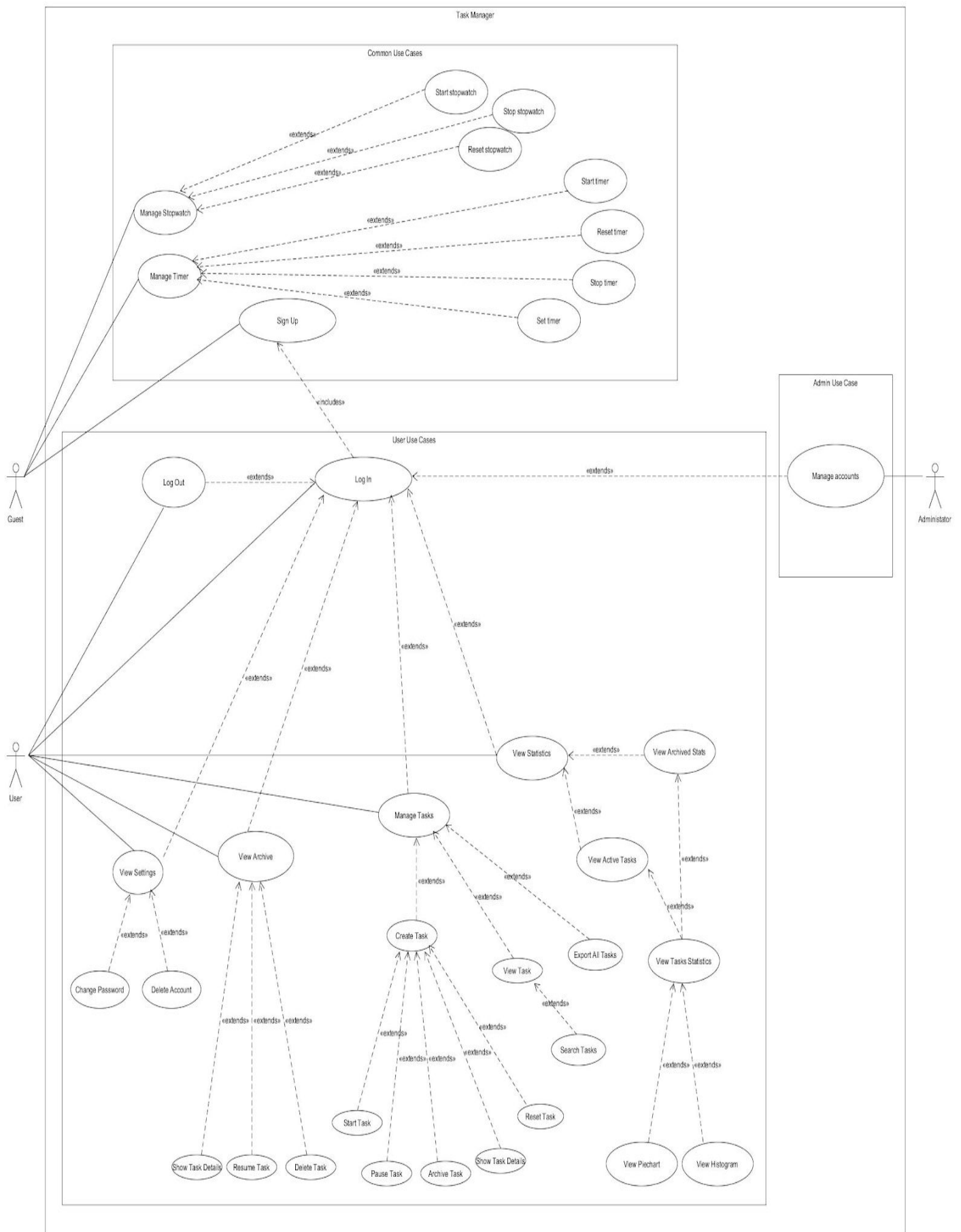# Task Manager

The goal of the project is to provide easy and convenient way to track time spending. System allows to create tasks, timers, stopwatches and see how much time was spent on different activities.

**User roles:** Guest, User, Admin.

| Role | Description |
|---|---|
| Guest | User that isn't logged in. Can manage stopwatches and timers, sing up and log in. |
| User | Logged in user. Can manage stopwatches and timers, create own tasks, edit and remove them, search tasks, view active/archive tasks, view statistics. Also can get info about his/her account, change password or delete his/her account. |
| Admin | Can manage stopwatches and timers, create own tasks, edit and remove them, search tasks, view active/archive tasks, view statistics, remove users, add admin role and remove it. |

| № | Action name | Role | Description | Function |
|---|---|---|---|---|
| 1 | SignUp | Guest | Guest can sign up in system | |
| 2 | LogIn | User | User can log in system | |
| 3 | LogOut | User | User can log out system | |
| 4 | CreateTask | User | User can create task | |
| 5 | StartTask | User | User can start task | |
| 6 | PauseTask | User | User can pause task | |
| 7 | ResetTask | User | User can reset task | |
| 8 | ArchiveTask | User | User can archive task | |
| 9 | AddAdminPrivilege | Admin | Admin can provide "admin privilege" for other users | |
| 10 | RemoveAdminPrivilege | Admin | Admin can remove "admin privilege" from user to whom he has granted admin rights | |

# Use case diagram

# Class diagram

**«Task»**
Package::DataAccess.Entities

+Id: int
+Name: string
+Description: string
+WatchType: WachType
+IsActive: bool
+IsRunning: bool
+UserId: int

**«User»**
Package::DataAccess.Entities

+Id: int
+Email: string
+FirstName: string
+LastName: string
+Password: string
+Tasks: ICollection<Task>
+UserRoles: ICollection<UserRole>

**«DateInfo»**
Package::DataAccess.Entities

+ElapsedTime: DataTime
+LastStartTime: DataTime