**TIME TRACKING – REFACTORING STATUS**

**DONE SO FAR**

* **DB:**
  + **Dynamic DB migrations** system across environments using Flyway tool.
  + **DB Schema refactoring** to SQL standard snake\_case column names and tables.
  + **DB Schema Foreign Keys** review & setting missing ones.
* **Spring Framework:**
  + Spring Framework updated version to latest minor.
* **Back End:**
  + **Utils, Constants, Enums** cleaning & refactoring into proper re-organized classes.
  + **Controllers & Services** general architecture refactoring (class & method coupling, encapsulation into private, layer responsibilities, etc.).
  + **Entities,** **DTOs (Business Objects) and local variables/parameters** refactoring to Java standard camelCase field names.
  + **Model** refactoring to make use of JPA Hibernate ORM features such as relational ManyToOne/OneToMany objects, making use of Foreign Keys relationships.
  + **Data Access Layer (DAO/Repository) layer** refactoring to make use of JPA Repository standard and use of modern method-derivation-based and annotated queries.
  + **Business Logic (Services) layer** refactoring to make use of ModelMapper and centralized custom casts for complex data objects (DTOs/BO).
  + **Administration page Reporting features** general refactoring on reports content generation and global shared and inconsistent data structures.
  + **Mailing service** methods refactoring.
  + **Session Attributes** cleaned from spread usage to more localized.
  + **Potential bug fixing** due to data inconsistence or code complexity.
* **Front End:**
  + **Unused static resources (JS/CSS)** third-party librariesremoved.
  + **Common static resources** refactored into global libraries file.
  + **JSP fragmentation** for common code blocks.
  + **JSP wrapping** for rollable code blocks into simpler iterable ones.
  + **JS common functions** unification for weekly views (Record Time & Approve per Week) and monthly views (Approve per Month & Monthly Summary) respectively.
  + **JS Global Utils** functions refactoring & cleaning.
  + **Refresh data** widespread functions unification into common pages or into a global utils function.

**TO DO (ESTIMATION & PRIORIZATION)**

* **Performance (High)**
  + **Optimization of Object Relational Mapping** (ORM) bidirectional relationship objects and cascade on update. (**Critical**) – 1 *week*
    - Usage of JPA ORM architecture to reduce number of queries and crossed calls between services & model entities. – 3 *days*
    - Ensure low coupling and high cohesion principle between Entities & DTOs. – *2 days*
* **Global refactoring (Medium)**
  + Review & refactor all back-end layers (controllers, services) after new architecture implementation for reducing code complexity. – *1 week*
  + Historic object refactoring in favor of DB synchronous calls – *1 week*
* **Technical Debt & Maintenance (Low)**
  + DB *status* column types unification – *2 days*
  + JS (inline) refactoring & cleaning – *1 week*
  + CSS files refactoring & cleaning. – *3 days*
  + Variables casing normalization on each programming language at back end and front end. – *2 days*