

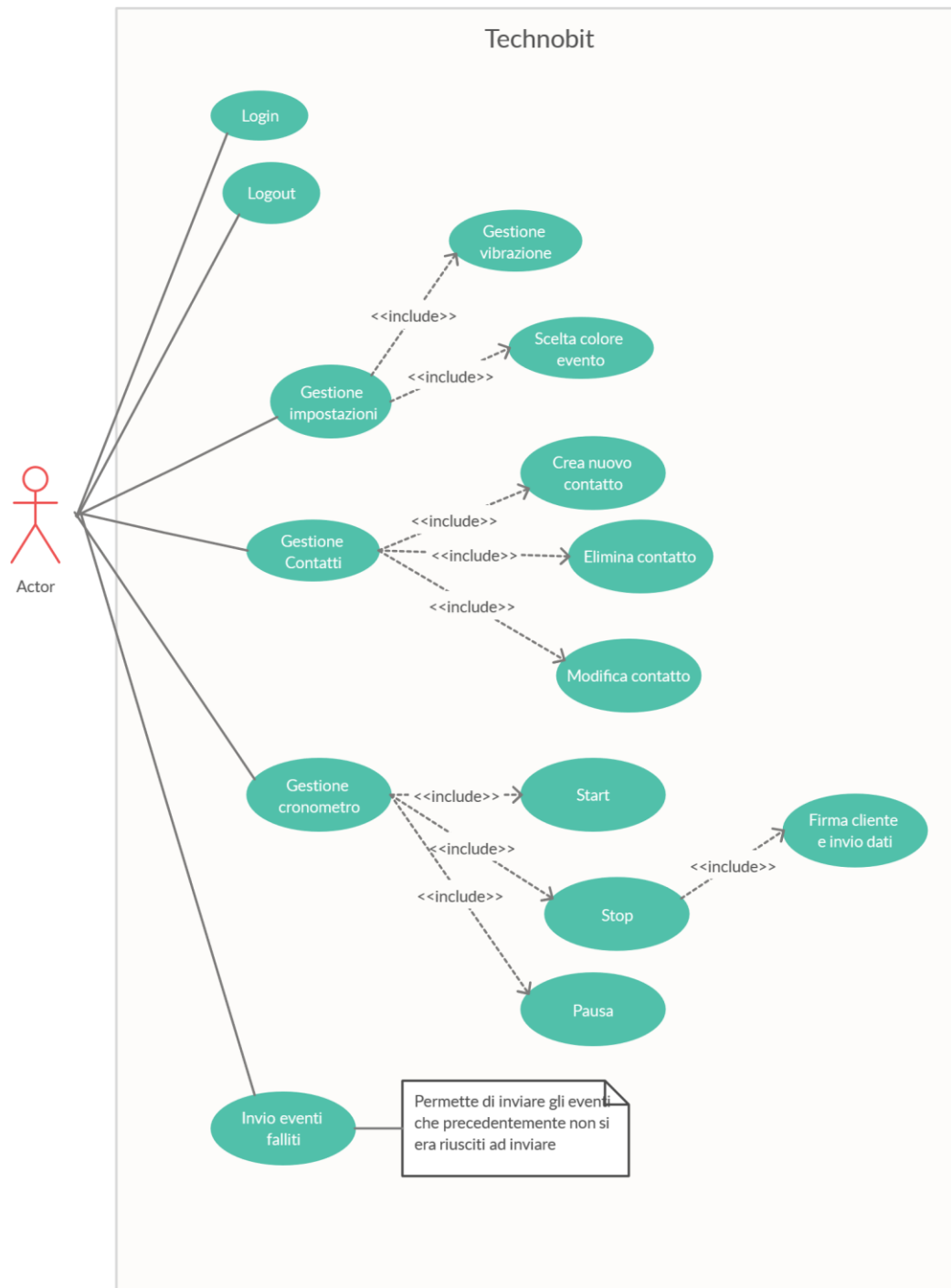
18/06/2020

DOCUMENTO DI DESIGN

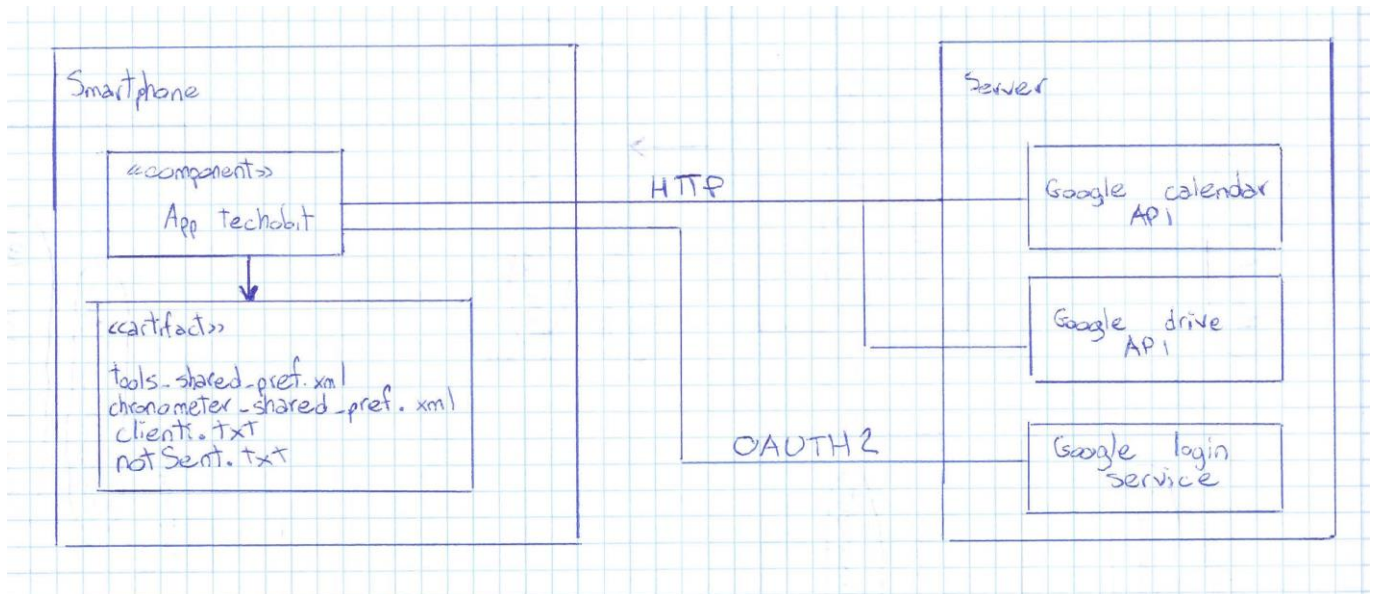
Funzionalità

L'obiettivo dell'applicazione è permettere all'utilizzatore di tenere traccia del tempo trascorso da un cliente per effettuare una riparazione e contestualmente alla fine dell'intervento poter inserire una breve descrizione del lavoro svolto e far firmare il cliente come forma di accettazione. Al termine del lavoro, una volta che il cliente ha firmato e accettato, l'applicazione provvederà ad inserire in Google Drive l'immagine della firma del cliente e in Google Calendar un evento con:

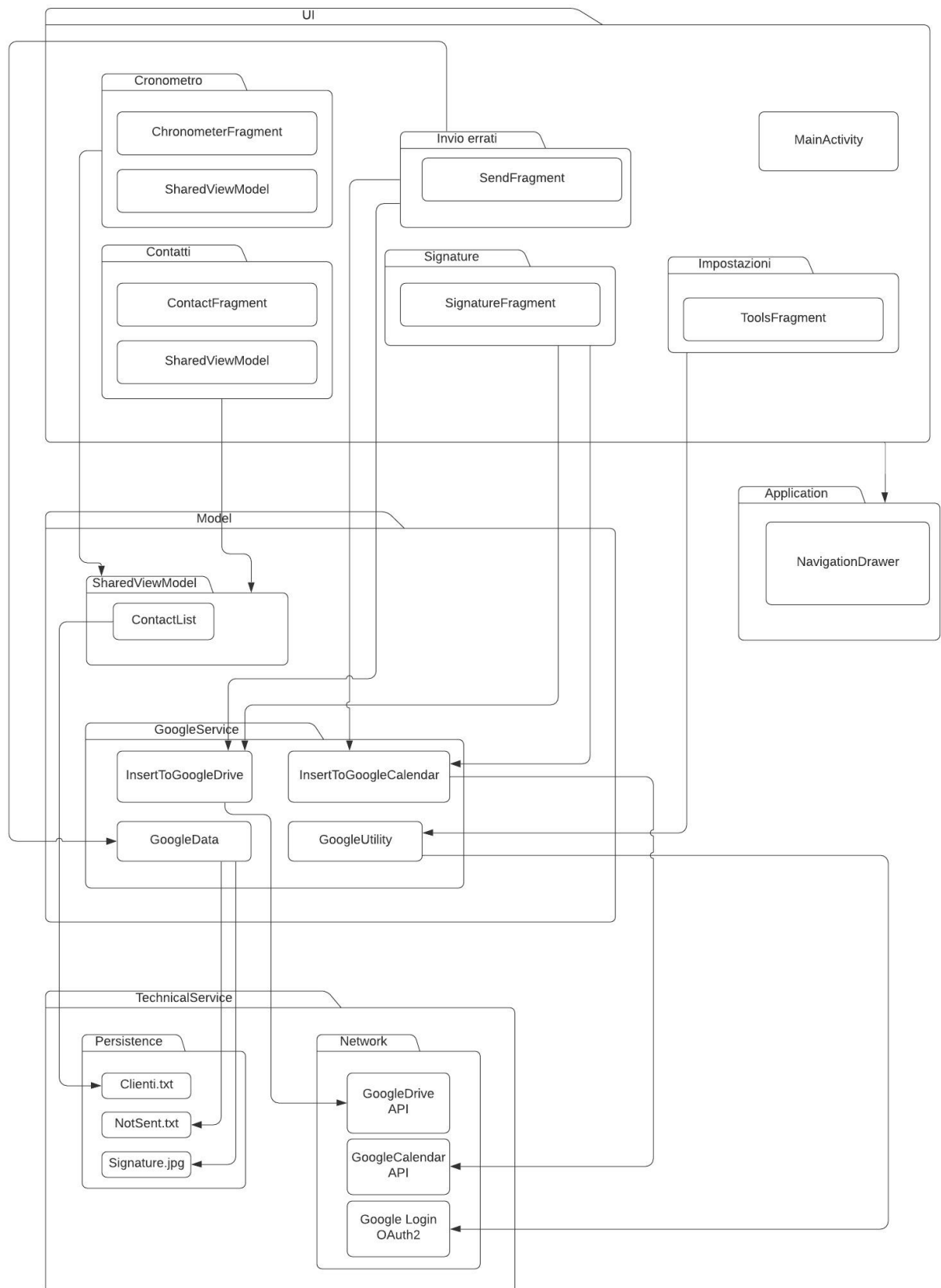
- orario di inizio e fine in base al periodo di lavoro
- la descrizione del lavoro fornita
- In allegato il link alla firma del cliente salvata precedentemente su Google Drive



Architettura



Design della soluzione



Informazioni sulla soluzione

Nel diagramma precedente sono rappresentate le componenti più significative ed importanti per l'applicazione e le interazioni tra esse. Nella componente "Persistence" in "Technical Service" sono state riportate anche NotSent.txt e Signature.jpg anche se questi file sono presenti soltanto in caso di errori di invio, ad esempio per mancanza di internet o per problemi dei servizi Google utilizzati. Questi dati vengono infatti salvati localmente finché non verranno inviati con successo e dunque automaticamente cancellati. Inoltre per quanto riguarda i file jpg delle signature dei clienti essi saranno tanti quanti gli eventi non inviati in quanto per ogni evento corrisponde una firma di un cliente. Per evitare problemi di duplicazione dei nomi ogni file jpg è nominato come NomeCliente_gg_mm_aa.jpeg.

API utilizzate

Per la realizzazione di questa applicazione sono state utilizzate le seguenti API Google:

- Google Drive API: Utilizzata per inserire l'immagine della firma del cliente su Google Drive dell'account selezionato e ottenere il webViewLink fornito da Google per lo stesso file
- Google Calendar API: Utilizzata per inserire un evento della durata dell'intervento effettuato, come allegato l'immagine della firma tramite webViewLink e come descrizione quella inserita dall'utente prima di confermare.
- Google Sign-In API: Per permettere all'utente dell'applicazione di loggarsi con il proprio account Google e far visualizzare e accettare i servizi necessari all'applicazione, cioè l'accesso ai servizi Drive e Calendar.

La scelta di caricare l'immagine su Google Drive e inserirla solo come "allegato" nell'evento del calendario è dovuta al fatto che Google non permette di caricare direttamente l'immagine su Calendar ma permette infatti di incorporare un'immagine già presente su Drive tramite link univoco. Un altro aspetto importante da notare è che per l'evento inserito su Calendar l'orario di inizio corrisponde esattamente al momento in cui è stato fatto partire il cronometro mentre l'orario di fine è ricalcolato sottraendo eventuali pause effettuate durante la riparazione. Ad esempio per una riparazione iniziata alle 9:00 e terminata alle 11:00 in cui è stata effettuata una pausa di venti minuti stoppando il cronometro l'orario di fine inserito per l'evento sarà 10.40. Questa è stata una scelta necessaria in modo da avere un controllo diretto sull'effettivo periodo lavorato per non addebitare al cliente ulteriori ore non effettuate. L'upload dei dati, sia su Calendar che su Drive, viene effettuato mediante dei Thread che lavorano in background questo perché le richieste richiedono l'utilizzo di internet e di una richiesta di tipo http. Inoltre non sapendo quanto tempo sia necessario per l'effettivo caricamento dei dati, utilizzando i Thread è possibile lasciare il controllo dell'applicazione all'utente in modo che mentre i dati vengono inviati possa effettuare altre operazioni. Ho scelto di implementare l'upload tramite Thread anziché tramite AsyncTask anche per rispettare le nuove direttive di Google che con Android 11 deprecherà gli AsyncTask.

Versione API Android

Per utilizzare questa applicazione è necessaria come versione minima la numero 21, quindi l'applicazione può essere eseguita sul 94.1% dei dispositivi. Mentre è stata correttamente testata su 2 emulatori e un dispositivo fisico sulla versione numero 29.