MeetMe Planner - Requirements Definition

Version	Description	Date	Authors
0.1	Initial Draft	04/11/2015	Mehdi
0.2	Update	05/11/2015	Mehdi Milica
0.3	Update	06/11/2015	Carolina Armindo Mehdi
0.4	Update	10/11/2015	Mehdi

1. Description and definition

1.1 Background

Today's planners do not meet the needs of a user. Person willing to easily schedule an event can stump into many problems in the process of organizing a meeting. The problem can become even more complex when the number of people willing to attend the meeting grows.

The proposed timeslots can be spread over few different meeting times, because each of the attendants have their own schedules they have to cope with and they have scheduled before. Because of the many constraints force upon the other attendances it is hard to find appropriate time slot that can work out for everyone. Usually in this situation the meeting is held over email conversation, which can become quite messy.

1.1Description

To solve these problems, we need to create a meeting planner that will manage to merge the information from all attendances calendars and manage to find a free time slot that fits everyone. MeetMe planner should be able to draw information from applications such as Google Calendars to provide enough data for finding an appropriate time slot for the meeting. And the application should be able to propose a time slot by respecting the privacy and security of the shared attendances calendars i.e. with minimum disclosure of the attendance plans.

1.2 Actors

Visitor — A visitor is a potential user of the system which can perform the following use cases :

- Registration

User — The system's user can perform the following use cases :

- Login
- Recover password
- Schedule meeting
- Add calendars
- Change settings
- Invite friends
- Choose attendance mandatory
- Add people without account

- Add people with account

Attendant An attendant is a user who chose to not share his private calendar which can perform following use case :

- Choose a time slot

2. Functional requirement

The following functional describe the system's behavior.

2.1Use cases

2.1.1 **V**isitor

2.1.1.1 Registration

Name	Registration
Actor	Visitor
Entry condition	The visitor does not already have an account
Flow of events	1. Visitor navigates to the registration page 2. Visitor fills the registration form 3. Visitor confirms registration 4. Systems add the new user to DB 5. Visitor receives a confirmation message of successful registration
Exit conditions	● Visitor's account is successfully store in DB ● Visitor is redirected to the login page
Exceptions	● Visitor enters already used email ● Visitor enters wrong input ● Visitor leaves compulsory fields empty ● Visitor refreshes the page without submit all inputs

2.1.2 User

2.1.2.1 LogIn

Name	LogIn
Actor	User
Entry condition	● User must be registered and not logged in
Flow of events	1. User navigates to the Login page

	2. User enters his email 3. User enters his password 4. User clicks LogIn button 5. System checks his credentials
Exit conditions	• User logged in and redirected to home page
Exceptions	• User enters a wrong credentials • User refreshes the page without submit all inputs • User forget his password

2.1.2.2 Logout

Name	Logout
Actor	User
Entry condition	• User already logged in
Flow of events	1. User clicks on Logout
Exit conditions	• User redirected to login page
Exceptions	

2.1.2.3 Change settings

Name	Change settings
Actor	User
Entry condition	• User already logged in
Flow of events	1. User clicks “Change Settings” button 2. User updates settings 3. User clicks confirmation button 4. System updates the changed data in DB
Exit conditions	• User redirected to home page
Exceptions	• User enters wrong input • User leaves compulsory fields empty • User refreshes the page without submit all inputs

2.1.2.4 Add calendar

Name	Add calendar
Actor	User
Entry condition	• User already logged in • User has Google Calendar account
Flow of events	1. User clicks on “Change Settings” button 2. User clicks on “linked my calendar” button 3. User enters his Google Calendar credentials

	4. Google Calendar API checks his credential 5. User confirms adding calendar to the system 6. System stored the calendar data in DB
Exit conditions	• User redirected to home page
Exceptions	• User enters wrong credentials • User refreshes the page without confirming

2.1.2.5 Invite friends

Name	Invite friends
Actor	User
Entry condition	• User already logged in
Flow of events	1. User clicks “Invite friends” button 2. User writes friend’s email 3. User clicks invite button 4. System sends Invitation via email
Exit conditions	
Exceptions	• User writes a wrong email • Email already exists in DB

2.1.2.6 Recover password

Name	Recover password
Actor	User
Entry condition	• User already registered to the system
Flow of events	• User clicks on “forgot password” • User enters his email address • System sends a reset link to provided email • User clicks on the link • User enters his new password • User re-enters his new password • User clicks on “OK” button • System updates the user password on the DB
Exit conditions	• User redirected to login page
Exceptions	• User enters nonexistent email on the DB • User re-enters password different than the first

2.1.2.7 Schedule meeting

Name	Schedule meeting
Actor	User
Entry condition	• User already logged in
Flow of events	• User clicks “Schedule a meeting” button • User enters meeting’s title • User enters meeting’s description • User selects attendants from friends list • User enters email of attendant without account • User selects mandatory attendants • User clicks “Create” button • System notifies attendants with account • System sends email to attendant without account with meeting’s informations and link to registration page • System stores meeting informations to the DB
Exit conditions	• User redirected to home page
Exceptions	• User leaves compulsory fields empty • User refreshes the page without submit all inputs

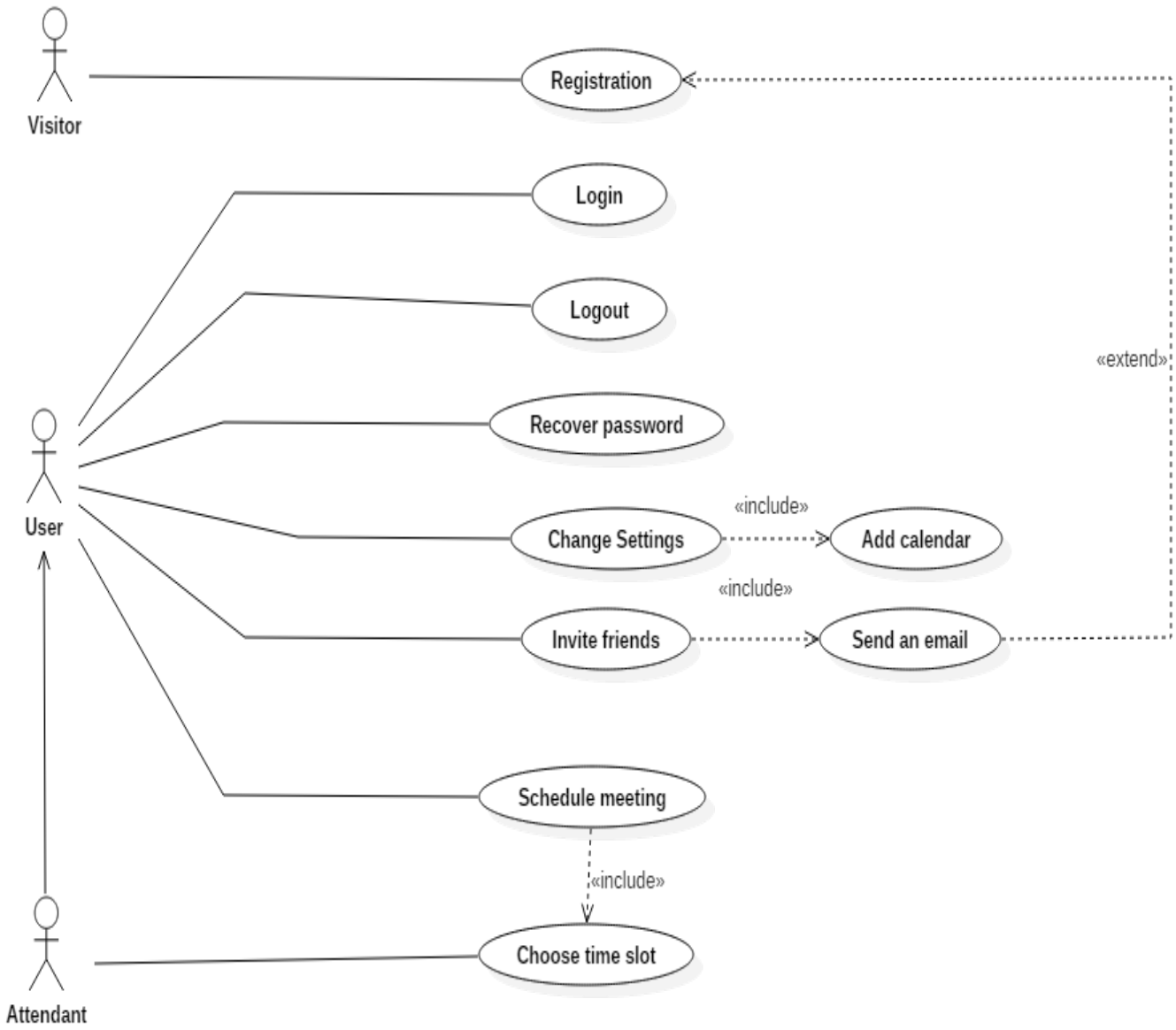
2.2.3 Attendant

2.2.3.1 Choose time slot

Name	Choose time slot
Actor	Attendant
Entry condition	• User already registered to the system • User choosed not to share his personal calendar
Flow of events	1. User selects the concerning meeting 2. User selects the time slots he desired 3. User clicks on “Confirm” button 4. System stores the informations on the DB
Exit conditions	• User redirected to home page
Exceptions	• User refreshes the page without confirming

2.2 Use cases diagram

The following use cases diagram represent graphical representation based on the User Cases mentioned above.



3. Non Functional Requirements

1.1.1 Performance Requirements

Performance of the system should be good enough to provide the user with fast responding software system. Response time should be small enough to enable good user experience.

1.1.2 Software System Attributes

- Availability, the application should be available to handle user's request at all times using any device with an installed web browser.
- Maintainability, The software system provide specific API for enabling future developers with option to add more services or fix bugs in the system.
- Portability, the application could be run on device with any OS that has access to Internet and has a web browser.
- User Interface, the web application should be intuitive so even the nontechnical users can use the system as simply and efficiently as possible.
- Security, application will not take advantage of users personal information and will respect the privacy policy related to information enclosed in his Google calendar account.

4. User stories

User stories are short and simple sentences that contain the features that customer expects to be present in the system. Product owner is responsible for them: team should create a system that meets the customer's expectations using user stories. The customer's requirements aren't equally important; for this reason high, average or low priority is attributed to each of them.

ID	USER STORY	PRIORITY

UserStory 1	As a visitor I want to be able to register to the system ● The user should be able to get in the web application and register in the network	High
UserStory 2	As an user I want to be able to login/logout at the application with my account ● The user should be able to login and logout of the web application whenever he wants to	High
UserStory 3	As an user I want to receive an e-mail with my new password if I forget the first one ● Whenever the user forget this password, the application should provide an way to allow the user recover his password	High
UserStory 4	As an user I want to schedule a meeting or an event ● The user should be able to plan a meeting by saying the its name, time slot and the attendants	High
UserStory 5	As an user I want to change settings ● All the users should have the option to change all the settings, for instance, change password, change profile picture, change email, change nickname and other privacy settings	Average
UserStory 6	As an user I want to receive notification when a meeting is scheduled ● The user should receive an notification by email or by web application whenever someone wants him as an attendant in a meeting	High
UserStory 7	As an user I want to invite my friends to use the application ● The application should provide some way to allow the users to invite other people to join the community by sending an email to user's friends.	Average
UserStory 8	As attendant I want to able to choose right time slot from me without sharing my calendar ● Whenever some user is marked as an attendant in some meeting/event, the application should check on user's calendar for the best time slot, but the privacy of the user must be respected. So no information of the user calendar can be shared with the other users.	High

5. Sub user stories

UserStory 4		
UserStory41	As an user I want to choose attendants in the meeting or event ● The user should be able to choose the attendants of the meeting by writing the email, nickname or real name	High
UserStory42	As an user I want to be able to receive meeting notifications ● Whenever a meeting as some updates like new attendants, new time slots or other details , all the attendants must be informed with an notification	Average
UserStory43	As an user I want to choose attendance mandatory ● The user that schedules a meeting should be able to choose if some user is mandatory to the meeting or not	Average
UserStory44	As an user I want to obtain the best time for the meeting using my and the other participants information in Google Calendar ● The application must check the user's Google Calendar to check if him doesn't have some overlap event in that time slot	High

UserStory 8		
UserStory81	As an user I want to distinguish between preferred meeting times and possible meetings times ● The user should be able to say which time slots are better for him, because sometimes we have availability in some time slot but we prefer another.	Low