TRACKITIME

1. WHAT?

How many hours did you use with your practice work? Or with your origami project? Trackitime is an application to solve this problem. With Trackitime one can easily create a new project, and add time periods spent with the project. Later some statistics can be seen of the project: what have been done, how much time have been spent altogether and so on.

2. HOW?

Trackitime is used with an easily-used web application, which is also supportive for mobile browsers. The users are identified with a logging system and personal accounts.

3. WITH WHAT?

The language Trackitime is build with is JavaScript with Node.js. Libraries used are Express.js (with all its dependencies) for creating and running the server and routing the requests. The user system is implemented in backend with Passport.js. All data is saved into a PostgreSQL database.

4. WHERE?

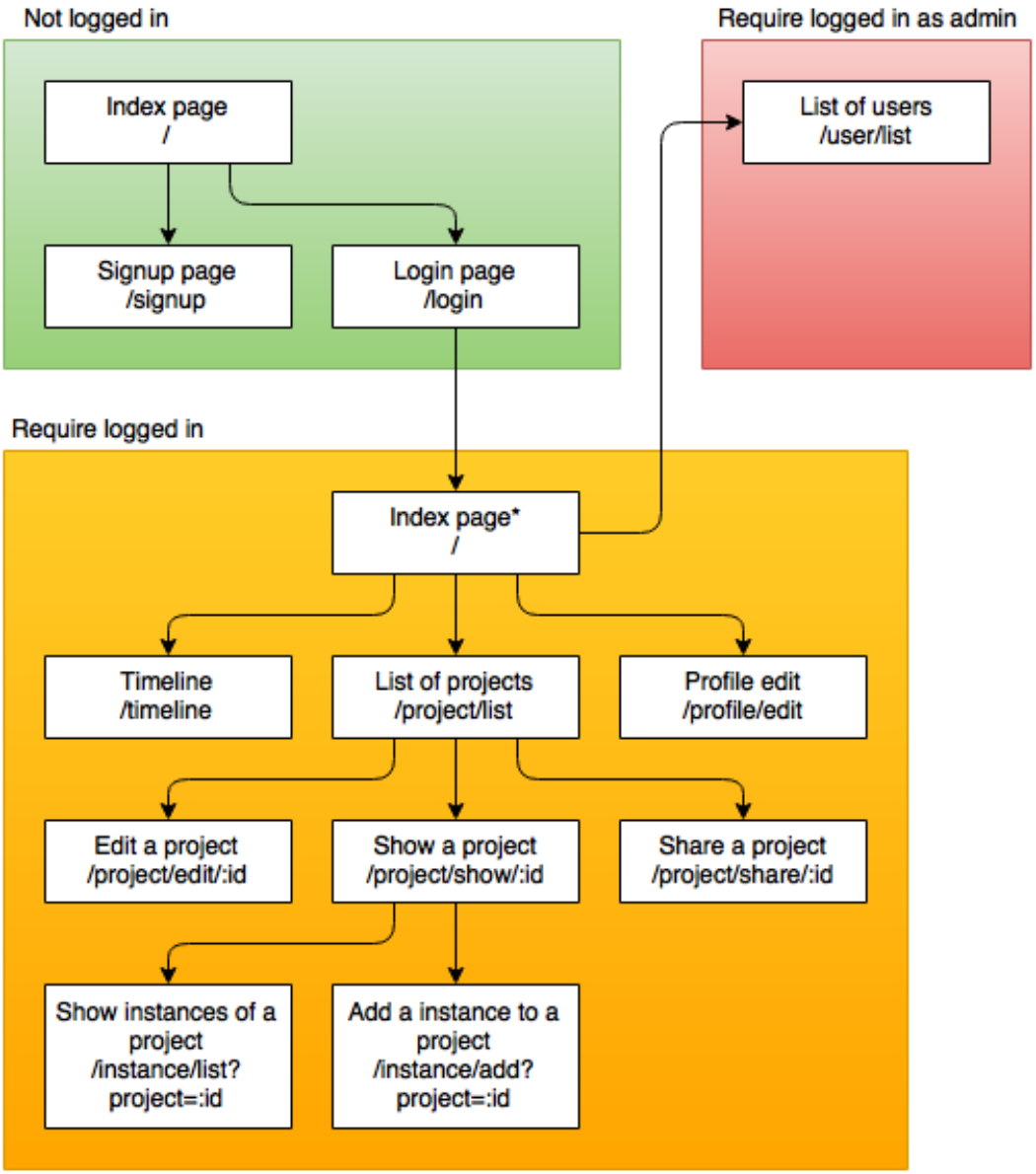
The code of Trackitime is stored in Git with a repository of the same name. The address of the repository is https://github.com/Aapzu/trackitime. At this moment the app is running in Heroku, and it can be found from the address https://trackitime.herokuapp.com/.

STRUCTURE OF THE APPLICATION

1. FILES AND FOLDER

The application follows the MVC model. Models are found from /models, views from /views and controllers from /routes. Configuration files/scripts are in folder /config, and other backend scripts in /app. All front-end related (except the views themselves) is found from /public.

2. SYSTEM COMPONENTS



* Different content whether user is logged in or not

USE CASES AND USERS

3. USERS

A. Unregistered user

B. Registered user

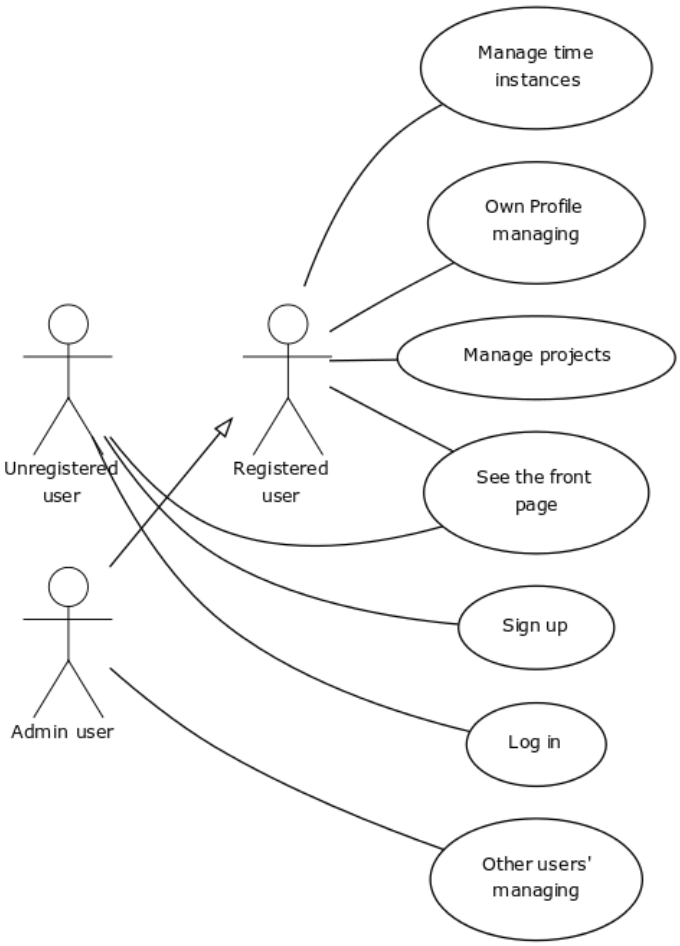
A registered user has signed up for the app and is logged in with their account. Almost all of the use cases require being logged in.

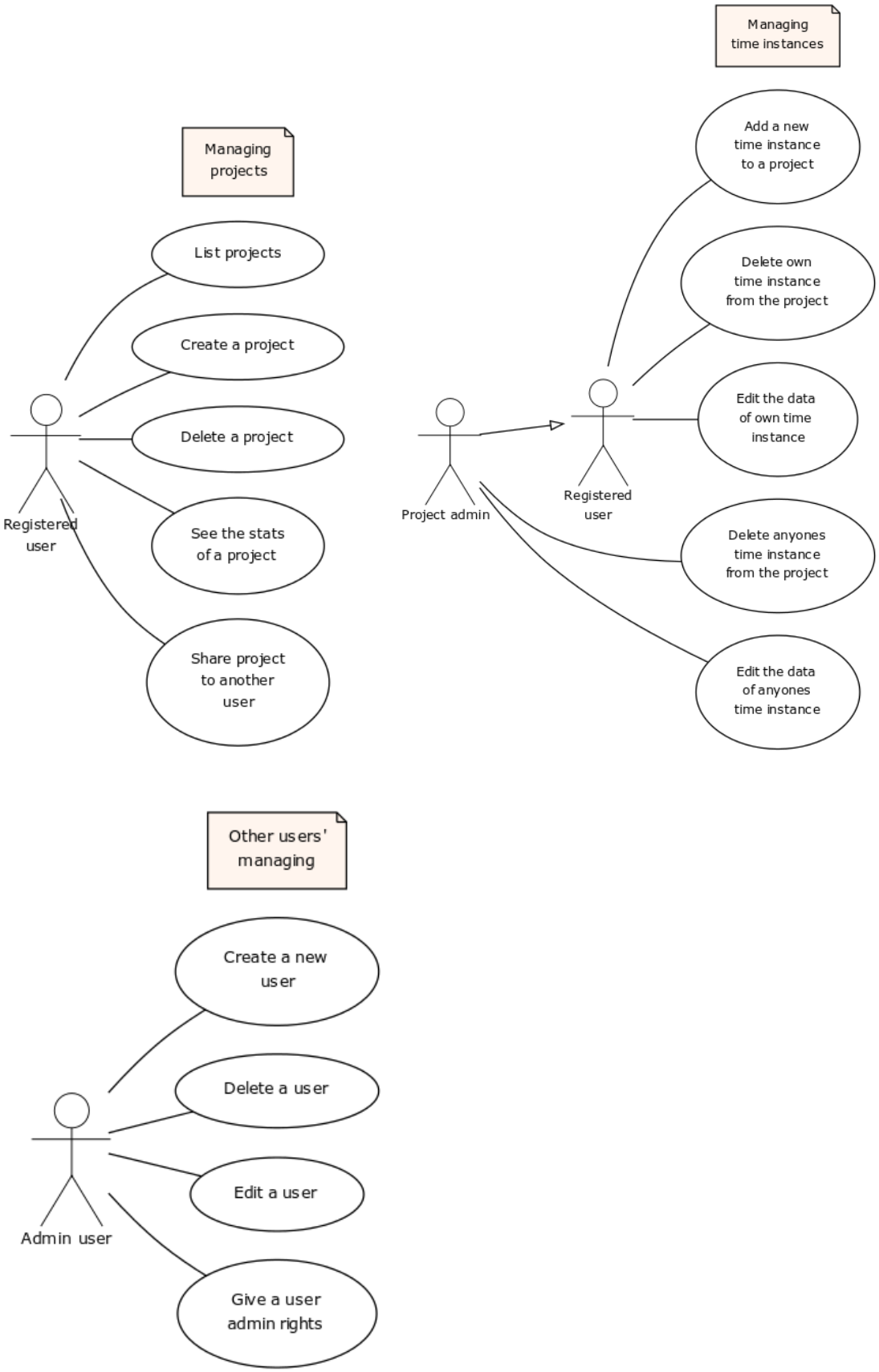
C. Admin user

Has all the rights of a registered user, but can also remove/edit other users.

4. USE CASES

A. Diagrams




Managing projects
List projects
Create a project
Delete a project
See the stats of a project
Share project to another user
Registered user
Managing time instances
Add a new time instance to a project
Delete own time instance from the project
Edit the data of own time instance
Delete anyones time instance from the project
Edit the data of anyones time instance
Project admin
Registered user
Other users' managing
Create a new user
Delete a user
Edit a user
Give a user admin rights
Admin user

B. Explanation for the most important use cases

Unregistered user

Sign up

By signing up an unregistered user creates an account for themselves. The real name, username and password are required for signing up.

Registered user

Create a project

A registered user can create projects. Projects can/must have some data, e.g. name, description and the date started. All time instances must belong to at least one project

Add a time instance to a project

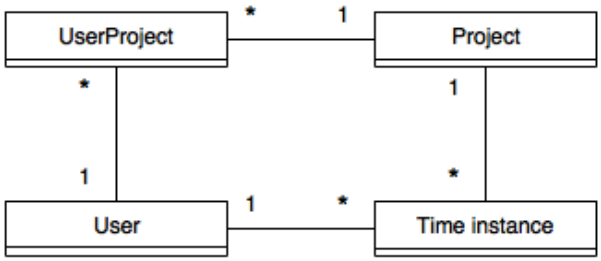
A time instance must have a date performed, description, duration and a project to belong to.

See the stats of a project

All the time instances used to a project can be seen from the statistics page of the project. The page can show at least the total amount of instances used, the total amount of time used, the average time spent per instance and so on.

DATA

1. DATA CONTENT



2. TABLES

A. User

Attribute	Type	Description	Foreign key?
id	Integer	Auto incremented id	
name	String, max 50 char	Real name of a user	
username	String, max 50 char	Unique username, cannot be empty	
password	String, max 50 char	Hashed password	

B. UserProject [connection table]

Attribute	Type	Description	Foreign key?
user	Integer		User.id
project	Integer		Project.id
isAdmin	Boolean	Is the user the admin of the project	

C. Project

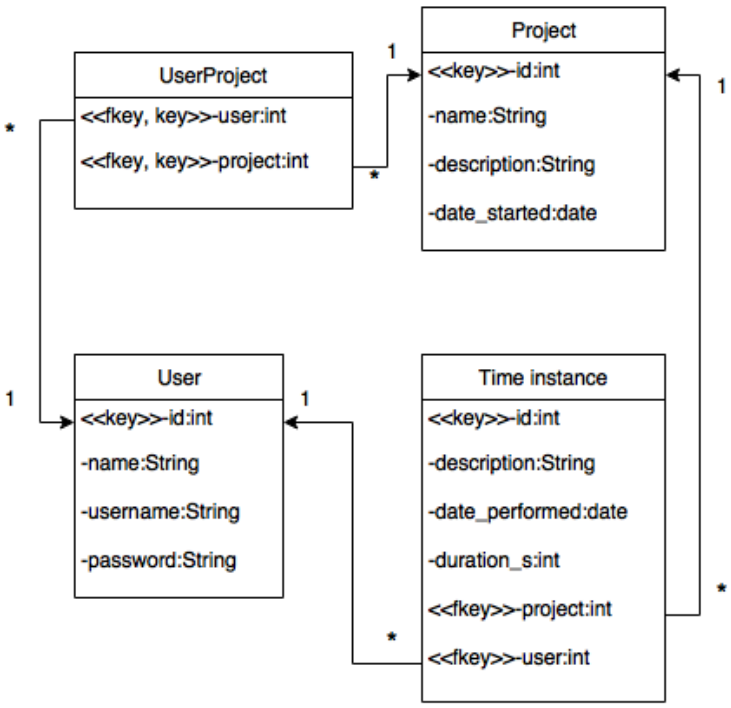
Attribute	Type	Description	Foreign key?
id	Integer	Auto incremented id	

name	String, max 50 char	Name of the project	
description	String, max 500 char	Description of the project	
date_started	Date	Date project is started	

D. TimeInstance

Attribute	**Type**	**Description**	**Foreign key?**
id	Integer	Auto incremented id	
description	String, max 500 char	Description of the time instance	
from	DateTime	DateTime when started	
to	DateTime	DateTime when stopped	
project	Integer	The project the instance belongs to	Project.id
user	Integer	The user who performed the time instance	User.id

3. DATABASE DIAGRAM



INSTALLATION AND USAGE

1. INSTALLATION LOCALLY

- `git clone https://github.com/Aapzu/trackitime`
- Go to the cloned directory
- Create a PostgreSQL database
- Create the tables by running sql/create_tables.sql to the database
- `npm install`
- `DATABASE_URL=<FULL URL> [PORT=<PORT>] npm start` where <FULL URL> is the PostreSQL url containing host, port, username and password and <PORT> the port app starts in (optional, default 8080)
- Go to http://localhost:8080 or http://localhost:<PORT>
- `sql/add_test_data.sql`

- You can also run `sql/add_test_data.sql` to the database. That creates two test users and a couple of projects and time instances to the database. Otherwise one can always create new users by signing up to the system (but not administrators).
    - Users:

Username	Password	Full Name	User Rights
admin	admin	Admin Tester	Administrator
tester	tester	Basic Tester	User

Username 'tester' also works in the published version in Heroku.