Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Canobbio |
| Data | 17.10.2019 |

|  |
| --- |
| Lavori svolti |
| Durante le prime due ore di lezione ho aggiunto un nuovo *handler* nella classe API che verrà utilizzato per le operazioni sui dati relativi alle prenotazioni.  Come si può vedere nello screenshot il metodo controlla anche se l’utente ha effettivamente i permessi necessari per l’utilizzo di questa funzione (l’utente necessita i permessi per l’inserimento di prenotazioni).  Dopo il controllo dei permessi utilizzo il nuovo metodo *add* creato all’interno della classe *BookingModel*. Questo nuovo metodo non fa altro che prendere i dati ricevuti dalle API, validarli e, se tutto va a buon fine, inserire i dati all’interno del database. Questo è il codice:    Successivamente ho aiutato per una quindicina di minuti Mattia Toscanelli e Mattia Lazzaroni per delle parti di codice.  Prima della fine della lezione mi sono ricordato di un requisito: *ci deve sempre essere almeno un amministratore*. Dato che nella gestione utenti non avevo implementato questo controllo ho creato un metodo nella classe *UserModel* che conta tutti gli amministratori presenti all’interno del database. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Non ho riscontrato nessun problema |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Sono in anticipo in anticipo rispetto alla pianificazione iniziale. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Creare il form per aggiungere gli utenti e testare il sistema. |