

UPCommons

Portal del coneixement obert de la UPC

<http://upcommons.upc.edu/e-prints>

Khafa, F. [et al.] (2016) On sharing and synchronizing groupware calendars under android platform. *2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2016, Fukuoka Institute of Technology (FIT), Fukuoka, Japan 6-8 July 2016: proceedings*. IEEE. Pp. 122-129. Doi: <http://dx.doi.org/10.1109/CISIS.2016.138>.

© 2016 IEEE. Es permet l'ús personal d'aquest material. S'ha de demanar permís a l'IEEE per a qualsevol altre ús, incloent la reimpressió/reedició amb fins publicitaris o promocionals, la creació de noves obres col·lectives per a la revenda o redistribució en servidors o llistes o la reutilització de parts d'aquest treball amb drets d'autor en altres treballs.

Xhafa, F. [et al.] (2016) On sharing and synchronizing groupware calendars under android platform. *2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2016, Fukuoka Institute of Technology (FIT), Fukuoka, Japan 6-8 July 2016: proceedings*. IEEE. Pp. 122-129. Doi: <http://dx.doi.org/10.1109/CISIS.2016.138>.

(c) 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

On Sharing and Synchronizing Groupware Calendars Under Android Platform

Fatos Xhafa*, Daniel Palou
Universitat Politècnica de Catalunya
Barcelona, Spain
Email: fatos@cs.upc.edu

Santi Caballé
Universitat Oberta de Catalunya
Barcelona, Spain
Email: scaballe@uoc.edu

Leonard Barolli
Fukuoka Institute of Technology
Fukuoka, Japan
Email: barolli@fit.ac.jp

Abstract—Sharing a calendar of tasks and events is a cornerstone in collaborative group work. Indeed, the individual work of the members of the group as well as the group work as a whole need the calendar to guide their activity and to meet the deadlines, milestones, deliverables of a project, etc. Additionally the members of the group should be able to work both offline and online, which arises when members of the group use smartphones and can eventually run out of Internet connection from time to time, or simply want to develop some activities locally. In the former case, they should have access to the calendar locally, while in the later case they should access the calendar online, shared by all members of the group. In both cases they should be able to see eventually the same information, namely the local calendars of the members should be synchronized with the group calendar. For the case of smartphones under Android system, one solution could be using the Google calendar, however, that is not easily tailorable to collaborative group work. In this paper we present an analysis, design and implementation of group work calendar that meets several requirements such as 1) sharing among all of members of the group; 2) synchronization among local calendars of members and global group calendar; 3) conflict resolution through a voting system; 4) awareness of changes in the entries (tasks, members, events, etc.) of the calendar and 5) all these requirements under proper privacy, confidentiality and security mechanisms. Moreover, we extend the sharing of calendars among different groups, a situation which often arises in enterprises when different groups need to be aware of other projects' development, or, when some members participate in more than one project at the same time.

Keywords: Calendar, Sharing, Synchronization, Coordination, Groupware, Android, Mobile Teams.

I. INTRODUCTION

With the fast development in Internet and mobile technologies, the collaborative group work is seeing a shift. Indeed, while in the recent past, collaborative group work was based on the web and peer-to-peer applications [17], [25], now most of the group work goes mobile. The use of mobile devices gives a new dimension to collaborative group work, which is not possible with Web-based group work, namely, the members of a group cannot only be geographically distributed, they can also do collaborative work *on the move*, anytime, anywhere [14] in both online

and offline modes of work [26]. Therefore, collaborative group work using smartphones has become commonplace in most collaborative activities, including business collaboration in enterprises, online collaborative learning, collaborative group work at remote areas, disaster management scenarios, crowd-sensing collaboration, etc.

The design of a group calendar requires overcoming several challenges that hinder collaborative group work. Several research works have addressed the development of Android applications that support group work and can be integrated with enterprise information systems [10], [20]. For disaster management, Android applications can provide with timely support as shown in [11], [15], [16]. Several proposal for supporting collaborative work have been proposed for online collaborative learning to effectively scaffold learners, and particularly online groups of learners by smartphones and tablets applications [18], [21], [24], [27]. Data visualisation in a group calendar is also an important feature [7] for handling multiple schedules and highlighting common free times, especially relevant when the number of users is rather large and scalability is to be achieved.

However, currently there are several issues not satisfactorily addressed for a group calendar to fully support the collaborative work. More precisely, in this work we are interested to develop a calendar that meets several requirements such as:

- 1) sharing among all of members of the group;
- 2) synchronization among local calendars of members and global group calendar;
- 3) conflict resolution through a voting system;
- 4) awareness of changes in the entries (tasks, members, events, etc.) of the calendar, and
- 5) all these requirements under proper privacy, confidentiality and security mechanisms.

Moreover, we are interested to extend the sharing of calendars among different groups, a situation which often arises in enterprises when different groups need to be aware of other projects' development, or, when some members participate in more than one project at the same time.

In this paper, we first analyse these requirements and then achieve a design for the group calendar, which can be implemented at a server side and at a client side (mobile

*SmartLearn Group, Universitat Oberta de Catalunya, Spain

device), being it Android or any other platform. We exemplify the approach for mobile devices under Android system and SugarCRM as a server.

The rest of the paper is organized as follows. In Section II we formalise the requirements for a group calendar supporting collaborative group work, the synchronisation mechanisms and its various types. The design of the calendar, its implementation and deployment are presented in Section III. We end the paper in Section IV with some conclusions and remarks for future work.

II. REQUIREMENTS

There are several requirements that a group calendar should satisfy to efficiently support collaborative group work. Among these, we could distinguish some functional and non-functional requirements. In the former group, some main requirements are: **a)** sharing the group calendar among all of members of the group; **b)** synchronization among local calendars of members and the global group calendar; **c)** conflict resolution mechanisms through a voting system; **d)** awareness services to members about the changes in the entries (tasks, members, events, etc.) of the calendar. In the later, we could list, **e)** privacy, confidentiality and security mechanisms for group calendar in order to avoid group activity tracking by non-group members; **f)** sharing of calendars among different groups, when different groups need to be aware of other projects' development activities and **g)** efficiency and scalability with regard to number of tasks and events in a calendar according to number of members of the group(s) and number of different group calendars being shared. It should be noted that some of these requirements can be achieved via a Google calendar, however, they cannot be addressed altogether, especially, conflict resolution mechanisms, privacy, confidentiality and multi-calendar multi-group management.

We briefly describe them next.

A. Sharing the group calendar among all of members of the group

Sharing is a fundamental requirement for a group calendar to support collaborative group work. Members of the group should be aware at anytime and anywhere about the previous, current and future states of the group project as well as of the associated documentation [22]. By sharing the calendars, the members of the group are able to follow up:

- project activities (tasks, deliverables, the project workflow, broadly speaking)
- collaborative group processes (meetings, voting, requests for help and scaffolding, etc.)
- availability and personal state changes.

Sharing should be able to be extended to members of the group for one or more calendars within the group and eventually among members of different groups.

B. Synchronization among local calendars of members and global calendar

Another key requirement is the synchronization among changes produced by a member in the local calendar and the general group or group calendars the member has rights to, that is, when a member of the group changes some entry in the local calendar, this should be reflected on the global calendar and all other mobile devices of members whose local calendar can be seen affected by the change. For members whose local calendar is not affected, it is desirable to warn them about the changes.

There are many aspects that influence the synchronization, such as decentralisation among members of the group (this issue has been much investigated for other models such as P2P Networks—see for instance [1]). By choosing to have the data spread across different devices, when a user makes a change to the data in its device, the change must be communicated directly to the other affected devices; or, the data can be stored centrally on a server and all the synchronization is made through it. In the former case, synchronization is more challenging due to the fact that the mobile devices can go on and off and data consistency is a real issue as it takes more time for the system to reach a consistent state when changes occur (usually *eventual consistency* is sought in this case). In the later case, using a server, it is easier and more reliable to manage the synchronization and achieve desired degree of consistency but to the price of keeping the system centralized.

Among the various strategies for synchronizing data, the most used ones are full and incremental synchronization.

C. Full synchronization

The full synchronization requires always all data to be synchronized. That is, if a device is to be synchronized with current server information, the server will always send all the data to the device (either in pull or push mode, according to setting parameters), and it is the device that should contrast that data received from the server against its data and update it accordingly, for example replace all old data by the newly received ones. This synchronization can be seen as *safe mode synchronization* in terms of synchronization errors, because by receiving all the information of the server side we can be sure that the device will contain exactly the information of the server. Obviously, this synchronization method implies a high cost of temporary data traffic, and it should be noted that part of this data traffic is redundant as the device has already part of it (usually small changes are done on the data locally). However, this kind of synchronization is necessary when a user connects for the very first time to the system or when it connects after a long period of time. In other cases, an incremental synchronization can be used, which is more efficient than full synchronization.

D. Incremental synchronization

The incremental synchronization requires knowing only the data that has changed since the last synchronization and the corresponding changes. Thus, a device receives only the changes occurred, not all information of the calendar. In this case, it is more likely to have a synchronization error or a higher degree of inconsistency. For example, if for some reason the synchronization fails, but through the process there was updated just the time/date of the last synchronization, the synchronization would be considered completed correctly. Clearly, this synchronization is more error-prone in case of mobile devices but on the other hand it is much more efficient both in space and in data traffic since it only sends the necessary changes of the information, not redundant data.

There is also possible to use both synchronization modes, that is, the member of the group uses incremental synchronization and from time to time, upon request, runs a full synchronization with the server.

E. Conflict resolution

In every system that uses data synchronization, conflicts may occur. In the case of the calendar, the most important type of the conflict arises when 1) two users may be modifying the same data and 2) a change done by a member cannot be done persistent in the group calendar until the change is approved by the members of the group (in this later case, this requirement is important for some critical data such as deadline delivery, adding a new member to the calendar who is not member of the group, etc.)

There are many strategies to resolve conflicts [2], [12], one simple strategy is voting. In our case, and due to the importance of correct conflict resolution to the collaborative group work, the conflict resolution is left to the members choice, namely, the affected users take a decision by a simple majority vote. In case of a tie, the voting will be repeated.

F. Notification awareness services

This requirement is necessary to keep the members of a group informed on the group activity and to timely collaborate with other members. Awareness can take various forms such as availability awareness (availability of resources, availability of members, ...), context awareness (where the actions/changes took place), group process awareness (changes with regard to project workflow), etc. [17], [25]. Implementation of these awareness forms enables a timely and effective collaboration and coordination of online groups.

Awareness is implemented via event notification services. Events are defined and linked to Calendar, tasks, groups and users. Then, based on events, two types of notification services are implementing: a) event notification that requires user action and b) warnings (which are simply informative). For example, a user is notified that there is a pending task

approval and is informed (*warning*) when a task is approved, completed and archived. A state diagram is defined for entities and events are linked to different states.

G. Privacy and confidentiality-preserved calendar activities

As with other human activities that generate data, ensuring the privacy and confidentiality is important [3]. On the one hand, an enterprise would like to keep confidential the ongoing projects and people working on them, and, on the other, members of the group may wish to not be tracked by non-group members (for their meeting times, places, etc.). While there are reported various models for Android devices [5], they are no tailored solutions to group calendars.

III. DESIGN, ARCHITECTURE, TECHNOLOGIES AND DEPLOYMENT

A. Design

We present next the design of the group calendar based on the main requirements, namely the design for the client side, i.e. a member of the group (see Fig. 1) and the one for the server side (see Fig. 2).

Further, we show the design of several methods (exemplified using the SugarCRM API), namely, *newCalendar* (see Fig. 3), *addMemberToCalendar* (Fig. 4), *addMemberToTask* (Fig. 5) and *confirmPendingApproval* (Fig. 6).

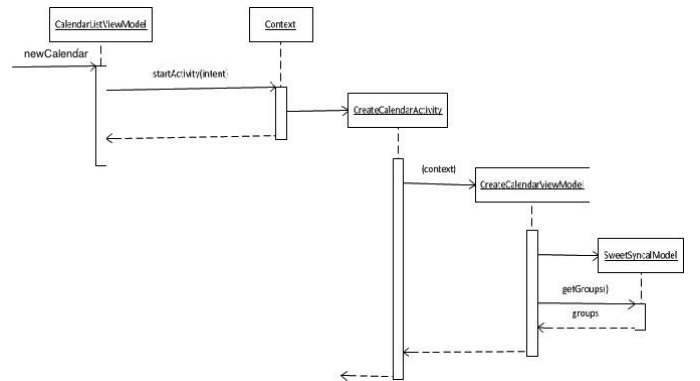


Figure 3. The sequence diagram of *newCalendar*.

B. Architecture

Regarding the architecture, it should be flexible, adaptive and inter-operable. Previous works have pointed out architectural concerns in distributed and mobile collaborative systems [6], [19]. Among them, achieving integration of process and workspace management with Peer-to-Peer (P2P) Middleware, Publish-Subscribe, and Community and User Management are identified as key requirements.

Therefore, the architecture of the software system supporting collaborative work should account for web servers, services, super-peer implementation and mobile peers. Some issues in developing such architectures have been addressed in [8], [9], [13], [23].

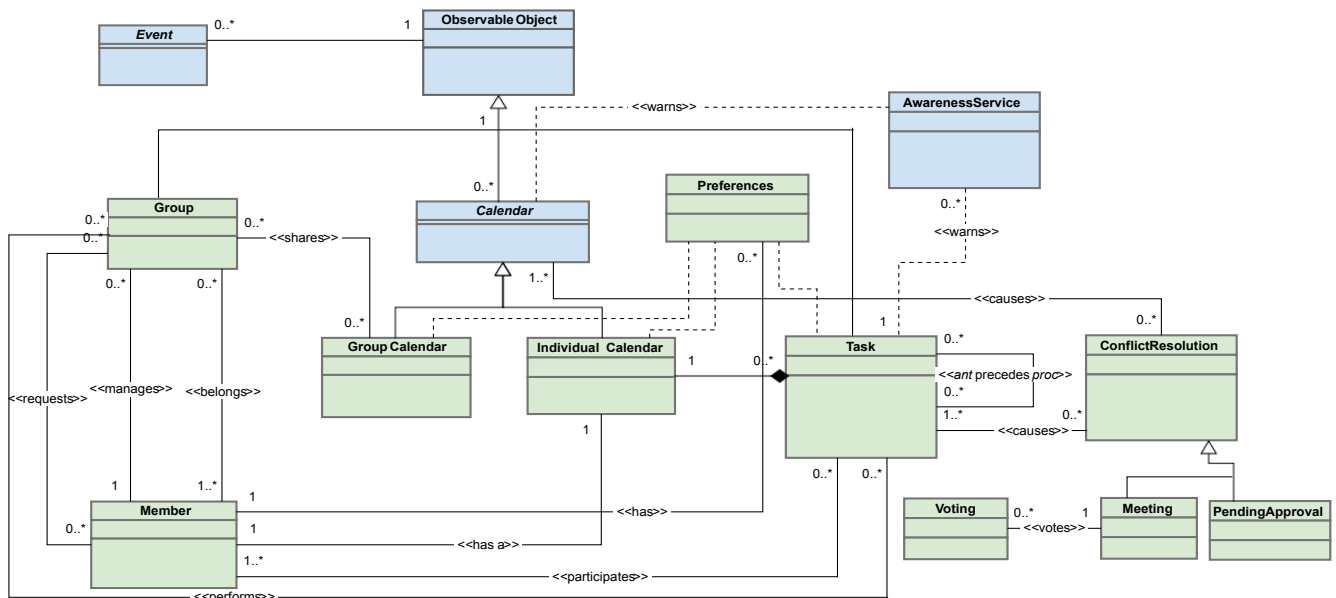


Figure 1. Calendar design at client side.

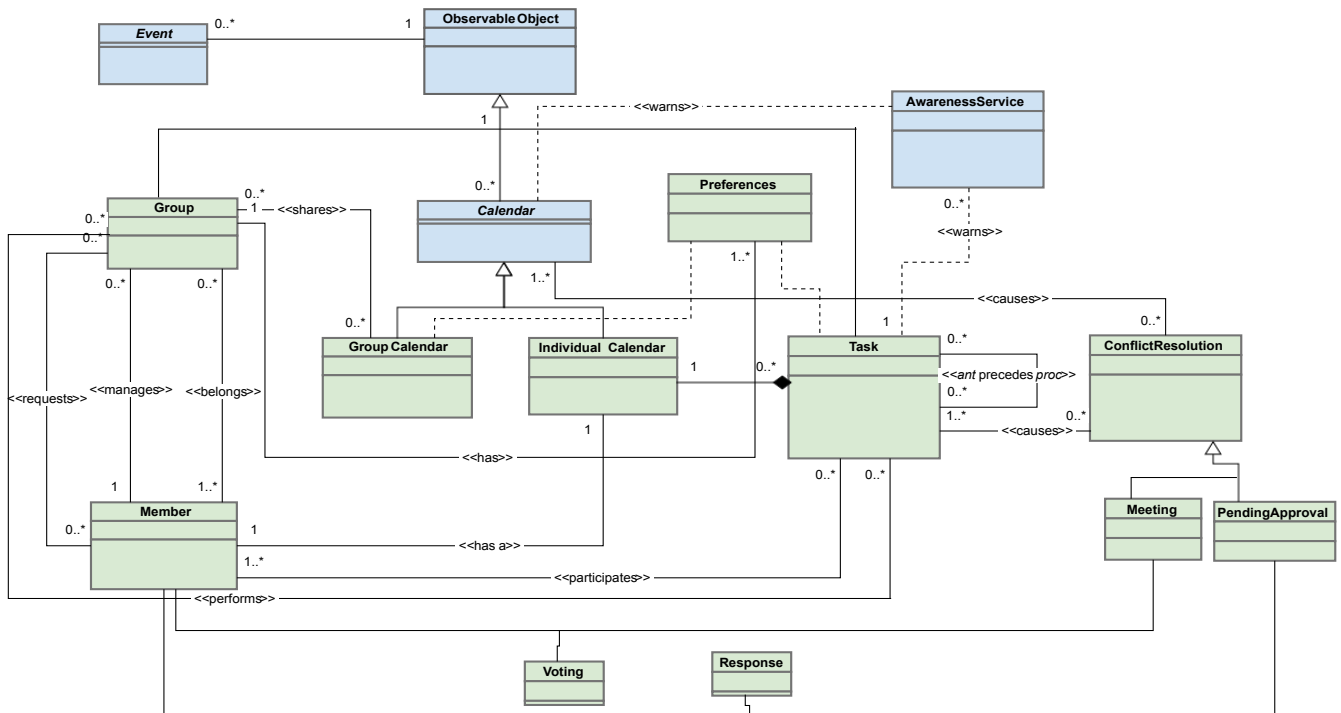


Figure 2. Calendar design at server side.

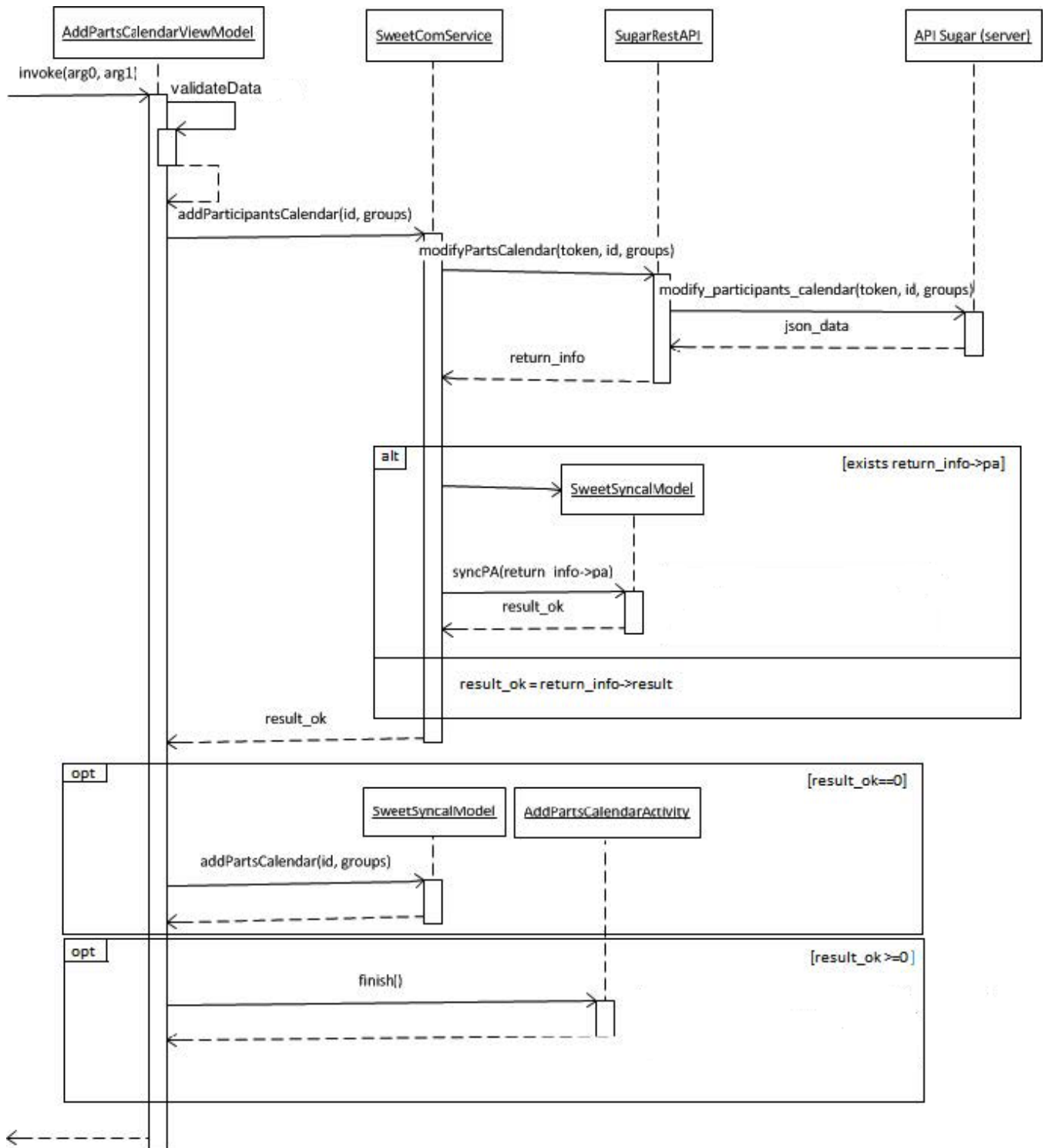


Figure 4. The sequence diagram of *addMemberToCalendar*.

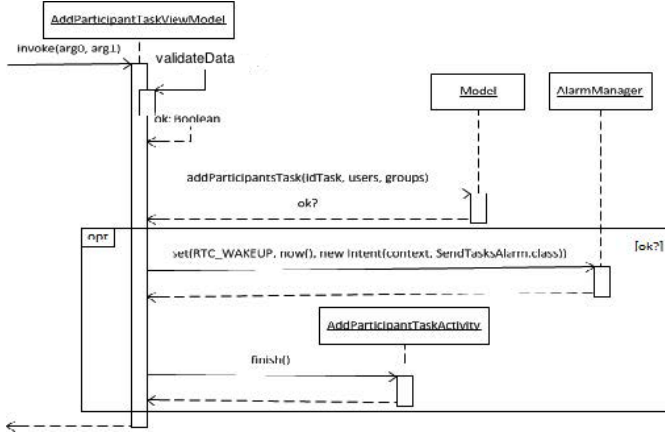


Figure 5. The sequence diagram of *addMemberToTask*.

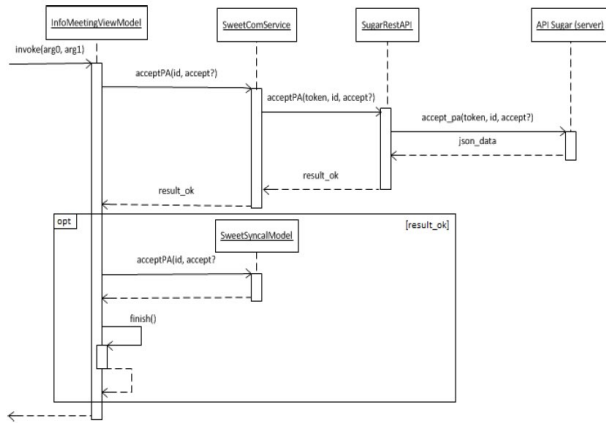


Figure 6. The sequence diagram of *confirmPendingApproval*.

The architecture pattern used is that of Model View ViewModel (see fig. 7).

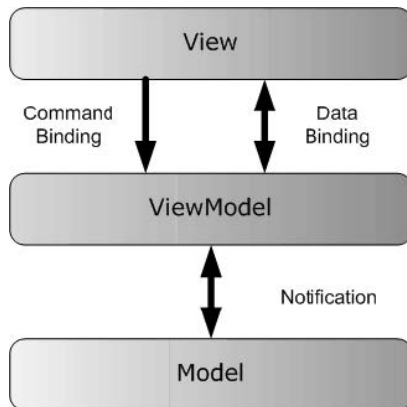


Figure 7. Model View ViewModel Pattern.

C. Implementation and technologies

The following technologies have been used to implement the system.

Java: The system is designed to run on Android, whose programming language is Java.

SQLite: We use SQLite as a local database because the resources are quite limited. Storing data in files can also be considered, but it was decided to use the database as a more effective solution.

Android: Because we want our application to use the maximum number of available devices, it was decided that the application will be implemented under Android, nevertheless, the system can be fully implemented using other operating systems.

Android-Binding: We use this external open source library to achieve a greater independence between the view and our system using the MVVM pattern.

SugarCRM: SugarCRM is commonplace in applications for SMEs businesses. By choosing this application server, we simulate the case of a real project development in enterprises and make our application potentially useful for real setting.

PHP: The server functionality must be extended in order to adapt it to our system. Because SugarCRM is implemented in PHP, it is then the programming language used to expand it.

Memcached: Memcached is a distributed system that is used for storing cached data or objects, thus reducing the need to access an external data source.

JXTA: JXTA (Juxtapose) is a P2P open source platform created by Sun Microsystems and defined as a set of XML-based protocols. These protocols allow devices connected to a network to exchange messages with each other regardless of the network topology. The definition of JXTA is abstract and based on a set of open protocols so that it can be brought to any modern programming languages. There are several implementations, the most advanced of which is JXSE the Java version. The version is called Java Micro Edition JXME. One important feature of JXTA is that it allows P2P communication even when peers are behind NATs or firewalls.

PeerDroid: PeerDroid is an open source library that implements protocols JXME for the Android platform. It allows to create applications for the Android platform using JXTA properties, creating a network of mobile and other traditional peers (computers, for example).

D. Deployment and evaluation

The system has been deployed at the RDLab infrastructure (<http://rdlab.cs.upc.edu/>, using SugarCRM server and has been evaluated according to different usability and performance criteria. The system is under full evaluation with real collaborative groups conducting software project development at online courses of a Virtual Campus.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we have presented an application for a calendar for collaborative group work to guide the individual and group activity in order to meet the deadlines, milestones, deliverables of a project, etc. We have first analysed the main functional requirements of the calendar such as sharing, synchronization, conflict resolution through a voting system and ability to support working both offline and online but also non-functional requirements such as privacy, confidentiality and security. Among these requirements, the synchronization is a challenging issue. We have formalized the requirements through the design of several algorithms, classes, modules and systems (client and server ones) and have implemented the proposed design using Android as a platform for smartphones and SugarCRM for the server. Preliminary evaluation of the prototype showed its efficiency and usability. We are currently evaluating the proposed systems using real collaborative groups at online courses of a Virtual Campus.

In our future work we would like to improve the group calendar in some aspects, for instance, with (fuzzy) meeting agent for decision support system, which would give better conflict resolution than current implemented method of majority voting.

REFERENCES

- [1] Achara, J.P., Imine, A. Rusinowitch, M. DeSCal - Decentralized Shared Calendar for P2P and Ad-Hoc Networks. *10th International Symposium on Parallel and Distributed Computing*. 223 - 231, DOI: 10.1109/ISPDC.2011.40, 2011
- [2] Samuel A. Ajila, S.A. and Al-Asaad, A. Mobile databases – Synchronization & conflict resolution strategies using SQL server. *IEEE International Conference on Information Reuse and Integration (IRI)*, 487 - 489, DOI: 10.1109/IRI.2011.6009598, 2011
- [3] Yu-Jia Chen, Y.J., Chia-Yu Lin, Ch.Y and Wang, L. Ch. A Privacy-Preserved Joint Group Time Scheduling Mechanism for Mobile Social Applications. *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 150 - 155, DOI: 10.1109/Trust-Com.2013.23, 2013.
- [4] De Melo, L.L., Zorzo, S.D. PUPDroid - Personalized user privacy mechanism for android. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 1479 - 1484, DOI: 10.1109/ICSMC.2012.6377944, 2012
- [5] Do, Q., Martini, B. and Choo, K.K.R. Enhancing User Privacy on Android Mobile Devices via Permissions Removal. *47th Hawaii International Conference on System Sciences*, 5070 - 5079, DOI: 10.1109/HICSS.2014.623, 2014
- [6] Dustdar, S.; Gall, H., Architectural concerns in distributed and mobile collaborative systems. *Proceedings of the Eleventh Euromicro Conference on Parallel, Distributed and Network-Based Processing*, 475-483, 2003.
- [7] Engin, B., Cetinkaya, M., Ayiter, E., Germen, M. and Balcişoy, S. Maestro: Design Challenges for a Group Calendar. *12th International Conference Information Visualisation*, 491 - 496, DOI: 10.1109/IV.2008.91, 2008
- [8] Franco, D. F., Neyem, A., Ochoa, S., Navarro, R. Supporting Mobile Virtual Team's Coordination with SOA-Based Active Entities. Establishing the Foundation of Collaborative Networks. *The International Federation for Information Processing* Vol. 243, 557-564, 2007.
- [9] Ghezzi, C, Cugola, G. and Picco, G. P. PeerWare: A Peer-to-Peer Middleware for Mobile TeamWork. *ERCIM News* No. 54, 2003.
- [10] Gryaznov, G. and Kovin, R. Development of the cross-platform mobile framework for integration with enterprise information systems. *7th International Forum on Strategic Technology (IFOST)*, 1 - 4, DOI: 10.1109/IFOST.2012.6357607, 2012.
- [11] Juhana, T., Widyani, R. N, and Mulyana, E. Mobile application for rapid disaster victim assessment. *7th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, 324 - 329, DOI: 10.1109/TSSA.2012.6366076, 2012.
- [12] Kedia, A. and Prakash, A. Data synchronization on Android clients. *IEEE International Conference on Communication Software and Networks (ICCSN)*, 212 - 216, DOI: 10.1109/ICCSN.2015.7296156, 2015
- [13] Kirda, E., Fenkam, P., Reif G., and Gall, H. A service architecture for mobile teamwork. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering (SEKE '02)*. ACM, New York, NY, USA, 513-518. DOI=10.1145/568760.568850 <http://doi.acm.org/10.1145/568760.568850>, 2002.
- [14] Leonardi, L., Mamei, M. and Zambonelli, F. A Physically Grounded Approach to Coordinate Movements in a Team. In *Proceedings of the 1st International Workshop on Mobile Teamwork*, 2002.
- [15] Nakamoto, Y. and Akiyama, Sh. A Proposal for Mobile Collaborative Work Support Platform Using an Embedded Data Stream Management System. *35th IEEE International Conference on Distributed Computing Systems Workshops*, 23 - 28, DOI: 10.1109/ICDCSW.2015.16, 2015.
- [16] Rahman, K.M., Alam, T. and Chowdhury, M. Location based early disaster warning and evacuation system on mobile phones using OpenStreetMap. *IEEE Conference on Open Systems (ICOS)*, 1 - 6, DOI: 10.1109/ICOS.2012.6417627, 2012.
- [17] Sapateiro, C., N. Baloian, P. Antunes and G. Zurita (2009) Developing Collaborative Peer-to-Peer Applications on Mobile Devices. *Proceedings of the 13th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2009)*, Santiago, Chile, 2009.
- [18] Djoni Haryadi Setiabudi, D.H., and Winsen, L.J.T. Mobile learning application based on hybrid mobile application technology running on Android smartphone and Blackberry. *International Conference on ICT for Smart Society (ICISS)*, 1 - 5, DOI: 10.1109/ICTSS.2013.6588081, 2013.

- [19] Su, X., Prabhu, B. S., Chi-Cheng Chu, Gadh, R. Middleware for Multimedia Mobile Collaborative System. *IEEE ComSoc , 3rd Annual Wireless Telecommunication Symposium (WTS 2004)*, California, USA, 2004.
- [20] Vojvodić, S., Zović, M., Režić, V., Maračić, H. and Kusek, M. Competence transfer through enterprise mobile application development. *37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 448 - 452, DOI: 10.1109/MIPRO.2014.6859609, 2014.
- [21] Wei, S. The Design and Implementation of a Mobile Learning Platform Based on Android. *International Conference on Information Science and Cloud Computing Companion (ISCC-C)*, 345 - 350, DOI: 10.1109/ISCC-C.2013.136, 2013.
- [22] Xhafa, F., Barolli, L., Fernandez, R., Daradoumis, Th., Caballé, S., Kolici, V. Extending JXTA Protocols for P2P File Sharing Systems. *CISIS 2008*: 35-40, 2008.
- [23] Xhafa, F., Barolli, B., Caballé, S., Fernandez, R.: Efficient peerGroup management in JXTA-Overlay P2P system for developing groupware tools. *The Journal of Supercomputing*, 53(1): 45-65, 2010.
- [24] Xhafa, F., Barolli, L., Caballé, S., Fernandez, R. Supporting Scenario-Based Online Learning with P2P Group-Based Systems. *NBiS 2010*: 173-180, 2010.
- [25] Xhafa, F. and Poullovassilis, A. Requirements for Distributed Event-Based Awareness in P2P Groupware Systems. *AINA Workshops 2010*: 220-225, 2010.
- [26] Yang, Y., Supporting online Web-based teamwork in offline mobile mode too. *Proceedings of the First International Conference on Web Information Systems Engineering*, Vol.1, 486-490, 2000.
- [27] Yang, H. Ch. and Wang, W.Y. Facilitating Academic Service-Learning with Android-Based Applications and Ubiquitous Computing Environment. *4th International Conference on Ubi-Media Computing (U-Media)*, 191 - 196, DOI: 10.1109/U-MEDIA.2011.29. 2011