

# Diario di lavoro

Luogo	Canobbio
Data	19.11.2019

## Lavori svolti

Durante la giornata di oggi ho iniziato a sviluppare il metodo di filtraggio dei posteggi. Ho svolto il filtraggio in base alla disponibilità.

Il risultato finale è il seguente:

Disponibilità	Data disponibilità	Numero di targa	Prenota
Mattina	02-12-2019	TI-234567	<button>Prenota</button>
Mattina	16-11-2019	TI-234234	<button>Prenota</button>
Proprietario	Disponibilità	Data disponibilità	Prenota

Per riuscire a fare il tutto ho aggiunto il nuovo input per la disponibilità alla pagina dei risultati. L'input è lo stesso della pagina di offerta con una modifica per aggiungere lo stato di selected al valore selezionato.

```
<div class="col-6">
  <p class="text-white font-weight-bold">Inserisci la disponibilità:</p>
  <form method="post">
    <select class="browser-default custom-select w-50" name="select_disp" onchange="filterDisp(this.value)" >
      <option value="Tutto il giorno" <?php echo ($selected == 'Tutto il giorno')?'selected':'?'>>Tutto il giorno</option>
      <option value="Mattina" <?php echo ($selected == 'Mattina')?'selected':'?'>>Mattina</option>
      <option value="Pomeriggio" <?php echo ($selected == 'Pomeriggio')?'selected':'?'>>Pomeriggio</option>
    </select>
  </form>
</div>
```

In seguito ho creato il metodo di filtraggio nel rispettivo Model.

```
/**
 * Funzione che filtra i posteggi disponibili in base alla disponibilità scelta dall'utente.
 *
 * @param $disp Disponibilità selezionata nel filtro.
 */
public static function filterByDisp($disp)
{
    self::$statement = Database::get()->prepare( statement: "select * from posteggio where data_disp is not null and disponibilita = :disp");
    self::$statement->bindParam( parameter: ':disp', &variable: $disp, data_type: \PDO::PARAM_STR);
    self::$statement->execute();
    self::$parcheggi = self::$statement->fetchAll( fetch_style: \PDO::FETCH_ASSOC);

    foreach(self::$parcheggi as &$row)
    {
        if(is_null($row['n_targa']))
        {
            $row['n_targa'] = "";
        }
        $inputDate = date_create($row['data_disp']);
        $inputDateFormat = date_format($inputDate, format: 'd-m-Y');
        $row['data_disp'] = $inputDateFormat;
    }
}
```

In questo metodo faccio la stessa select di base che carica tutti i posteggi disponibili, ma gli aggiungo anche il valore disponibilita da controllare.

Successivamente ho modificato il codice presente nel controller per fare in modo di visualizzare il cambiamento dei dati una volta filtrati.

```
/**
 * Funzione che richiama il metodo per ricevere i posteggi offerti e poi reindirizza verso la pagina che li mostra.
 */
public function index()
{
    $selected = null;
    if(isset($_GET['filter_disp']))
    {
        $disp = Validator::testInput($_GET['filter_disp']);
        RicercaModel::filterByDisp($disp);
        $selected = $disp;
    }
    else {
        RicercaModel::getParcheggiDisponibili();
    }
    ViewLoader::load( template: 'ricerca/index', array('parcheggi'=>RicercaModel::$parcheggi, 'selected' => $selected));
}
```

Tutto ciò per funzionare ha bisogno della seguente funzione javascript che prende il valore selezionato nella select e lo invia tramite GET al controller.

```
function filterDisp(disp) {
    window.location.href = "/ricerca/index/?filter_disp=" + disp;
}
```

**Problemi riscontrati e soluzioni adottate**

-

**Punto della situazione rispetto alla pianificazione**

In linea.

**Programma di massima per la prossima giornata di lavoro**

Finire il filtraggio in base a un range di date e continuare i lavori assegnatimi dal docente.