



ToDo - System for managing user's To Do list

Engenharia Informática e Computadores
Projecto e Seminário - 2020/21

Bernardo Rodrigues, n. 42130
A42130@alunos.isel.pt

Supervisor:
Nuno Leite, ISEL, nuno.leite@isel.pt

1 Introduction

In today's fast and complex world people try to schedule days or weeks. We create small reminders, either virtually or physically, to do something, whatever that may be. Those little notes we give to ourselves, those "to-do's" are often forgotten and not fully fulfilled. As such this application will attempt to mend that by creating an environment where a reminder is easily created but hardly forgotten or skipped over. This application will allow the users to create an account and then create reminders by either writing them or taking a picture of a post-it.

2 State of the Art

The need to organize our daily lives isn't a new concept and as such, to-do applications are not new. A few examples of some of these are *Todoist* [2], and Google Calendar's reminder system [1]. The main difference between these applications and the one being developed in this project is the image recognition component as well as the prioritization algorithm.

3 Proposed Solution

The goal of the project is to create a web application using current practices and technologies. The application will be comprised of a web *API*, a database and front-end application as shown below.

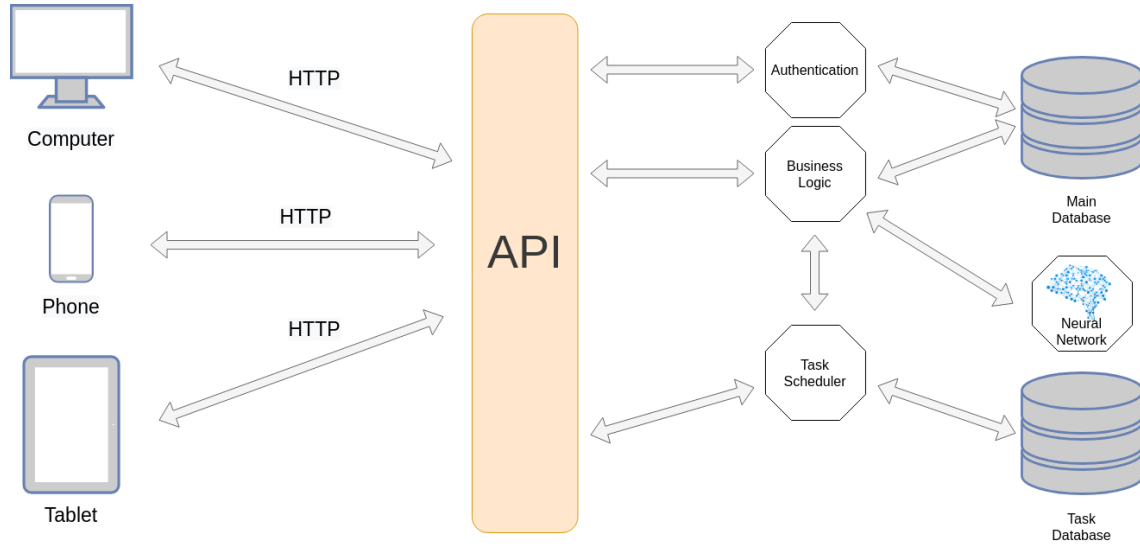


Fig. 1: Abstract architecture of the solution

The proposed solution is intended to be generic and expandable for future use. The user will be able to create, delete and edit new reminders or "to-do's" and assign them a priority as well as a date. The application will then rank these to-do's and arrange them in a way where the user will not forget them and will actually fulfill them. This will be achieved using a *REST* [3] *API* with a push notifications component.

There will be a ranking algorithm designed to prioritize these "to-do's" and then remind the user to fulfill them. This *API* will have an image recognition component that will recognize writing on post-it's for a better user experience.

The image recognition component will be achieved by creating, modeling and training a neural network[5]. The training will be achieved using a *dataset* [6] with 657 different writers.

4 Required Functionalities

1. Allow users to sign up, log in, log out and delete the account.
2. Create, edit and erase to-do's.
3. To-do's ranking algorithm.
 - 3.1. Rank to-do's based on date and priority
 - 3.2. Push notifications.
4. Image recognition creation based on post-it's (English only).

5 Optional Functionalities

1. *OpenId* [4] log in.
2. Profile Picture.

6 Schedule

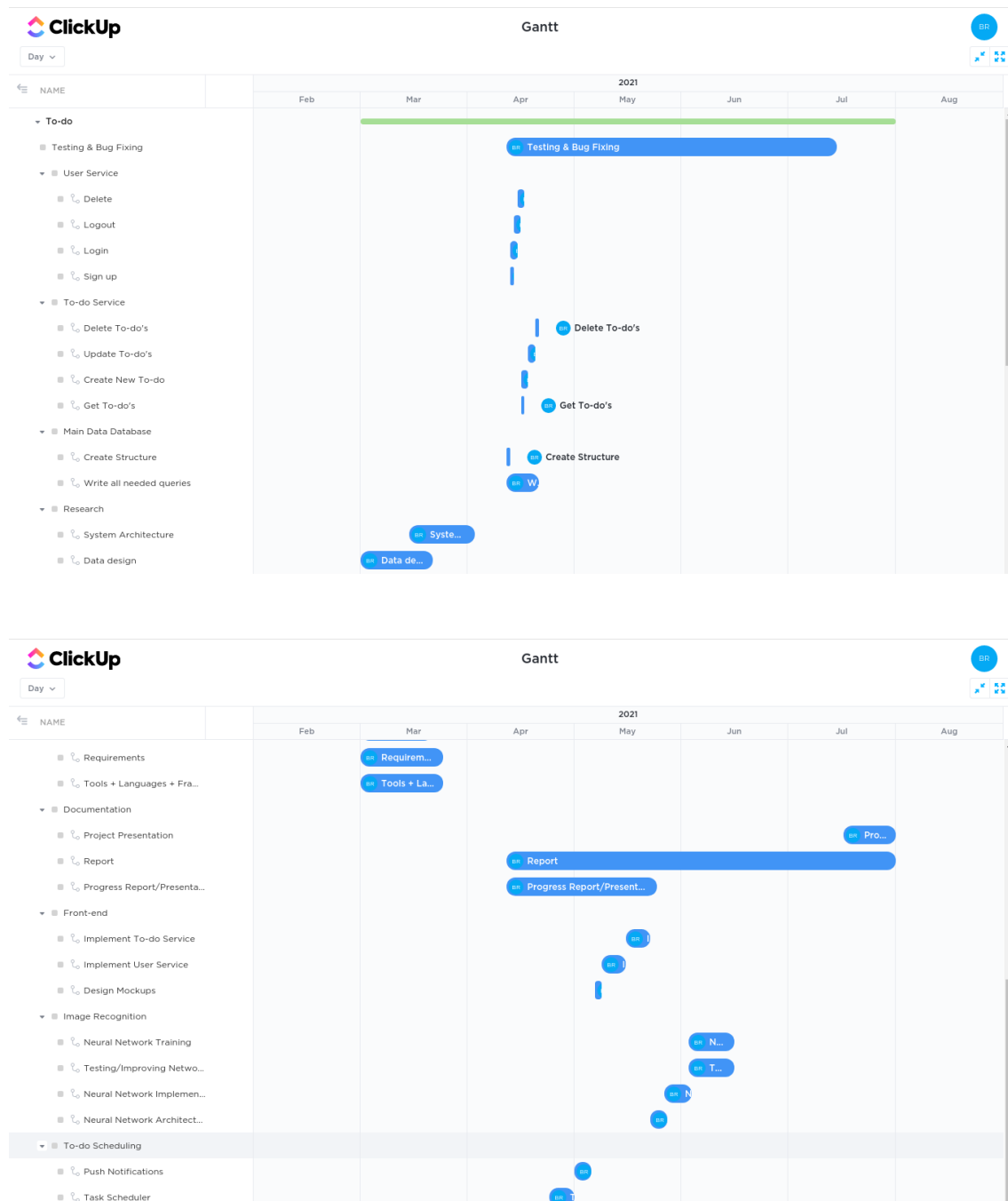


Fig. 2: Gantt chart

For a better look at the Gantt chart

References

- [1] Google calendar.
- [2] Doist. Todoist. <https://todoist.com/>.
- [3] Fielding, Roy Thomas. Rest, 2000. https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm.
- [4] Fitzpatrick, Brad. OpenId Connect, 2007. <https://openid.net/>.
- [5] Gurney, Kevin. Neural Networks. 1997. http://www.macs.hw.ac.uk/~yjc32/project/ref-NN/Gurney_et_al.pdf.
- [6] U. Marti, H. Bunke. The iam-database: An english sentence database for off-line handwriting recognition. <https://fki.tic.heia-fr.ch/databases/iam-handwriting-database>.