



Laboratorio di Fullstack

Cecilia Baccolo, Francesco D'Alessandro, Samira Boutabt, Mohamed Osman.



Obbiettivi

- 1) Creazione del **modulo** e del suo **report** per un registro elettronico scolastico
- 2) **Fase analitica:** target, personas, benchmarking, alberatura sito
- 3) **Fase operativa:** creazione wireframe e mockup desktop e mobile
- 4) Creazione del **database**
- 5) Creazione dell'**ambiente di lavoro** su Eclipse e **implementato le logiche**
- 6) Inserimento dell'applicativo all'interno di **Docker**



Piano di lavoro

Linguaggi di programmazione:

- HTML, CSS
- MySQL
- Java
- JQuery con AJAX

Tecnologie Storage usate:

- Xampp
- PHPMyAdmin

Framework Utilizzati:

- Bulma

Creazione del Wireframe:

- Figma

Repository:

- Github



Fase analitica

Target

Il target è diviso in **preside/personale scolastico**(dai **50** ai **55** anni) e **alunni** (dai **15** ai **19**), nella quale sono presenti sia uomini che donne. Abbiamo suddiviso in due categorie perché nel registro elettronico accedono persone con diverse età.

Personas

Mario Bianchi



Dario Flanco



Chiara Giuliani





Personas 1

Mario Bianchi (56 anni)

Provenienza: Milano, Italia

Stato: Sposato

Ruolo: Preside dell'istituto

Istituto: IIS Manzoni (Milano)

Cosa si aspetta dal registro? Facile da utilizzare e soprattutto leggibile.





Personas 2

Dario Flanco (18 anni)

Provenienza: Milano, Italia

Stato: Nubile

Ruolo: Studente

Istituto: IIS Manzoni (Milano)

Cosa si aspetta dal registro? Che sia UserFriendly.





Personas 3

Chiara Giuliani (16 anni)

Provenienza: Milano, Italia

Stato: Fidanzata

Ruolo: Studentessa

Istituto: IIS Manzoni (Milano)

Cosa si aspetta dal registro? Velocità e usabilità.



Benchmarking

Abbiamo analizzato 3 competitor:





Velocità

Leggibilità

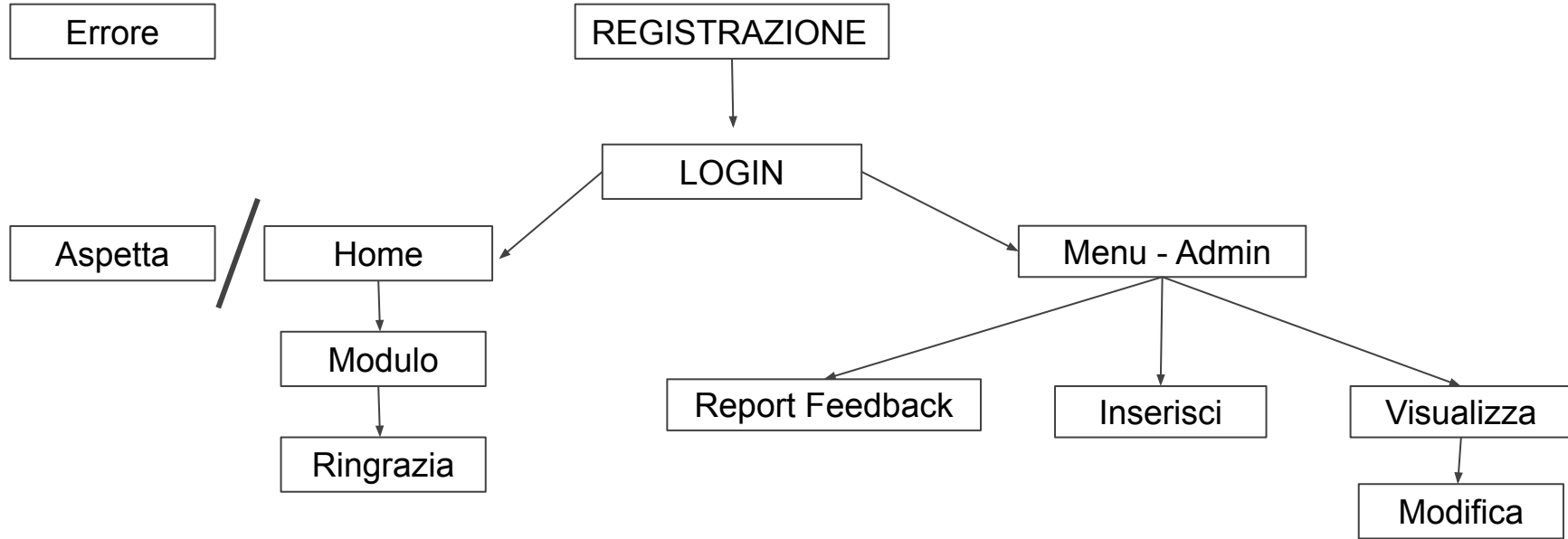
Comprensibilità

Estetica

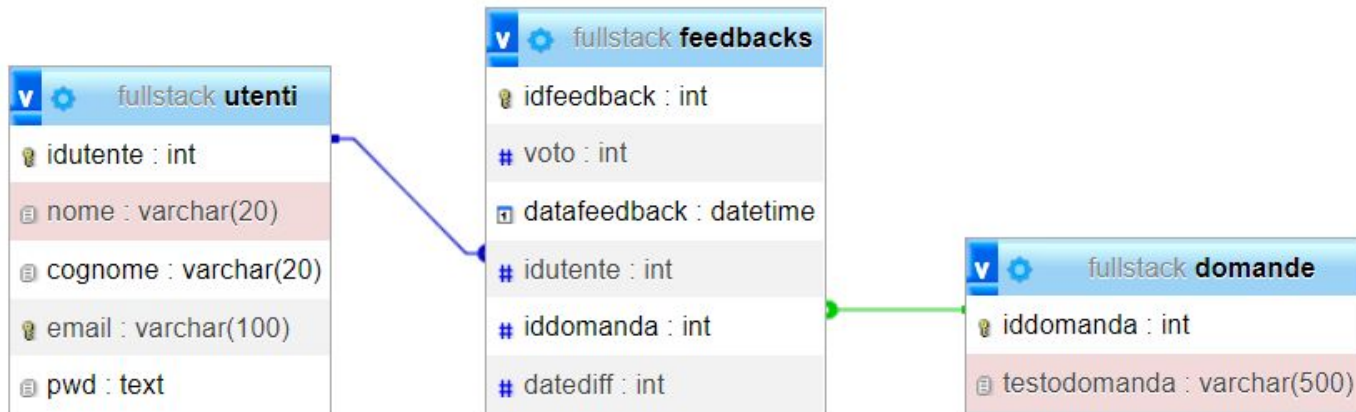
Mobile



Alberatura del sito



Creazione del database





Creazione dell'ambiente di lavoro

Nella creazione del progetto abbiamo inserito delle dipendenze:

- **Thymeleaf, SpringWeb**
- **Spring Data JPA**
- **MYSQL Driver**
- **Hibernate**

Una volta importato il progetto su Eclipse, abbiamo creato nei diversi Package i **Model** (Utente, Domande, Feedbacks), i **Dao** (Utente, Domande e Feedbacks) e il **Controller**.



Creazione delle logiche lato studente

1. Lo studente ha una e-mail che viene registrata dalla scuola e i suoi dati sono già inseriti
2. Lo studente può fare una volta ogni 7 giorni un modulo e se riprova a fare il modulo, saprà tra quanto potrà rifarlo
3. Il modulo di 4 domande viene generato randomicamente sulla base di 8 domande inserite nel db ed è anonimo (vengono visualizzate le percentuali per ogni risposta e la media)



Creazione delle logiche lato admin

1. L'admin accede attraverso la e-mail e si ritrova nel **menù principale** dalla quale potrà svolgere diverse **azioni (modifica, visualizza report, aggiungi e rimuovi)**
2. Visualizzando il **report del modulo** si vedono le sue **statistiche** quindi le percentuali per singola risposta, quante risposte sono state date e la sua media
3. **Rimuovendo** uno studente, si rimuove tutto il suo **record** quindi anche il suo id tramite `removeById()`, invece modificandolo i suoi dati vengono sovrascritti su quelli già esistenti con `set_____()`;
4. Aggiungendo uno studente, si inseriscono nome, cognome, e-mail e la password che viene impostata in NULL con `save()`



Creazione delle logiche sicurezza

1. L'accesso all'applicazione è **limitato** soltanto alle persone che frequentano la scuola quindi il personale e gli studenti non possono registrare nuove e-mail, ma soltanto l'admin
2. L'accesso alle varie pagine è reso sicuro attraverso la creazione della **Session**:
 - se admin e studente non sono loggati, non possono accedere alle loro pagine
 - l'admin non può entrare nelle pagine dello studente e viceversa
3. Le password vengono **criptate**
4. Se lo studente dimentica la password, solo la segreteria può modificarla
5. Facendo il **log-out** si chiude la session