

# Assignment 4

## Team

Full name	Innomail	Tasks
Ruslan Belkov	r.belkov@innopolis.university	Managing, Backend
Saveliy Lekhtin	s.lekhtin@innopolis.university	Backend
Mikhail Dudinov	m.dudinov@innopolis.university	Frontman
Daniil Nikulin	d.nikulin@innopolis.university	UI Design, Frontend
Artem Bulgakov	art.bulgakov@innopolis.university	DevOps, Frontend

## Git process

We continue to use the practices from the last sprint because we satisfied with them.

## Sprint

### Sprint Planning

#### Sprint Goal

- The updated system should be able to be delivered to end-user;
- The user should not be frustrated by website interface defects;
- Developers should get a convenient API for the next sprint.

#### Deadline


Until Wednesday, 02.07.2023 (3 days after sprint planning).

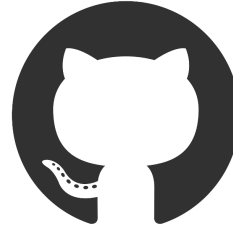
## Backlog

Sprint #2 backlog:

InNoHassle • one-zero-eight

System for storing and managing events in Innopolis

 <https://github.com/orgs/one-zero-eight/projects/4/views/1?filterQuery=milestone%3A%22Sprint+2%22>



## Milestone

On GitHub every repository has to have separate milestones:

- [InNoHassle-Events](#)
- [InNoHassle-Website](#)
- [InNoHassle-Parsers](#)

## Sprint Review

What considered as *Increment*:

- the system update from previous sprint can be delivered to the end user;
- the website interface fixed, now it has no obvious display defects;
- the API code is refactored and ready for further development.

## Sprint Retrospective

What went well:

- our DevOps is very good;

The problems we encountered:

- short sprints(3 days) are too inconvenient despite the task size limitation(specifying concrete *Sprint Goal*);
- development in small team is highly dependent on external factors (illness of a team member, general busyness of the person); There are probably a lot of disadvantages to developing in a sprint format outside of main job, for example, pet projects as a hobby.

# Quality attributes

## Portability (Adaptability)

*degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments.*

**Scenario:** User uses the system from the specific device, and system provides responsive interface

**Measure:** Stability of the system for desktop and mobile devices with various browsers

## Usability (User interface aesthetics)

*degree to which a user interface enables pleasing and satisfying interaction for the user.*

**Scenario:** User interacts with the system, the system provides it with a user-friendly and pleasant interface

**Measure:** Overall satisfaction with interface view based on A/B testing

## Usability (Operability)

*degree to which a product or system is easy to operate, control and appropriate to use.*

**Scenario:** User interacts with the system, the system provides fast access(in measure of user time) to operations

**Measure:** time, which user spent to perform specific operation in his/her first time, based on A/B testing

## Security (Authenticity)

*degree to which the identity of a subject or resource can be proved to be the one claimed.*

**Scenario:** User accesses the system using Innopolis SSO authentication, and the system verifies the authenticity of the user's identity

**Measure:** Success rate of authenticating users through Innopolis SSO and average time spent on authorization

## Reliability (Availability)

*degree to which a product or system is operational and accessible when required for use.*

**Scenario:** Users attempt to access the system during study hours, and the system remains accessible without interruptions

**Measure:** Percentage of uptime measured over a specific period (e.g., a month)

## Satisfaction (Usefulness)

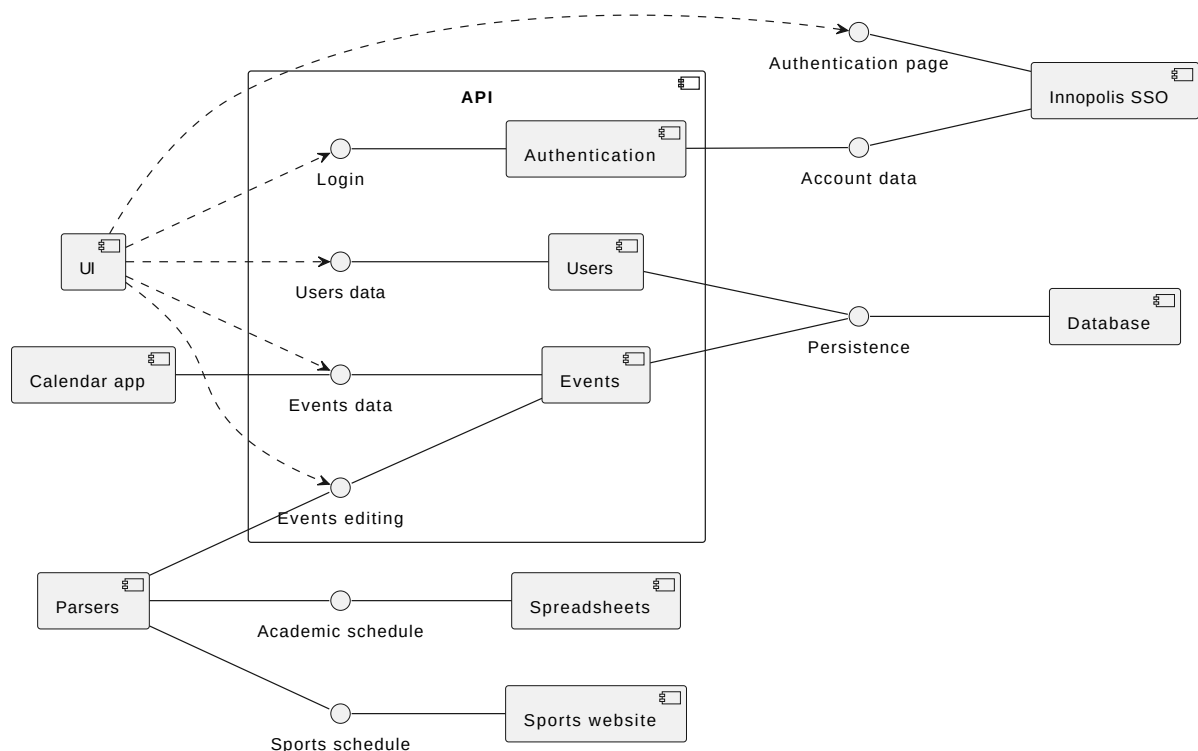
*degree to which a user is satisfied with their perceived achievement of pragmatic goals, including the results of use and the consequences of use. Quality in Use*

Scenario: User uses the system, and system collect log this visit

Measure: statistics about amount of users

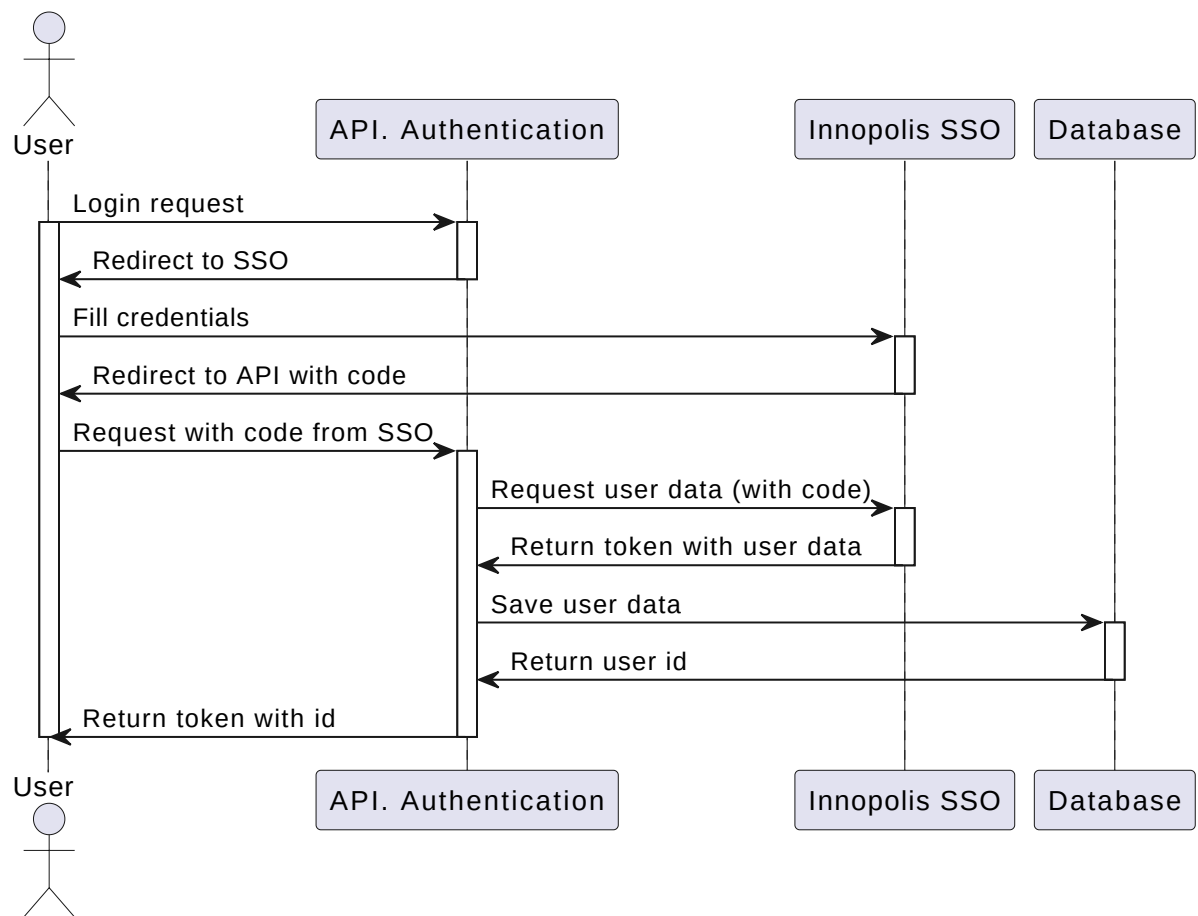
# Architectural view

## Static

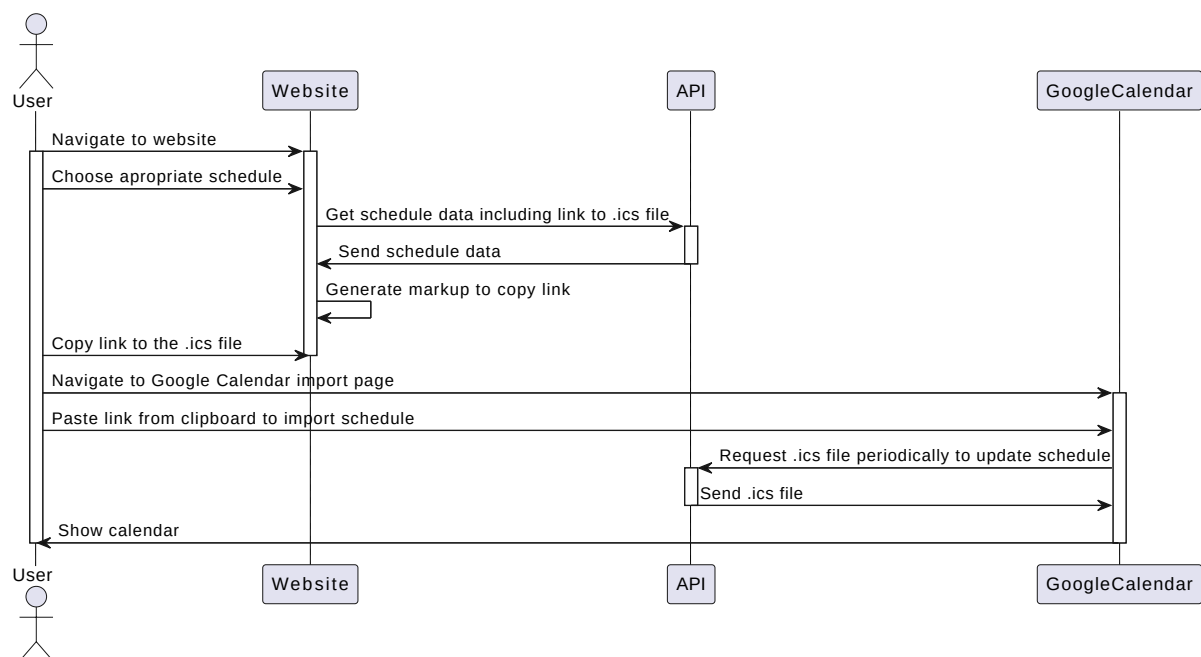


## Dynamic

### User authentication flow



Import schedule to the **GoogleCalendar** (other app have similar approach)



*The rest of the functionality can be considered obvious.*