FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

**Human-Machine Interaction**

2017/2018

# Study Helper

**Stage 4:** Computational Prototype


**Authors:**

45356, António Ferreira
45412, Francisco Cunha
45640, Luís Martins

**Lab Class: P4**

**Professor:**
Teresa Romão

## Briefing

Oftentimes students suffer from a lack of organization and methodology on their studies. Study Helpers' main focus is to increase these students' productivity and motivation by making their time management easier. It will help planning study schedules and routines as students will be allowed to create tasks and divide them over specific periods of time. Also, they'll be able to create a calendar with the important dates and will be notified on unfinished tasks, upcoming exams and presentations. Lastly, they won't ever be caught off- guard with deadlines and with regular feedback on their performance over time, there won't be any risk on falling behind schedule throughout the coursework.

## The 3 Scenarios

Here we display the three scenarios that we're going to present to our application's testers (in separate cheats, of course). The reason why we show them here again is because they've been slightly modified since Stage 2, in order to avoid ambiguities and make them as clear as possible.

### Scenario 1

Today Johnny was surfing the web when he thought to himself *"I should be more productive today..."*. He stopped for a while and tried to think about the main tasks that he should add to his to-do list and complete for the day, coming up with the following two:

*"I need to study one hour for the upcoming Geography exam."*
*"I also need to email over my Physics report. It should take me 3 minutes tops!"*

He started the first task, paying close attention to the time elapsed. Because of how focused he was, he got all his studying done in much less time, and so happily proceeded to check off the task.

With all the excitement, he did the mistake of checking-off his second task as well...! What a bummer!

Shaking his head, he went to the all-time task history and recovered the task.

## Scenario 2

After his afternoon math class, Josh was super happy because he had received an 18 in the maths exam. So, he decided it would be a good idea to update his grades cheat with his latest math grade.

Afterwards, while on the bus, Josh decided to check his performance statistics throughout the last month, and realised how much his math grade had increased his overall performance.

## Scenario 3

At lunch time, Jane reminded Mark of that super difficult Physics exam that was going to take place on the 21st of December.

Mark had no idea about this event, and so he decided to add a new task, on his calendar, for the 21st of December - *"Physics exam"*, tagging it with the *Exam* category. This way, he'd start studying earlier and wouldn't forget about this important date.

## Start-up Instructions

**NOTE:** Our app is intended to be executed on **macOSX only!**

We had some trouble packaging our application. Thus, the only way for you to grab it is by cloning it from our public GitHub repository - sorry!

To get started, you should *Node.js* installed. If you don't, please go here and download the latest version. With *Node* installed, navigate to your desired location, via the command line, and execute the following commands:

- *git clone https://github.com/Study-Helper/study-helper.git*
- *cd study-helper*
- *npm install && npm start*

If everything goes right, the application should open up, greeting you with a friendly splash screen. If any errors occur, please contact us as soon as possible.

## Accessing Project History

To access our project history, go on the following link:

- https://study-helper.github.io/website/

Here you'll be able to see an in-depth record of our project's evolution.

## Incomplete Parts of Our Prototype

The incomplete parts of our application are the following:

- We removed the *Schedule* functionality and replaced it with the *Categories* functionality;
- The *Categories* functionality is fully implemented **visually**, although its backend is **not** implemented;
- Along with the *Categories* functionality, our *Settings* functionality is as

well visually implemented without its backend part;

- The "*Most used category*", "*Average task duration*" and "*Total time spent*" information in the *Statistics* functionality is not being processed by our backend, it is instead hardcoded;
- The search boxes in *History*, *Home* and *Calendar* functionalities don't work;

## On the Development Side

For this project's development, we used Electron with React. Even though being cross-platform is one of Electron's selling points, we decided to perfect it for macOS only. We also used ES6 JavaScript with Babel for transpiling.

As for the styling part, we used plain CSS and CSS-in-JS techniques (no fancy pre-processors!), and used Material-UI for our general styling toolkit.