# Requirements for Mini Meco

Summary

- Mini Meco is a software application that provides agile methods teams simple tooling for their projects.
- For us, here, agile methods tooling is what we need for our course AMOS (agile methods and open source).
- In addition, Mini Meco serves as a teaching example in our course ADAP (advanced design and programming).

Tech stack

- Mini Meco uses as few frameworks as possible to make learning the code base as simple as possible.
- Mini Meco is a NodeJS application with a React-based user interface.
- All data is stored in an SQLite database.

Key data structures

Users and projects
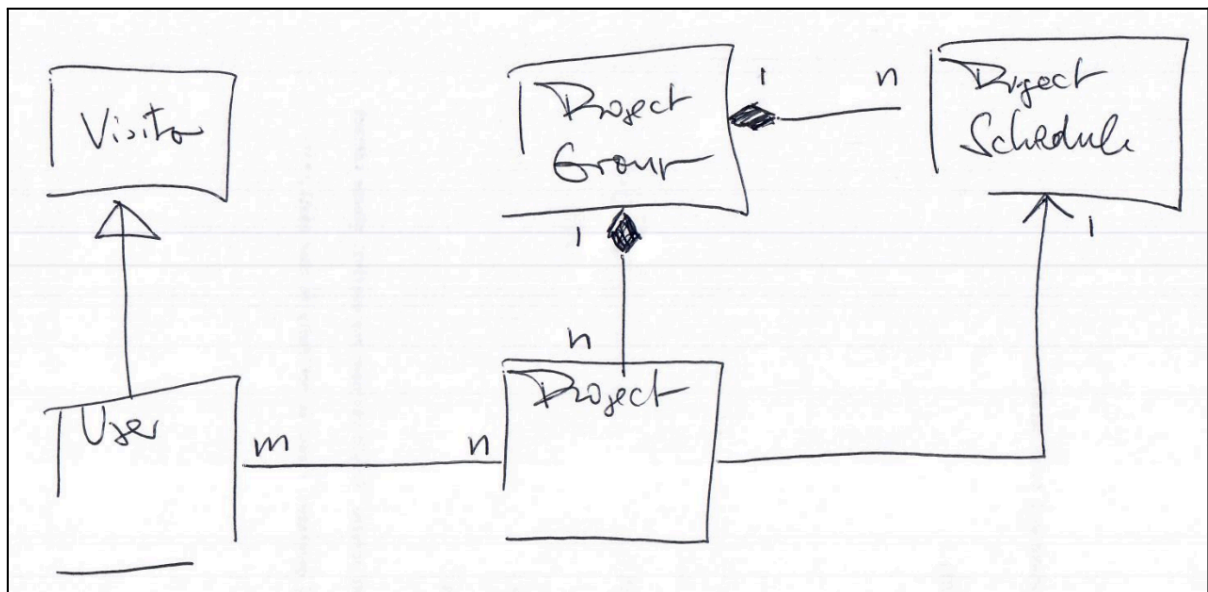
User

- A user has their email address as their name, and a password
- There are two types of users, regular and admin
- A user has a flag identifying them as an admin
- A user has a status life-cycle: Unconfirmed, active, deactivated
- A user can be a member of any number of projects

Project

- A project is a software development project
- It has a set of members (users)
- It belongs to exactly one project group

Project group

- A project group is a collection of projects
- The group defines a schedule (of timeboxes a.k.a. sprints) for the projects

Example project groups are AMOS #22 (summer 2024) or ADAP course, example projects / classes specific amos projects or a particular ADAP class instance.

Views and models

Views

Views are React components or components that hide/wrap React; they have one model.

Models

Models represent the domain functionality of the view/model component; they come as a three-level hierarchy:

- Dashboard is the top-level i.e. the main screen
- Categories are the main sections on the main screen
- Features are large functionalities within a category

See below for examples.

Plugins

- Plugins are view/model combinations
- Plugins start with feature view/model combinations

Functionality

Login screen

- Visitors see a welcome page with a login opportunity.
- Visitors can register to become a user and login.
- Users can trigger a password reset email.
- There is one predefined admin user email address, dirk@riehle.org.

Main screen
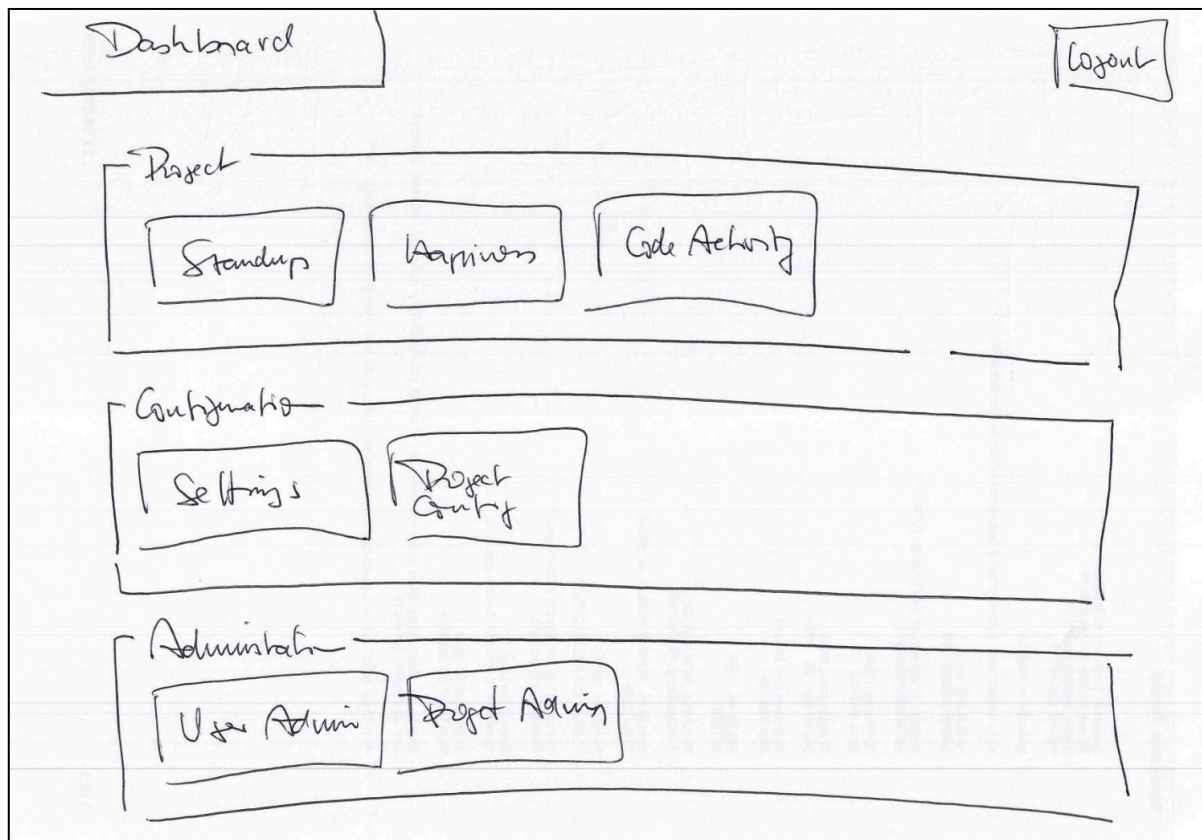
- The app is hierarchically structured into components; each component has its own view (UI) and its own model
- The component hierarchy has three layers: Dashboard, category, feature

The main screen (after login) shows the dashboard; the dashboard groups categories on the screen; within categories, there are features

- Dashboard
  - Projects
    - Standups
    - Happiness
    - Code activity
  - Configuration
    - Settings
    - Project config
  - Administration
    - User admin
    - Project admin

Dashboard

- The dashboard has a current user attribute
- The dashboard contains categories as subcomponents
- Visually, the dashboard is the main screen

Project category

- The projects category is a subcomponent of the dashboard
- It offers all features for software development (standups, happiness, coding) as subcomponents
- It has a current project attribute (that can be changed to another project for which the user is

registered)
- Visually a category like projects is a subsection of the dashboard and a frame around the features

Projects category

- In all project feature screens, the user can change their current project from the list of projects they are registered to
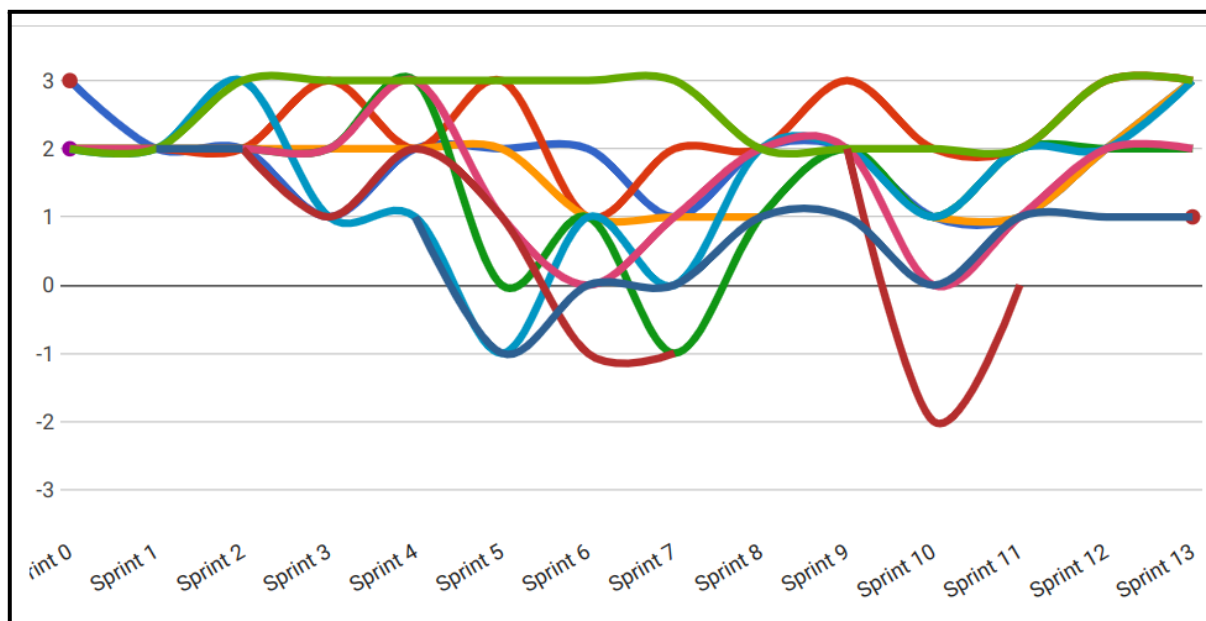
Projects / standups

- The standups feature lets a user send everyone a standup email

Projects / happiness

- The happiness feature lets a user declare their happiness for the current time box.
- It shows the project team's happiness over time (as a graph)
- All users on the happiness index are anonymous to each other (just curves)

Admin interface for project happiness index

Projects / Code activity

- The coding feature shows the users # of commits for a given timebox (as a graph)

Configuration category

Configuration / settings

- Regular users can change their email address, admins cannot
- Users can change their password
- Users can join or leave a project; they can see all projects

Configuration / project config

- The user can select a project they are a member of
- The user can set and change the git URL for the project

Administration category

- The administration category and features require that the user be an admin

Administration / user admin

- The admin can see three separate lists of users, by status (unconfirmed, active, deactivated)
- For unconfirmed users
    - The admin can send unconfirmed users a password reset email
- For active users
    - The admin can deactivate an active user
    - The admin can deactivate all active regular users
- For deactivated users
    - The admin can activate a user

Administration / project admin

- The admin can CRUD a project group

- Delete is not possible as long as there are projects in the group
- The admin can CRUD a project
  - A project has a name and belongs to a group

Code organization

All code belonging to one domain is captured as one plugin and gets one folder for its source code.