Technical University of Moldova

Faculty of Computers, Informatics and Microelectronics

Software Engineering and Automation Department

The Basics of Software Application Development

Modular Organizer

Authors:
Branzeanu Marinela
Cambur Dumitru
Evstafiev Nicolae
Gavirlita Ion
Nichiforov Maxim

Supervisor:
Popa Dorin

Chișinău, 2020

Contents

1 Abstract

With the development of internet and online services, as well as PaaS (platform as a service) the current lack of organizational and/or time management platforms is still not filled due to the intentions of big tech leading companies trying to outperform each other by trying to lock in the user in their ecosystem. The following report aims to analyze and give a better insight on the way a comfortable and powerful management platform should look like.

2 Introduction

2.1 Problem formulation

For our project we decided to tackle the problem of the big quantity of the calendars, reminders, task apps, in day-by-day life. This is a big problem that we encounter and this is what we base on.

From our experience, we regularly get tasks which contains deadlines. The problem is that we get them on different platforms from different organizations like university, work or other places. We get many daily events which take place on separate platforms. We can take and rewrite and introduce the same information multiple times on different calendars and this will take a lot of time and efforts. And all this can increase the chance of error occurrence, much less people that are working with many clients and customers which can daily organize different events on different platforms. They deal with focus loss and this can influence their and other lives like a forgotten meeting, undone work or missed deadline.

2.2 Solution concept

An online platform designed for vast groups of users for day-by-day life activities to more specific work flows and study schedules. The aim of the platform is to have a calendar with a list of tasks and a lot of modules that can be chosen and integrated by choice as desired by anyone. Users will have the option to connect from a catalogue of helpful modules which will then be integrated in the main calendar app for a better and more personal user experience. Like a FAQ tab or a 'get-started' option we will plan to introduce a platform wide smart-assistant that will help the user step-by-step how to utilize the platform. The end goal being a platform which will help people to organize time, as combining many platforms in one will improve work speed and will help with reminders which will make planning effortless and a lot more enjoyable than using the traditional organizational application stacks.

2.3 Motivation

Our second aim of this platform is to have all schedules arranged on one screen where users can easily interact with them as well due to the introduced assistant, interaction with the platform will be more interactive and intuitive. We want users to be able to manage their time-efficient using our platform, due to installed custom modules there are no limits on the site's abilities in terms of interaction with users, it can show incoming tasks, show the traffic situation, shows the bus in the traffic, tell about weather conditions and recommendation about the weather. Also, it can communicate with other sites via API (Application Programming Interface) (for syncing purposes) to increase even more the possibilities of user managing time using our platform like booking a place to a café, make invitations for events, booking seats to the movie theatre.

Furthermore, to motivate users to use all possibilities of the platform as well as experiment with new features we will introduce a challenges/ranking system based on how much they use all feature of the web app.

Not to get too messy at the start assistant will companion users through all necessary steps to understand how to integrate and use modules as well as how to connect and manage new calendars and how to use the all the potential of the platform.

3 Domain Analysis

3.1 Impact of the problem over the domain of study

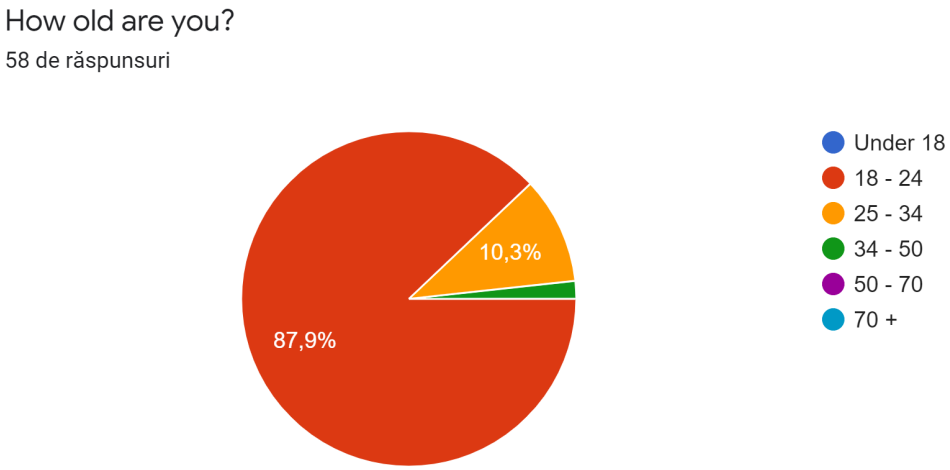
We decided to go with this project idea due to the sheer amount of learning possibilities it offers – the main requirement would be a better understanding of OOP/PO/CRUD concepts, as they will be main way of building our codebase for the project.

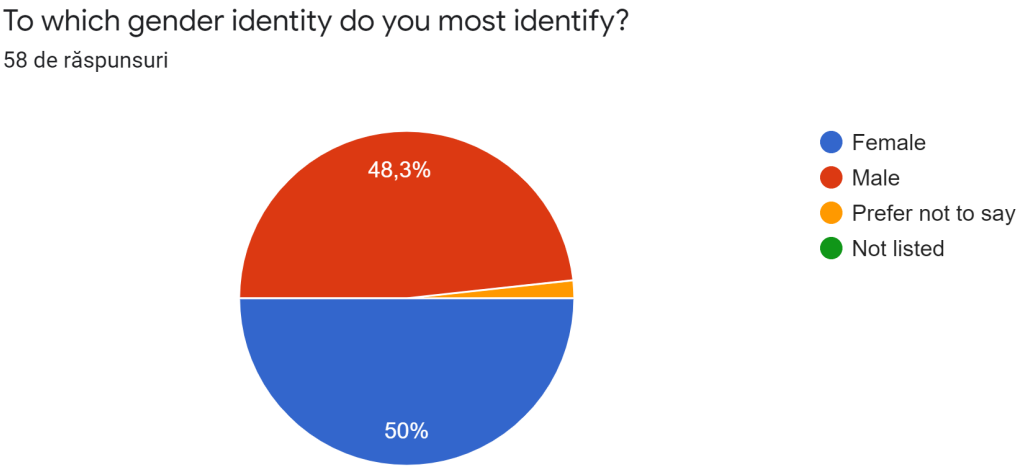
The usage of sessions and relational data-bases for storing session and user related information which has a steep learning curve but is well worth the investment as it is a fundamental practice in building todays web services and application.

And finally, the backend and technology stack used for delivering the project – this requires a good understanding of CI/CD practices as well as different frameworks and a good knowledge of the server-side part and how things work over the network (GET/POST/HTTPS requests, etc...)

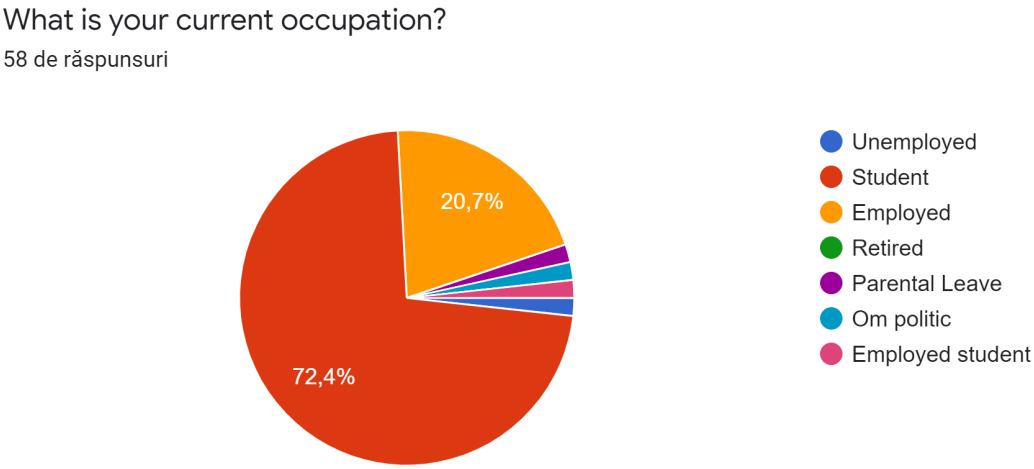
3.2 Target group

To understand the target group and customer interest in such a product, we conducted a survey in which 58 persons took part. This survey helped us to understand to which age group, gender and occupation future customers will identify.





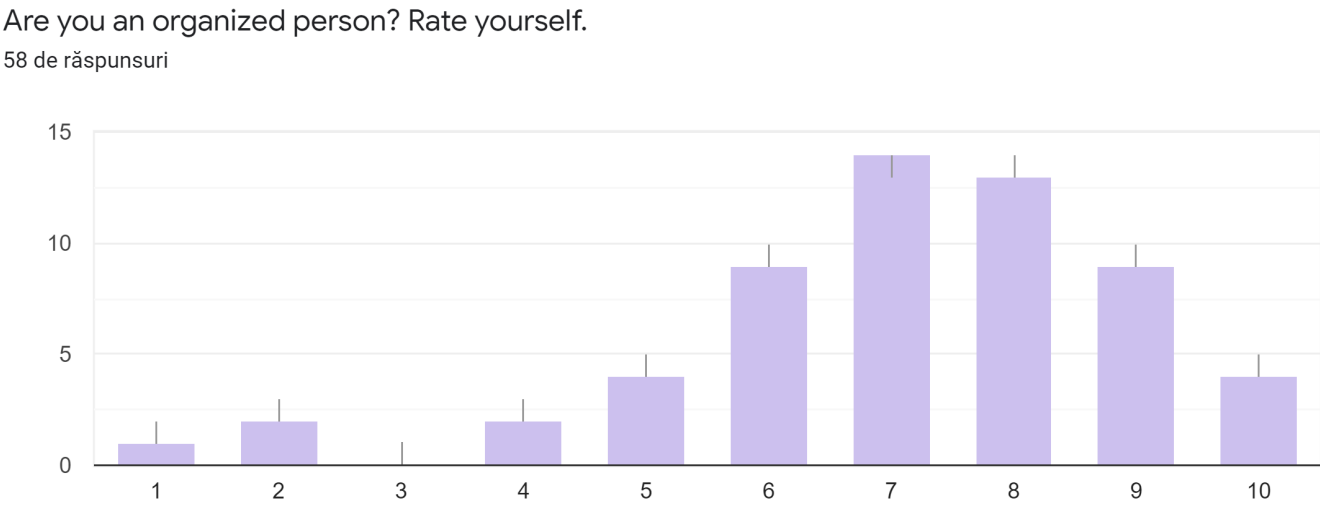
The main advantage of this survey that gender majorities are in equal parts and we have a fair opinion about our product.



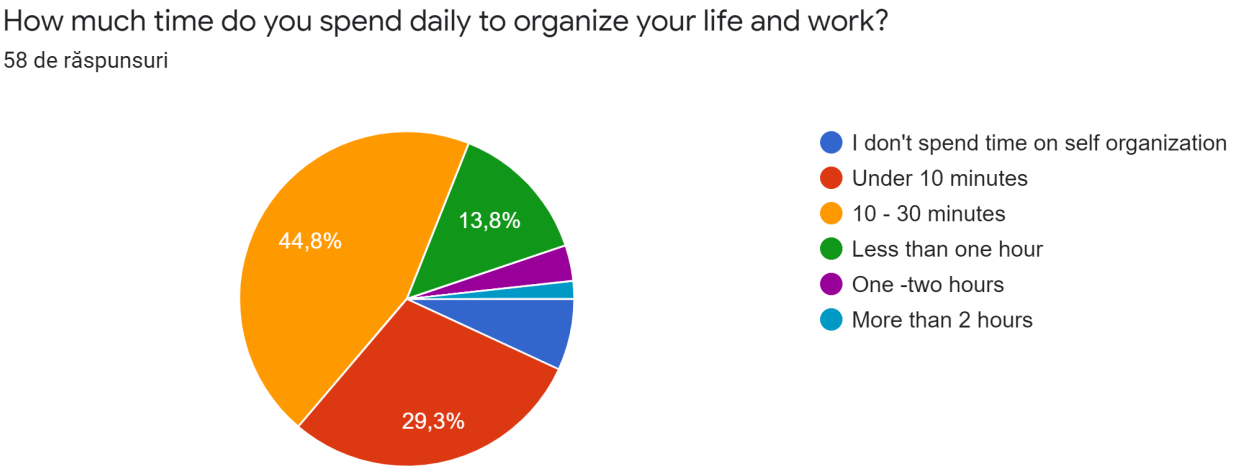
Here we can see that the main parts of audience will be students and employed people. At the end of survey, we had the input line for suggestions, ideas and someone that took part in survey had written to add one more field with employed student. We thought that better will be to add 'others' field such that people will have more freedom to describe their current occupation. This way, we have politicians here. After these 3 questions we understood that the leading audience of the app will be English speaking active study-working people aged between 16 and 35 which have a lot of tasks events which are related to teams or companies that work on different platforms, or for startuppers who have an extremely flexible schedule and need a platform which will help to organize all tasks and events on one board.

3.3 Customer validation

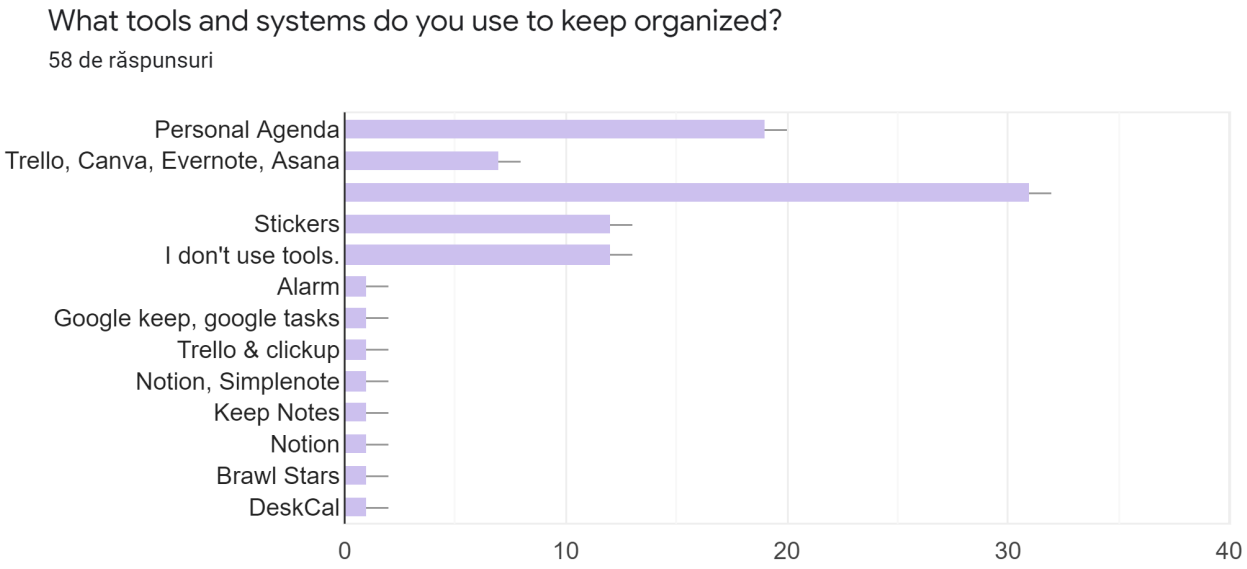
Another part of survey had the aim to find out if people are interested in such an app or not. Firstly, we had to find out which are their habits about organization and time management.



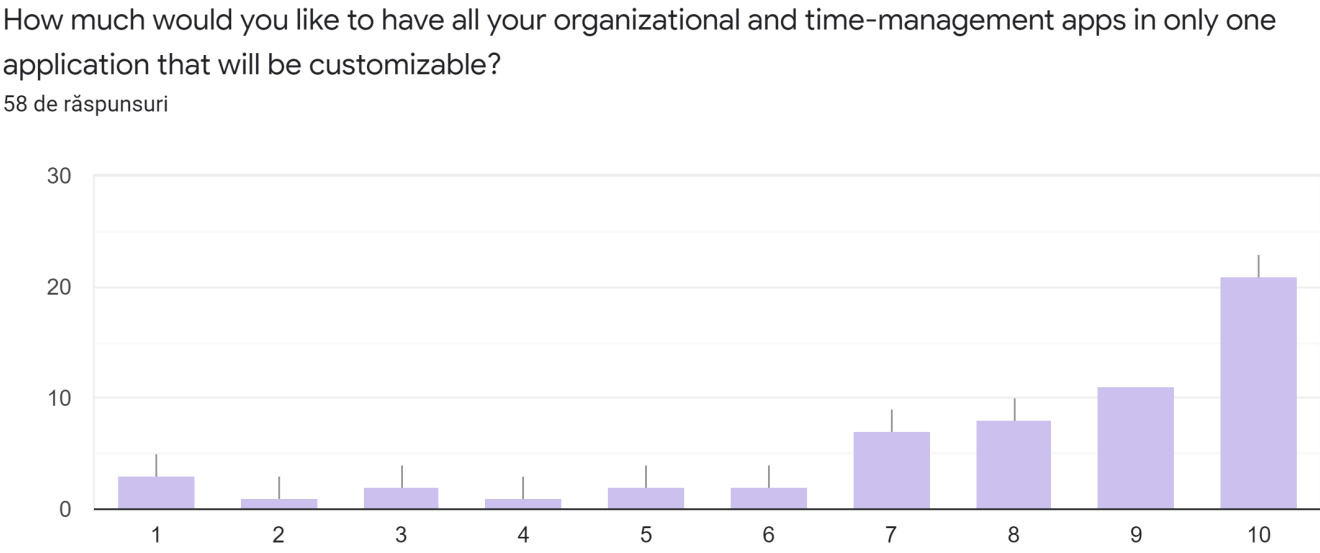
As we can see most people consider themselves moderately organized, because more than 25 people grade themselves with and 7 or 8. So we understand that people are not enough satisfied of their coordination.



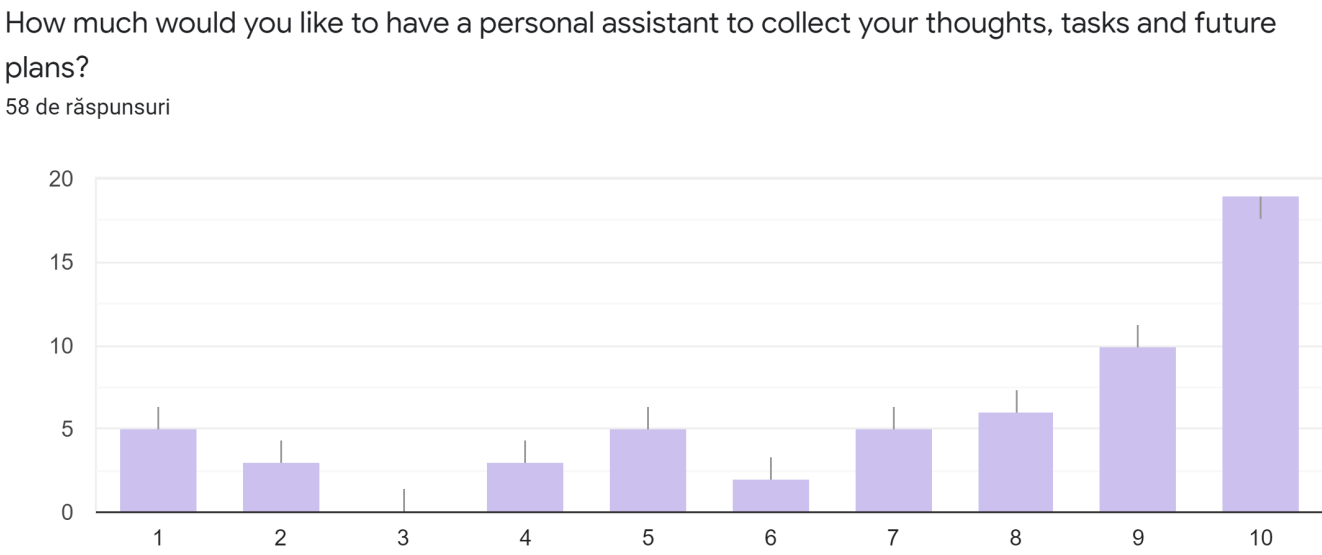
44,8 % of people spend 10-30 minutes per day to plan their life and work. Per year they spend 7300 minutes on this. It is 121 hours, 5 days. In their entire life they spend 250 days on arranging. There is a lot.



31 humans use Google Calendar, Outlook Calendar, Todoist. This means that people like to view their task, activities, meetings in a calendar way of displaying. The second place take agendas with 19 votes. This means that even that we live in very digitalized world people choose to use the paper-based type of managing their activities and thoughts.



30 assume that their desire of having an organizational and time-management app that will be customizable and will be integrated with other calendar apps is graded with 9 or 10. This is more than a half of total number of people took part and this means that the interest in such an application is highly shown. This means that developing such a concept has the right to be in production.



29 people shown surely that they want to have a personal assistant to collect their thoughts, tasks and plans. It can be understood that future customers are highly interested in both of our main advantages over competitors.

3.4 Competition

a) Google calendar is not the most intuitive app of using tasks and schedules it has a lot of special conditions to get the desired result

b) SuperSaaS is good at creating group schedule, but because of the design of the site is hard to understand what happens, saves the situation the support and tutorials page, it supports the work with forms directly when accessing the schedule which is very efficient in scope to get additional information

c) Outlook to get all the possibilities of the calendar you need to pay, is free for students and have many integrated applications which can communicate direct whit each other that increase productivity it is a combination of calendar and email it is very easy to share about events and set tasks and reminders for groups or selected people

d) Jorte Calendar theme changing, can use free but to get the chance to change the appearance of the calendar you need to pay subscription

e) Calendly it is app which works very good with establishing new meetings and events because it gave you to choose the rules that all participants can choose between the permission you establish it is time zone but the free users are limited to 1 calendar and basic functions which do not let user to feel the power of this app

4 System Design

4.1 Objectives

The system design objectives are:

- Establish the final expected result as idea/sketch
- Sketch the functionality scheme for each module needed in detail.
- Setup a server with the necessary dependencies.
- Setup a database used for storing user records
- Site mockup for: home page, plugin page, calendar page, login page, register page, account page
- Implement basic interactions between user and platform: registration, login, CRUD (create-read-update-delete) for records, implement calendar feature, import Google Calendar, add plugins
- Implement Companion feature.
- Create the possibility to add user-made plugins
- Live testing and deployment.

4.2 Requirements specification

Requirements to this platform are:

a) Intuitive and easy to use design

b) Assistant which will companion the user all the time

c) FAQ page

d) User Friendly

e) Adaptive web design

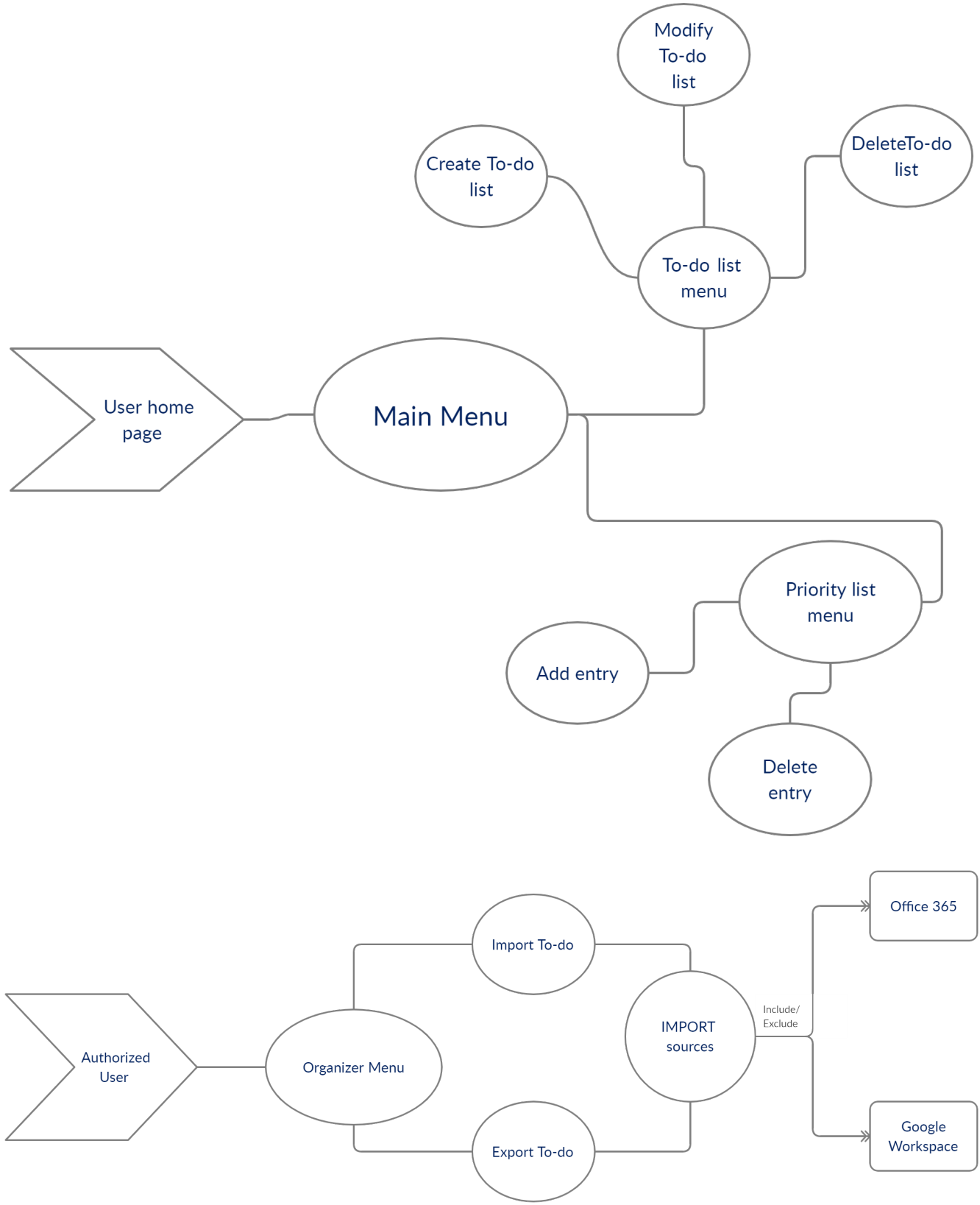
f) Deep personalization

g) Seamlessly API integration.

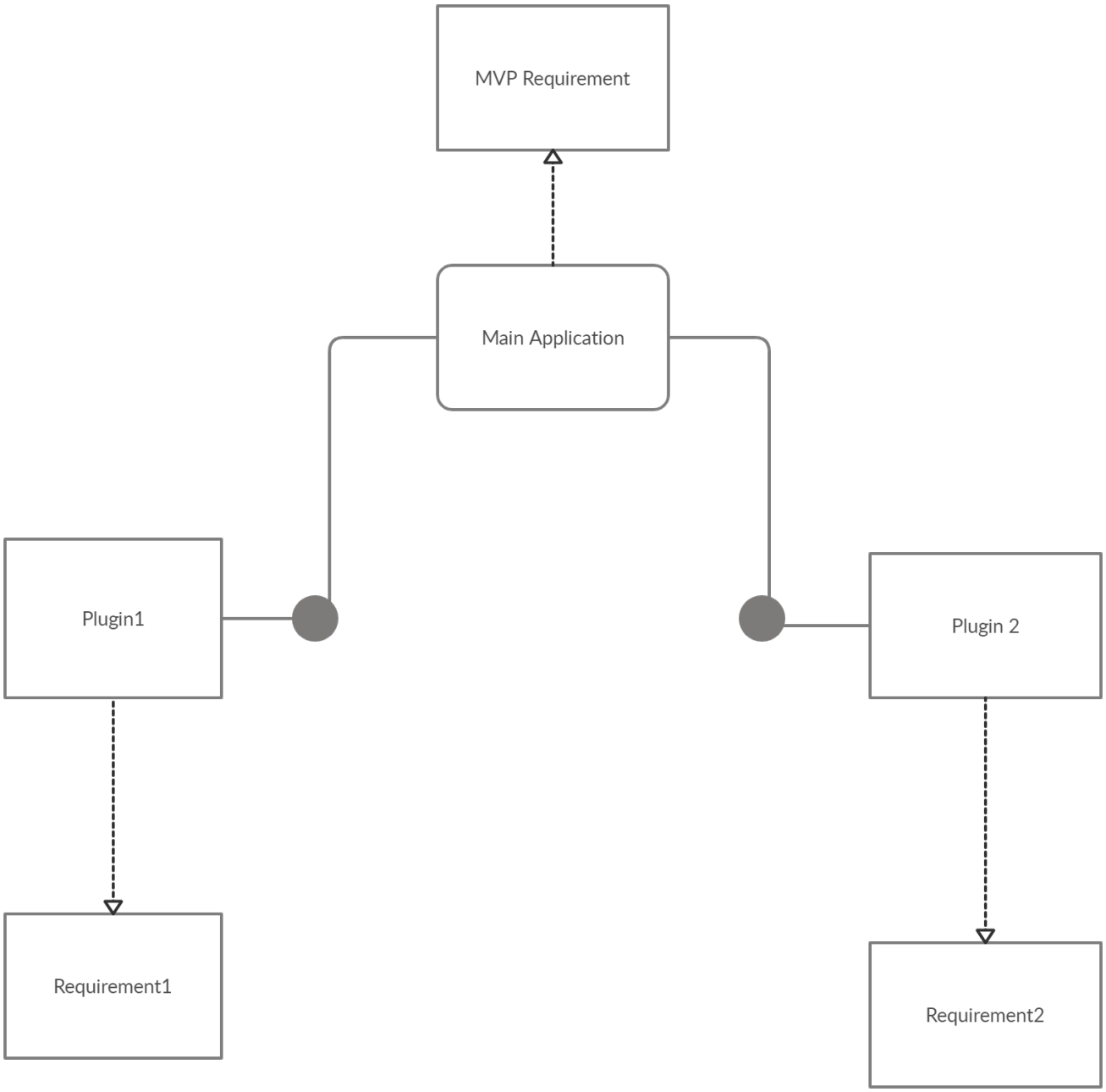
h) Community custom made plugins.

4.3 UML Diagrams

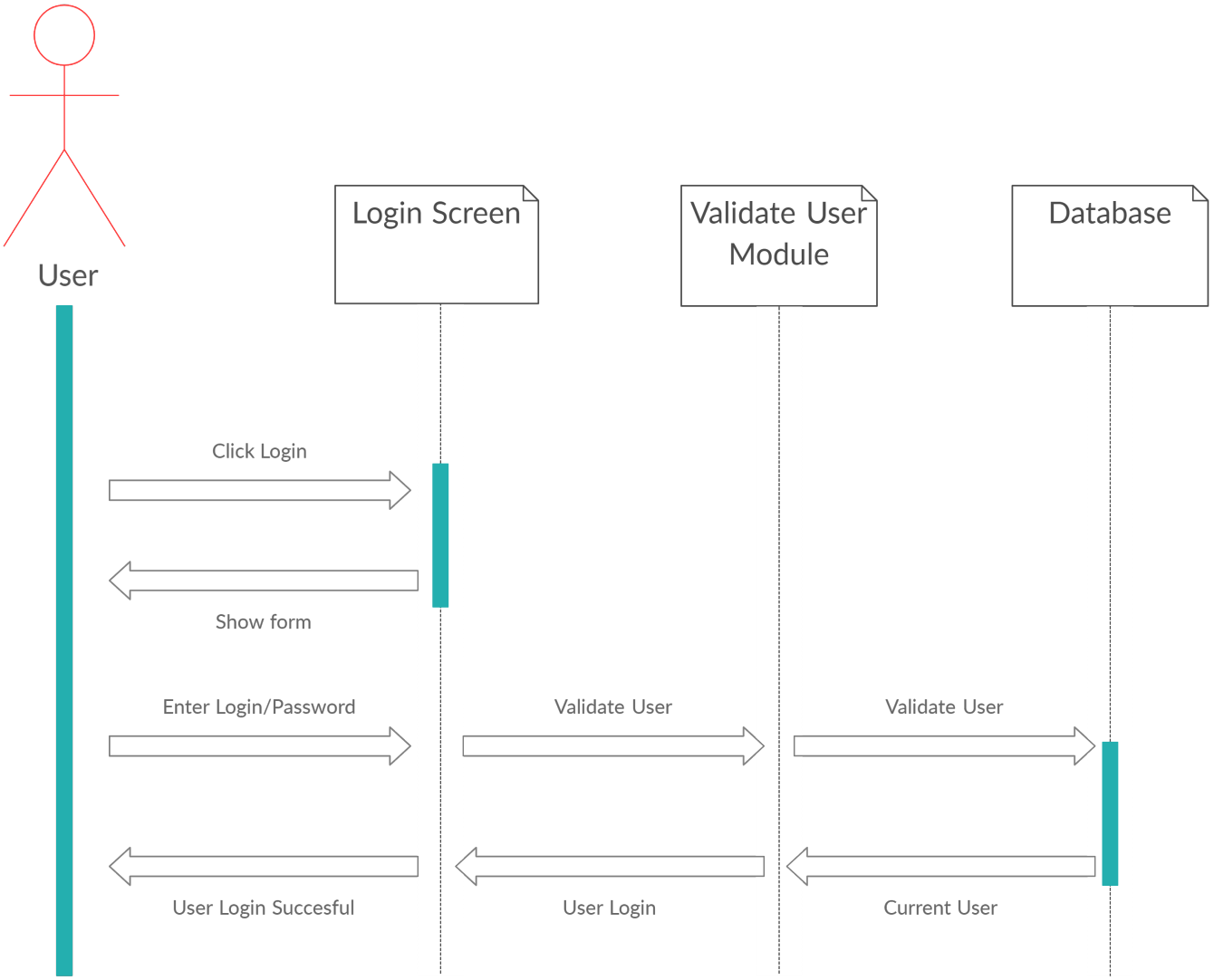
4.3.1 Use case

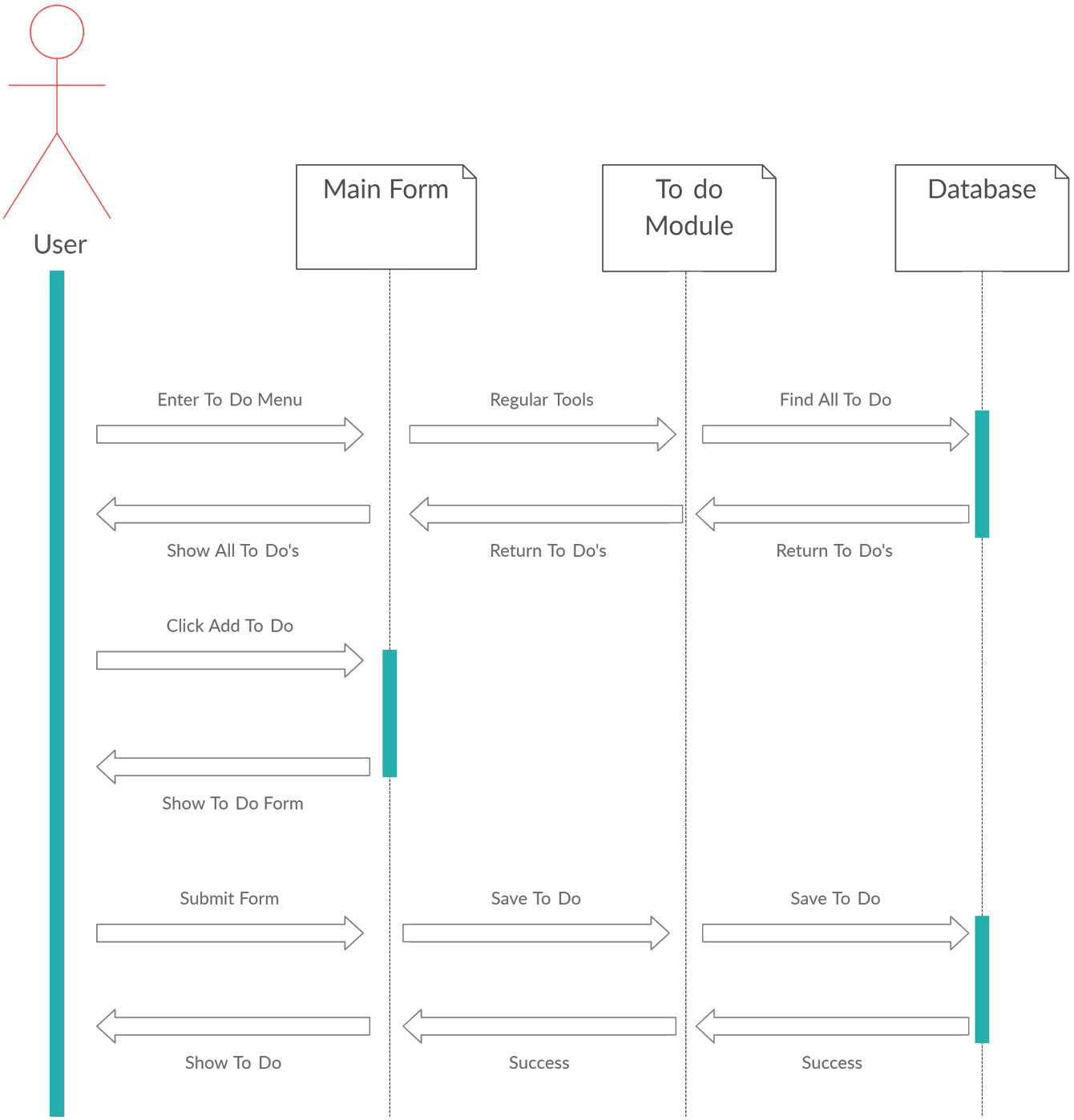


4.3.2 Component



4.4 Sequence


User
Login Screen
Validate User Module
Database
Click Login
Show form
Enter Login/Password
Validate User
Validate User
User Login Succesful
User Login
Current User


User
Main Form
To do Module
Database
Enter To Do Menu
Regular Tools
Find All To Do
Show All To Do's
Return To Do's
Return To Do's
Click Add To Do
Show To Do Form
Submit Form
Save To Do
Save To Do
Show To Do
Success
Success

5 Implementation

5.1 Solution Strategy

5.1.1 Architecture decisions

Our project is based on REST architecture. It consists of client and web server parts. Without much detail, our clients send a request to the server (like GET and POST HTTP requests), and depending on conditions server-side gives a response. For example, the client wants to see some protected page, for example, the client sends a request to the webserver. If the request is valid (for instance, the user is logged in), then we give him access to this part of the info. It is worth mentioning, that the main advantage of REST architecture is in task separation. Generally, all related to UI and Backend is separated into two parts, and interaction is done via JSON or something like XML. Besides, it makes code less chaotic and also more secure.

5.1.2 Technology stack

After a long consideration, we decided to use the next technology stack:

a) html/CSS + React (frontend);

b) Python with flask framework (backend);

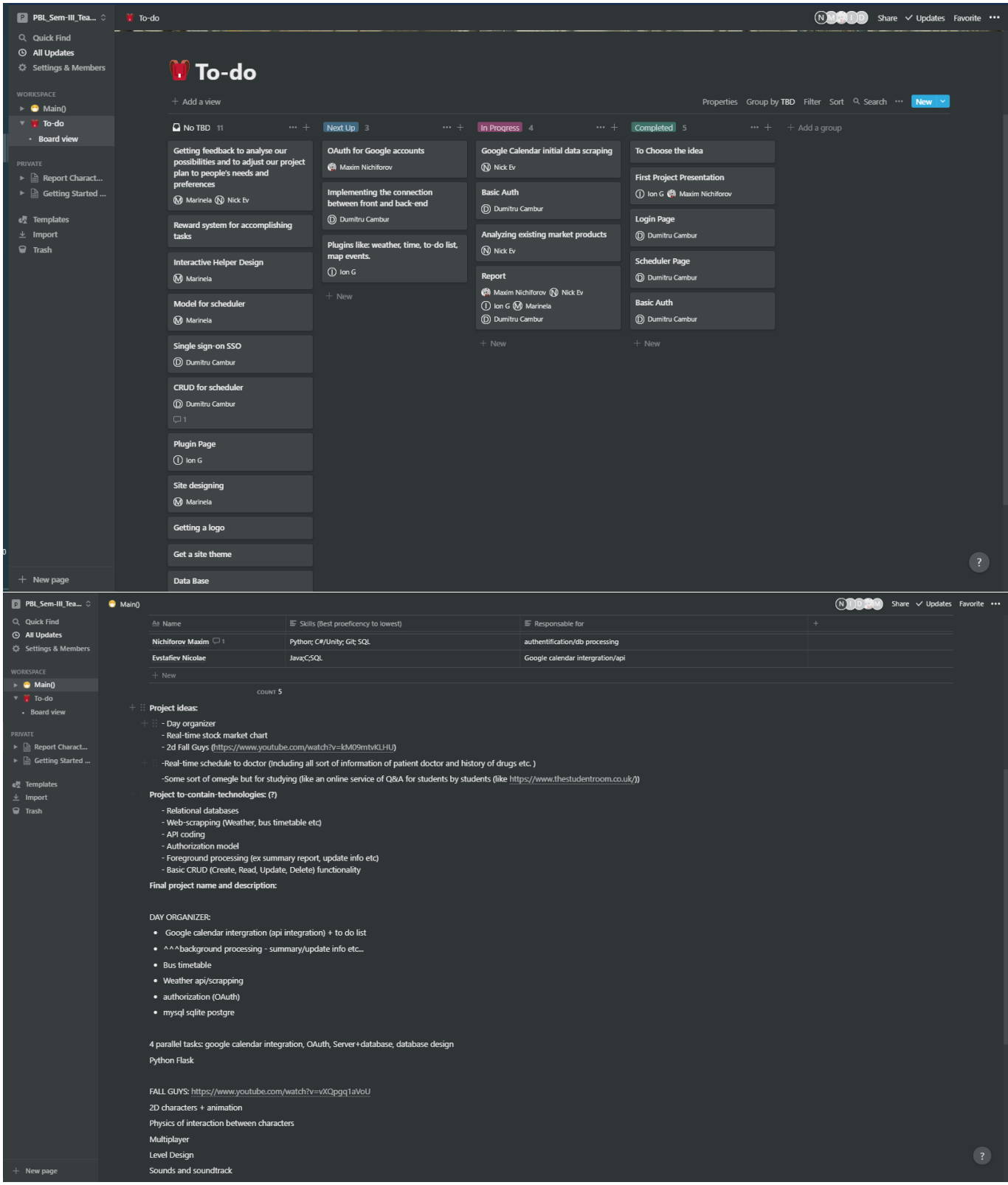
c) Nginx + Gunicorn + dokku + drone.io (Nginx - proxy for frontend and gunicorn backend, gunicorn - python wsgi server, dokku - deployment, drone.io - continuous integration)

d) Gitea (similar to GitHub) this is like architecture decision + technology stack

5.2 Project initialization

5.2.1 Task management/distribution

As management tool we choose to use notion.so - mainly because it offers students a pro version, as well as allow us to use the Notion platform as a common blackboard, to-do list and to keep easy track of what everyone has done or what is supposed to do next. Below is a snapshot of our notion dashboard:



5.2.2 Distributed Version Control of the involved repositories

We choose to use Git and to host our code base on Gitea over GitHub, as it offers better overall features that we may need in future upon developing further the project. Gitea is an open-source forge software package for hosting software development version control using Git as well as other collaborative features like bug tracking, wikis and code review. It supports self-hosting but also

provides a free public first-party instance hosted on DiDi's cloud. Source code hosted on Gitea:
Front-end part:`https://git.bitcore.nz/dima/day_organizer_fe`
Back-end part: `https://git.bitcore.nz/dima/DayOrganizer-be`
Initial work done on GitHub (no longer maintained):`https://github.com/waffle4everyone/PBL_Organizer`

Conclusion

Paragraph

References

1 Example Author 0, *Name 0*, `www.google.com`

2 Example Author 1, *Name 1*, 2002