

## RELAZIONE PROGETTO

Il progetto consiste in un'applicazione Android sviluppata per un tecnico informatico. Il datore ha un contratto ad ore con il Comune ed ha bisogno di tenere traccia delle ore rimanenti dopo i vari interventi. Inoltre ha bisogno di tenere traccia degli interventi effettuati a privati senza contratti, per completezza si è deciso di tenere traccia anche di eventuali contratti ad ore con privati seppure siano sporadici.

L'applicazione consente di creare:

- 1) **Contratto per il Comune** con una **banca ore**, le **ore rimanenti** e il **valore monetario**. Al contratto sono associate le relative **schede intervento** per tenere traccia dei lavori effettuati.
- 2) **Scheda Comune** contiene un **numero intervento** incrementato automaticamente, un **id del contratto** a cui è associato, la **durata** dell'intervento, una **descrizione**, l'**ufficio** in cui è stato svolto l'intervento, **data e ora**.
- 3) **Schede privato** contengono gli stessi attributi di quelle per il Comune ma con alcune differenze: al posto dell'ufficio c'è il **cliente** a cui è stato svolto l'intervento, l'**id non è presente** in quanto non associato a nessun contratto. Il **numero intervento** viene incrementato indipendentemente dal cliente.
- 4) **Schede privato per contratto** è una scheda privato che contiene l'**id del contratto associato**.
- 5) **Contratto Privato** è un contratto con **banca ore**, **ore rimanenti**, **valore monetario**, il **cliente** con cui è stato siglato ed un **id** che sarà associato alle schede.

Nella schermata di Creazione di ogni scheda è possibile selezionare un'immagine da associare all'intervento.

Una volta create le schede l'app effettua uno screenshot in automatico e lo salva nella galleria con il nome "Mit\_<Tipo Scheda>\_<Numero Intervento>". Su richiesta del datore è stato scelto di fare lo screenshot come prova dell'avvenuto intervento, si è scelto di usare lo screenshot per una sua maggiore comodità.

L'applicazione è scritta in **Flutter**, come database è stato utilizzato **SQLite** per la sua compatibilità con Flutter e perché permette di effettuare le operazioni in locale.

E' stato implementato un pattern **MVC** dividendo così i vari file:

### MODEL:

- 1) **ContrattoComune.dart** rappresenta un contratto Comune
- 2) **ContrattoPrivato.dart** rappresenta un contratto privato
- 3) **SchedaComune.dart** rappresenta una scheda intervento per il Comune
- 4) **SchedaPrivato.dart** rappresenta una scheda intervento per un privato

## CONTROLLER:

- 1) **SchedeController.dart** Controller che si occupa della creazione delle istanze del model e usare il dbManager per salvarli sul database o leggere da esso
- 2) **dbManager.dart** Controller che si occupa di gestire il database

## VIEW:

- 1) **CreaContrattoComune.dart** prende i dati dall'utente per creare un contratto Comune
- 2) **CreaContrattoPrivato.dart** prende i dati dall'utente per creare un contratto privato
- 3) **CreaSchedaComune.dart** prende i dati dall'utente per creare una scheda Comune e poi effettua uno screenshot
- 4) **CreaSchedaPrivato.dart** prende i dati dall'utente per creare una scheda privato e poi effettua uno screenshot
- 5) **CreaSchedaPrivatoContratto.dart** prende i dati dall'utente per creare una scheda privato da associare ad un contratto privato infine effettua uno screenshot
- 6) **ErrorPage.dart** viene visualizzata in caso di errore, creata per poterla usare anche con Widget Stateless
- 7) **DrawerWidget.dart** Widget per inserire il drawer menu nelle varie pagine
- 8) **MainPage.dart** pagina iniziale, funge da pagina principale per la creazione delle schede
- 9) **MainPageContratti.dart** pagina principale per la creazione dei contratti

Ci sono un paio di differenze tra implementazione e la succitata descrizione degli oggetti causate dall'utilizzo di un database per il salvataggio dei dati:

Anche se ne viene usato uno solo per volta anche il contratto Comune ha un ID per rispettare la proprietà di unicità del database.

Per le schede privato non associate ad un contratto è comunque presente l'ID contratto per non creare un'altra tabella virtualmente identica tranne che per un valore. L'ID viene messo pari a 0 per segnalare l'assenza di contratto.