[DONE] Verificare che i giorni di lavoro aggiunti siano lavorativi. Se ci sono sabati, domeniche o festivi segnati, segnalarlo ed eventualmente aggiungerli come straordinari.

[DONE] Verificare che le ore siano 8; se sono di più aggiungerle come straordinari mentre se sono meno aggiungerle come permesso. Se sono sabato e domenica segnarle tutte come straordinari.

[DONE-DA TESTARE]Se per uno stesso giorno vengono segnate più attività verificare che le info siano congrue. Esempio in nello stesso giorno non ci possono essere info di smartworking e malattia.

COPPIE AMMESSE:

smartworking - permesso

trasferta - permesso

ufficio - permesso

NEXT-STEPS:

creare un Dialog/Modal per segnalare warnings e errori

CLIENT:

PAGINE:

* LoginPanel.js: pagina contenente il form di login; permette all'utente di loggarsi all'interno del portale e, eventualmente di modificare password e email.
* DashboardPanel.js : pagina che contiene l'intera dashboard; permette ai consulenti di aggiungere gli orari lavorativi; mostra sia il dettaglio delle ore lavorative sia un resoconto a seconda del tipo di ore segnate.
* SettingPanel.js: pagina per la gestione degli utenti, del database e degli orari dei consulenti. E' utilizzabile solo dal service manager. Permette di esportare il file excel con il rendiconto di tutti i consulenti e di convalidare i vari mesi in modo da non essere più modificabili. Mostra i consulenti che non hanno compilato gli orari per l'ultimo mese valido e permette di inviare mail reminder ai consulenti.

COMPONENTI:

AddWDButton.js: bottone che serve per aggiungere un nuovo record della tabella work\_day sia in modalità massiva che singolarmente.

PROPS:

* OnSingleAWDClick --> utilizzato per richiamare la funzione della pagina DashboardPanel.js per aggiungere un nuovo record work\_day.
* type --> s se il bottone inserisce un record alla volta; m se invece apre il pannello per l'inserimento massivo.

VARIABILI DI STATO:

* lastID --> utilizzato per creare un nuovo id per il nuovo record.

FUNZIONI:

* addSingleRowClicked --> funzione per l'aggiunta di un singolo record work\_day. Richiama tramite props OnSingleAWDClick(newRow) la funzione della pagina DashboardPanel.js wtAddNewRow(newRow).

DialogBox.js: dialog per mostrare eventuali messaggi di errore, warning o stato dei processi

PROPS: nessuna.

VARIABILI DI STATO:

showModal --> true per mostrare il modale, false per nasconderlo.

title --> stringa che verrà visualizzata come titolo/header del dialog.

body --> stringa che verrà visualizzata come corpo del testo del dialog.

FUNZIONI:

handleShow --> funzione per mostrare il dialog; come parametri titolo e testo del dialog

handleClose --> funzione per nascondere il dialog.

ExportWDButton.js [TODO]: pulsante che attiva l'esportazione del rapportino.

MonthComboBox.js: combobox contente i mesi per la quale sono stati registrati dei dati.

PROPS:

month --> ultimo mese in cui esistono dati.

usrID --> id dell'utente utilizzato per recuperare tutti i mesi in cui esistono dati

OnMonthChange --> metodo utilizzato dalla pagina DashboardPanel.js per la gestione degli altri componenti al cambio del mese selezionato

VARIABILI DI STATO:

selectedMonth --> variabile che salva il mese selezionato.

month --> variabile che storicizza i mesi in cui ci sono dei dati .

FUNZIONI:

useEffect --> metodo per caricare i dati al caricamento del componente.

setTimeName --> chiamata al server per settare la lingua del db (e quindi il nome dei mesi) in italiano.

fetchMonths --> chiamata al server per recuperare i mesi dell'utente tramite la variabile usrID

changeHandler --> evento scatenato al cambio del mese selezionato che a sua volta fa partire il metodo props OnMonthChange del DashboardPanel.js

SaveWDButton.js: tasto per il salvataggio delle aggiunte/modifiche dei vari record

PROPS:

OnSaveClick --> metodo utilizzato dalla pagina DashboardPanel.js per la gestione del salvataggio dei dati