



## **LEDAS FreeTime Overview**

***Version: 1.0***

*LEDAS Ltd.  
Novosibirsk  
Russia*

<http://ledas.com>  
*Mail-to: sales@ledas.com*  
*+7 383 335 65 04*

Copyright LEDAS Ltd., 2004-2005  
All brand and product names are the service marks, trademarks, or registered  
trademarks of their respective owners.

# TABLE OF CONTENTS

---

---

Table of contents .....	2
1. Introduction .....	3
2. Purpose of FreeTime and its place in the market .....	4
3. Who benefits from FreeTime .....	6
4. FreeTime strengths .....	7
5. Main FreeTime definitions and functionalities .....	8
6. Using FreeTime within the Evaluation Program .....	10
7. Technical requirements.....	11
8. Invitation to joint development.....	12
9. A detailed sample scenario .....	13
10. The product architecture.....	15
11. Algorithms in the FreeTime core.....	17

## 1. Introduction

---

FreeTime is an end-user software solution for computer-aided optimal scheduling of meetings.

FreeTime was developed to coordinate meetings with participation of people from a joint team using resources from a joint pool. The term “team” is not limited to industrial groups but includes the entire spectrum of the meanings that cover commercial and non-profit organizations where scheduling is necessary. At the same time, the term “meeting” should also be extended and could correspond to every “thing to do” in everyday life: meetings, small talks, short trips, phone calls, and even coffee-breaks. Similarly, the term “resource pool” is meant to include all available rooms, projectors, computers or other equipment that can be useful for “meetings”.

The distributed FreeTime system helps users to choose an optimal timeframe and place for each meeting in real time:

- A user can specify any rough timeframe for a meeting, like “this afternoon”, “before the end of this week”, “each month”, etc – the FreeTime system will automatically choose an exact time depending on the particular schedules of all participants,
- If a meeting demands some resources – rooms, special tools or devices - the FreeTime system will automatically take it into account and choose an appropriate distribution of all resources among meetings
- If some participants cannot be available, even in a given “rough” timeframe, the FreeTime system will automatically re-schedule other meetings to find a time and place for all meetings.

The competitive power of this new LEDAS product is based on a unique combination of its features:

- FreeTime provides automatic schedule optimization and resolution of conflicts,
- FreeTime is based on the client-server protocol with all scheduling done by a powerful server,
- FreeTime enables the planning of meetings by using notebooks, PDAs, and mobile phones,
- FreeTime supports true and efficient interaction among those who are involved in the planning process,
- FreeTime includes a broad set of high level end-user capabilities,
- FreeTime is cost-efficient software with a flexible pricing policy.

The advantages listed above make the LEDAS scheduling solution powerful, flexible, and mobile – we see these characteristics as the keys to the success of FreeTime at the market.

## **2. Purpose of FreeTime and its place in the market**

---

### **2.1. Typical meeting scheduling functionality in the market**

There are plenty of the meeting-scheduling solutions on the market. The main idea of any of them is to provide users the means to organize events in timeframe appropriate for all attendees with regards to their personal free time. There are stand-alone meeting planning systems (e.g. Meeting Maker by Meeting Maker, Inc.) and scheduling modules within the framework of office packages like Microsoft Office Outlook by Microsoft. The basic functionality of all such solutions is almost the same: they allow a group of users to share their calendars and to find an appropriate timeframe for some future meeting or just to place a meeting at some fixed time and let attendees accept or decline this time for that meeting. The existing solutions suppose that each team member has his own schedule that is made up of busy (intervals occupied by previously scheduled events) or free intervals of time. The scheduling goal is then to find a timeframe which is free for scheduling of all necessary attendees or to just ignore the fact that some of the attendees are busy during the chosen time and will not be able to attend. Such scheduling is done by a person – a meeting organizer – who is responsible for checking personal agendas and for the decision to ignore conflicts if necessary. Different visual and analytical tools are provided to make locating a desirable time-slot easier, but scheduling itself remains a manual task for the meeting organizers. The organizer is also responsible for re-scheduling a meeting if one of the attendees who “must attend” finds the chosen time unavailable.

### **2.2. Advantages of FreeTime**

The main idea of FreeTime is to make the meeting planning process more collaborative, more flexible and more convenient for all team/project members. Unlike most other systems, FreeTime allows its users to define a timeframe for meetings instead of a single fixed start time, end time and duration. Then the software automatically locates a time for all meetings in all personal schedules. The algorithms incorporated into the software can resolve scheduling conflicts of various kinds and even re-schedule some already defined meetings if it is needed. The software switches into a manual-scheduling mode only if it cannot automatically find a schedule without conflicts.

Automatic optimization of schedules and plans is still not common even for big and expensive industrial software solutions. There are even fewer affordable mobile solutions available for SME and small groups. Therefore, another goal of FreeTime development is to make highly automated scheduling immediately available to everyone to respond to everyday planning challenges in a timely manner wherever you are. For this reason, the powerful scheduling capabilities of FreeTime have been made available not only for PCs, but also for hand-held devices and mobile phones. Now you can organize meetings and group events while traveling, from you home or in any other situation just when you decided to do so. You will also be able to reply to invitations to some new event. By using FreeTime you can respond promptly and align your plans with this upcoming event or reject an inappropriate timeframe, leaving time for the system to reschedule.

Compared to existing systems, FreeTime supports all the basic terms and data representations that are common for such systems, adding more flexibility, collaboration and mobility.

### **2.3. How FreeTime works: a short sample scenario**

The short scenario below gives an idea of the styles in which FreeTime can be used. In Section 9, a more complete scenario is given to explain in detail how FreeTime works.

*Peter works in the marketing department. While on a business trip, he receives an attractive offer to collaborate with a well-known automaker. He is in the airport waiting for his flight home.*

*17:00 - Peter takes his PDA and, using FreeTime, plans to hold a 1-hour meeting with the business development director and chief engineer of his company as invited participants. He specifies that it is necessary to hold the meeting tomorrow before the end of the working day.*

*After specification of all required parameters, Peter synchronizes them with the FreeTime server located in the office of his company (to connect the server, Peter uses the WiFi-spot in the airport). The server automatically finds the most appropriate time and site for this meeting – from 14:00 to 15:00, since it is clear from the schedule for tomorrow that the chief engineer is busy in the morning up to 12:30 at meetings with the chiefs of the construction bureau and the production department, and he usually has lunch from 13:00 to 14:00.*

*17:10 - Andrew, the chief engineer of the company, is holding a meeting with the IT-manager and the engineer-designers. From time to time, he takes a look at his laptop connected to the local network. After Peter's query was processed by the server, Andrew receives an invitation to the meeting to be held tomorrow from 14:00 to 15:00 and immediately confirms it.*

*17:15 - Nick, the business development director, is going to the investment bank by taxi. With the help of the FreeTime client application on his mobile phone, he receives an invitation to an important meeting with Peter and Andrew tomorrow from 14:00 to 15:00. Nick has scheduled negotiations on a credit application in another bank for tomorrow and he realizes that, after his talks in the bank and business lunch, it is impossible to get to his office by 14:00, so he rejects the proposed time but says that he is quite free from 15:00 to the end of the working day.*

*The server immediately processes the message from Nick sent via GPRS. The program sees that there are several meetings with Peter and Andrew already scheduled for tomorrow from 15:00 to 19:00 but, shifting some of them, finds a free interval from 16:30 to 17:30.*

*17:20 - Andrew and Nick, with their laptop and mobile phone, synchronously receive a proposition to move tomorrow's meeting to 16:30 and confirms it. The FreeTime server sends this information to Peter who is now in the airplane. When he sees that, as a result of successful synchronization with his colleagues, the meeting is scheduled for 16:30, Peter opens the file folder and starts preparations for the meeting...*

### **3. Who benefits from FreeTime**

---

FreeTime is, first of all, aimed at business use. It is intended for companies, where much time is spent for such team events as meetings:

- This may be a dynamic company conducting a lot of projects, where meetings help to track ongoing activities and share knowledge among employees,
- This also may be a trading or service company, where regular personnel-to-customers, or management-to-partners meetings are vital for successful business,
- This finally can be virtually any company: in the current business environment ongoing activities require regular training sessions to constantly increase staff proficiency, meetings to exchange knowledge and elaborate best practices, and seminars to keep employees informed about company directions and strategy.

FreeTime can also help to organize personal free time:

- Social clubs can use FreeTime to keep members informed about sessions, events etc,
- Some fun groups can also use FreeTime to organize joint activities like football matches, snowboarding travels, or bridge tournaments.

Scheduling events and meetings is an essential part of everyday life and so everyone could potentially benefit from FreeTime. Its ease of use, flexibility and mobility can save a lot of time and effort by simplifying everyday planning and allow more free time for personal work and fun.

## **4. FreeTime strengths**

---

Automated scheduling of meetings is a big advantage over manual finding and assigning time. It is much easier; it saves a lot of time, and it allows more collaboration from all the participants. This automated scheduling becomes possible because of a powerful computational engine called the LEDAS Scheduler. The Scheduler is a core technology capable of solving all kinds of scheduling tasks using state-of-the art algorithms packaged into a flexible infrastructure, which allows integration into a broad class of applications.

Therefore, one of the major strengths of FreeTime is its computational power provided by LEDAS Scheduler: FreeTime is capable of high performance when finding non-contradictory schedules for hundreds of events with hundreds of participants. FreeTime processing of inputted data is always preserving “naturalness” of the resulting schedule – namely for each next meeting it tries to find a schedule to affect as few already scheduled events as possible and simultaneously to respect as many constraints defined for the new event as possible.

Another important thing about FreeTime is its mobility. Since all the scheduling is done by the server, each client interface remains very lightweight and simple, keeping it powerful enough to comfortably define events and view personal agendas. Therefore, FreeTime clients for PDAs and even mobile phones with direct access to the scheduling server are available providing scheduling right at your fingertips.

Each client, be it for mobile phone, PDA, or web, provides all means to create new events, receive and provide feedback for invitations and notifications of events, scheduled by others and standard views for personal agendas.

LEDAS is planning to perform integration of FreeTime with Microsoft Outlook and Lotus Notes as well as to build interaction between PDA client and Palm OS built-in organizer. There are also similar plans regarding the built-in organizers for mobile phones.

## **5. Main FreeTime definitions and functionalities**

---

We give here a short overview of the terminology used in FreeTime, along with a brief characteristic of its functionalities. After getting acquainted with this section, it is recommended to look thorough the sample scenario in Section 9.

To work with FreeTime, the following main terms for the entities are used:

### **Person**

A person, or participant, is the basic entity of our system, since the whole system is designed in order to help a user schedule meetings with his/her colleagues – the other users of the system.

The administrator of the company sets up all persons at the server and inputs their logins and passwords. He also defines which of the users would have supervisor rights. Ordinary users can see in the schedule views only their own private jobs and meetings they participate in, which makes the scheduling process easier and faster, while supervisors can see all planned events of the company. This information can be useful, for example, for managers. During the work in the system, all persons may create entities of all types (see description below), except resources and other persons.

### **Resource**

Resource is a place, object, or another tool or equipment that may be required to hold a meeting. Resources are divided in two categories: Rooms and Equipment. Any resource can be assigned to a meeting.

Like persons, all resources are created at the server by the system administrator.

### **Private job**

This entity represents some kind of activity that does not allow a user to participate in meetings during that period of time. A private job can be edited by its creator only but it is also shown to all users with supervisor rights.

### **Meeting**

A meeting is created by a person. The creator defines a set of resources required for a meeting and invites other participants. When a meeting is created, notifications are sent to everybody invited and they should confirm participation. Only registered company employees can be invited to a meeting.

A meeting, until it is confirmed by all its participants, doesn't have a fixed start time and is described by a time interval ("Begin after" and "End before") and a desirable starting time ("Optimal start"). The server can change this assumed starting time to resolve conflicts with other meetings or with a private job of one of the participants. Other attributes of the meeting (duration, list of the participants, etc.) may be changed only by the creator of the meeting.

Each person can add a personal restriction on his participation in this meeting.

### **Restriction**

A restriction allows a user to show that he or she is not able to participate in a particular meeting at that time. The difference between a personal restriction and a private job is that the user remains available for other meetings during a specified period of time.

### **Notification**

A notification is a message about a change in a meeting's status. It is created every time

information about the meeting is changed. A notification can have two states – read (or confirmed) and unread. If the notification has the state “read,” then it will be deleted during the next data synchronization and the server will be informed about confirmation of the notification by the user. The server can also send an e-mail message for each notification, informing the user that he has got some changes in his schedule.

## **Calendar**

A weekly calendar is an entity that is specified by system administrator. He defines which of the days are working and then, for each of these days, sets up to three working intervals. The system will automatically schedule all the meetings into working intervals.

The main functions of FreeTime include the following:

0. System preparation
1. Installation and launch of a user application
2. Entering and editing of user settings
3. Synchronization with the server
4. Work with the calendar
  - 4.1 Viewing a month schedule
  - 4.2 Viewing a week schedule
  - 4.3 Viewing a day schedule
5. Viewing the information about a particular meeting
6. Creation and deleting of private jobs
7. Creation and deletion of a personal restriction
8. Creation and editing of a meeting
9. Work with the notifications

All the functions are described in detail in the users guide.

## 6. Using FreeTime within the Evaluation Program

---

In order to allow potential users of the FreeTime system to evaluate the added value of this software, LEDAS provides the Evaluation Program. In the framework of this program the demo versions of client applications and documentation for them can be downloaded from the LEDAS web site. The server part of FreeTime as well as the web-application will be hosted on the LEDAS site. They will be delivered to our customers together with mobile client applications after the signing of the license agreement.

Before entering the FreeTime Evaluation Program, please become acquainted with the technical requirements for the system (Section 7 below).

To start working with the FreeTime system, you need to register in Evaluation Program at the FreeTime site (<http://freetime.ledas.com>) and do the following:

- 1) Read the license agreement and confirm that you agree to use the FreeTime system under the specified conditions
- 2) Fill in the form suggested on the next page where you must enter you name, e-mail address and the name of your company and push the “sign up” button

From this moment, you have authority to enter the FreeTime Evaluation Program site using your password and e-mail address as login. After that you're allowed to load documentation and client versions of the FreeTime applications, to send a user report, and get news about system development.

Using your password and e-mail address as login, you are also authorized to use the IMS (Incident Management System, see <http://ims.ledas.com>) of the LEDAS Company for sending messages about bugs in the FreeTime software.

On receiving the confirmation letter, you become the administrator of one of the newly created companies on the FreeTime server. From this moment, you have access to the administrator's component of the FreeTime web-application using the name of your company, password entered at registration and login “admin”. There you can create, modify or delete persons, resources and weekly calendar of the company. After creation of all needed persons and resources, all client applications of FreeTime system are ready to work.

## 7. Technical requirements

---

### **FreeTime server technical requirements**

- Microsoft Windows NT4/2000/XP
- Internet Information Services 4 or 5 (it is a part of Microsoft Windows) switched on
- PHP 4.3.11
- Internet connection by HTTP protocol
- SMTP server (for e-mail notifications only)
- 10 Mb of disk space

### **Web client technical requirements**

You should:

- use Internet browsers Internet Explorer 4.0, Opera 8.0 or Mozilla 1.3 and higher versions
- switch on JavaScript support (not earlier than version 1.2)
- use CSS technology in your browser
- work with screen resolution not less than 1024x768

More than 95% of Internet users are known to work in this environment, which suits the FreeTime web-client requirements perfectly.

### **PDA client technical requirements**

- Operation system PalmOS 3.5 or higher
- Internet connection module (for example, Wi-Fi, GPRS or Bluetooth)
- Screen resolution 160x160 or 320x320
- At least 16 gray scale palette

Our analysis of the market shows that all PalmOS devices being produced today have these features.

### **Mobile phone client technical requirements**

You should:

- have the support of Java technology MIDP 2.0
- be able to load Java application distributive of size up to 128KB
- use network gata support (GPRS)
- have 5-way joystick
- use color depth of 16 bit or greater
- have 2 soft keys
- work with a screen resolution not less than 128x128

In fact, most of contemporary mobile phones of business class (such as the popular Siemens S65 or Nokia 6230) satisfy all the requirements formulated above.

## **8. Invitation to joint development**

---

An initial release of the FreeTime solution is already available. However, the development of the FreeTime system is still intensively going on. LEDAS has its own vision of an ideal meeting scheduling solution, but this vision can be adjusted based on the feedback from the users. We invite all interested parties to contribute to the FreeTime development by sending us their vital requirements, wishes and thoughts about an ideal meeting scheduler.

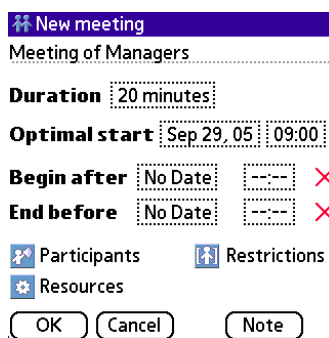
We also have a partnership program for all the parties that want to have that ideal solution or any other solutions based on FreeTime and Scheduler technologies, gain more control over FreeTime and Scheduler enhancement, and to make it faster. Creation of a technology comparable to the LEDAS Scheduler is very expensive: licensing of a third-party solver consumes funds, increases product price and affects supportability because your solution contains uncontrolled third-party code. We offer a reasonable alternative: by investing in the development of the Scheduler you leave the development of a solver to the field experts, gain some control over the development process, can license the final version of LEDAS Scheduler at a special price and cover this licensing by your investments (your investment will be regarded as the licensing fee), control support, further development and enhancement trends with access to the latest releases.

For all who need some extra functionality not covered by FreeTime, LEDAS offers consultancy and outsourcing services. FreeTime team can implement features not presented in the public FreeTime version but close by nature to the tasks for which FreeTime is designed. Such development can be done on terms varying from pure outsourcing (with all rights for this FreeTime version belonging to the customer) to joint development of a FreeTime-based or Scheduler-based market product, which can be distributed by the customer or cooperatively by the customer and LEDAS.

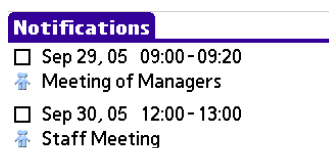
## 9. A detailed sample scenario

To demonstrate how FreeTime operates, in this section we describe a detailed sample scenario, which is illustrated by screenshots taken from a real-life FreeTime version. It is recommended to reproduce this scenario while working with commercial or demo version of the FreeTime. The terminology used in the scenario is explained in the Section 5, all users in the scenario have supervisor rights.

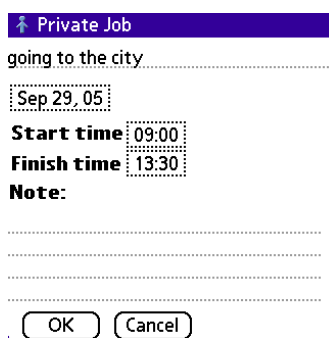
Below you can find an example of scheduling meetings in a two-day timeframe for a small IT project.



Screenshot 1a



Screenshot 2a



Screenshot 3a

### STEP 1: Wednesday, 13:20

Project manager Paul Owen creates Meeting of Managers (see screenshots 1a and 1b) with participants Max Twain and Paul Owen on Thursday, 9:00, with duration of 20 minutes.

Paul Owen creates Staff Meeting with participants Max Twain, Paul Owen, John Taylor and Joe Fisher on Friday, 12:00, with duration of 60 minutes.

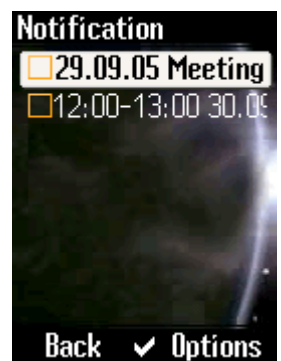
Paul Owen synchronizes at 13:27.

Paul Owen gets notifications (see screenshots 2a and 2b) of Meeting of Managers (9:00, Thursday) and Staff Meeting (12:00, Friday).

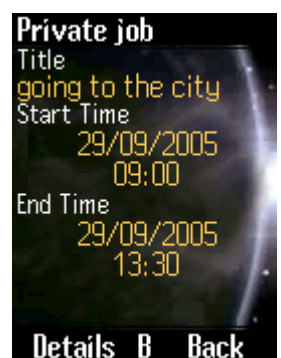
Paul Owen confirms them at 13:27.



Screenshot 1b



Screenshot 2b



Screenshot 3b

### STEP 2: Wednesday, 13:25

Vice-manager of the project Max Twain creates a private job “going to the city” (see screenshots 3a and 3b) on Thursday, 9:00-13:30.

Max Twain synchronizes at 13:28.

FreeTime moves Meeting of Managers to 13:30, Thursday.

Max Twain gets notifications of Meeting of Managers (13:30, Thursday) and Staff Meeting (12:00, Friday).

Max Twain confirms them at 13:29.

### STEP 3: Wednesday, 13:35

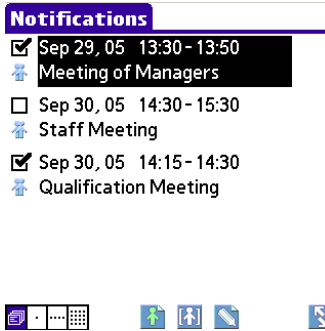
First developer John Taylor creates a private job, Weekly Report Preparation, on Friday, 10:00-13:00.

John Taylor synchronizes at 13:36.

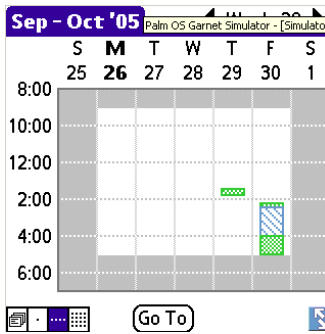
FreeTime moves Staff Meeting to 13:00, Friday.

John Taylor gets notifications of Staff Meeting (13:00, Friday).

John Taylor confirms them at 13:36.



Screenshot 4a



Screenshot 5a



Screenshot 6a



Screenshot 7a

**STEP 4: Wednesday, 13:40**

Second developer Joe Fisher synchronizes and gets notification of Staff Meeting scheduled at 13:00, Friday.

Joe Fisher creates personal restriction on Staff Meeting on Friday, 9:00-14:00.

Joe Fisher synchronizes at 13:42.

FreeTime moves Staff Meeting to 14:00, Friday.

Joe Fisher gets notifications of Staff Meeting (14:00, Friday).

Joe Fisher confirms them at 13:43.

**STEP 5: Wednesday, 13:45**

Paul Owen creates Qualification Meeting with participants Paul Owen, Joe Fisher on Friday, 14:15, with duration of 15 minutes.

Paul Owen synchronizes at 13:46.

FreeTime moves Staff Meeting to 14:30 (removing the conflict between Staff Meeting and Qualification Meeting).

Paul Owen gets notifications of Meeting of Managers (13:30, Thursday), Staff Meeting (14:30, Friday) and Qualification Meeting (14:15, Friday).

Paul Owen confirms Meeting of Managers and Qualification Meeting at 13:47 (Meeting of Managers becomes confirmed, see screenshots 4a and 4b).

Paul Owen creates a private job, Business Lunch, on Friday, 14:30-16:00.

Paul Owen synchronizes at 13:48.

FreeTime moves Staff Meeting to 16:00.

Paul Owen gets notification of Staff Meeting (16:00, Friday) and confirms it.

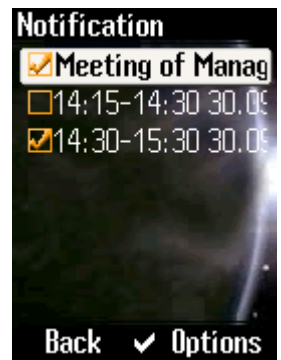
**STEP 6:**

Max Twain synchronizes, gets notification of Staff Meeting (16:00, Friday) and confirms it.

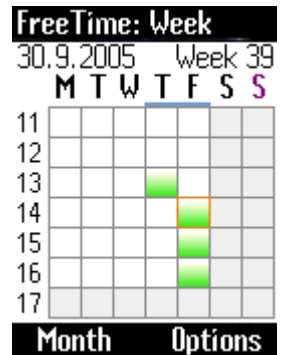
Max Twain synchronizes, gets notification of Staff Meeting (16:00, Friday) and confirms it.

Joe Fisher synchronizes, gets notifications of Staff Meeting (16:00, Friday), Qualification Meeting (14:15, Friday) and confirms them – all meetings become confirmed.

Final schedule (Paul Owen's as an example) can be seen in week view on screenshots 5a and 5b, in month view on screenshots 6a and 6b, and in notification view on screenshots 7a and 7b.



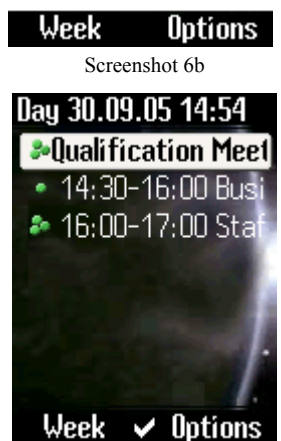
Screenshot 4b



Screenshot 5b



Screenshot 6b



Screenshot 7b

## 10. The product architecture

Mobile client applications (PDA, phone) are connected to the server via the FreeTime protocol, which is a set of commands and notifications in a textual form that are sent via HTTP-protocol. The Internet-client application is implemented in PHP and is accessible from all the browsers that support JavaScript and CSS. Client applications for personal computers can also be connected to the server via the FreeTime protocol.

The server consists of the following components:

### FTVM

The FreeTime virtual machine FTVM performs all operations on schedule editing. FTVM provides a high-level object-oriented interface (API) which is a set of abstract C++ classes. There are special-purpose classes for scheduling objects, and supplementary classes for work with companies, sessions, etc. This API allows a user to create several sessions simultaneously, to load data for different companies, to modify data, to obtain solutions using different solvers, to save a modified model and obtained solutions, to receive notifications about changes in the project and to combine the projects. To save/read data, FTVM interacts with the data provider SDataProvider, and to recalculate a schedule, it uses the calculation provider FTCalcProvider.

### FTSrv

The module FTSrv is the FreeTime server that contains implementation of the FreeTime protocol. It interacts with FTVM and provides maintenance, serialization and deserialization of objects, messages and notifications.

### FTX

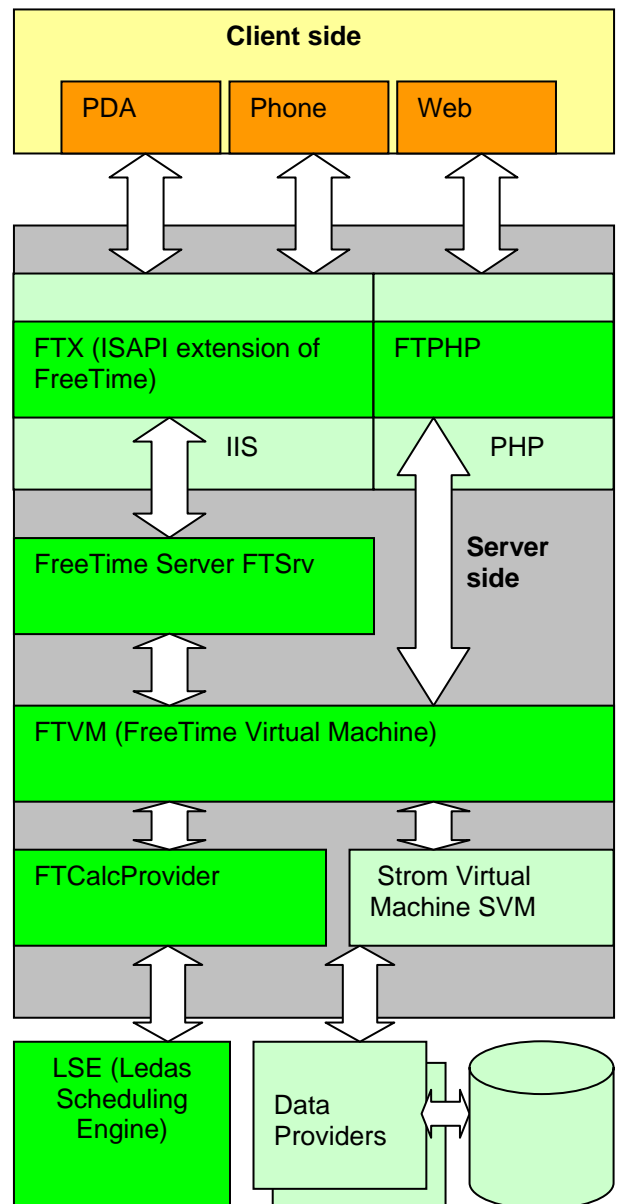
This module is an ISAPI extension for the IIS web-server. It presents implementation of the transport component of the FreeTime protocol and provides all needed transport functionality for FTSrv.

### FTPHP

This module implements an extension for PHP4. It allows all classes and methods of FTVM to be accessible in php-scripts that are used in the FreeTime web-client.

### FTCalcProvider

This is a set of modules for connection to different solvers. A module composes a task in terms of a concrete provider using the data from FTVM, starts computations and transfers the obtained solution to FTVM. At present, the provider for connection to the LSE computational engine (which is a part of the LEDAS Scheduler) is used.



## **SVM and SDataProvider**

For data manipulation and storage, a library Strom (Structured Object Modeler) is used, which consists of an SVM and of the set SDataProvider.

An SVM - virtual machine of Strom allows any structured data to be stored as objects with slots. The SVM also provides a high-level object-oriented programming interface.

SDataProvider is a set of modules that supports a common protocol of data exchange with the SVM module. Each module from SDataProvider provides access to a certain data format/source. At present, an SFileProvider module is implemented for data storage in the files of a special-purpose-format Strom.

## **11. Algorithms in the FreeTime core**

---

FreeTime is capable of automatically reacting to all new data entered by the participants of the meeting under its coordination. The schedule can thus be adopted with minimal manual operation and with less extra coordination between participants. This fundamental advantage of the product is based on the scheduling technology developed by LEDAS, in particular on the powerful LEDAS Scheduling Engine (LSE) incorporated into the FreeTime server.

After the user has made some changes in a schedule, he or she must synchronize them with the server, which calls the LSE computational kernel in order to check and resolve conflicts in a new schedule.

LSE is an independent scheduling engine, which is designed to be integrated into scheduling systems and tools of different complexity. Its interface functions allow to create a scheduling problem, which contains such entities as jobs (single or group), resources and resource pools, constraints (release/due dates, precedence relations) and assignments of resources on jobs. The engine works with these problems, trying to solve them with a variety of universal and specialized methods: a critical path algorithm, serial and stochastic serial methods, a conflict eliminator algorithm, branch and bound method, and so on.

The main purpose of LSE is to provide scalable and time-efficiency algorithms for resource-constrained project scheduling problems, which are NP-hard in the general case. Thus, LSE contains fast, optimal (i.e. providing true minimal completion time for the project), specialized algorithms for some particular cases and a special universal heuristic scheme for general case.

One of the fastest and most powerful methods is the serial scheme, which performs one-by-one scheduling of jobs using a greedy method with carefully developed efficient heuristics. Its modification – a stochastic serial method – allows one to get even more optimal solutions using a randomized heuristic-based way of output solution computation. All these methods are extremely fast – a problem with about 1000 jobs is solved in less than one second.

The main method used in FreeTime is an original conflict elimination method. It takes into account initial starting times on jobs, finds conflicts in this initial schedule (such as resource overloading or finishing a job after its due date) and locally shifts starting times to resolve these conflicts. It is very important that an output schedule is the closest to the initial schedule. That means that the solver will make as few changes as possible to resolve all conflicts and that the new schedule would not crucially change someone's plans. So, it provides excellent support for obtaining more "natural" solution. Another great advantage of this method is that it is very fast - its run-time and consumption of memory is quadratic with respect to the size of the initial problem.