



Documentazione progetto Ingegneria del Software

Software di gestione dei pazienti diabetici

Università di Verona

Imbriani Paolo - VR500437

Irimie Fabio - VR501504

5 luglio 2025

Indice

1	Requisiti e Use Case	3
1.1	Note generali	3
1.2	Casi d'uso del paziente	3
1.2.1	Inserimento dei dati giornalieri	4
1.3	Casi d'uso del medico	5
1.3.1	Visualizzare i dati del paziente	5
1.3.2	Modificare i dati del paziente	6
1.4	Ricezione delle notifiche	7
1.5	Casi d'uso dell'amministratore	8
1.5.1	Gestione dell'utente	9
1.5.2	Gestione delle richieste di registrazione	9
1.6	Visualizzazione della traccia dei medici	9
1.7	Activity diagram	10
1.7.1	Registrazione di un utente	10
1.7.2	Attività del paziente	11
1.7.3	Attività del medico	12
1.7.4	Attività dell'amministratore	13
2	Sviluppo	13
2.1	Processo di sviluppo	13
2.2	Progettazione e pattern architetturali usati	15

1 Requisiti e Use Case

1.1 Note generali

Il software che si andrà a sviluppare è un sistema di telemedicina di un servizio clinico per la gestione di pazienti diabetici. Gli attori principali del sistema sono i *medici* (diabetologi) e i *pazienti*; questi ultimi hanno credenziali di accesso al sistema fornite dagli amministratori del servizio con cui possono autenticarsi. Se l'autenticazione va a buon fine allora l'utente verrà indirizzato alla propria *home page* in cui potrà visualizzare le informazioni relative al loro ruolo. Nel seguente diagramma dei casi d'uso sono rappresentati i principali attori e le loro interazioni con il sistema. Notare che diamo per scontato che tutti gli autori siano già autenticati per semplificare il diagramma.

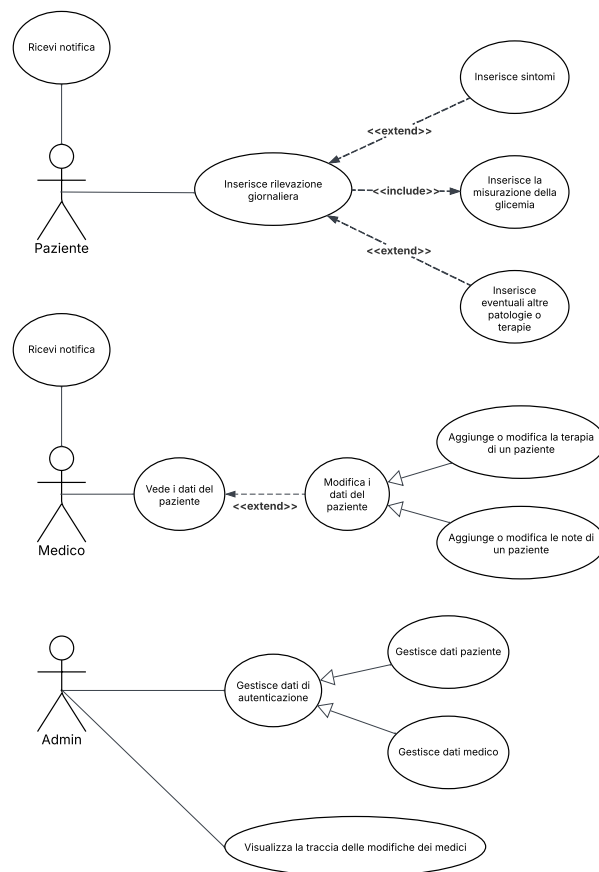


Figura 1: Diagramma dei casi d'uso

1.2 Casi d'uso del paziente

Una volta che il paziente si è autenticato ed è entrato all'interno della sua area riservata, egli può inserire i dati giornalieri relativi alla sua glicemia (prima e

dopo ogni pasto); per fare questo ha bisogno di inserire dati quali: sintomi, rilevazione glicemica, eventuali altre patologie o terapie.

1.2.1 Inserimento dei dati giornalieri

Attore: Paziente

Precondizioni: Il paziente deve essere autenticato

Passi:

1. Il paziente accede alla sua home page
2. Il paziente entra dentro l'area di inserimento dei dati giornalieri
3. Il paziente inserisce il dato della sua glicemia prima pasto e dopo pasto
 - Il paziente può aprire un ulteriore finestra per inserire i sintomi, le terapie e le patologie, con opportuna data
 - Può anche inserire le assunzioni di insulina o qualsiasi farmaco prescritto dal diabetologo, specificandone giorno, ora, farmaco e quantità assunta
4. Il paziente inserisce la data e ora di rilevazione
5. Il paziente conferma l'inserimento dei dati

Postcondizioni: La rilevazione è inserita

Andiamo a specificare come viene gestito **l'inserimento dei dati** del paziente: i dati glicemici sono quelli che il paziente inserisce giornalmente e obbligatori per inviare le rilevazioni. Dopo aver inserito i dati, l'utente può anche:

- inserire specifiche sulla patologia, terapia o sintomi
- inserire le assunzioni di insulina o qualsiasi altro farmaco prescritto dal diabetologo

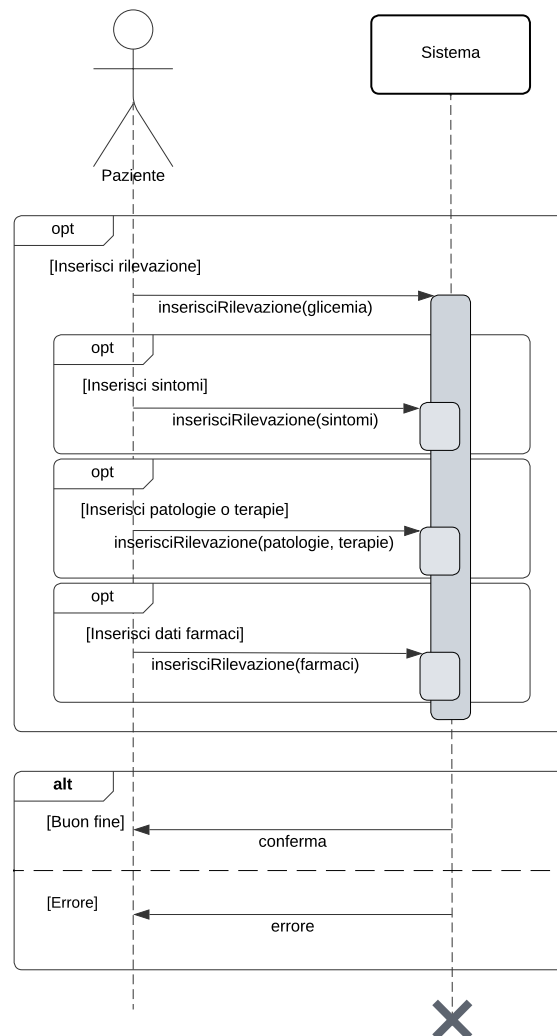


Figura 2: Sequence Diagram della rilevazione del paziente

1.3 Casi d'uso del medico

1.3.1 Visualizzare i dati del paziente

Attore: Medico

Precondizioni: Il medico deve essere autenticato

Passi:

1. Il medico accede alla sua area riservata
2. Il medico può accedere alla lista dei pazienti
3. Il medico può visualizzare i dati del paziente e lo storico delle rilevazioni

Postcondizioni: nessuna

Il medico una volta che ha acceduto alla sua area riservata può visualizzare i dati di tutti i pazienti.

1.3.2 Modificare i dati del paziente

Attore: Medico

Precondizioni: Il medico deve essere autenticato

Passi:

1. Il medico accede alla sua area riservata
2. Il medico accede alla lista dei pazienti
3. Il medico seleziona il paziente che vuole gestire
4. Il medico può decidere tra le seguenti opzioni:
 - Aggiungere o modificare la terapia del paziente
 - Aggiungere o modificare le note di un paziente
5. Il medico conferma le modifiche
6. Il sistema aggiorna i dati del paziente

Postcondizioni: La modifica è effettuata

Sequenza alternativa 1: il medico può in qualunque momento decidere di annullare le modifiche e ritornare alla lista dei pazienti

Postcondizioni: La modifica non è effettuata

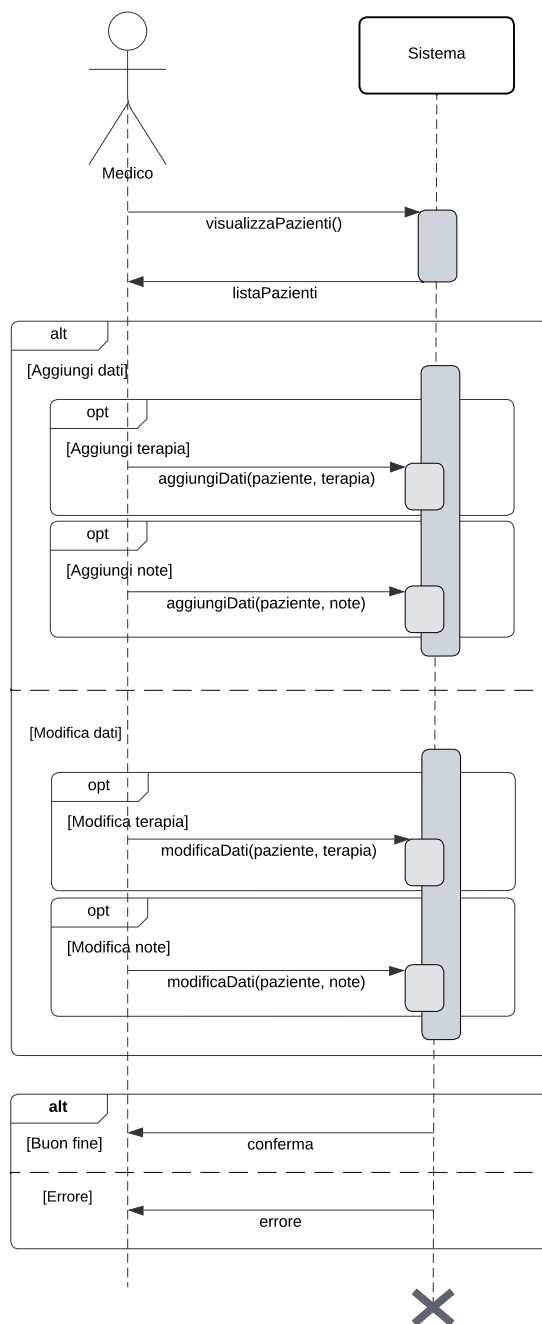


Figura 3: Sequence Diagram del medico

1.4 Ricezione delle notifiche

Il paziente può ricevere notifiche dal sistema; se il paziente si dimentica di assumere i farmaci il sistema può inviare una notifica per ricordarglielo.

Attore: Paziente o Medico

Precondizioni: Il paziente o medico deve essere autenticato

Passi:

1. Il paziente o medico accede alla sua home page
2. Il paziente o medico entra dentro la sezione delle notifiche
3. Il paziente o medico può visualizzare:
 - Notifiche già lette
 - Notifiche non lette

Postcondizioni: Se viene visualizzata una notifica, questa viene marcata come letta

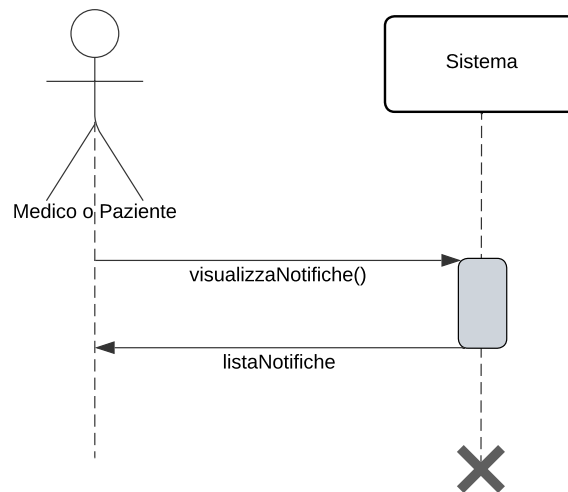


Figura 4: Sequence Diagram della notifica con lista

Il sistema invia delle notifiche ai seguenti attori:

- **Al paziente:** per ricordargli di inserire i dati giornalieri
- **Al medico di riferimento del paziente:** per avvisare che il paziente non ha seguito per più di tre giorni consecutivi le prescrizioni.
- **A tutti i medici:** per segnalare i pazienti che registrano livelli di glicemia sopra le soglie indicate.

1.5 Casi d'uso dell'amministratore

L'amministratore del servizio può gestire gli utenti del sistema. Si occupa di creare, modificare e cancellare gli account dei pazienti e dei medici. Per agevolare l'aggiunta di utenti, ci si può registrare tramite un form di registrazione (che specificherà il ruolo), ma l'amministratore deve comunque approvare la registrazione per rendere l'utente attivo nel sistema.

1.5.1 Gestione dell'utente

Attore: Amministratore

Precondizioni: L'amministratore deve essere autenticato

Passi:

1. L'amministratore accede alla sua area riservata
2. L'amministratore visualizza la lista degli utenti
3. L'amministratore può:
 - Creare un nuovo utente
 - Modificare un utente
 - Cancellare un utente

Postcondizioni: L'utente è creato, modificato o cancellato

1.5.2 Gestione delle richieste di registrazione

Attore: Amministratore

Precondizioni: L'amministratore deve essere autenticato

Passi:

1. L'amministratore accede alla sua area riservata
2. L'amministratore visualizza la lista delle richieste
3. Seleziona una richiesta di registrazione
4. L'amministratore può:
 - Accettare una registrazione
 - Rifiutare una registrazione

Postcondizioni: L'utente è accettato o rifiutato

1.6 Visualizzazione della traccia dei medici

Il sistema tiene traccia di ogni cambiamento o modifica che i medici effettuano sui pazienti, per ragioni di sicurezza.

- L'amministratore può visualizzare questa traccia per verificare che i medici non stiano effettuando operazioni non autorizzate.
- Il medico può visualizzare la traccia per verificare le operazioni effettuate precedentemente su un paziente da parte di altri medici

Attore: Amministratore

Precondizioni: L'amministratore deve essere autenticato

Passi:

1. L'amministratore accede alla sua area riservata
2. L'amministratore accede alla pagina di visualizzazione delle tracce

Postcondizioni: nessuna

Attore: Medico

Precondizioni: Il medico deve essere autenticato

Passi:

1. Il medico accede alla sua area riservata

2. Il medico accede alla pagina di visualizzazione dei pazienti
3. Il medico seleziona il paziente di cui vuole visualizzare la traccia
4. Il medico accede alla schermata di visualizzazione della traccia dell'utente

Postcondizioni: nessuna

1.7 Activity diagram

Per semplicità non verrà rappresentato il fatto di poter ripetere le operazioni più volte, ma si darà per scontato che l'utente possa tornare indietro e ripetere senza riavviare il software. Vengono quindi rappresentate soltanto le singole attività di operazione.

Si dà inoltre per scontato che per ogni attività (tranne la registrazione) l'utente sia già autenticato e che non ci siano errori di autenticazione.

1.7.1 Registrazione di un utente

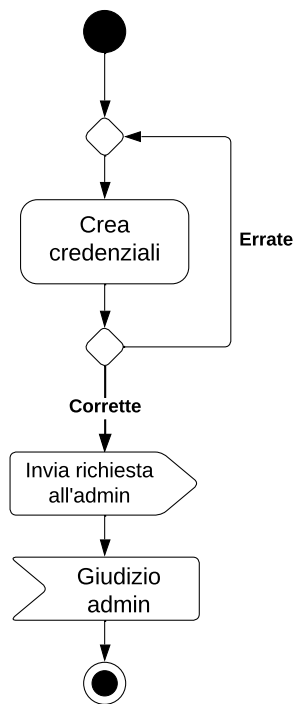


Figura 5: Activity diagram della registrazione di un utente

1.7.2 Attività del paziente

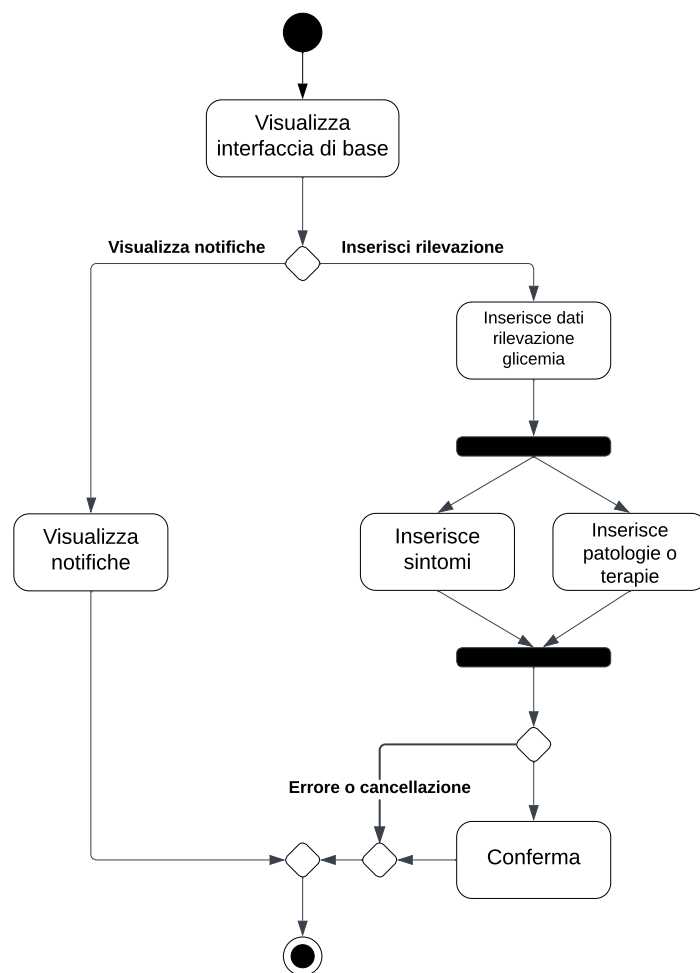


Figura 6: Activity diagram del paziente

1.7.3 Attività del medico

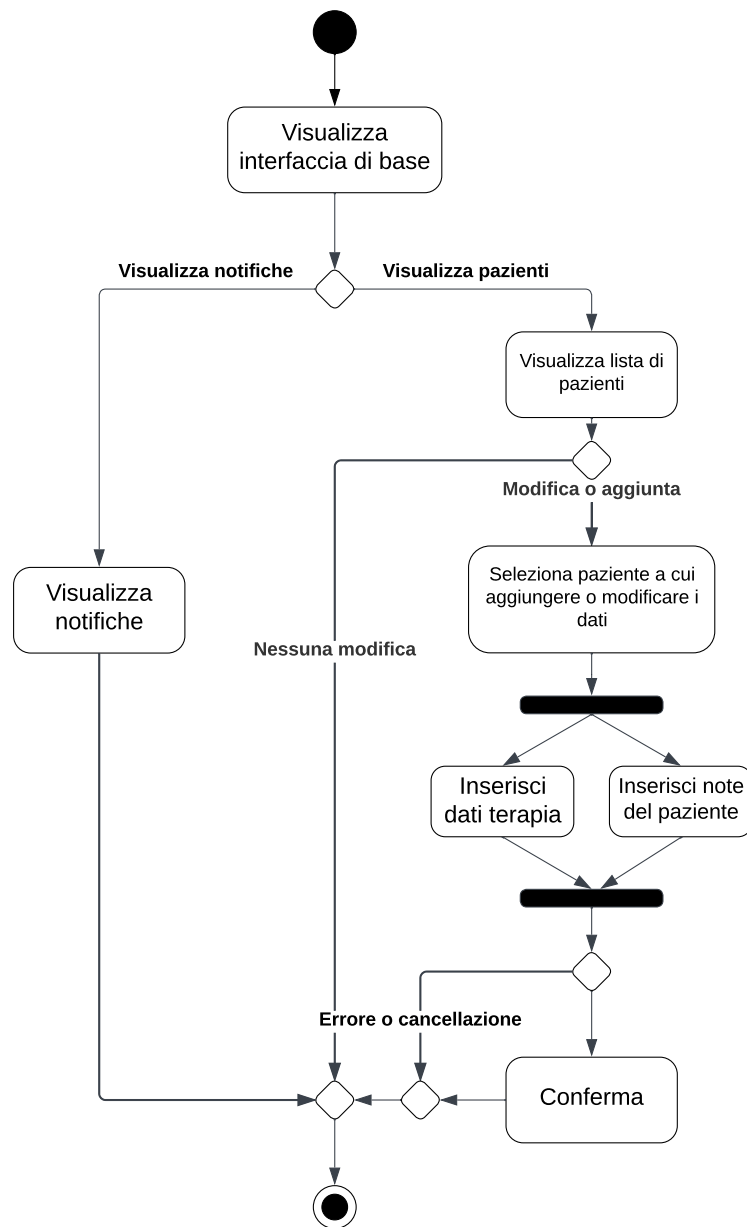


Figura 7: Activity diagram del medico

1.7.4 Attività dell'amministratore

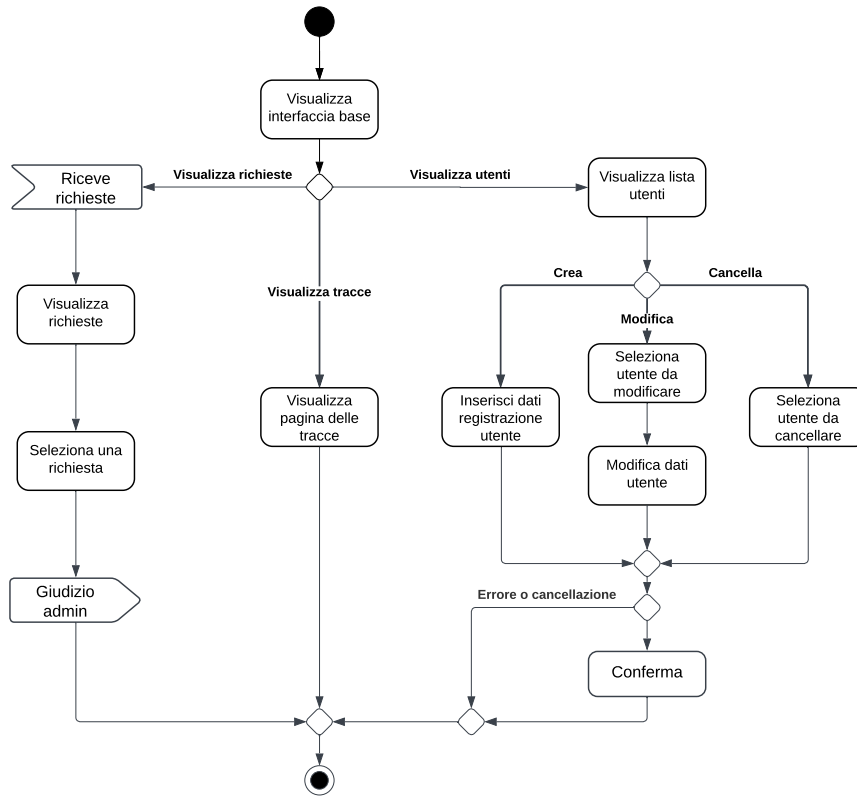


Figura 8: Activity diagram dell'amministratore

2 Sviluppo

2.1 Processo di sviluppo

Il processo di sviluppo del software è stato realizzato seguendo il modello *Agile* con metodologia *Scrum*. Il team è composto da due sviluppatori, che hanno lavorato in parallelo su diverse funzionalità del software. Lo Scrum ci ha permesso di organizzare il lavoro in sprint precisi, durante i quali abbiamo sviluppato le funzionalità richieste, testandole e integrandole nel software. Gli sprint venivano pianificati in base alle funzionalità da implementare e alle priorità stabilite all'inizio, facendo una scrematura delle funzionalità da implementare in base al tempo a disposizione. Alla fine di ogni sprint, abbiamo effettuato una revisione del lavoro svolto, testando le funzionalità implementate e correggendo eventuali bug. *Product Backlog* e *Sprint Backlog* non sono nient'altro che una lista di "to-do" task che bisogna affrontare per la scelta delle funzionalità da implementare e su come implementarle rispettivamente. Tuttavia non vi è un *Scrum Master*

in quanto il team è composto da due persone e non vi è la necessità di avere una figura che coordini il lavoro.

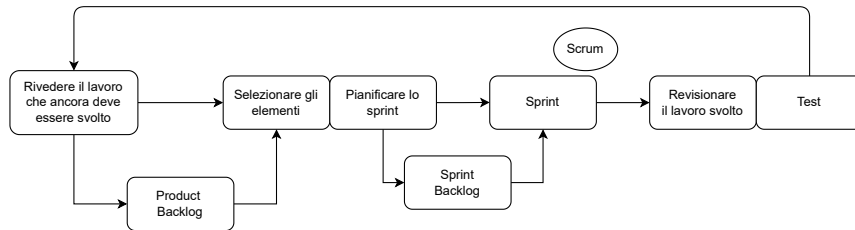


Figura 9: Diagramma del processo di sviluppo

Prima di cominciare il ciclo di Scrum però, è stata effettuata una fase di *analisi dei requisiti*, dove tutto il team di Scrum si è unito per analizzare le specifiche del progetto e poi sono stati definiti i casi d'uso (vedi figura 1). Man mano che venivano implementate le classi e le funzionalità, venivano costruiti gli UML per rappresentare le classi e le relazioni tra di esse. Per la gestione del codice sorgente, è stato utilizzato *Git* come sistema di versionamento, con un repository su *GitHub* per facilitare la collaborazione tra i membri del team. Abbiamo utilizzato anche *Github Planner* per tenere traccia delle attività da svolgere e dello stato di avanzamento del progetto.

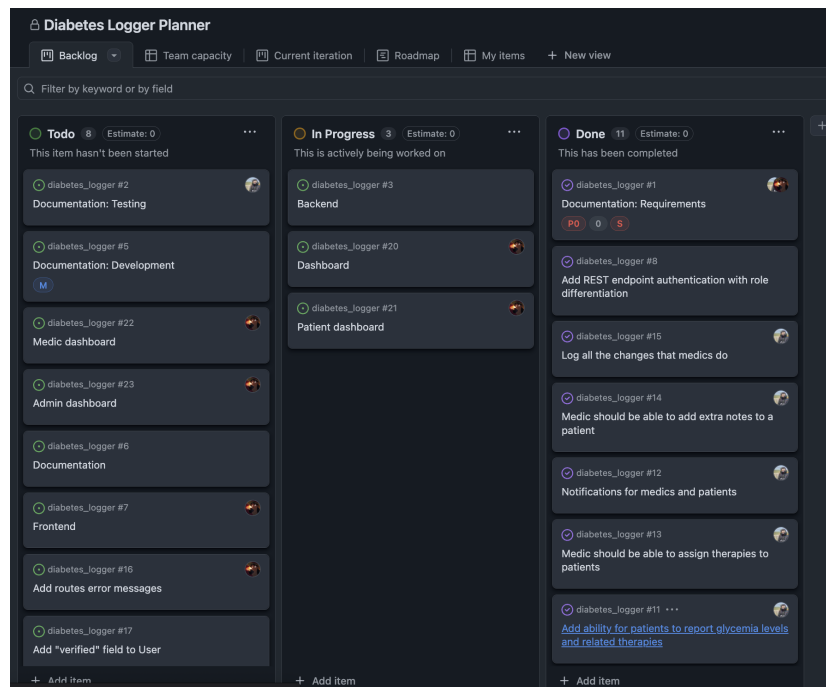


Figura 10: Github Planner del progetto

2.2 Progettazione e pattern architetturali usati

Il software è stato progettato seguendo il pattern architetturale *Model-View-Controller* (MVC), che separa la logica di business, la presentazione e la gestione degli eventi. Questo tipo di pattern ci ha permesso di mantenere il codice ben organizzato e facilmente manutenibile. Abbiamo scelto questo pattern poiché *Spring Boot* (framework di Java utilizzato per il backend) ha gli strumenti necessari per implementare questo pattern in modo semplice e veloce:

- **Model:** rappresenta la logica di business e i dati del software. In questo caso, il model è composto da classi che rappresentano lo user, i pazienti, i medici, le rilevazioni, le notifiche e il change log.
- **View:** rappresenta la parte di presentazione del software. In questo caso, la view è composta da pagine costruite con SvelteKit, un framework per fare siti web.
- **Controller:** rappresenta la parte di gestione degli eventi e della logica di business. Risponde alle chiamate e alle sollecitazioni della View e gestisce le richieste degli utenti.

Tramite richieste HTTP, il controller comunica con il model e la view. Quindi il controller riceve la richiesta dalla view e poi interagisce con il model per ottenere i dati necessari oppure modificarli in base alla richiesta dell'utente. Nelle prossime sezioni verrà analizzato in dettaglio il modello Model poiché contiene le classi che rappresentano gli attori del sistema e le loro interazioni.

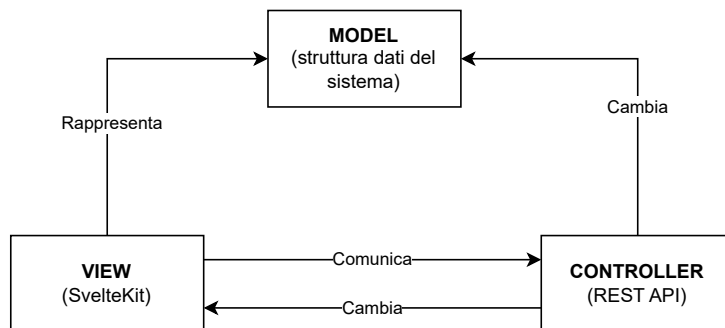


Figura 11: Architettura MVC del software