



Planner Design Description

Version 1.0

Revision History

Date	Version	Description	Author
04/11/2015	0.1	Initial Draft	Joan Josep Crespí Villalonga
04/11/2015	0.2	Introduction	Joan Josep Crespi
12/11/2015	0.3	Background, Architecture, Back-End, User Interface, Futures extensions	Jean Barré
13/11/2015	1.0	Cleaning of the document. Added technology part.	Giorgio Pea

Table of contents

1. Introduction

1.1 Purpose of this document

This document aims to show and formalize the design and the architecture of planner. The contents of this document are not immutable: changes can happen during the implementation of the application and so new versions of this document will be produced. This document is made taking in consideration the Plunner's Requirements Definition Document

1.2 Intended audience

The intended audience is:

- Customer
- Team members
- Supervisors

This document represents the guide team members of Plunner must use to develop the application so that the objectives of the application itself are achieved and the final customer is satisfied.

1.3 Objectives of the application

Plunner aims to:

- Simplify the organization of meetings in companies
- Optimize the date and time for meetings in companies
- Automate the process of planning meetings in companies

1.4 Target of the application

Plunner targets small and medium companies. This choice affects in a significant way how Plunner is conceived and supposed to work

1.5 Definitions and acronyms

1.5.1 Definitions

Keyword	Definitions
Use Case	List of action or event steps, typically defining the interactions between a role and a system, to achieve a goal.
Mock-Up	Drawing illustrating our vision of the view.

1.5.2 Acronyms and abbreviations

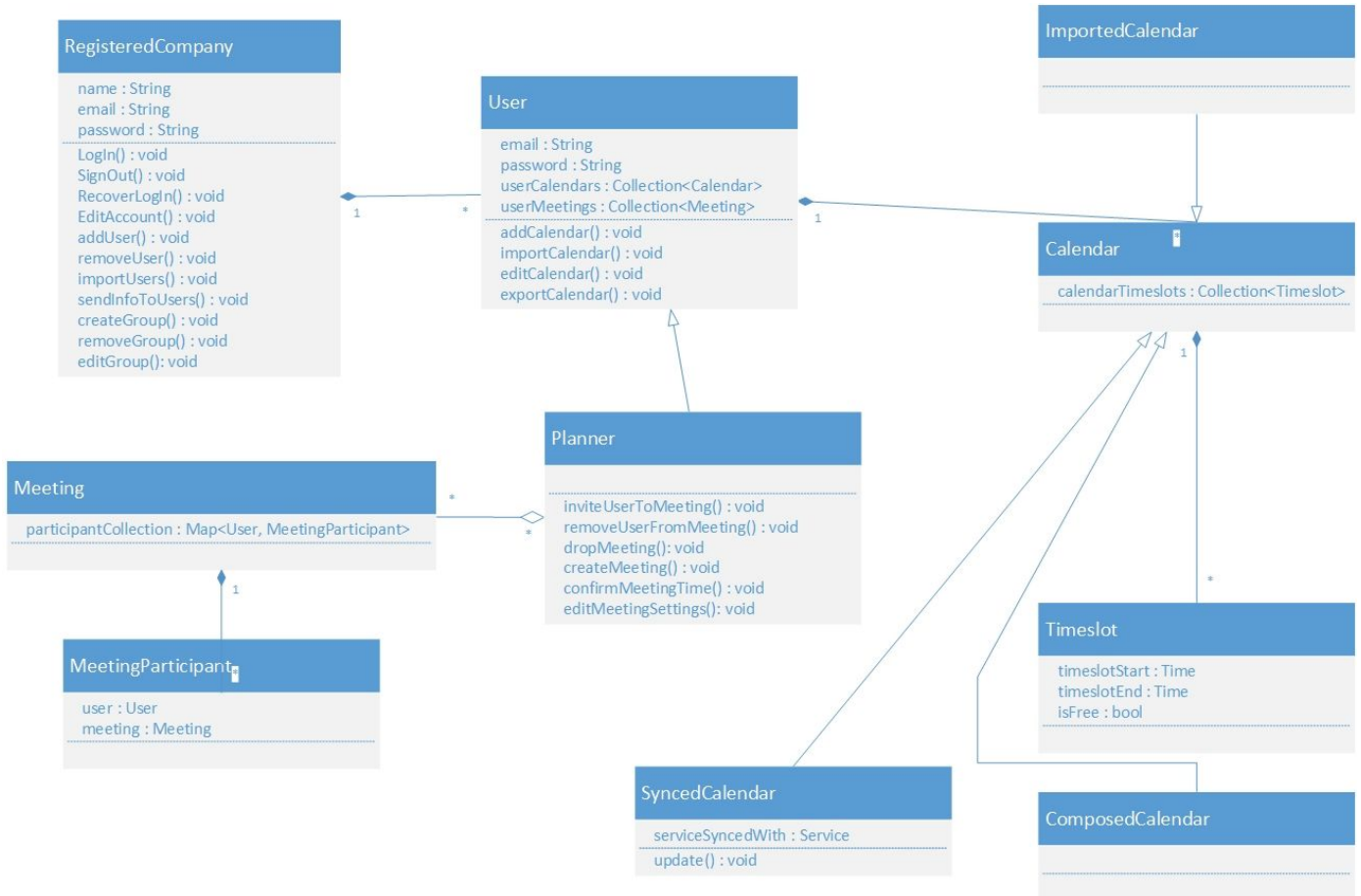
Acronym or abbreviation	Definitions
DSD	Distributed Software Development. It's the course in which this project is being developed
FER	Faculty of Electrical Engineering and Computing, Zagreb, Croatia
MDH	Mälardalen University, Västerås, Sweden
EXT	Extension
calDAV	HTTP protocol, extending webDAV, standardised automated scheduling transport.
MVC	Model, View, Controller
RD	Plunner's Requirements Definition Document

1.6 Actors

- Company(self explanatory)
- Registered company(see RD)
- User(see RD)
- Planner(see RD)

3. Back-End design

Our model will respect the following figure.



4. Architectural style and patterns

4.1 The main design

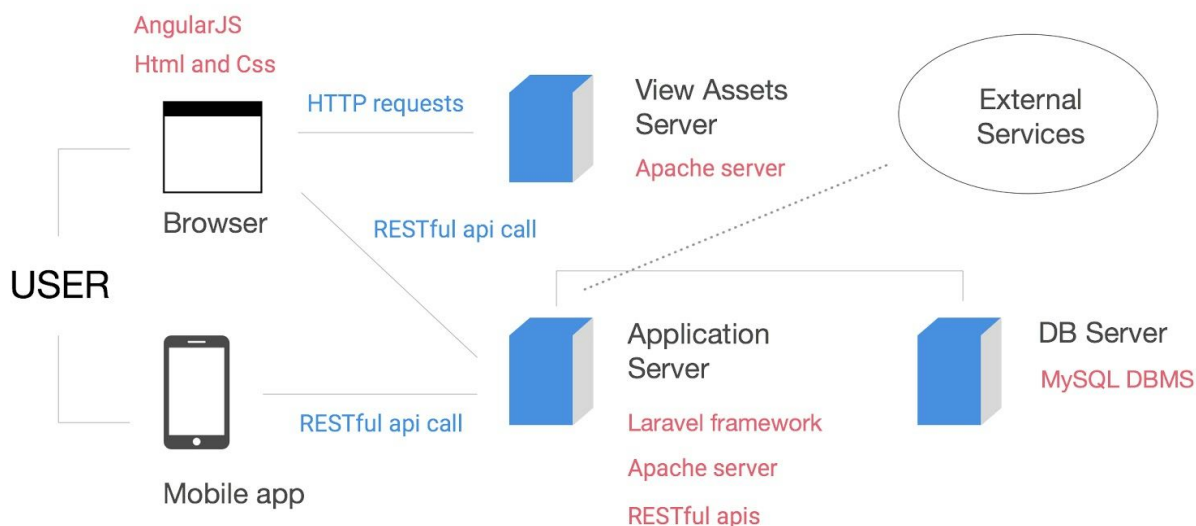


Figure 2 : Architecture Scheme

From the client point of view, HTTP(S) calls our HTML, CSS and Angular files. This repertory will be stored on an Apache Server to constitute the WebApp and the Front-End side of our project.

The use of JavaScript and AJAX should allow us to support the most popular and modern browsers.

Requesting our API (via RESTful) which one will reply in JSON format.

On the Back-end, our API server and task service will be implemented in PHP, as it appeared to be a shared knowledge in the team, using Laravel's framework. From there our DataBase will be called by MySQL request.

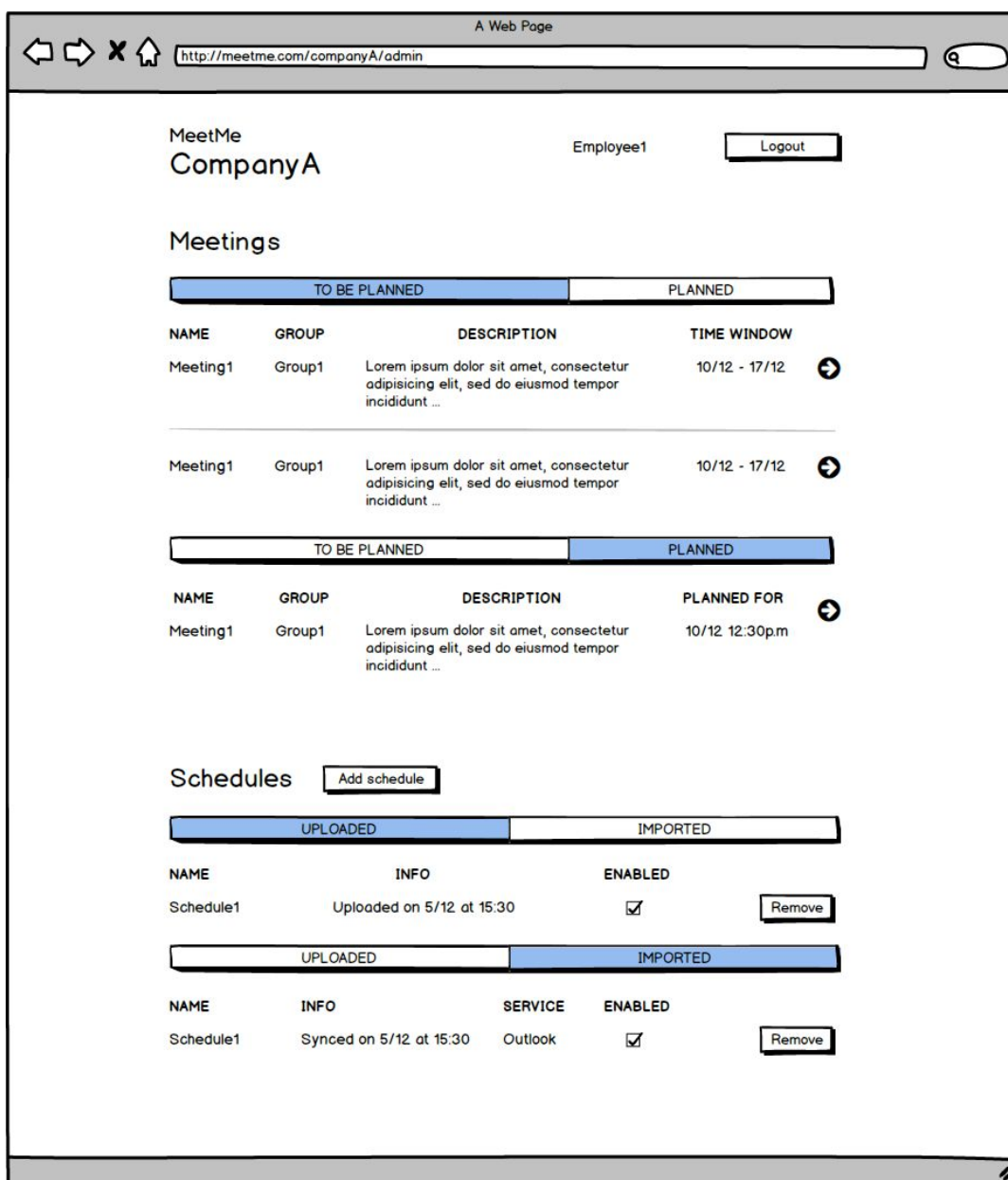
4.2 The MVC pattern

The View is managed by the Client server. User's laptop uploads all the HTML and style sheet files. The behaviour of the user on those pages is then managed by the Controller implemented in our API. From this API server, all the data retrieved and stored in an SQL database are called using our Model.

5. Graphical User Interface

Regarding the user interface, mock-ups have been designed. Only the most important ones are displayed

5.1 User View



5.2 Planner View

A Web Page
http://meetme.com/companyA/admin

MeetMe
CompanyA

Employee1 Logout

Meeting 1

Title

Brief description

lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Meeting place

Export the planned meeting date to your imported schedules

Temporal window

NOVEMBER 2015

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Meeting repetition

Planning deadline

Participants Add participant

NAME	EMAIL	REQUIRED	
User1	User1@companyA.com	<input checked="" type="checkbox"/>	Remove

Your schedules for the meeting Add schedule

NAME	INFO	
Schedule1	Uploaded on 5/12 at 15:30	Don't use

Your free time slots for the meeting

View and edit

Save changes

Figure 4 : Planner Setting View

5.3 Administrator View

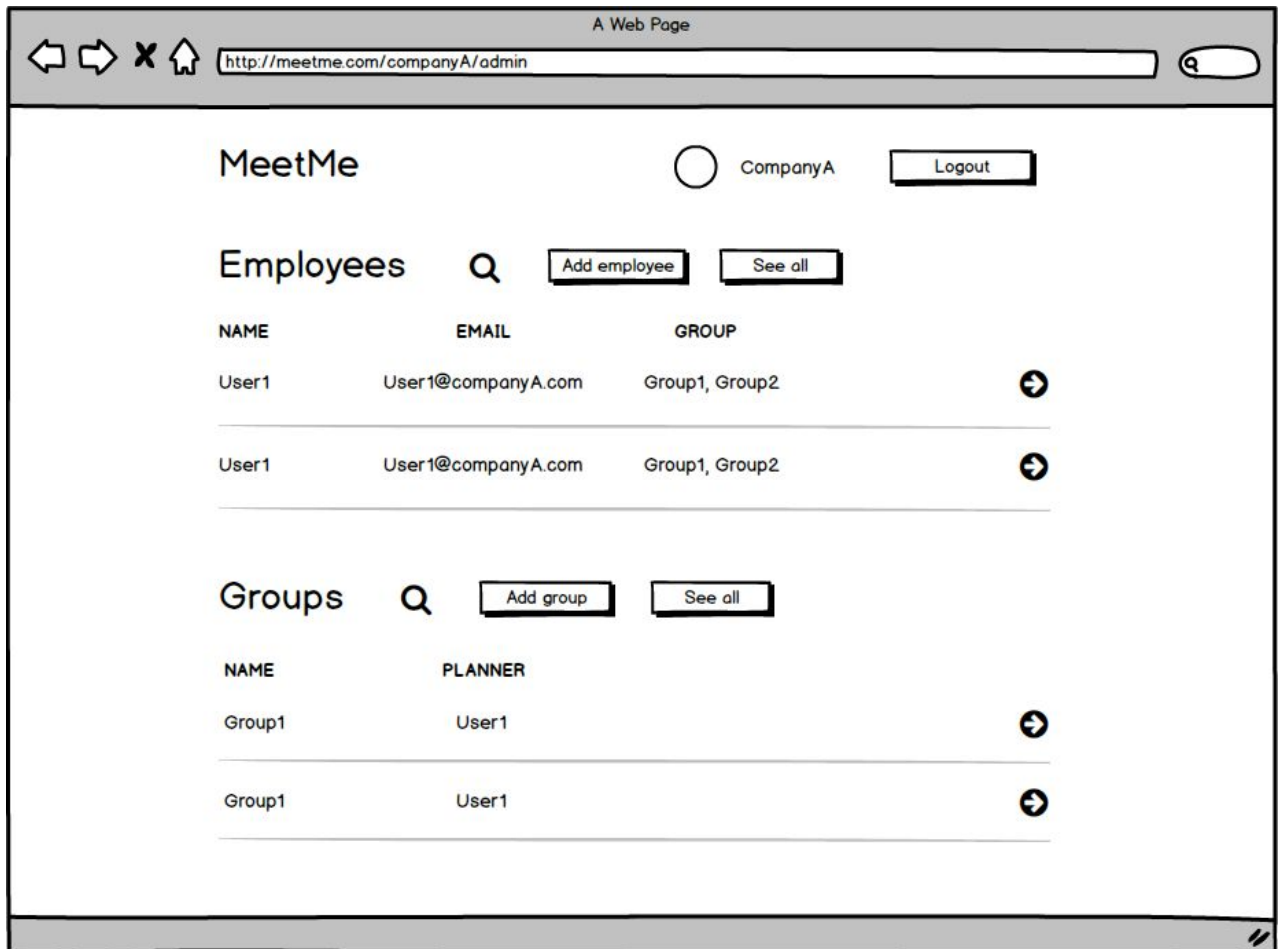


Figure 5 : Administrator Edit View

6. Technologies

- Html
- Css
- JavaScript
- AngularJS framework
- PHP
- Laravel framework
- MySQL
- JSON

7. External Services and other design constraints

Calendar services to be integrated:

- Google Calendar
- iCal
- Windows Calendar
- Outlook Calendar

Protocols for exchanging calendar data:

- calDav(see Definitions)

7. Future Extensions

This part is optional and represents the future extensions that the project may have, in order to try to improve or create new functionalities.

Graphical interface is planned to be responsive, so it shouldn't be a problem to extend our target to mobile device as presented in the drawing. We could then set the start page of the web application as the target URL for the device's browser. It would be a quick solution for mobile integration if we lack of time. As native application offers a better user experience but is longer to develop, the mobile side of our architecture remains optional.

ID	EXT001
Name	Mobile Integration 1
Description	mobile integration using the device browser to target our web application
Reason	Make our system mobile quickly
Priority	Low : we would rather a native version for user experience
Effort	Low : our implementation is web responsive

ID	EXT002
Name	Mobile Integration 2
Description	mobile integration using full native tools of iOS and or Android operating systems
Reason	Make our system mobile
Priority	Medium
Effort	High

