Project: My Available Time (MAT) Part one – Publishing of MAT by the mailing and sharing by Google+

WEB Application with possibility porting to the mobile applications

Front-end Server

Beck-end Server

* Web service for maintaining MATT (My Available Time Table)
* Web service for working with Social Networks (Google+)

1. Accounting on MAT
   1. Name
   2. Family Name
   3. Email address (MVP Assumption: MAT’s user should have account on Gmail)
   4. Password
   5. Login (MVP Assumption: Login only through Gmail account for SSO (Single Sign On) initial implementation. Otherwise the user should manipulate with several passwords and the MAT account should have several passwords that is not so acceptable)
2. Creation of new MATT
   1. Parameters:
      1. Name
      2. Number of days
      3. Start date (using calendar widget) /ending date will be added automatically
      4. Starting hour
      5. Ending hour
      6. Time slot (1 hour or half an hour)
      7. Password (if the MATT is a private)
   2. Getting available time from Google
      1. Creating MAT calendar in Google by sync all Google calendars
   3. Updating available time by demarking free time slots
      1. Updating MAT calendar in Google
3. Getting of existing MATT by name
   1. Password for private tables
4. Editing of existing MATT (all parameters except name may be edited)
5. Deleting of existing MATT
6. Sharing via Google+
   1. Creating circle
   2. Sending mail with URL of MATT into circle of contacts
   3. iFrame creation ???
7. Invitation of the booking event by Email
   1. Adding email addresses from the Gmail contacts

Questions:



MAT DB will contain MATT’s in the form of views and the time slots marked as occupied by the MAT user

MAT is formed on basis of sync with all Social Network (SN) calendars of the MAT’s user and MAT DB

Pros: Actual up to now information

Cons: Timing and authorization against all Social calendars per each request

Interface 1 – between FES and BES1

Interface 3 –between FES and BES2

* Registration

Standard registration through email

User’s profile (First name, Last name, email, password , SN name, username (unique))

* Login

Username/password of MAT account ***mat\_login(String username, String password): bool***

Username/password of SN ***sn\_login(String sn\_name***, ***String username): bool***

* Creation of MATT

***create\_matt(MattData data, String username):bool***

* Getting of MATT

***get\_matt(String matt\_name, String username):MattData***

* Update MATT

***update\_matt(String matt\_name, String username, MattData data):bool***

* Sharing MATT (interface 3)

***share\_matt(SocialNetwork network, String matt\_name, String username):bool***

Sharing MATT by email (interface 3)

***getContacts(String username):String[]***

***share\_matt\_email(String username,String urlMatt, String [] to):void***

Interface 2 – between BES1 and BES2

* Identity Service

***setIdentity(String snUsername, String matUsername, String snName):bool***

* Getting calendars

***getCalendars(String username, String [] snName, DayInterval interval): MyCalendar;***

22/09/14

1. Discussed Demo
   1. Registration/Login – Component test FES/BES1
   2. Authorization –Component test inside BES2
   3. Viewing/sharing URL – ***http://localhost:8080/myavailabletime/viewMatt?table=<MATT name>&username=<username>***
   4. Sergey Z is responsible for controller function with @RequestMapping (“/viewMatt”). This function will call getMatt with parameters: MATT name and user name and show MATT
   5. Person contains all authorized and selected Social calendars
   6. No polling for viewing updated and synchronized MATT’s
   7. Functions createMatt and getMatt are checking Social networks for the person. In the case if none social networks exists the function getSlotes is not called
   8. Function setMatCalendar implies working with Google Calendar
   9. Function setMatCalendar is called from the function saveMatt. List of actual MATT’a is passed to the function setMatCalendar