MeetMe – Design, Implementation and Testing of a Real Time Meeting Planner for Efficient, Convenient and Fast Scheduling with a Minimum Disclosure of Personal Information and high security standard.

Natascha Kania, Danny Friedrich, Manuel Gollmann,
Ronald Robertson, Malte Kloß
MAM-IM 3
University of Applied Sciences Neu-Ulm
{Natascha.M.b.Kania, Danny.Friedrich, Manuel.Gollmann,
Ronald.Robertson, Malte.T.Kloss}@student.hs-neu-ulm.de

January 2016

Abstract

This project is part of an assignment from the course *Enterprise Application Engineering*, supervised by Prof. Dr. Philipp Brune (Contact email: Philipp.Brune@hs-neu-ulm.de) and part of the master programme *Master of Advanced Management* from students specializing in Information Management at the University of Applied Sciences Neu-Ulm in Germany.

Scheduling meetings in business, study or private environments nowadays requires the coordination of multiple schedules from different people, mostly more than two. The fast technological development in internet based communication enabled computer-mediated collaboration for many people. Web applications featuring these services especially for small handheld devices like smartphones are therefor growing in popularity.

Many people are relying on calendar applications to schedule meetings. Meeting planning applications like Doodle are often used to coordinate meeting dates for bigger groups, but face challenges as when more time constraint loaded schedules get involved, more time for finding common dates is needed.

As an improvement a new kind of meeting planning application named *MeetMe* is developed. This project covers the concept, design, implementation and testing of the application. MeetMe locates the next possible appointment for all participants in an by automatically scanning the calendar of every participant and suggesting the next possible meeting time, within the constraints defined by the organizer. For the meeting organizer this proposes a more efficient, convenient and faster way of scheduling meetings. Besides the usability and convenience aspects, the application is designed with great respect to user privacy and in consideration of high international security standards for web applications, like save password encryption and AES-encryption of all user data.

The project was divided in different phases, beginning with the design phase of the overall concept and an online survey to verify or falsify some of the assumptions made. Based on the knowledge gained from the survey, the development phase started, which again was divided in different successive stages, where the design and functionality of the application was developed, with great respect to security aspects. During the development stages, different usability tests showed potential challenges for users, which were considered for the development of the user interface. As a final test a penetration test is planned to verify the high security standards.

Contents

List of Figures

List of Tables

Statutory of Declaration

We declare that we have developed and written the enclosed paper entirely by ourself and have not used sources or means without declaration in the text. Any thoughts or quotations which were inferred from these sources are clearly marked as such. This paper was submitted in a shortened version to the SCORE Contest 2016.

Neu-Ulm, 28. January 2016

(Place, Date)

(Danny Friedrich)

(Manuel Gollmann)

(Natascha Kania)

(Malte Kloß)

(Ronald Robertson)

1 Introduction

Finding a suitable time for a meeting between multiple parties is a common practical problem in busy and increasingly diversified lives of everyone. Generally the meeting scheduling process usually involves sending and receiving meeting time proposals back and forth between the participants until a suitable time is found. If more people get involved in a meeting the complexity and time consumption of the meeting planning increases exponentially.

When scheduling meetings the primary challenge is to manage the individual perspectives on available time slots as each person involved only nows his or her available time slots which need to be discreetly and adequately communicated with other meeting participants. Accomplishing this requires the laboriously coordination using many different communication tools like by phone or mail.

Nowadays software tools like Microsoft Outlook can help people share calendar information more easily and are often used for scheduling business meetings by determining when a group of people have a common free time slot. This faces a challenge when different communication and calendaring software collide from different users or different organizations. If this happens, the meeting hosts will need to resort to e-mail or telephone discussions to organize their event. Furthermore not all application calendar tools work with other calendar applications and not everyone wants to share his calendar or availability options with everyone in particular when considering different hierarchy levels in companies.

Web applications like Doodle offer solutions to find a suitable time slot for all participants. In this type of scheduling web applications a meeting organizer will suggest several meeting times in which the event may take place. Afterwards the application sends out an invitation to all participants with a link to the voting site. Following the link, the participants can choose the meeting in which they can be present. The scheduling web application keeps track of the results and the meeting organizer is able to establish the time that works best for most of the participants.

Generally, Doodle is a web-based application tool that helps people to solve the organization problem with multiple parties in a simple way without emails and phone calls. The free scheduling service is used worldwide by over 20 million users monthly and is especially popular in Switzerland, Germany, France and the US [1].

Nevertheless, Doodle permits the invitees only to decide the suggested times for the event and if none of them is suitable, again a flood of e-mail communication follows or the meeting has to be fixed via phone calls. The whole process can be very time-consuming, inefficient and inconvenient. Additionally when the number of doodle participants is higher than three or four it can lead to the following two scenarios:
A survey is spread with a few suggested meeting times. Each participant has conflicts with at least one of the proposed times. A flurry of email follows, then either a new poll is distributed with agreed candidate times or else the meeting time is just settled by email. To avoid the first scenario, a second scenario would be possible where the survey is spread with many time slots. It is more difficult to fill out, but it would be possible to find an acceptable meeting time in a single round.

Although the large offering of scheduling applications are available, there doesn't seem to be an optimal solution that would allow easy meeting scheduling across multiple parties of different organizations, different calendaring applications and with respect to a high user privacy.

This project covers the concept, design, implementation and testing of the application. The meeting planner locates the next possible appointment for all participants in an automated process, which for the meeting organizer will be a more efficient, convenient and faster way. Besides the usability and convenience aspects, the application is designed with great respect to user privacy and in consideration of high international security standards.

2 Related Work

Besides *Doodle*, there are some similar web-based applications like *Kulibri* or *Couchkiller* for meeting scheduling, which work in the same way [2]. An approach to automate the process is *Timebridge*. The organizer chooses different possible times in his calendar and then send it to the participants he would like to invite. The participants can choose on which dates they are available and which date option has the highest priority. After every participant replied to the request, Timebridge checks the answers of everybody automatically, selects a date which is suitable for all participants and sends an invitation to everybody. It also send automatically reminders to the participants who did not reply within a required time [3]. This application already simplifies the scheduling process for the user and makes it more convenient, but the organizer still has to check his or her calendar for availability to make some suggestions and the participants have to check their calendars and to reply, making the process not fully automatic.

Doodle is often criticized for the lack of privacy and secrecy by some companies and therefore it is forbidden to use for these companies employees [4]. Doodle is collecting data with different tools like Google Analytics for analyses and for user-based advertisement on their page [5]. By voting every Doodle user has to publish at which time he or she is available and thereby providing a very sensitive information. To limit this information out flow, Doodle offers *hidden pools* where the user has more privacy and only the organizer can see the availability of each participants still leaving him or her to see time preferences of colleagues. Herlea et al. made a comparison in 2001 of different privacy protocol approaches and introduced a custom-made protocol approach [6] to face this problem. Kellermann's approach to tackle this problem in his work [7, 8] was by defining requirements for privacy enhanced event scheduling and derived *Dudle*, a web based event scheduling tool, which works in a similar way as Doodle, but with great respect to privacy aspects. An other approach of a privacy enhanced alternative is *DFNTerminplaner* from DFN, the German national research and education network. It works in the same way as Doodle, but it does not collect any date of the user and does not show any advertisements. Also for scheduling the organizer has to define an expiration date where all the data from the scheduling will be deleted from the server of DFN [9].

Bilogrevic et al. introduced in their paper three different protocols for the privately computing of user availability, using well-known cryptosystems by implementing and testing them on a mobile device. However the user still has to select at which time slots he or she is free [10].

Marcinowski designed an automatic algorithm for event scheduling. The algorithm is based on *Constraint Logic Programming*. The input of that algorithm is the users' schedules and preferences, like specification of the time, which suits the best and which should be avoided. The output of the algorithm are several suggestions, from which the user can define how many he would like to have, for a meeting time which suits best to all participants. The schedule can be sourced from an online calendar like Google Calendar, unfortunately this option is not implemented in the library. Therefore there is no automatic synchronization of the online calendar [11].

The above mentioned approaches shows that there are already some tools with concentration on privacy and security, but all of them lack an fully automatic scheduling process. Also there is no official working web-based application so far. The approach introduced in this paper is the design, implementation and testing of a real time meeting planer for efficient convenient and fast scheduling with a minimum disclosure of personal information and high security standards.

3 Project Management

Before starting the project, the team decided to use *scrum* [12] as a management and control framework to focus on agile software development. Therefore several key data needed to be defined like a vision of the web application, the roles in the scrum team and the *Definition of Done* (DOD), which are described in this chapter.

3.1 Vision

The vision is a meeting planer, which locates the next possible appointment for all participants in an automated process, which for the meeting organizer will be a more efficient, convenient and faster way of scheduling meetings. Besides the usability and convenience aspects, the application is designed with great respect to user privacy and in consideration of high international security standards.

3.2 Sprints

The development team started its very first scrum project so in this case there is no historical velocity to plan the sprints. For this reason the velocity is relatively low which is shown in table 3.2. The team had several lectures every week which are related to the project. For this reason the sprint length is one week and the sprint planning is every Wednesday, which marks the beginning of a new sprint. This day was chosen on purpose so that during the sprint planning, the team still has all the results of the last sprint in mind and the team can directly begin with the next sprint without having a weekend in between.

Table 1: Sprint Key Data

Sprint Length	7 days
Sprint Planning	Wednesday
Velocity	1.5

3.3 Roles

In the scrum team you have the three roles developer team, scrum master and product owner [12]. The product owners of the project are Natasha Kania and the sponsor from the Score contest Michal Young. Ms Kania understands the business and market requirements, has a clear vision of the product and professional experience in it security and rights issues. Her main responsibilities in the team are:

- Build and manage the product backlog
- Ensure that the team understands the work items in the product backlog
- Give the team clear guidance on which features to deliver next [13]

The members of the developer team are Danny Friedrich, Ronald Robertson and Malte Kloß. The developer team drives the plan for each sprint and forecasts how much work they believe they can complete over the iteration. The team members have differing skill sets and support each other in the delivery of work to ensure a successful sprint completion.

The scrum master role is assigned to Manuel Gollmann. He coaches the team, the product owner, and the business on the scrum process. He schedules the needed human and logistical resources for sprint planning, stand-up, sprint review, and the sprint retrospective. He also takes care that the development team does not get disturbed from external, environmental factors during the sprint.

3.4 Definition of Done

The Definition of Done is a list of criteria that the software will need to meet for each product backlog item to be considered as done [12]. For this project a backlog item is done if it is:

1. implemented in all previous increments,
2. tested by the developer team,
3. documented for the user,
4. deployed on the test web server.

An item is finished if it is integrated in the complete web application. During the development the team rents a web server in which the application can be deployed. The item is done, if everything is tested on the deployed web application by all developers. This decision was made, because the application interacts with several users and therefore the team needs a web server.

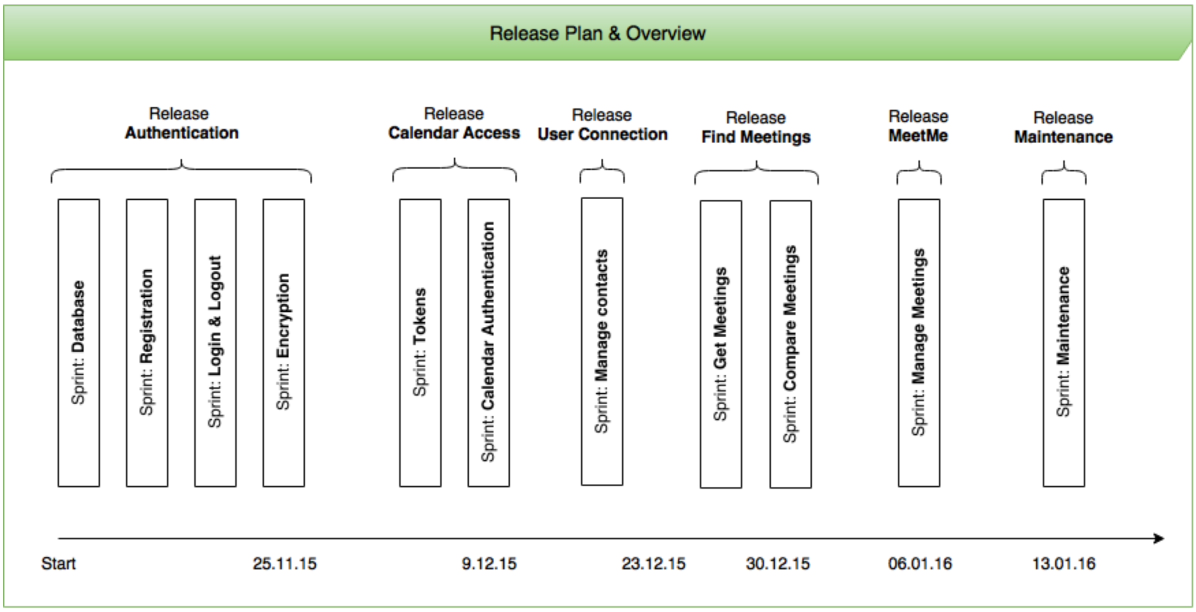


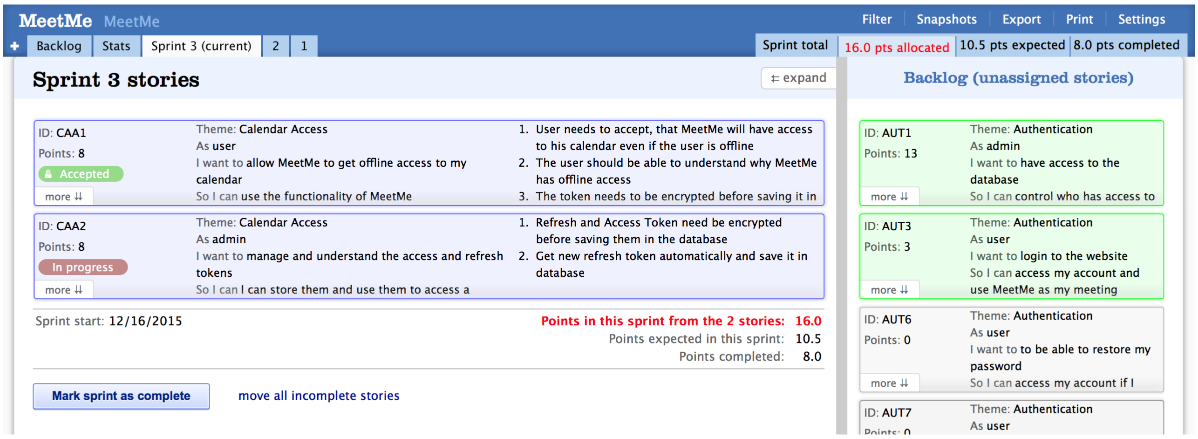
Figure 1: Release Plan and Log

3.5 Release Management

The kick off meeting of the project was in the middle of october. The first sprint planning took place on the 25th October 2015. During development the product owner has an idea of the next two to three sprints. If something unforeseen comes up, which is common in agile software development, the upcoming sprints can be changed after every sprint. As shown in figure 1 the team released 6 big packages until now. The first release contains the basic database and encryption structure of the web application as well as the login and registration process. With the release calendar access it is possible to access a user's calendar. The "Find Meetings" - Release contains the functionality to get meetings from several user's calendars and includes the comparison algorithm to find a free meeting for all participants. With the next release it is possible to invite and delete other contacts. The first release in 2016 makes it possible to use full functionality of MeetMe as described in the vision. The last release contains maintenance functionality like deleting inactive or unregistered users from the database. All the releases are described in more detail in the following chapters.

3.6 Tools

To set up and manage the product backlog, the team uses the web application *easyBacklog* [14]. This tool is for now free to use and makes it possible to manage and plan the sprints by providing aids to streamline the agile process such as printing user story cards, exporting the backlog to an excel file or creating reports with a burn down and burn up chart as well as a velocity log. A backlog item contains a theme, item ID, user story, acceptance criteria, comments, effort points and the expected days to fulfill the item. The user sets the velocity of the team and guesses the effort of an item by giving points. The application will then calculate the expected days. A sprint can be easily started by dragging an item from the product backlog directly and dropping it into the current sprint. A screenshot of the tool is shown in figure 2.



Sprint total | 16.0 pts allocated | 10.5 pts expected | 8.0 pts completed

Sprint 3 stories

ID: CAA1
Points: 8
Accepted
Theme: Calendar Access
As user
I want to allow MeetMe to get offline access to my calendar
So I can use the functionality of MeetMe

1. User needs to accept, that MeetMe will have access to his calendar even if the user is offline
2. The user should be able to understand why MeetMe has offline access
3. The token needs to be encrypted before saving it in

ID: CAA2
Points: 8
In progress
Theme: Calendar Access
As admin
I want to manage and understand the access and refresh tokens
So I can I can store them and use them to access a

1. Refresh and Access Token need be encrypted before saving them in the database
2. Get new refresh token automatically and save it in database

Sprint start: 12/16/2015

Points in this sprint from the 2 stories: 16.0
Points expected in this sprint: 10.5
Points completed: 8.0

Backlog (unassigned stories)

ID: AUT1
Points: 13
Theme: Authentication
As admin
I want to have access to the database
So I can control who has access to

ID: AUT3
Points: 3
Theme: Authentication
As user
I want to login to the website
So I can access my account and use MeetMe as my meeting

ID: AUT6
Points: 0
Theme: Authentication
As user
I want to to be able to restore my password
So I can access my account if I

ID: AUT7
Points: 0
Theme: Authentication
As user

Figure 2: Screenshot of the tool Easy Backlog

4 Requirements

4.1 General Requirments

In the first place the web application should meet the requirements which are described in the project scope of the SCORE contest [15]. The project describes the targeted application to be *a better meeting planner* and thereby offer the same functionality of other web applications such as *Doodle*, but with more efficiency in terms of the duration of meeting planning. In addition the web application should meet the following requirements:

- It should draw from Google Calendar.
- It should work for users who do not use the same calendaring application.
- It should respect privacy and security preferences.

4.2 Use Case Requirements

The *MeetMe Overview* diagram in figure 3 shows the range of functions MeetMe will offer.

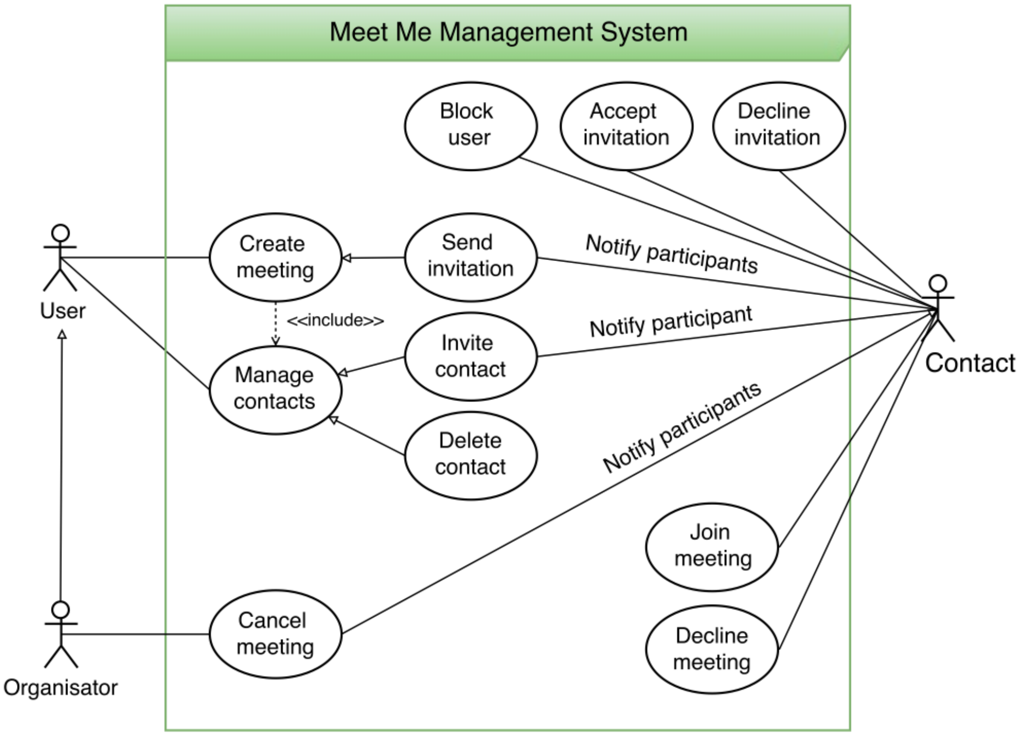


Figure 3: MeetMe - Use Case Diagram

Before using MeetMe every user needs to register on the system first. The user then can access the authorized area of MeetMe by using his or her e-mail and password in the login form. The use case diagram for the authorization process is not shown in this document. After the registration every user can manage his contacts and has the possibility to invite contacts, who will then be contacted by e-mail. The invitee can confirm or decline the invitation. For simplicity reasons, this is not shown in the figure. Any contact can also be deleted afterwards.

A MeetMe user can take an organizational role, as well as the role of a participant. By creating a meeting the organizer has to specify the meeting name, location, time slot, starting time, duration of the meeting and the participants. A meeting organizer has also the possibility to make a *Quick Search* in which the next possible meeting will be found by just adding the participants and a time slot. By automatically searching the calendar of the participants and matching the available times in the background, MeetMe locates the next possible appointment for all participants within the constraints of the individual calendars. Afterwards an invitation to the participants is sent. Despite having a free time slot in his calendar, an invitee has the possibility to accept or decline the meeting. In this case the organizer will be informed via email. The organizer has also the possibility to cancel a meeting in which a notification will be sent to the participants.

4.3 User Stories

In the project MeetMe every team member can access the different types of requirements of the MeetMe project and his or her individual tasks in *easyBacklog*. It contains *User Stories* of different sizes and levels of detail in a form of an arranged list. With regard to the project MeetMe the requirements are split in six themes *Start Page, Authentication, Calendar Access, Meetings, Contacts* and *Maintenance*. In every single requirement an user story is documented with acceptance criteria, which is defined by the product owner, followed with a priority grouping and an estimation of story points.

The *Product Backlog* is divided in *Backlog Items* with different priority groupings like *must have, would have, won‘t have* and *nice to have*. The tool will be explained in more detail in the chapter 3 Project Management.

The *must have* are defined and finished with regard to the project MeetMe. All the user stories are attached in the final document. In scope of this document, two user stories are presented in more detail. For instance the *start page* - user story tells that "As user I want to read more about MeetMe. So I can understand the functionality of MeetMe.". This leads to the acceptance criteria, that the start page of MeetMe needs to provide information by a text and a video. This criteria is defined as a *must have* by the product owner.

A second example from the development process is the *authentification* - user story "As admin I want to have access to the database. So I can control who has access to the secure areas.". The acceptance criteria are that an admin can delete a user, can differentiate between an activated and a pending account and can see how many users are registered. However, an admin is not allowed to see any personal data of a user. This user story is a *must have* with an high estimation of story points (13).

4.4 German Law And Privacy

Defining the privacy of the users as a high priority for MeetMe, the application is designed with respect to the regulations of the German *Federal Data Protection Act (FDPA)*. Due to this the collecting of personal data of a person is generally forbidden. However, there are some cases in which it is allowed to save personal data of a user. In case of MeetMe a user gives the application permission to use his data, by agreeing to the terms and conditions of the MeetMe service when registering an account. Nevertheless with respect to the *FDPA*, MeetMe is only allowed to collect the data, if the data is required for the fulfillment of the proposed service. For this reason only the name, surname, e-mail address and the calendar data of the user are collected and saved. Considering this, MeetMe fully meets the regulations of the German *FDPA* [16].

5 Design

This chapter describes the architecture model of the application.

5.1 Overview

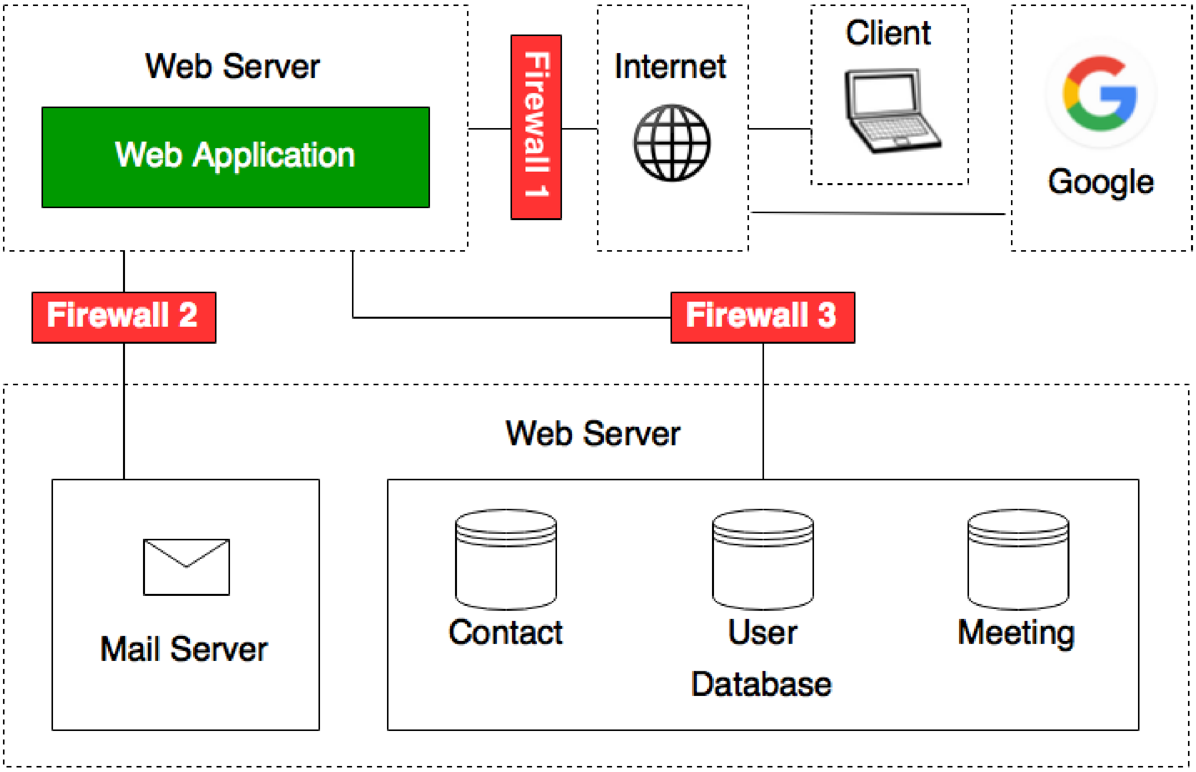


Figure 4: Architecture overview

Figure 4 shows an overview of the architecture of the web application. MeetMe is deployed on a web server and has access to a second web server to access data from the database and to send emails via the mail server. It also communicates with the Google servers to get access to the calendars of the registered users of MeetMe. A user can use the web application via a internet browser on his client computer. In the following chapters, the individual functions and the single design elements will be explained in more detail.

5.2 Patterns

5.2.1 Model-View-Controller

Meet Me uses the *Model-View-Controller* (MVC) pattern [17], which separates user interface logic (front end) from business logic (back end). This decision was made to have a clear separation of the modeling of the domain, the presentation, and the actions based on user input. The development team was distributed in the same pattern with Danny Friedrich being responsible for the back end and Ronald Robertson and Malte Kloß responsible for the front end.

5.2.2 Data Access Objects

In addition the application uses the *Data Access Object* (DAO) [18] pattern to abstract away the details of persistence in an application. The reason for this is that the domain logic speaks to a DAO layer instead of directly to the database. In this DAO layer the application uses entity managers to communicate with the underlying persistence services.

5.3 Security

MeetMe saves and uses personal data of the registered users to offer the described calendar functionality, even if the user is offline. For this reason the web application needs to meet modern IT security standards and needs to be conform to the German *Federal Data Protection Act*. Security and privacy are two of the main focuses on the development process. More detailed information about the conformity to the German law is described in chapter 4.4. Even though the priority of the web application is safety (**Defense in Depth**), no online application can be 100 % secure. For this reason the team defined actions in case the application gets attacked to reduce the potential damage for a user and the product owner. The actions are described in chapter 6.4.8 Worst Case Scenarios.

5.3.1 ESAPI Framework

It is recommended to use only one security framework in an web application [19]. The team decided to use the *Enterprise Security API* framework which is used for authentication and authorization, session management (session handling, session ID and session fixation), output escaping and password hashing [20]. The decision was made because the framework is open source, supported by the OWASP community, and widely used and tested by an active community. Another reason is to avoid *Security by Obscurity*, with the open source framework the development team can check the source code and knows exactly what happens in the framework.

5.3.2 Web Application Firewall

In addition, a *web application firewall* (WAF), which is a server plugin, is set up to configure a set of rules to the HTTP conversation to cover common attacks such as *cross-site scripting* (XSS) and *SQL injection*. This firewall needs to be maintained but can identify and block many attacks [19].

5.3.3 Transport Layer Secutiy

The web application uses the cryptographic protocols *Transport Layer Security* (TLS), which is the successor of *Secure Sockets Layer* (SSL), to secure all communication between the server and the web browser of the user [19]. This decision was made to provide privacy and data integrity between two communicating JavaServer Pages within the application. In addition, MeetMe uses a special response header which specifies *HTTP Strict Transport Security* (HSTS). This is to prevent any communications from being sent over HTTP to the specified domain in the browser. Instead, all communication will be sent over HTTPS [19].

5.3.4 Encryption

The web application uses different encryption standards when saving the password of a user and all other data into the database. Regarding the password encryption the *National Institute of Standards and Technology* (NIST) recommends to hash passwords at least with the cryptographic checksum procedure *SHA-256* [21]. In addition a random string, which is called *salt* [22], should be attached to the password and the key needs to be stretched with the *Password-Based Key Derivation Function 2* (PBKDF2) [23]. Regarding saving data in the database, the web application uses the *Advanced Encryption Standard* (AES) [24] to encrypt and save all data with a unique, user specific key, which is combined with a server key, into the database.

5.4 Calendar Interfaces

5.4.1 Calendar Provider

In scope of this work the web application is compatible with the email service *google calendar*. The software architecture is designed to allow other calendar providers like for instance *outlook365* to be easily implemented by just adding a new DAO. Subsequently the *Google Application Programming Interfaces* (Google APIs) are described, which are necessary to provide the calendar functionality.

5.4.2 Google OAuth2 API

Before getting all functionality of a user's calendar the web application needs to have permanent access to it. Therefore the usage of the *Google OAuth API* is necessary for authorization and authentication. Basically the application obtains OAuth 2.0 credentials from the *Google Developers Console* and can then receive a refresh and an access token to get permanent access to the owner's calendar. This permission, even if the user is offline, needs to be confirmed by the user [25].

5.4.3 Google Calendar API

Once the application has access to the user's calendar, the *Google Calendar API* is needed to display, create and modify calendar events. The application can then get the calendar data of every user and compare their calendar events to find a suitable meeting [26].

5.5 Tools

5.5.1 Unified Modeling Language

The software architecture is described with the *Unified Modeling Language* (UML). The application is separated into the main functions Authentication, Calendar, Meetings, Contacts and Maintenance. During a sprint the team creates detailed *use case and class diagrams* before implementing the actual code. For creating the UML diagrams the team uses the free online tool *Draw.io* which is available on *www.draw.io* [27].

6 Implementation

6.1 Technology

MeetMe is a *Java EE* business application, uses *JavaServer Faces* (JSF) in the back end and *HTML, Bootstrap, JavaScript and jQuery* in the front end. It is developed with *Eclipse Mars*. This decision was made to develop a component based application so that the java source code can be the base for mobile applications like for instance an android app. An additional reason was to use the strength of the different team members and thus build the teams *Front End* and *Back End* to have a very clear programming interface between the client and the web server. The web application is hosted on a test web server and runs with a *Tomcat 7* [28].

6.2 Graphical User Interface

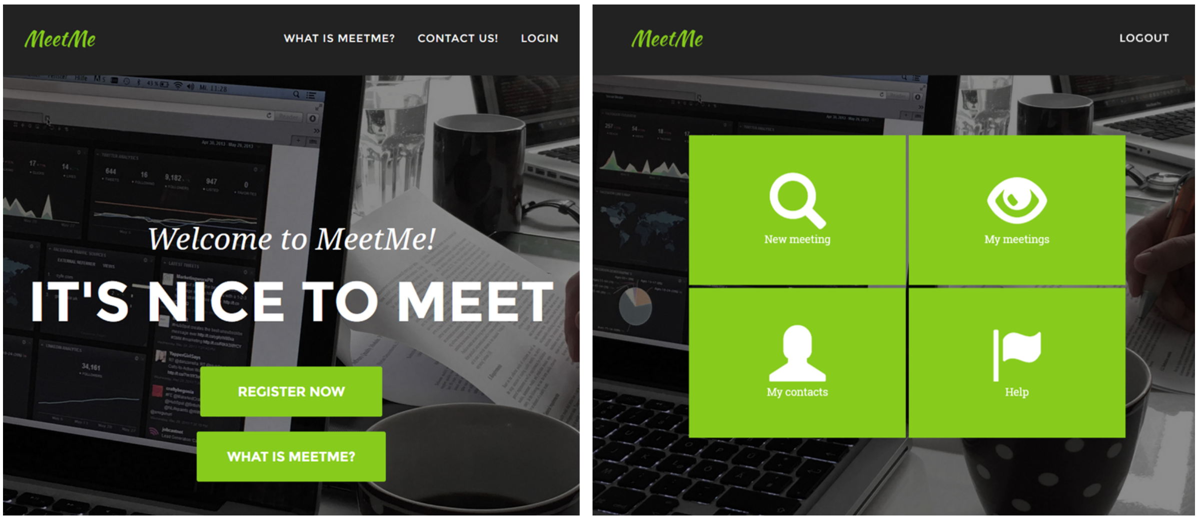


Figure 5: Screenshots MeetMe Landing Page and Main Menu

The web design of MeetMe can be seen in figure 5 and was developed using the front end framework Bootstrap which provides components through *HTML, CSS* and *JavaScript* files which allow the design of a responsive web-application. A responsive web design consists of a flexible grid-based layout, flexible multimedia components and *CSS3 Mediaqueries* [29]. Each site of the MeetMe web application should work equally well on smartphone screens as it would on a bigger laptop screens without an impairment of usability. In terms of color composition the MeetMe design appears in a mixture of green and white with some dark accents. This color scheme was decided based on the popularity of the color green [30] and its distinction to other scheduling service providers like outlook or Doodle who both use blue as their main color.

6.2.1 Navigation

Visiting MeetMe for the first time a new user will find the landing page which was designed using the bootstrap template *Agency*. On the landing page a user may get information about

MeetMe, contact the MeetMe Team about further questions or get registered on MeetMe. After registration a user will find the home menu which consists of four 2D blocks of flat equally sized tiles like found in the Microsoft Design Language *Metro*. The tiles represent the four functions *New Meeting*, *My Meeting*, *My Contacts* and *Help/Settings*.

6.2.2 Create New Meeting

When clicking on *New Meeting* a user can plan a new meeting by providing firstly the meeting details such as name and location of meeting. Afterwards the host specifies the start and end date, starting time and duration of the meeting. For the selection of the start and end date a date picker will appear and the date can be chosen from a calendar widget. The starting time for the meeting is grouped in daytimes ranging from morning to late evening whereas the duration is to be typed in an input field in hours and minutes. Lastly the participants for the meeting are selected, given they are already registered on MeetMe and were added to the contacts previously. After these dates are specified the user clicks *Find Meeting* and MeetMe will start looking for suitable meeting slots for all selected participants including the host. During this process the user will see a wait window and if MeetMe found a suitable meeting slot the host will see the details of the found date and can then send an invitation to all participants. The host will then be notified by mail if the participants accepted the invitation or declined it.

6.2.3 My Meetings

Next to the New Meeting screen a user has the option to administer his or her meetings by selecting the My Meetings option in the main menu. Meetings are grouped in the three categories *Planned Meetings*, *Pending Meetings* and *Past Meetings* which show the current timely status of a meeting. Each meeting has a details page showing detailed information and offers the function to delete the meeting after which MeetMe will inform the participants about the cancellation.

6.2.4 My Contacts

Under the section *My Contacts* a user can view his contacts and add new contacts, view existing contacts or delete contacts. Adding a new contact is done by entering the e-mail address. The invited user will receive an invitation email and can either accept the invitation by clicking the confirmation link for which he or she will need to be registered on MeetMe. The invited user in this case requires a google account the use the web application.

6.3 Calendar

6.3.1 Calendar Access

The complete authorization (token management) and access to a user's calendar is managed by Google. The very first time, the user wants to create a meeting, he will get redirected to Google via a link and will be asked to give MeetMe permission to access his calendar. Once the user confirms the permission request, the web application gets an authorization code from the Google OAuth 2 API. This authorization code gets exchanged for a refresh token and an access token. The refresh token gets encrypted with the unique, user specific

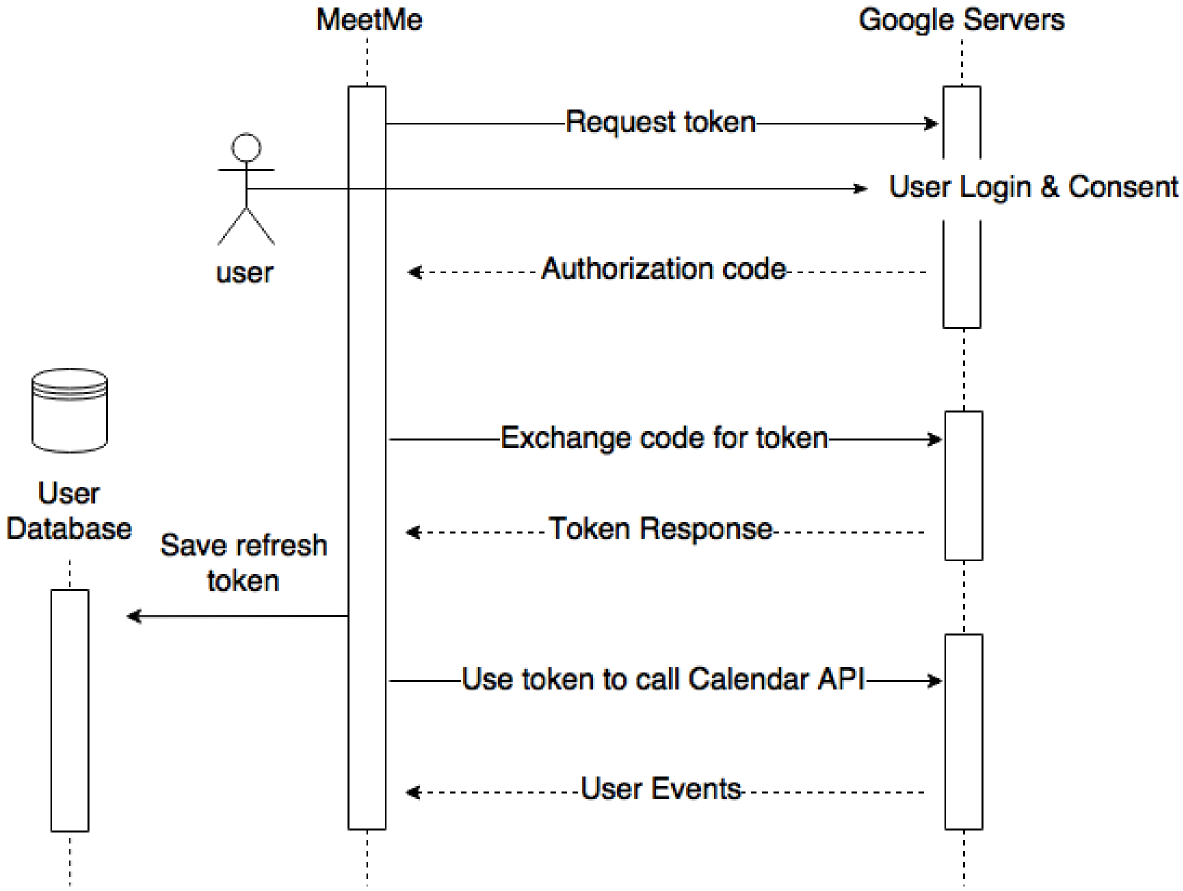


Figure 6: Token Management

key and is saved in the database. The access token will then be send to the Google Calendar API to modify the user's calendar events. However, this access token can only be used for one request and expires after 4000 seconds. The application can get a new access token by using the refresh token, which is saved in the database [31]. This process is shown in figure 6.

6.3.2 Comparison Algorithm

Once the application has access to every participants calendar, it is able to compare the participant's events to find a suitable meeting for everyone. The application has the following parameters to find a suitable meeting:

- Participants
- Earliest and latest possible day
- Earliest and latest time
- Duration of the meeting

The application first of all creates a temporary array in which the relevant data will be saved. The single indexes, values and datatypes of the array are shown in table 2. MeetMe then starts to check the first possible date and saves all participants who are free and all who are occupied in the array. In the last index of the array the amount of free participants gets saved. The application now checks the next possible time slot in 10 minute intervals and saves the relevant data in the pending array until the latest date is reached.

Table 2: Pending array to compare events

Description	Participants Free	Participants Occupied	Participants - Counter
Datatype	String	String	Integer
Value	""	""	1

The application now checks the pending array. If the amount of free participants is higher than "1", a suitable meeting is found, else no possible meeting was found in the given time slot. MeetMe then suggests the earliest, possible time slot with the most free participants. The visualized process "MeetMe - Compare Algorithm" is attached at the end of this paper.

6.4 Security

6.4.1 Presentation Layer Access Control

The application uses a folder structure in Eclipse in which the different access roles are defined. MeetMe differentiates between 2 different roles:

- Authorized Access
- Public Access

All public JavaServer Pages can be used and accessed by any user who visits the website of MeetMe. All other JavaServer Pages can only be accessed if the user authorizes himself via email and password.

6.4.2 Authentication and Authorization

If the user wants to use the functionality of MeetMe, he needs to register first. The web application will check if the email already exists, generate a registration link and send the link to the user's email address. By clicking the registration link, the user verifies his email address. The user then needs to login with his email address and his password to get access to the authorized area of the application.

6.4.3 Encryption

All data in the database is AES - encrypted with a 60 character strong key, hereinafter named *server key*. This key is combined with the email address of the user to personalize it to every user. For instance, instead of saving the name *Daniel Miller* in clear text in the database, the data gets encrypted and is saved as shown in table 3. The data can only be decrypted by the web application with the *server key*, so only the user himself can decrypt the data by using his email as a key. The password itself is never saved in clear text anywhere. Instead, the password gets hashed with the cryptographic checksum procedure SHA-256. Before the hashing procedure, a salt with the length of 64 bit is attached to the password and then gets stretched with PBKDF2 in 20.000 rounds.

Table 3: Encryption of data in mySQL database

Daniel	Miller	daniel.example@email.com
eMVxf258zagZhafbT734	AhQt38alhha1015nFKKh	av3afkfplymwifasjgONNV99h

6.4.4 Session Management

Beside using the TLS/SSL protocols and HSTS, the application also focuses on Session Fixation and the Session ID. Once the user is logged in, the application creates a Session ID by using the ESAPI framework. The app creates a random reference which is saved in the Session. Every time the application redirects to an internal *JavaServer Page* (JSP), the reference gets passed and checked by the redirected JSP. The reason for this is to prevent *Cross Site Request Forgery* vulnerabilities. In addition, before creating a new HTTP-Session, the current HTTP session gets invalidated. Another action to secure communications within the application is to secure the session-cookies and to configure a session timeout of 30 minutes [19].

6.4.5 Input Validation

Before using data input in the back end, the data needs to be validated. For the data validation the web application uses whitelists instead of blacklists. The advantage is that MeetMe is better maintainable if security requirements change in the future. An example for input validation with a whitelist is that an input for the first name can only contain capital and small letters. All inputs need to be validated, such as link parameters like the registration ID, to protect the application first of all against SQL injections. MeetMe uses *Bean Validation*, which is part of Java EE, to implement the whitelists [19].

6.4.6 Output Escaping

To handle the output of the application it uses the encoders of the ESAPI framework. It processes content that is about to be HTML or JavaScript output so that dangerous characters are made safe. The encoders are used to prevent *Cross-site Scripting* (XSS) vulnerabilities [19].

6.4.7 Limitations And Privacy

The application uses the following rules when searching for a meeting:

1. The app does not access any data of an user's event when searching for a time slot. It only checks whether a time slot is free/empty in the calendar.

2. A user can only start 5 search requests with the same participant(s) within 15 minutes. This is to avoid that someone can check single time slots of a contact systematically.

6.4.8 Worst Case Scenarios

Even though the priority of the web application is safety, no web application can be 100 % secure. [19] For this reason several actions are defined in case the application gets attacked to reduce the potential damage for a user and the product owner:

1. The Client Secret of the Google Developer Account needs to be rotated, so an attacker won't have access to the calendar of the users because the refresh token expires.
2. All users will get notified to change their Google and MeetMe password.
3. The AES server key need to be changed and all data in the database need to be updated.
4. All log files need to be checked to find, as well as remove, the vulnerabilities.

6.4.9 Contact All Users

Once a user clicked the registration link and activates his account, his email and user ID will be saved in the so named *email database*. All emails are AES encrypted. If the app wants to send an email to all users, it needs to decrypt the emails with the *server key* first. MeetMe only sends an email to all users in case that a worst case scenario occurs.

6.5 Contact Management

When adding a new contact the application checks wether the user is already in the contact database. If the user does not exist in the database, an invitation will be send in form of an email to the new contact. The invitee can then confirm, decline or block the invitation. If the invitee accepts the invitation the application saves given name, last name and e-mail address and encrypts the data with a combination of the server key and the email address of the initiator. If a user wants to delete another contact, the complete record of this contact gets deleted from the contact database.

Every invitation email contains three links which the potential new contact can click. The links contain an invitation ID and they end with a *confirm, decline* or *block* -parameter. If a registered user confirms the invitation, the app will connect both users by writing a *connected* into the contact database in which the invitation IDs are saved. If the user accepts the invitation, but is not yet registered, the application will redirect the user to the registration form and add the invitationID to the link. The new user does the registration like every other user, but the invitation ID will as well be saved in the database during the registration. When the user activates his account he will already have the initiator contact in his contact list. If the invitee wants to block the initiator, the application saves *blocked* into the contact database of the initiator. In this case it is not possible for the initiator to add the contact again, because the email already exists in his contact database. If the invitee declines the invitation, the complete request of the initiator will be deleted in the database so that a new invitation can be requested by the initiator. The visualized process "MeetMe - Contact Management" is attached at the end of this paper.

6.6 Meeting Management

The meeting management works in principle as the contact management. The user types in the parameters to find a meeting and the application searches for a suitable meeting. Once a meeting is found all participants get invited to the meeting via email. However, the email notification can be turned off in the settings. The application saves the meeting to the *meeting database* which contains a meeting ID, all data input of the initiator and the potential meeting participants. In this stadium all invitees have the status *pending*. Every meeting invitation can be confirmed or declined by clicking one of the attached links.The links contain the meeting ID including the end parameters *confirm* or *decline*. Depending on the invitees action, the status of this user will be updated in the database by saving *confirm* or *decline*in the status index of the *meeting database*. If a participant confirmed a meeting, it is shown in the meeting list of that account. The visualized process "MeetMe - Meeting Management" is attached at the end of this paper.

6.7 Maintenance

In the first place the maintenance function of the application is to remove old data from all databases and to update the *server key* for better security. The application

1. deletes all pending users, if they do not click the registration link within 24 hours
2. deletes all meetings, which are older than 31 days
3. changes the server key and updates / encrypts all data every 72 hours

7 Testing

At the beginning of the project the team defined quality aspects which were seen as important for MeetMe, and identified methods how to measure them. This section describes how the quality aspect usability for every user group was tested and how it will be improved continuously. Also it is shown how the acceptance of MeetMe among the target audience was tested. Lastly it will give an outlook on tests which will be made further in further development stages such as a security test of the web application ensuring to ensure the safety of its users and therefore securing the continual usage of MeetMe using a penetration test.

7.1 Acceptance Test - Online Survey

7.1.1 Execution

An online survey was conducted to analyze the participant's acceptance and reactions to MeetMe. The survey targeted all user groups and should deliver suggestions from the participants, as well as challenges in the understanding of the concept and the intended usage of the application. The target group are users of online calendars who often plan appointments with multiple participants in their business or private environment. The survey was conducted for two weeks and was distributed via Facebook and e-mail and consisted of open and closed questions. The closed questions are designed to create data that is easily quantifiable and furthermore, the information gained by closed-ended questions allows the MeetMe Team to categorize respondents into groups based on the options they have selected. According to this open questions provide reach qualitative data. They provide the MeetMe Team with an opportunity to gain insight on the opinions on a topic they are not familiar with.

The first section of the survey includes demographical information like gender, age and occupation. The second part contained a video, which explains the idea and advantages of MeetMe. Afterwards questions about the MeetMe concept are asked and a mock-up of the *Create New Meeting* page is presented and the survey participant is asked questions about his or her impression of the mock-up. In the end four general questions complete the survey.

7.1.2 Results

The following section presents the results of the online survey, starting with the demographical information of the participants.

Demographical Information

The result of the online survey showed that 79 participants completed the online survey. They consisted of 69,6 % female and 30,4 % male participants. The average age is 26 and the biggest participant group are students with 81 % followed by 13 % employees.

Concept of MeetMe

The second part contains a video, which explains the idea and advantages of MeetMe. Afterwards four questions and their results about the MeetMe concept are represented. On the closed question *Do you think MeetMe would be useful for you?*, 76 % of the participants response with *yes* and 24 % with *no*. In the following are some examples of the reasons from the participants why they think MeetMe would not be useful for them:

- I have my dates in my paper based calendar.
- Outlook and its calendar function (with invitations) work fine for me.
- It does not look like it is adding any new functionality compared to Doodle or Outlook. The fact that it selects the timeslot best suitable for all participants automatically is not a big plus as I can do the same in Outlook (just slightly more manually). The privacy aspect is invalid as well, Outlook offers the same.

The second question in this part is an open question about *What do you like most on MeetMe?*. Example answers of the participants were:

- Timesaving
- Seems simple to me
- That I don't have to remind my friends to check their calendar.
- The automated search and synchronization of availabilities does not sound bad.
- No discussions about appointments. MeetMe does everything for you. Very easy.

The third question *How would you describe MeetMe in one word?*, is a closed question with five possible answers. The most participant described MeetMe as *easy*, in figure 7 you can see all results of the question.

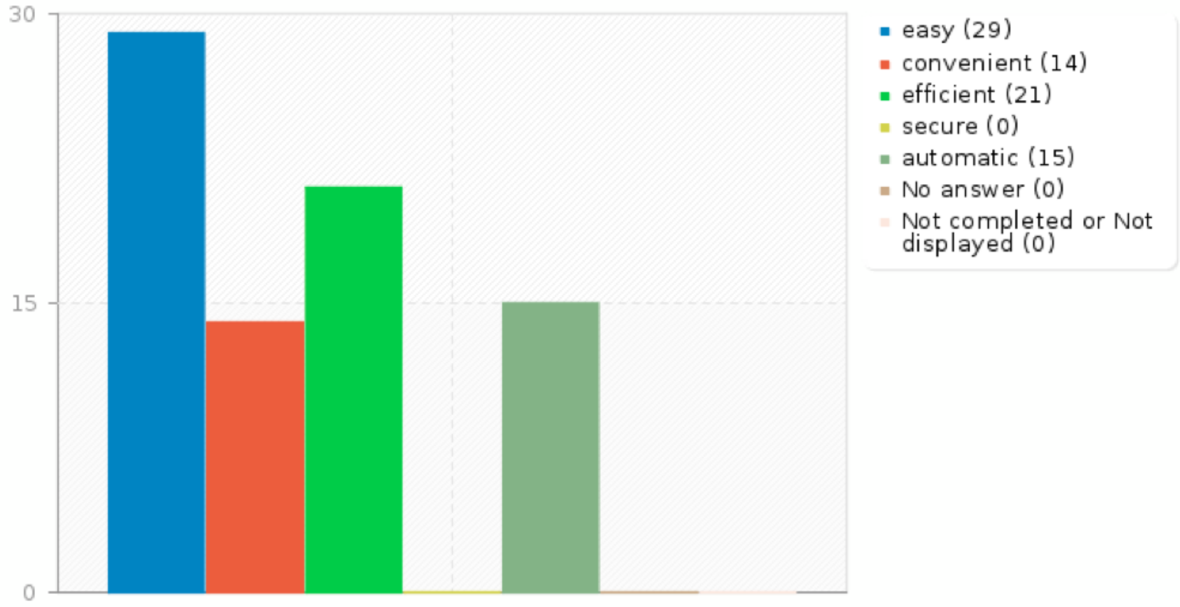


Figure 7: Survey result *How would you describe MeetMe in one word?*

The last question about the concept of MeetMe was *What do you think could be improved on MeetMe to suit your personal planning needs?* It is asked as an open question and 33 % of the participants answered it. Below the answers can be seen.

- Especially for after work activities it would be great if one could schedule a reoccurring event (I know Outlook does that) that sends reminder mails automatically one or two days prior and asks who will attend and in addition allows everyone to see who attends. For example: Soccer every thursday, 25 possible participants. I don't want to send out a request every week and I would like people to automatically check who will show up.
- Maybe include some kind of chat.
- Clear interface, easy comprehensibility, high usability, self-explainable usage, implementation as plug-in in existing mail accounts (outlook, gmail, etc).

Creating a new meeting

Then a Mock-up *Creating a new meeting*, which is seen in figure 8 was presented and six questions were asked about it. The basic question is *How useful would you rate the following options for your personal usage of MeetMe?* The participants had the possibility to rate the different options from one to five, with 1 (not useful), 3 (nice to have) and 5 (very useful).

The **Option to select multiple time slots** is rated as very useful for the participants with 21 %, 16 % of the participants rated this option as nice to have and for 6 % of them that it is not useful.

The **Option to select optional participants** is rated as useful for the most of the participants with 31 %, for 15 % it is very useful, 8 % of the participants rated this option as a nice to have and for 8 % of them that it is not useful.

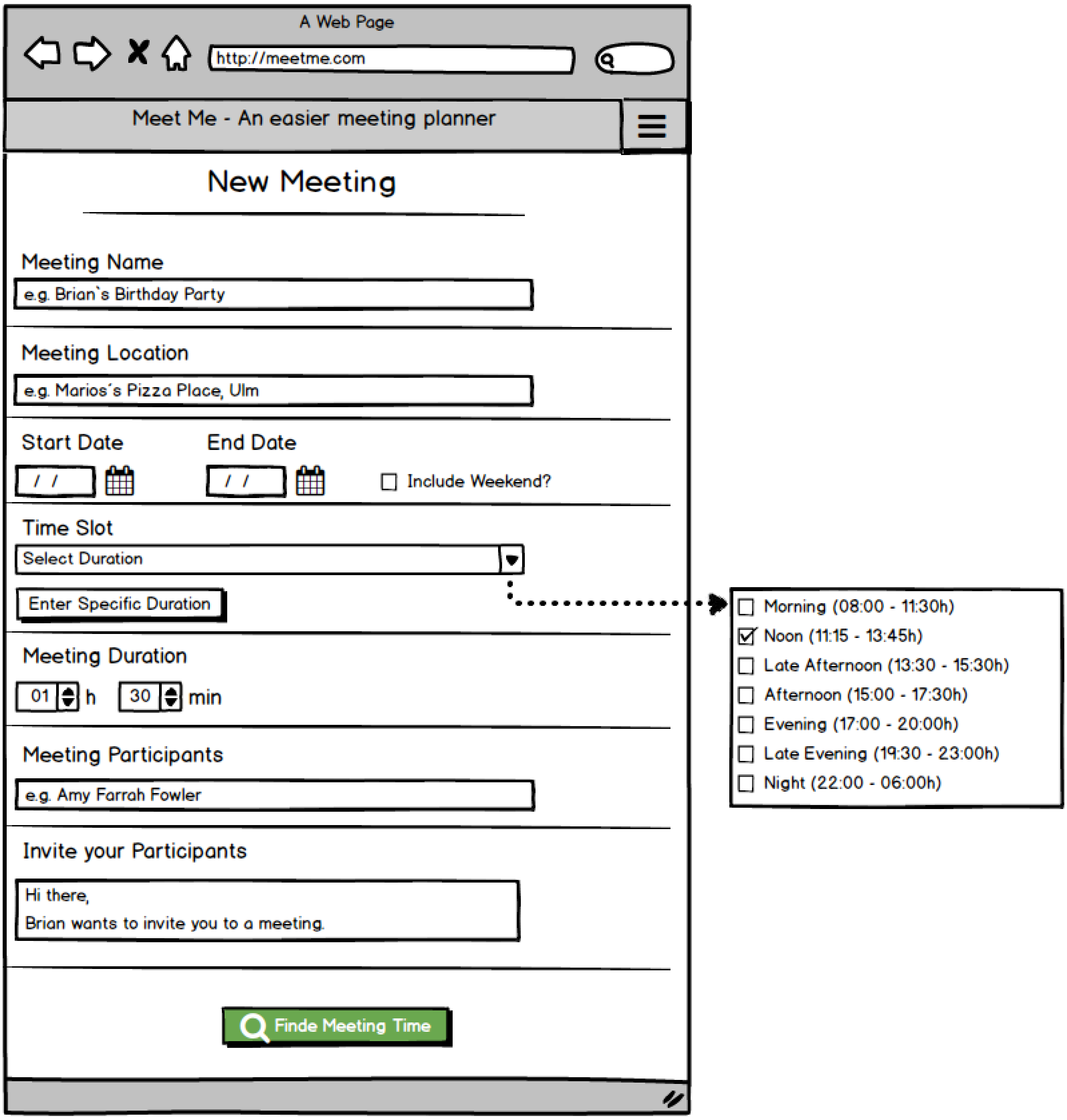
The **Option to add pictures to a Meeting** is rated as nice to have with 24 %, for 21 % of the participants it is useful and 12 % rated this option as not useful.

The **Option to receive multiple time suggestions** is rated for 16 % as very useful, for 29 % it is useful, 12 % of the participants rated this option as nice to have and for 4 % of them it was not seen useful.

The **Option to receive a push notification** is rated for 26 % as very useful, for 24 % it is useful, 9 % of the participants rated this option as nice to have and for 4 % of them it was not seen useful.

As a last question in this section the participants are asked *Do you miss anything or do you have any further suggestions for improvement?*, which answered 11,4 %. Some of the answers can be seen below.

- Maybe a priority ranking.
- Description to find the meeting location (route).
- Things to bring with what are necessary for a meeting.
- What is the "Include weekend" checkbox about? I think it would be more intuitive to just enter start and end date -> thus, the tool is applicable in business as well as private contexts.



New Meeting

Meeting Name
e.g. Brian`s Birthday Party

Meeting Location
e.g. Marios´s Pizza Place, Ulm

Start Date / /
End Date / /
□ Include Weekend?

Time Slot
Select Duration
Enter Specific Duration

- □ Morning (08:00 - 11:30h)
- ☑ Noon (11:15 - 13:45h)
- □ Late Afternoon (13:30 - 15:30h)
- □ Afternoon (15:00 - 17:30h)
- □ Evening (17:00 - 20:00h)
- □ Late Evening (19:30 - 23:00h)
- □ Night (22:00 - 06:00h)

Meeting Duration
01 h 30 min

Meeting Participants
e.g. Amy Farrah Fowler

Invite your Participants
Hi there,
Brian wants to invite you to a meeting.

Figure 8: Moock-up *Creating a new meeting*

Further coments

In the end four general questions end this online survey. The first closed question is *Do you trust MeetMe?* From the participants 45.6 % of the participants would trust MeetMe, 21.5 % do not.

To the question *Would you give MeetMe access to your calendar?* 43 % of the participants answered with yes and 24 % with no.

The third closed question *Which calendaring service do you use?* The results shows that with 15.2 % the most participants use *Outlook.com* and only 9 % use the *Google calendar*.

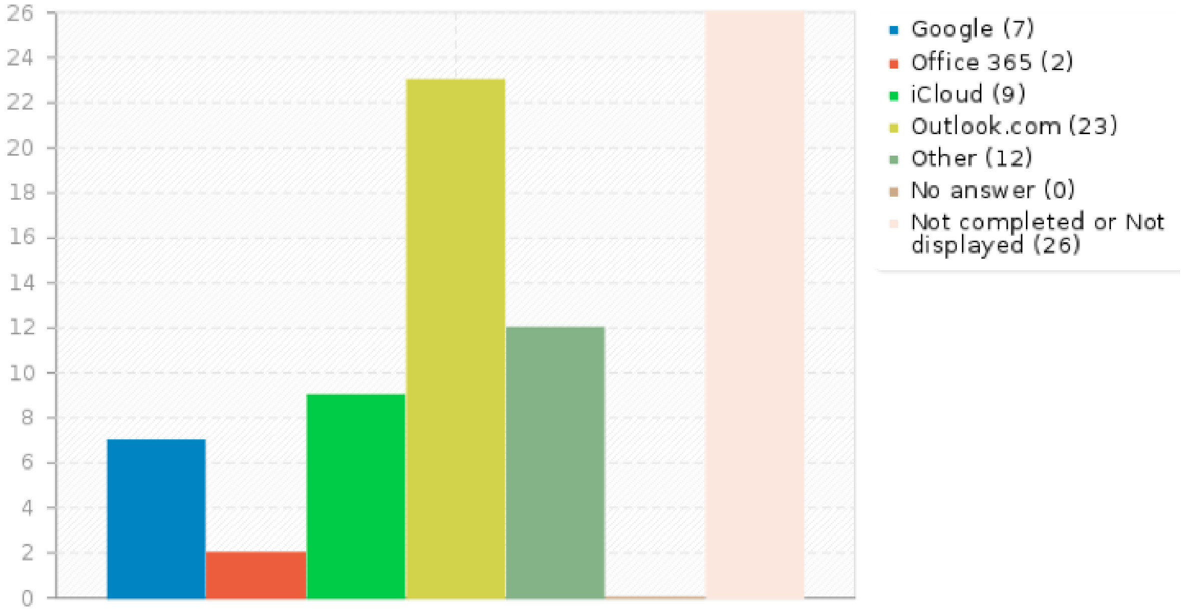


Figure 9: Survey result *Which calendaring service do you use?*

Nevertheless the Google calendar is first implemented in this project, because it is a requirement of the project sponsor from the SCORE project. In figure 9 the complete results of the question are seen.

The very last open question *Do you have any further suggestions/ comments or something else you would like to let us know?*, gives the survey participant in the end a last opportunity for a final feedback, which only 6.3 % of the participants used. Here some of the final comments of the participants are given.

- You just asked, whether I trust MeetMe, how should I? Except the advertising video I have no information about it. Advertising is not very trustworthy I think. Maybe some more ideas have to be developed to make the user feel safe.
- It would be really interesting, if MeetMe could check all the calendars of one person, e. g. I am using three services, because work has their own, university has their own, etc. If MeetMe could check it all, it would be really great.
- I'm not sure about trusting MeetMe and providing access to my calendar yet. Further information about the security would be helpful.
- Is it for free?

The online survey also shows, that it is difficult to explain the concept in a short video, so additional information should be provided to the user to understand the concept and to gain the trust in the security and privacy of MeetMe which can be derived from the last comments of the survey.

7.2 Backend Test

FindBugs is an Eclipse Plugin which is used during the development. The plugin detects defects by using static analysis to look for more than 200 bug patterns. The decision to use *FindBugs* was made because the tool is open source and is used by many major companies and financial institutions. For repeatable unit tests the team uses *JUnit*, which can be downloaded on *junit.org*.

7.3 Usability Test

Usability testing was seen as a key process and determinant for the creation of the user interface to transform the concept of MeetMe into an interface which seamlessly translates the proposed functionality. For both iterations of the usability testing an example from the German *Usability Professionals Association* (UPA) [32] was used and modified to feature the key aspects of MeetMe. The template gives recommendations about the execution of an usability test and is compliant with most of the standards described in the *ISO/IEC CD 25066.2 Common Industry Format for Usability Evaluation Report*.

7.3.1 1. Iteration

In the first run of the usability test, the test persons were given a quick story representing a typical scenario for a registered MeetMe user with eight consecutive tasks. The execution of the tasks was evaluated in a spreadsheet with symbols showing if a tester completed the task *without any problems*, *with some problems*, *with major problems* or if the tester did *not complete* the task or *completed it incorrectly*. The testing was done on the laptop of one of the developers which was monitored using *TeamViewer* and the screen activities were recorded for later analysis.

Table 4: Evluation of Test Results

Symbol	Meaning
✓	Task solved without any problems
O	Task solved with some problems
!	Task solved with major problems
X	Task not solved or completed falsely

Before testing the testers were instructed to *think out loud* [33] during the testing and the comments were written down by the development team who was present during the test. The test persons were assured that the usability test is merely an evaluation of the software and not of the test persons. Main objective of the usability test was identifying major challenges in the functions of MeetMe before MeetMe will be introduced publicly. Therefore four of the test persons were chosen based on their experience with web applications with two being professors from this subject as their expertise would deliver valuable insights and concrete recommendations for later implementations in the graphical user interface. They were also aware of the thinking out loud method and commented openly without restrictions. For testers 5 and 6 the *Hawthorne Effect* [34] would need to be considered and it could be assumed that being family members from one of the team members they e.g. were less inclined to abort a task when frustrated.

Table 5: Test Persons

No.	Gender	Approx. Age	Occupation	Knows MeetMe?
1	Male	20-30	Master Student, Information Management	Yes
2	Female	20-30	Master Student, Information Management	Yes
3	Female	40-50	Professor, Information Management	Yes
4	Male	40-50	Professor, Information Management	Yes
5	Female	40-50	Self-Employed, Vendor	No
6	Male	10-20	Pupil	No

Table 7 shows an overview of the test persons who were mostly familiar with the concept of MeetMe. Testers 5 and 6 were not familiar with the concept and had no previous experience in web application engineering and user interface design. The results of these two persons gave insights about the accessibility of MeetMe by different types of users across ages and professions [35].

Table 6: Results of Usability Test

Participant	1	2	3	4	5	6
Task 1: Add Contact	O	✓	✓	O	O	✓
Task 2: Edit Contact	✓	✓	✓	O	✓	✓
Task 3: Plan Meeting	O	✓	✓	✓	✓	✓
Task 4: Meeting Details and Participants	O	O	✓	✓	✓	O
Task 5: Define Time slot	✓	!	!	✓	X	X
Task 6: Search for Meeting	✓	O	✓	✓	✓	✓
Task 7: Edit Meeting	O	✓	✓	✓	✓	✓
Task 8: Logout	✓	✓	✓	✓	✓	✓

Considering the background of test persons 1, 2, 3 and 4 there were a few „real“ problems with the handling of the web application but the testers simulated problems to show improvement potential. Looking at Task 5 it comes clear that most problems occurred when the testers should define a time slot with testers 2 and 3 having major problems and testers 5 and 6 completing the task falsely. Reasons for this were wording of the sections *Selection of Time Slot* and *Time Slot* sounding too similar, the definition of daytimes and the option *Include Weekend* not being prominent on the page. Another problem area was the section *Add Contact* for which users first had to open *My Contacts* to find this option.

The usability test showed many improvement possibilities for the *Create New Meeting* page which will be included in the final version of MeetMe. For example the duration of a meeting will be changed from a text input field to a slider. Also the invitation text field at the end will be removed as participants who cannot find a suitable meeting time would write an invitation text which will not be used later. The arrangement of the input fields and the wording of the headers were adjusted based on preferences of the testers.

7.3.2 2. Iteration

After remodelling the user interface, based on the results of the previous test a second usability test was in order to see if the changes resulted in a better outcome in the task performance. Also a new landing page was added which included different information about MeetMe and a link to the login and registration function and a task was added to evaluate this new screen. The formulation of test questions was changed from explicit instructions, to more general tasks which should show how intuitive MeetMe works without a clear guidance.

For the second iteration, the test team had access to the usability lab of the university of applied sciences Neu-Ulm which provided a stationary eye-tracking device to monitor eye movement and focus of the participants with gaze charts and heat maps. This was especially useful to see if the arrangement of objects on the new landing page would lead the user to find the required information and stepping stones on other pages of the web-application.

The evaluation was done like described in subsection 7.3.1 and this time exclusively with students with an information management background like shown in the table below.

Table 7: Test Persons

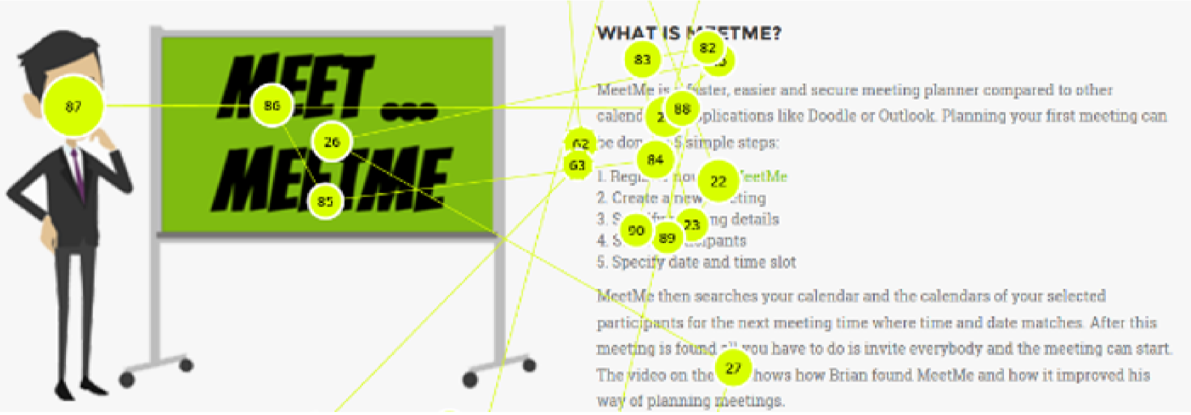
No.	Gender	Approx. Age	Occupation	Knows MeetMe?
1	Female	20-30	Master Student, Information Management	Yes
2	Female	20-30	Master Student, Information Management	Yes
3	Male	20-30	Master Student, Information Management	Yes
4	Male	20-30	Master Student, Information Management	Yes
5	Female	20-30	Master Student, Information Management	Yes

In table 8 you can see the results of the second iteration of the usability test.

Table 8: Results of 2. Usability Test

Participant	1	2	3	4	5
Task 1: Key Features and Video	X	X	X	O	X
Task 2: Log-In	✓	✓	✓	✓	✓
Task 3: Check Contact	✓	✓	✓	✓	✓
Task 4: Create a Meeting	O	✓	✓	✓	X
Task 5: Log-Out Mobile	✓	✓	✓	✓	✓

The table shows that participants had fewer problems creating a meeting which included defining a timeslot. Only one participant was mislead by a help text which will need to be better phrased. The task involving the landing page was only solved by one participant who was irritated about the location of the video. Other participants criticized the location and amount of text involved to discribe MeetMe.



WHAT IS M[illegible]ETME?

MeetMe is [illegible] faster, easier and secure meeting planner compared to other calend[illegible] [illegible]plications like Doodle or Outlook. Planning your first meeting can be do[illegible] [illegible] 5 simple steps:

1. Regi[illegible] no[illegible] [illegible]eetMe
2. Create a new [illegible]eting
3. S[illegible] [illegible]ng details
4. S[illegible] [illegible]ipants
5. Specify date and time slot

MeetMe then searches your calendar and the calendars of your selected participants for the next meeting time where time and date matches. After this meeting is found [illegible] you have to do is invite everybody and the meeting can start. The video on the [illegible] shows how Brian found MeetMe and how it improved his way of planning meetings.

Figure 10: Gaze Chart - Landing Page

This Gaze Chart from *Task 1 Key Features and Video* shows how testers even though noticing the picture, they did not suspect the video to be under this link. The paragraph below the five steps description was almost not looked at even though this was a testing situation. Therefore this text needs to be shortened, and the video needs to be embedded so users can see a frame and a play button for the video.

The Heat Map of *Task 4 Create a Meeting* gives insight on where the focus of a meeting creator is during the process, which is the left side where the title of each form can be found. Dropdown, slider and input fields which expand over the screen to the right side are unhandily as they feel unnecessarily long. In future they should be reduced to half of the container width of this page.

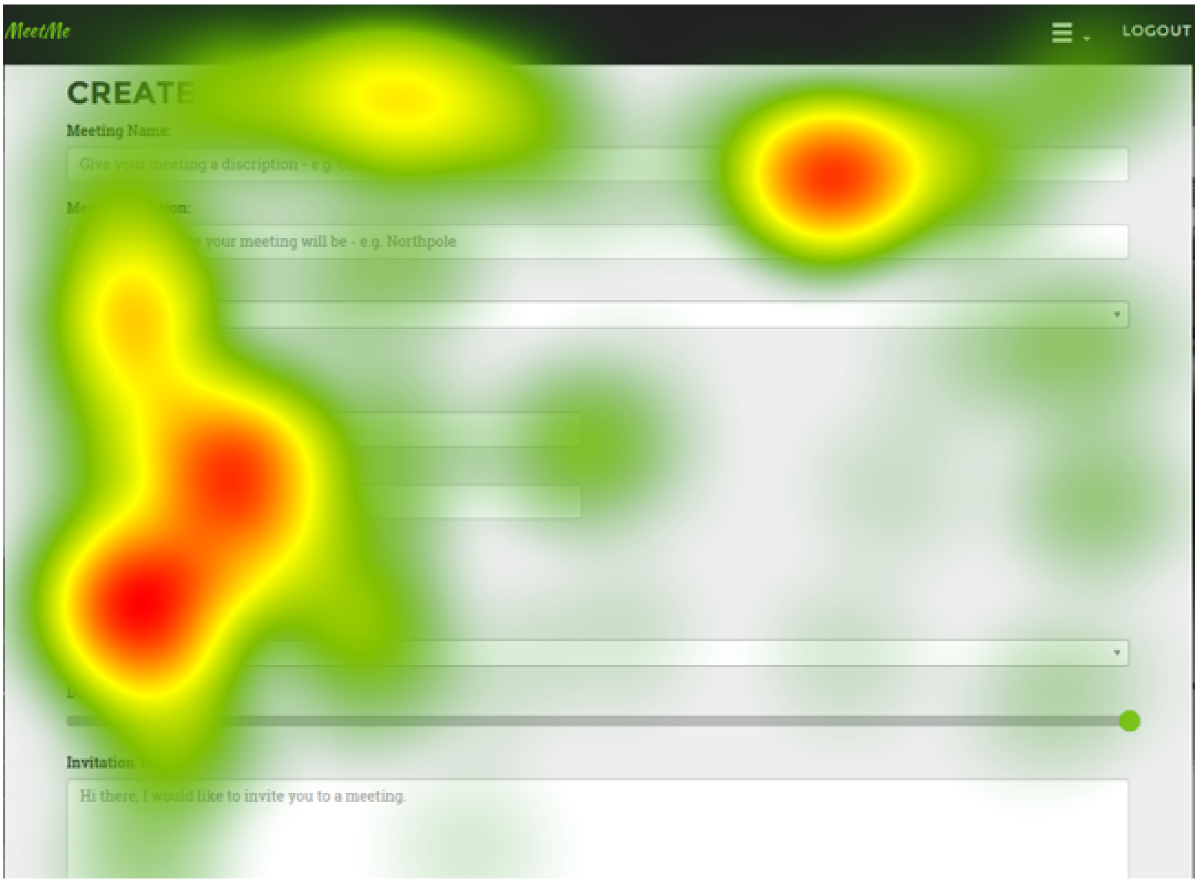

MeetMe
LOGOUT
CREATE
Meeting Name:
Give your meeting a discription - e.g. G
your meeting will be - e.g. Northpole
Invitation
Hi there, I would like to invite you to a meeting.

Figure 11: Heat Map - Create New Meeting

7.4 Upcoming Tests

As the above mentioned tests were already conducted there are more tests planned in the next stages of the development. Before a first release of MeetMe will be made public a penetration test from the german IT company Atos is planned to test the implemented security measures in a real attack situation. After the complete web-application is finished in development, a last usability test will be done to see if the changes made at the landing page were successful and to test latency or other backend related problems which would result in a lack of usability.

8 Evaluation

8.1 Lessons Learned

One of the lessons learned is the importance to secure an application against vulnerabilities, because an attack can easily lead to image damage of a company or in the worst case to bankrupt a cooperation especially a small start-up company. Injections vulnerabilities can be prevented easily but they are still in the OWASP Top 10 list of vulnerabilities in 2013 [36]. In addition, when working with user data, the laws of the specific countries where the web application is published need to be considered.

Regarding scrum, the team needs to work together for several weeks to make a good guess on how much effort a task needs in scrum which was not expected to be a challenge. The next project will work more smoothly once the team has more experience.

Another lesson learned is that the concept for new users of MeetMe due to its high complexity needs a solid explanation. To make the user interface more intuitively it needs to be adapted after each iteration of a usability test to improve the interface logic and accessibility for different user groups. The use of correct wording and phrasing in the communication between the web design and user is paramount for using MeetMe.

8.2 Limitations

One limitation of this project was due to the time and all authors being information management master students, there was in genreal a big focus on students and local university personnell. This resulted in most of the respondents of the online survey being students and university personnell.

The usability tests were conducted mostly with fellow information management master students, who were able to give valuable and professional feedback and even though it was tried to give general feedback, the overall results however do not represent a typical user but more technically experienced users. This and the fact that the usability testers were exclusively peers and relatives, make the results less objective and critical for general statements. Considering the available version of MeetMe for the different iterations of the usability tests it needs to be considered that usually a cut-down version with no backend functionality was available for testing.

8.3 Further Research

The application MeetMe is not only thought for private use or use, but also for business scenarios. To get insights into these scenarios, it would be necessary to expand the research by broadening the scope of the acceptance survey test and include a higher diversity of business users as well as users without any IT background.

One of the proclaimed key features of MeetMe is the swiftness in which a meeting can be found, so further research will be done to improve this feature. For this reason the possibilities of creating a native application for Android and iOS smartphones will be evaluated to incorporate push-notifications in order to synchronize the communication between host and participant of an event as some e-mails remain unchecked.

Also an expansion of the offered calendar support is planned which can only be realized if the same level of security in the calendar access is reached as the current security status of MeetMe predefines.

As MeetMe is planned to be implemented as an enterprise application, certain IT governmental aspects need to be considered and researched. From an organizational view, it needs to be tested which companies are ready to implement such a web-application, as it is assumable that companies with steeper hierarchies tend to be critical towards an a calendaring app which lets an employee check the calendar of his or her supervisor. Looking at the reverse scenario, a supervisor who checks the calendars of his employees may be viewed at as a threat by personnel law. Considering these aspects, qualitative interviews with employees from different companies and in different hierarchy levels will be conducted and evaluated to gather insight on this subject.

8.4 Conclusion

In this paper a concept of MeetMe, a web application of an improved meeting planner with an automated scheduling algorithm is introduced. The concept is designed in consideration of the Federal Data Protection Act, so only personal data which is really necessary for the application to work are collected and the security standards are defined. To test the concept an online survey as an acceptance test was conducted as well as other tests to continuously improve MeetMe.

The authentication is already fully implemented, the calendar and the meeting process are partially implemented and the contact as well as the maintain process are not implemented yet. To assess the accessibility and usage for every type of potential user, several usability tests were made and the results were considered when remodelling the gui.

References

[1] doodlemyke, “Doodle crosses the 20M user mark.” http://en.blog.doodle.com/2014/01/29/doodle-crosses-the-20m-user-mark/, 29.01.2014 [14.01.2016].

[2] D. Albrecht, “Doodle: Die besten Alternativen zur Termin-Findung.” http://praxistipps.chip.de/doodle-die-besten-alternativen-zur-termin-findung_30828, 2015 [4.01.2016].

[3] Timebridge, “Timebridge.” www.timebridge.com, 2016 [4.01.2016].

[4] T. Albers-Heinemann, “Doodle-Alternativen mit Blick auf Datenschutz.” https://www.medienpaedagogik-praxis.de/2013/01/08/dfn-kalender-als-doodle-alternative/, 2013 [4.01.2016].

[5] Doodle, “Datenschutzrichtlinie.” https://doodle.com/de/datenschutzrichtlinie, 12.03.2015 [4.01.2016].

[6] T. Herlea, J. Claessens, B. Preneel, G. Neven, F. Piessens, and B. De Decker, “On securely scheduling a meeting,” in *Trusted Information*, pp. 183–198, Springer, 2001.

[7] B. Kellermann, *Open Research Questions of Privacy-Enhanced Event Scheduling*, vol. 6555 of *Lecture Notes in Computer Science*, book section 2, pp. 9–19. Springer Berlin Heidelberg, 2011.

[8] B. Kellermann, *Privacy-Enhanced Web-Based Event Scheduling with Majority Agreement*, vol. 354 of *IFIP Advances in Information and Communication Technology*, book section 19, pp. 235–246. Springer Berlin Heidelberg, 2011.

[9] J. Pattloch, “DFNTerminplaner - Einfach und schnell Termine abstimmen.” https://www.dfn.de/dienstleistungen/dfnterminplaner/, 7.09.2015 [4.01.2016].

[10] I. Bilogrevic, M. Jadliwala, J.-P. Hubaux, I. Aad, and V. Niemi, “Privacy-preserving activity scheduling on mobile devices,” pp. 261–272, ACM, 2011.

[11] B. Marcinowski, “Flexible meeting scheduling,” tech. rep., University of Edinburgh, 2014.

[12] R. Dräther, H. Koschek, and C. Sahling, *Scrum kurz & gut*. O’Reilly Germany, 2013.

[13] D. Radigan, “A brief introduction to scrum.” https://www.atlassian.com/agile/scrum, 2016 [14.01.2016].

[14] Easybacklog, “Easybacklog.” https://easybacklog.com, 2016 [15.01.2016].

[15] M. Young, “Score project: Meetme.” http://score-contest.org/2016/projects/meetme.php, 2015 [15.01.2016].

[16] “Bundesdatenschutzgesetz,” 20.12.1990 2009.

[17] A. Leff and J. T. Rayfield, “Web-application development using the model view controller design pattern,” in *Enterprise Distributed Object Computing Conference, 2001. EDOC’01. Proceedings. Fifth IEEE International*, pp. 118–127, IEEE, 2001.

[18] S. Haines, “Java Reference Guide.” http://www.informit.com/guides/content.aspx?g=java&seqNum=1 14.03.2003 [15.01.2016].

[19] D. Schadow, *Java-Web-Security - Sichere Webanwendungen mit Java entwickeln*. dpunkt.verlag, 2014.

[20] C. Schmidt, “Category:OWASP Enterprise Security API.” https://www.owasp.org/index.php/Category: OWASP_Enterprise_Security_API, 16.12.2015 [14.01.2016].

[21] Q. Dang, *Recommendation for applications using approved hash algorithms*. US Department of Commerce, National Institute of Standards and Technology, 2012.

[22] Defuse Security, “Salted Password Hashing - Doing it Right.” https://crackstation.net/hashing-security.htm#salt, 6.08.2014 [15.01.2016].

[23] M. S. Turan, E. Barker, W. Burr, and L. Chen, “Recommendation for password-based key derivation,” *NIST special publication*, vol. 800, p. 132, 2010.

[24] NIST-FIPS, “Announcing the advanced encryption standard (aes),” *Federal Information Processing Standards Publication*, vol. 197, pp. 1–51, 2001.

[25] Google Developers, “Using OAuth 2.0 to Access Google APIs.” https://developers.google.com/identity/protocols/OAuth2, 4.01.2016 [14.01.2016].

[26] Google Developers, “Get Started with the Calendar API.” https://developers.google.com/google-apps/calendar/overview, 20.11.2015 [14.01.2016].

[27] draw.io, “draw.io.” https://www.draw.io, 2016 [14.01.2016].

[28] B. Prof. Dr. Schiefer, “Entwicklung von Web-Anwendungen auf JAVA EE Basis.” http://www.fh-kl.de/ schiefer/lectures/download/javaee/javaee_01_intro.pdf, [14.01.2016].

[29] E. Marcotte, *Responsive web design*. No. 4, Editions Eyrolles, 2011.

[30] M. Wäger, *Grafik und Gestaltung: das umfassende Handbuch ; [perfekte Drucksachen erstellen: Form, Farbe, Schrift und Bild ; alle Prinzipien und Layouttechniken sicher im Griff ; DTP-Grundlagen: Auflösung, Farbmanagement, Druckverfahren u.v.m.]*. Galileo Design, Galileo Press, 2010.

[31] Google Developers, “Oauth 2.0 playground.” https://developers.google.com/oauthplayground/, 2016 [14.01.2016].

[32] C. Riedmann, L. Daske, and R. Molich, *Usability Testbericht, Beispiel*. German UPA e.V., Arbeitskreis Qualitätsstandards, 2014.

[33] J. S. Dumas and J. Redish, *A practical guide to usability testing*. Intellect Books, 1999.

[34] J. G. Adair, "The Hawthorne effect: A reconsideration of the methodological artifact.," *Journal of applied psychology*, vol. 69, no. 2, p. 334, 1984.

[35] W. O. Galitz, *The essential guide to user interface design: an introduction to GUI design principles and techniques*. John Wiley & Sons, 2007.

[36] OWASP, "Top 10 2013-top 10." https://www.owasp.org/index.php/Top_10_2013-Top_10, 21.08.2015 [15.01.2016].