



Commandline Team - Progetto "TuTourSelf"

Allegato Tecnico di Product Baseline

Versione	2.0.0
Approvazione	Nicola Agostini
Redazione	Nicola Agostini Alberto Battistini Giulia Corò Marco Giollo Giovanni Motterle Daniele Penazzo Michele Tagliabue
Verifica	Daniele Penazzo Michele Tagliabue Giovanni Motterle
Stato	Approvato
Uso	Interno
Destinato a	Commandline Team Prof. Tullio Vardanega Prof. Riccardo Cardin

Descrizione

Questo documento riporta i diagrammi UML dell'architettura del sistema, oltre ai design pattern applicati e contestualizzati alle scelte effettuate.

Contenuto

1	Changelog	3
2	Struttura Generale Del Progetto	4
3	Struttura Generale Frontend	5
3.1	Visione generale	5
3.2	Interfacce Utente	9
4	Registrazione	13
4.1	Registrazione: back-end	13
4.2	Registrazione: front-end	19
5	Login	22
5.1	Login: back-end	22
5.2	Login: front-end	26
6	Modifica del profilo	29
6.1	Modifica del profilo: back-end	29
6.2	Modifica del profilo: front-end	34
7	Chat	39
7.1	Chat: back-end	39
7.2	Chat: front-end	46
8	Proposta	52
8.1	Proposta: back-end	52
8.2	Proposta: front-end	56
9	Eliminazione di un accordo	61
9.1	Eliminazione di un accordo: back-end	61
9.2	Eliminazione di un accordo: front-end	64
10	Rilascio dei feedback	66
10.1	Feedback: back-end	66
10.2	Feedback: front-end	70
11	Apertura di una controversia	74
11.1	Controversia: back-end	74
11.2	Controversia: front-end	78

12 Gestione degli eventi	81
12.1 Gestione degli eventi: back-end	81
12.2 Gestione degli eventi: front-end	86
13 Ricerca per keywords	89
13.1 Ricerca per keywords: back-end	89
13.2 Ricerca per keywords: front-end	92
14 Ricerca su mappa	94
14.1 Ricerca su mappa: back-end	94
14.2 Ricerca su mappa: front-end	96
14.3 Ricerca locali su mappa	99
14.4 Ricerca eventi su mappa	101
15 Pannello dell'impiegato	103
15.1 Pannello dell'impiegato: back-end	103
15.2 Pannello dell'impiegato: front-end	109
16 Segnalazione utente	112
16.1 Segnalazione utente: back-end	112
16.2 Segnalazione utente: front-end	115

1 Changelog

Versione	Data	Nominativo	Ruolo	Descrizione
2.0.0	2018-05-09	Nicola Agostini	Responsabile	Approvazione del documento
1.1.0	2018-05-09	Giovanni Motterle	Verificatore	Verifica del documento
1.0.1	2018-05-02	Daniele Penazzo	Progettista	Rimozione Factory, inserimento visuale generale del progetto.
1.0.0	2018-04-05	Giovanni Motterle	Responsabile	Approvazione documento
0.1.0	2018-04-05	Daniele Penazzo Michele Tagliabue	Verificatore	Verifica documento
0.0.9	2018-04-05	Giovanni Motterle	Responsabile	Aggiunta diagrammi login
0.0.8	2018-04-05	Nicola Agostini	Progettista	Aggiunti diagrammi frontend proposta, ui e segnalazione utente
0.0.7	2018-04-05	Michele Tagliabue	Progettista	Aggiunti diagrammi controversia, accordo e proposta (back-end)
0.0.6	2018-04-05	Daniele Penazzo	Progettista	Inserimento e correzione diagrammi chat, mappe e registrazione
0.0.5	2018-04-04	Giulia Corò	Progettista	Aggiunta parte front-end Struttura generale §3, Chat §7.2 e Registrazione §4.2
0.0.4	2018-04-04	Alberto Battistini	Progettista	Aggiunta parte back-end e front-end Ricerca su mappa
0.0.3	2018-04-04	Daniele Penazzo Marco Giollo	Progettista	Aggiunta parte BE Chat e Registrazione
0.0.2	2018-04-05	Marco Giollo	Progettista	Creazione di tutte le sezioni ed inserimento dei requisiti che esse soddisfano.
0.0.1	2018-04-04	Daniele Penazzo	Progettista	Creazione Template

Tabella 1: Changelog di questo documento

2 Struttura Generale Del Progetto

Questa sezione contiene una visione "a volo d'uccello" del progetto nella propria completezza, dimostrando come le varie componenti vadano ad integrarsi tra loro e come si interfacciano con moduli esterni.

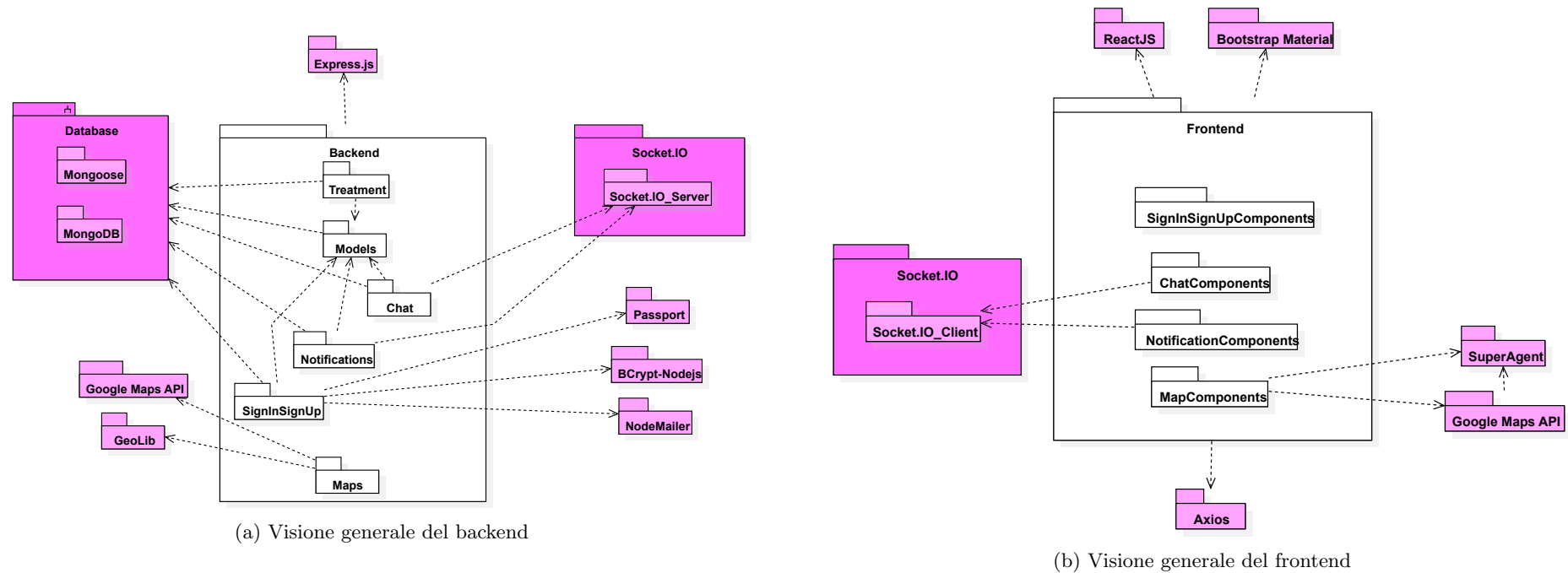


Immagine 1: Visione generale del progetto

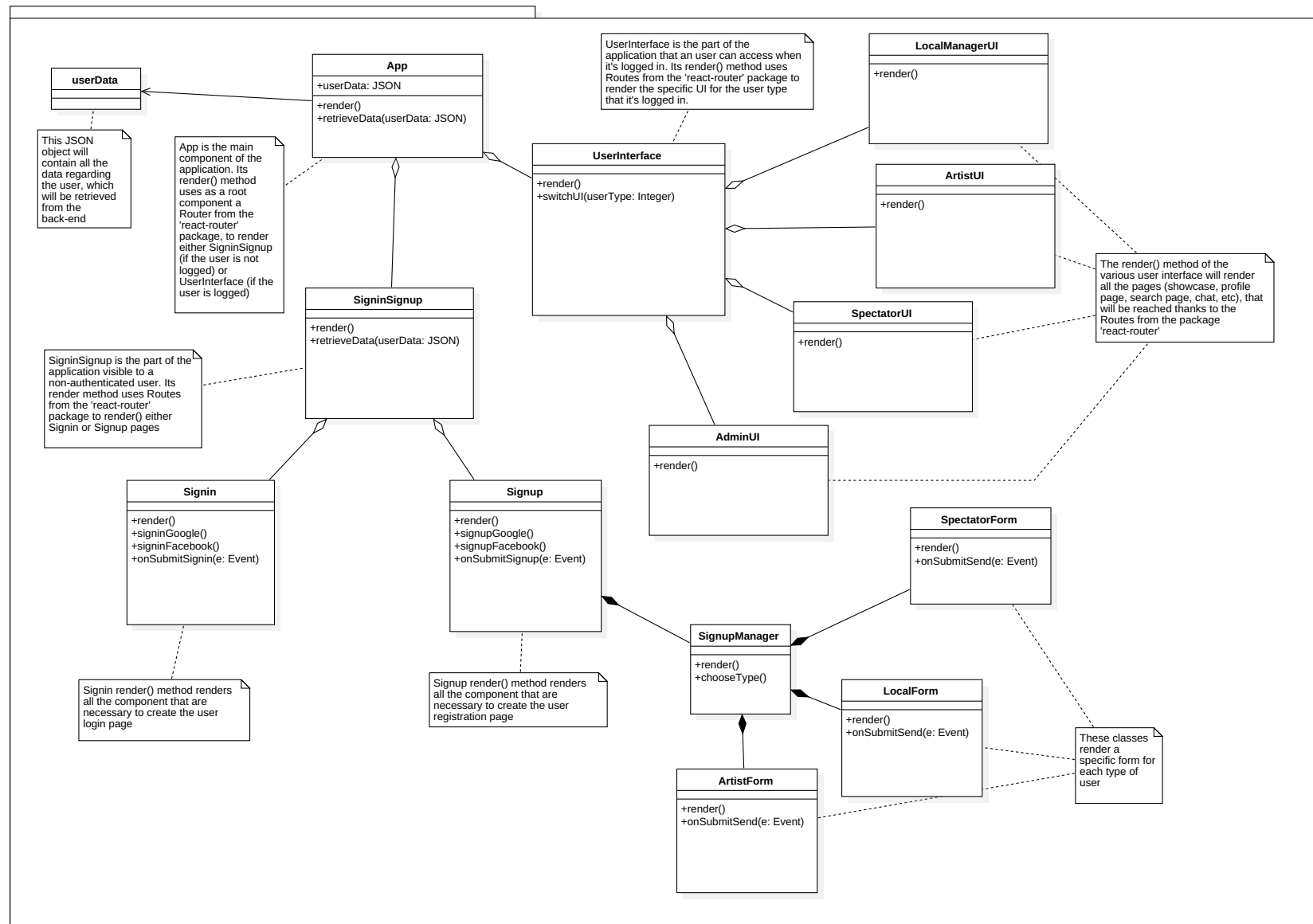
3 Struttura Generale Frontend

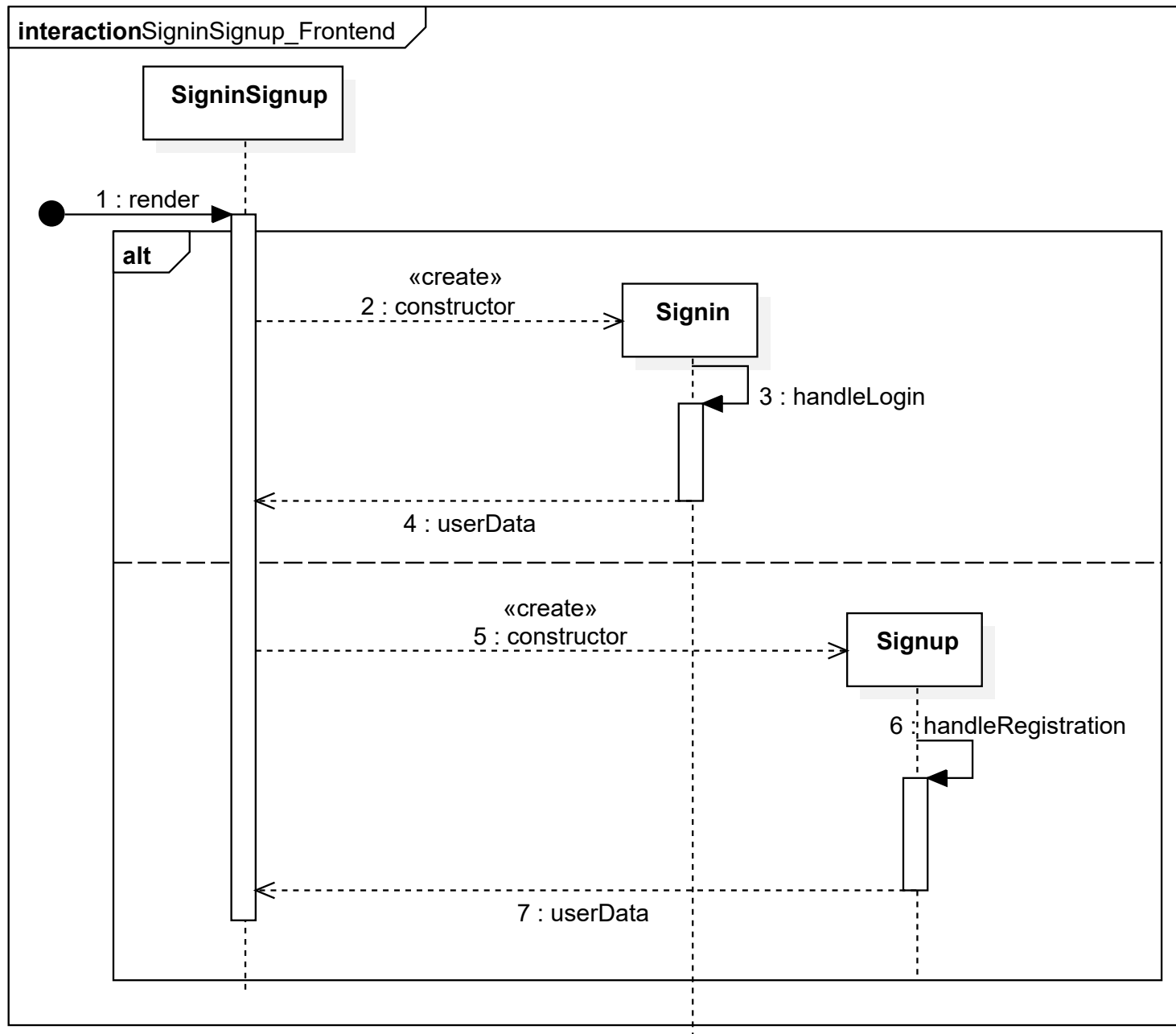
Di seguito è riportato il diagramma che rappresenta la struttura utilizzata per le classi e le sequenze di interazioni del frontend.

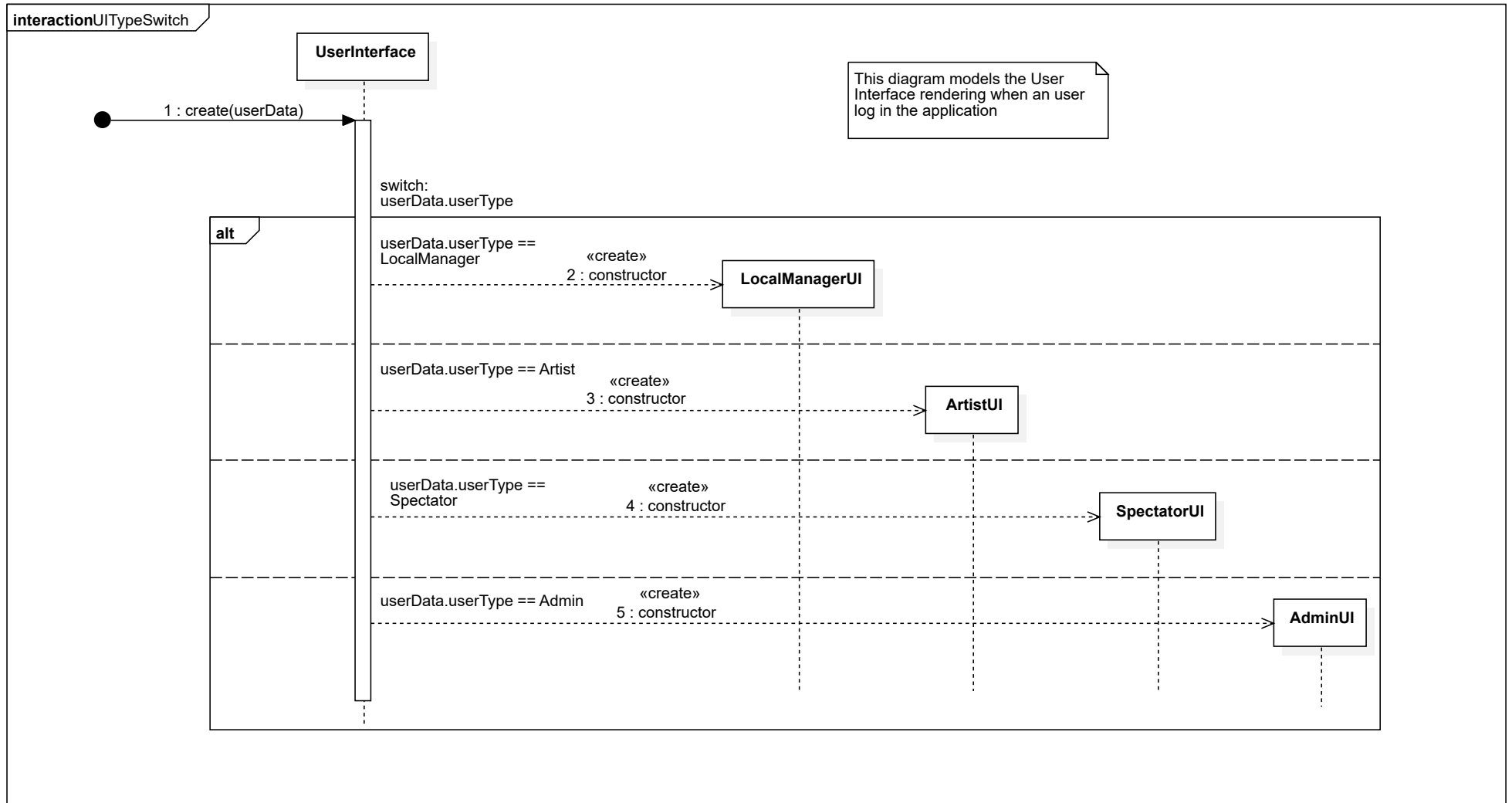
3.1 Visione generale

La componente principale dell'applicazione è la classe *App* di React. Essa usa nel suo metodo *render()* l'elemento *Router* del package 'react-router' di React, per instradare le varie pagine dell'applicazione. Nello stato della componente *App* sono conservati i dati dell'utente che sta interagendo con il sistema.

Nel caso l'utente non abbia effettuato la login, viene instradato tramite una *Route* sulla componente *SigninSignup*, che allo stesso modo instrada l'utente su *Signin* o su *Signup*, tramite il click di un apposito link sull'AppBar. Se invece l'utente effettua o ha effettuato la login, viene instradato, sempre usando una *Route*, sulla componente *UserInterface*, che allo stesso modo, tramite le informazioni sul tipo di utente che si trovano nel JSON *UserData*, identifica di che tipo di utente si tratta e ne renderizza la rispettiva interfaccia (*ArtistUI*, *LocalManagerUI*, *SpectatorUI* o *AdministratorUI*).

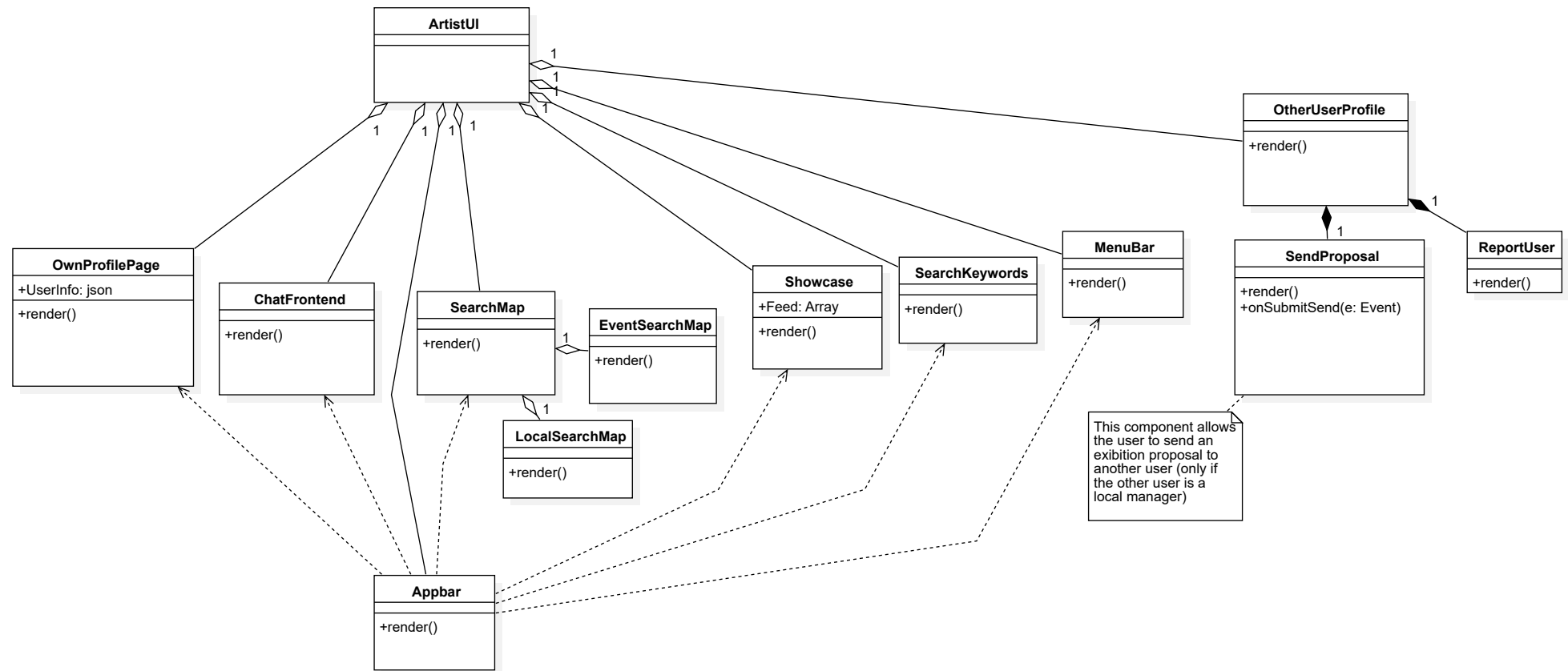


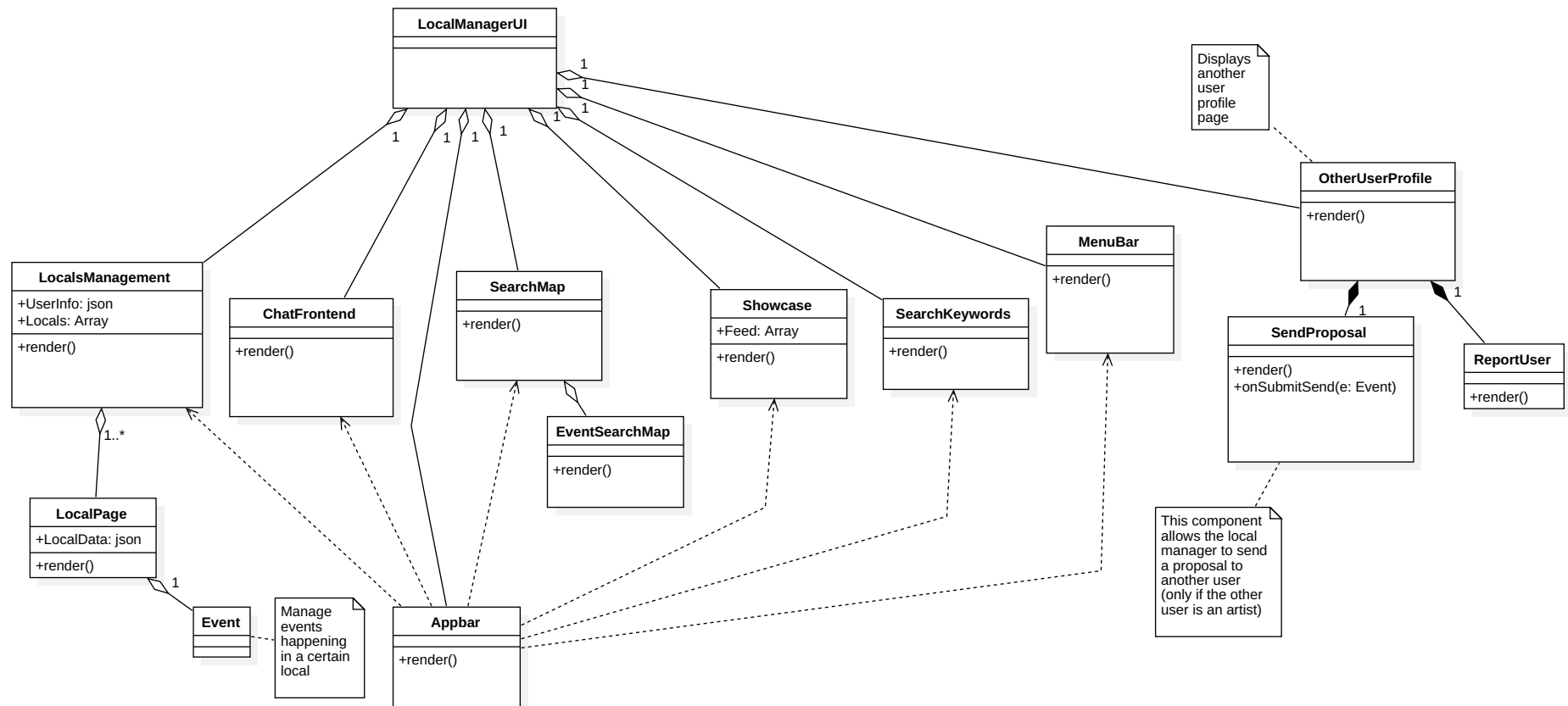


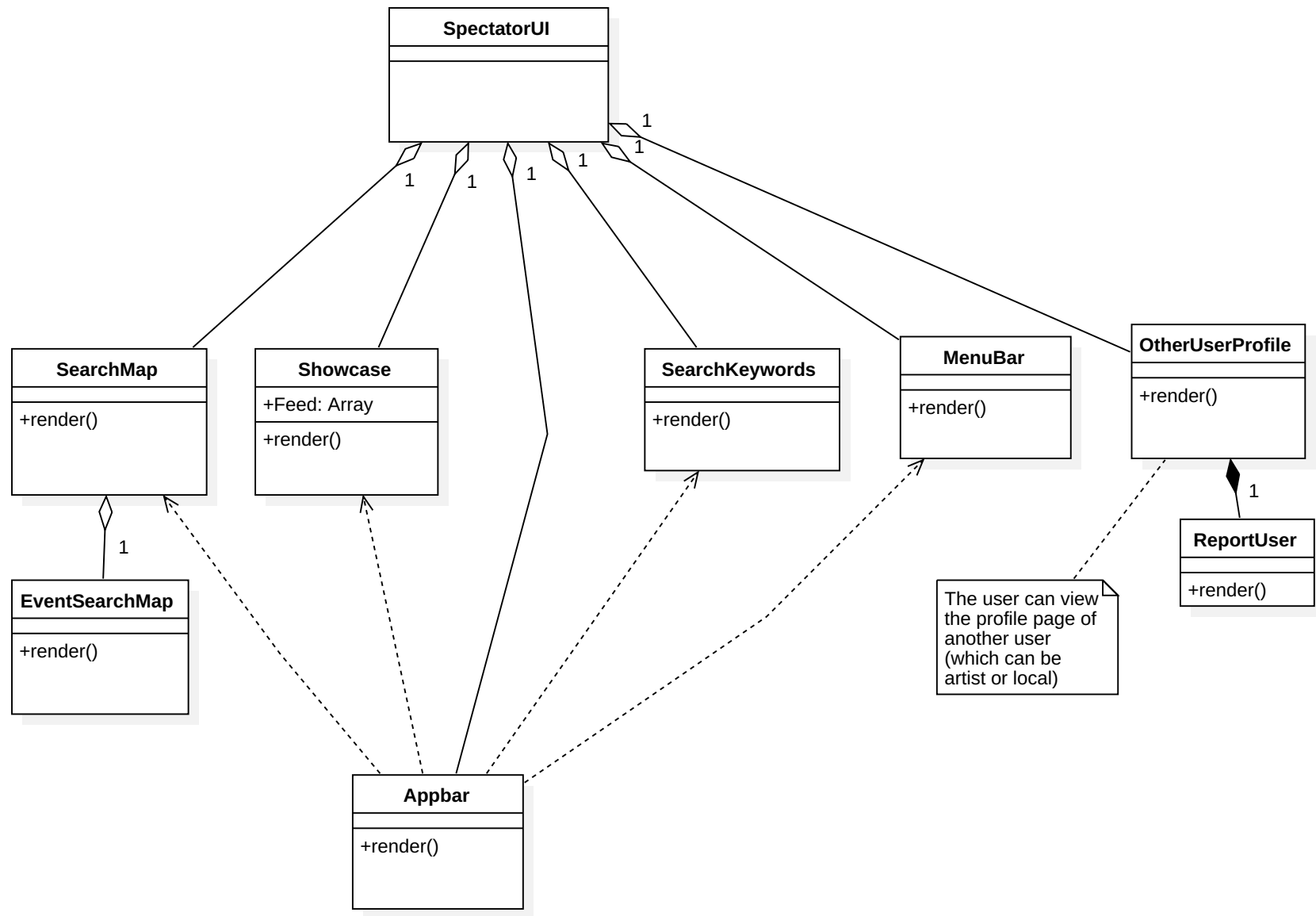


3.2 Interfacce Utente

Di seguito vengono riportati i diagrammi delle classi riguardanti le principali interfacce utente, ossia, in ordine, *ArtistUI*, *LocalManagerUI* e *SpectatorUI*. L'interfaccia amministratore non viene qui riportata perché sarà presentata in seguito con tutte le sue funzionalità in §15.







4 Registrazione

Con lo sviluppo e la progettazione di questo modulo, il gruppo si prende in carico i seguenti requisiti:

- | | | |
|-------------|-------------|-------------|
| • RF0-1 | • RF0-1.4 | • RF1-2.1 |
| • RF0-1.1 | • RF1-1 | • RF1-2.1.2 |
| • RF0-1.1.1 | • RF1-1.1 | • RF1-2.2 |
| • RF0-1.1.2 | • RF1-1.1.1 | • RF1-2.2.1 |
| • RF0-1.2 | • RF1-1.1.2 | • RF1-2.2.2 |
| • RF0-1.2.1 | • RF1-1.2 | • RF1-2.3 |
| • RF0-1.2.2 | • RF1-1.2.2 | • RF1-2.3.1 |
| • RF0-1.3 | • RF1-1.3 | • RF1-2.3.2 |
| • RF0-1.3.1 | • RF1-1.3.2 | |
| • RF0-1.3.2 | • RF1-2 | |

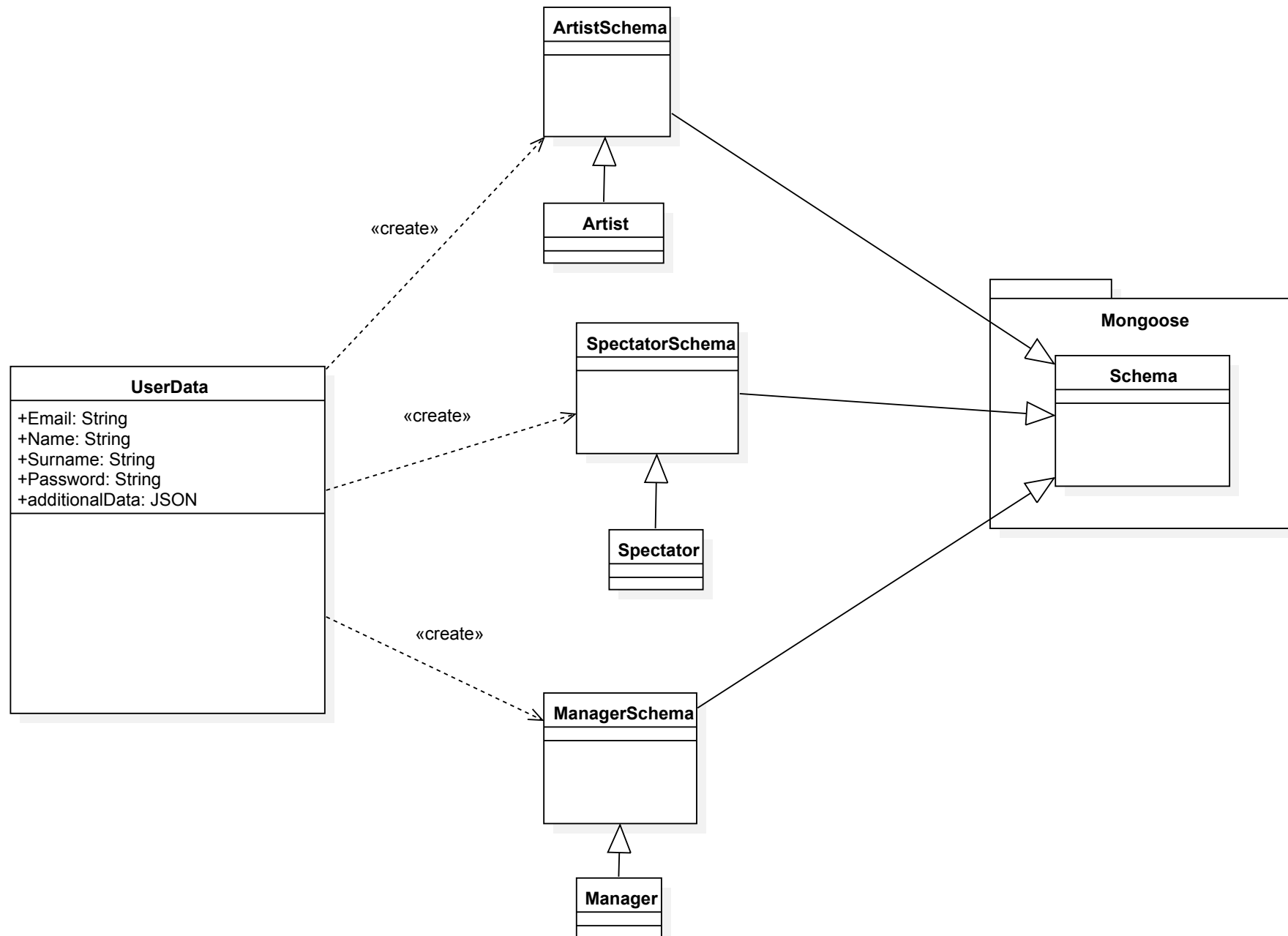
4.1 Registrazione: back-end

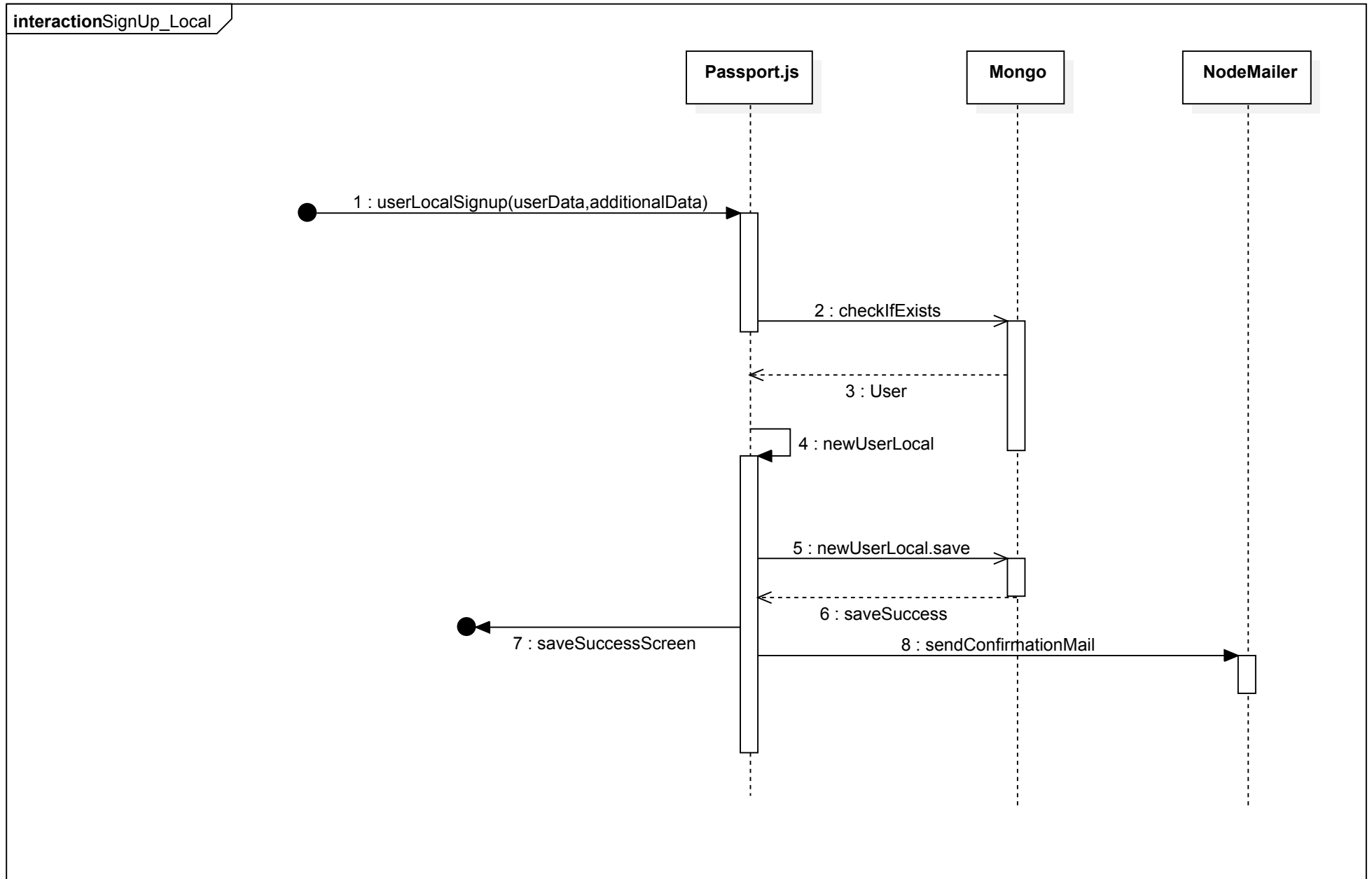
Per la gestione della registrazione abbiamo deciso di usare Passport, che include da sè uno Strategy Pattern interno per la gestione della registrazione e autenticazione tramite diversi protocolli.

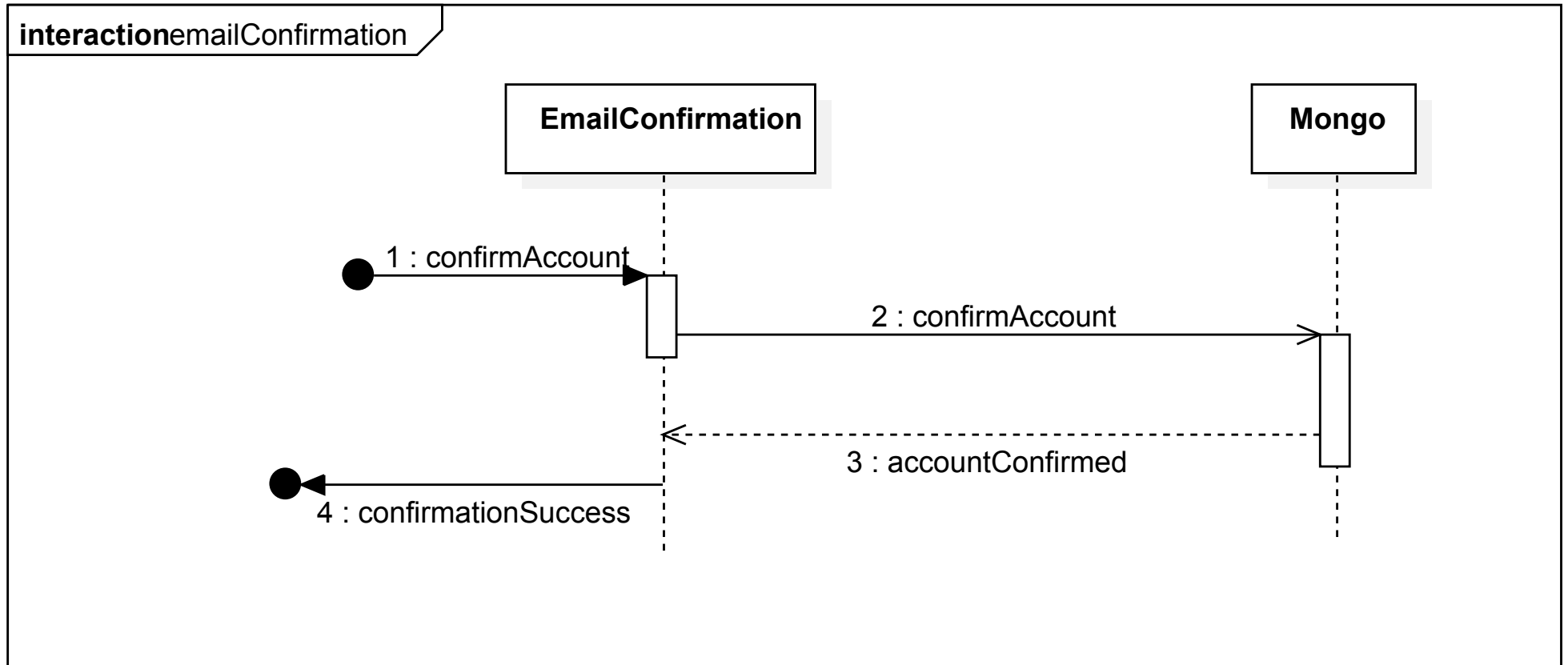
Le strategie usate saranno:

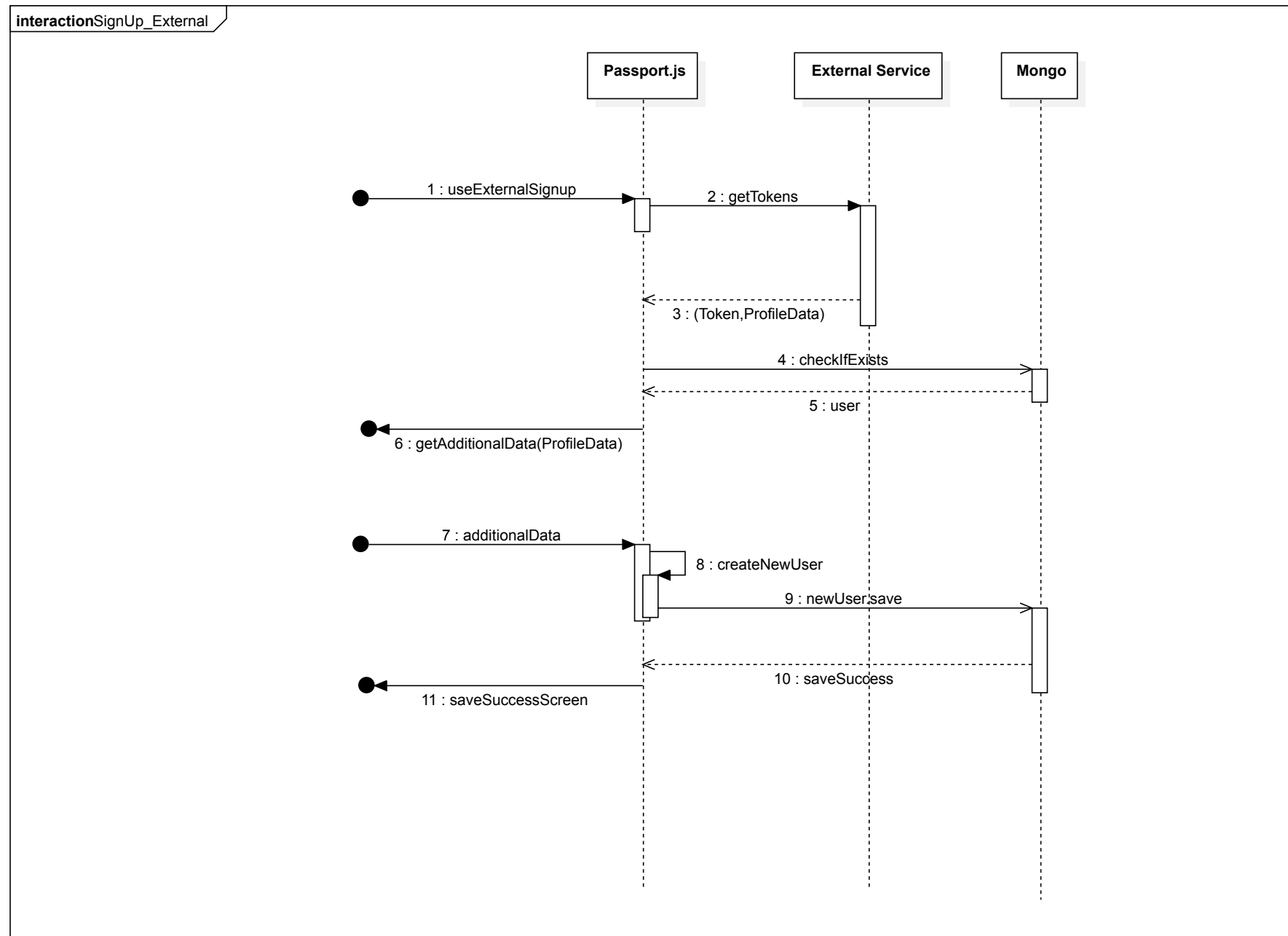
- LocalStrategy
- FacebookStrategy
- GoogleStrategy

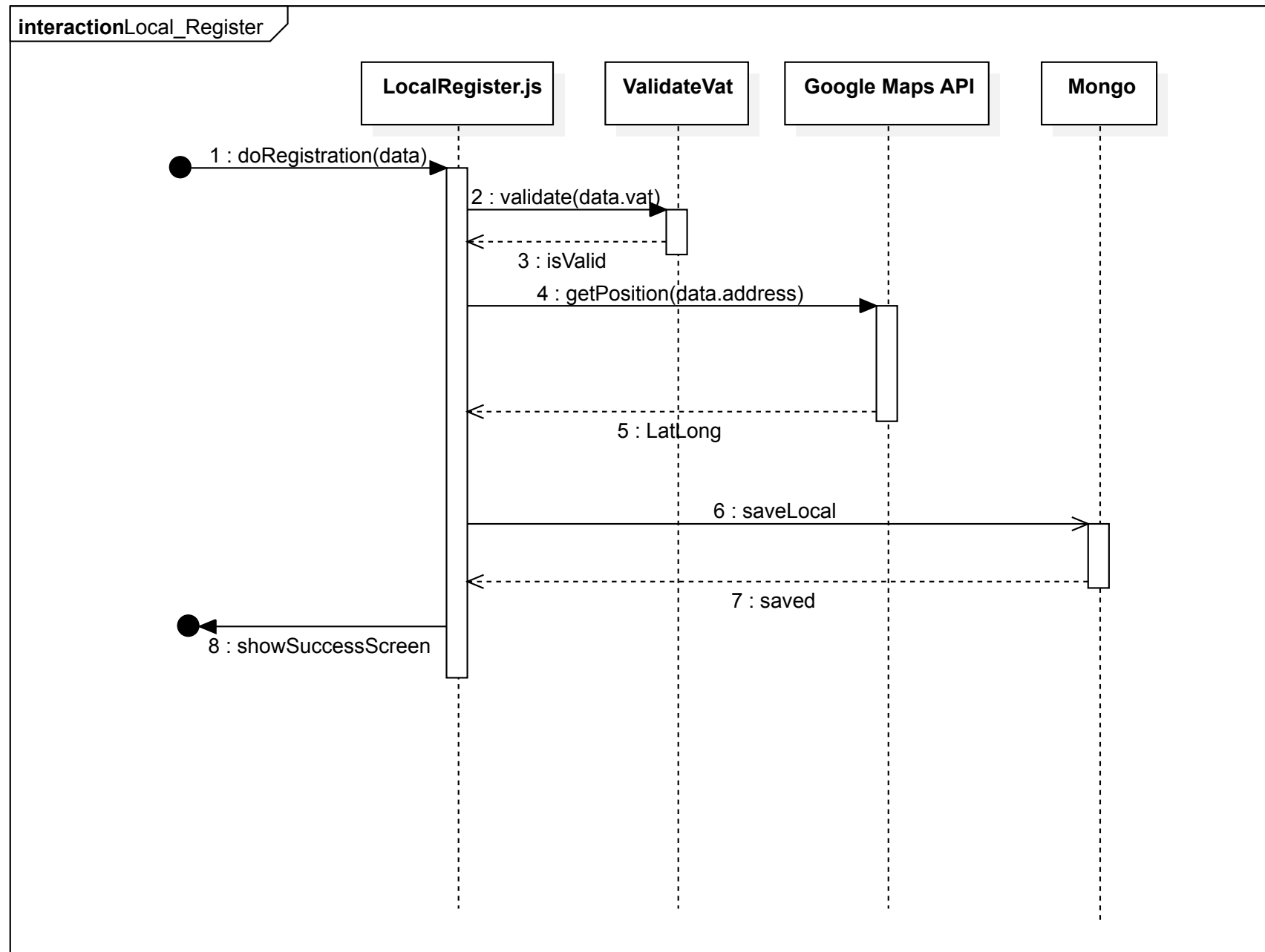
Si farà anche uso dell'Abstract Factory Pattern per la creazione degli utenti a seconda dei dati forniti in fase di registrazione dal front-end. Inoltre si farà uso di MongoDB e BCrypt, rispettivamente per la memorizzazione dei dati e il salted hashing delle password.









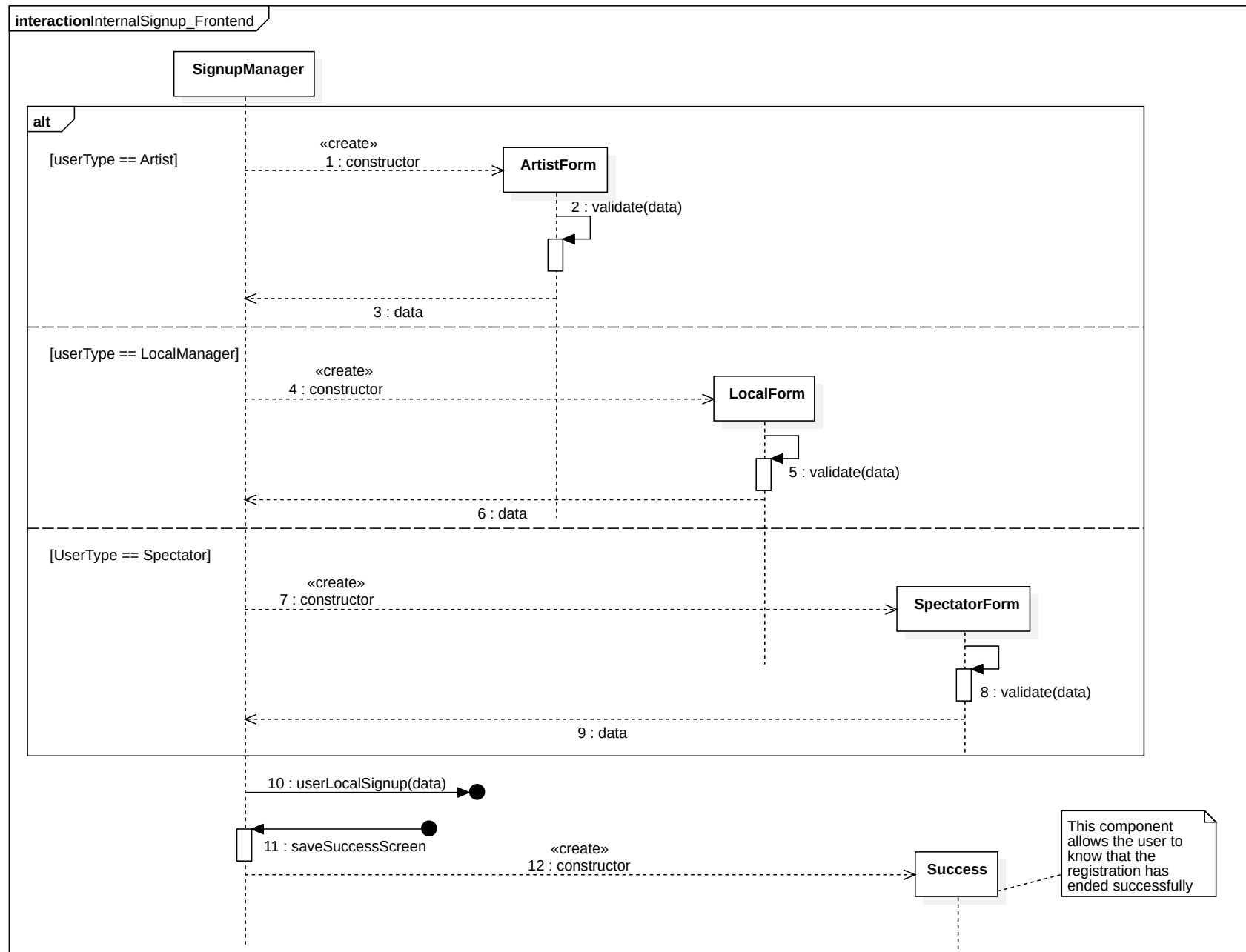


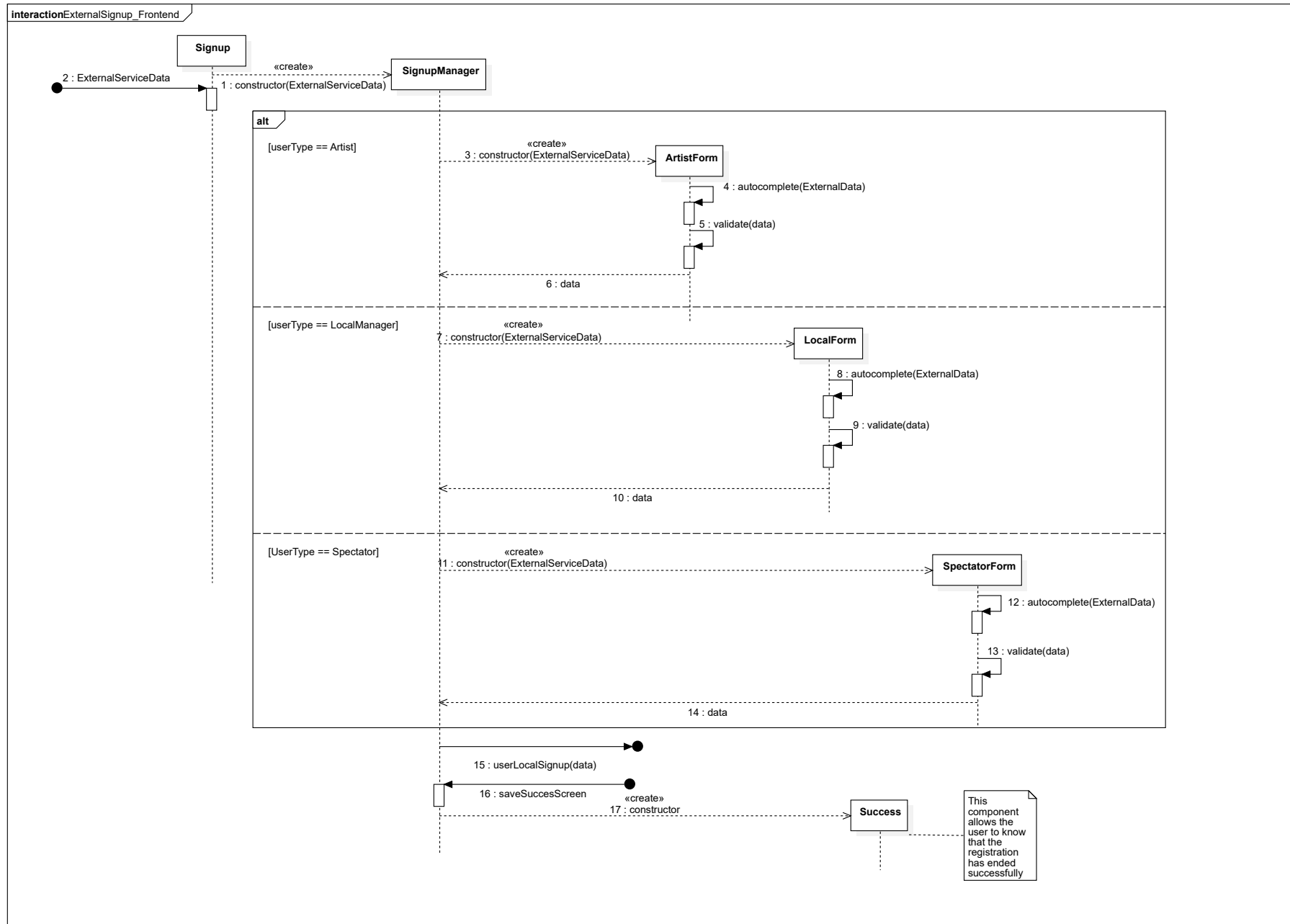
4.2 Registrazione: front-end

Per quanto riguarda i diagrammi di sequenza, la registrazione è stata rappresentata con due diagrammi:

- **La registrazione interna:** verrà innanzitutto richiesto all'utente di inserire quale tipo di registrazione vuole effettuare (come artista, gestore di un locale o spettatore), ed egli verrà quindi reindirizzato ad una form specifica per il suo tipo di utente. Quando verrà cliccato su conferma, i dati della form verranno validati, e se validi inviati al back-end. Se la registrazione verrà validata anche nel lato back-end, verrà quindi mostrato all'utente un messaggio di avvenuta registrazione;
- **La registrazione esterna,** ovvero quella eseguita tramite servizio esterno, nello specifico Facebook e Google: dopo aver recuperato i dati dell'utente dal servizio esterno, verrà richiesto all'utente di inserire quale tipo di registrazione vuole effettuare (come artista, gestore di un locale o spettatore), ed egli verrà quindi reindirizzato ad una form specifica per il suo tipo di utente, dove i dati recuperati dal servizio esterno verranno autocompletati, e l'utente potrà inserire i dati mancanti. Quando verrà cliccato su conferma, tutti i dati della form verranno validati, e se validi inviati al back-end. Se la registrazione verrà validata anche nel lato back-end, verrà quindi mostrato all'utente un messaggio di avvenuta registrazione.

Le dipendenze della componente Signup con le altre componenti del sistema, sono state precedentemente illustrate nel diagramma delle classi in §3.





5 Login

I diagrammi e la progettazione di questo modulo portano alla presa in carico dei requisiti:

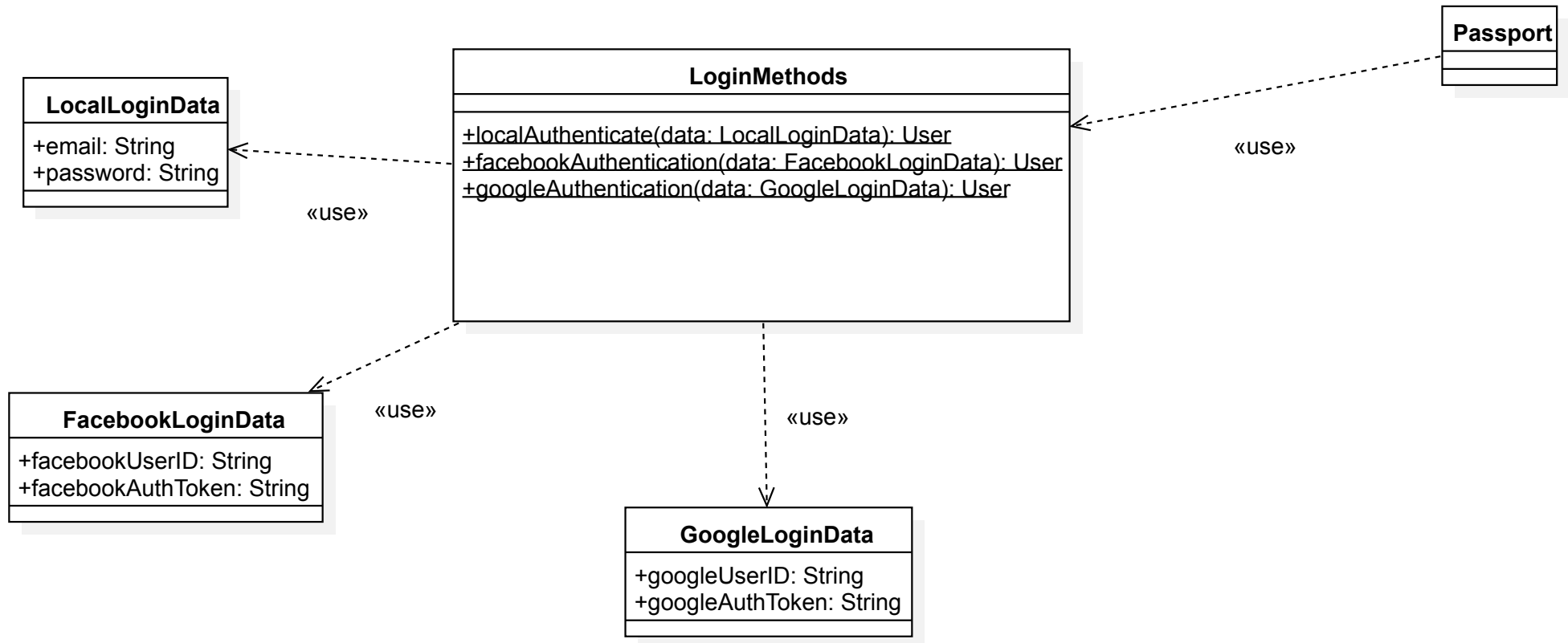
- RF0-2
- RF0-2.1
- RF0-2.2
- RF1-3
- RF1-4

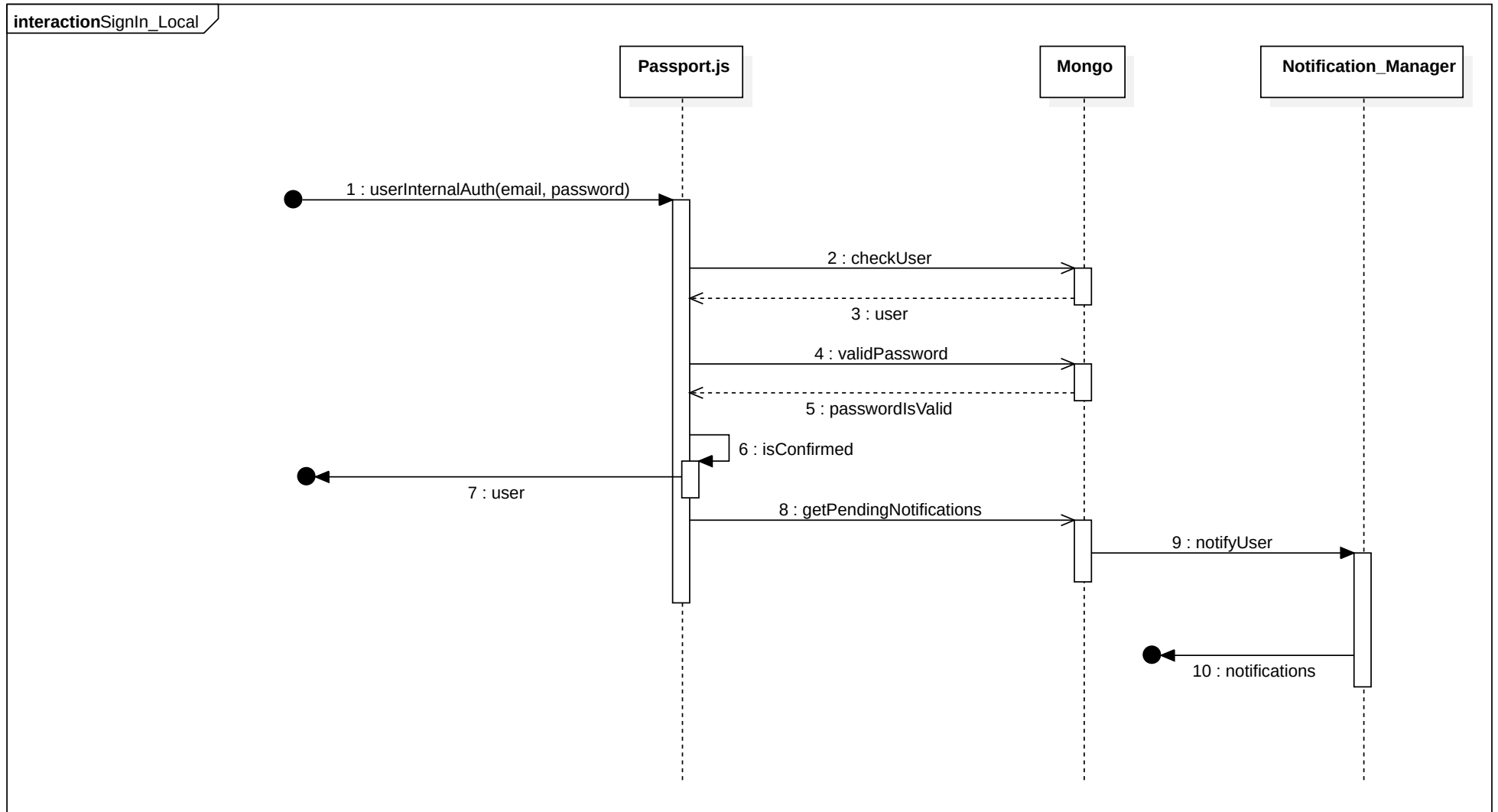
5.1 Login: back-end

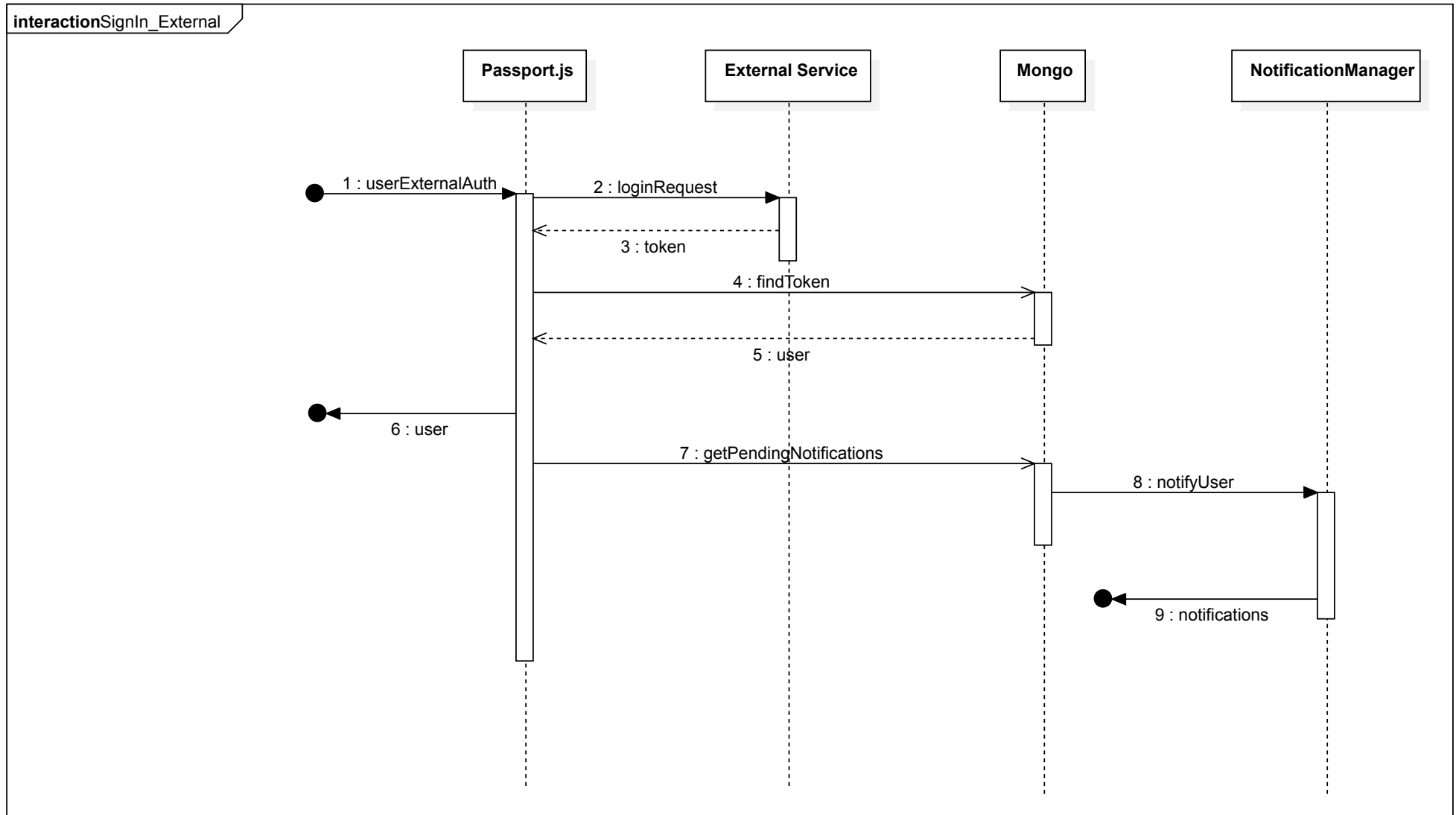
Per la gestione della login abbiamo deciso di utilizzare Passport, che include da sè uno Strategy Pattern interno per la gestione dell'autenticazione tramite diversi protocolli:

- LocalStrategy
- FacebookStrategy
- GoogleStrategy

Viene fatto uso di MongoDB e BCrypt per la verifica dei dati inseriti e il salted hashing delle password.





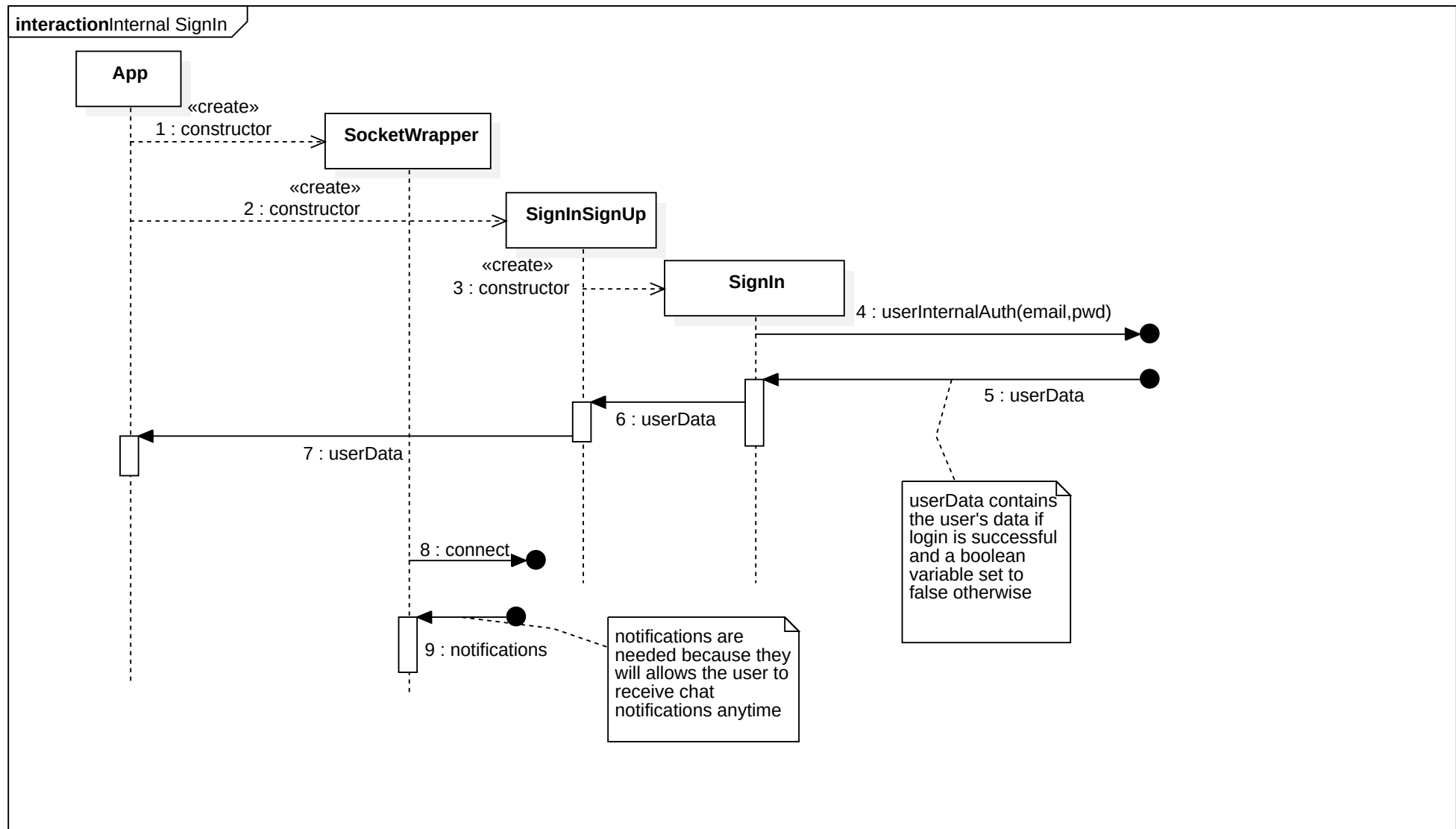


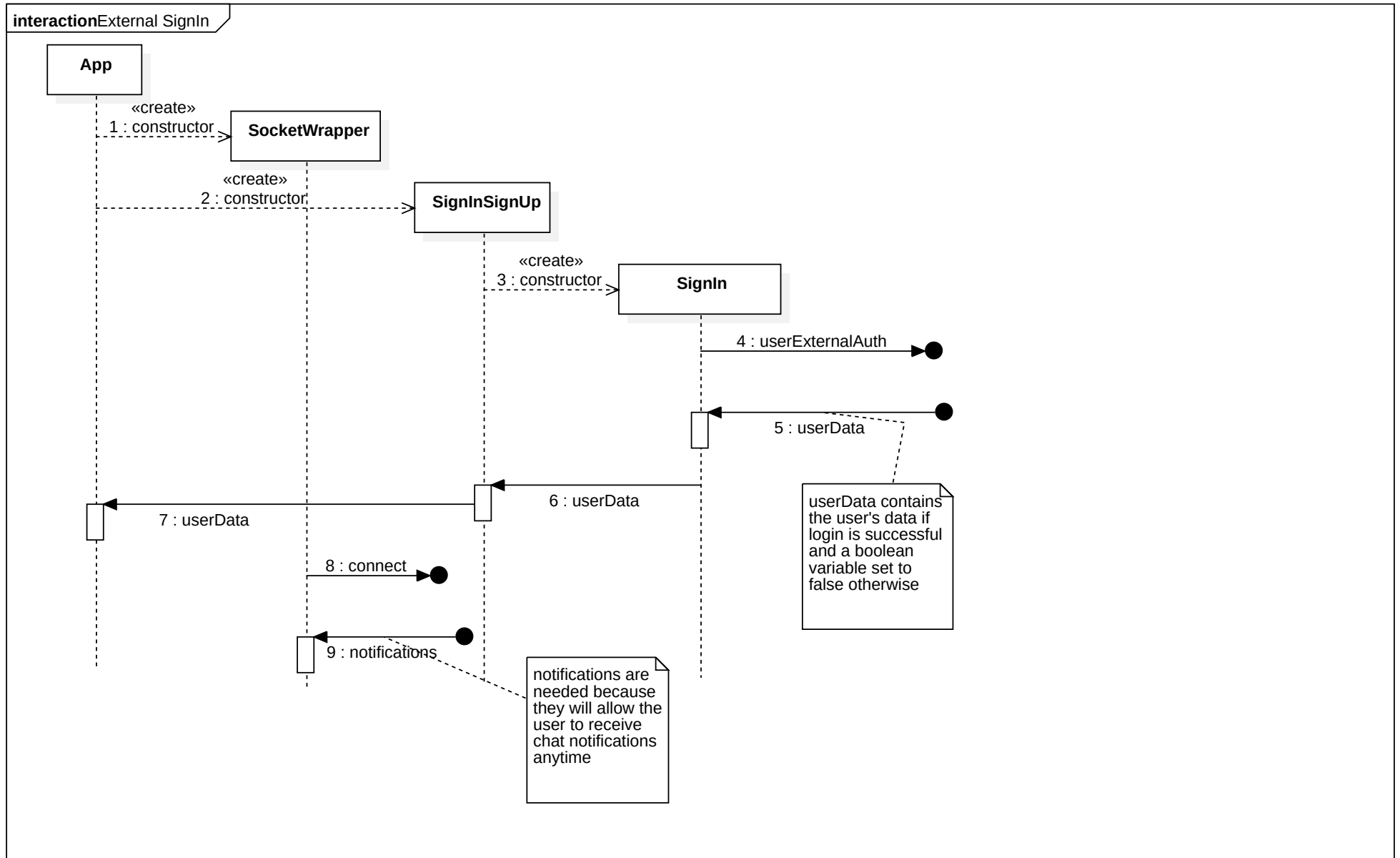
5.2 Login: front-end

Per quanto riguarda i diagrammi di sequenza, la login è stata rappresentata con due diagrammi:

- La login interna;
- La login esterna, ovvero quella eseguita tramite servizio esterno, nello specifico Facebook e Google.

Le dipendenze della componente Signin con le altre componenti del sistema sono state precedentemente illustrate nel diagramma delle classi in §3.





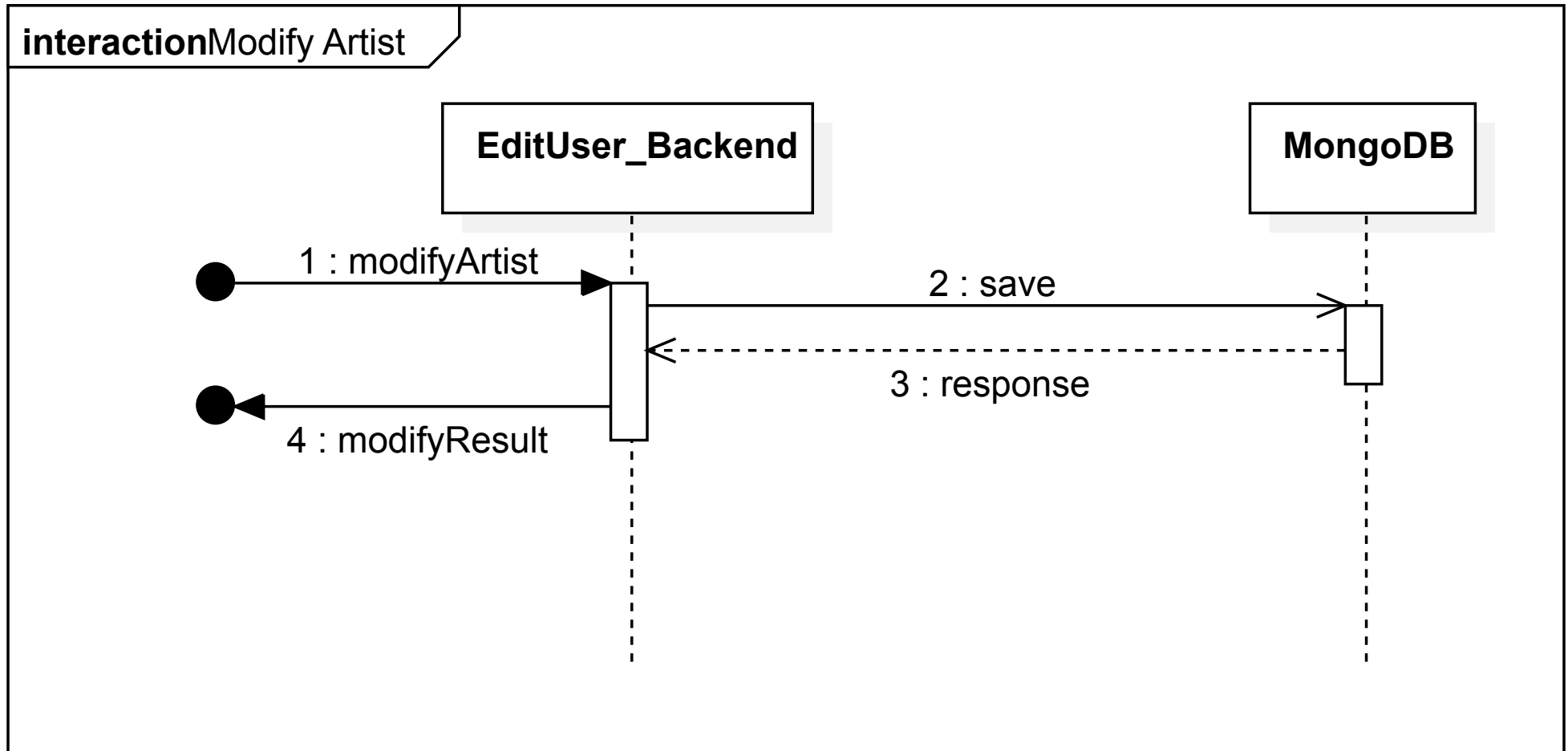
6 Modifica del profilo

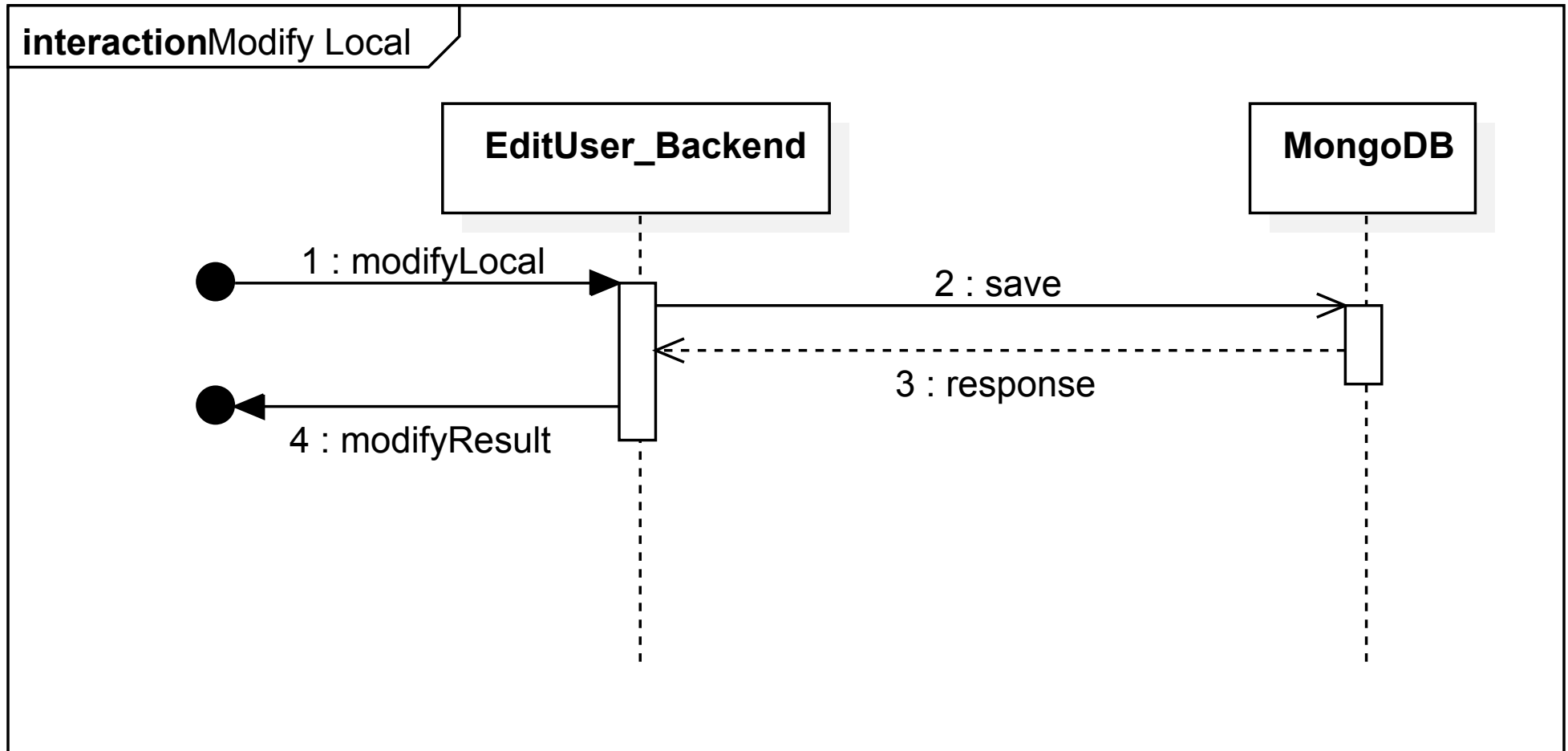
Con lo sviluppo e la progettazione di questo modulo, il gruppo si prende in carico i seguenti requisiti:

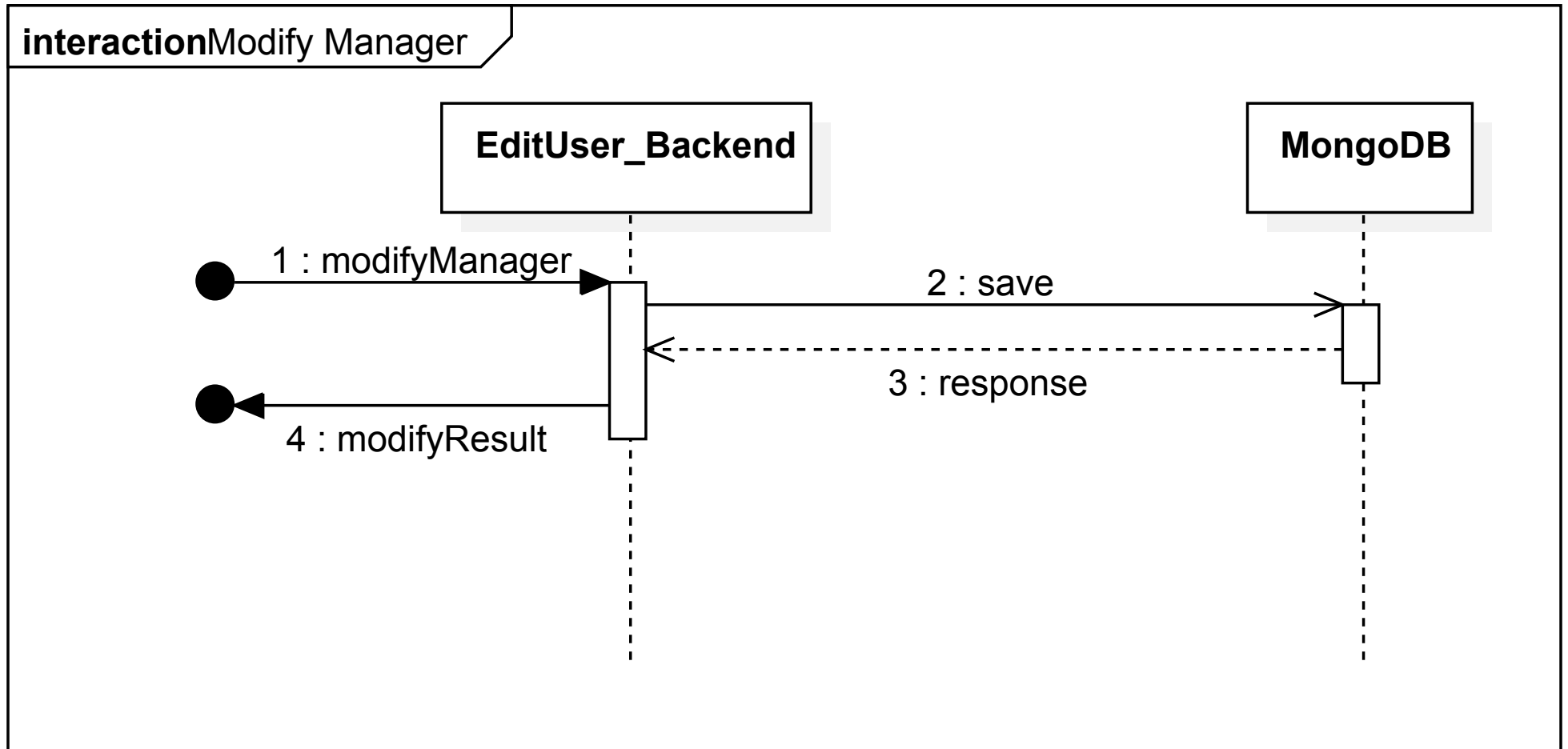
- RF0-10
- RF0-10.1
- RF0-10.1.1
- RF0-10.1.2
- RF0-10.2.1
- RF0-10.2.2
- RF0-10.3
- RF0-11

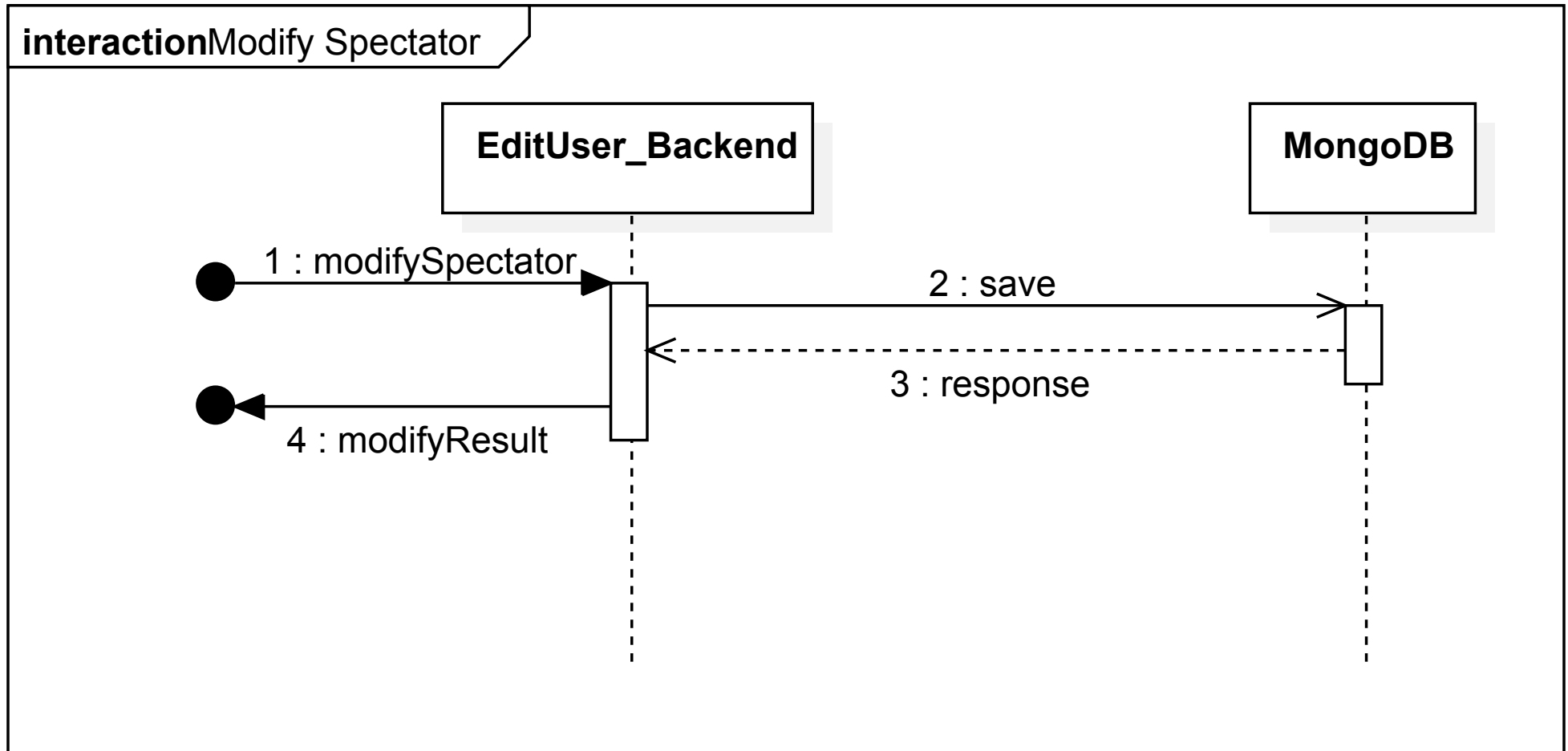
6.1 Modifica del profilo: back-end

A seguito di una richiesta di modifica del profilo, il back-end salverà le modifiche apportate in database. Pertanto viene utilizzato MongoDB per il salvataggio delle modifiche.







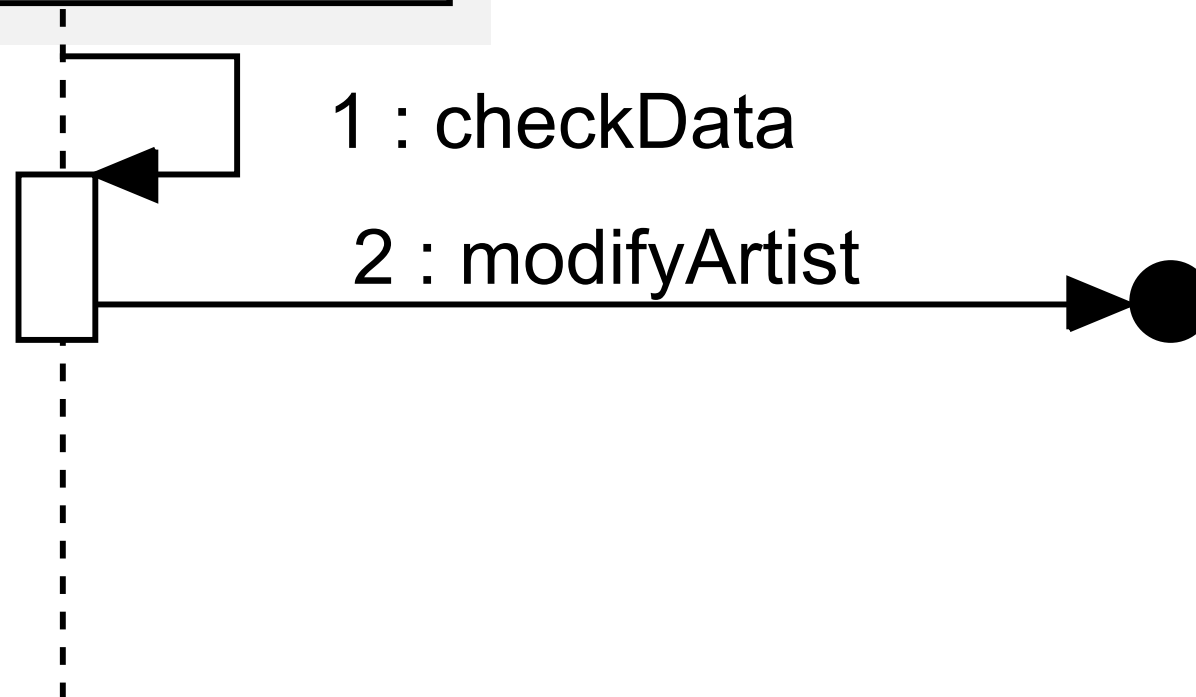


6.2 Modifica del profilo: front-end

A seguito di una richiesta di modifica da parte dell'utente, il front-end verifica la validità dei dati e in caso di esito positivo effettuerà una richiesta di modifica e salvataggio al back-end.

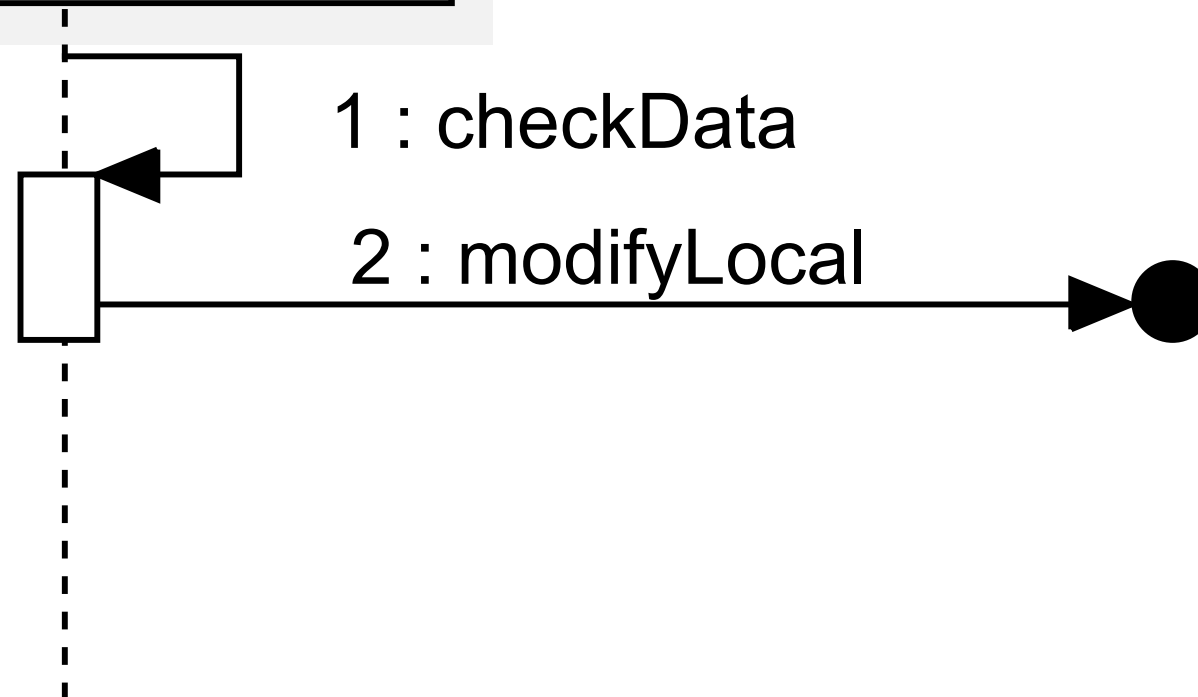
interactionArtist Profile Modification

Modify_frontend



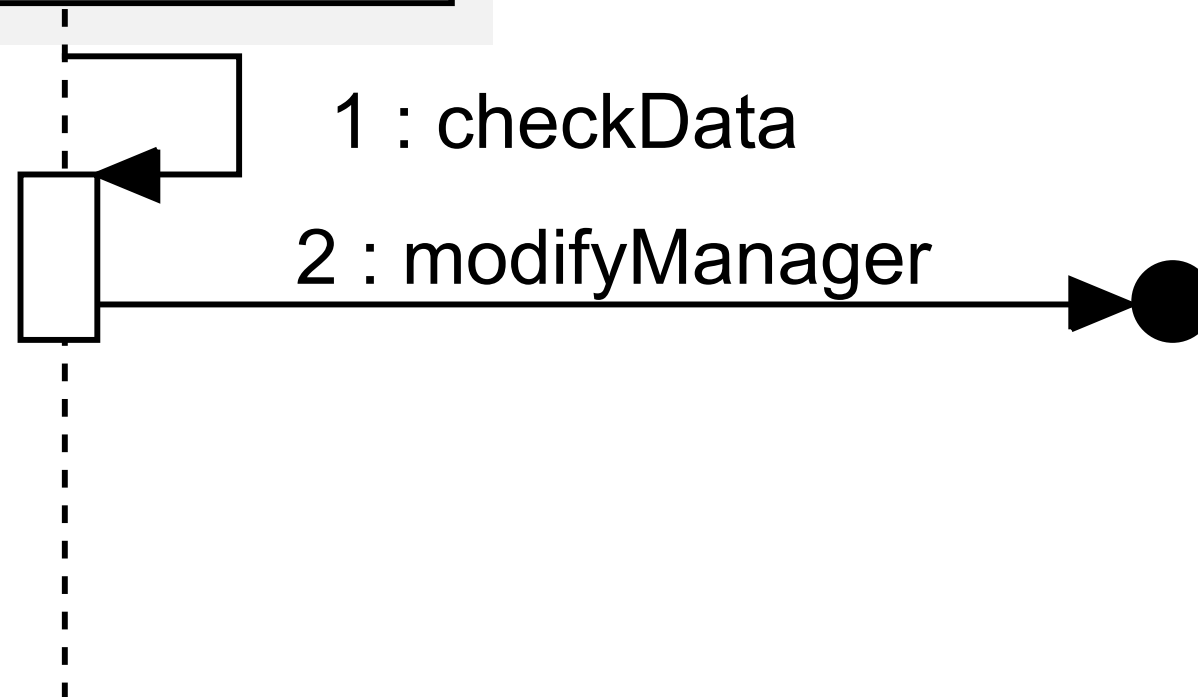
interactionLocal Profile Modification

Modify_frontend



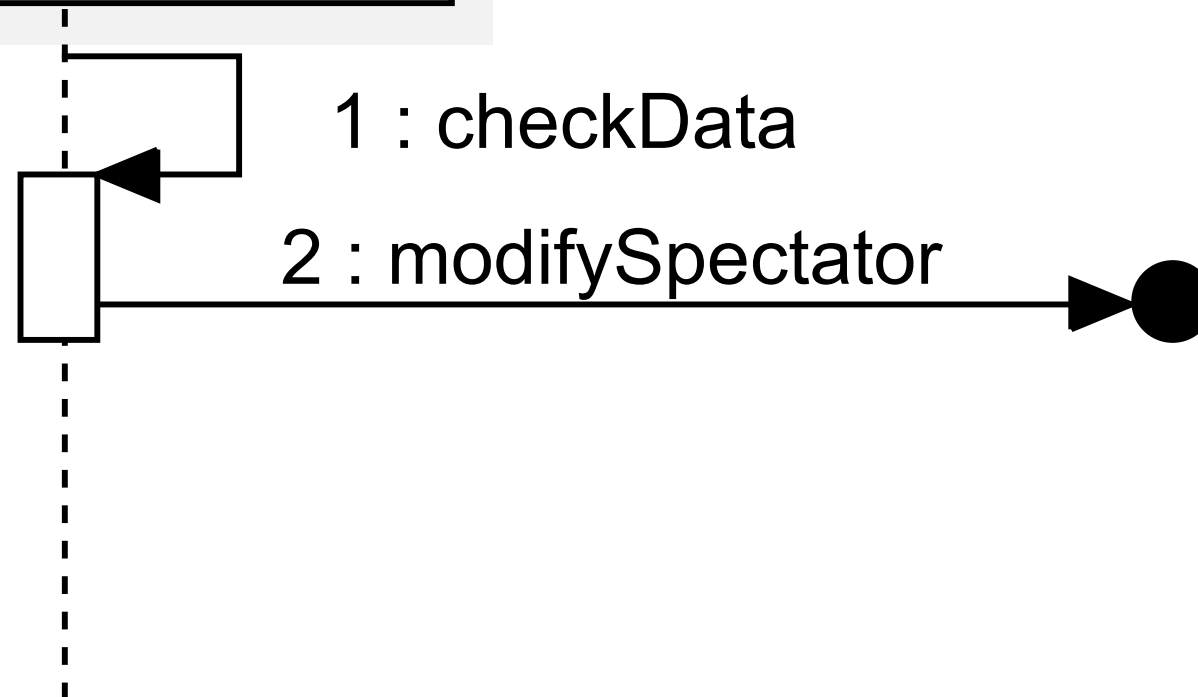
interactionManager Profile Modification

Modify_frontend



interactionSpectator Profile Modification

Modify_frontend



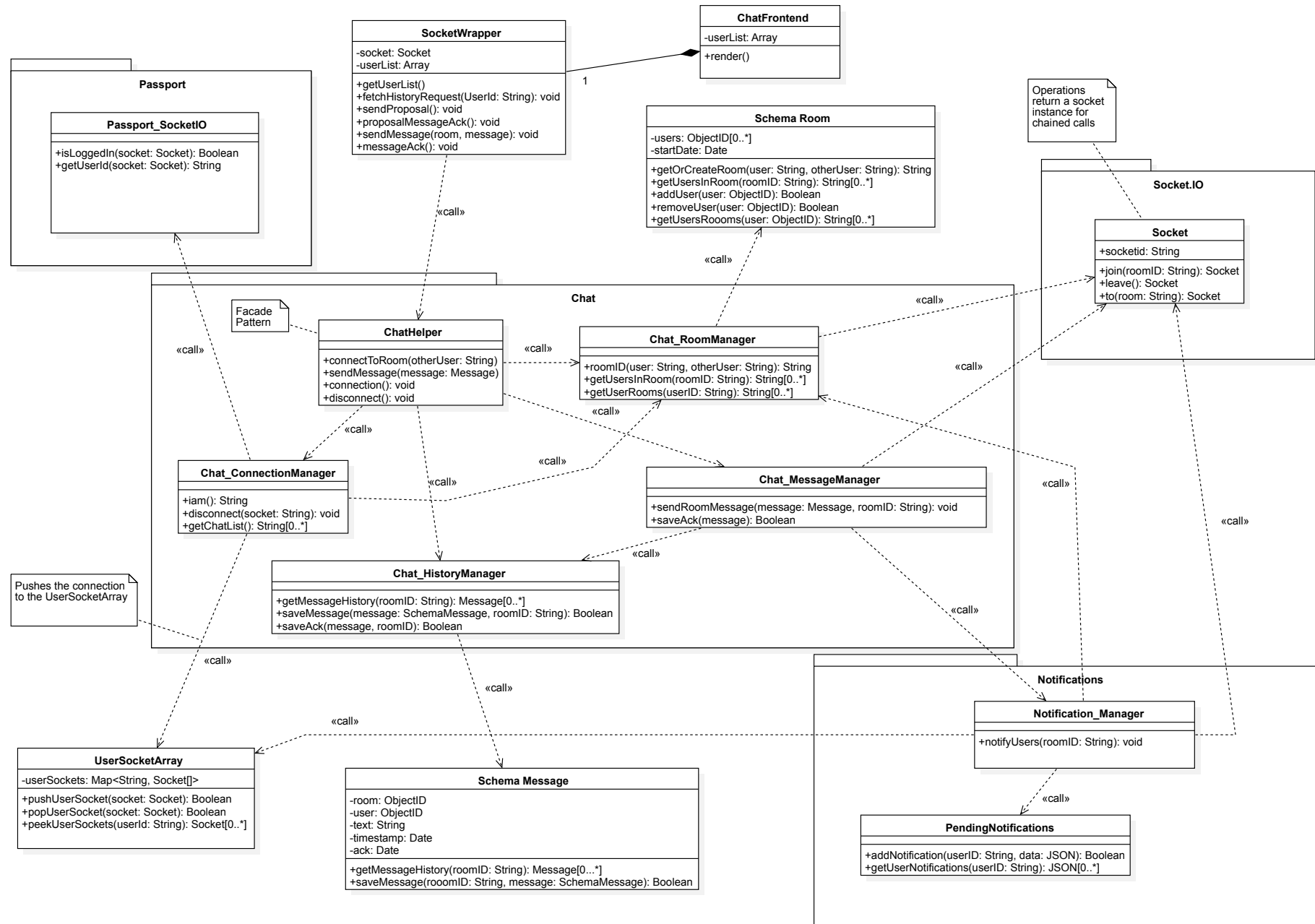
7 Chat

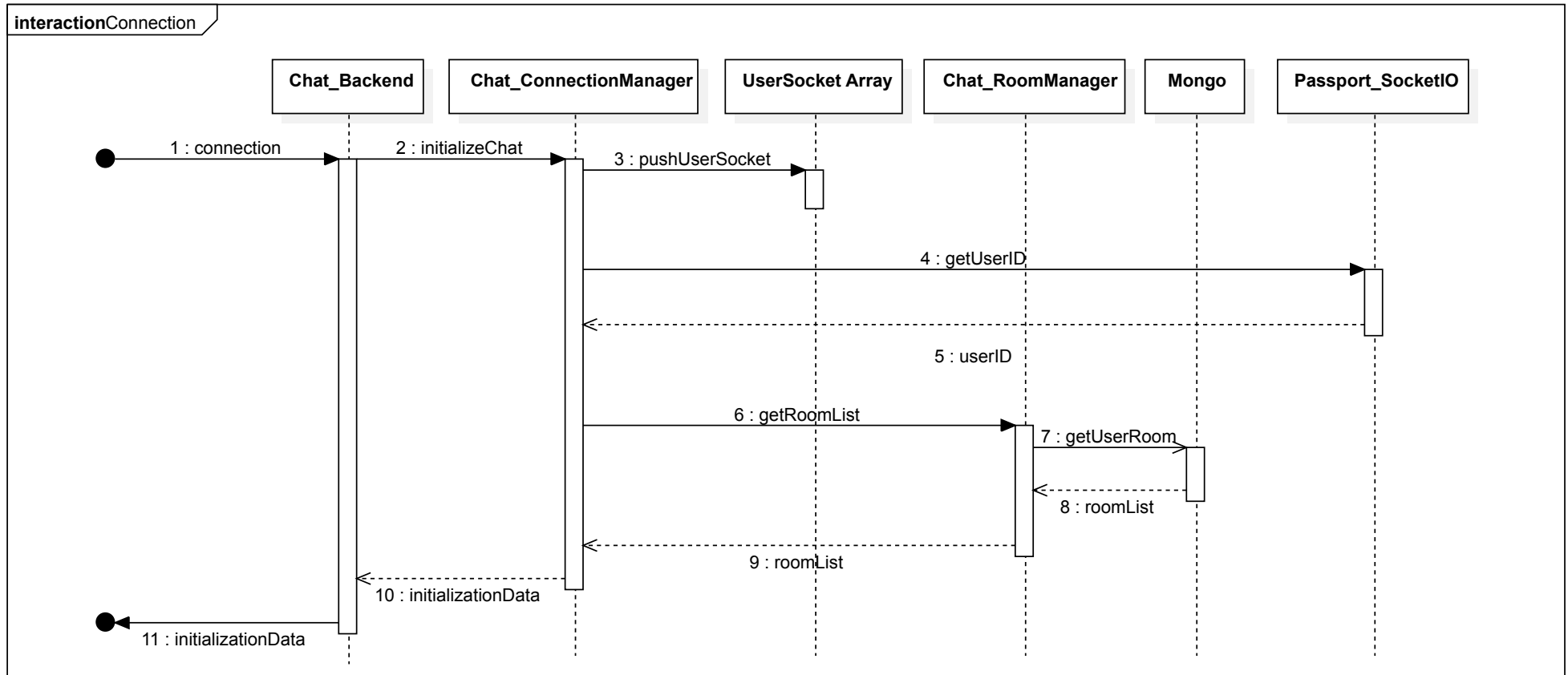
I diagrammi e la progettazione di questo modulo portano alla presa in carico dei requisiti:

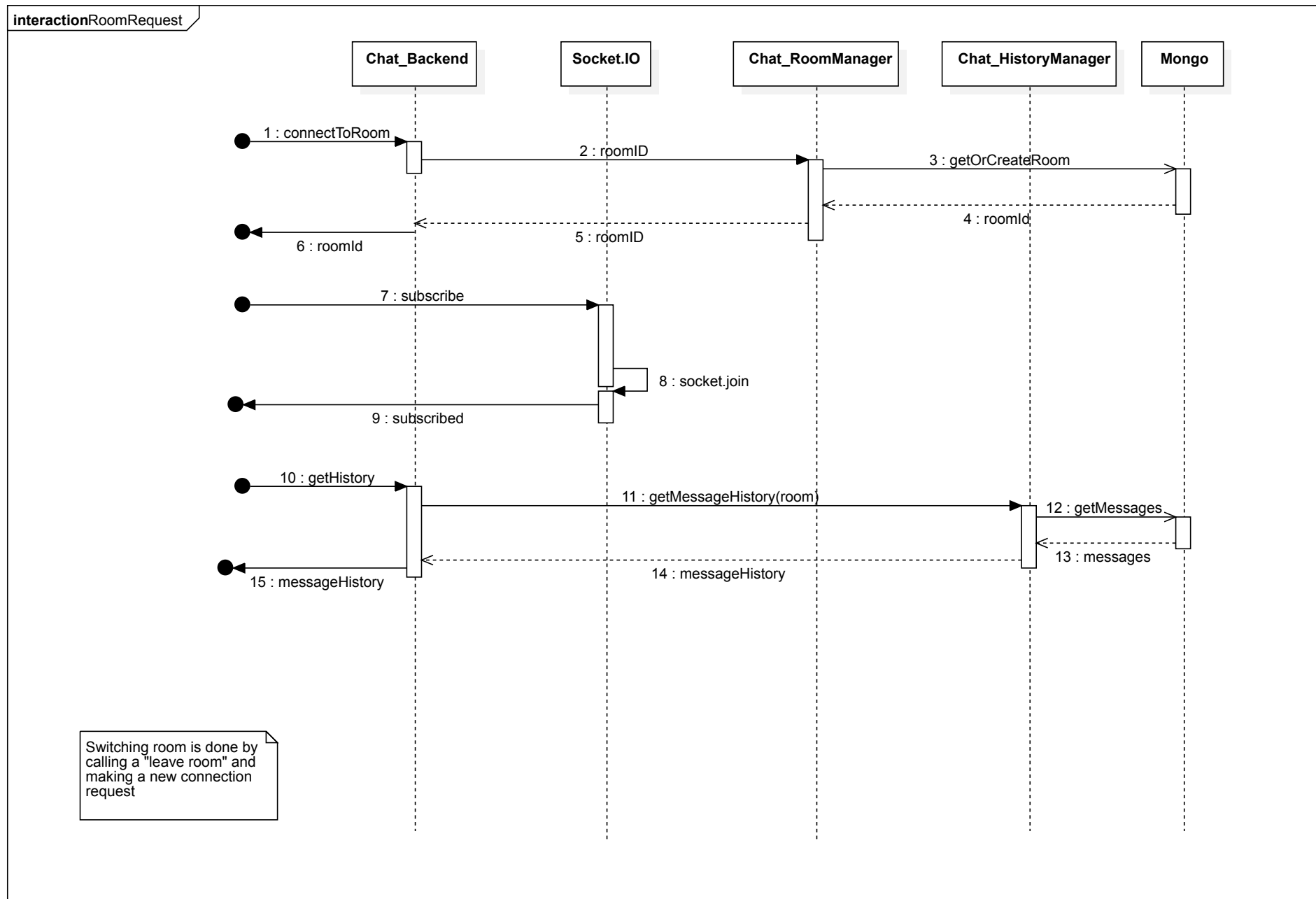
- RF0-6.1.2.1
- RF0-6.2.2.1
- RF0-29
- RF0-29.1

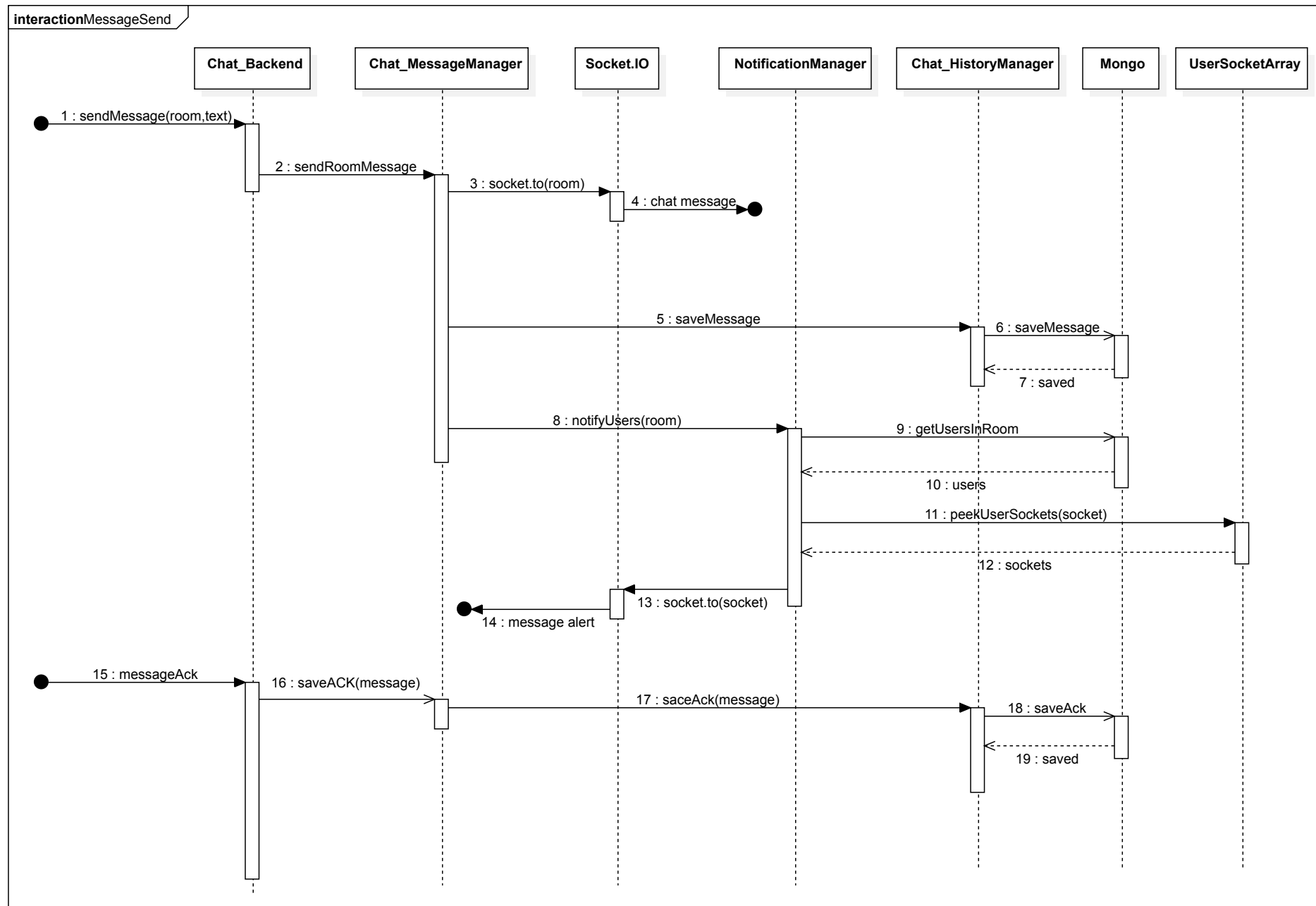
7.1 Chat: back-end

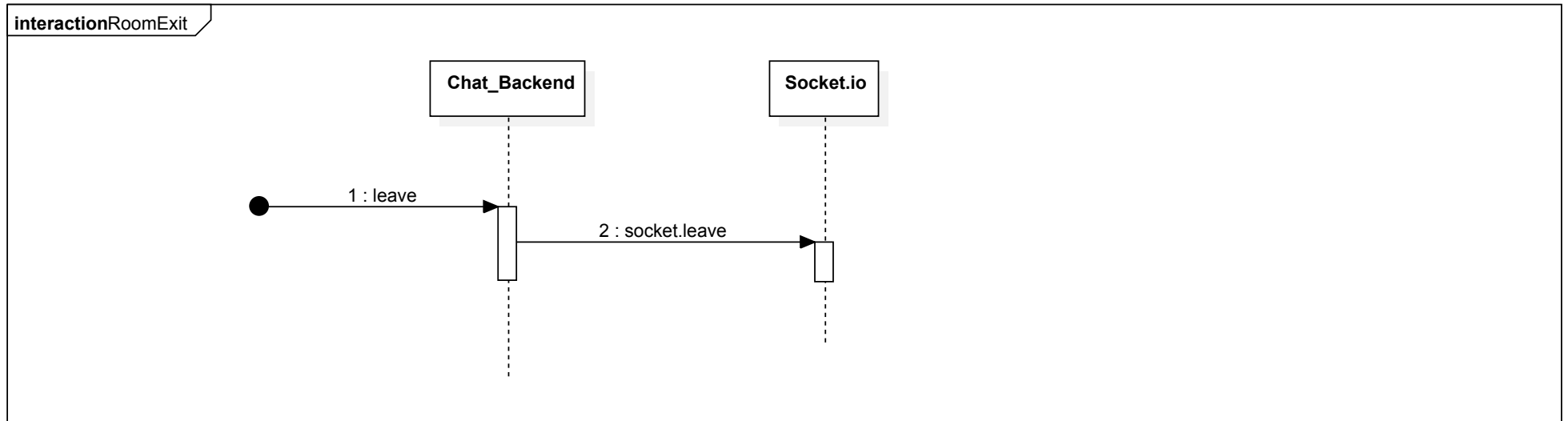
Per la realizzazione della chat, abbiamo deciso di fare uso di Socket.io, che ci permette di avere un framework semplice per la comunicazione in tempo reale. Allo scopo di fornire un'interfaccia semplice al front-end, il back-end fa uso del Facade Pattern. Inoltre si fa uso di MongoDB per la memorizzazione delle stanze e dei messaggi.

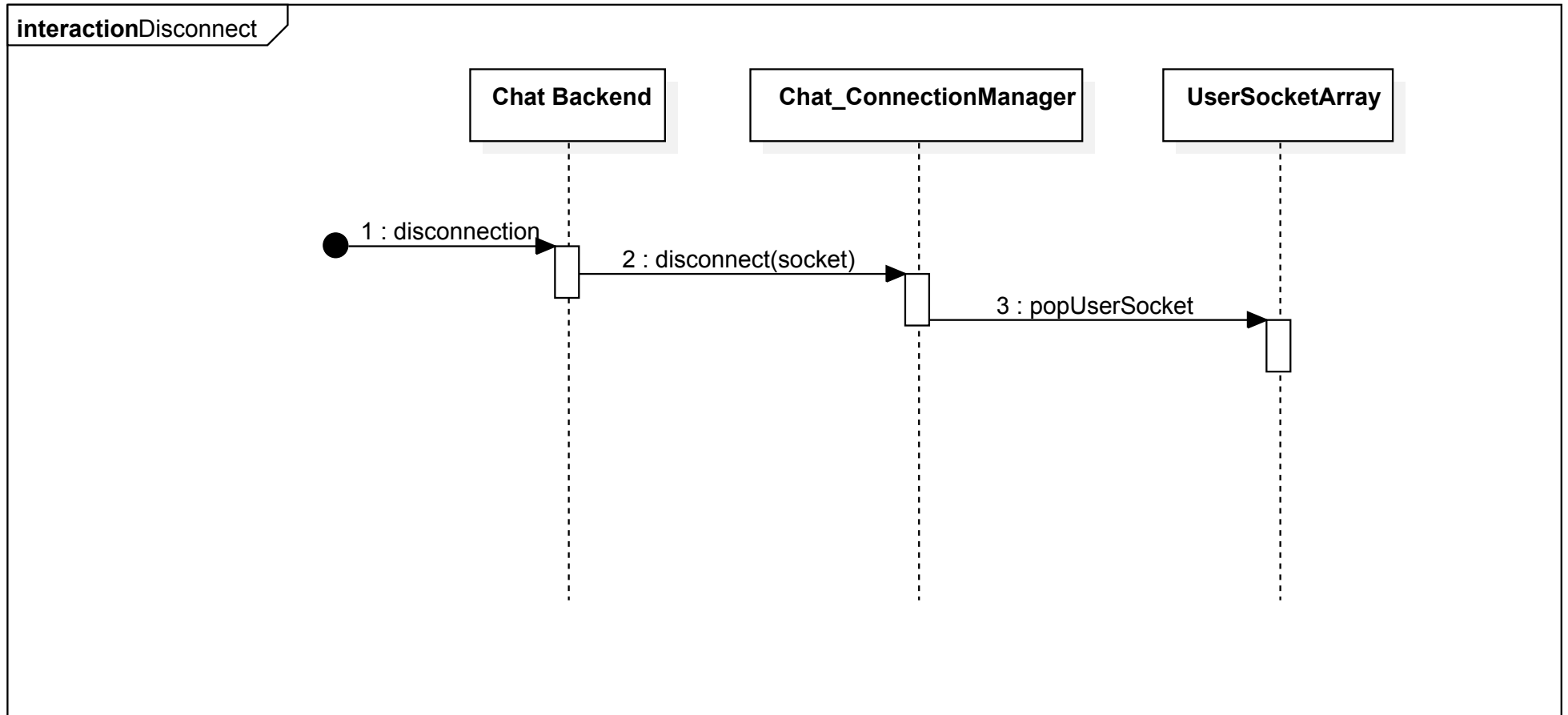






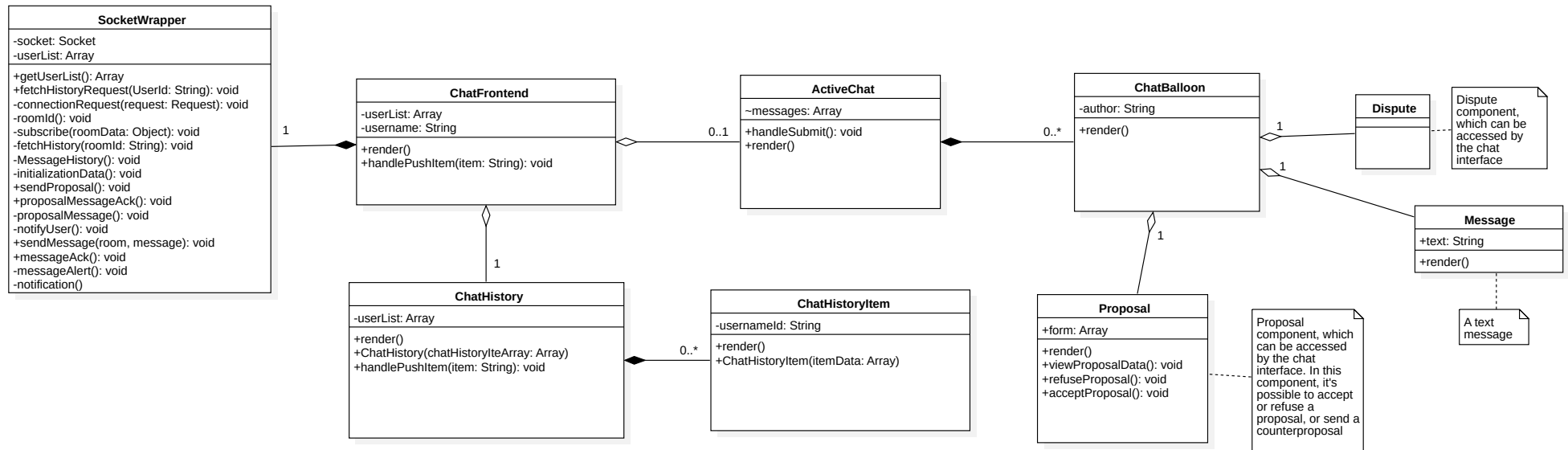


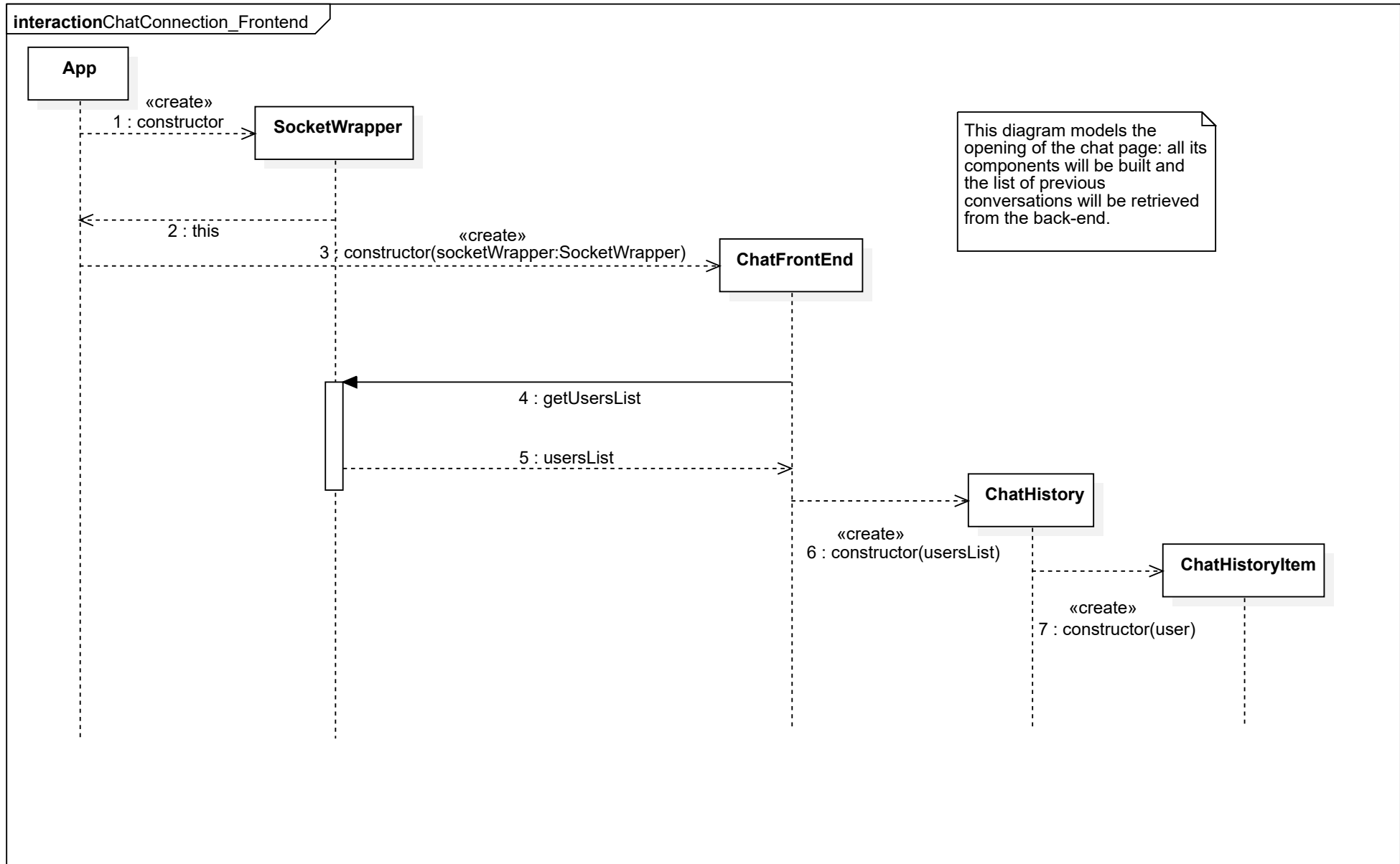


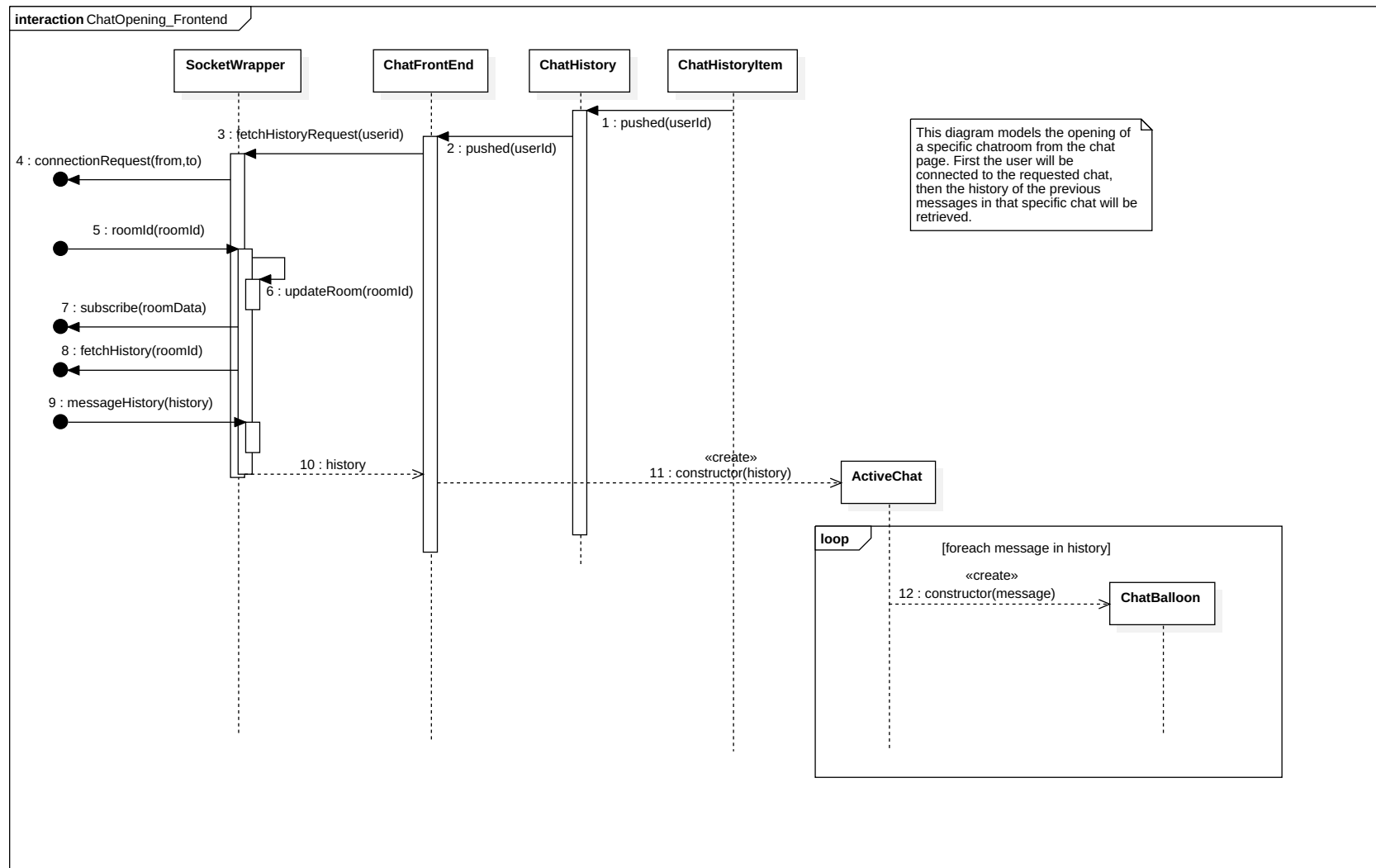


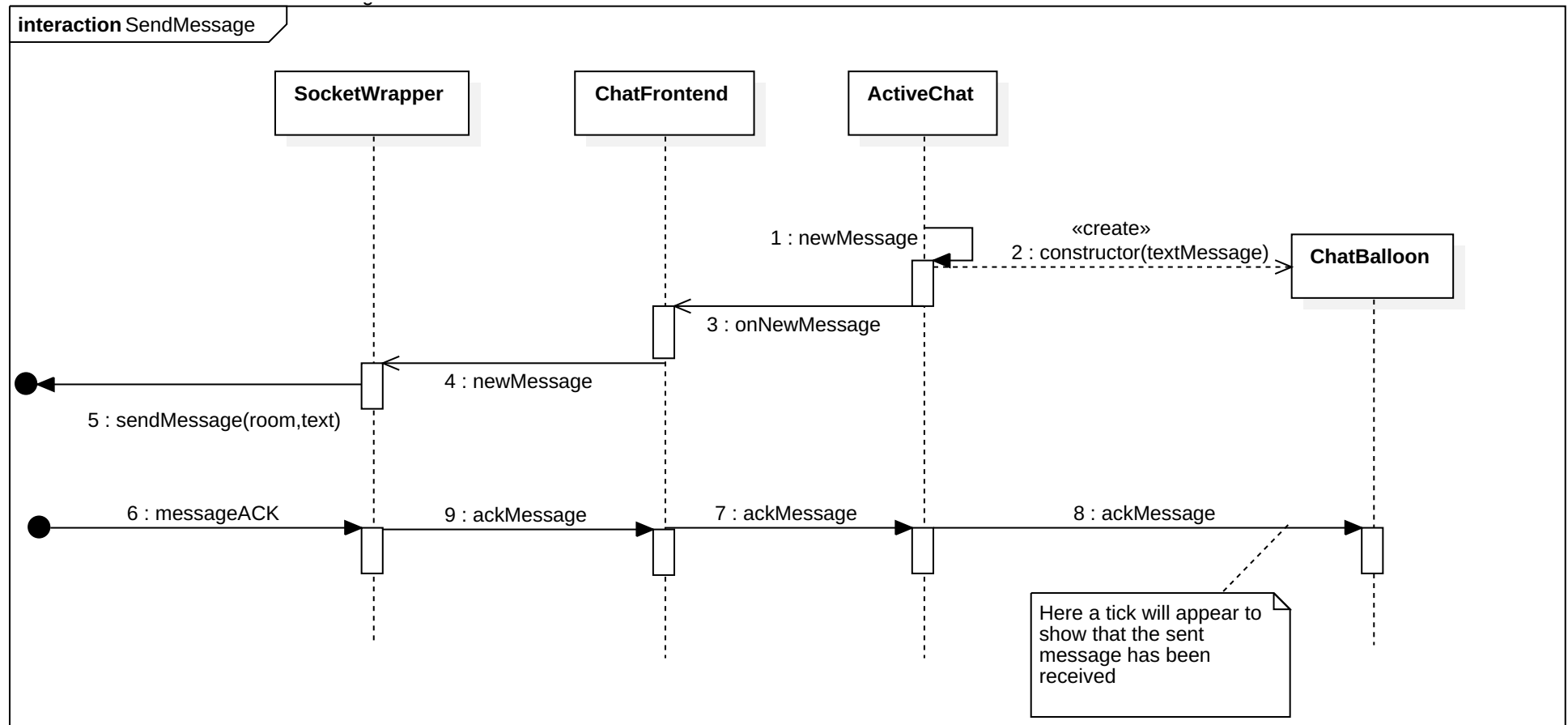
7.2 Chat: front-end

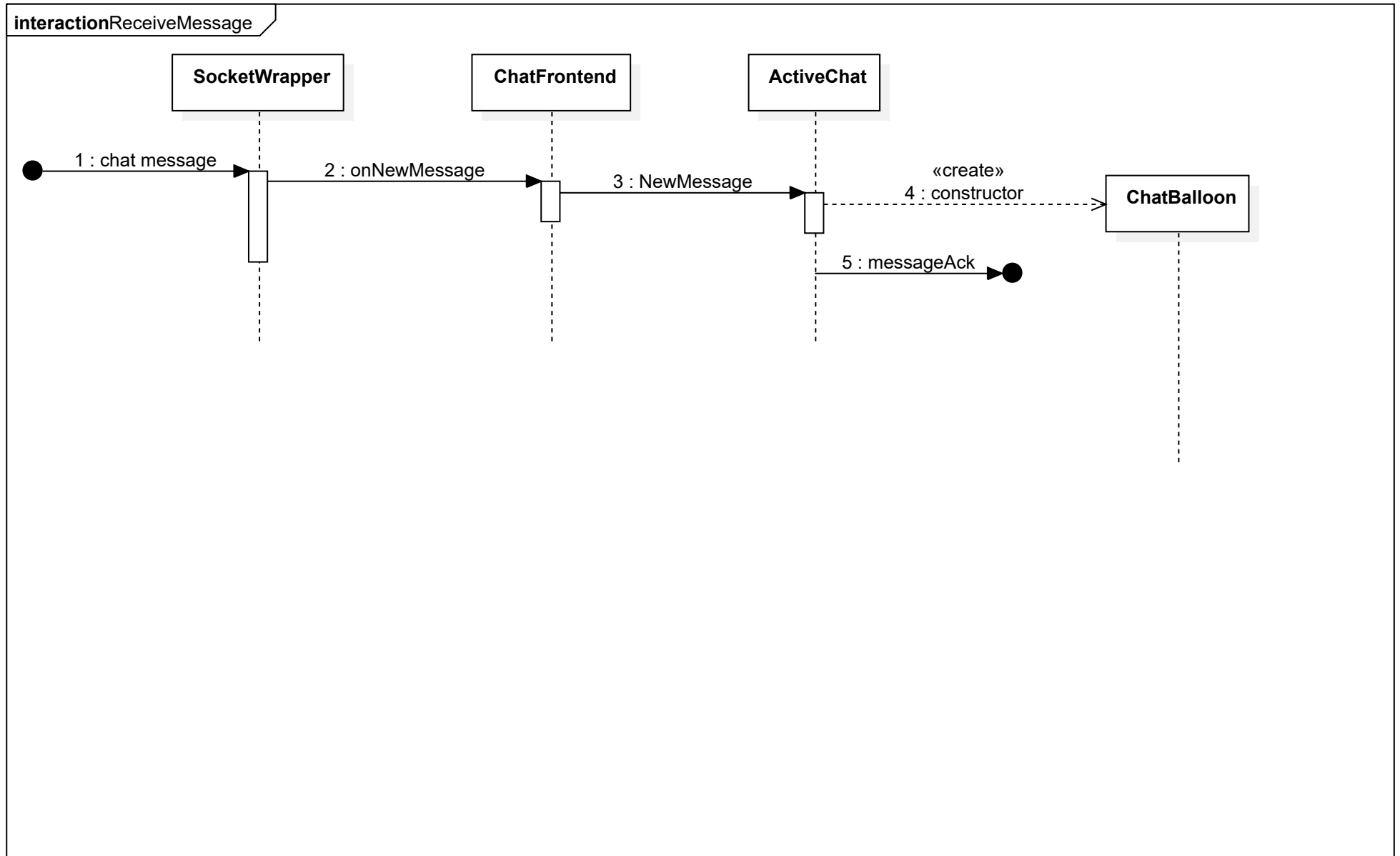
Per la realizzazione della parte frontend della chat, abbiamo deciso di usare una classe JavaScript *SocketWrapper* che incapsulerà tutte le interazioni con i socket, gestendole, per poi interagire con le varie componenti React.











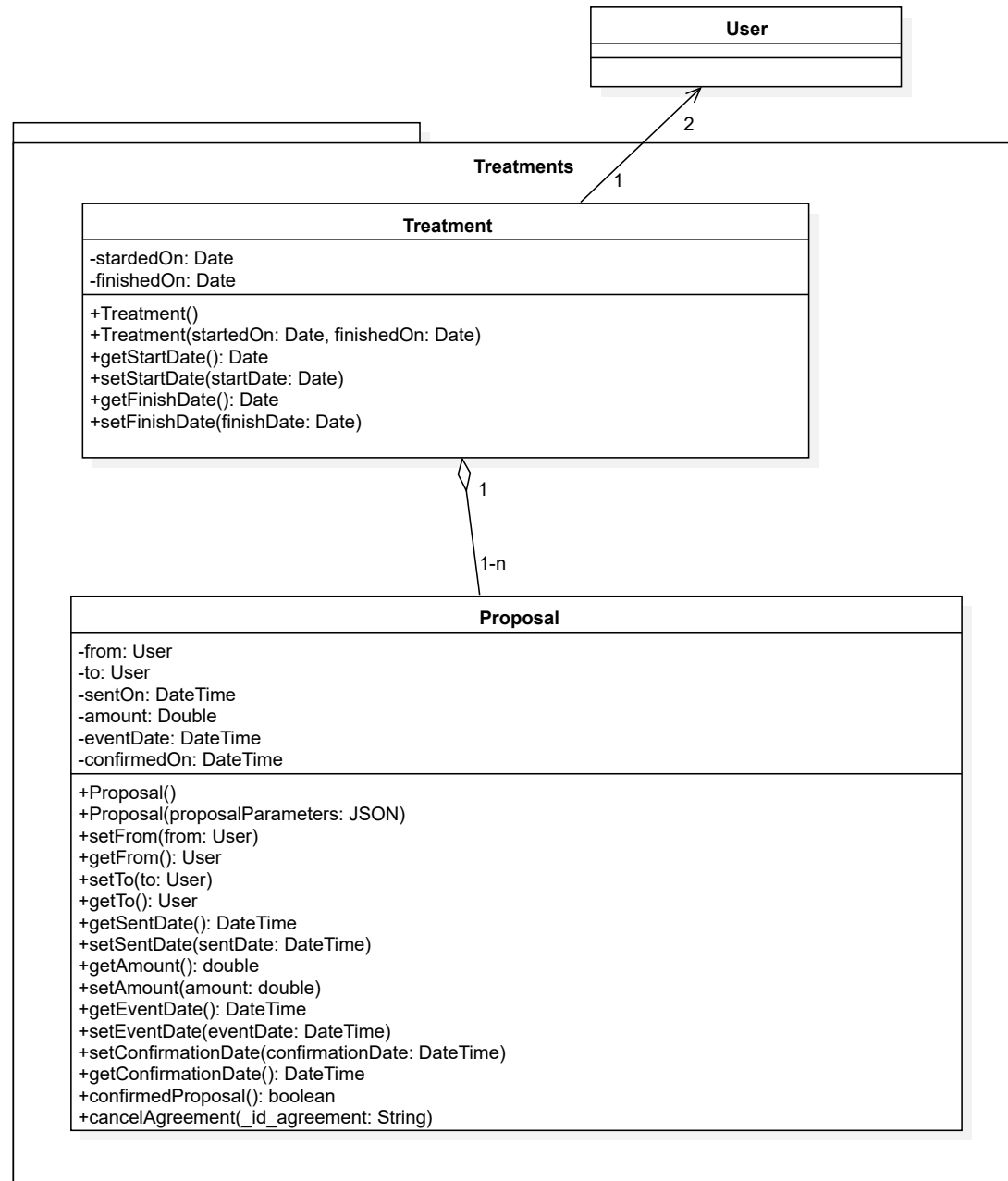
8 Proposta

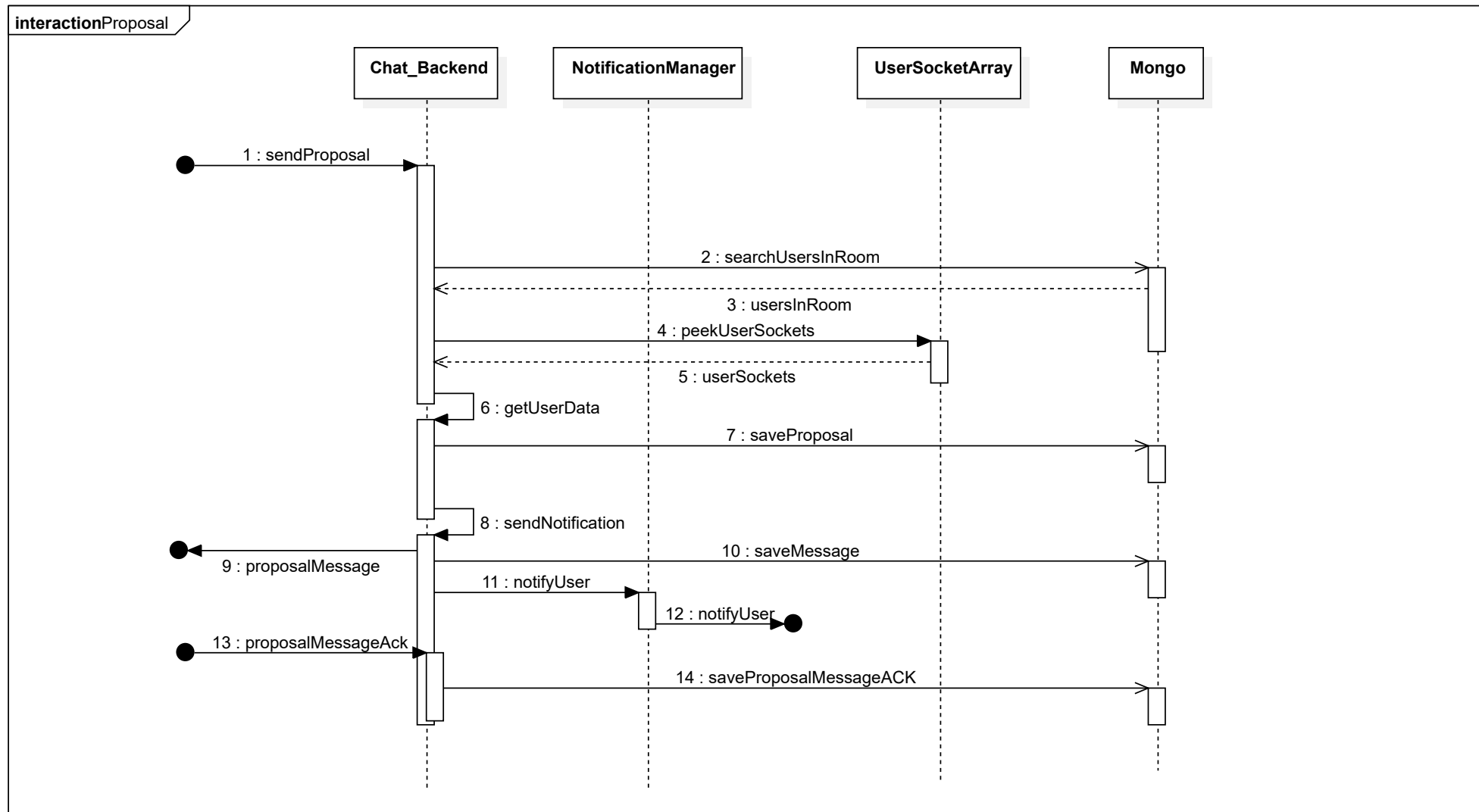
Con lo sviluppo e la progettazione di questo modulo, il gruppo si prende in carico i seguenti requisiti:

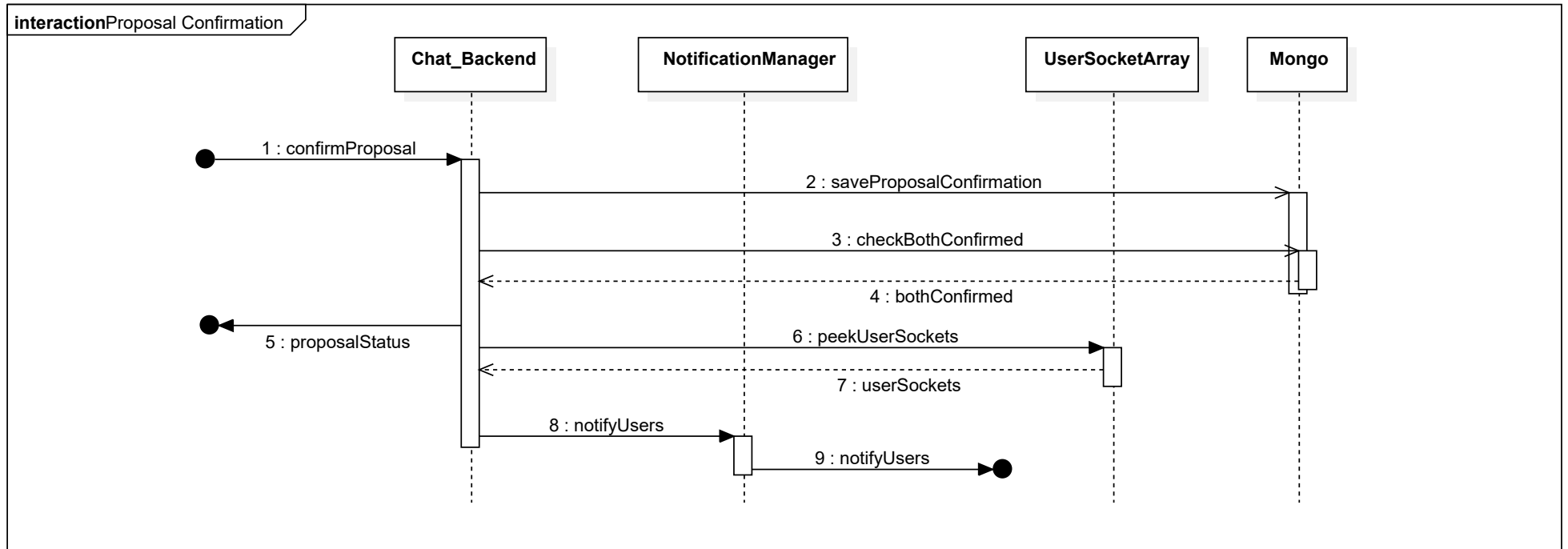
- RF0-6
- RF0-6.1
- RF0-6.1.1
- RF0-6.1.2.1
- RF0-6.1.3
- RF0-6.2
- RF0-6.2.1
- RF0-6.2.2.1
- RF0-6.5
- RF0-6.6
- RF0-14.1
- RF0-14.2
- RF0-18
- RF0-19
- RF0-20
- RF0-22
- RF0-23
- RF0-24

8.1 Proposta: back-end

A seguito di un invio di una proposta tramite chat, il back-end si occuperà di notificare la controparte della presenza di una nuova proposta. MongoDB si occuperà del salvataggio della proposta in database.

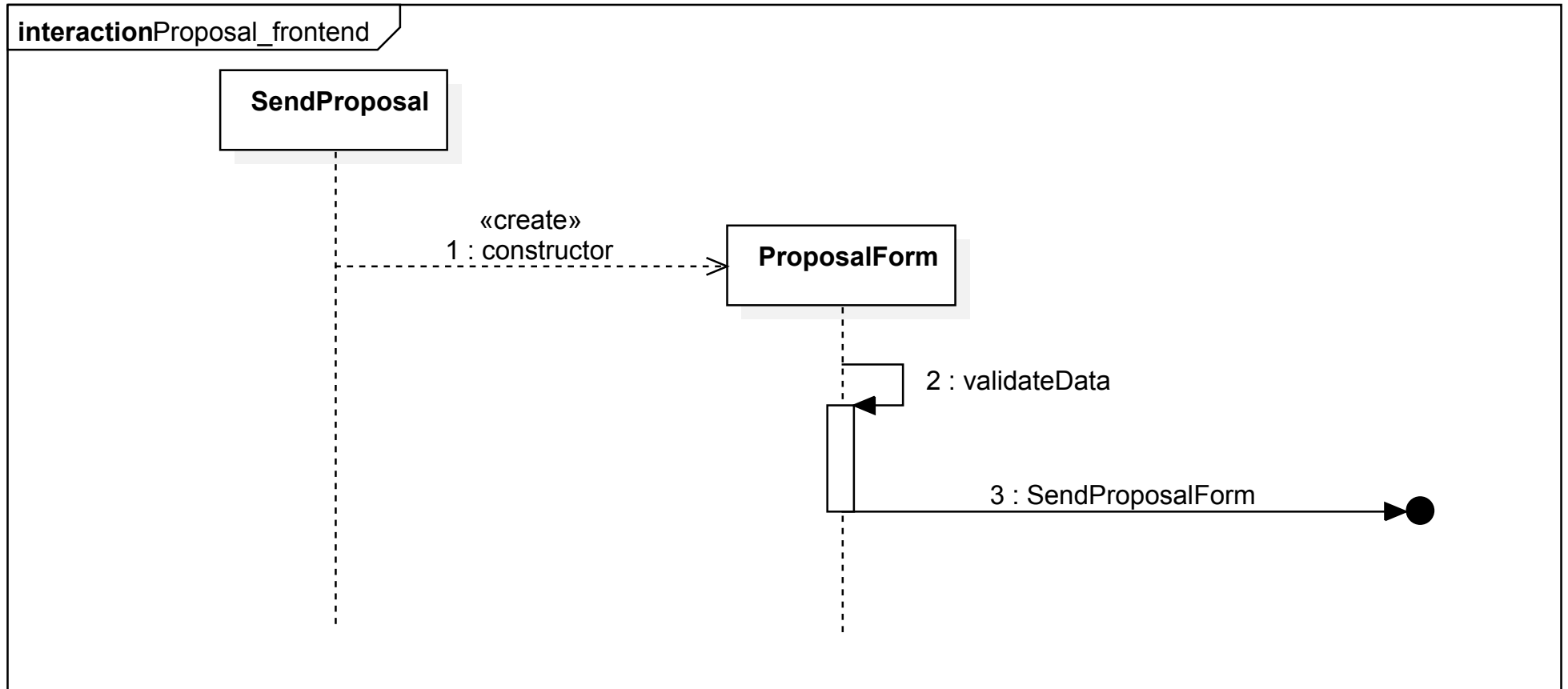


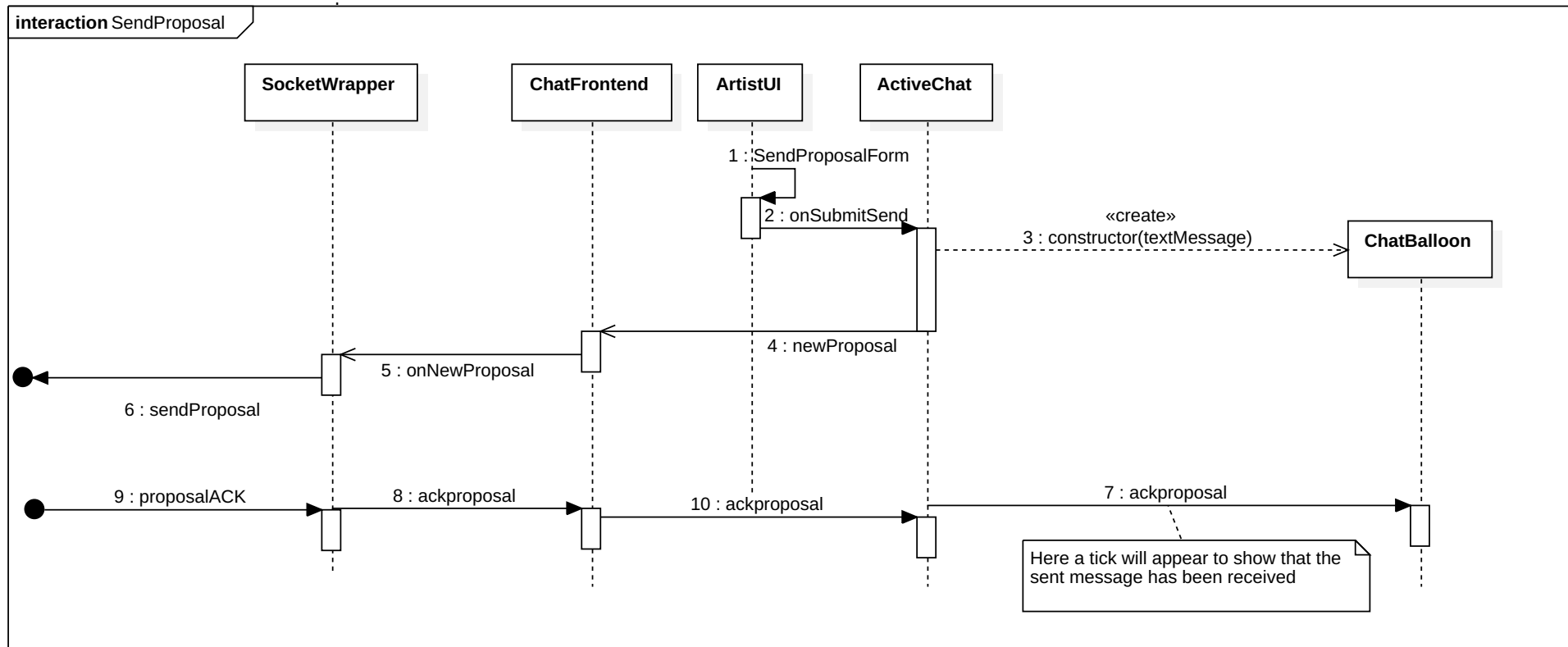


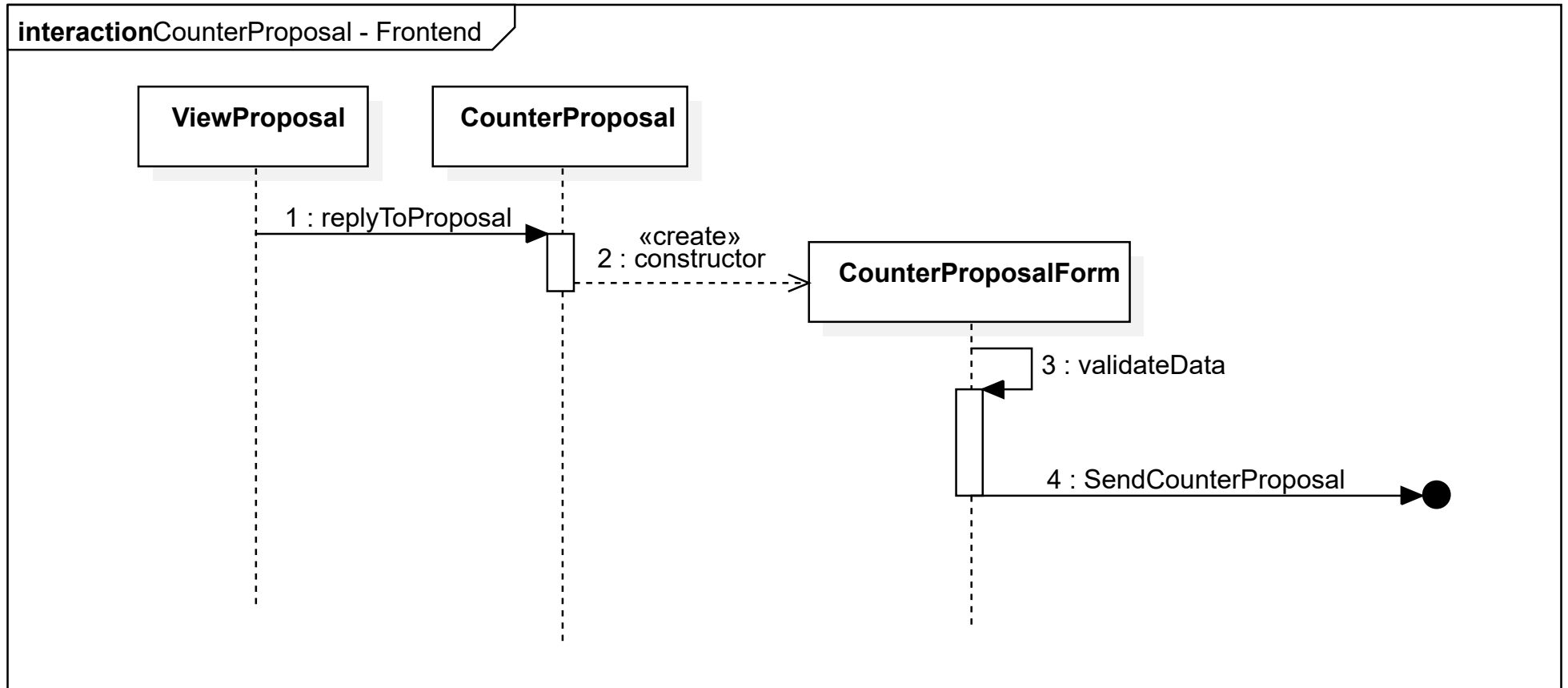


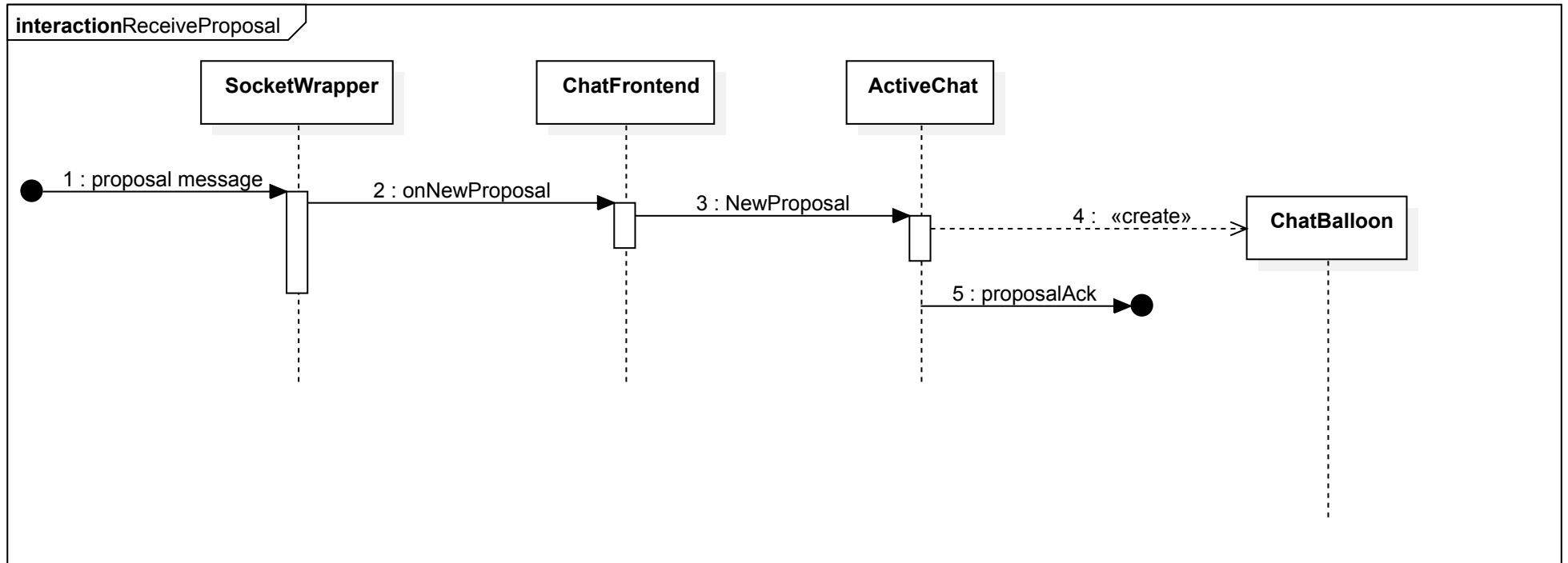
8.2 Proposta: front-end

Per la realizzazione della parte frontend della proposta, nel diagramma delle classi, abbiamo deciso di usare una classe React *Proposal* che è visibile nella chat 7.2 e che ha le funzionalità per accettare, rifiutare, rispondere e leggere una proposta mentre per inviare una proposta è possibile farlo nella User Interface di un altro utente. Per i diagrammi di sequenza vi è il diagramma per fare una controproposta, quello in cui viene modellata la ricezione e l'invio di una proposta.









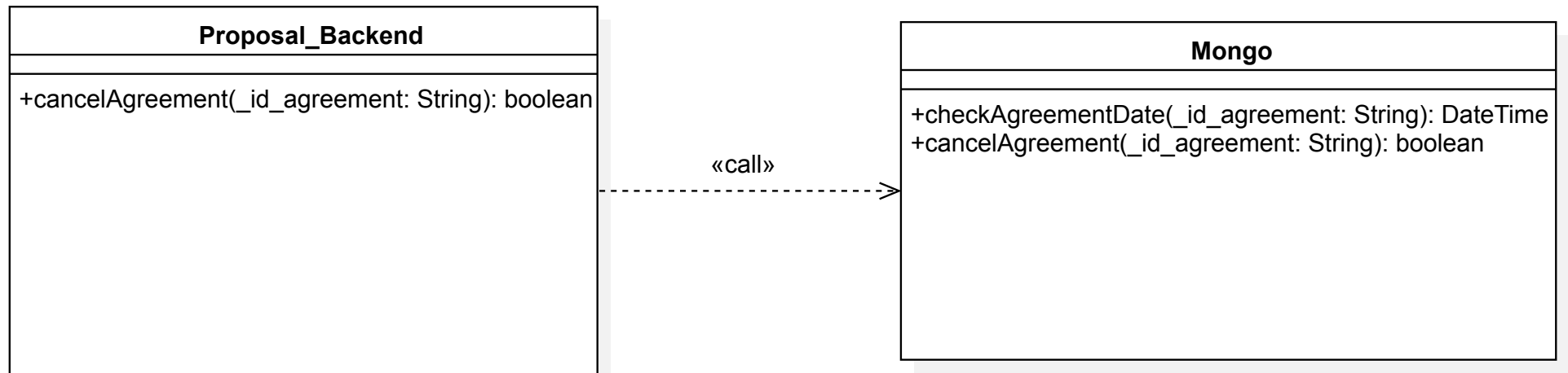
9 Eliminazione di un accordo

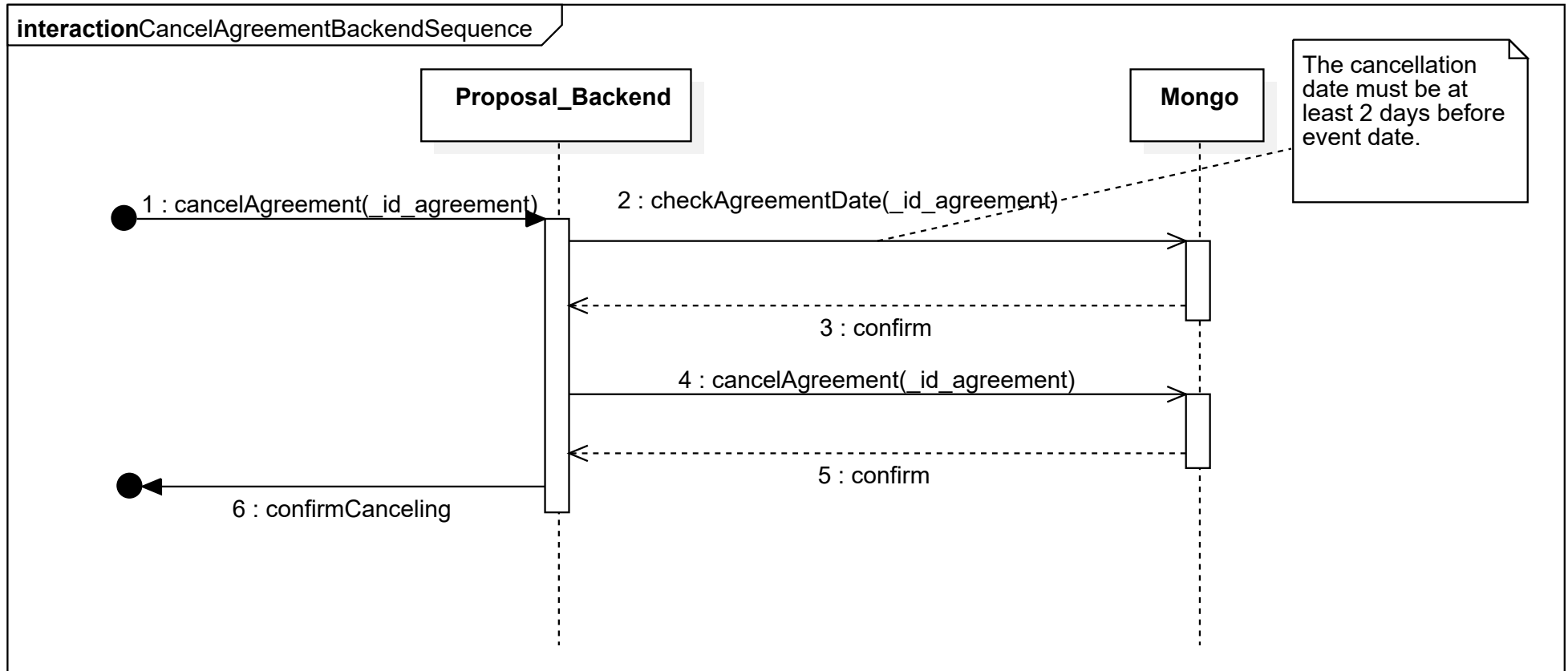
Con lo sviluppo e la progettazione di questo modulo, il gruppo prende in carico i seguenti requisiti:

- RF0-6.3
- RF0-6.3.1
- RF0-6.3.1

9.1 Eliminazione di un accordo: back-end

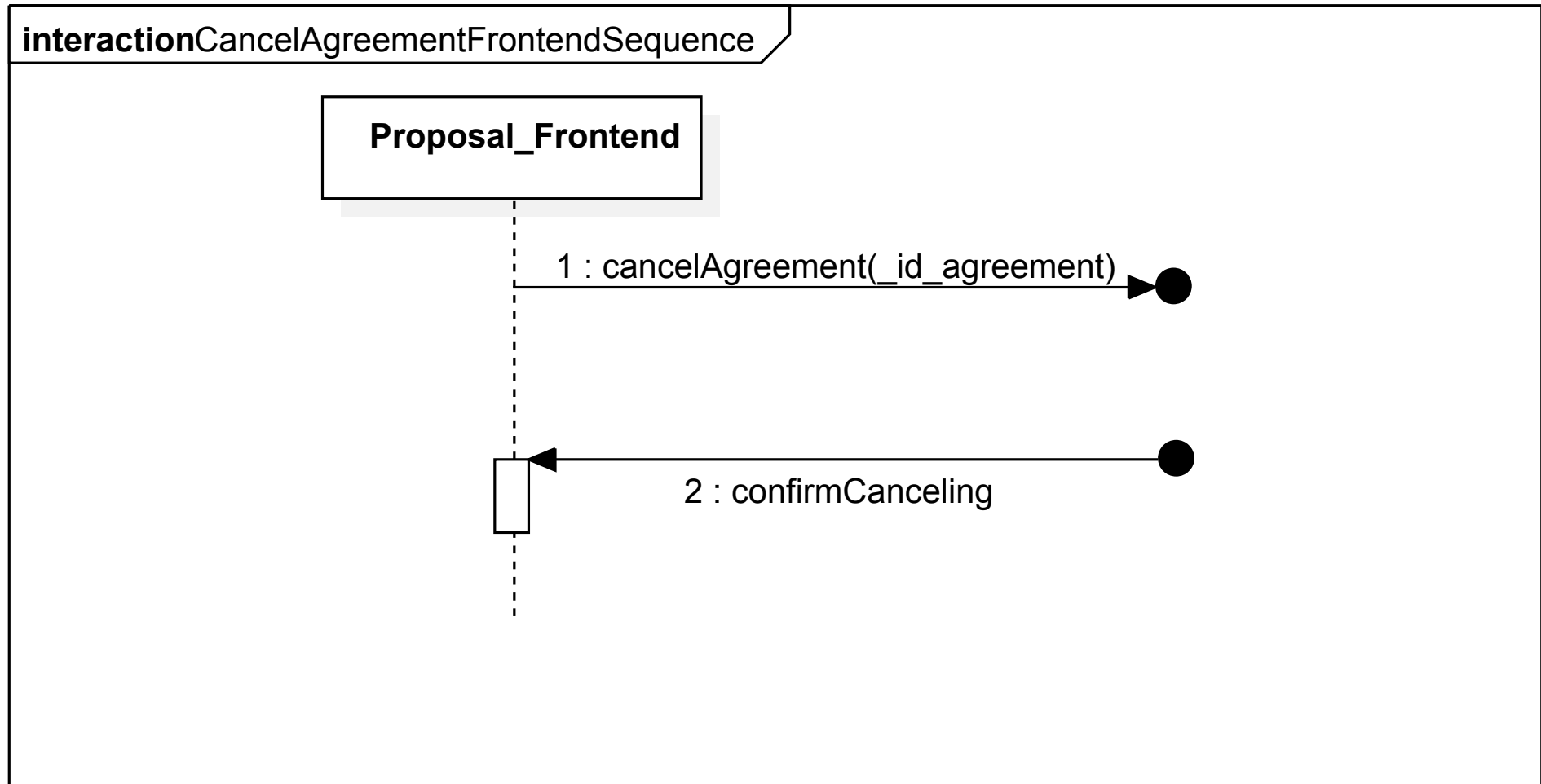
A seguito di una richiesta di eliminazione accordo, il back-end andrà a rimuovere l'accordo precedentemente confermato.





9.2 Eliminazione di un accordo: front-end

A seguito di una richiesta di eliminazione accordo, il front-end inoltra la richiesta al back-end.



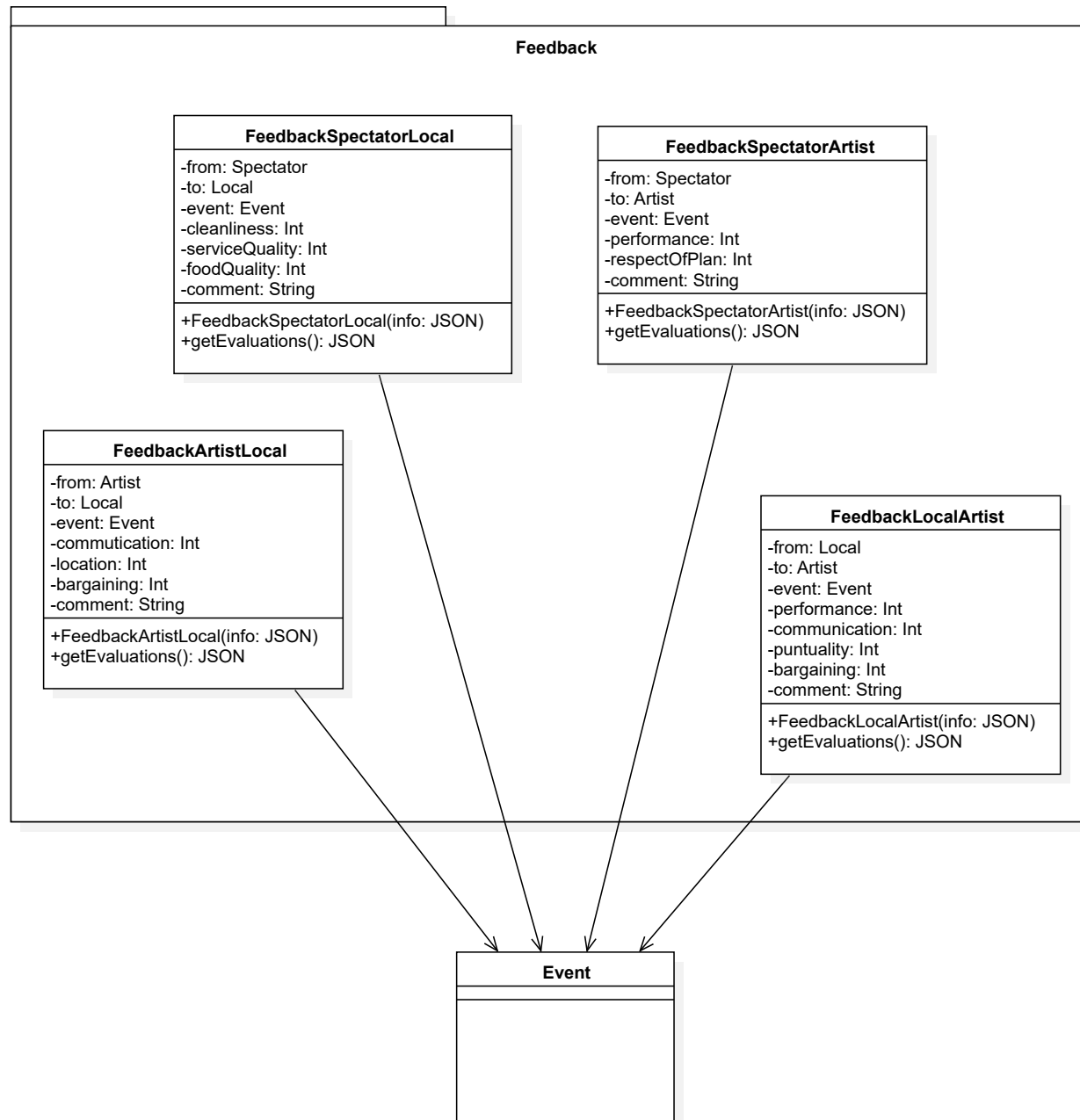
10 Rilascio dei feedback

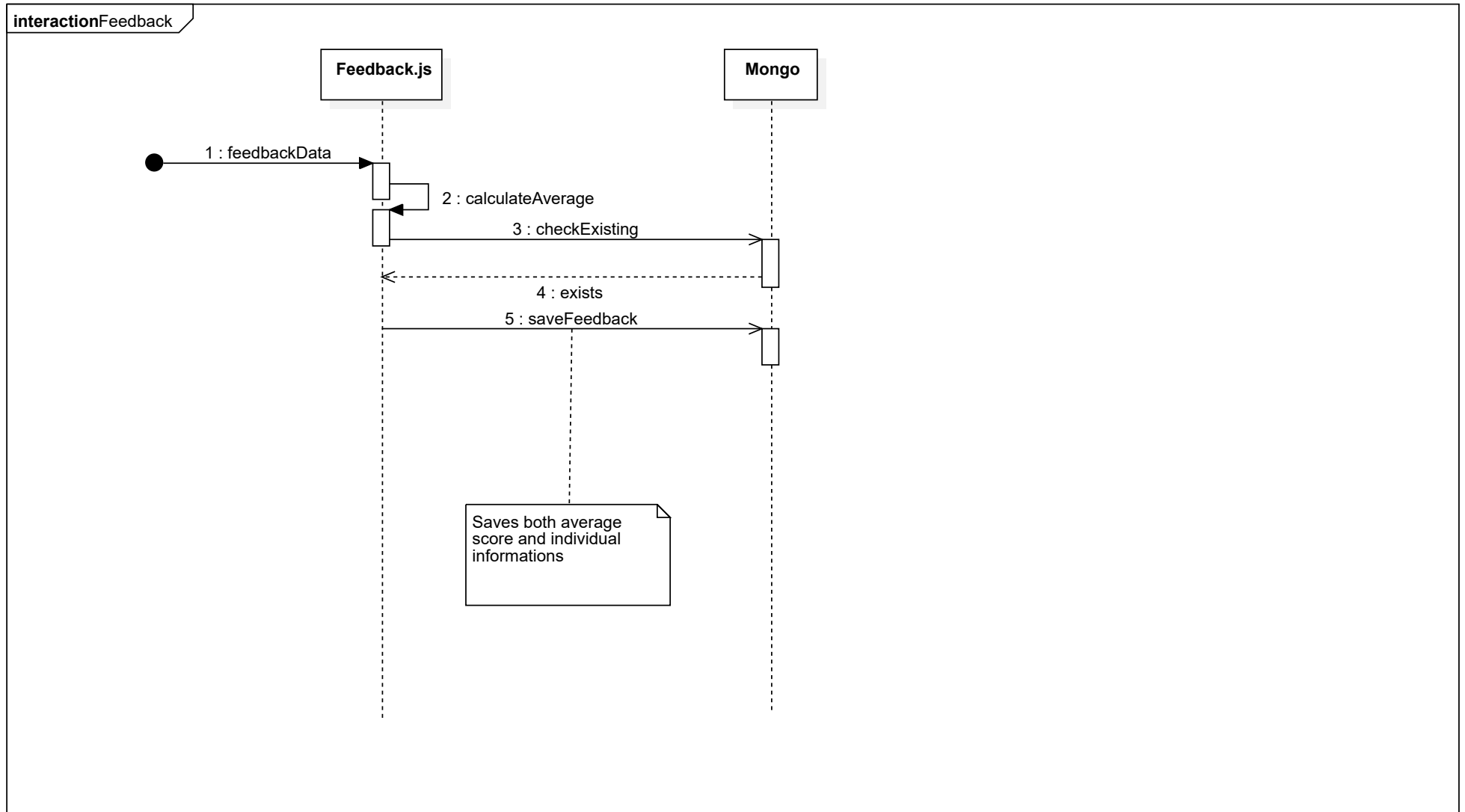
Con lo sviluppo e la progettazione di questo modulo, il gruppo prende in carico i seguenti requisiti:

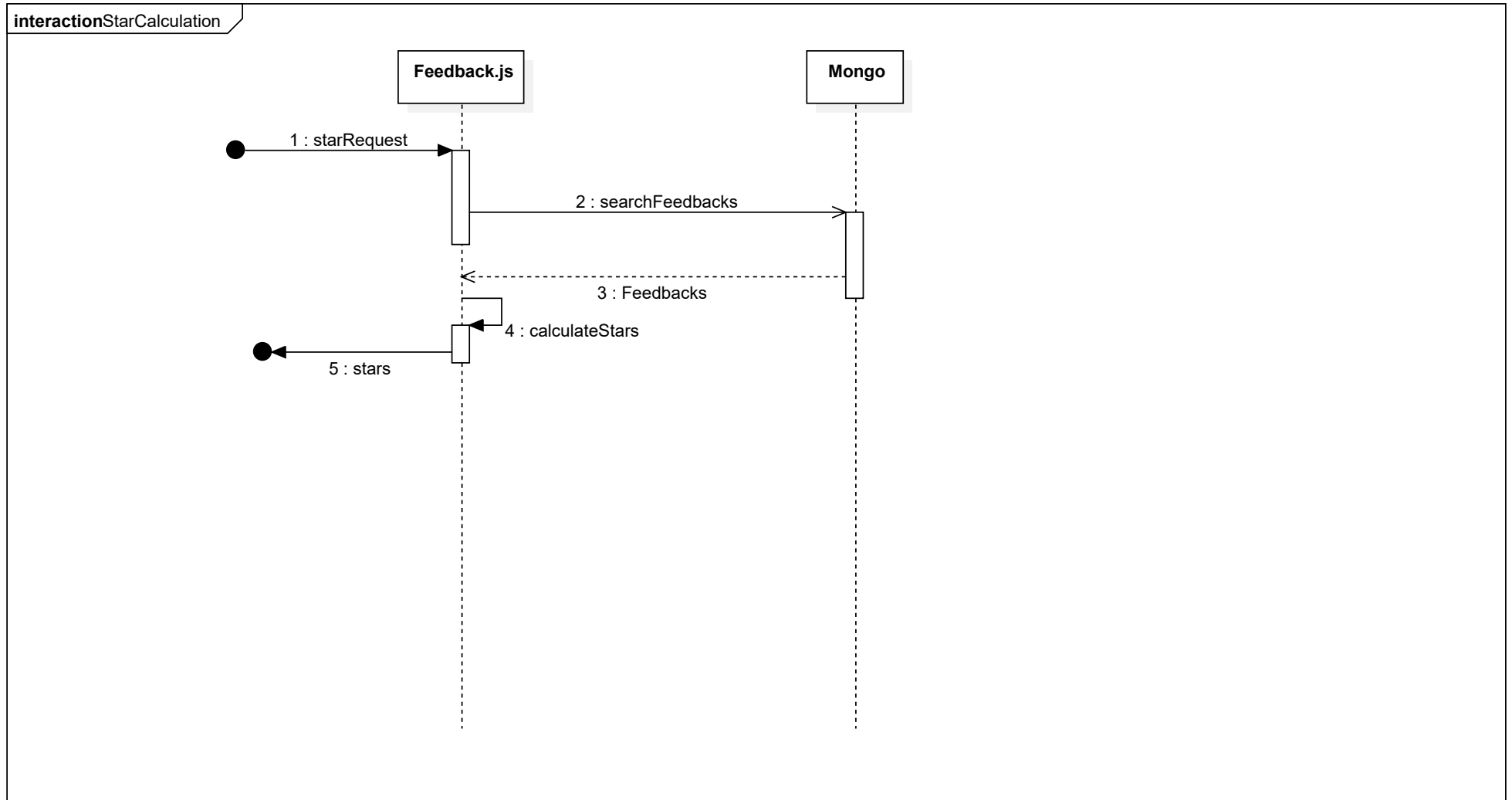
- RF0-8.2
- RF0-8.2.1
- RF0-8.3
- RF0-8.3.1
- RF0-8.4
- RF0-8.4.1
- RF0-8.5
- RF0-8.5.1
- RF0-8.7

10.1 Feedback: back-end

A seguito di una richiesta di rilascio feedback, il back-end elaborerà la richiesta andando a controllare tramite MongoDB se il rilascio feedback sia già avvenuto. In caso di esito negativo, salverà in database tramite MongoDB le informazioni del feedback appena rilasciato.

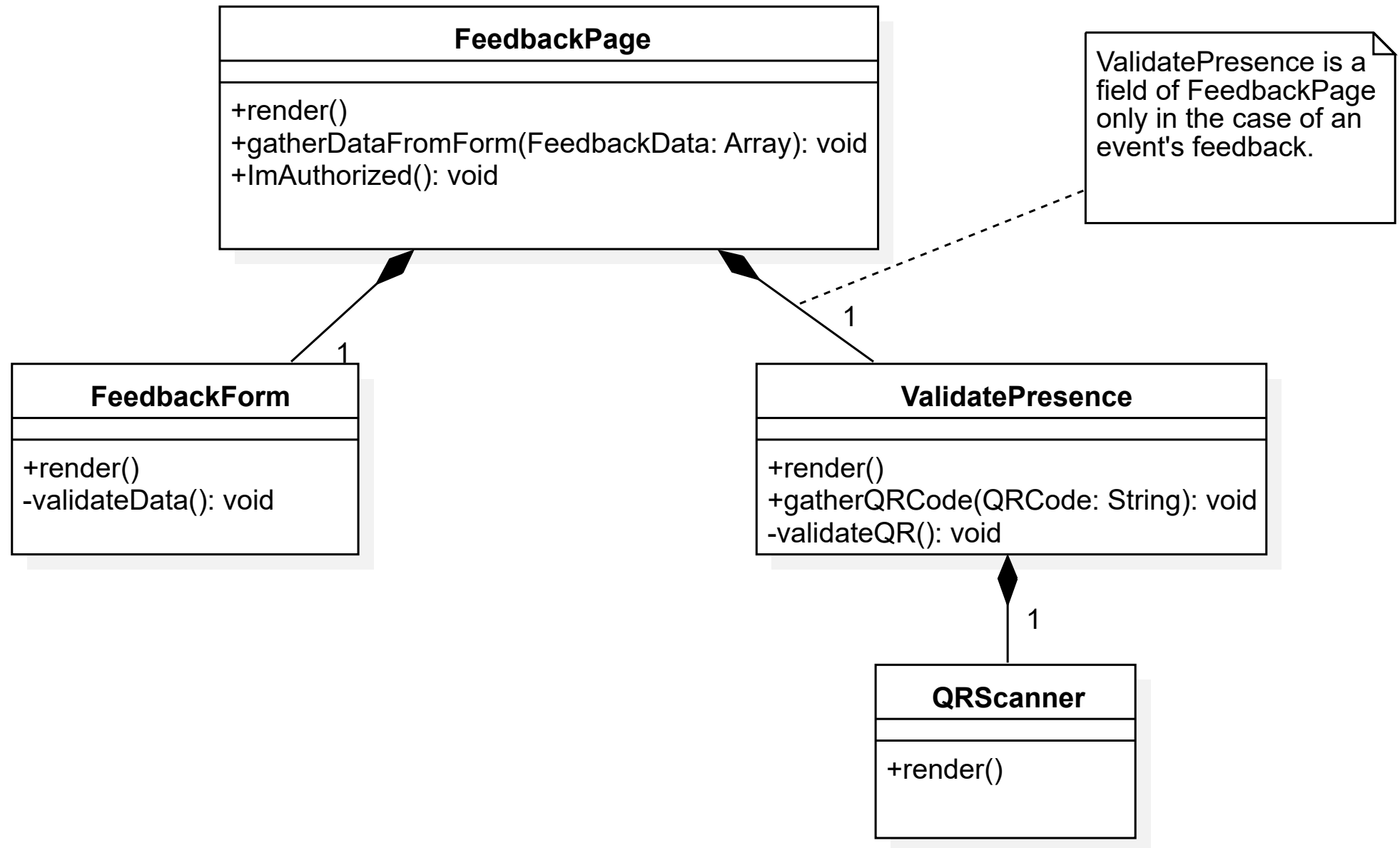


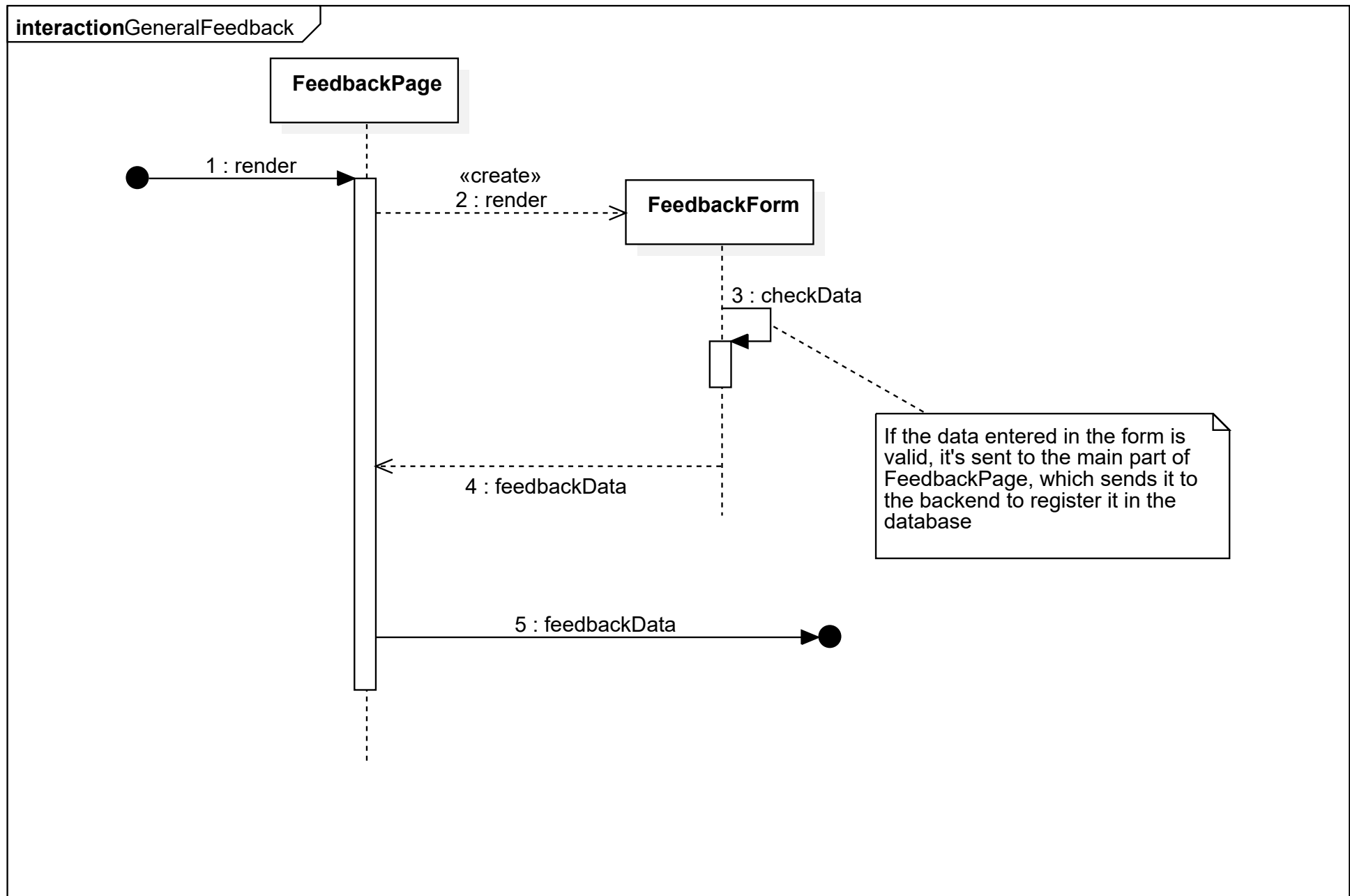


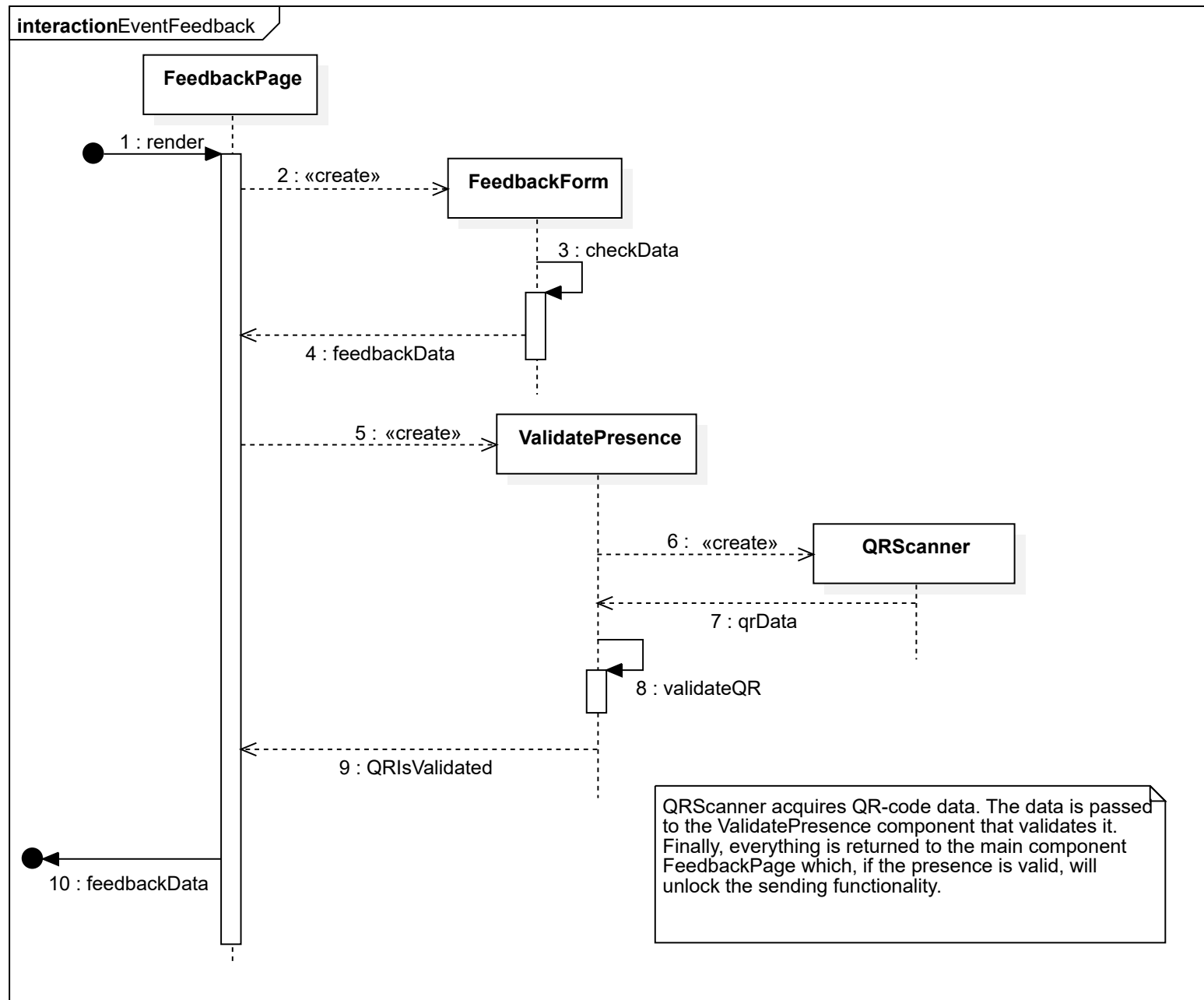


10.2 Feedback: front-end

A seguito di una richiesta di rilascio feedback ad un evento, il front-end controlla la validità dei dati inseriti nella form. Viene verificato inoltre che l'utente abbia partecipato all'evento tramite il controllo di validità del codice QR.







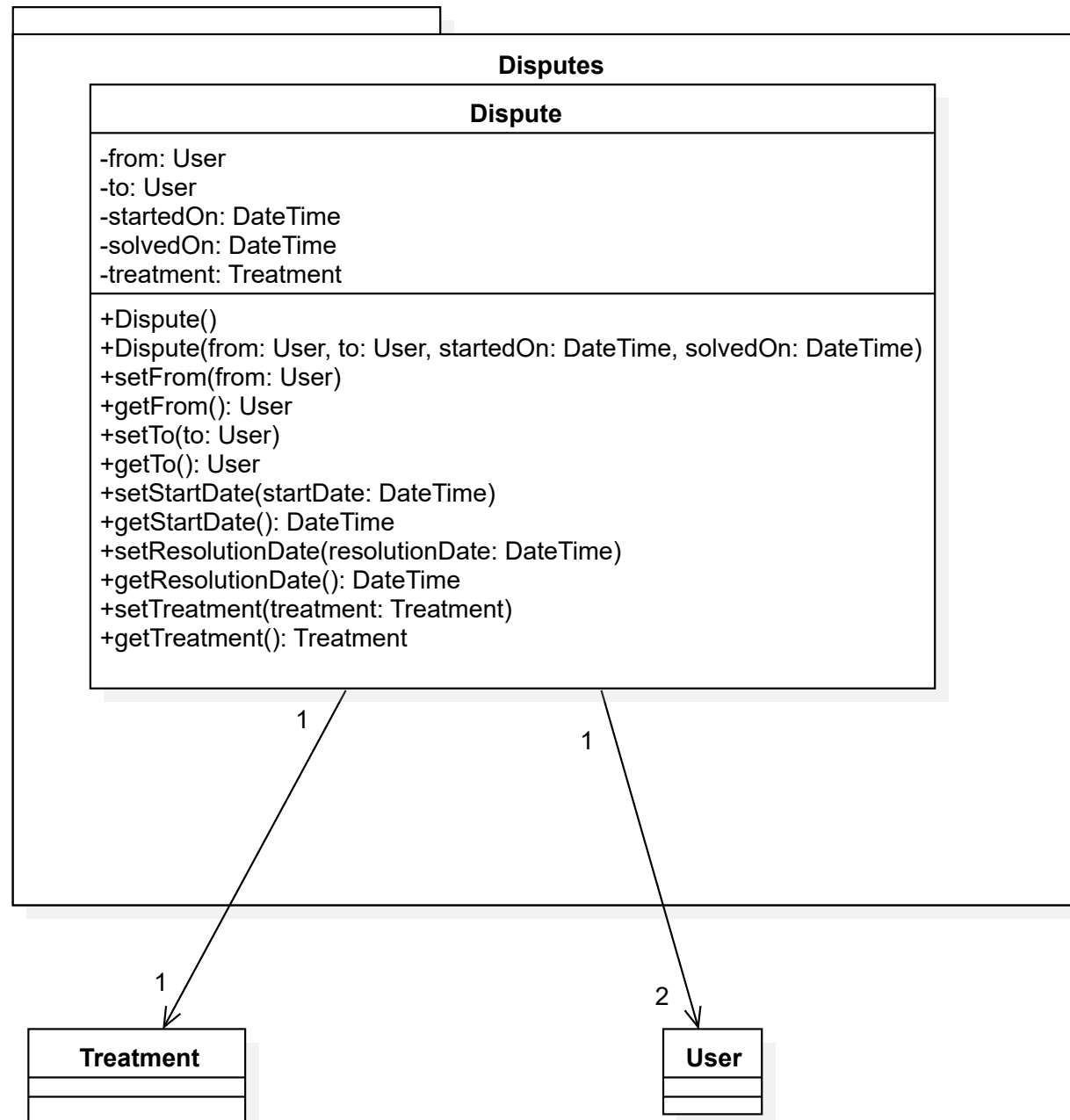
11 Apertura di una controversia

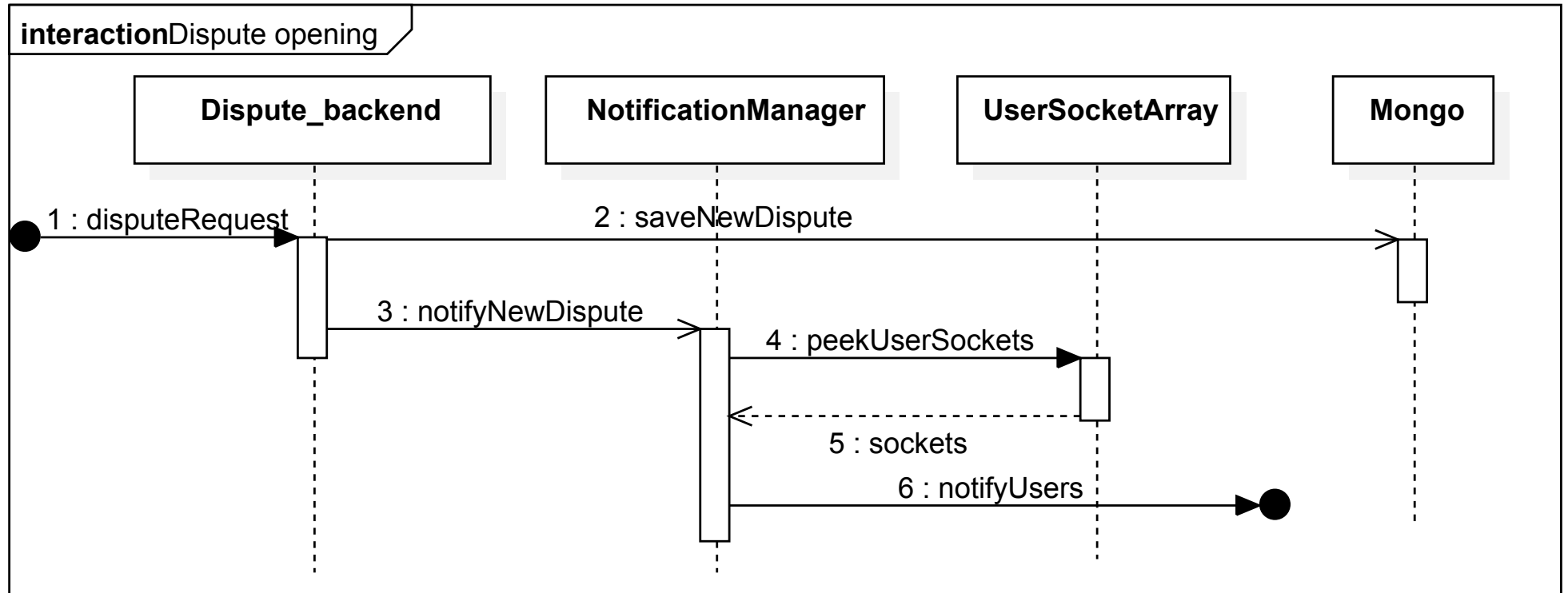
Con lo sviluppo e la progettazione di questo modulo, il gruppo si prende in carico dei seguenti requisiti:

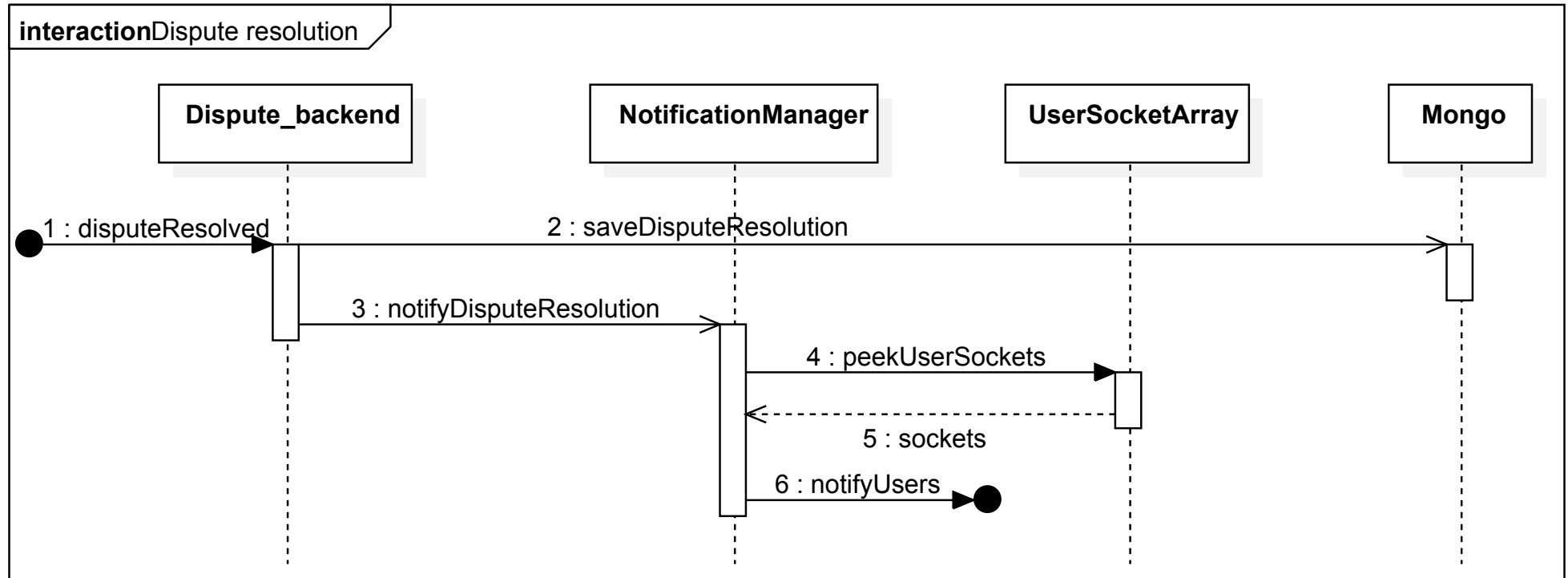
- RF0-7
- RF0-7.1
- RF0-7.2
- RF0-7.3
- RF0-7.4
- RF0-7.4.1
- RF0-7.4.2

11.1 Controversia: back-end

A seguito di un'apertura controversia, il back-end salverà le informazioni su database tramite MongoDB e successivamente notificherà la controparte della presenza di un'apertura controversia.

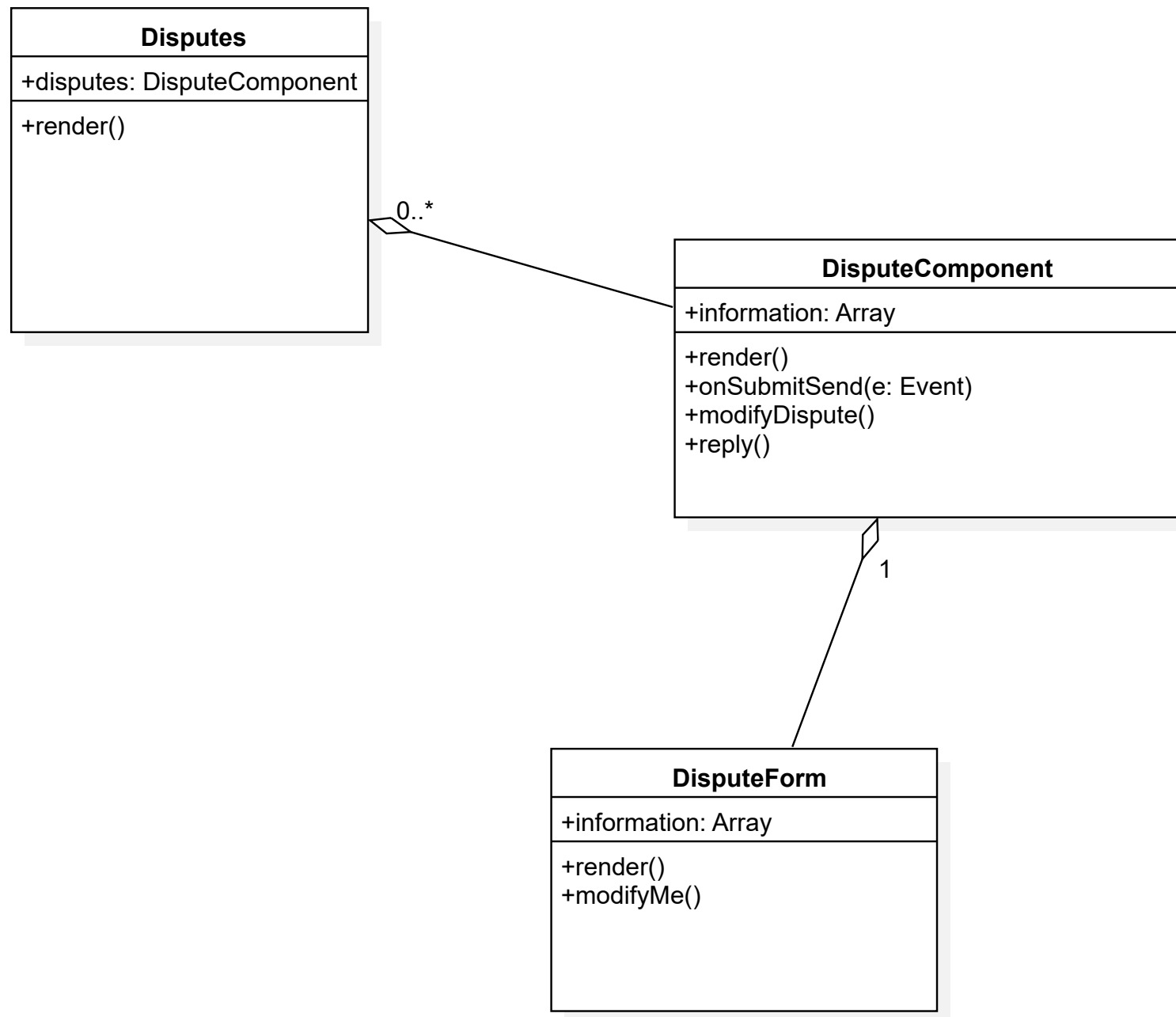


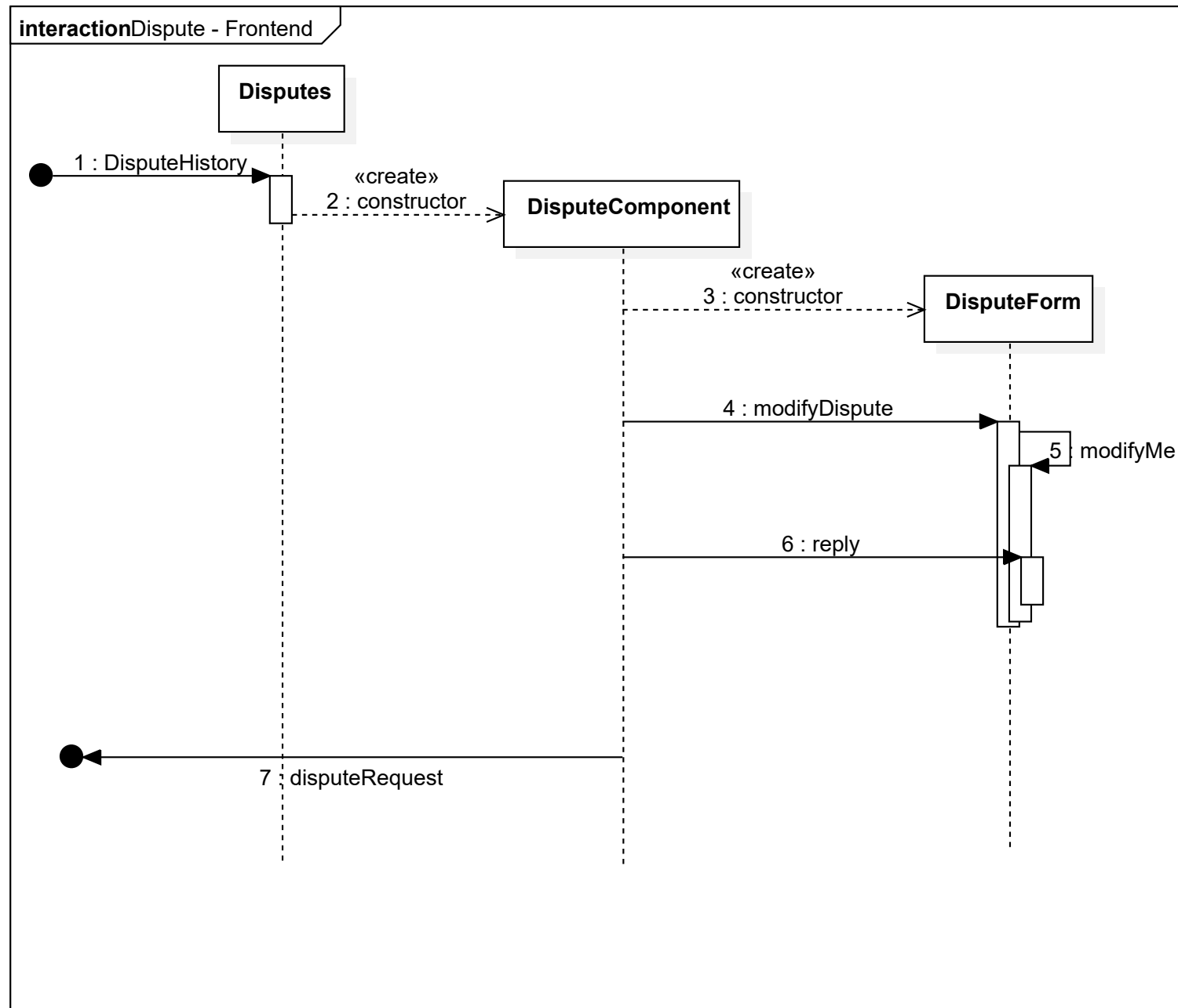




11.2 Controversia: front-end

A seguito di una richiesta di apertura controversia, il front-end verificherà la validità dei dati inseriti ed in caso positivo inoltrerà la richiesta di salvataggio della controversia al back-end.





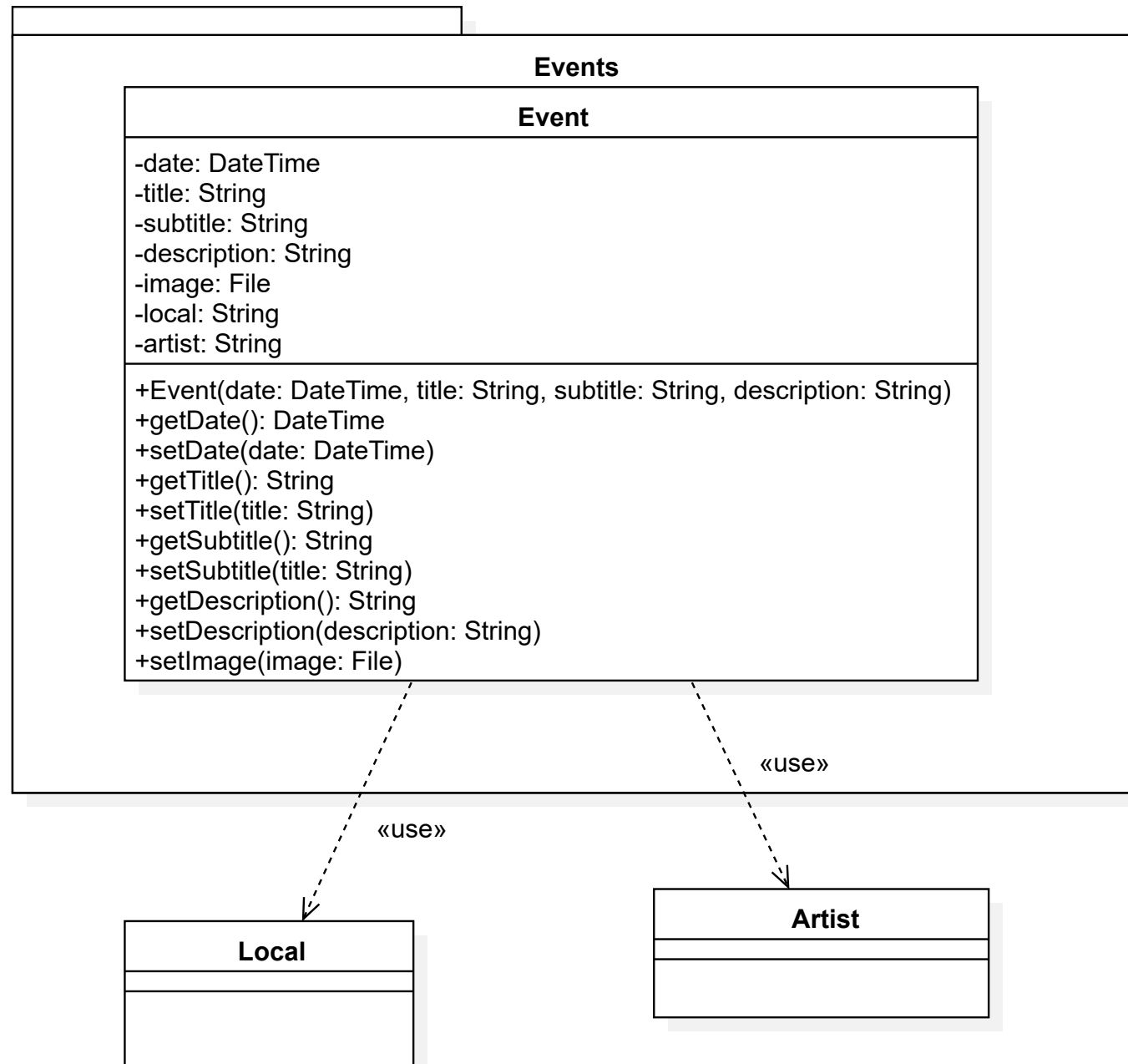
12 Gestione degli eventi

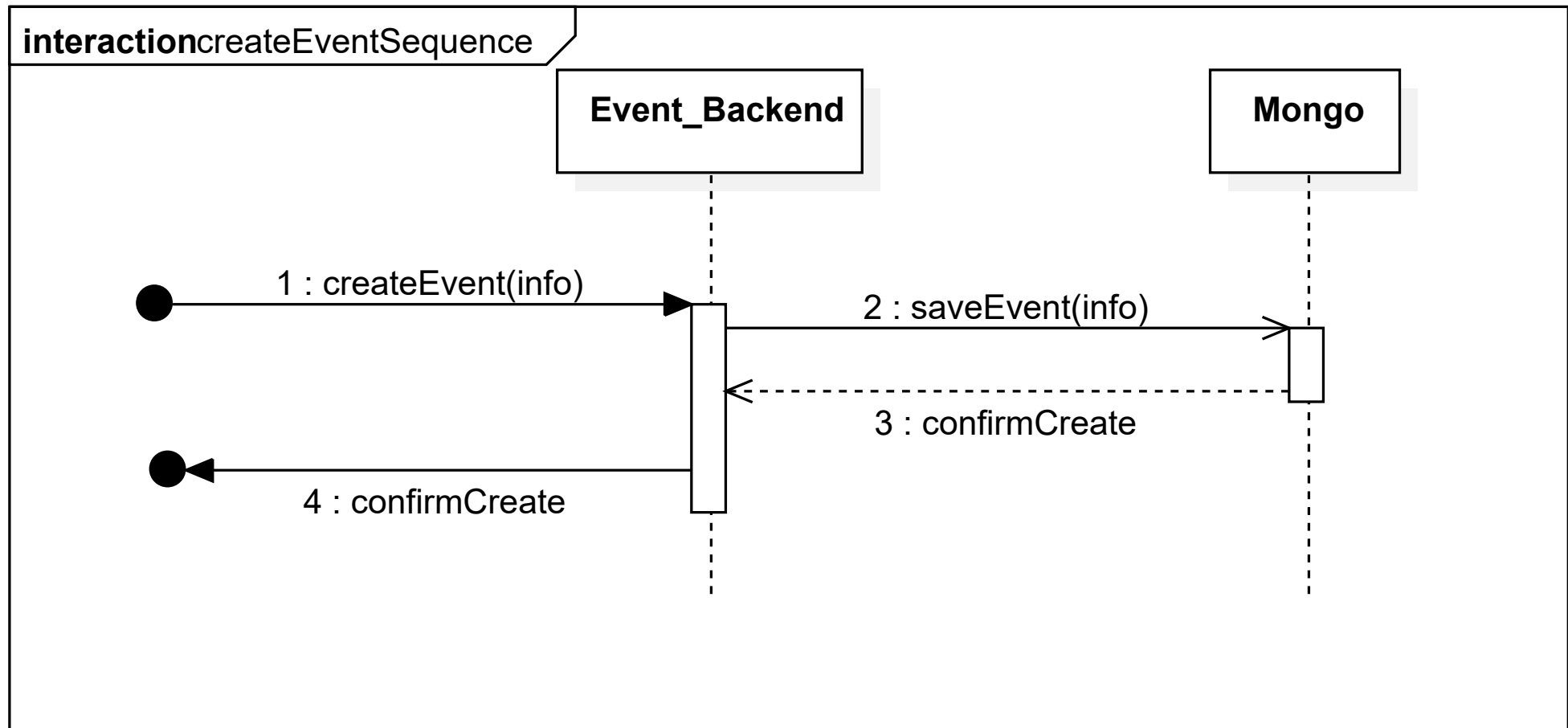
Con lo sviluppo e la progettazione di questo modulo, il gruppo si prende in carico dei seguenti requisiti:

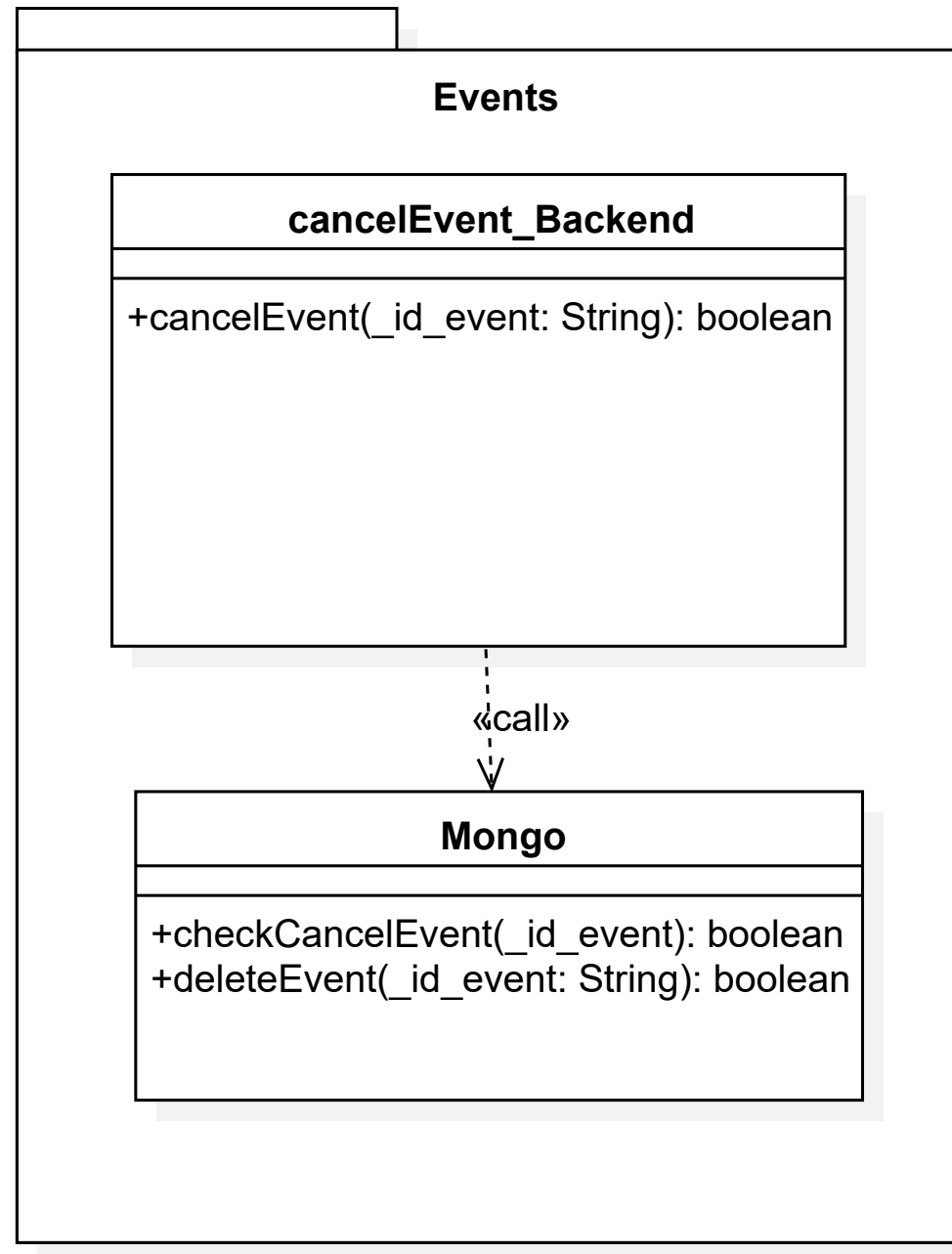
- RF0-8
- RF0-8.1
- RF0-8.1.1
- RF0-8.1.2
- RF0-8.6

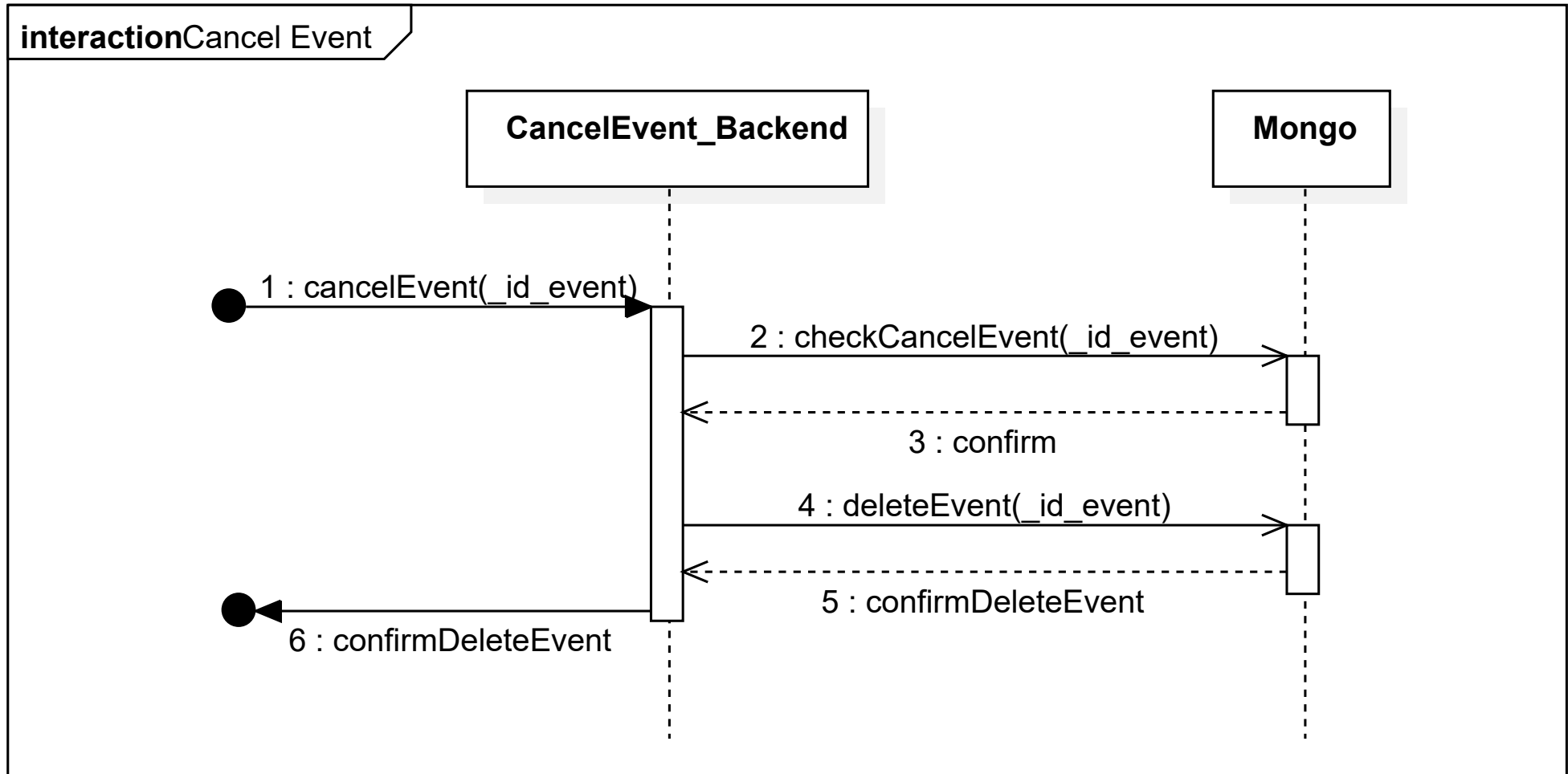
12.1 Gestione degli eventi: back-end

A seguito di una richiesta di creazione o cancellazione di un evento, il back-end effettuerà l'operazione richiesta salvando le modifiche effettuate in database tramite MongoDB. In entrambi i casi restituirà l'esito dell'operazione al front-end.



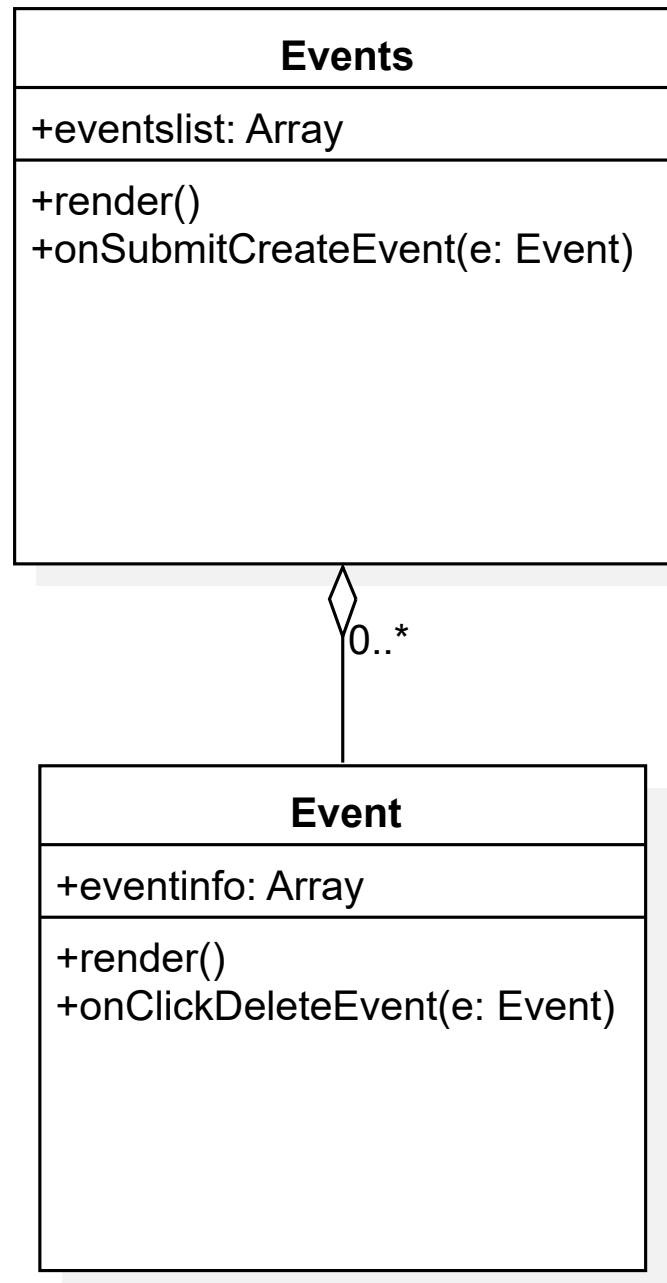


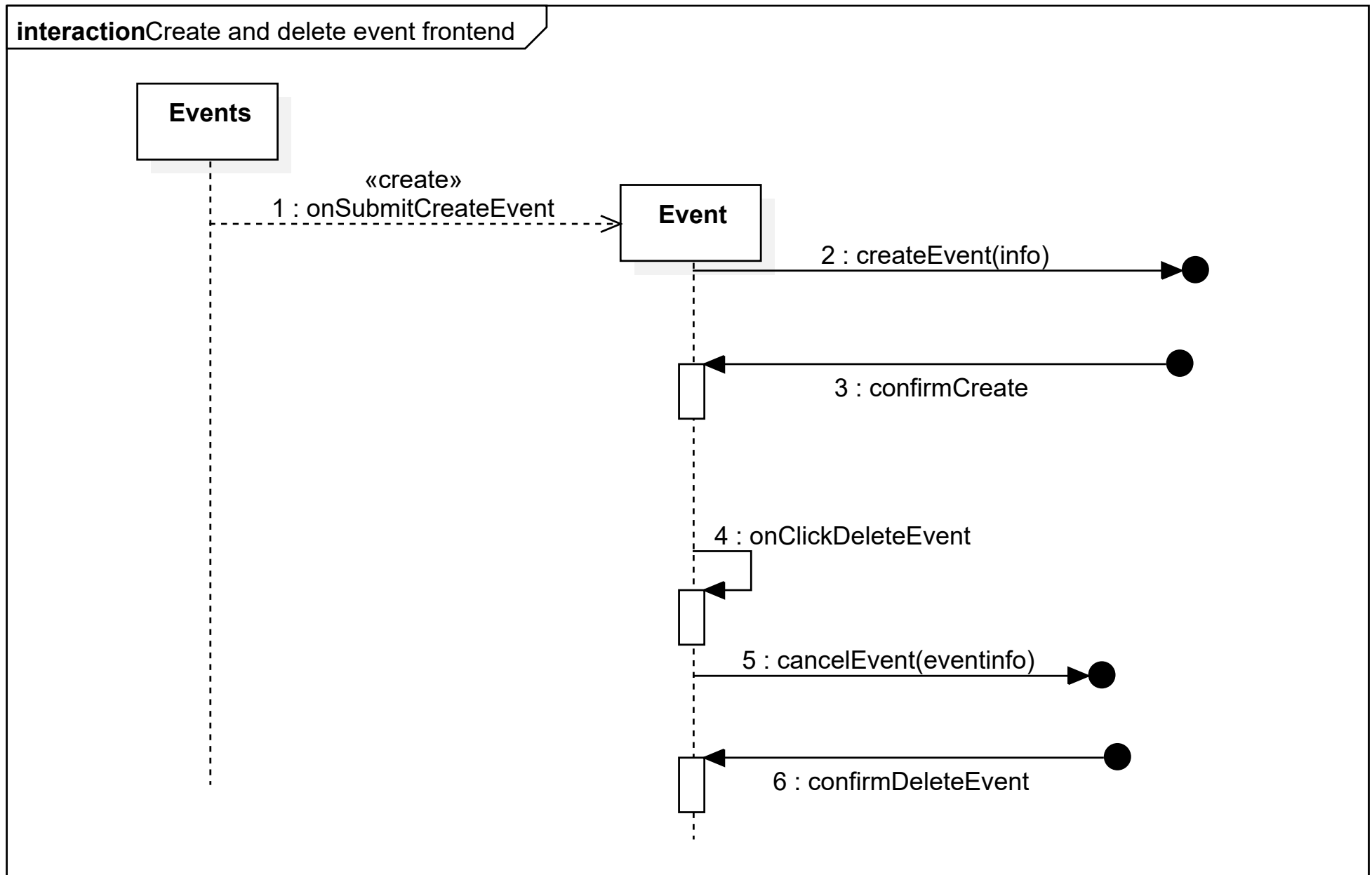




12.2 Gestione degli eventi: front-end

A seguito di una richiesta di creazione o cancellazione di un evento, il front-end inoltrerà la richiesta al back-end il quale ritornerà l'esito dell'operazione richiesta.





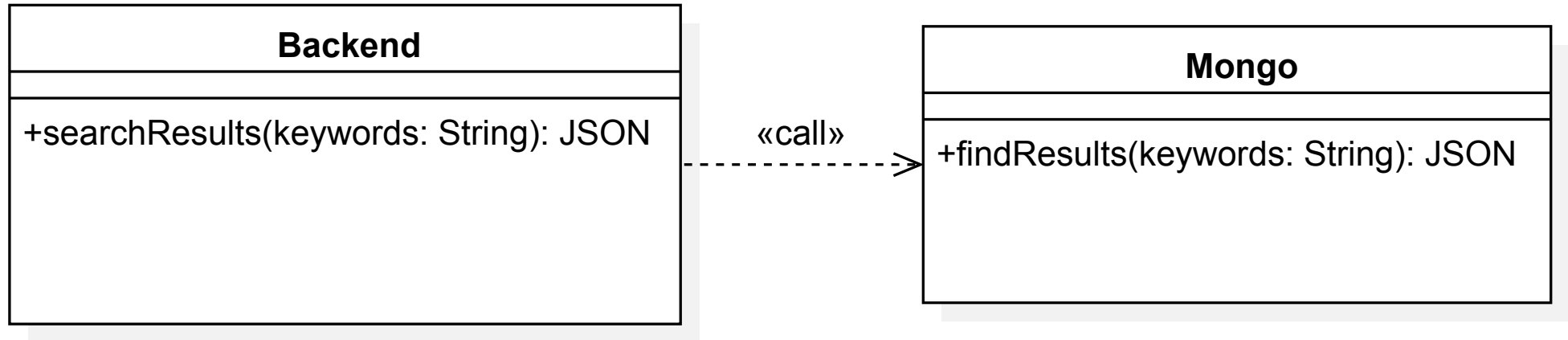
13 Ricerca per keywords

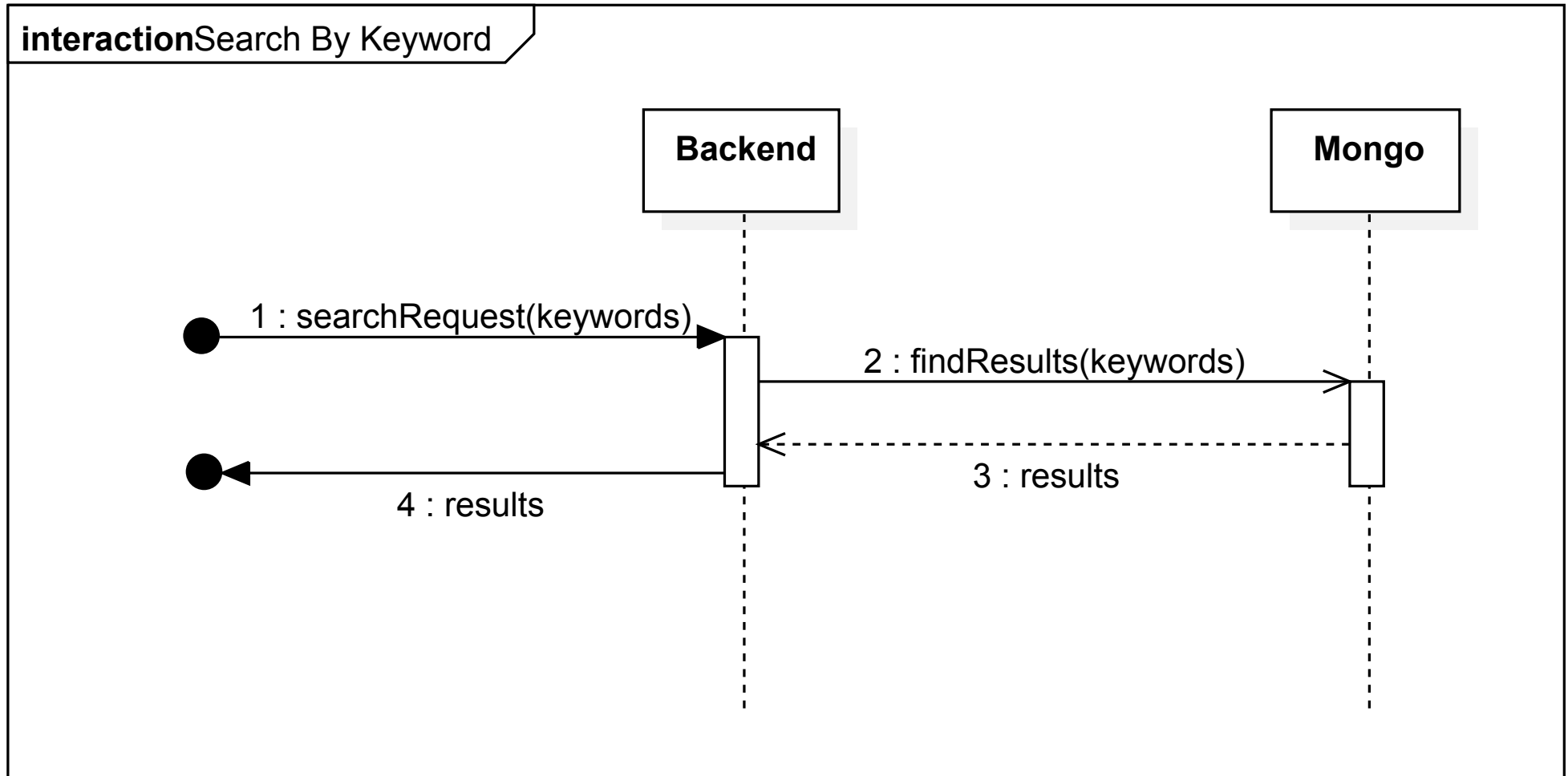
I diagrammi e la progettazione di questo modulo, portano il gruppo alla presa in carico dei seguenti requisiti:

- RF0-3
- RF0-3.1
- RF0-4
- RF0-4.1
- RF0-5
- RF0-5.1

13.1 Ricerca per keywords: back-end

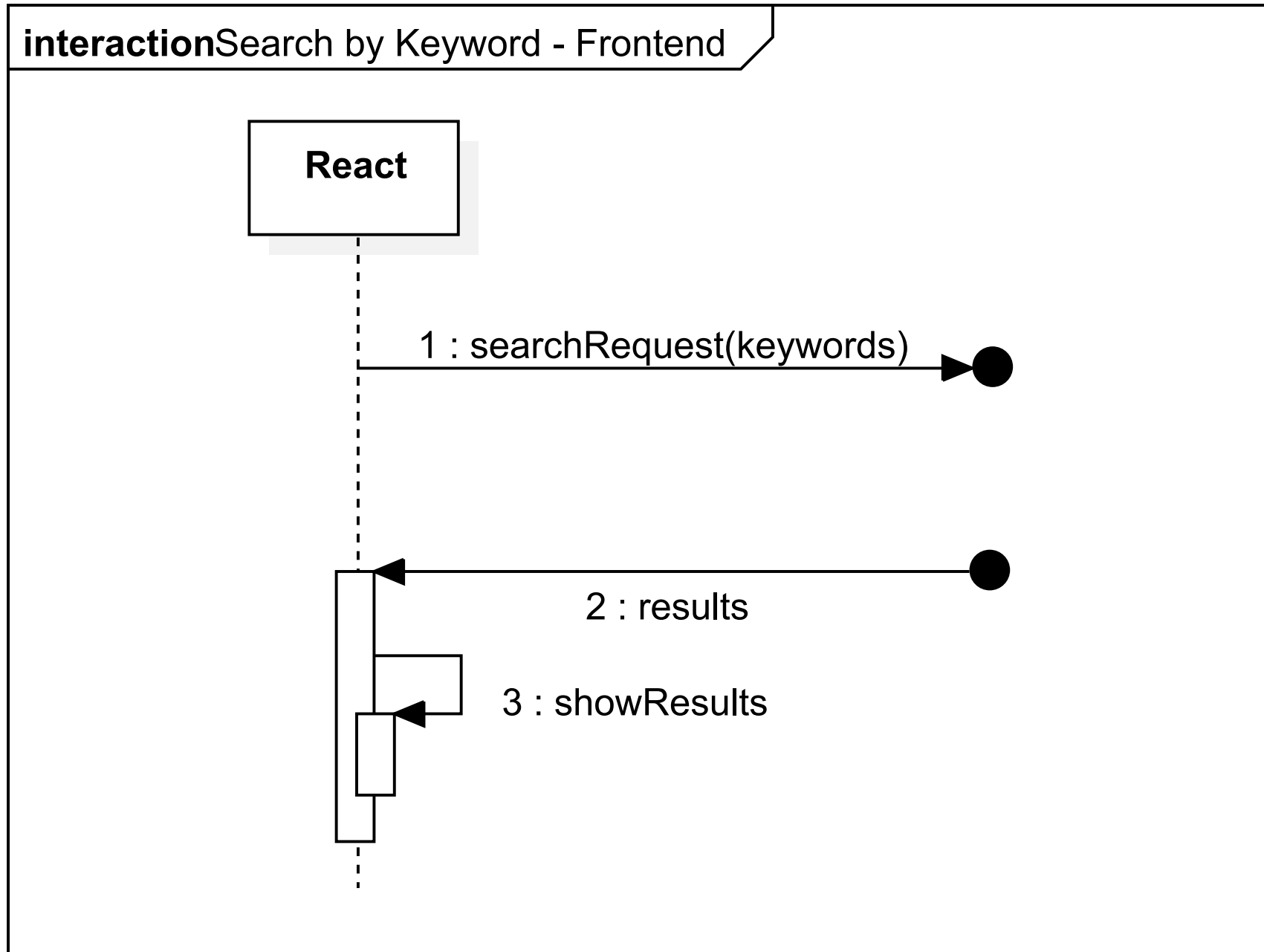
Per la ricerca per keywords facciamo uso di MongoDB per il recupero dei locali su ed eventi in base alle keywords scelte.





13.2 Ricerca per keywords: front-end

A seguito di una ricerca, il front-end si interfaccia con il back-end il quale restituisce i risultati della ricerca per keywords.



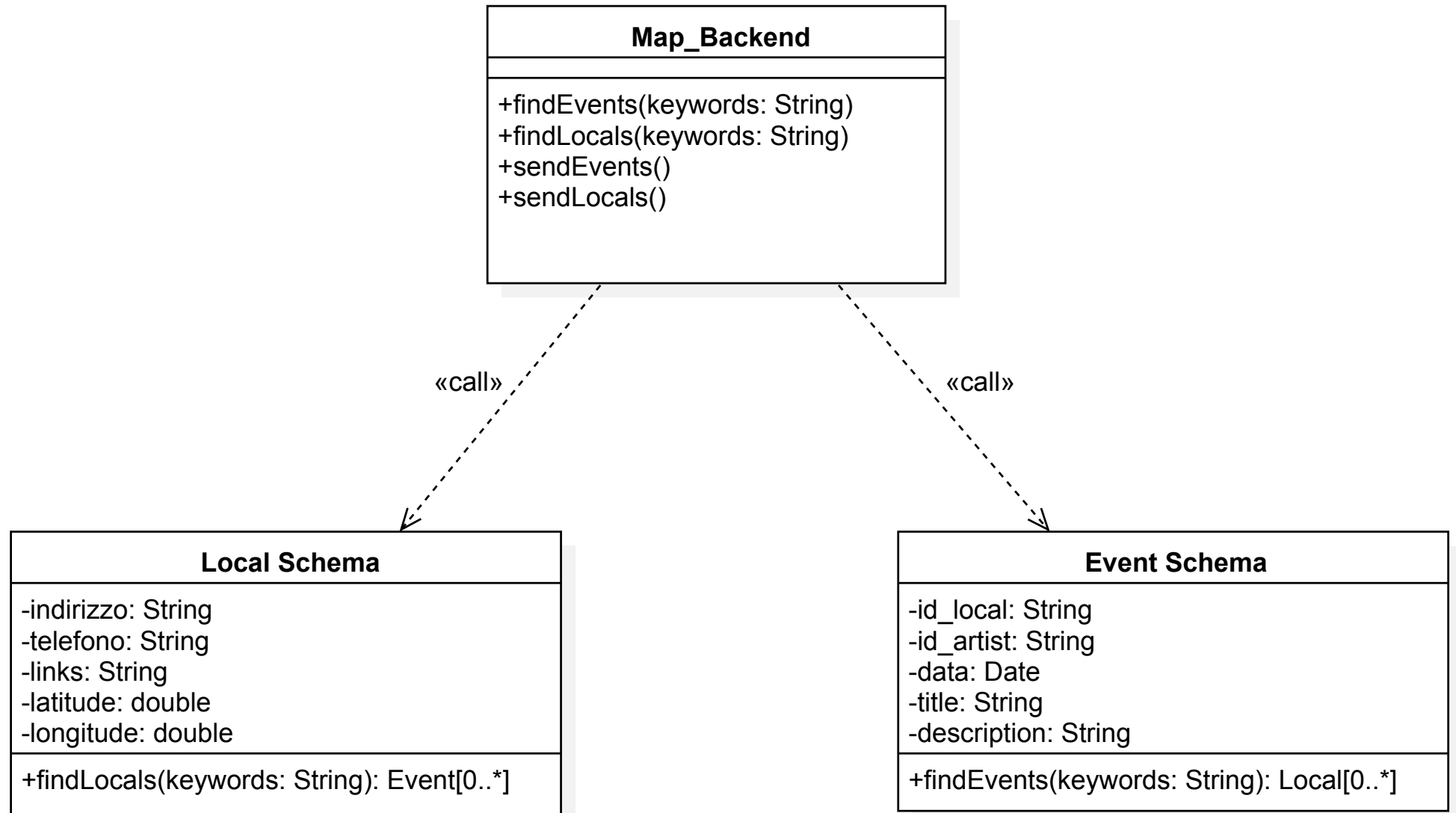
14 Ricerca su mappa

I diagrammi e la progettazione di questo modulo portano alla presa in carico dei requisiti:

- RF1-5
- RF1-5.2
- RF1-6
- RF1-6.2

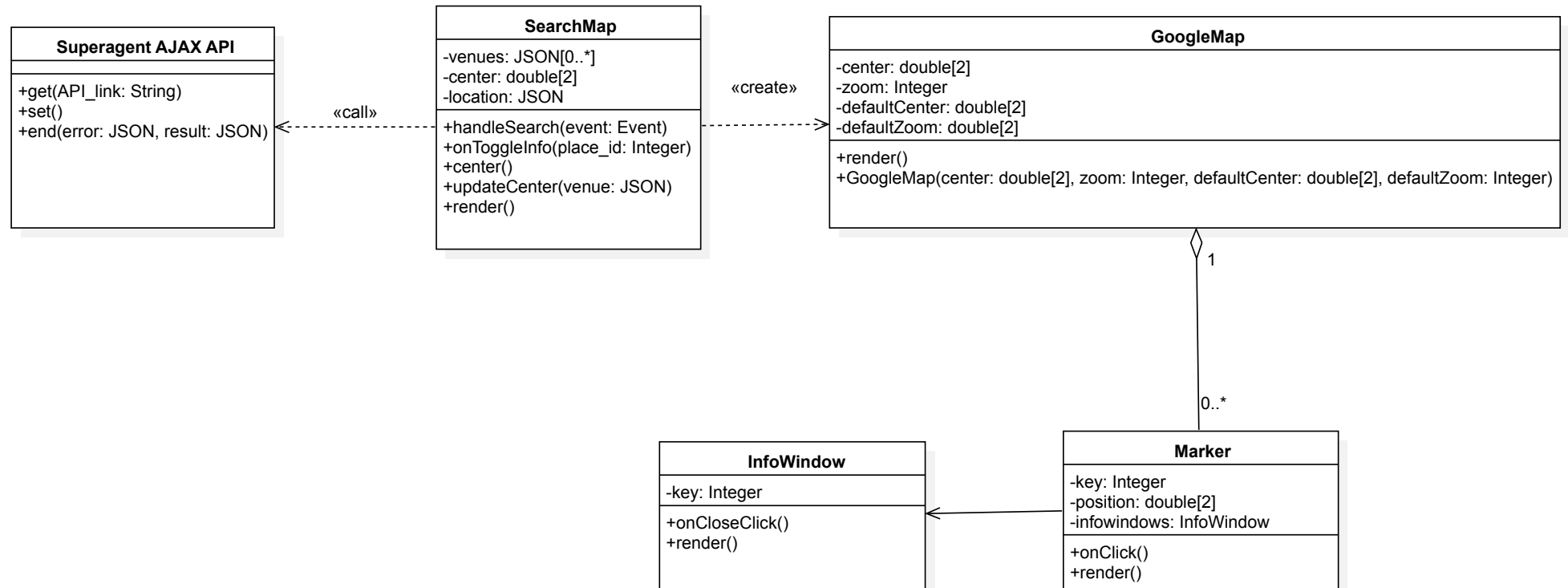
14.1 Ricerca su mappa: back-end

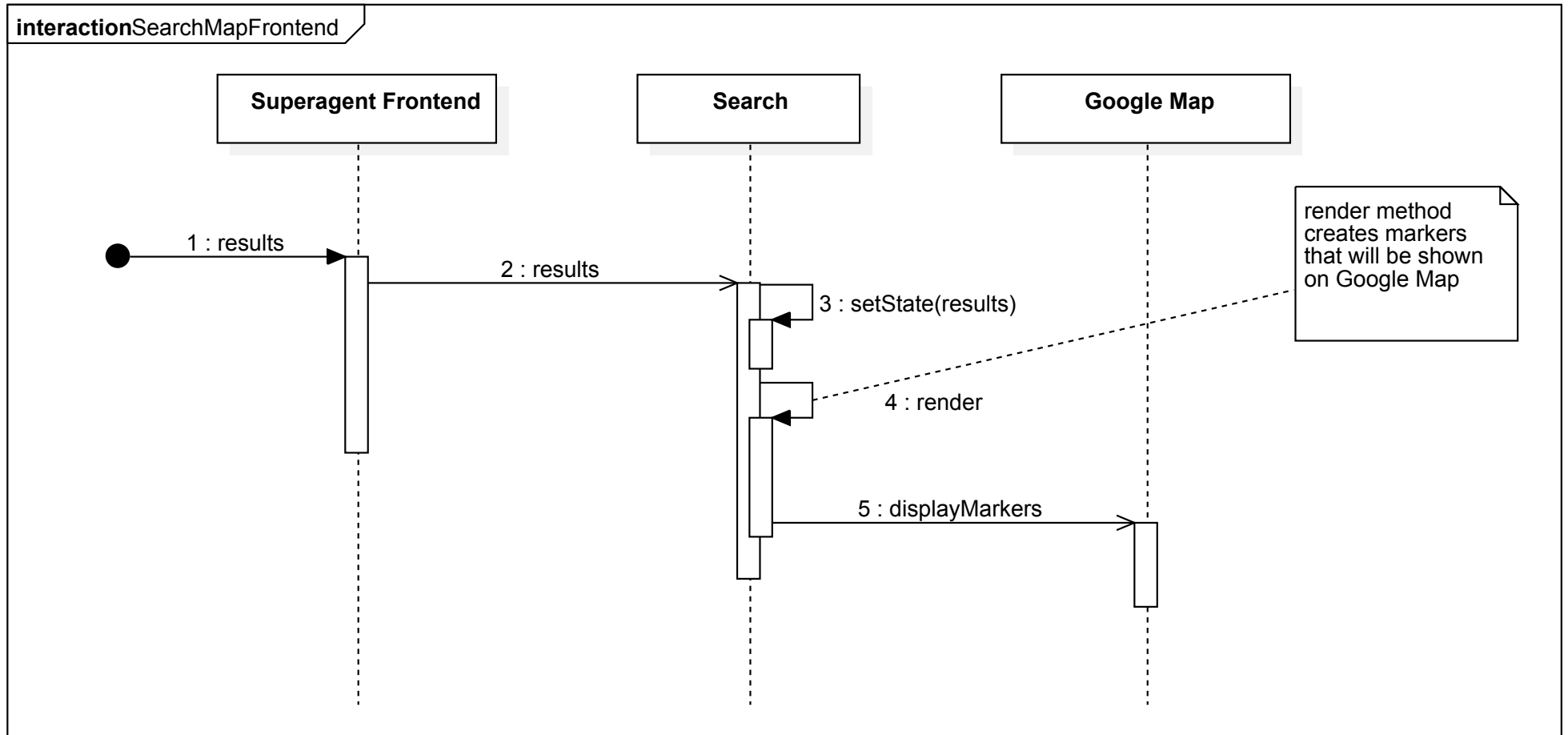
A seguito di una richiesta pervenuta dal front-end, il back-end recupera i risultati attraverso MongoDB e li restituisce al front-end.



14.2 Ricerca su mappa: front-end

A seguito di una ricerca su mappa, il front-end invia una richiesta al back-end e una volta ottenuti i risultati da quest'ultimo, li mostrerà sulla mappa.

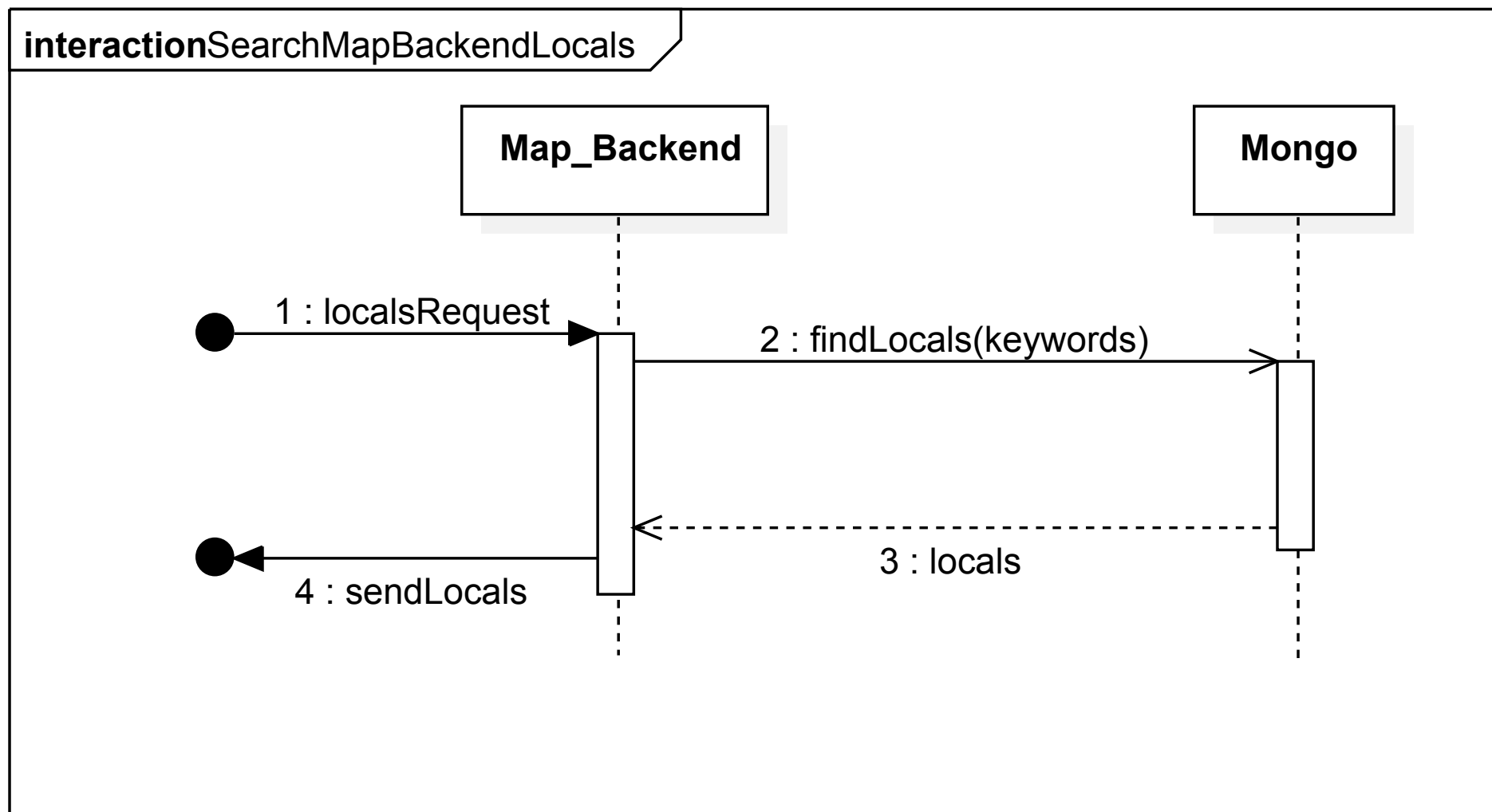




14.3 Ricerca locali su mappa

Per la realizzazione della ricerca locali su mappa, abbiamo deciso di fare uso del modulo react-google-maps, il quale ci permette di avere dei componenti React già predisposti per visualizzare dei locali su una mappa.

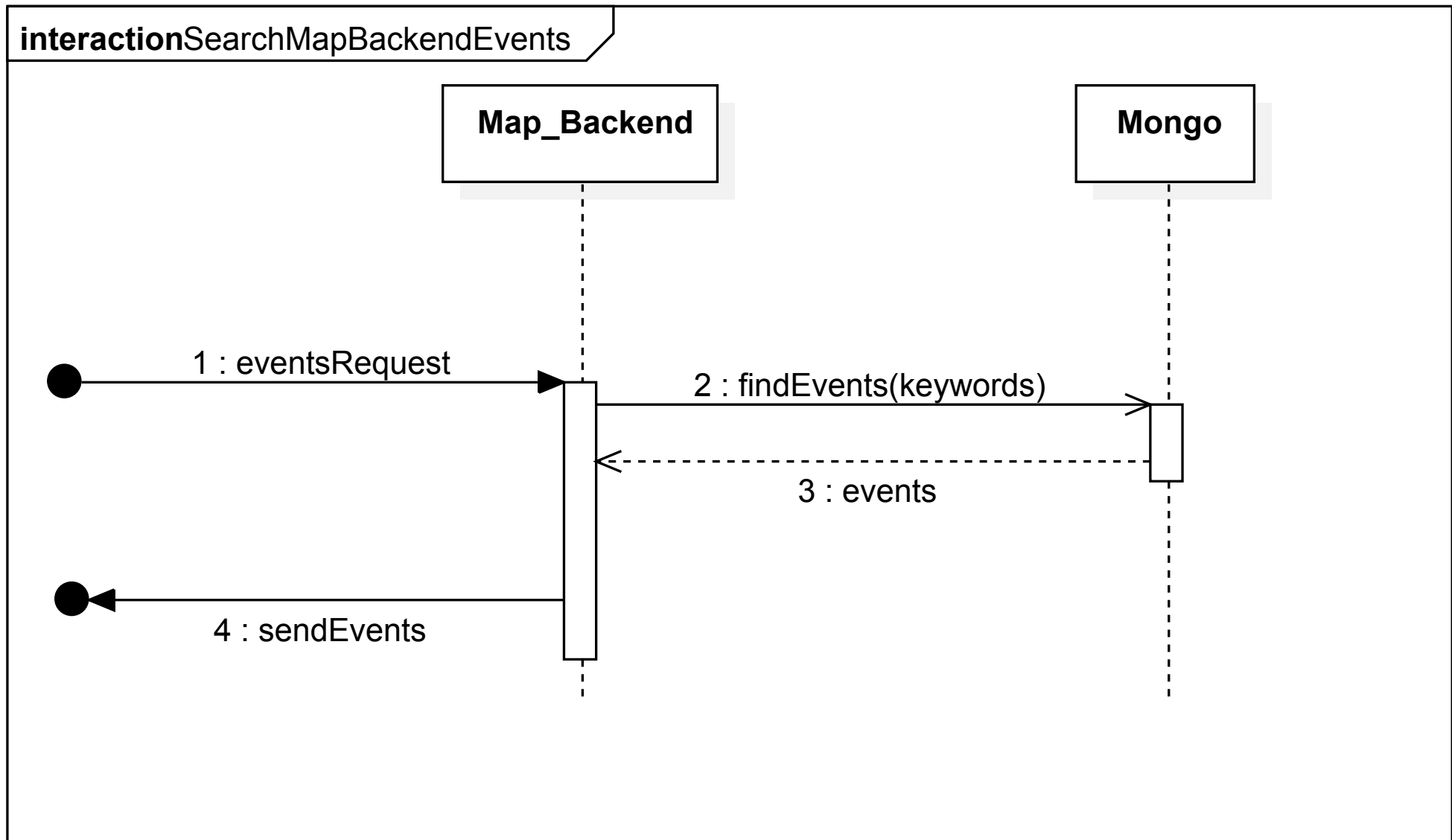
Inoltre si fa uso di MongoDB per il recupero delle coordinate geografiche dei locali.



14.4 Ricerca eventi su mappa

Per la realizzazione della ricerca eventi su mappa, abbiamo deciso di fare uso del modulo react-google-maps, il quale ci permette di avere dei componenti React già predisposti per visualizzare degli eventi su una mappa.

Inoltre si fa uso di MongoDB per il recupero delle coordinate geografiche degli eventi.



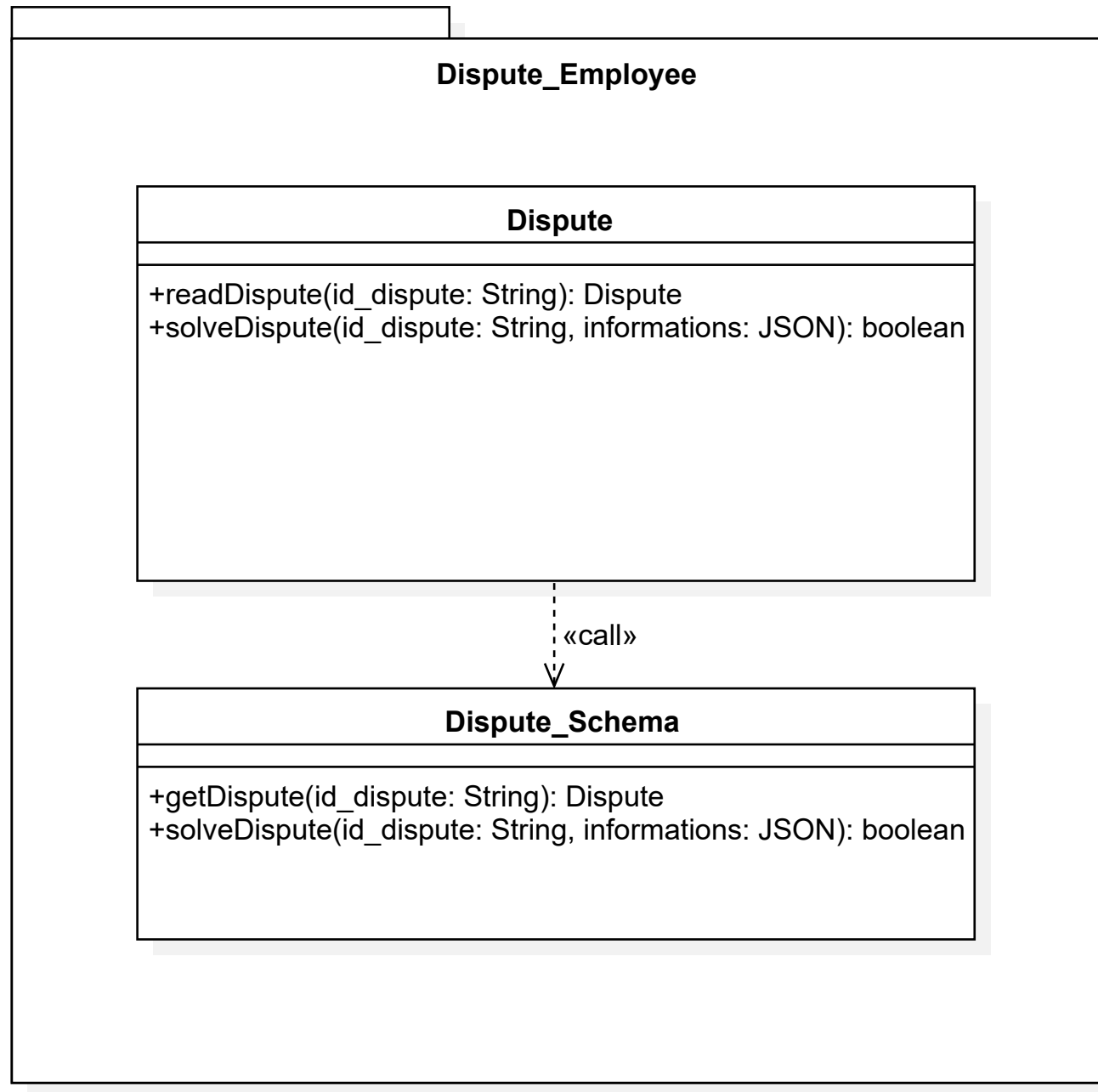
15 Pannello dell'impiegato

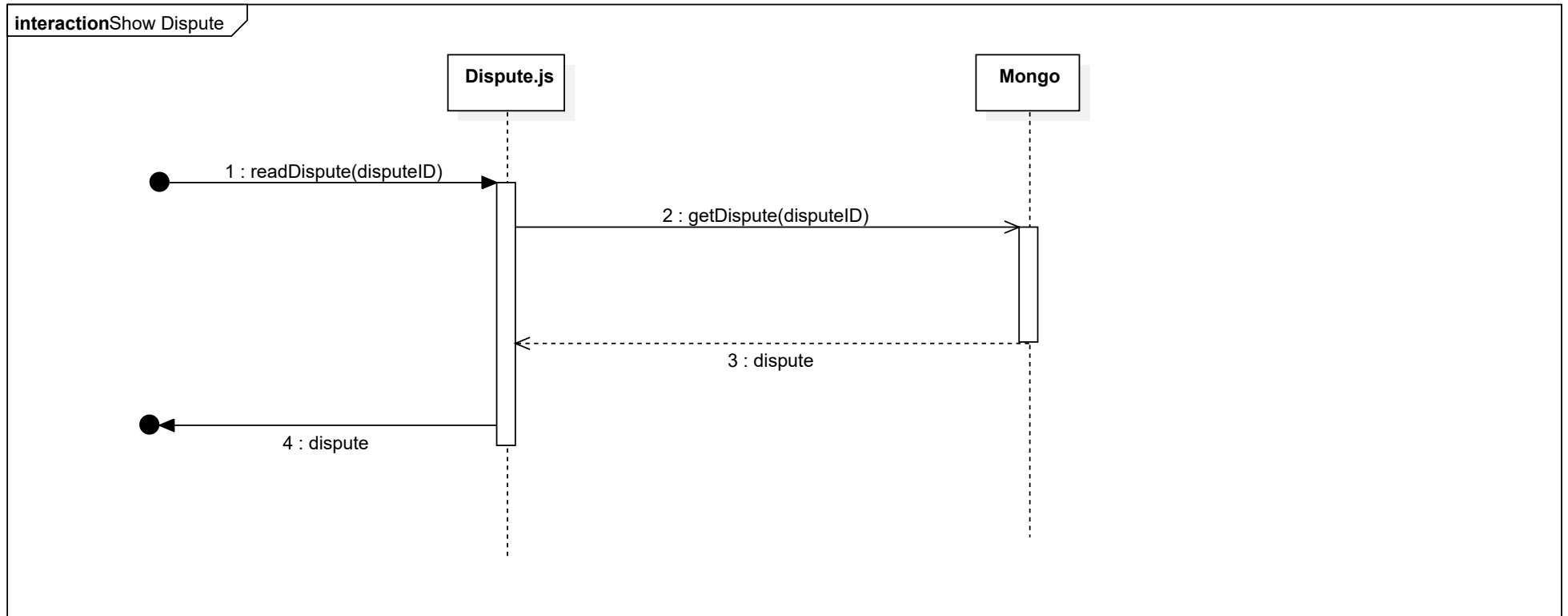
Con lo sviluppo e la progettazione di questo modulo, il gruppo si prende in carico i seguenti requisiti:

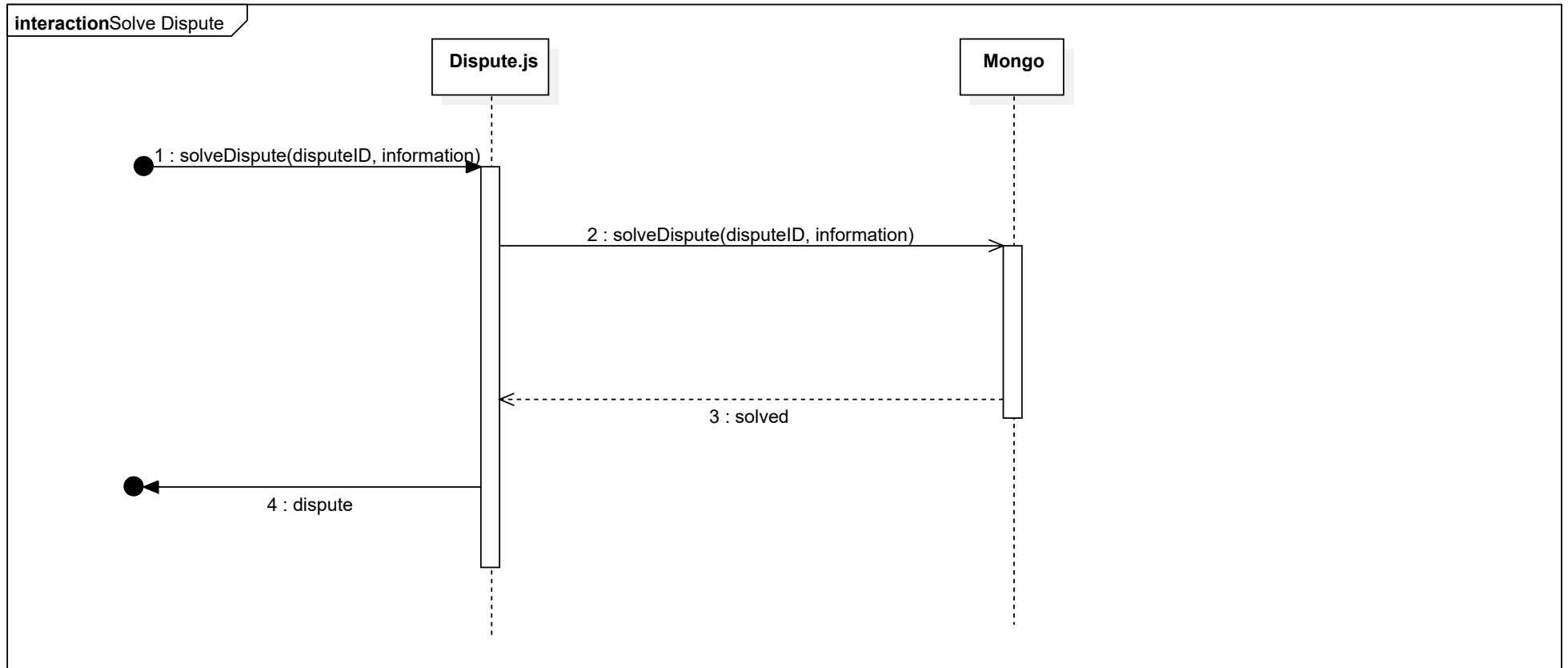
- RF0-15
- RF0-15.1
- RF0-15.2
- RF0-15.3

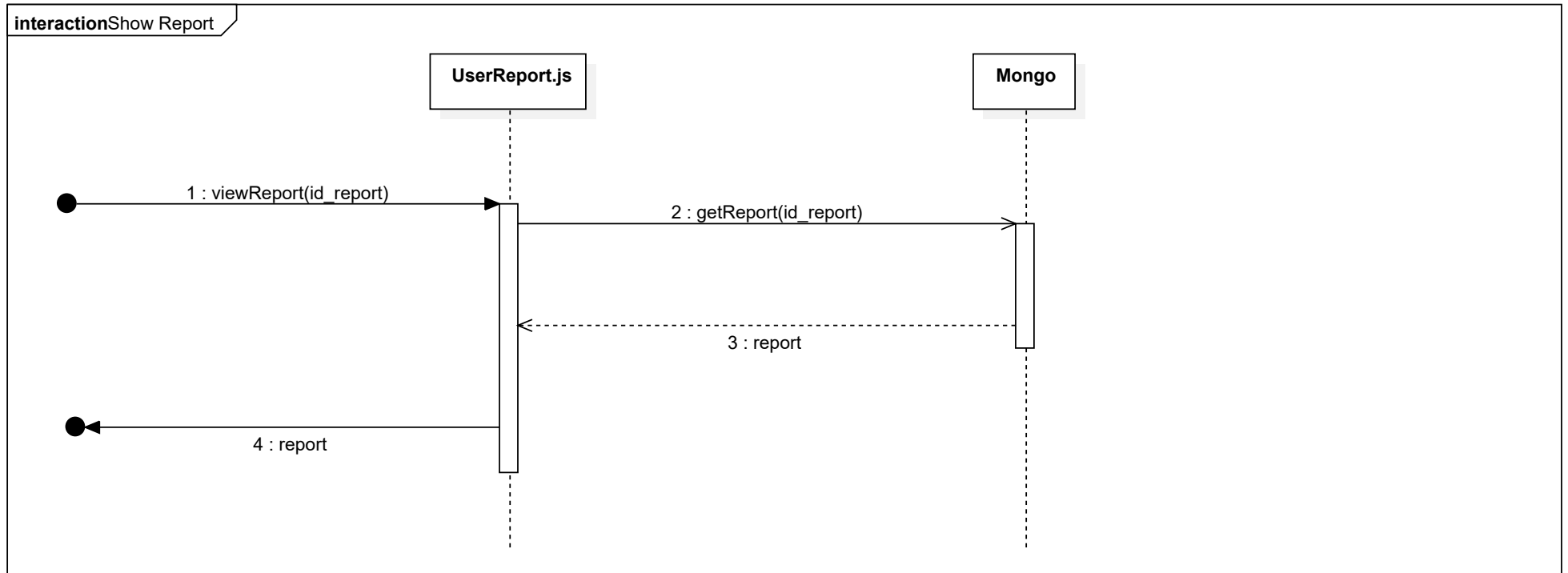
15.1 Pannello dell'impiegato: back-end

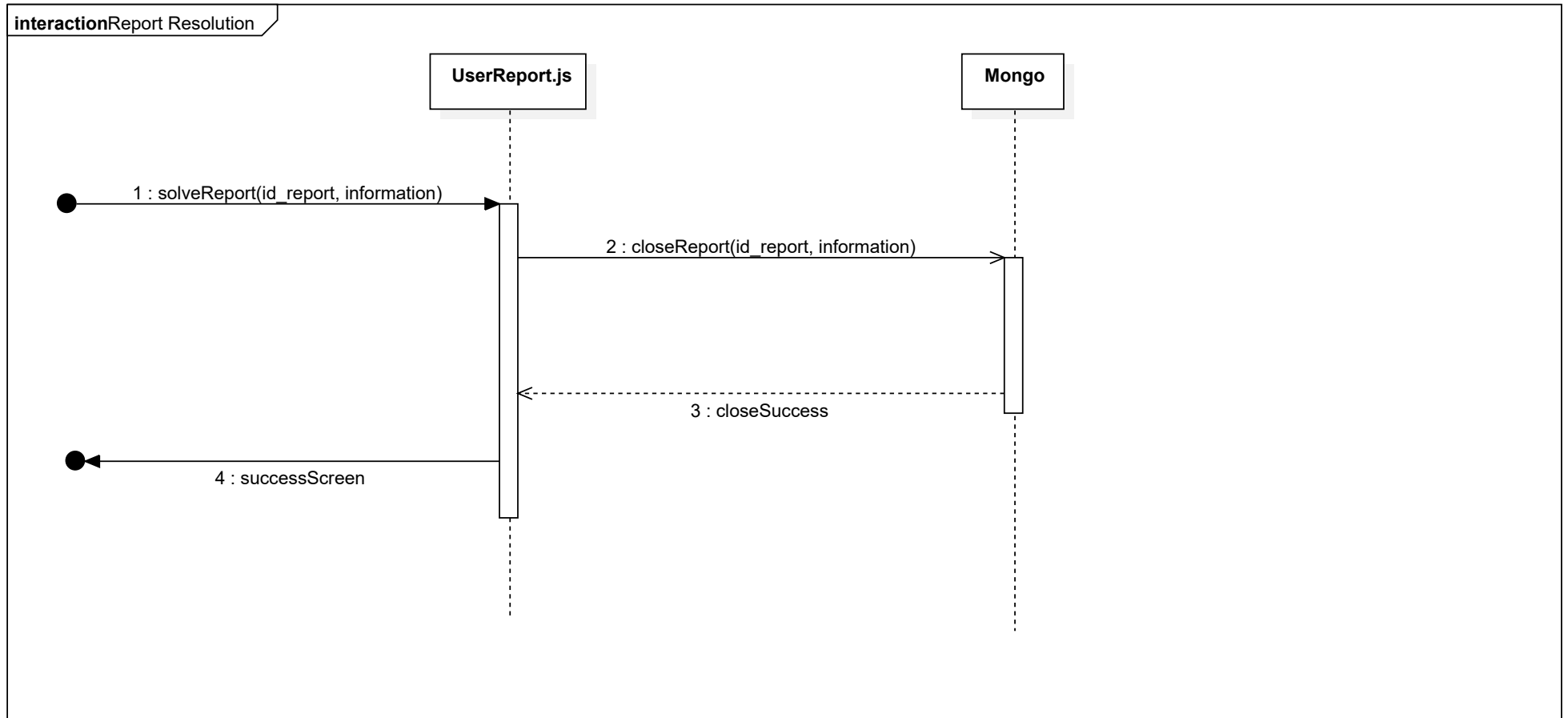
Ricevute le richieste dal front-end, il back-end si occupa della memorizzazione delle controversie e dei risultati della loro risoluzione, oltre alle segnalazioni degli utenti.





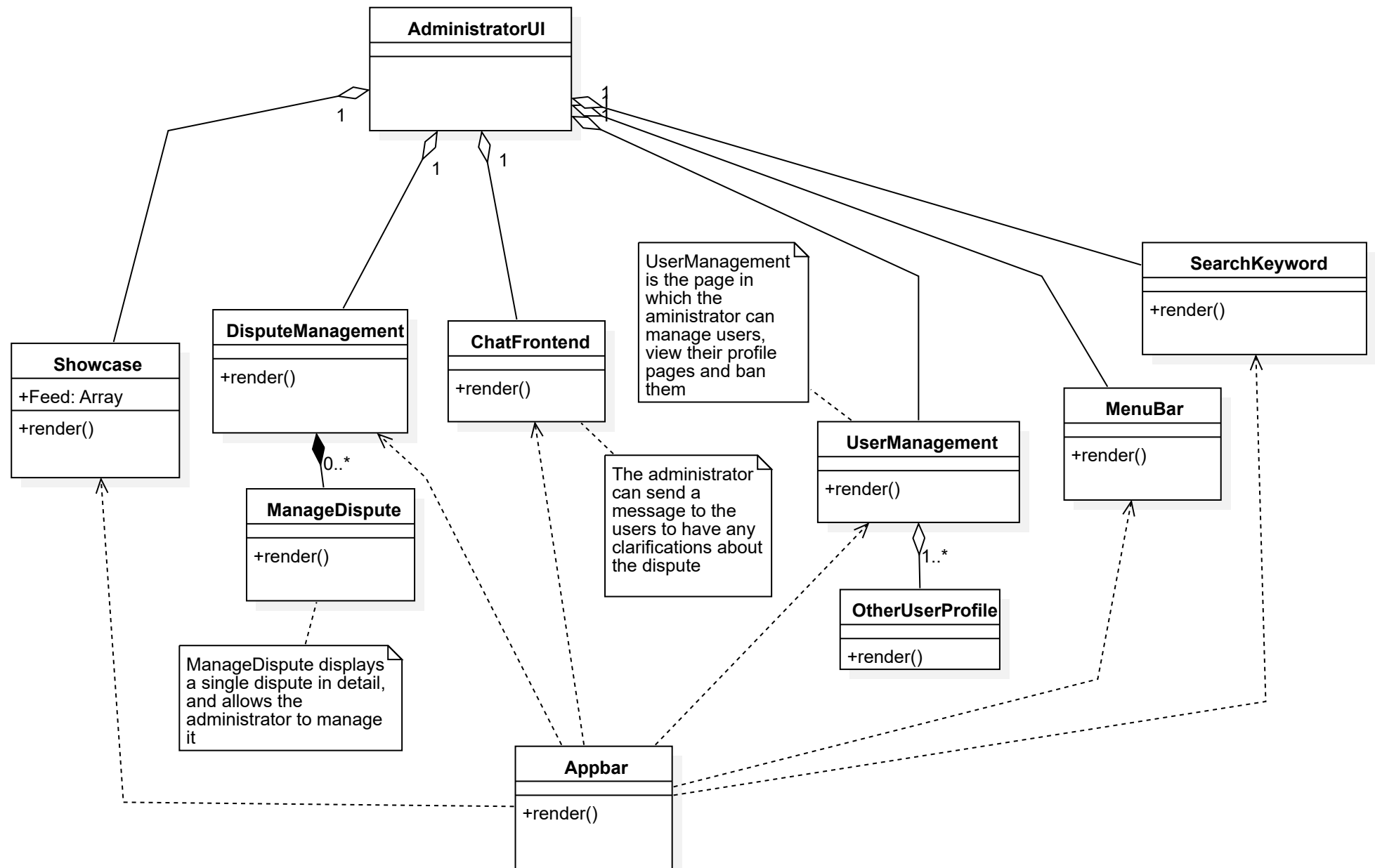


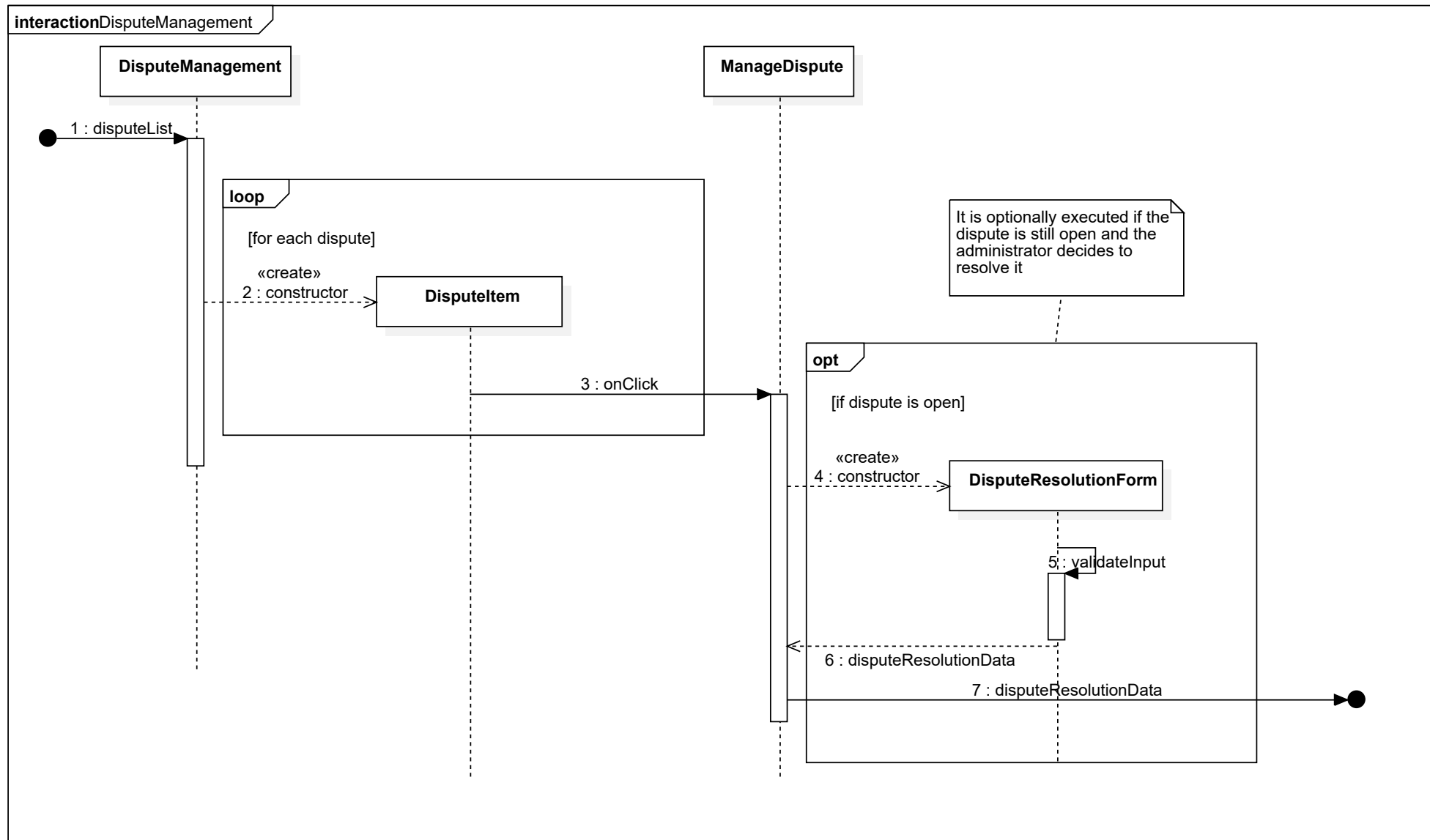




15.2 Pannello dell'impiegato: front-end

Nel pannello dell'impiegato, l'amministratore TuTourSelf può visualizzare le controversie, inviare messaggi agli altri utenti per avere chiarimenti sulle controversie e ovviamente la sua sezione per amministrare i profili degli utenti.





16 Segnalazione utente

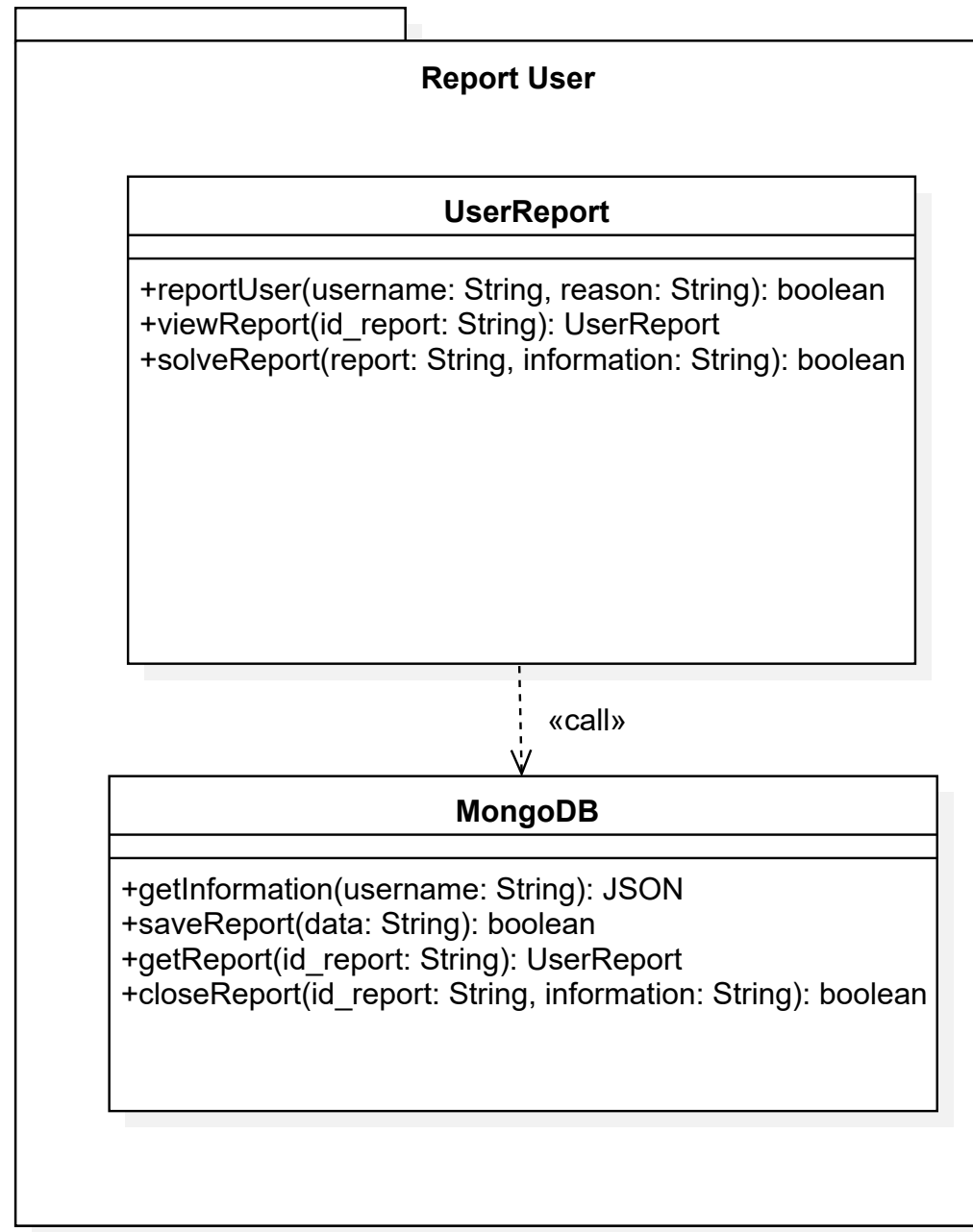
I diagrammi e la progettazione di questo modulo portano alla presa in carico dei requisiti:

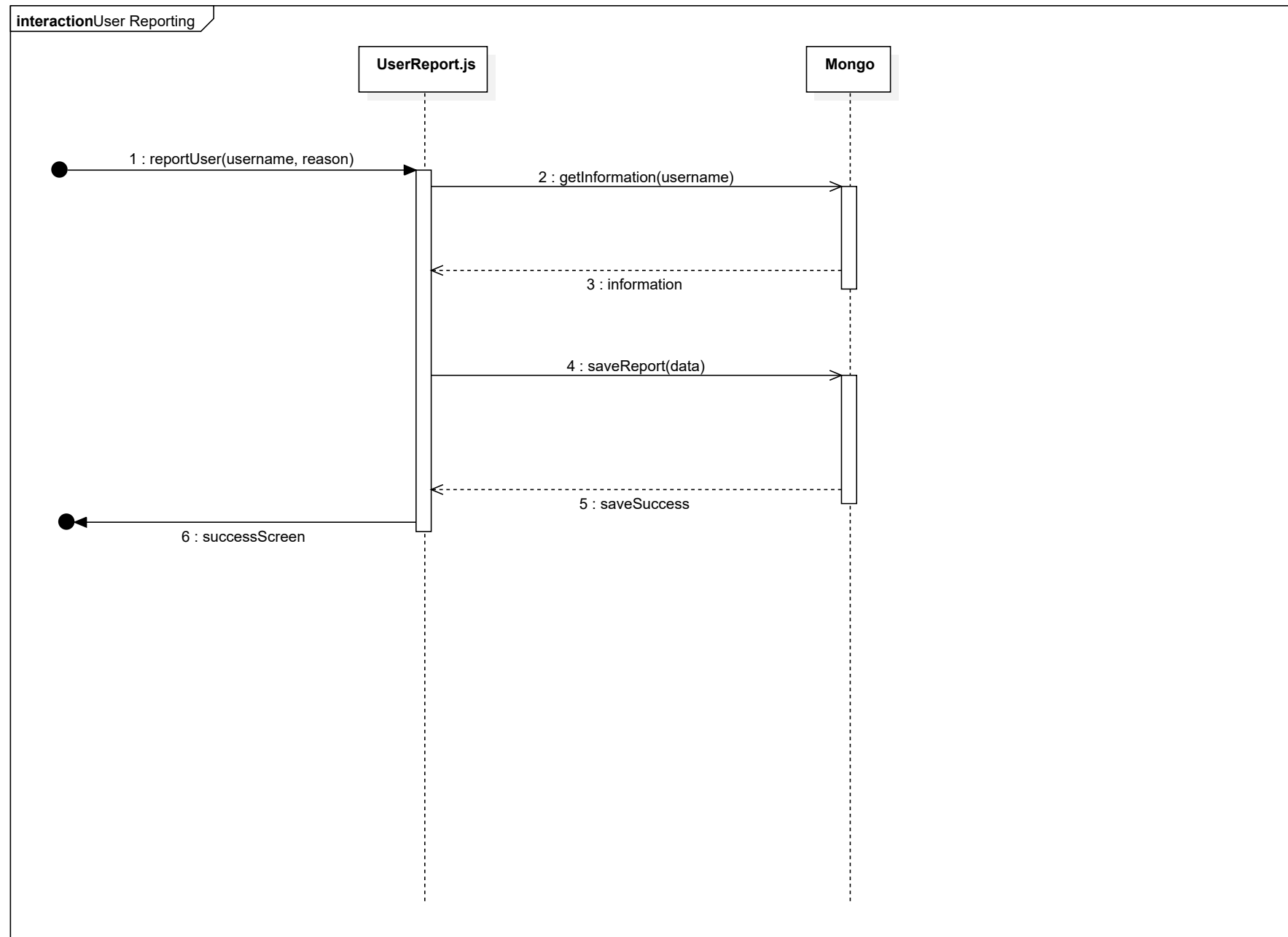
- RF1-11
- RF1-11.1
- RF1-11.2
- RF1-11.3

16.1 Segnalazione utente: back-end

Un utente può segnalare il comportamento truffaldino di un altro utente. Quando viene fatta una segnalazione essa viene registrata all'interno del database con tutte le informazioni relative all'utente e alla segnalazione.

Viene fatta, quindi, una iterazione con MongoDB.





16.2 Segnalazione utente: front-end

Per i diagrammi delle classi e di sequenza della segnalazione utente si fa riferimento a § 3, in quanto un utente per segnalare un altro utente deve recarsi nella sua pagina profilo.