

# Services documentation for Team utilisation Monitor v1

Mamphasa Agape

June 6, 2022

## 1 Introduction

This document will detail all of the Backend business logic required for the Team utilisation Monitor. Please take note that the documentation is subject to change as more requirements are added and more improvements are made. The core services has been implemented using the CQRS design pattern. This separates services into Queries and Commands. A short description will be given on each functionality and the name of the Command or Query and it's respective handler.

## 2 Services

I've decided to classify the functions into 4 classes:

- Queries: Used to get data from the database
- Commands: Used to mutataate the database
- Authentication: Used to verify the user's credentials
- Functions: Provide some general functionality e.g calculateUtilisation()

## 3 Queries

Queries are generally used to get data from the database

### **3.1 getAllPersonsQuery**

This function returns all the people in the database

### **3.2 getAllPersonsQuery: UserPerson[]**

This function returns all Objects of all people in the database

### **3.3 getOnePersonQuery(email:string): UserPerson**

This function returns one person Object based on their email returns null if person doesn't exist.

### **3.4 getCompany(companyName:string): UserCompany**

This function returns the company's Object based on name of company. Returns null if company was not found.

### **3.5 getUserId(email:string):Int**

This function returns the user's id based on their email. Returns -1 if user doesn't exist.

### **3.6 getCompanyId(companyName:string): Int**

This function returns the company's id based on it's name Returns -1 if company doesn't exist.

### **3.7 getAdminsForCompany(companyName:string):Admin[]**

This function returns the admins for the company. Returns null if admins don't exist.

### **3.8 getEmployees(companyName:string):Person[]**

This function returns an array of employees that work in the company. Returns an empty array if a company doesn't have employees.

### **3.9 getPendingRequests(companyName:string):Person[]**

This functions returns an array of people who hold the company inviteCode and who's approved is still false. Returns an empty array if no pending requests.

### **3.10 getNumberOfTeams(companyName:string): Int**

This function returns the number of teams for a company

### **3.11 getTeamMembers(teamName:string):Person[]**

This function returns all the team members of a team.

## **4 Commands**

Commands are used to mutataate values in the database. Or change the state of the database.

### **4.1 createUserCommand(name:string,surname:string,email:string,inviteLink:string)**

This function is used to create a User on the database system.

### **4.2 createAdminCommand(name:string,surname:string,email:string,companyName:string)**

This function is used to create an Admin on the database.

### **4.3 assignToProject(id:user)**

This function is used to assign a user to a project using their ID.

### **4.4 assignToTeam(id:user)**

This function is used to assign a user to a team.

#### **4.5 createProject(projectName:string,ownerId:Int,manhours:Int)**

This function is used to create a new project and add it to the company's portfolio of projects.

#### **4.6 createTeam(teamName:string,companyId:Int,projectId:Int)**

This function is used to create a new team and add it to the company's set of teams.

#### **4.7 editProfile()**

We will hav to implement this later on!

#### **4.8 addSkills()**

We will hav to implement this later on!

### **5 Authentication**

Authentication functions are used to authenticate some field against the database

#### **5.1 login(email:string,name:password):UserPerson**

This function is used to login by taking the email and password of the admin/user.

#### **5.2 VerifyCode(inviteCode:string):boolean**

This function is used to verify an invitation code that was sent.

### **6 Functions**

Functions are used to provide standard business logic that might be needed in the front-end or Backend.

## **6.1 CalculateUtilisation(ID:Int):Int**

This function calculates the utilisation of an employee and takes in the employee's ID

## **6.2 generateInviteCode(companyName:string):string**

This function creates a unique invite code that'll be used by the user to create an account.