste tecnologie verrà analizzato dal Producer associato in modo da inserirli nel Topic appropriato.

2.3.1.1 Producer GitLab

Gli eventi relativi alle REPOSITORY_G di GitLab presi in considerazione sono:

- Push
- ISSUE_G, che verranno suddivisi in:
 - Apertura issue
 - Chiusura issue

Le segnalazioni sono generate in base alle KEYWORD_G contenute nei messaggi di commit e nelle etichette assegnate alle issue. Da quest'ultime verrà creato il Topic corrispondente in caso non sia già esistente.

2.3.1.2 Producer Redmine

Gli eventi relativi ai progetti di Redmine presi in considerazione, sono:

- Issue, che verranno suddivisi in:
 - Apertura issue
 - Chiusura issue

2.3.2 Apache Kafka

Il ruolo di Apache Kafka è quello di restare in ascolto dei messaggi provenienti dai Producer, e smistarli in determinati Topic, in base al contenuto degli stessi. Il Gestore Personale potrà poi reperire i messaggi dai Topic specifici e, in base ai Topic ai quali gli utenti sono abbonati, immettere quei messaggi rielaborati alle code specifiche del Broker relative ai Consumer, che si occuperanno di inoltrare il messaggio all'utente finale.

2.3.3 Consumer

Come per gli strumenti precedentemente elencati, anche per quelli su cui il messaggio andrà ad essere inoltrato, è necessaria la creazione di un microservizio in grado di fare da tramite tra Broker e `CLIENTG` dello strumento di messaggistica scelto dall'utente. Le tecnologie verso le quali vengono inoltrati i messaggi rielaborati dal sistema sono elencate di seguito in ordine di priorità in base a quanto richiesto dalla proponente.

1. Telegram
2. Email
3. Slack

Allo stesso modo dei Producer, abbiamo scelto di sviluppare i Consumer per le prime due applicazioni in base alle priorità suggeriteci da Imola Informatica nel capitolato, lasciando l'applicativo relativo a Slack come opzionale.

2.3.3.1 Consumer Telegram

Il Consumer Telegram ha il ruolo di restare in ascolto del Topic specifico per Telegram di Apache Kafka e, usando come tramite un bot Telegram dedicato, inoltrare i messaggi agli utenti che hanno impostato Telegram come piattaforma di messaggistica preferita.

Questo richiederà uno step aggiuntivo per gli utenti, che dovranno mettersi in contatto con il bot la prima volta in modo che quest'ultimo abbia il permesso di inviare loro i messaggi tramite la piattaforma.

2.3.3.2 Consumer Email

Il Consumer Email ha il ruolo di restare in ascolto del Topic specifico per le Email di Apache Kafka e, interfacciandosi con un server e-mail dedicato, inoltrare i messaggi agli utenti finali che hanno impostato Email come preferenza per la ricezione.

2.3.4 Gestore Personale

Il Gestore Personale è quel `COMPONENTEG` che permette agli utenti di impostare le proprie preferenze e i dati personali tramite interfaccia web, installata in un server dell'azienda.

Questo è accessibile dai dipendenti interessati, i quali potranno quindi modificare, aggiungere e rimuovere, determinate impostazioni riguardanti le proprie preferenze. Quest'ultime sono:

- Scelta dell'applicativo sul quale ricevere le notifiche (Telegram / Email)
- Iscrizione / disiscrizione a determinati Topic
- Aggiunta / rimozione di determinate keyword per i commit di GitLab
- Selezione dei giorni di calendario in cui l'utente segnala la sua assenza
- Aggiunta / rimozione / modifica utenti

- Aggiunta e rimozione progetti di interesse
- Modifica priorità dei progetti

Nell'aggiunta o modifica di un utente i campi da poter inserire sono:

- Nome
- Cognome
- Contatto Email
- Contatto Telegram

Per comprendere meglio il sistema di notifica e inoltro delle segnalazioni, bisogna tener presente che ad una segnalazione fanno sempre riferimento:

- Uno o più label, nel caso di issue, o ad una o più keyword nel caso di commit
- Un progetto di GitLab o Redmine

La scelta di poter selezionare i progetti di interesse e la loro priorità permette ad un utente di non dover ricevere le segnalazioni di un certo tipo legate a tutti i progetti, ma solo a quelli in cui si lavora. Inserendo i vari progetti, nel momento di iscrizione ad un Topic, verranno visualizzati solo i Topic esistenti all'interno dei progetti selezionati.

L'inserimento della priorità per progetto serve per poter notificare solo gli utenti con la priorità più alta, in modo tale da poter suddividere le segnalazioni tra quelle di primaria importanza e quelle secondarie.

Un utente può ricevere delle segnalazioni se è iscritto ai progetti ad esse relativi. In particolare, gli utenti più indicati per la ricezione di una notifica sono quelli che, oltre ad essere interessati ai progetti, sono anche interessati ai Topic legati alla segnalazione.

2.3.4.1 Inoltro notifiche

Nel caso in cui un utente debba ricevere una segnalazione ma non si riveli essere disponibile, l'inoltro della notifica deve funzionare in tale modo:

- Se è presente almeno un altro utente iscritto al Topic legato alla segnalazione, disponibile il giorno dell'invio della notifica e con la stessa priorità di progetto, non viene inviata la notifica all'utente in questione perché almeno un altro utente riceverà la segnalazione
- Se non è disponibile nessun altro utente con la stessa priorità iscritto al Topic legato alla segnalazione, la notifica viene inoltrata ad un utente sempre iscritto allo stesso Topic e disponibile il giorno dell'invio della segnalazione, ma con una priorità di progetto inferiore
- Se non è disponibile alcun utente iscritto al Topic legato alla segnalazione e disponibile il giorno dell'invio della notifica, verrà notificata una persona iscritta al progetto indicato con priorità massima tra quelli disponibili

2.3.4.2 Informazioni aggiuntive

Dato che il nostro sistema di accesso non consiste in un login con username e password, gli utenti possono essere aggiunti al sistema di Butterfly solo da un altro utente. Al primo utilizzo dell'applicazione sarà perciò già presente un utente, di cui verrà rilasciato l'identificativo, per poter aggiungere di seguito gli altri utenti che utilizzeranno Butterfly.

3 Casi d'uso

Questa sezione elenca le funzionalità offerte da Butterfly descritte attraverso il linguaggio UML_G. Butterfly può essere visto come l'insieme di più sottosistemi che verranno in seguito elencati e che sono stati descritti in modo molto generale anche attraverso la Figura 1.

Questa mostra in maniera un po' più chiarificante la suddivisione e ne facilita l'analisi per la stesura dei casi d'uso. Abbiamo quindi come attori non soltanto le applicazioni che mandano messaggi al sistema, ma anche componenti quali i vari Producer e Consumer che interagiscono col Gestore Personale.

All'interno dei casi d'uso il termine "sistema" è inteso come l'intera applicazione Butterfly, mentre "piattaforma di messaggistica" fa sempre riferimento a Telegram o Email e con "indisponibilità" e "irreperibilità" di una persona si intendono quei giorni in cui una persona non è raggiungibile nè disponibile per proseguire con le proprie mansioni e quindi non vuole ricevere notifiche.

Il termine "identificativo" fa riferimento indifferentemente al contatto Email o all'ID Telegram. In ogni momento ogni utente deve garantire che almeno uno degli identificativi (Email o Telegram) sia configurato.

Inizialmente è già inserito manualmente al database un utente predefinito per poter inserire successivamente gli altri utenti.

3.1 Attori

- Redmine
- GitLab
- Producer Redmine
- Producer GitLab
- Utente non acceduto
- Utente non registrato
- Utente (inteso come acceduto e registrato nel sistema)
- Consumer Telegram
- Consumer Email
- Telegram
- Email

3.2 Sistemi e sottosistemi

Per indicare i vari sottosistemi nei codici dei casi d'uso sono allegati delle sigle per rendere più chiara la posizione del caso d'uso all'interno di Butterfly.

- **PR**: Producer Redmine.
- **PG**: Producer GitLab.
- **GP**: Gestore Personale.
- **CT**: Consumer Telegram.
- **CE**: Consumer Email.



- **BT**: bot Telegram.
- **SE**: server Email.

3.3 Elenco casi d'uso

3.3.1 UC1-PR - Redmine invia segnalazione al Producer Redmine

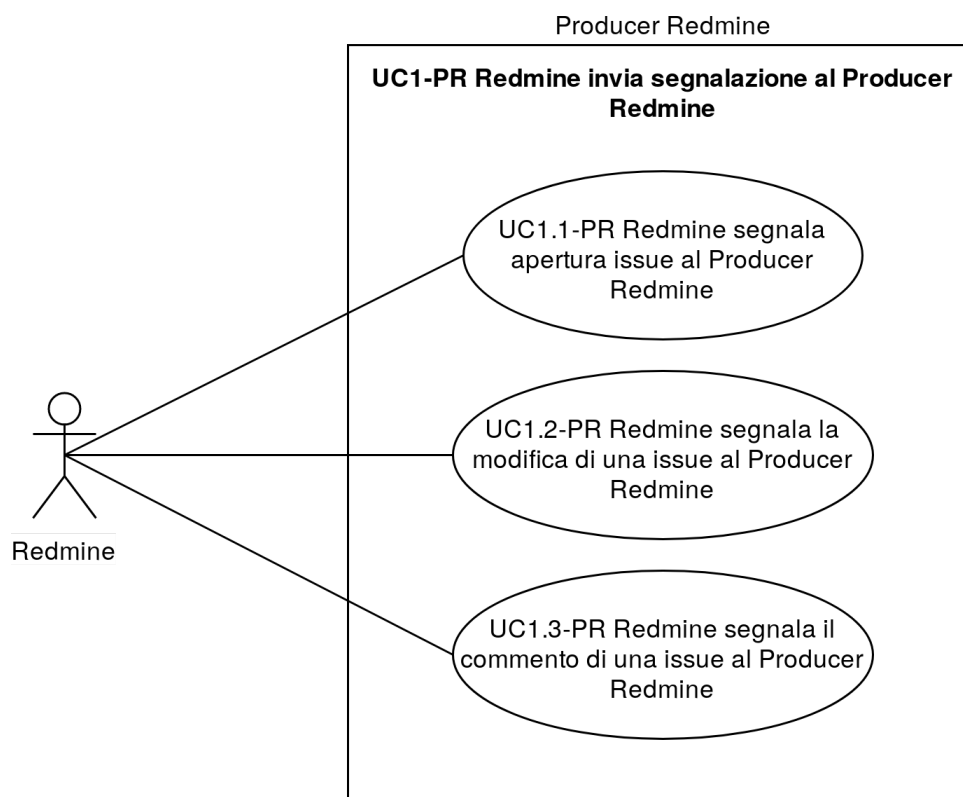


Figura 2: UC1-PR - Redmine invia segnalazione al Producer Redmine

- **Codice:** UC1-PR.
- **Titolo:** Redmine invia segnalazione al Producer Redmine.
- **Attori primari:** Redmine.
- **Descrizione:** Redmine invia a Butterfly una segnalazione.
- **Precondizione:** su Redmine viene eseguita un'operazione che scatuisce una segnalazione da inviare a Butterfly.
- **Postcondizione:** il Producer Redmine riceve la segnalazione da Redmine.
- **Scenario principale:**
 1. Viene eseguita un'operazione in Redmine da far scaturire l'invio di una segnalazione
 2. Redmine procede all'invio della segnalazione al Producer Redmine
 3. La segnalazione viene ricevuta dal Producer Redmine

3.3.2 UC1.1-PR - Redmine segnala apertura issue al Producer Redmine

- **Codice:** UC1.1-PR.
- **Titolo:** Redmine segnala apertura issue al Producer Redmine.

- **Attori primari:** Redmine.
- **Descrizione:** Redmine segnala a Butterfly l'apertura di una nuova issue tramite webhook. L'apertura di una issue in un particolare progetto su Redmine contiene i seguenti campi di interesse:
 - Object kind
 - Action
 - Project
 - Description
 - Author
 - Title
 - Label
 - Assignees

Il campo action conterrà al suo interno "opened".

- **Precondizione:** su Redmine viene eseguita un'operazione che scatuisce una segnalazione da inviare a Butterfly.
- **Postcondizione:** il Producer Redmine riceve la segnalazione di apertura issue da Redmine.
- **Scenario principale:**
 1. Viene aperta una nuova issue su Redmine compilando i campi indicati
 2. Redmine procede all'invio della segnalazione di issue al Producer Redmine
 3. Il Producer Redmine riceve la segnalazione di apertura issue

3.3.3 UC1.2-PR - Redmine segnala la modifica di una issue al Producer Redmine

- **Codice:** UC1.2-PR.
- **Titolo:** Redmine segnala la modifica di una issue al Producer Redmine.
- **Attori primari:** Redmine.
- **Descrizione:** quando una issue viene modificata, l'invio di tale segnalazione avviene da parte di Redmine tramite webhook. I campi di interesse sono gli stessi dell'apertura di una issue, con la differenza che necessariamente il campo action contiene ora "updated".
- **Precondizione:** su Redmine viene eseguita un'operazione che scatuisce una segnalazione da inviare a Butterfly.
- **Postcondizione:** il Producer Redmine riceve la segnalazione di modifica issue da Redmine.
- **Scenario principale:**
 1. Viene modificata una issue già esistente su Redmine
 2. Redmine procede all'invio della segnalazione di modifica issue al Producer Redmine
 3. Il Producer Redmine riceve la segnalazione di modifica issue

3.3.4 UC1.3-PR - Redmine segnala il commento di una issue al Producer Redmine

- **Codice:** UC1.3-PR.
- **Titolo:** Redmine segnala il commento di una issue al Producer Redmine.
- **Attori primari:** Redmine.
- **Descrizione:** quando una issue viene commentata, l'invio di tale segnalazione avviene da parte di Redmine tramite webhook. I campi di interesse sono gli stessi dell'apertura di una issue, con la differenza che necessariamente il campo action contiene ora "updated".
- **Precondizione:** su Redmine viene eseguita un'operazione che scatuisce una segnalazione da inviare a Butterfly.
- **Postcondizione:** il Producer Redmine riceve la segnalazione del commento di una issue da Redmine.
- **Scenario principale:**
 1. Viene commentata una issue già esistente su Redmine
 2. Redmine procede all'invio della segnalazione di commento issue al Producer Redmine
 3. Il Producer Redmine riceve la segnalazione di commento in una issue

3.3.5 UC2-PG - GitLab invia segnalazione al Producer GitLab

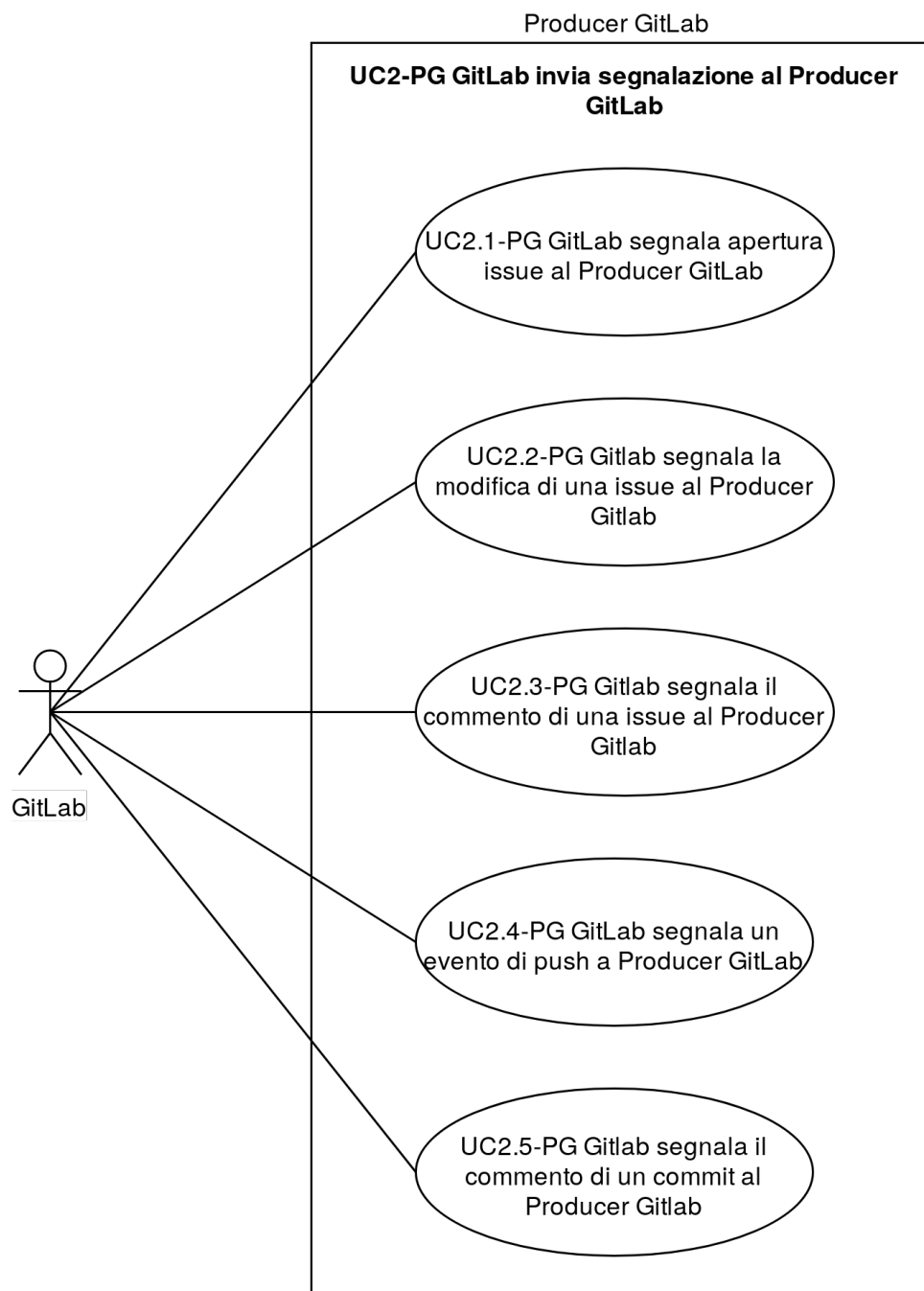


Figura 3: UC2-PG - GitLab invia segnalazione al Producer GitLab

- **Codice:** UC2-PG.
- **Titolo:** GitLab invia segnalazione al Producer GitLab.
- **Attori primari:** GitLab.
- **Descrizione:** GitLab invia una segnalazione tramite webhook a Butterfly.
- **Precondizione:** su GitLab viene eseguita una operazione che scaturisce una segnalazione da inviare a Butterfly.

- **Postcondizione:** il Producer GitLab riceve la segnalazione da GitLab.
- **Scenario principale:**
 1. Viene eseguita un'operazione
 2. GitLab procede all'invio della segnalazione al Producer GitLab
 3. Il Producer GitLab riceve la segnalazione

3.3.6 UC2.1-PG - GitLab segnala apertura issue al Producer GitLab

- **Codice:** UC2.1-PG.
- **Titolo:** GitLab segnala apertura issue al Producer GitLab.
- **Attori primari:** GitLab.
- **Descrizione:** GitLab segnala l'apertura di una nuova issue tramite webhook a Butterfly. L'apertura di una issue su GitLab contiene:
 - Object kind
 - Action
 - Project
 - Title e opzionalmente:
 - * Label
 - * Milestone
 - * Assignees
 - * Due Date

Il campo object kind ci permette di capire il tipo di oggetto. In questo caso infatti contiene "issue", mentre il campo action ha "open".

- **Precondizione:** su GitLab viene eseguita una operazione che scaturisce una segnalazione da inviare a Butterfly.
- **Postcondizione:** il Producer GitLab riceve la segnalazione di apertura issue da GitLab.
- **Scenario principale:**
 1. Viene aperta una nuova issue su GitLab
 2. GitLab procede all'invio della segnalazione di issue al Producer GitLab
 3. Il Producer GitLab riceve la segnalazione di apertura issue

3.3.7 UC2.2-PG - Gitlab segnala la modifica di una issue al Producer Gitlab

- **Codice:** UC2.2-PG.
- **Titolo:** Gitlab segnala la modifica di una issue al Producer Gitlab.
- **Attori primari:** GitLab.
- **Descrizione:** GitLab segnala la modifica di una issue esistente tramite webhook a Butterfly. I campi di interesse sono gli stessi dell'apertura issue, con la differenza che il campo action contiene "update".

- **Precondizione:** su GitLab viene eseguita una operazione che scaturisce una segnalazione da inviare a Butterfly.
- **Postcondizione:** il Producer GitLab riceve la segnalazione di modifica issue da GitLab.
- **Scenario principale:**
 1. Viene modificata una issue già esistente su GitLab
 2. GitLab procede all'invio della segnalazione di modifica issue al Producer GitLab
 3. Il Producer GitLab riceve la segnalazione di modifica issue

3.3.8 UC2.3-PG - Gitlab segnala il commento di una issue al Producer Gitlab

- **Codice:** UC2.3-PG.
- **Titolo:** Gitlab segnala il commento di una issue al Producer Gitlab.
- **Attori primari:** GitLab.
- **Descrizione:** GitLab segnala il commento di una issue esistente tramite webhook a Butterfly. I campi di interesse sono gli stessi dell'apertura issue, con la differenza che il campo action contiene "issue-note".
- **Precondizione:** su GitLab viene eseguita una operazione che scaturisce una segnalazione da inviare a Butterfly.
- **Postcondizione:** il Producer GitLab riceve la segnalazione di commento issue da GitLab.
- **Scenario principale:**
 1. Viene commentata una issue già esistente su GitLab
 2. GitLab procede all'invio della segnalazione di commento issue al Producer GitLab
 3. Il Producer GitLab riceve la segnalazione di commento di una issue

3.3.9 UC2.4-PG - GitLab segnala un evento di push a Producer GitLab

- **Codice:** UC2.4-PG.
- **Titolo:** GitLab segnala un evento di push a Producer GitLab.
- **Attori primari:** GitLab.
- **Descrizione:** GitLab segnala un evento di push tramite webhook a Butterfly. L'evento di push può essere composto da uno o più commit. I campi di interesse sono:
 - Object kind
 - Project
 - Repository
 - Commits

E in questo caso il campo object kind contiene "push".

- **Precondizione:** su GitLab viene eseguita una operazione che scaturisce una segnalazione da inviare a Butterfly.
- **Postcondizione:** il Producer GitLab riceve la segnalazione di push da GitLab.

- **Scenario principale:**

1. Viene effettuato un push in GitLab
2. GitLab procede all'invio della segnalazione di push al Producer GitLab
3. Il Producer GitLab riceve la segnalazione dell'evento di push

3.3.10 UC2.5-PG - Gitlab segnala il commento di un commit al Producer Gitlab

- **Codice:** UC2.5-PG.

- **Titolo:** Gitlab segnala il commento di un commit al Producer Gitlab.

- **Attori primari:** GitLab.

- **Descrizione:** GitLab segnala il commento di un commit esistente tramite webhook a Butterfly. I campi di interesse sono gli stessi dell'apertura issue, con la differenza che il campo action contiene "commit-note".

- **Precondizione:** su GitLab viene eseguita una operazione che scatuisce una segnalazione da inviare a Butterfly.

- **Postcondizione:** il Producer GitLab riceve la segnalazione di commento di commit da GitLab.

- **Scenario principale:**

1. Viene commentato un commit già esistente su GitLab
2. GitLab procede all'invio della segnalazione di commento commit al Producer GitLab
3. Il Producer GitLab riceve la segnalazione di commento di un commit

3.3.11 UC3-GP - Producer Redmine invia messaggio al Gestore Personale

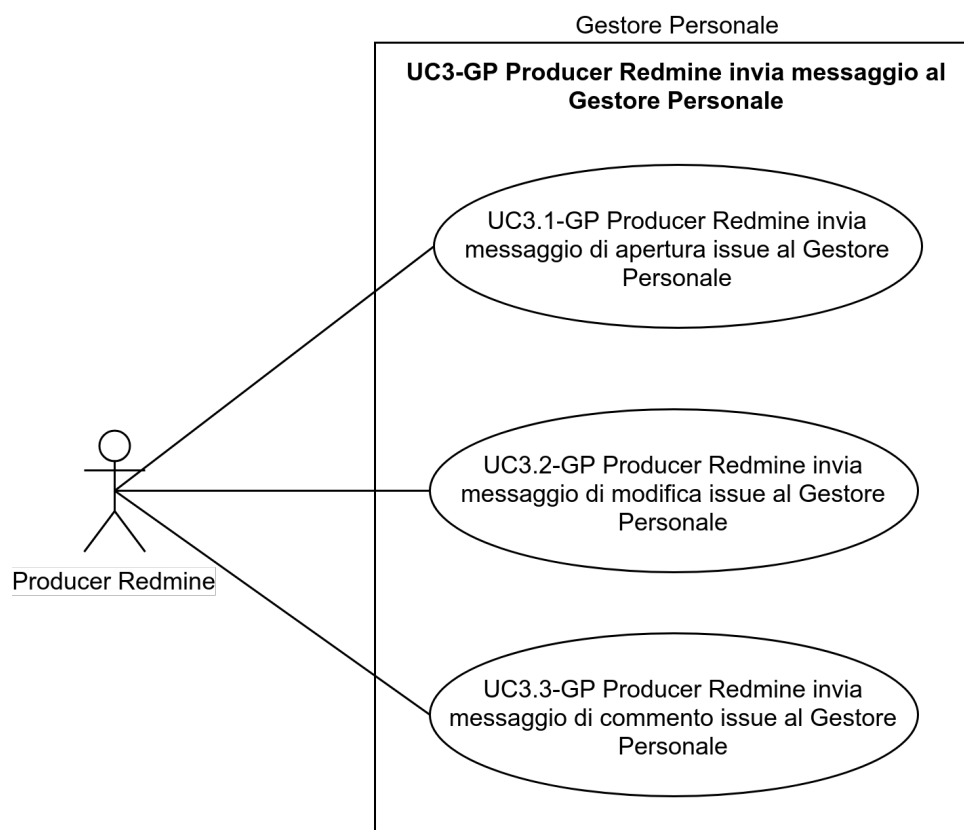


Figura 4: UC3-GP - Producer Redmine invia messaggio al Gestore Personale

- **Codice:** UC3-GP.
- **Titolo:** Producer Redmine invia messaggio al Gestore Personale.
- **Attori Pimari:** Producer Redmine.
- **Descrizione:** il Producer Redmine, dopo aver ricevuto una segnalazione da Redmine, elabora il messaggio e lo invia al Gestore Personale.
Il messaggio elaborato conterrà i campi:
 - App
 - Object_kind
 - Title
 - Project_id
 - Project_name
 - Author
 - Assignees
 - Action
 - Description
 - Labels
 - Update

Se il Producer Redmine riceve una segnalazione che non è in grado di analizzare, questa viene scartata.

- **Precondizione:** il Producer Redmine ha ricevuto una segnalazione da Redmine.
- **Postcondizione:** il Producer Redmine ha elaborato e inviato al Gestore Personale il messaggio.
- **Scenario principale:**
 1. Il Producer Redmine riceve la segnalazione di issue da Redmine
 2. Il Producer Redmine prepara il messaggio di issue in modo che venga inserito correttamente nel Gestore Personale
 3. Il Producer Redmine invia il messaggio di issue al Gestore Personale
 4. Il Gestore Personale riceve il messaggio dal Producer Redmine

3.3.12 UC3.1-GP - Producer Redmine invia messaggio di apertura issue al Gestore Personale

- **Codice:** UC3.1-GP.
- **Titolo:** Producer Redmine invia messaggio di apertura issue al Gestore Personale.
- **Attori Primari:** Producer Redmine.
- **Descrizione:** il Producer Redmine, dopo aver ricevuto una segnalazione di apertura issue da Redmine, elabora il messaggio e lo invia al Gestore Personale.
- **Precondizione:** il Producer Redmine ha ricevuto una segnalazione da Redmine.
- **Postcondizione:** il Producer Redmine ha elaborato e inviato al Gestore Personale il messaggio di apertura issue.
- **Scenario Principale:**
 1. Il Producer Redmine riceve la segnalazione di apertura issue da Redmine
 2. Il Producer Redmine prepara il messaggio in modo che venga inserito correttamente nel Gestore Personale
 3. Il Producer Redmine invia il messaggio di apertura issue al Gestore Personale
 4. Il Gestore Personale riceve il messaggio di apertura issue

3.3.13 UC3.2-GP - Producer Redmine invia messaggio di modifica issue al Gestore Personale

- **Codice:** UC3.2-GP.
- **Titolo:** Producer Redmine invia messaggio di modifica issue al Gestore Personale
- **Attori Primari:** Producer Redmine.
- **Descrizione:** il Producer Redmine, dopo aver ricevuto una segnalazione di modifica issue da Redmine, elabora il messaggio e lo invia al Gestore Personale.
- **Precondizione:** il Producer Redmine ha ricevuto una segnalazione da Redmine.
- **Postcondizione:** il Producer Redmine ha elaborato e inviato al Gestore Personale il messaggio di modifica issue.

- **Scenario Principale:**

1. Il Producer Redmine riceve la segnalazione di modifica issue da Redmine
2. Il Producer Redmine prepara il messaggio in modo che venga inserito correttamente nel Gestore Personale
3. Il Producer Redmine invia il messaggio di modifica issue al Gestore Personale
4. Il Gestore Personale riceve il messaggio di modifica issue

3.3.14 UC3.3-GP - Producer Redmine invia messaggio di commento issue al Gestore Personale

- **Codice:** UC3.3-GP.

- **Titolo:** Producer Redmine invia messaggio di commento issue al Gestore Personale

- **Attori Primari:** Producer Redmine.

- **Descrizione:** il Producer Redmine, dopo aver ricevuto una segnalazione di commento issue da Redmine, elabora il messaggio e lo invia al Gestore Personale.

- **Precondizione:** il Producer Redmine ha ricevuto una segnalazione da Redmine.

- **Postcondizione:** il Producer Redmine ha elaborato e inviato al Gestore Personale il messaggio di commento issue.

- **Scenario Principale:**

1. Il Producer Redmine riceve la segnalazione di commento issue da Redmine
2. Il Producer Redmine prepara il messaggio in modo che venga inserito correttamente nel Gestore Personale
3. Il Producer Redmine invia il messaggio di commento issue al Gestore Personale
4. Il Gestore Personale riceve il messaggio di commento issue

3.3.15 UC4-GP - Producer GitLab invia messaggio al Gestore Personale

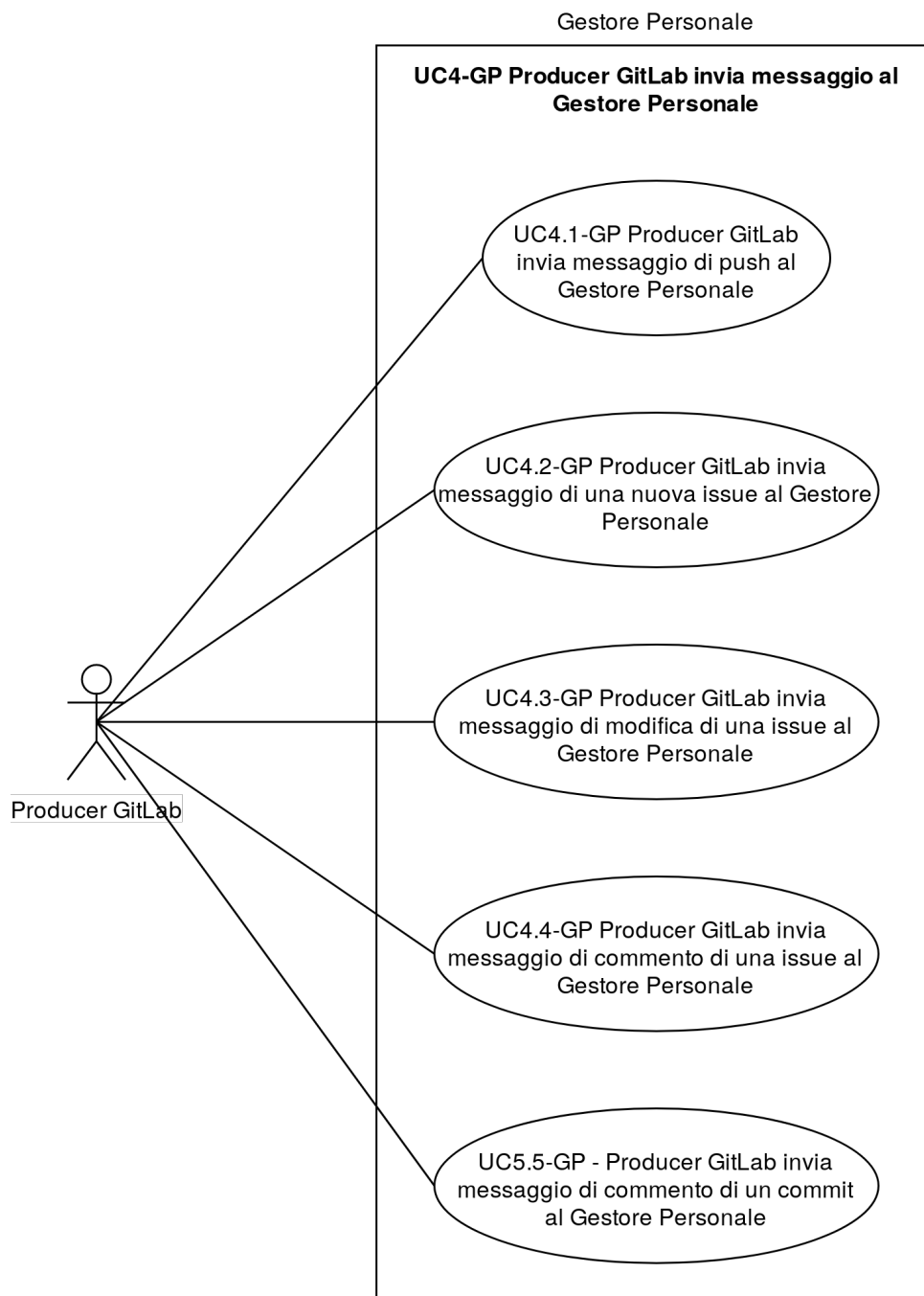


Figura 5: UC4-GP - Producer GitLab invia messaggio al Gestore Personale

- **Codice:** UC4-GP.
- **Titolo:** Producer GitLab invia messaggio al Gestore Personale.
- **Attori primari:** Producer GitLab.
- **Descrizione:** il Producer GitLab, dopo aver ricevuto una segnalazione da GitLab, elabora un messaggio da inviare al Gestore Personale.
Il messaggio elaborato conterrà i campi:

- App
- Object_kind
- Title
- Project_id
- Project_name
- Author
- Assignees
- Action
- Description
- Labels
- Update

Se il Producer Gitlab riceve una segnalazione che non è in grado di analizzare, questa viene scartata.

- **Precondizione:** il Producer GitLab ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer GitLab ha inviato al Gestore Personale il messaggio elaborato.
- **Scenario principale:**
 1. Il Producer GitLab riceve una segnalazione da GitLab
 2. Il Producer GitLab prepara il messaggio in modo che venga inserito correttamente nel Gestore Personale
 3. Il Poducer GitLab invia il messaggio al Gestore Personale
 4. Il Gestore Perosonale riceve il messaggio dal Producer GitLab

3.3.16 UC4.1-GP - Producer GitLab invia messaggio di push al Gestore Personale

- **Codice:** UC4.1-GP.
- **Titolo:** Producer GitLab invia uno o più messaggi di commit al Gestore Personale.
- **Attori primari:** Producer GitLab.
- **Descrizione:** il Producer GitLab, dopo aver ricevuto una segnalazione di push da GitLab, suddivide la segnalazione per i commit in più messaggi che verranno catalogati sotto il Topic “commits”.
- **Precondizione:** il Producer GitLab ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer GitLab ha inviato uno o più messaggi di commit.
- **Scenario principale:**
 1. Il Producer GitLab riceve la segnalazione di un push da GitLab
 2. Il Producer GitLab prepara i messaggi in modo che vengano inseriti correttamente nel Gestore Personale
 3. Il Producer GitLab invia i messaggi di commit al Gestore Personale
 4. Il Gestore Personale riceve i messaggi di commit

3.3.17 UC4.2-GP - Producer GitLab invia messaggio di una nuova issue al Gestore Personale

- **Codice:** UC4.2-GP.
- **Titolo:** Producer GitLab invia messaggio di una nuova issue al Gestore Personale.
- **Attori primari:** Producer GitLab.
- **Descrizione:** il Producer GitLab, dopo aver ricevuto una segnalazione di una nuova issue da GitLab, elabora il messaggio da inoltrare al Gestore Personale.
- **Precondizione:** il Producer GitLab ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer GitLab ha inviato al Gestore Personale il messaggio elaborato di nuova issue.
- **Scenario principale:**
 1. Il Producer GitLab riceve la segnalazione di una nuova issue da GitLab
 2. Il Producer GitLab prepara il messaggio di una nuova issue in modo che venga inserito correttamente nel Gestore Personale
 3. Il Producer GitLab invia il messaggio di una nuova issue al Gestore Personale
 4. Il Gestore Personale riceve il messaggio di apertura issue dal Producer GitLab

3.3.18 UC4.3-GP - Producer GitLab invia messaggio di modifica di una issue al Gestore Personale

- **Codice:** UC4.3-GP.
- **Titolo:** Producer GitLab invia messaggio di modifica di una issue al Gestore Personale.
- **Attori primari:** Producer GitLab.
- **Descrizione:** il Producer GitLab, dopo aver ricevuto una segnalazione di una modifica issue da GitLab, elabora il messaggio da inoltrare al Gestore Personale.
- **Precondizione:** il Producer GitLab ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer GitLab ha inviato al Gestore Personale il messaggio elaborato di modifica issue.
- **Scenario principale:**
 1. Il Producer GitLab riceve la segnalazione di issue da GitLab
 2. Il Producer GitLab prepara il messaggio di modifica issue in modo che venga inserito correttamente nel Gestore Personale
 3. Il Producer GitLab invia il messaggio di issue al Gestore Personale
 4. Il Gestore Personale riceve il messaggio di modifica issue dal Producer GitLab

3.3.19 UC4.4-GP - Producer GitLab invia messaggio di commento di una issue al Gestore Personale

- **Codice:** UC4.4-GP.
- **Titolo:** Producer GitLab invia messaggio di commento di una issue al Gestore Personale.
- **Attori primari:** Producer GitLab.
- **Descrizione:** il Producer GitLab, dopo aver ricevuto una segnalazione di commento di una issue da GitLab, elabora il messaggio da inoltrare al Gestore Personale.
- **Precondizione:** il Producer GitLab ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer GitLab ha inviato al Gestore Personale il messaggio elaborato di commento issue.
- **Scenario principale:**
 1. Il Producer GitLab riceve la segnalazione di commento issue da GitLab
 2. Il Producer GitLab prepara il messaggio di commento issue in modo che venga inserito correttamente nel Gestore Personale
 3. Il Producer GitLab invia il messaggio di commento issue al Gestore Personale
 4. Il Gestore Personale riceve il messaggio di commento issue dal Producer GitLab

3.3.20 UC4.5-GP - Producer GitLab invia messaggio di commento di un commit al Gestore Personale

- **Codice:** UC4.5-GP.
- **Titolo:** Producer GitLab invia messaggio di commento di un commit al Gestore Personale.
- **Attori primari:** Producer GitLab.
- **Descrizione:** il Producer GitLab, dopo aver ricevuto una segnalazione di commento di un commit da GitLab, elabora il messaggio da inoltrare al Gestore Personale.
- **Precondizione:** il Producer GitLab ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer GitLab ha inviato al Gestore Personale il messaggio elaborato di commento commit.
- **Scenario principale:**
 1. Il Producer GitLab riceve la segnalazione di commento commit da GitLab
 2. Il Producer GitLab prepara il messaggio di commento commit in modo che venga inserito correttamente nel Gestore Personale
 3. Il Producer GitLab invia il messaggio di commento commit al Gestore Personale
 4. Il Gestore Personale riceve il messaggio di commento issue dal Producer GitLab

3.3.21 UC5-CT - Gestore Personale invia il messaggio finale al Consumer Telegram

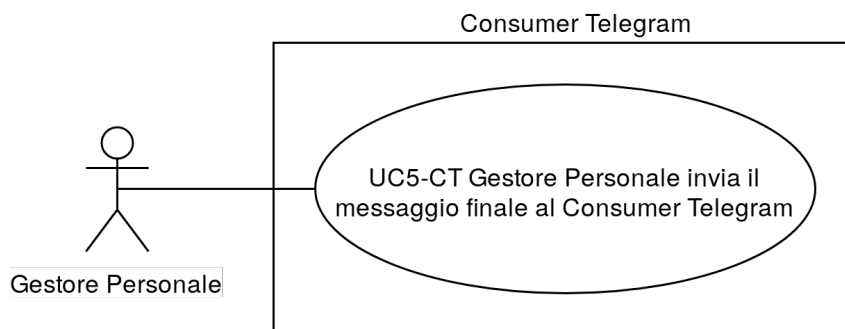


Figura 6: UC5-CT - Gestore Personale invia il messaggio finale al Consumer Telegram

- **Codice:** UC5-CT.
- **Titolo:** Gestore Personale invia il messaggio finale al Consumer Telegram.
- **Attori primari:** Gestore Personale.
- **Descrizione:** il Gestore Personale, dopo aver ricevuto il messaggio elaborato dal Producer Redmine o GitLab, controlla i Topic del messaggio, gli utenti iscritti, la loro disponibilità e se la loro preferenza è Telegram. Se tutte queste condizioni sono verificate, viene preparato il messaggio finale da inviare successivamente al Consumer Telegram. Se il destinatario è iscritto a quel Topic ma non è disponibile, si verifica se è presente almeno un altro utente iscritto allo stesso Topic e con la stessa priorità di progetto, in caso contrario la segnalazione viene inviata a chi è iscritto al Topic indicato e con una priorità del progetto più bassa. Se non è disponibile alcuna persona iscritta al Topic indicato, il messaggio verrà inviato alla persona con più alta priorità nel progetto.

Il messaggio finale, una volta elaborato, conterrà i campi:

- Id della chat del destinatario
- Applicazione di provenienza
- Ora di invio
- Tipo di segnalazione(commit, issue o commento)
- Project
- Topic
- Description
- Subject e opzionalmente
 - * Due date
 - * Milestone
 - * Assignee
- **Precondizione:** il Gestore Personale ha ricevuto il messaggio elaborato dal Producer Redmine o GitLab.
- **Postcondizione:** Il Gestore Personale ha inviato il messaggio finale al Consumer Telegram.
- **Scenario principale:**

1. Il Gestore Personale riceve un messaggio dal Producer Redmine o dal Producer GitLab

2. Il Gestore Personale valuta quali utenti sono iscritti al Topic del messaggio ricevuto e se vogliono ricevere il messaggio tramite Telegram
3. Il Gestore Personale procede all'invio del messaggio finale al Consumer Telegram
4. Il Consumer Telegram riceve il messaggio dal Gestore Personale

3.3.22 UC6-CE - Gestore Personale invia il messaggio finale al Consumer Email

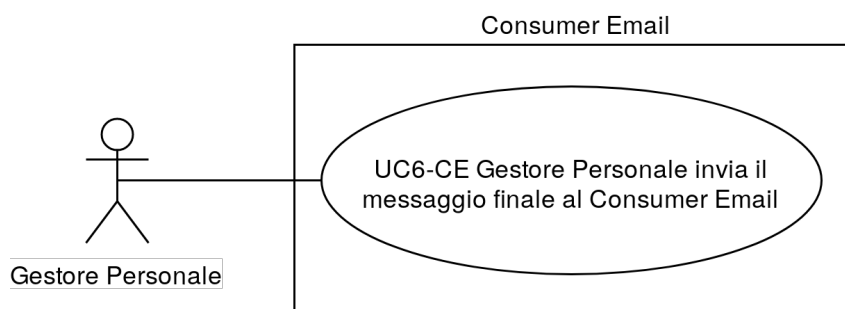


Figura 7: UC6-CE - Gestore Personale invia il messaggio finale al Consumer Email

- **Codice:** UC6-CE.
- **Titolo:** Gestore Personale invia il messaggio finale al Consumer Email.
- **Attori primari:** Gestore Personale.
- **Descrizione:** il Gestore Personale, dopo aver ricevuto il messaggio elaborato dal Producer Redmine o GitLab, controlla i Topic del messaggio, gli utenti iscritti, la loro disponibilità e se la loro preferenza è Email. Se tutte queste condizioni sono verificate, viene preparato il messaggio finale da inviare successivamente viene inviato al Consumer Email. Se il destinatario è iscritto a quel Topic ma non è disponibile, si verifica se è presente almeno un altro utente iscritto allo stesso Topic e con la stessa priorità di progetto, in caso contrario la segnalazione viene inviata a chi è iscritto al Topic indicato e con una priorità del progetto più bassa. Se non è disponibile alcuna persona iscritta al Topic indicato, il messaggio verrà inviato alla persona con più alta priorità nel progetto.

Il messaggio finale, una volta elaborato, conterrà i campi:

- Email del destinatario
 - Applicazione di provenienza
 - Ora di invio
 - Tipo di segnalazione(commit, issue o commento)
 - Project
 - Topic
 - Description
 - Subject e opzionalmente
 - * Due date
 - * Milestone
 - * Assignee
- **Precondizione:** il Gestore Personale ha ricevuto il messaggio elaborato dal Producer Redmine o GitLab.

- **Postcondizione:** Il Gestore Personale ha inviato il messaggio finale al Consumer Email.
- **Scenario principale:**
 1. Il Gestore Personale riceve un messaggio dal Producer Redmine o dal Producer GitLab
 2. Il Gestore Personale valuta quali utenti sono iscritti al Topic del messaggio ricevuto e se vogliono ricevere il messaggio tramite Email
 3. Il Gestore Personale procede all'invio del messaggio finale al Consumer Email
 4. Il Consumer Email riceve il messaggio dal Gestore Personale

3.3.23 UC7-BT - Consumer Telegram inoltra il messaggio finale al bot Telegram

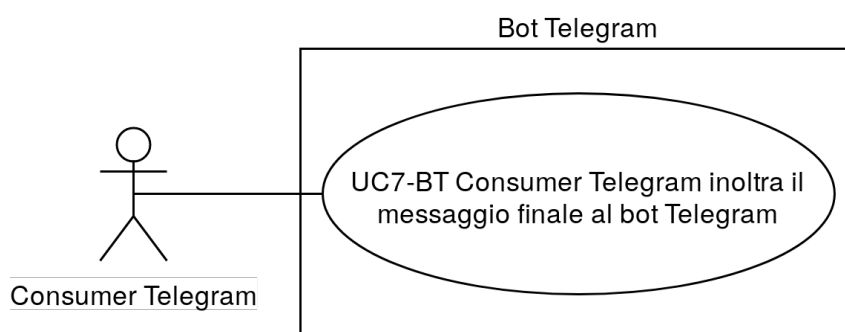


Figura 8: UC7-BT - Consumer Telegram inoltra il messaggio finale al bot Telegram

- **Codice:** UC7-BT.
- **Titolo:** Consumer Telegram inoltra il messaggio finale al bot Telegram.
- **Attori primari:** Consumer Telegram.
- **Descrizione:** il Consumer Telegram inoltra il messaggio finale al bot Telegram, il quale notifica il destinatario finale attraverso Telegram.
- **Precondizione:** il Consumer Telegram ha ricevuto almeno un messaggio.
- **Postcondizione:** il bot Telegram ha ricevuto il messaggio finale.
- **Scenario principale:**
 1. Il Consumer Telegram riceve un messaggio dal Gestore Personale
 2. Il Consumer Telegram inoltra il messaggio finale al bot Telegram
 3. Il bot Telegram riceve il messaggio dal Consumer Telegram

3.3.24 UC8-SE - Consumer Email inoltra il messaggio finale al server Email

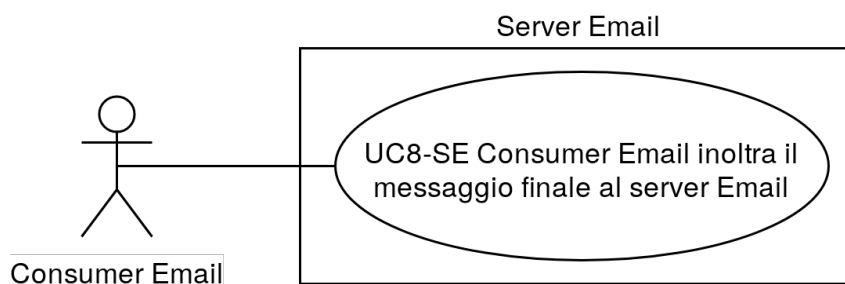


Figura 9: UC8-SE - Consumer Email inoltra il messaggio finale al server Email

- **Codice:** UC8-SE.
- **Titolo:** Consumer Email inoltra il messaggio finale al server Email.
- **Attori primari:** Consumer Email.
- **Descrizione:** il Consumer Email inoltra il messaggio finale al server Email, il quale notifica il destinatario finale attraverso una Email.
- **Precondizione:** il Consumer Email ha ricevuto almeno un messaggio.
- **Postcondizione:** il server Email ha ricevuto il messaggio finale.
- **Scenario principale:**
 1. Il Consumer Email riceve un messaggio dal Gestore Personale
 2. Il Consumer Email procede all'inoltro del messaggio finale al server Email
 3. Il server Email riceve il messaggio dal Consumer Email

3.3.25 UC9-GP - Accesso

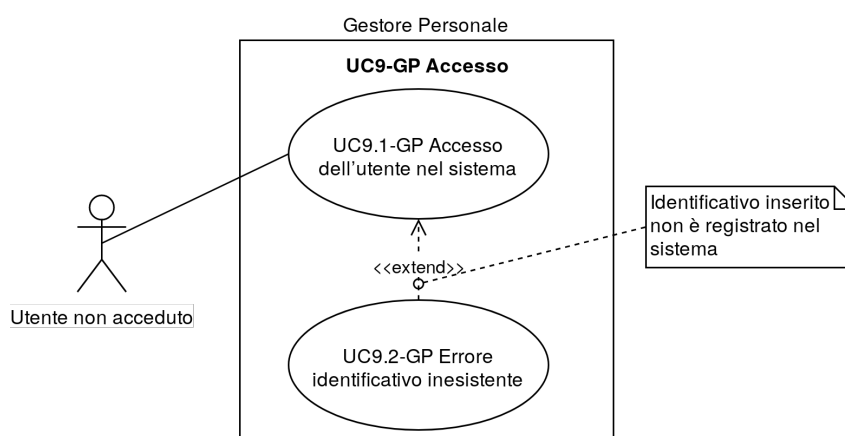


Figura 10: UC9-GP - Accesso

- **Codice:** UC9-GP.
- **Titolo:** accesso.

- **Attori primari:** utente non acceduto.
- **Descrizione:** l'utente non acceduto richiede di accedere al sistema attraverso l'inserimento del proprio identificativo.
- **Precondizione:** il sistema considera l'utilizzatore come un utente non acceduto.
- **Postcondizione:** il sistema riconosce l'utilizzatore di esso come utente acceduto.
- **Scenario principale:**
 1. L'utente non ancora riconosciuto dal sistema procede all'accesso inserendo il proprio identificativo

3.3.26 UC9.1-GP - Accesso dell'utente nel sistema

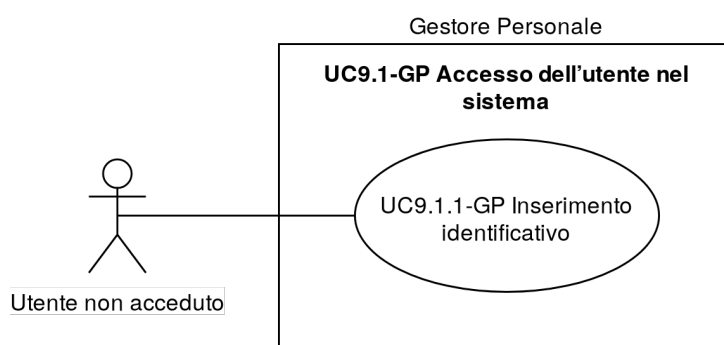


Figura 11: UC9.1-GP - Accesso dell'utente nel sistema

- **Codice:** UC9.1-GP.
- **Titolo:** accesso dell'utente nel sistema.
- **Attori primari:** utente non acceduto.
- **Descrizione:** l'utente non acceduto accede al sistema inserendo il proprio identificativo e inviandone la conferma.
- **Precondizione:** il sistema riconosce l'utilizzatore come un utente non acceduto.
- **Postcondizione:** il sistema riconosce l'utilizzatore di esso come utente acceduto.
- **Scenario principale:**
 1. L'utente non ancora riconosciuto dal sistema richiede l'accesso inserendo il proprio identificativo
 2. L'utente conferma l'invio dei dati
 3. L'utente è riconosciuto dal sistema
- **Estensioni:**
 1. L'accesso non va a buon fine e viene visualizzato un errore per avvisare l'utente [UC9.2-GP]

3.3.27 UC9.1.1-GP - Inserimento identificativo

- **Codice:** UC9.1.1-GP.
- **Titolo:** inserimento identificativo.
- **Attori primari:** utente non acceduto.
- **Descrizione:** l'utente inserisce l'identificativo.
- **Precondizione:** il sistema riconosce l'utilizzatore come un utente non acceduto.
- **Postcondizione:** l'utente ha inserito l'identificativo desiderato dal sistema.
- **Scenario principale:**
 1. L'utente inserisce l'identificativo per autenticarsi

3.3.28 UC9.2-GP - Errore identificativo inesistente

- **Codice:** UC9.2-GP.
- **Titolo:** errore identificativo inesistente.
- **Attori primari:** utente non acceduto.
- **Descrizione:** l'utente viene avvisato che ha inserito un identificativo non riconosciuto.
- **Precondizione:** il sistema riconosce l'utilizzatore come un utente non acceduto.
- **Postcondizione:** il sistema comunica all'utilizzatore l'errore.
- **Scenario principale:**
 1. L'utente inserisce un identificativo non esistente
 2. L'utente conferma l'invio dei dati
 3. L'utente visualizza il messaggio d'errore

3.3.29 UC10-GP - Uscita dell'utente dal sistema

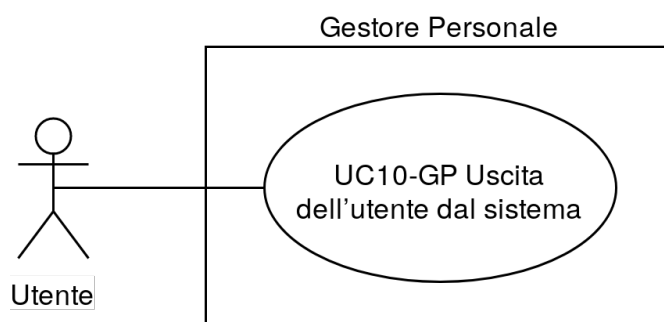


Figura 12: UC10-GP - Uscita dell'utente dal sistema

- **Codice:** UC10-GP.
- **Titolo:** uscita dell'utente dal sistema.
- **Attori primari:** utente.

- **Descrizione:** l'utente esce dal sistema ed ha la possibilità di rientrarci come un diverso utente o come lo stesso di prima.
- **Precondizione:** l'utente è all'interno del sistema.
- **Postcondizione:** l'utente è ora un utente non acceduto e si trova a poter accedere nuovamente nel sistema.
- **Scenario principale:**
 1. L'utente riconosciuto dal sistema effettua l'uscita dallo stesso

3.3.30 UC11-GP - Inserimento utente

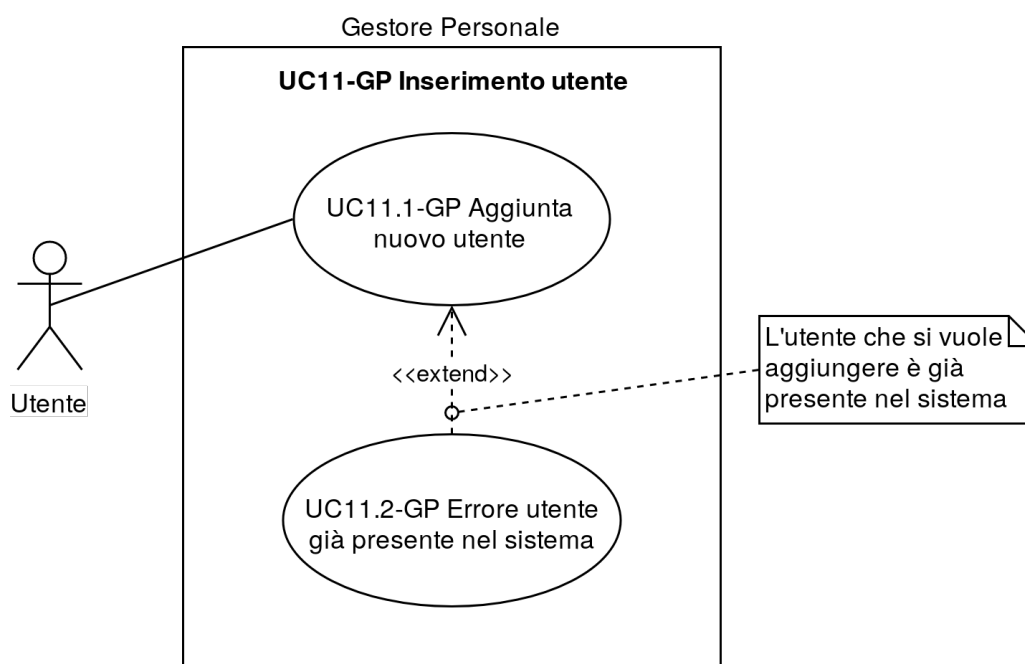


Figura 13: UC11-GP - Inserimento utente

- **Codice:** UC11-GP.
- **Titolo:** inserimento utente.
- **Attori primari:** utente.
- **Descrizione:** l'utente, acceduto al sistema, aggiunge un nuovo utente nel sistema. Non è possibile che un utente non acceduto si iscriva da solo per la prima volta. In un primo momento è presente solo un utente predefinito che può aggiungere gli altri utenti.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** un utente viene aggiunto al sistema.
- **Scenario principale:**
 1. L'utente aggiunge un nuovo utente

3.3.31 UC11.1-GP - Aggiunta nuovo utente

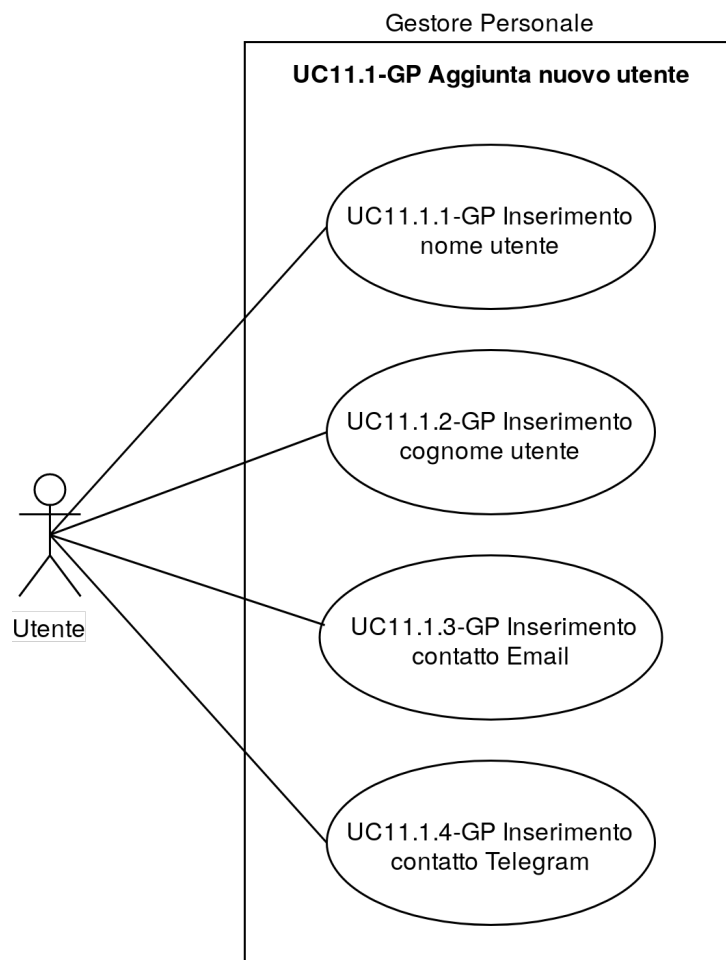


Figura 14: UC11.1-GP - Aggiunta nuovo utente

- **Codice:** C11.1-GP.
- **Titolo:** Aggiunta nuovo utente.
- **Attori primari:** utente.
- **Descrizione:** un nuovo utente viene inserito nel sistema.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** un utente viene aggiunto al sistema.
- **Scenario principale:**
 1. L'utente inserisce i dati del nuovo utente da aggiungere
 2. L'utente conferma l'invio dei dati
 3. Il nuovo utente ora è riconosciuto dal sistema
- **Estensioni:**
 1. Errore utente già presente nel sistema [UC11.2-GP]

3.3.32 UC11.1.1-GP - Inserimento nome utente

- **Codice:** UC11.1.1-GP.
- **Titolo:** inserimento nome utente.
- **Attori primari:** utente.
- **Descrizione:** l'utente inserisce il nominativo dell'utente da aggiungere.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** il nome del nuovo utente è stato aggiunto nel sistema.
- **Scenario principale:**

1. L'utente aggiunge il nominativo del nuovo utente

3.3.33 UC11.1.2-GP - Inserimento cognome utente

- **Codice:** UC11.1.2-GP.
- **Titolo:** inserimento cognome utente.
- **Attori primari:** utente.
- **Descrizione:** l'utente inserisce il cognome dell'utente da aggiungere.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** il cognome del nuovo utente è stato aggiunto nel sistema.
- **Scenario principale:**

1. L'utente aggiunge il cognome del nuovo utente

3.3.34 UC11.1.3-GP - Inserimento contatto Email

- **Codice:** UC11.1.3-GP.
- **Titolo:** inserimento contatto Email.
- **Attori primari:** utente.
- **Descrizione:** l'utente inserisce il contatto Email dell'utente da aggiungere. L'inserimento di questo campo è opzionale, ma è obbligatorio che l'utente da aggiungere possieda un contatto Email o un contatto Telegram.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** il contatto Email è stato aggiunto.
- **Scenario principale:**

1. L'utente aggiunge il contatto Email del nuovo utente

3.3.35 UC11.1.4-GP - Inserimento contatto Telegram

- **Codice:** UC11.1.4-GP.
- **Titolo:** inserimento contatto Telegram.
- **Attori primari:** utente.
- **Descrizione:** l'utente inserisce il contatto Telegram dell'utente da aggiungere. L'inserimento di questo campo è opzionale, ma è obbligatorio che l'utente da aggiungere possieda un contatto Email o un contatto Telegram.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** il contatto Telegram è stato aggiunto.
- **Scenario principale:**
 1. L'utente aggiunge il contatto Telegram del nuovo utente

3.3.36 UC11.2-GP - Errore utente già presente nel sistema

- **Codice:** UC11.2-GP.
- **Titolo:** errore utente già presente nel sistema.
- **Attori primari:** utente.
- **Descrizione:** l'utente viene avvisato che i contatti Telegram o Email immessi non sono univoci all'interno del sistema.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** il sistema comunica all'utilizzatore l'errore e l'utente non viene inserito.
- **Scenario principale:**
 1. L'utente visualizza il messaggio d'errore

3.3.37 UC12-GP - Rimozione utente

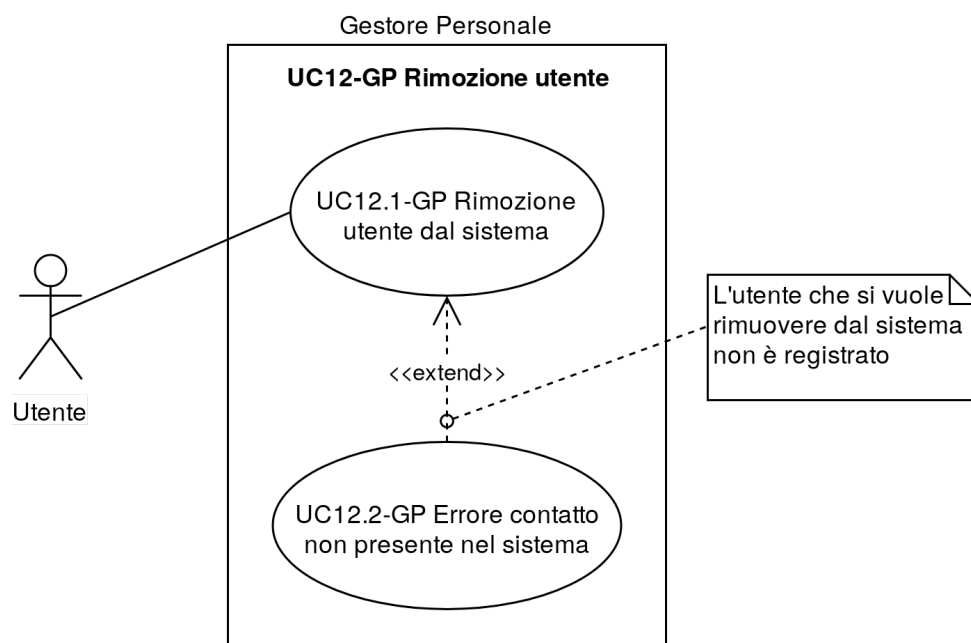


Figura 15: UC12-GP - Rimozione utente

- **Codice:** UC12-GP.
- **Titolo:** rimozione utente.
- **Attori primari:** utente.
- **Descrizione:** l'utente rimuove l'utente selezionato dal sistema.
- **Precondizione:** un utente già presente nel sistema deve essere rimosso.
- **Postcondizione:** un utente viene rimosso dal sistema e non è ora più riconoscibile.
- **Scenario principale:**
 1. L'utente rimuove l'utente selezionato

3.3.38 UC12.1-GP - Rimozione utente dal sistema

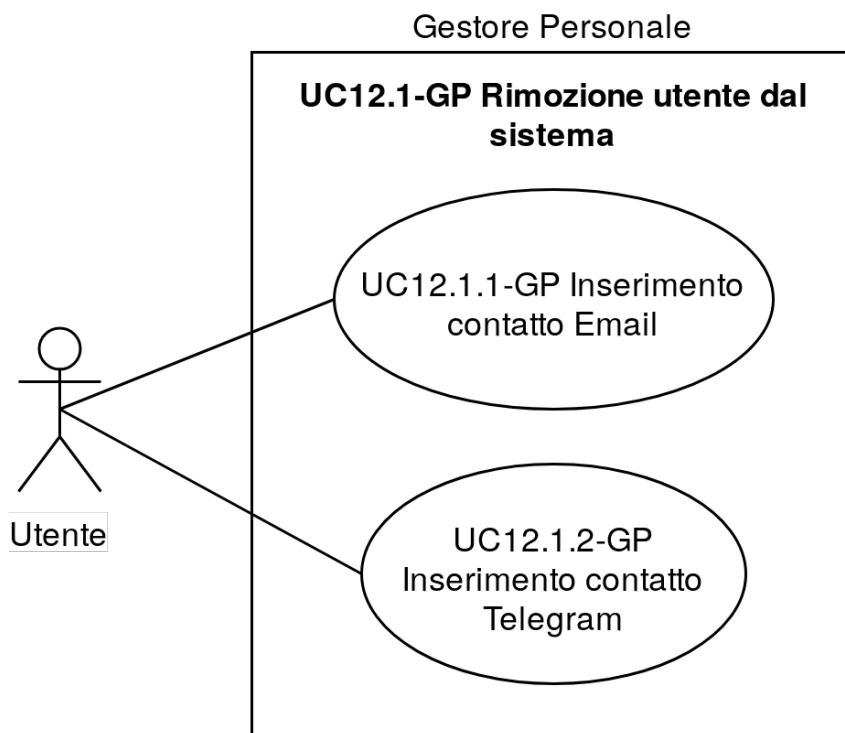


Figura 16: UC12.1-GP - Rimozione utente dal sistema

- **Codice:** UC12.1-GP.
- **Titolo:** rimozione utente dal sistema.
- **Attori primari:** utente.
- **Descrizione:** un utente, acceduto al sistema, rimuove un utente presente nel sistema tramite l'inserimento del suo contatto Email o Telegram. Questo utente può essere anche se stesso.
- **Precondizione:** un utente già presente nel sistema deve essere rimosso.
- **Postcondizione:** un utente con il contatto Email o Telegram inserito viene rimosso dal sistema.
- **Scenario principale:**
 1. L'utente inserisce il contatto dell'utente da rimuovere
 2. L'utente conferma l'invio dei dati
 3. L'utente da rimuovere è stato rimosso dal sistema
- **Estensioni:**
 - Errore contatto non presente nel sistema [UC12.2-GP]

3.3.39 UC12.1.1-GP - Inserimento contatto Email

- **Codice:** UC12.1.1-GP.
- **Titolo:** inserimento contatto Email.
- **Attori primari:** utente.
- **Descrizione:** l'utente ha aggiunto il contatto Email relativo all'utente che vuole rimuovere. L'inserimento di questo campo non avviene se viene invece inserito il contatto Telegram.
- **Precondizione:** un utente già presente nel sistema deve essere rimosso.
- **Postcondizione:** il contatto Email dell'utente da rimuovere è stato inserito.
- **Scenario principale:**
 1. L'utente inserisce il contatto Email dell'utente da rimuovere

3.3.40 UC12.1.2-GP - Inserimento contatto Telegram

- **Codice:** UC12.1.2-GP.
- **Titolo:** inserimento contatto Telegram.
- **Attori primari:** utente.
- **Descrizione:** l'utente ha aggiunto il contatto Telegram relativo all'utente che vuole rimuovere. L'inserimento di questo campo non avviene se viene invece inserito il contatto Email.
- **Precondizione:** un utente già presente nel sistema deve essere rimosso.
- **Postcondizione:** il contatto Telegram dell'utente da rimuovere è stato inserito.
- **Scenario principale:**
 1. L'utente inserisce il contatto Telegram del utente da rimuovere.

3.3.41 UC12.2-GP - Errore contatto non presente nel sistema

- **Codice:** UC12.2-GP.
- **Titolo:** errore contatto non presente nel sistema.
- **Attori primari:** utente.
- **Descrizione:** l'utente viene avvisato che il contatto inserito non è presente nel sistema.
- **Precondizione:** un utente già presente nel sistema deve essere rimosso.
- **Postcondizione:** il sistema comunica all'utente l'errore e nessun utente viene rimosso.
- **Scenario principale:**
 1. L'utente inserisce il contatto dell'utente da rimuovere
 2. L'utente selezionato attraverso il contatto Telegram o Email non viene rimosso perché non presente nel sistema.

3.3.42 UC13-GP - Modifica utente

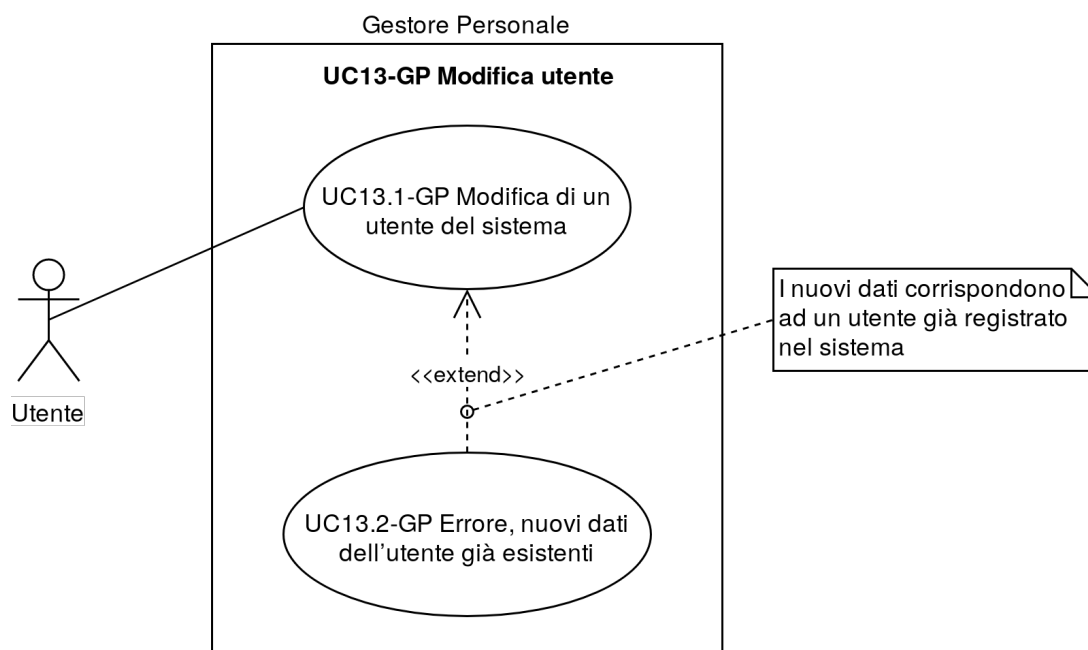


Figura 17: UC13-GP - Modifica utente

- **Codice:** UC13-GP.
- **Titolo:** modifica utente.
- **Attori primari:** utente.
- **Descrizione:** l'utente vuole modificare le informazioni a esso relative.
- **Precondizione:** l'utente vuole modificare alcuni dei dati presenti nel sistema.
- **Postcondizione:** i campi dell'utente sono stati modificati correttamente.
- **Scenario principale:**
 1. L'utente modifica i dati

3.3.43 UC13.1-GP - Modifica di un utente del sistema

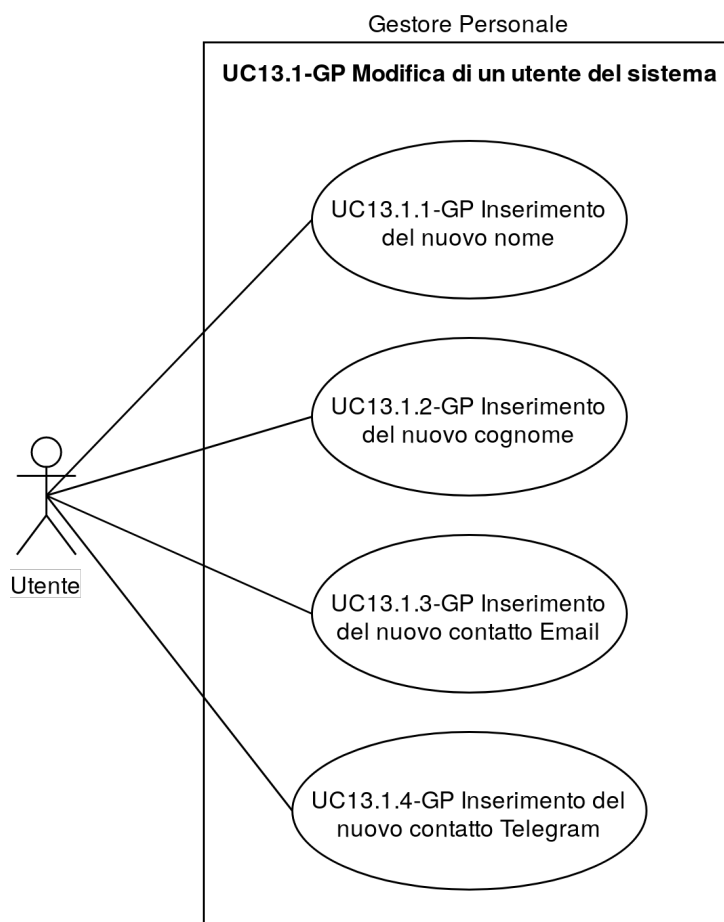


Figura 18: UC13.1-GP - Modifica di un utente del sistema

- **Codice:** UC13.1-GP.
- **Titolo:** modifica di un utente del sistema.
- **Attori primari:** utente.
- **Descrizione:** l'utente vuole modificare i suoi dati compilando i relativi campi.
- **Precondizione:** l'utente vuole modificare alcuni dei dati presenti nel sistema.
- **Postcondizione:** l'utente è stato modificato.
- **Scenario principale:**
 1. L'utente inserisce i nuovi dati che vuole rimpiazzare ai precedenti
 2. L'utente viene modificato
- **Estensioni:**
 - Errore, nuovi dati dell'utente già esistenti [UC13.2-GP]

3.3.44 UC13.1.1-GP - Inserimento del nuovo nome

- **Codice:** UC13.1.1-GP.
- **Titolo:** inserimento del nuovo nome.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge il nuovo nome che vuole modificare.
- **Precondizione:** l'utente vuole modificare alcuni dei dati presenti nel sistema.
- **Postcondizione:** il nome è stato inserito.
- **Scenario principale:**
 1. L'utente inserisce il suo nuovo nome

3.3.45 UC13.1.2-GP - Inserimento del nuovo cognome

- **Codice:** UC13.1.2-GP.
- **Titolo:** inserimento del nuovo cognome.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge il nuovo cognome che vuole modificare.
- **Precondizione:** l'utente vuole modificare alcuni dei dati presenti nel sistema.
- **Postcondizione:** il cognome è stato inserito.
- **Scenario principale:**
 1. L'utente inserisce il suo nuovo cognome

3.3.46 UC13.1.3-GP - Inserimento del nuovo contatto Email

- **Codice:** UC13.1.3-GP.
- **Titolo:** inserimento del nuovo contatto Email.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge il nuovo contatto Email rimpiazzando il suo contatto precedente o aggiungendone uno per la prima volta.
- **Precondizione:** l'utente vuole modificare alcuni dei dati presenti nel sistema.
- **Postcondizione:** il contatto Email è stato inserito.
- **Scenario principale:**
 1. L'utente inserisce il nuovo contatto Email

3.3.47 UC13.1.4-GP - Inserimento del nuovo contatto Telegram

- **Codice:** UC13.1.4-GP.
- **Titolo:** inserimento del nuovo contatto Telegram.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge il nuovo contatto Telegram rimpiazzando il suo contatto precedente o aggiungendone uno per la prima volta.
- **Precondizione:** l'utente vuole modificare alcuni dei dati presenti nel sistema.
- **Postcondizione:** il contatto Telegram è stato inserito.
- **Scenario principale:**
 1. L'utente inserisce il nuovo contatto Telegram

3.3.48 UC13.2-GP - Errore, nuovi dati dell'utente già esistenti

- **Codice:** UC13.2-GP.
- **Titolo:** errore, nuovi dati dell'utente già esistenti.
- **Attori primari:** utente.
- **Descrizione:** i nuovi dati dell'utente da modificare, che sono stati inseriti, sono già presenti nel sistema. Questo vuol dire che i nuovi campi corrispondono a quelli di un utente già esistente. In particolare il contatto Telegram o Email, perchè una persona può avere lo stesso nome e cognome di un altro, ma non la stessa Email e nemmeno lo stesso identificativo Telegram.
- **Precondizione:** l'utente vuole modificare alcuni dei dati presenti nel sistema.
- **Postcondizione:** l'utente non è stato modificato e viene visualizzato un messaggio di errore.
- **Scenario principale:**
 1. L'utente inserisce i campi richiesti dal sistema per la modifica
 2. L'utente visualizza un messaggio di errore

3.3.49 UC14-GP - Aggiunta preferenze

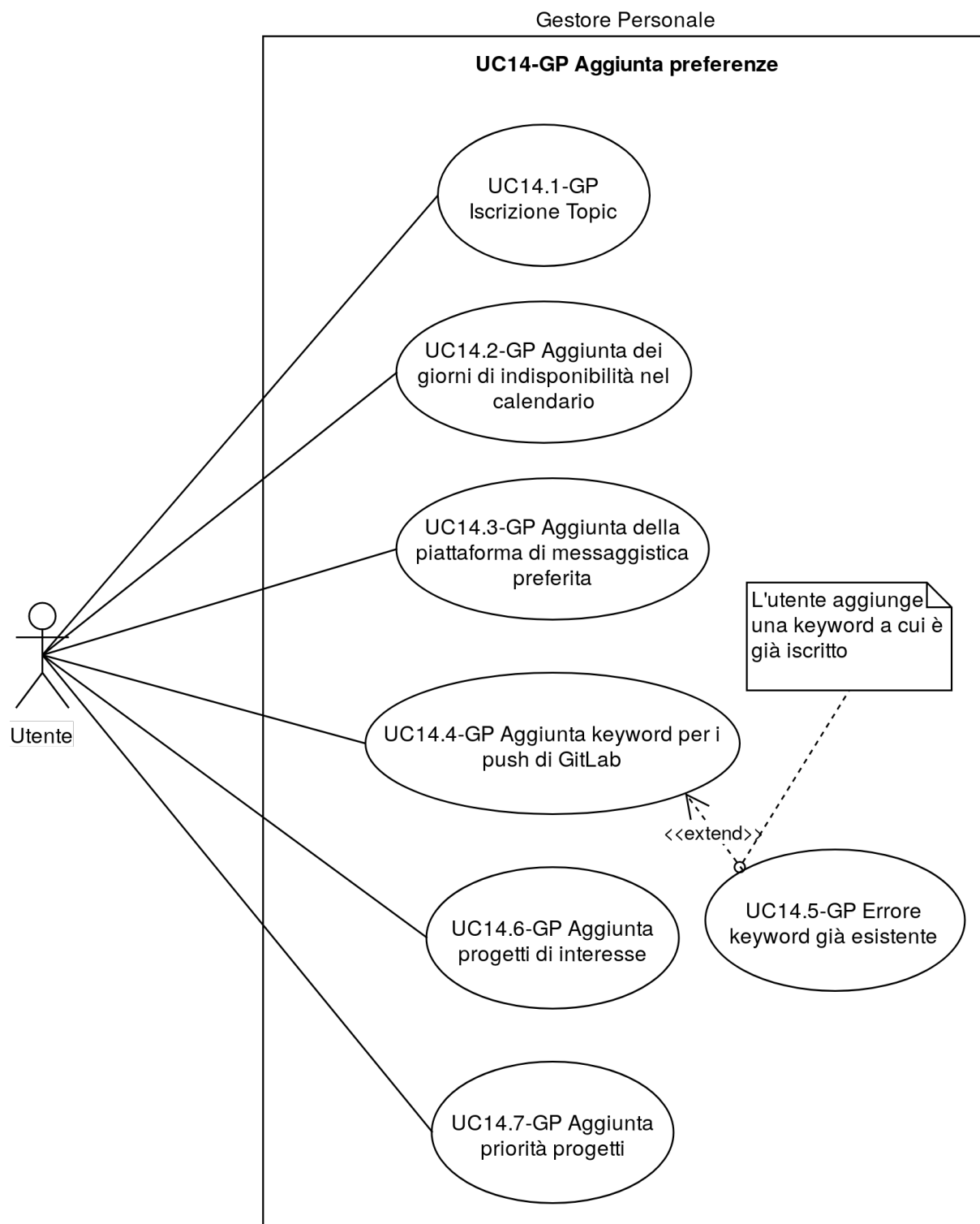


Figura 19: UC14-GP - Aggiunta preferenze

- **Codice:** UC14-GP.
- **Titolo:** aggiunta preferenze.
- **Attori primari:** utente.

- **Descrizione:** l'utente, date le varie opzioni per configurare Butterfly, aggiunge una preferenza tra Topic, giorni di calendario e la piattaforma di messaggistica (Telegram o Email) preferita.
- **Precondizione:** l'utente ha acceduto con l'identificativo corretto nel sistema e non ha già selezionato tutte le preferenze possibili proposte da Butterfly.
- **Postcondizione:** la nuova configurazione contiene una o più preferenze in aggiunta rispetto a quella precedente.
- **Scenario principale:**
 1. L'utente aggiunge una o più preferenze

3.3.50 UC14.1-GP - Iscrizione Topic

- **Codice:** UC14.1-GP.
- **Titolo:** iscrizione Topic.
- **Attori primari:** utente.
- **Descrizione:** data la lista di Topic presenti per i progetti a cui è iscritto l'utente, esso ne seleziona uno o più a cui è interessato, per poterne ricevere le notifiche. I Topic sono divisi per categoria e comprendono label (quali per esempio "bug" o "fix", ecc) o "commit", con progetto di appartenenza e applicazione di provenienza: Redmine o GitLab.
- **Precondizione:** l'utente ha acceduto con l'identificativo corretto nel sistema e non ha già selezionato tutte le preferenze possibili proposte da Butterfly.
- **Postcondizione:** il numero di Topic a cui è interessato l'utente è aumentato.
- **Scenario principale:**
 1. L'utente procede all'iscrizione di uno o più Topic

3.3.51 UC14.2-GP - Aggiunta dei giorni di indisponibilità nel calendario

- **Codice:** UC14.2-GP.
- **Titolo:** aggiunta dei giorni di indisponibilità nel calendario.
- **Attori primari:** utente.
- **Descrizione:** dato il calendario lavorativo, l'utente aggiunge uno o più giorni in cui non è reperibile e non vuole ricevere notifiche.
- **Precondizione:** l'utente ha acceduto con l'identificativo corretto nel sistema e non ha già selezionato tutte le preferenze possibili proposte da Butterfly.
- **Postcondizione:** il numero di giorni in cui l'utente non si rende disponibile è aumentato.
- **Scenario principale:**
 1. L'utente procede all'inserimento di uno o più giorni di indisponibilità

3.3.52 UC14.3-GP - Aggiunta della piattaforma di messaggistica preferita

- **Codice:** UC14.3-GP.
- **Titolo:** aggiunta della piattaforma di messaggistica preferita.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge la sua preferenza tra Telegram e Email dove vuole ricevere le notifiche. Questo non comporta l'aggiunta di identificativo Telegram o del proprio contatto Email, ma solo della selezione della preferenza tra le due tecnologie.
- **Precondizione:** l'utente ha acceduto con l'identificativo corretto nel sistema e non ha già selezionato tutte le preferenze possibili proposte da Butterfly.
- **Postcondizione:** il numero di piattaforme di messaggistica selezionate dall'utente è aumentato.
- **Scenario principale:**
 1. L'utente procede all'aggiunta di una o più piattaforme di messaggistica

3.3.53 UC14.4-GP - Aggiunta keyword per i push di GitLab

- **Codice:** UC14.4-GP.
- **Titolo:** aggiunta keyword per i push di GitLab.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge le keyword che vuole che siano contenute nei messaggi di commit dei push di cui vuole ricevere la notifica.
- **Precondizione:** l'utente ha acceduto con l'identificativo corretto nel sistema e non ha già selezionato tutte le preferenze possibili proposte da Butterfly.
- **Postcondizione:** nelle nuove configurazioni dell'utente selezionato sono presenti una o più nuove keyword per ricevere le notifiche da push di GitLab di interesse.
- **Scenario principale:**
 1. L'utente procede all'aggiunta di una o più nuove keyword
- **Estensioni:**
 1. Errore keyword già esistente [UC14.5-GP]

3.3.54 UC14.5-GP - Errore keyword già esistente

- **Codice:** UC14.5-GP.
- **Titolo:** errore keyword già esistente.
- **Attori primari:** utente.
- **Descrizione:** l'utente vuole aggiungere una keyword che ha già aggiunto in precedenza.
- **Precondizione:** l'utente ha acceduto con l'identificativo corretto nel sistema e non ha già selezionato tutte le preferenze possibili proposte da Butterfly.

- **Postcondizione:** il sistema comunica all'utilizzatore l'errore della keyword.
- **Scenario principale:**
 1. L'utente procede all'aggiunta di keyword
 2. Il sistema rileva che queste sono già presenti
 3. L'utente visualizza un messaggio di errore

3.3.55 UC14.6-GP - Aggiunta progetti di interesse

- **Codice:** UC14.6-GP.
- **Titolo:** inserimento progetti di interesse.
- **Attori primari:** utente.
- **Descrizione:** l'utente seleziona da una lista i progetti da cui è interessato ricevere delle notifiche.
- **Precondizione:** l'utente ha acceduto con l'identificativo corretto nel sistema e non ha già selezionato tutte le preferenze possibili proposte da Butterfly.
- **Postcondizione:** almeno un progetto è stato aggiunto.
- **Scenario principale:**
 1. L'utente aggiunge uno o più progetti

3.3.56 UC14.7-GP - Aggiunta priorità progetti

- **Codice:** UC14.7-GP.
- **Titolo:** inserimento progetti di interesse.
- **Attori primari:** utente.
- **Descrizione:** l'utente, per ogni progetto a cui è interessato, seleziona la priorità dei progetti.
- **Precondizione:** l'utente ha acceduto con l'identificativo corretto nel sistema e non ha già selezionato tutte le preferenze possibili proposte da Butterfly.
- **Postcondizione:** la priorità per progetto è stata aggiunta.
- **Scenario principale:**
 1. L'utente aggiunge le priorità i progetti di interesse

3.3.57 UC15-GP - Rimozione preferenze

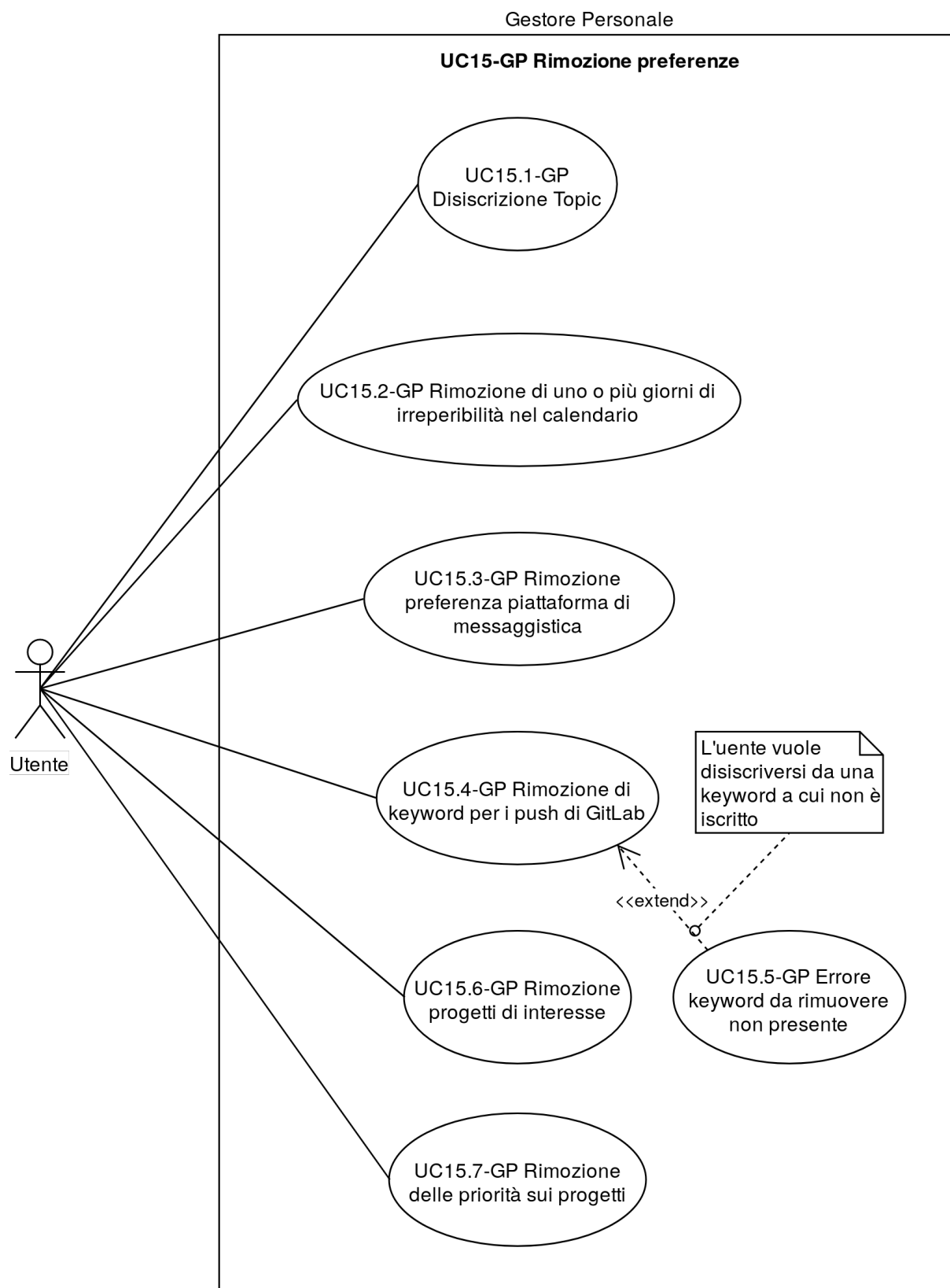


Figura 20: UC15-GP - Rimozione preferenze

- **Codice:** UC15-GP.

- **Titolo:** rimozione preferenze.
- **Attori primari:** utente.
- **Descrizione:** l'utente, dopo aver selezionato delle preferenze dalle opzioni di configurazione, ne rimuove una o più. Le preferenze consistono in Topic, date di calendario e piattaforma di messaggistica (Telegram e email).
- **Precondizione:** l'utente ha eseguito l'accesso nel sistema ed è presente almeno una preferenza selezionata tra quelle proposte da Butterfly.
- **Postcondizione:** la nuova configurazione contiene una o più preferenze in meno rispetto a quella precedente.
- **Scenario principale:**
 1. L'utente procede alla rimozione di una o più preferenze

3.3.58 UC15.1-GP - Disiscrizione Topic

- **Codice:** UC15.1-GP.
- **Titolo:** disiscrizione Topic.
- **Attori primari:** utente.
- **Descrizione:** l'utente si disiscrive da uno o più Topic dai quali prima riceveva delle notifiche.
- **Precondizione:** l'utente ha eseguito l'accesso nel sistema ed è presente almeno una preferenza selezionata tra quelle proposte da Butterfly.
- **Postcondizione:** il numero di Topic a cui è iscritto l'utente è diminuito.
- **Scenario principale:**
 1. L'utente procede alla disiscrizione di uno o più Topic

3.3.59 UC15.2-GP - Rimozione di uno o più giorni di irreperibilità nel calendario

- **Codice:** UC15.2-GP.
- **Titolo:** rimozione di uno o più giorni di irreperibilità nel calendario.
- **Attori primari:** utente.
- **Descrizione:** l'utente rimuove i giorni di calendario in cui precedentemente non era reperibile, tornando disponibile.
- **Precondizione:** l'utente ha eseguito l'accesso nel sistema ed è presente almeno una preferenza selezionata tra quelle proposte da Butterfly.
- **Postcondizione:** il numero di giorni di calendario in cui l'utente non è reperibile è diminuito.
- **Scenario principale:**
 1. L'utente procede alla rimozione di uno o più giorni di irreperibilità

3.3.60 UC15.3-GP - Rimozione preferenza piattaforma di messaggistica

- **Codice:** UC15.3-GP.
- **Titolo:** rimozione preferenza piattaforma di messaggistica.
- **Attori primari:** utente.
- **Descrizione:** l'utente rimuove una o più preferenze tra Telegram e Email dalle quali non vuole più ricevere notifiche tramite Butterfly. Questo non comporta la rimozione dell'identificativo Telegram o del proprio contatto Email, ma solo della deselection della preferenza tra le due tecnologie.
- **Precondizione:** l'utente ha eseguito l'accesso nel sistema ed è presente almeno una preferenza selezionata tra quelle proposte da Butterfly.
- **Postcondizione:** il numero di piattaforme di messaggistica da cui l'utente vuole ricevere notifiche è diminuito.
- **Scenario principale:**
 1. L'utente procede alla rimozione di una o più piattaforme di messaggistica

3.3.61 UC15.4-GP - Rimozione di keyword per i push di GitLab

- **Codice:** UC15.4-GP.
- **Titolo:** rimozione di keyword per i push di GitLab.
- **Attori primari:** utente.
- **Descrizione:** l'utente seleziona e rimuove una o più keyword già presente nel sistema per non ricevere la notifica di push in cui i messaggi di commit contengono la keyword rimossa.
- **Precondizione:** l'utente ha eseguito l'accesso nel sistema ed è presente almeno una preferenza selezionata tra quelle proposte da Butterfly.
- **Postcondizione:** nelle nuove configurazioni dell'utente selezionato sono state rimosse delle keyword precedentemente presenti.
- **Scenario principale:**
 1. L'utente rimuove una o più keyword per cui non vuole più ricevere messaggi di push
- **Estensioni:**
 1. Errore keyword da rimuovere non presente [UC15.5-GP]

3.3.62 UC15.5-GP - Errore keyword da rimuovere non presente

- **Codice:** UC15.5-GP.
- **Titolo:** errore keyword da rimuovere non presente.
- **Attori primari:** utente.
- **Descrizione:** l'utente vuole rimuovere una keyword che non ha mai inserito o che ha rimosso precedentemente.

- **Precondizione:** l'utente ha eseguito l'accesso nel sistema ed è presente almeno una preferenza selezionata tra quelle proposte da Butterfly.
- **Postcondizione:** viene visualizzato un messaggio d'errore con indicato che la keyword selezionata che non è presente nel sistema.
- **Scenario principale:**
 1. L'utente inserisce la keyword che vuole rimuovere
 2. Il sistema rileva che non è presente
 3. L'utente visualizza il messaggio di errore

3.3.63 UC15.6-GP - Rimozione progetti di interesse

- **Codice:** UC15.6-GP.
- **Titolo:** rimozione progetti di interesse.
- **Attori primari:** utente.
- **Descrizione:** l'utente modifica la lista di progetti a cui è iscritto rimuovendone uno.
- **Precondizione:** l'utente ha eseguito l'accesso nel sistema ed è presente almeno una preferenza selezionata tra quelle proposte da Butterfly.
- **Postcondizione:** la lista di progetti a cui un utente è iscritto è diminuita.
- **Scenario principale:**
 1. L'utente rimuove un progetto a cui è iscritto

3.3.64 UC15.7-GP - Rimozione delle priorità sui progetti

- **Codice:** UC15.7-GP.
- **Titolo:** rimozione delle priorità sui progetti.
- **Attori primari:** utente.
- **Descrizione:** l'utente modifica la priorità dei progetti a cui è iscritto rimuovendo la priorità precedentemente inserita per quel progetto. Un progetto però non può restare tra le preferenze dell'utente senza aver segnalato una priorità, per cui sarà necessario aggiungerne un'altra subito dopo la rimozione.
- **Precondizione:** l'utente ha eseguito l'accesso nel sistema ed è presente almeno una preferenza selezionata tra quelle proposte da Butterfly.
- **Postcondizione:** la priorità di un progetto presente nella lista dei progetti è stata cambiata.
- **Scenario principale:**
 1. L'utente modifica la priorità della lista di progetti a cui è iscritto

4 Requisiti

Ad ogni requisito viene assegnato il codice identificativo univoco:

$$R[\text{Numero}] [\text{Tipo}] [\text{Priorità}]$$

in cui ogni parte ha un significato preciso:

- **R**: requisito.
- **Numero**: numero progressivo che segue la struttura dei documenti.
- **Tipo**: la la tipologia di requisito che può essere di:
 - **F**: funzionalità.
 - **Q**: QUALITÀ_G.
 - **V**: vincolo.
- **Priorità**: indica il grado di urgenza di un requisito di essere soddisfatto, come:
 - **0**: opzionale.
 - **1**: desiderabile.
 - **2**: obbligatorio.

Esempio: R2Q1 indica il secondo requisito di qualità ed è desiderabile.

4.1 Requisiti di funzionalità

Codice	Requisito	Fonte
R1F2	Redmine deve poter inviare una segnalazione al Producer Redmine	Interno UC1-PR
R1.1F2	Redmine deve poter inviare la segnalazione di apertura issue al Producer Redmine	Interno UC1.1-PR
R1.2F2	Redmine deve poter inviare la segnalazione di modifica issue al Producer Redmine	Interno UC1.2-PR
R1.3F2	Redmine deve poter inviare la segnalazione di commento di una issue al Producer Redmine	Interno UC1.3-PR
R2F2	GitLab deve essere in grado di inviare una segnalazione al Producer GitLab	Interno UC2-PG
R2.1F2	GitLab deve essere in grado di segnalare l'apertura di issue al Producer GitLab	Interno UC2.1-PG
R2.2F2	GitLab deve essere in grado di segnalare la modifica issue al Producer GitLab	Interno UC2.2-PG
R2.3F2	GitLab deve essere in grado di segnalare il commento di una issue al Producer GitLab	Interno UC2.3-PG
R2.4F2	GitLab deve poter segnalare un evento di push al Producer GitLab	Interno UC2.4-PG
R2.5F2	GitLab deve poter segnalare un evento di commento di commit al Producer GitLab	Interno UC2.5-PG
R3F2	Il Producer Redmine deve essere in grado di inviare un messaggio al Gestore Personale	Interno UC3-GP
R3.1F2	Il Producer Redmine deve essere in grado di inviare un messaggio di apertura issue al Gestore Personale	Interno UC3.1-GP
R3.2F2	Il Producer Redmine deve essere in grado di inviare un messaggio di modifica issue al Gestore Personale	Interno UC3.2-GP
R3.3F2	Il Producer Redmine deve essere in grado di inviare un messaggio di commento issue al Gestore Personale	Interno UC3.3-GP

Tabella 1: Elenco dei requisiti di funzionalità (1)

Codice	Requisito	Fonte
R4F2	Il Producer GitLab deve essere in grado di inviare un messaggio al Gestore Personale	Interno UC4-GP
R4.1F2	Il Producer GitLab deve essere in grado di inviare un messaggio di push al Gestore Personale	Interno UC4.1-GP
R4.2F2	Il Producer GitLab deve essere in grado di inviare un messaggio di nuova issue al Gestore Personale	Interno UC4.2-GP
R4.3F2	Il Producer GitLab deve essere in grado di inviare un messaggio di modifica issue al Gestore Personale	Interno UC4.3-GP
R4.4F2	Il Producer GitLab deve essere in grado di inviare un messaggio di commento di issue al Gestore Personale	Interno UC4.4-GP
R4.5F2	Il Producer GitLab deve essere in grado di inviare messaggi di commento di commit al Gestore Personale	Interno UC4.5-GP
R5F2	Il Gestore Personale deve poter inviare il messaggio finale al Consumer Telegram	Interno UC5-CT
R6F2	Il Gestore Personale deve poter inviare il messaggio finale al Consumer Email	Interno UC6-CE
R7F2	Il Consumer Telegram deve poter inoltrare il messaggio finale al bot Telegram	Interno UC7-BT
R8F2	Il Consumer Email deve poter inoltrare il messaggio finale al server Email	Interno UC8-SE
R9F0	L'utente può eseguire l'accesso al Gestore Personale	Interno UC9.1-GP
R9.1F0	L'utente può inserire il proprio identificativo all'interno del sistema	Interno UC9.1.1-GP
R10F0	Butterfly deve saper notificare l'utente se il tentativo di accesso non è andato a buon fine	Interno UC9.2-GP
R11F0	L'utente acceduto deve poter uscire dal sistema	Interno UC10-GP

Tabella 2: Elenco dei requisiti di funzionalità (2)

Codice	Requisito	Fonte
R12F0	L'utente acceduto deve poter aggiungere un nuovo utente	Interno UC11.1-GP
R12.1F0	L'utente acceduto deve poter inserire il nome dell'utente da aggiungere	Interno UC11.1.1-GP
R12.2F0	L'utente acceduto deve poter inserire il cognome dell'utente da aggiungere	Interno UC11.1.2-GP
R12.3F0	L'utente acceduto deve poter inserire il contatto Email dell'utente da aggiungere	Interno UC11.1.3-GP
R12.4F0	L'utente acceduto deve poter inserire il contatto Telegram dell'utente da aggiungere	Interno UC11.1.4-GP
R13F0	L'utente acceduto deve poter visualizzare un messaggio di errore se il nuovo contatto Telegram inserito corrisponde a quello di un utente già registrato nel sistema	Interno UC11.2-GP
R14F0	L'utente acceduto deve poter visualizzare un messaggio di errore se il nuovo contatto Email inserito corrisponde a quello di un utente già registrato nel sistema	Interno UC11.2-GP
R15F0	L'utente acceduto deve poter rimuovere un utente presente nel sistema	Interno UC12.1-GP
R15.1F0	L'utente acceduto deve poter inserire il contatto Email dell'utente da rimuovere	Interno UC12.1.1-GP
R15.2F0	L'utente acceduto deve poter inserire il contatto Telegram dell'utente da rimuovere	Interno UC12.1.2-GP
R16F0	L'utente acceduto deve poter visualizzare un messaggio di errore se l'utente da rimuovere non è presente nel sistema	Interno UC12.2-GP
R17F0	L'utente acceduto deve poter modificare le proprie informazioni	Interno UC13.1-GP
R17.1F0	L'utente acceduto deve poter modificare il nome	Interno UC13.1.1-GP
R17.2F0	L'utente acceduto deve poter modificare il cognome	Interno UC13.1.2-GP
R17.3F0	L'utente acceduto deve poter modificare il contatto Email	Interno UC13.1.3-GP
R17.4F0	L'utente acceduto deve poter scegliere il nuovo contatto Telegram	Interno UC13.1.4-GP
R18F0	L'utente acceduto deve poter visualizzare un messaggio di errore se il contatto Email inserito corrisponde a quello di un utente presente nel sistema	Interno UC13.2-GP

Tabella 3: Elenco dei requisiti di funzionalità (3)

Codice	Requisito	Fonte
R19F0	L'utente acceduto deve poter visualizzare un messaggio di errore se il contatto Telegram inserito corrisponde a quello di un utente presente nel sistema	Interno UC13.2-GP
R20F0	L'utente acceduto deve poter aggiungere le proprie preferenze nel sistema	Interno UC14-GP
R20.1F0	L'utente acceduto deve poter aggiungere nuovi Topic	Interno UC14.1-GP
R20.2F0	L'utente acceduto deve poter aggiungere nuovi giorni di indisponibilità nel calendario	Interno UC14.2-GP
R20.3F0	L'utente acceduto deve poter aggiungere una nuova piattaforma di messaggistica in cui ricevere le notifiche	Interno UC14.3-GP
R20.4F0	L'utente acceduto deve poter aggiungere nuove keyword di interesse per i messaggi di commit di GitLab	Interno UC14.4-GP
R20.5F0	L'utente acceduto deve poter visualizzare un messaggio di errore se la keyword inserita era già nella sua lista	Interno UC14.5-GP
R20.6F0	L'utente acceduto deve poter aggiungere i progetti a cui è interessato	Interno UC14.6-GP
R20.7F0	L'utente acceduto deve poter aggiungere la priorità ai progetti a cui è interessato	Interno UC14.7-GP
R21F0	L'utente acceduto deve poter rimuovere le proprie preferenze dal sistema	Interno UC15-GP
R21.1F0	L'utente acceduto deve poter rimuovere i Topic a cui è iscritto	Interno UC15.1-GP
R21.2F0	L'utente acceduto deve poter rimuovere giorni di indisponibilità nel calendario	Interno UC15.2-GP
R21.3F0	L'utente acceduto deve poter rimuovere una piattaforma di messaggistica in cui ricevere le notifiche	Interno UC15.3-GP
R21.4F0	L'utente acceduto deve poter rimuovere keyword di interesse per i messaggi di commit di GitLab	Interno UC15.4-GP
R21.5F0	L'utente acceduto deve poter visualizzare un messaggio di errore se la keyword da rimuovere è assente dalla sua lista	Interno UC15.5-GP
R21.6F0	L'utente acceduto vuole rimuovere un progetto a cui non è più interessato	Interno UC15.6-GP
R21.7F0	L'utente acceduto vuole rimuovere la priorità di un progetto a cui è interessato	Interno UC15.7-GP

Tabella 4: Elenco dei requisiti di funzionalità (4)

4.2 Requisiti di qualità

Codice	Requisito	Fonte
R1Q1	È stabilito un numero massimo di giorni di ritardo per la chiusura di una issue	Interno QPR001
R2Q1	L'INDICE DI GULPEASE _G di ogni documento deve rientrare all'interno di un intervallo stabilito	Interno QPD001
R3Q1	Una frequenza minima di commit devono essere effettuati in una settimana	Interno QPR004
R4Q1	Tutte le norme scelte dallo standard PEP8 devono essere rispettate	Interno QPS005
R5Q1	Un numero stabilito di requisiti desiderabili deve essere soddisfatto	Interno QPR006
R6Q1	Ogni documento deve attraversare tutte le fasi previste dal suo CICLO DI VITA _G	Interno QPR008
R7Q1	Viene stabilito il numero massimo di modifiche che può ricevere un prodotto prima di essere verificato	Interno QPR009
R8Q1	Ogni modulo deve essere prima progettato e solo successivamente codificato	Interno QPR011
R9Q1	L'intera progettazione del progetto non deve contenere un numero di pattern più alto di quanto segnalato nel <i>PianoDiQualifica v4.0.0_D</i>	Interno QPR012
R10Q1	Tutti i moduli devono possedere una suite di test allegata	Interno QPR013
R11Q2	Tutte le NORME _G inserite nelle <i>NormeDiProgetto v4.0.0_D</i> devono essere rispettate	Interno
R12Q2	Tutti i vincoli presenti nel <i>PianoDiQualifica v4.0.0_D</i> devono essere rispettati	Interno
R12.1Q2	Le applicazioni sviluppate devono rispettare i fattori trattati in The Twelve-Factor App segnati nel <i>PianoDiQualifica v4.0.0_D</i>	Capitolato <i>VE_2018-12-12_D</i>
R13Q2	È necessario presentare il BUG _G reporting per ogni componente	Capitolato
R14Q2	Deve essere redatta la documentazione sulle scelte progettuali effettuate	Capitolato
R14.1Q2	Ogni scelta descritta nella documentazione deve essere correlata dalle relative motivazioni	Capitolato

Tabella 5: Elenco dei requisiti di qualità (1)

Codice	Requisito	Fonte
R15Q2	È necessario testare ogni prodotto software considerando ogni sistema di riferimento e interazione tra le sue parti, perciò con test d'unità, d'integrazione e di sistema	Capitolato
R15.1Q2	È necessario fornire test d'unità per ogni componente applicativo	Capitolato
R15.2Q2	È necessario fornire test d'integrazione per ogni componente applicativo	Capitolato
R15.3Q2	È necessario testare interamente il sistema con test di sistema	Capitolato
R16Q1	Per ogni problema aperto documentato, si allegano delle soluzioni da attuare in futuro	Capitolato
R16.1Q2	Deve essere redatta una documentazione su eventuali problemi riscontrati rimasti ancora aperti al termine del progetto	Capitolato
R17Q2	È necessario presentare un file README _G per ogni componente	Capitolato
R17.1Q2	I file README delle componenti applicative devono contenere la documentazione delle API _G esposte dal servizio	Capitolato
R17.2Q2	I file README delle componenti applicative devono contenere le istruzioni per il loro utilizzo	Capitolato
R17.3Q2	È necessario presentare un file README per il Dockerfile	Capitolato
R17.3.1Q2	Il file README per il Dockerfile deve contenere le istruzioni per l'avvio	Capitolato
R17.3.2Q2	Il file README per il Dockerfile deve contenere la documentazione delle configurazioni custom scelte	Capitolato

Tabella 6: Elenco dei requisiti di qualità (2)

4.3 Requisiti di vincolo

Codice	Requisito	Fonte
R1V2	Devono essere sviluppati due componenti applicativi Producer tra Redmine 3.3.9, GitLab 11.7 e SonarQube 6.7	Capitolato
R1.1V0	È possibile avere un terzo componente applicativo Producer oltre ai due obbligatori	Capitolato
R2V2	Devono essere sviluppati due componenti applicativi Consumer tra Telegram (Bot API 4.0), Email e Slack 3.3.8	Capitolato
R2.1V0	È possibile avere un terzo componente applicativo Consumer oltre ai due obbligatori	Capitolato
R3V2	Docker 18.09 deve essere la tecnologia di riferimento per l'istanziamento di tutte le componenti	Capitolato
R3.1V2	È necessario presentare un Dockerfile per ogni componente	Capitolato
R4V1	Per lo sviluppo dei componenti applicativi è possibile usare come linguaggio JAVA _G 8 o una versione più recente, PYTHON 3.6.7 _G o NODE.JS _G 10.15.1	Capitolato
R5V1	È possibile utilizzare Apache Kafka 2.12 come Broker	Capitolato
R6V0	L'interfaccia è sviluppata utilizzando HTML5 e CSS3	Interno
R7V0	Il database viene sviluppato utilizzando MongoDB 4.0.6	Interno
R8V0	Il server web viene sviluppato utilizzando Python 3.6.7 con la libreria CHERRYPY _G	Interno
R9V0	Gli URL dell'interfaccia web devono rispettare lo standard REST	Interno
R10V0	Ciascun componente Docker viene istanziato tramite un file Docker Compose 3.7	Interno
R11V2	Le componenti Consumer devono essere in grado di inviare i messaggi provenienti da un Topic verso il corretto destinatario	Capitolato
R12V2	Le componenti Consumer devono essere in grado di abbonarsi ai Topic scelti	Capitolato
R13V2	Le segnalazioni devono poter essere gestite in maniera automatica e personalizzabile	Capitolato
R14V2	Nel sistema deve essere presente un Broker che istanzia e gestisce le segnalazioni organizzandole per Topic	Capitolato
R15V2	Le componenti Producer devono riuscire a pubblicare le segnalazioni recuperate sotto forma di messaggi secondo i Topic corretti	Capitolato

Tabella 7: Elenco dei requisiti di vincolo

4.4 Tracciamento

4.4.1 Tracciamento fonte-requisito

Fonte	Requisito
Capitolato	R12.1Q2
	R13Q2
	R14Q2
	R14.1Q2
	R15Q2
	R15.1Q2
	R15.2Q2
	R15.3Q2
	R16Q1
	R16.1Q2
	R17Q2
	R17.1Q2
	R17.2Q2
	R17.3Q2
	R17.3.1Q2
	R17.3.2Q2

Tabella 8: Elenco dei requisiti del capitolato (1)

Fonte	Requisito
Capitolato	R1V2
	R1.1V0
	R2V2
	R2.1V0
	R3V2
	R3.1V2
	R4V1
	R5V1
	R11V2
	R12V2
	R13V2
	R14V2
	R15V2

Tabella 9: Elenco dei requisiti del capitolato (2)

Fonte	Requisito
UC1-PR	R1F2
UC1.1-PR	R1.1F2
UC1.2-PR	R1.2F2
UC1.3-PR	R1.3F2
UC2-PR	R2F2
UC2.1-PG	R2F2.1
UC2.2-PG	R2F2.2
UC2.3-PG	R2F2.3
UC2.4-PG	R2F2.4
UC2.5-PG	R2F2.5
UC3-GP	R3F2
UC3.1-GP	R3.1F2
UC3.2-GP	R3.2F2
UC3.3-GP	R3.3F2
UC4-GP	R4F2
UC4.1-GP	R4.1F2
UC4.2-GP	R4.2F2
UC4.3-GP	R4.3F2
UC4.4-GP	R4.4F2
UC4.5-GP	R4.5F2
UC5-CT	R5F2
UC6-CE	R6F2
UC7-BT	R7F2
UC8-SE	R8F2
UC9.1-GP	R9F0
UC9.1.1-GP	R9.1F0
UC10-GP	R11F0

Tabella 10: Elenco dei requisiti per i casi d'uso (1)

Fonte	Requisito
UC11.1-GP	R12F0
UC11.1.1-GP	R12.1F0
UC11.1.2-GP	R12.2F0
UC11.1.3-GP	R12.3F0
UC11.1.4-GP	R12.4F0
UC11.2-GP	R13F0
	R14F0
UC12.1-GP	R15F0
UC12.1.1-GP	R15.1F0
UC12.1.2-GP	R15.2F0
UC12.2-GP	R16F0
UC13.1-GP	R17F0
UC13.1.1-GP	R17.1F0
UC13.1.2-GP	R17.2F0
UC13.1.3-GP	R17.3F0
UC13.1.4-GP	R17.4F0
UC13.2-GP	R18F0
	R19F0
UC14-GP	R20F0
UC14.1-GP	R20.1F0
UC14.2-GP	R20.2F0
UC14.3-GP	R20.3F0
UC14.4-GP	R20.4F0
UC14.5-GP	R20.5F0
UC14.6-GP	R20.6F0
UC14.7-GP	R20.7F0

Tabella 11: Elenco dei requisiti per i casi d'uso (2)

Fonte	Requisito
UC15-GP	R21F0
UC15.1-GP	R21.1F0
UC15.2-GP	R21.2F0
UC15.3-GP	R21.3F0
UC15.4-GP	R21.4F0
UC15.5-GP	R21.5F0
UC15.6-GP	R21.6F0
UC15.7-GP	R21.7F0

Tabella 12: Elenco dei requisiti per i casi d'uso (3)

Fonte	Requisito
QPR001	R1Q1
QPD001	R2Q1
QPR004	R3Q1
QPS005	R4Q1
QPR006	R5Q1
QPR008	R6Q1
QPR009	R7Q2
QPR011	R8Q1
QPR012	R9Q1
QPR013	R10Q1
<i>VE_2018-12-12_D</i>	R12.1Q2

Tabella 13: Elenco dei requisiti per gli obiettivi di qualità e verbali

Fonte	Requisito
Interno	R1F2
	R1.1F2
	R1.2F2
	R1.3F2
	R2F2
	R2.1F2
	R2.2F2
	R2.3F2
	R2.4F2
	R2.5F2
	R3F2
	R3.1F2
	R3.2F2
	R3.3F2
	R4F2
	R4.1F2
	R4.2F2
	R4.3F2
	R4.4F2
	R4.5F2
	R5F2
	R6F2
	R7F2
	R8F2
	R9F0
	R9.1F0
	R10F0
	R11F0

Tabella 14: Elenco dei requisiti da fonte interna (1)

Fonte	Requisito
Interno	R12F0
	R12.1F0
	R12.2F0
	R12.3F0
	R12.4F0
	R13F0
	R14F0
	R15F0
	R15.1F0
	R15.2F0
	R16F0
	R17F0
	R17.1F0
	R17.2F0
	R17.3F0
	R17.4F0
	R18F0
	R19F0
	R20F0
	R20.1F0
	R20.2F0
	R20.3F0
	R20.4F0
	R20.5F0
	R20.5F0
	R20.6F0
	R20.7F0

Tabella 15: Elenco dei requisiti da fonte interna (2)

Fonte	Requisito
Interno	R21F0
	R21.1F0
	R21.2F0
	R21.3F0
	R21.4F0
	R21.5F0
	R21.6F0
	R21.7F0
	R1Q1
	R2Q1
	R3Q1
	R4Q1
	R5Q1
	R6Q1
	R7Q1
	R8Q1
	R9Q2
	R10Q1
	R11Q1
	R12Q1
	R6V0
	R7V0
	R8V0
	R9V0
	R10V0

Tabella 16: Elenco dei requisiti da fonte interna (3)

4.4.2 Tracciamento requisito-fonte

Requisito	Fonte
R1F2	Interno UC1-PR
R1.1F2	Interno UC1.1-PR
R1.2F2	Interno UC1.2-PR
R1.3F2	Interno UC1.3-PR
R1.4F2	Interno UC1.4-PR
R2F2	Interno UC2-PG
R2.1F2	Interno UC2.1-PG
R2.2F2	Interno UC2.2-PG
R2.3F2	Interno UC2.3-PG
R2.4F2	Interno UC2.4-PG
R2.5F2	Interno UC2.5-PG

Tabella 17: Elenco dei requisiti funzionali in rapporto alle fonti (1)

Requisito	Fonte
R3F2	Interno UC3-GP
R3.1F2	Interno UC3.1-GP
R3.2F2	Interno UC3.2-GP
R3.3F2	Interno UC3.3-GP
R4F2	Interno UC4-GP
R4.1F2	Interno UC4.1-GP
R4.2F2	Interno UC4.2-GP
R4.3F2	Interno UC4.3-GP
R4.4F2	Interno UC4.4-GP
R4.5F2	Interno UC4.5-GP
R5F2	Interno UC5-CT
R6F2	Interno UC6-CE
R7F2	Interno UC7-BT
R8F2	Interno UC8-SE

Tabella 18: Elenco dei requisiti funzionali in rapporto alle fonti (2)

Requisito	Fonte
R9F2	Interno UC9.1-GP
R9.1F0	Interno UC9.1.1-GP
R10F0	Interno UC9.2-GP
R11F0	Interno UC10-GP
R12F0	Interno UC11.1-GP
R12.1F0	Interno UC11.1.1-GP
R12.2F0	Interno UC11.1.2-GP
R12.3F0	Interno UC11.1.3-GP
R12.4F0	Interno UC11.1.4-GP
R13F0	Interno UC11.2-GP
R14F0	Interno UC11.1-GP
R15F0	Interno UC12.1-GP
R15.1F0	Interno UC12.1.1-GP
R15.2F0	Interno UC12.1.2-GP
R16F0	Interno UC12.2-GP

Tabella 19: Elenco dei requisiti funzionali in rapporto alle fonti (3)

Requisito	Fonte
R17F0	Interno UC13.1-GP
R17.1F0	Interno UC13.1.1-GP
R17.2F0	Interno UC13.1.2-GP
R17.3F0	Interno UC13.1.3-GP
R17.4F0	Interno UC13.1.4-GP
R18F0	Interno UC13.2-GP
R19F0	Interno UC13.2-GP
R20F0	Interno UC14-GP
R20.1F0	Interno UC14.1-GP
R20.2F0	Interno UC14.2-GP
R20.3F0	Interno UC14.3-GP
R20.4F0	Interno UC14.4-GP
R20.5F0	Interno UC14.5-GP
R20.6F0	Interno UC14.6-GP
R20.7F0	Interno UC14.7-GP

Tabella 20: Elenco dei requisiti funzionali in rapporto alle fonti (4)

Requisito	Fonte
R21F0	Interno UC15-GP
R21.1F0	Interno UC15.1-GP
R21.2F0	Interno UC15.2-GP
R21.3F0	Interno UC15.3-GP
R21.4F0	Interno UC15.4-GP
R21.5F0	Interno UC15.5-GP
R21.6F0	Interno UC15.6-GP
R21.7F0	Interno UC15.7-GP

Tabella 21: Elenco dei requisiti funzionali in rapporto alle fonti (5)

Requisito	Fonte
R1Q1	Interno QPR001
R2Q1	Interno QPD001
R3Q1	Interno QPR004
R4Q1	Interno QPS005
R5Q1	Interno QPR006
R6Q1	Interno QPR008
R7Q1	Interno QPR009
R8Q1	Interno QPR011
R9Q1	Interno QPR012
R10Q1	Interno QPR013
R11Q2	Interno
R12Q2	Interno
R12.1Q2	Capitolato VE_2018-12-12 _D
R13Q2	Capitolato
R14Q2	Capitolato
R14.1Q2	Capitolato

Tabella 22: Elenco dei requisiti di qualità in rapporto alle fonti (1)

Requisito	Fonte
R15Q2	Capitolato
R15.1Q2	Capitolato
R15.2Q2	Capitolato
R15.3Q2	Capitolato
R16Q1	Capitolato
R16.1Q2	Capitolato
R17Q2	Capitolato
R17.1Q2	Capitolato
R17.3Q2	Capitolato
R17.3.1Q2	Capitolato
R17.3.2Q2	Capitolato

Tabella 23: Elenco dei requisiti di qualità in rapporto alle fonti (2)

Requisito	Fonte
R1V2	Capitolato
R1.1V0	Capitolato
R2V2	Capitolato
R2.1V0	Capitolato
R3V2	Capitolato
R3.1V2	Capitolato
R4V1	Capitolato
R5V1	Capitolato
R6V0	Interno
R7V0	Interno
R8V0	Interno
R9V0	Interno
R10V0	Interno
R11V2	Capitolato
R12V2	Capitolato
R13V2	Capitolato
R14V2	Capitolato
R15V2	Capitolato

Tabella 24: Elenco dei requisiti di vincolo in rapporto alle fonti

4.5 Riepilogo

Tipologia	Obbligatori	Desiderabili	Opzionali
Di funzionalità	24	0	38
Di qualità	17	11	0
Di vincolo	9	2	7

Tabella 25: Riepilogo dei requisiti

A Approfondimenti tecnologici

A.1 Webhook

I webhook sono un pattern HTTP leggero che fornisce un modello publisher/subscriber semplice per collegare tra loro Web API. Quando si verifica un evento in un servizio, viene inviata una notifica ai sottoscritti registrati sotto forma di una richiesta HTTP. La richiesta contiene informazioni sull'evento che rende possibile per il ricevitore agire di conseguenza. Le tecnologie Redmine e GitLab li integrano e utilizzando una specifica porta, dove i nostri Producer sono in ascolto, viene inviato un file `JSONG` contenente tutte le informazioni riguardanti l'evento. Questa tecnologia ha forma tale da fornirci collezioni di dati organizzati, facilitandoci così la lettura e la scrittura.

Seguono `DIZIONARI PYTHONG` chiamati genericamente `obj` contenenti i campi dei webhook di nostro interesse:

- GitLab Issue Event:

```
1  obj ["object_kind"]
2  obj ["project"] ["id"]/obj ["object_attributes"] ["project_id"]
3  obj ["project"] ["name"]
4  obj ["assignees"] [k] ["nome utente"]
5  obj ["object_attributes"] ["action"]
6  obj ["object_attributes"] ["description"]
7  obj ["labels"] [k] ["title"]
8  obj ["labels"] [k] ["project_id"]
9  obj ["changes"] ["title"]
10 obj ["changes"] ["labels"] ["previous"] [k] ["title"]
11 obj ["changes"] ["labels"] ["current"] [k] ["title"]
```

- GitLab Push Event:

```
1  obj ["object_kind"]
2  obj ["utente_id"]
3  obj ["utente_nome utente"]
4  obj ["project_id"]
5  obj ["commits"] [k] ["id"]
6  obj ["commits"] [k] ["message"]
7  obj ["commits"] [k] ["timestamp"]
8  obj ["commits"] [k] ["author"] ["name"]
9  obj ["total_commits_count"]
```

- Redmine Issue Event:

```
1  obj ["payload"] ["issue"] ["priority"] ["name"]
2  obj ["payload"] ["issue"] ["tracker"] ["name"]
3  obj ["payload"] ["issue"] ["subject"]
4  obj ["payload"] ["issue"] ["status"] ["name"]
5  obj ["payload"] ["issue"] ["description"]
6  obj ["payload"] ["issue"] ["project"] ["id"]
7  obj ["payload"] ["issue"] ["project"] ["name"]
8  obj ["payload"] ["issue"] ["action"]
```

A.2 Redmine

Ciascuna istanza di Redmine permette l'utilizzo di webhook¹ che inviano segnalazioni al Producer alla modifica del progetto. Queste vengono ricevute dal server tramite una componente che resta

¹Riferirsi alla voce "3. Webhook di Redmine" alla sezione §1.4.2

in ascolto, aggiornando in base a ciò che riceve dai dati presenti sul Gestore Personale e inoltrando i messaggi ai Consumer interessati.

A.3 GitLab

Ciascuna istanza di GitLab, online o in un server locale, mette a disposizione la configurazione di webhook² che, alla modifica di una repository, manda un messaggio con le informazioni relative a quell'evento specifico. Questo messaggio arriva a una componente applicativa capace di aggiornare i dati presenti nel Gestore Personale e, come per Redmine, inoltrare le notifiche ai Consumer interessati.

A.4 Apache Kafka

Apache Kafka è un software OPEN SOURCE_G che permette la lettura e la scrittura di messaggi su differenti canali di comunicazioni per i dati. Questi messaggi arrivano dai Producer che ricevono le notifiche di applicazioni di terze parti immettendole nel Broker. Questo le elabora analizzandone il contenuto e contrassegnandole con Topic che verranno utilizzati per l'inoltro ai Consumer e successivamente agli utenti finali, i quali possono abbonarsi a più Topic.

A.5 MongoDB

È un DBMS_G non relazionale, orientato ai documenti, libero e open source. Classificato come un database di tipo NoSQL, MongoDB si allontana dalla struttura tradizionale basata su tabelle dei database relazionali in favore di documenti in stile JSON_G con schema dinamico, rendendo l'integrazione di dati di alcuni tipi di applicazioni più facile e veloce.

A.6 Telegram

Telegram permette l'interazione in maniera automatica con gli utenti tramite bot³ che possono essere configurati per mandare messaggi ricevuti da strumenti di terze parti, in questo caso Butterfly. Il Consumer interroga il Broker per acquisire i messaggi da inoltrare e li trasmette al bot di Telegram, che si occuperà dell'invio all'utente finale.

A.6.1 BotFather

Per mandare messaggi tramite Telegram all'utente finale, è necessario utilizzare un bot creato con BotFather. Quest'ultimo è esso stesso un bot offerto da Telegram che permette la creazione e la configurazione di bot che possono fare da tramite tra utenti e servizi esterni dai quali ricevere, o a cui mandare, informazioni o comandi.

A.7 Email

Per inoltrare le Email agli utenti finali utilizziamo un Consumer associato che sfrutta il SERVER SMTP_G di posta in modo tale da poter inoltrare i messaggi ottenuti dal Broker all'indirizzo specificato.

A.8 Docker

L'azienda ci consiglia di utilizzare DOCKER_G per la semplicità di utilizzo e per l'adattamento all'architettura a microservizi. La configurazione avverrà tramite un DOCKERFILE_G in cui verranno specificate informazioni come sistema operativo, script di avvio, numero di istanze ed altri parametri specifici.

²Riferirsi alla voce "4. Webhook di GitLab alla sezione" §1.4.2

³Riferirsi alla voce "5. Bot di Telegram" alla sezione §1.4.2

A.8.1 Container software

Un $\text{CONTAINER SOFTWARE}_G$ simula un ambiente virtuale dove è possibile testare e mantenere le proprie applicazioni, permettendo di aumentare l'efficienza riducendone i costi e simulando l'esecuzione di sistema operativo su una macchina con RISORSE_G condivise.

A.8.1.1 Differenza tra container e macchina virtuale

A differenza delle $\text{MACCHINE VIRTUALI}_G$, dove lo stato dell'ambiente viene salvato su disco, occupando memoria, i container si adattano in maniera più performante all'applicativo richiesto, in quanto il loro scopo è quello di massimizzare la quantità delle applicazioni in esecuzione riducendo al minimo il numero delle macchine per eseguirla. Sono quindi più leggeri, occupando meno memoria su disco e impiegando meno risorse.

A.8.2 DockStation

Per la gestione dei container in locale è stato possibile utilizzare DOCKSTATION_G in quanto fornisce una GUI dalla quale si possono effettuare i comandi impartiti a Docker via Terminale. È inoltre possibile avere la visione dello stato di ciascun container e vedere in tempo reale i log che questo genera, oltre che a poter controllare le risorse utilizzate da ciascun container.

A.9 Rancher

Per la gestione dei container in remoto, la proponte Imola Informatica ci ha messo a disposizione un CLUSTER_G con due server sui quali è stato installato Rancher. Questo è un software di gestione di oggetti di KUBERNETES_G che fornisce un'interfaccia grafica più ricca rispetto a quella di Dockstation e funzionalità maggiori. Da qui possiamo quindi gestire i nostri container installando le immagini direttamente da DOCKERHUB_G senza aver bisogno di file di configurazione come ad esempio quello necessario al docker-compose.


<div>  <div> butterfly Default </div> <div> Workloads Catalog Apps Resources Namespaces Members </div> </div>			
<div> Workloads Load Balancing Service Discovery Volumes Pipelines </div>			
<div> Redeploy Pause Orchestration Download YAML Delete </div>			
<div> <input type="checkbox"/> State Name Image </div>			
Namespace: gitlab			
<input type="checkbox"/>	▶ Active	gitlab 30001/tcp	gitlab/gitlab-ce:latest 1 Pod / Created 14 days ago
Namespace: kafka			
<input type="checkbox"/>	▶ Active	kafka-kafka	confluentinc/cp-kafka:4.0.1-1 1 Pod / Created 16 days ago
<input type="checkbox"/>	▶ Active	kafka-kafka-rest	confluentinc/cp-kafka-rest:4.0.0 1 Pod / Created 16 days ago
<input type="checkbox"/>	▶ Active	kafka-kafka-topics-ui 80/http, 31140/tcp	landoop/kafka-topics-ui:latest 1 Pod / Created 16 days ago
<input type="checkbox"/>	▶ Active	kafka-manager 9000/tcp	sheepkiller/kafka-manager 1 Pod / Created 14 days ago
<input type="checkbox"/>	▶ Active	kafka-schema-registry	confluentinc/cp-schema-registry:4.0.1 1 Pod / Created 16 days ago
<input type="checkbox"/>	▶ Active	kafka-zookeeper	confluentinc/cp-zookeeper:4.1.1 1 Pod / Created 16 days ago
Namespace: redmine			
<input type="checkbox"/>	▶ Active	postgres-redmine	postgres:11-alpine 1 Pod / Created 16 days ago
<input type="checkbox"/>	▶ Active	redmine 30000/tcp	redmine:3.3.9 1 Pod / Created 16 days ago

Figura 21: Alcuni servizi attivi su Rancher