PROGETTO

AnAvis

by “**SuperDiciotto**”

Leonardo Mogianesi, Lorenzo Brancaleoni, Brian Bernardini

VISION DI PROGETTO

E' stato realizzato un sistema in cui è permesso ai donatori Avis di prenotare le donazioni on-line.

Il sistema proposto renderebbe più veloce la pratica di prenotazione e risulterebbe più comodo per i donatori scegliere fra tutte le date e gli orari disponibili. Per quanto riguarda il modello per la prenotazione a una donazione, esso sarà disponibile per la compilazione on-line da parte del donatore al momento della prenotazione.

Il sistema sarà accessibile anche ai dottori che riceveranno i dati sulle analisi del sangue e potranno aggiungere un commento che poi il donatore potrà visionare per email.
Inoltre il sistema permetterà ai dottori di inviare una richiesta di emergenza chiamata “donazione d'emergenza” .

Le sedi Avis interagiranno con il sistema solo per inserire gli orari e le date disponibili per le donazioni, eventualmente rimuovere delle date e controllare la lista delle prenotazioni.
Allo stesso tempo le sedi Avis avranno a disposizione una “dashboard” che permetterà di visualizzare l’andamento delle donazioni e dati vari al fine di evidenziare potenziali criticità della sede.

METODO DI LAVORO

Per la realizzazione del progetto si è cercato di seguire il modello del processo unificato (UP) e lavorare tramite iterazioni.
I diagrammi che sono stati realizzati (Use Case Diagram, Class Diagram e Sequence Diagram) chiariscono in maniera efficiente come il sistema lavora e cosa è in grado di fare.

ITERAZIONE 1

Nella prima iterazione (Avvio) si è cercato di stabilire gli attori che possono interagire con il sistema e gli obbiettivi da raggiungere. Sono stati realizzati i principali casi d'uso che avrebbero composto il sistema e aggiunta la descrizione comprendendo le info, le pre e post condizioni e il flow of events.
Infine è stato redatto un class diagram molto semplice con le entity del sistema e le relazioni tra di loro (basandoci solo sullo use case diagram) per avere le idee più chiare su come procedere.

ITERAZIONE 2

Nella seconda iterazione (Elaborazione) sono state apportate modifiche e migliorie ai casi d'uso realizzati nella precedente iterazione descrivendo in modo più chiaro possibile cosa il sistema è in grado di fare. Sono stati aggiunti ulteriori casi d'uso in quanto mancavano delle funzionalità ritenute importanti (ad esempio il logout) e sono stati gestiti meglio gli attori del sistema in modo che non venissero create ambiguità con i casi d'uso.
Inoltre è stato ridefinito da zero il class diagram con le varie entity presenti nel sistema e spiegato in modo dettagliato come le varie classi del back-end lavorano e sono in relazione tra loro basandoci sul codice prodotto.
Contemporaneamente sono stati sviluppati dei casi d'uso (ritenuti importanti per il funzionamento del sistema) tramite dei sequence diagram molto dettagliati enfatizzando la sequenza temporale degli scambi di messaggi tra diverse linee di vita.
Quindi alla fine di questa iterazione è stato redatto l'use case diagram in modo definitivo, il package delle classi che compongono il back-end e dei sequence diagram su alcuni casi d'uso.

ITERAZIONE 3

Nella terza iterazione (Costruzione) il focus è stato il miglioramento di alcuni aspetti del sistema. Il principale è stato il login in quanto si era reso necessario strutturarlo diversamente da come pensato in precedenza. Inoltre è stata definita un'altra funzionalità importante, quella di mandare una mail con il protocollo smtp che era prevista in alcuni casi d'uso (Controllo delle analisi, donazione d'emergenza).
Sono stati realizzati i componenti del front-end che devono interagire con il back-end. Quindi nel class diagram è stata aggiunto la parte del front-end specificando le varie componenti e le loro relazioni e aggiornato qualche componente della parte del back-end.
Procedendo con il lavoro e la scrittura del codice, sono stati realizzati i sequence diagram dei casi d'uso rimanenti.
Alla fine di questa iterazione la parte del back-end e del front-end sono state collegate tra loro così da avere un prodotto quasi finito. Ovviamente è stato realizzato il class diagram definitivo, comprendendo un package per il front-end, basandoci sul codice scritto. Per quanto riguarda i sequence diagram, oltre a completare quelli rimanenti è stato realizzato più nel dettaglio il sequence diagram del login più chiaro e dettagliato e aggiornati i sequence diagram dei casi d'uso che prevedevano l'integrazione della funzionalità della mail, così da evidenziare come le linee di vita si scambiano i messaggi

ITERAZIONE 4

Nella quarta e ultima iterazione (Transizione/Rilascio) l'obiettivo principale è stato sistemare dei problemi sul codice e organizzarlo in maniera più chiara e precisa.
Completata questa iterazione, il progetto è stato ultimato e le varie funzionalità pensate nelle iterazioni precedenti sono implementate completamente e funzionano.

AVVIARE IL PROGETTO

All'interno della repository sono presenti tre cartelle principali:

- Avis: in cui sono presenti tutti i file riguardanti la parte front-end(Angular8)
- idsdatabase: in cui è presente la parte back-end(Spring-boot)
- VisualParadigmIDS: in cui sono presenti i file di Visual Paradigm(16.0);

Il file dbprova.sql che dovrà essere importato su phpMyAdmin, inoltre dovrà creare un nuovo utente per avere accesso ai privilegi del database impostando le seguenti credenziali:

- **user: ids**
- **password: ids**

Per avviare il progetto sono richieste i seguenti componenti:

Front-end:

1. Angular 8
2. Node.js
3. Node Modules
4. Bootstrap
5. ngx-bootstrap

Back-end:

1. Xampp
2. Apache
3. Tomcat
4. Maven
5. Spring-boot
6. Hibernate

Gli step da seguire per avviare il progetto sono i seguenti:
Per il front-end:

1. Su Visual Studio, caricato il progetto, aprire il terminale e digitare "cd Avis" e in seguito digitare "ng serve –open"
2. per scaricare i node modules: "npm install"

Per il back-end:

1. Avviare Xampp e premere start su Apache e MySQL
2. Caricare su MySQL il file .sql messo su GitHub(come precedentemente descritto)
3. Su Visual Studio scaricare l'estensione Spring-boot Dashboard, dopo averlo scaricato premere “start” nel menù a tendina che si creerà e attendere il caricamento.

ACCOUNT ESISTENTI:
Dottore:
username: dottore
password: dottore

Secondo dottore:
username: secondodottore
password: secondodottore

Avis:
username: avis
password: avisavis

User 1:
username: lorenzo
password: lorenzo

User 2:
username: brian
password: brianbernardini

User 3:
username: leonardo
password: leonardo

User 4:
username: professore
password: professore

User 5:
username: nicola
password: lattanzi