

Разработка мобильных версий органайзера TimeMaster – техническое задание

Версия 0.3 от 10.04.2013

Лист изменений

[illegible]

Оглавление

Лист изменений	2
Термины и определения	6
1. Общая информация.....	7
1.1 Описание приложения	7
1.2 Ограничение на используемые технологии.....	7
1.3 Требования к используемым в приложениях ресурсам (строкам, картинкам и т.п.)	7
2. Основные сущности программы	8
2.1 Профиль пользователя	8
2.2 Задача	11
2.2.1 Основная сущность задачи	11
2.2.2 Напоминания	15
2.2.3 Рабочие группы.....	17
2.3 Контекст.....	17
2.4 Контакт	19
2.5 Файл.....	23
2.6 Рабочая группа.....	26
2.7 Дневник	29
2.8 Отношения между сущностями	30
3. Описания бизнес-процессов и алгоритмов в программе.....	33
3.1 Вход в систему	33
3.2 Создание пользователя	34
3.3 Создание задачи.....	35
3.4 Завершение задачи	36
3.5 Открыть завершенную ранее задачу	38
3.6 Редактирование задачи	Ошибка! Закладка не определена.
3.7 Удаление задачи.....	38
3.8 Создание контекста	38
3.9 Редактирование контекста	39
3.10 Удаление контекста.....	39
3.11 Создание контакта.....	40
3.12 Редактирование контакта	41
3.13 Удаление контакта	41

3.14	Создание записи дневника	41
3.15	Редактирование записи дневника	42
3.16	Удаление записи дневника	42
3.17	Создание рабочей группы	42
3.18	Добавление пользователя в рабочую группу	Ошибка! Закладка не определена.
3.19	Удаление пользователя из рабочей группы	Ошибка! Закладка не определена.
3.20	Комментирование задачи	43
3.21	Добавление файла	43
3.22	Редактирование файла	43
3.23	Удаление файла	43
3.24	Синхронизация данных	44
4.	Описания интерфейсов	45
4.1	Общая информация и требования к интерфейсам	47
4.2	Специфические требования для Android	48
4.3	Специфические требования для IOS	50
4.4	Специфические требования для Windows Phone 8	50
4.5	Стартовый экран	50
4.6	Вход в систему	50
4.7	Регистрация	51
4.8	Настройки	52
4.9	День	Ошибка! Закладка не определена.
4.10	Неделя	58
4.11	Месяц	59
4.12	ToDo	59
4.13	Контакты	60
4.14	Контексты	61
4.15	Файлы	61
4.16	Дневник	61
4.17	Завершенные задачи	62
5.	Этапы разработки	64
5.1	Первый этап	66
5.2	Второй этап	66
5.3	Третий этап	66
5.4	Четвертый этап	66
6.	Требования к разработке, документированию и оформлению кода	67

6.1	Требования к оформлению кода	67
6.2	Правила разработки	67

Термины и определения

[illegible]

1. Общая информация

1.1 Описание приложения

Приложение представляет собой версию органайзера TimeMaster (<http://time-master.ru/organizer>) для мобильных устройств.

Приложение работает без подключения к интернету со своим хранилищем данных и периодической синхронизацией данных с сервером time-master.

Данное задание разрабатывается для приложений на платформах IOS, Android и Windows Phone 8.

Все 3 приложения на перечисленных выше платформах имеют приблизительно одинаковый интерфейс, одинаковые функции и одинаковую логику работы.

1.2 Ограничение на используемые технологии

Приложение должно быть разработано для всех платформ на языке C#.

При этом для разработки под платформу Android предполагается использовать технологию Mono For Android <http://xamarin.com/monoformandroid>. Разработка должна вестись в среде Visual Studio 2010.

Для IOS должна использоваться технология Mono Touch <http://xamarin.com/monotouch>. В качестве среды разработки может быть использована Xamarin Studio <http://xamarin.com/studio> или Visual Studio 2010 с соответствующими плагинами от Xamarin.

Разработка для Windows Phone 8 должна вестись средствами Visual Studio 2010.

1.3 Требования к используемым в приложениях ресурсам (строкам, картинкам и т.п.)

Все используемые в приложении строки не должны быть «захардкожены». Следует использовать для хранения строк отдельные файлы ресурсов для простой поддержки множества языков в будущем.

2. Основные сущности программы

В данном разделе даны описания основных сущностей и их атрибутов.

Раздел служит для формирования общего представления о сущностях в разрабатываемом приложении и позволяет построить физическую модель данных для мобильного устройства.

Операции над сущностями, алгоритмы и бизнес-правила рассмотрены в разделе 3.

2.1 Профиль пользователя

Профиль пользователя – сущность, описывающая пользователя в системе. Помимо основных данных, таких как имя, срок окончания премиум-версии, количество доступного места для загрузки файлов и т.п. содержит дочерние сущности – список компаний в которых находится пользователь, контакты пользователя для напоминаний.

В отличие от других сущностей, профили и связанные с ними данные не существуют офлайн, то есть они создаются сразу на сервере и при работе с ними необходимо оперировать идентификаторами сервера.

Теоретически может возникнуть ситуация, когда существует профиль с одним и тем же именем на разных серверах синхронизации (например, основной сервер time-master и коробочное решение у одного из клиентов). Поэтому, профиль пользователя должен быть привязан к серверу синхронизации. При смене сервера синхронизации все связанные с профилем данные – задачи, контакты и т.п. должны быть удалены и заново получены с нового сервера.

Синхронизация профилей и связанных с ними данных соответственно сводится к получению последних актуальных данных о профиле с сервера.

Структура профиля и связанных с ним сущностей приведена на рисунке 2.1.

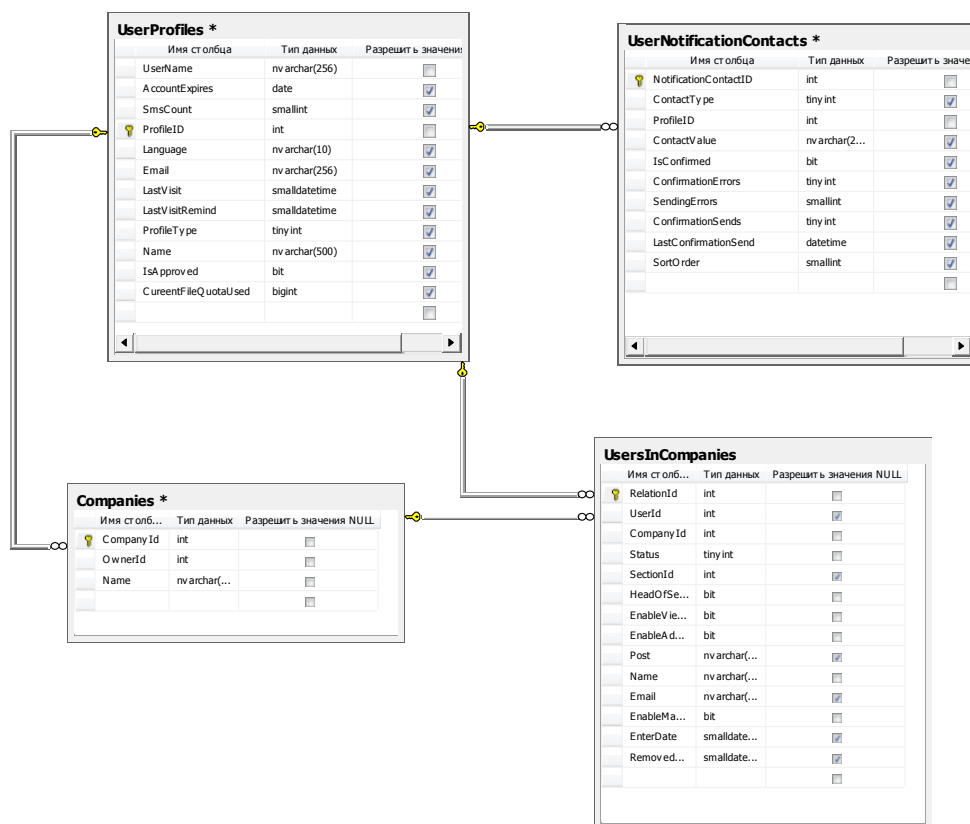


Рисунок 2.1 – Профиль пользователя и связанные с ним сущности

Пояснения к полям приведены ниже в таблицах

Таблица 2.1.1 – Описание полей для профиля на сервере (UserProfiles)

Название поля	Тип данных	Пояснение
UserName	Строка(256)	Имя пользователя. Используется для входа. В последних версиях совпадает с адресом электронной почты, но для более ранних пользователей это логин.
AccountExpires	Дата	Дата окончания срока действия премиум версии по московскому времени. Если текущая дата больше указанной – пользователь считается премиумом.
SmsCount	Число(макс. Значение 2000)	Количество СМС для напоминаний у пользователя. Используется, чтобы показать информацию о том, сколько осталось СМС при задании напоминания для задачи.
ProfileID	Число	ИД профиля на сервере
Language	Строка(10)	Язык, используемый пользователем. В настоящее время не используется пока никак.
Email	Строка(256)	Основной адрес электронной почты пользователя. Для новых пользователей совпадает с логином.

LastVisit	Дата и время	Дата и время последнего посещения
-----------	--------------	-----------------------------------

Таблица 2.1.2 – Описание контакта для напоминаний на сервере (UserNotificationContacts)

Название поля	Тип данных	Пояснение
NotificationContactID	Число	Уникальный идентификатор контакта для напоминания
ContactType	Байт	Возможные значения: 0 – Email 1 – SMS
ProfileID	Число	Идентификатор профиля, к которому относится контакт
ContactValue	Строка(256)	Значение контакта (т.е. адрес почты или телефон)
IsConfirmed	Булевый	Подтвержден ли контакт 1 – Контакт подтвержден и на него можно отправлять напоминания 0 – Контакт не подтвержден
ConfirmationErrors	Число (1 байт)	Количество ошибок при попытке подтвердить контакт
SendingErrors	Число (2 байта)	Количество ошибок при отправке сообщения для подтверждения
ConfirmationSends	Число (1 байт)	Количество отправок сообщения для подтверждения
LastConfirmationSend	Дата и время	Дата и время последней отправки сообщения для подтверждения
SortOrder	Число (2 байта)	Порядок сортировки в списке контактов для напоминаний

Таблица 2.1.3 – Описание компаний, принадлежащих пользователю (Companies)

Название поля	Тип данных	Пояснение
CompanyId	Число	Уникальный идентификатор компании, которой руководит пользователь
OwnerId	Число	Уникальный идентификатор пользователя. В мобильной версии должна быть предусмотрена возможность хранить в этом поле NULL т.к. на мобильное устройство в частности будет загружаться информация о компаниях, в которых пользователь не является владельцем.
Name	Строка(255)	Название компании

Таблица 2.1.4 – Описание пользователей в компаниях (UsersInCompanies)

Название поля	Тип данных	Пояснение
RelationId	Число	Уникальный идентификатор пользователя в компании
UserId	Число	Уникальный идентификатор пользователя
CompanyId	Число	Уникальный идентификатор компании
Status	Байт	Статус пользователя в компании: 1 – приглашен 2 – в компании 6 – удален

SectionId	Число	Не используется в текущей версии
HeadOfSection	Булевый	Не используется в текущей версии
EnableViewBills	Булевый	Не используется в текущей версии
EnableAddUsers	Булевый	Может ли данный пользователь добавлять в компанию других пользователей
Post	Строка(255)	Должность пользователя в компании
Name	Строка(255)	Имя пользователя в компании
Email	Строка(255)	Адрес электронной почты, если пользователь был приглашен по почте
EnableManage	Булевый	Не используется в текущей версии
EnterDate	Дата и время	Дата вступления в компанию
RemovedDate	Дата и время	Дата выхода из компании

Для поддержания механизмов синхронизации в структуру данных мобильной версии органайзера следует добавить поле **LocalProfileSyncDate**, которая будет содержать дату последней синхронизации профиля по московскому времени.

Также для профиля следует добавить поле **SyncServer**, которое будет содержать адрес сервера синхронизации (для пользователей коробочной версии органайзера). При смене адреса сервера синхронизации необходимо удалить все сохраненные для профиля данные. Это необходимо для поддержания корректности данных – если это переезд сервера, то данные будут выгружены в любом случае, а если именно смена сервера – удаление данных позволит сохранить корректность.

Если последняя дата синхронизации профиля меньше даты окончания премиум версии (поле AccountExpires) и текущая дата превышает значение поля AccountExpires более, чем на 7 дней (т.е. пользователь последний раз синхронизировался, когда у него был «премиум», затем не синхронизировался с момента окончания «премиума» 7 дней) – пользователь считается переведенным на стандартную версию.

Профиль пользователя и соответствующая информация не может никак изменяться внутри мобильного приложения. Она только подгружается с сервера при синхронизации. Единственное исключение – регистрация пользователя.

2.1.1 Задача

2.2.1 Основная сущность задачи

Задачи – основная и наиболее часто используемая сущность в системе. Представляет собой задачу пользователя (или задачу назначенную другому пользователю или полученную от другого пользователя).

Задачи имеют древовидную структуру, которая достигается за счет того, что каждая задача имеет поле ParentTaskId, которое содержит идентификатор родительской задачи. Если ParentTaskId не заполнен – задача находится на верхнем уровне дерева.

Поля задачи, хранимые на сервере, представлены на рисунке 2.2.1:

Tasks				
	Имя столбца	Тип данных	Длина	Разрешить значения NULL
	TaskId	int	4	☐
	UserName	nvarchar(256)	256	☐
	UserTaskId	int	4	☐
	Text	nvarchar(2000)	2000	☐
	FullText	ntext	16	☒
	Type	smallint	2	☒
	ParentTaskId	int	4	☒
	StartDate	date	3	☒
	EndDate	date	3	☒
	StartTime	time(7)	5	☒
	EndTime	time(7)	5	☒
	Length	int	4	☒
	FactLength	int	4	☒
	Period	int	4	☒
	PeriodUnit	smallint	2	☒
	Priority	smallint	2	☒
	Completed	bit	1	☒
	CompletedPercent	smallint	2	☒
	CompletedTime	datetime	8	☒
	Deleted	bit	1	☒
	SortOrder	int	4	☒
	CompletedInCurPeriod	bit	1	☒
	RepeatCount	int	4	☒
	RepeatOnMonday	bit	1	☒
	RepeatOnTuesday	bit	1	☒
	RepeatOnWednesday	bit	1	☒
	RepeatOnThursday	bit	1	☒
	RepeatOnFriday	bit	1	☒
	RepeatOnSaturday	bit	1	☒
	RepeatOnSunday	bit	1	☒
	DeepLevel	smallint	2	☒
	ChainChildren	bit	1	☒
	LastMod	smalldatetime	4	☒
	UID	nvarchar(255)	255	☒
	TimezoneOffset	smallint	2	☒
	OriginalTaskId	int	4	☒
	IsInput	bit	1	☒
	IsOutput	bit	1	☒
				☐

Рисунок 2.2.1 – поля сущности «Задача» на сервере

Пояснения к полям приведены ниже в таблице 2.2.1

Таблица 2.2.1 – Описание полей для задачи на сервере

Название поля	Тип данных	Пояснение
TaskId	число	Уникальный идентификатор задачи, предоставляемый сервером при сохранении задачи в базу данных сервера.
UserName	строка(256)	Имя пользователя (логин) автора задачи
UserTaskId	число	Идентификатор задачи пользователя (уникальный в рамках пользователя) предоставляемый сервером при сохранении задачи в базу данных сервера
Text	строка(2000)	Название задачи, обязательное поле
FullText	текст	Описание задачи
Type	число (2 байта)	Тип задачи. Обязательное поле. Допускает следующие значения: 1 – обычная задача 2 - проект (в настоящий момент не используется, просто зарезервировано)

		3 – событие Задачи и события отличаются с точки зрения бизнес-правил для обработки этих сущностей и поведением на интерфейсе. Более подробно эти отличия будут разобраны в разделах ниже.
ParentTaskId	число	Идентификатор родительской задачи. Используется для организации древовидной структуры задач.
StartDate	дата	Дата начала задачи.
EndDate	дата	Дата окончания задачи.
StartTime	время	Время начала задачи. Если дата задачи задана, а время нет, временем начала задачи считается начало дня.
EndTime	время	Время окончания задачи. Если дата окончания задано, а время окончания нет, временем окончания считается конец дня.
Length	число	Длительность задачи в минутах.
FactLength	число	Фактически затраченное пользователем на задачу время
Period	число	Период повторения (для повторяющихся событий)
PeriodUnit	число (2 байта)	Единица измерения для периода повторений Принимает следующие значения: 1 – час 24 – день 168 – неделя 4 – месяц 5 – год
Priority	число (2 байта)	Приоритет задачи. Принимает следующие значения 0 – обычный (значение по умолчанию) 1 – низкий 2 – высокий
Completed	булевое	Флаг, показывающий, завершена ли задача 0 - не завершена 1 - завершена
CompletedPercent	число (2 байта)	Процент завершения задачи. Принимает значение от 0 до 100. Если значение равно 100, задача считается завершенной и должен быть выставлен флаг Completed
CompletedTime	дата и время	Дата и время завершения задачи. Для повторяющегося события – время последнего завершеного повторения. (пояснения в последующих разделах)
Deleted	булевое	Флаг, показывающий, удалена ли задача 0 – не удалена 1 – удалена
SortOrder	число	Порядок сортировки задачи в дереве. При построении дерева задач задачи сортируются внутри каждого уровня по этому параметру
CompletedInCurPeriod	булевое	Не используется, оставлен для поддержки старых версий
RepeatCount	число	Максимальное количество повторений для повторяющегося события
RepeatOnMonday	булевое	Флаг, показывающий, будет ли задача повторяться по

Примечание [L1]: Решить как работать с этим полем

		понедельникам в случае задания недельного периода повторений с указанием конкретных дней
RepeatOnTuesday	булевое	Флаг, показывающий, будет ли задача повторяться по вторникам в случае задания недельного периода повторений с указанием конкретных дней
RepeatOnWednesday	булевое	Флаг, показывающий, будет ли задача повторяться по средам в случае задания недельного периода повторений с указанием конкретных дней
RepeatOnThursday	булевое	Флаг, показывающий, будет ли задача повторяться по четвергам в случае задания недельного периода повторений с указанием конкретных дней
RepeatOnFriday	булевое	Флаг, показывающий, будет ли задача повторяться по пятницам в случае задания недельного периода повторений с указанием конкретных дней
RepeatOnSaturday	булевое	Флаг, показывающий, будет ли задача повторяться по субботам в случае задания недельного периода повторений с указанием конкретных дней
RepeatOnSunday	булевое	Флаг, показывающий, будет ли задача повторяться по воскресеньям в случае задания недельного периода повторений с указанием конкретных дней
DeepLevel	число (2 байта)	Глубина задачи в дереве. На верхнем уровне будет 0, далее будет возрастать в зависимости от уровня. При построении дерева можно сортировать записи сначала по глубине, потом по порядку, чтобы иметь возможность строить дерево в обычном цикле
ChainChildren	булевое	Флаг, показывающий, должны ли подзадачи выполняться последовательно 1 – подзадачи будут выполняться последовательно 0 – подзадачи выполняются параллельно (значение по умолчанию)
LastMod	дата и время	Дата и время последнего изменения в задаче
UID	строка(255)	Уникальный идентификатор задачи, если она была импортирована из другой системы. Для задач созданных в самом тайм-мастере не заполняется.
TimezoneOffset	число (2 байта)	Отклонение часового пояса задачи от UTC0 в минутах. Например, для Москвы это -240
OriginalTaskId	число	ИД оригинальной задачи, если задача была поставлена другим пользователем. Имеется ввиду значение TaskID для пользователя, который поставил задачу.
IsInput	булевое	Флаг, является ли задача входящей 1 – является 0 – не является
IsOutput	булевое	Флаг, является ли задача исходящей 1 – является 0 – не является

Для поддержки логики работы на мобильном устройстве и синхронизации данных необходимо добавить следующие поля:

LocalTaskId – идентификатор задачи пользователя на мобильном устройстве. При создании задачи, которая еще не синхронизирована с сервером, задачи следует идентифицировать именно по этому полю. Соответственно, задачи, у которых есть LocalTaskId но нет TaskId не существуют на сервере и если такая задача была удалена – ее не следует синхронизировать.

LocalParentTaskId – идентификатор родительской задачи на мобильном устройстве. Служит для создания древовидной структуры для задач, которые не синхронизированы с сервером.

LocalCreateTime – дата и время создания сущности на мобильном устройстве. Используется, чтобы выстроить правильный порядок синхронизации объектов.

LocalChangeDateTime – дата и время последнего изменения задачи на мобильном устройстве. Должна храниться в московском часовом поясе (UTC +4)

2.2.2 Напоминания

Для задач должна быть реализована возможность задавать напоминания. Напоминания могут быть как локальные (только для мобильного устройства, не синхронизируются с сервером), так и синхронизируемые – в этом случае во время напоминания сервер пришлет по указанному адресу Email или SMS. Для одной задачи может быть задано произвольное количество напоминаний (рисунок 2.2.2).

Для напоминаний по аналогии с задачами следует добавить поля для поддержки синхронизации:

LocalRemindId – идентификатор напоминания на мобильном устройстве.

LocalTaskId – идентификатор задачи, для которой создается напоминание, на мобильном устройстве.

LocalCreateTime – дата и время создания сущности на мобильном устройстве. Используется, чтобы выстроить правильный порядок синхронизации объектов.

LocalChangeDateTime – дата и время последнего изменения напоминания на мобильном устройстве. Должна храниться в московском часовом поясе (UTC +4)

Для отправки напоминаний используются контакты пользователя, заданные в профиле. При этом контакты должны быть обязательно предварительно подтверждены.

Следует понимать, что напоминание, которое предполагает отправку SMS или Email и не синхронизировано с сервером не может быть отправлено т.к. сервер о нем ничего не знает. Необходимо предупреждать пользователя о таком моменте.

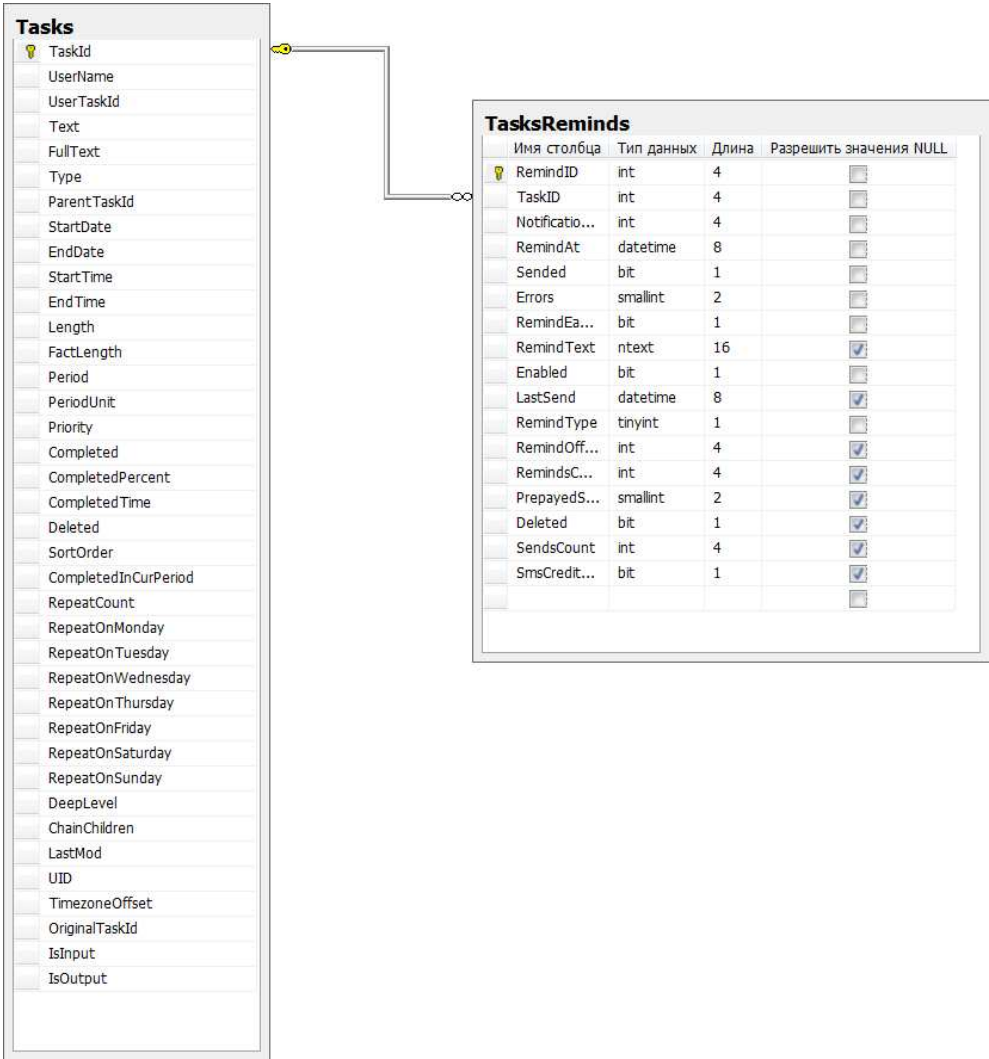


Рисунок 2.2.2 – Задачи и напоминания на сервере

Пояснения к полям сущности приведены в таблице 2.2.2.

Таблица 2.2 – Описание полей для напоминания на сервере

Название поля	Тип данных	Пояснение
RemindID	Число	Уникальный идентификатор напоминания на сервере
TaskID	Число	Серверный идентификатор задачи для которой задано напоминание
NotificationContactID	Число	Идентификатор контакта, для которого задано напоминание. (см. Таблица 2.1.2, строка 1)
RemindAt	Дата и время	Дата и время напоминания (заданное или рассчитанное) по Москве (+4)
Sended	Булевый	Флаг, отправлено ли напоминание 1 – отправлено

Errors	Число (2 байта)	Количество ошибок при отправке напоминания. Если ошибок больше 3 – напоминание больше не отправляется
RemindEachIteration	Булевый	Флаг, указывающий, следует ли отправлять напоминание для каждого повторения задачи. 1 – отправлять для каждого повторения 0 – ограничиться одним отправлением
RemindText	Текст	Текст напоминания
Enabled	Булевый	Флаг – активно ли напоминание 1 – напоминание активно 0 – напоминание отменено
LastSend	Дата и время	Дата и время последней отправки по москве
RemindType	Байт	Тип напоминания. Принимает одно из значений: 0 – напоминать в указанное время 1 – напоминать за указанный промежуток времени до начала задачи или события (при этом при сдвиге или переносе задачи или события время должно быть пересчитано)
RemindOffset	Число	Время в минутах до начала события, когда нужно отправить напоминание
RemindsCount	Число	Не используется
PrepayedSmsCount	Число (2 байта)	Количество SMS для отправки текста напоминания.
Deleted	Булевый	Флаг, удалено ли напоминание 1 – удалено 0 – не удалено
SendsCount	Число	Количество раз, когда напоминание было отправлено
SmsCreditsError	Булевый	Флаг, указывающий, что СМС напоминание не было отправлено из-за того, что у пользователя недостаточно СМС кредитов.

2.2.3 Рабочие группы

Рабочие группы используются для совместной работы над задачами с другими пользователями.

Подробнее сущность описана в разделе 2.6 Рабочая группа.

2.2 Контекст

Контексты в органайзере используются в основном для группировки различных связанных с ними сущностей. Можно привести синоним слову контекст – папка, метка, тег. Примеры использования:

1. Список задач по контексту
2. Список контактов по контексту
3. Список файлов по контексту.

Отличием можно считать то, что контексты имеют иерархическую структуру по аналогии с задачами, поэтому при поиске по ним применяются дополнительные правила, описанные в разделе 2.8.

Также, в зависимости от типа, контекст может быть связан с компанией, в которой находится пользователь. В этом случае при работе с контекстом существуют дополнительные правила, которые будут описаны в разделе 3.

Структура сущности контекста на сервере изображена на рисунке 2.2

Contexts			
Имя столб...	Тип данных	Разрешить значения NULL	
 ContextId	int	☐	
ContextNa...	nvarchar(...	☐	
UserName	nvarchar(...	☐	
Description	nvarchar(...	☒	
ParentCon...	int	☒	
SortOrder	int	☒	
Mode	smallint	☒	
Deleted	bit	☒	
DeepLevel	smallint	☒	
Company Id	int	☒	
O riginalId	int	☒	
IsCompany	bit	☒	
IsSection	bit	☒	
SectionTy pe	tiny int	☒	
		☐	

Рисунок 2.2 – Сущность «Контекст» на сервере

Пояснения к полям сущности приведены в таблице 2.2

Таблица 2.2 – Описание полей для сущности контекст на сервере

Название поля	Тип данных	Пояснение
ContextId	Число	Уникальный идентификатор контекста на сервере
ContextName	Строка(256)	Название контекста. Обязательное поле
UserName	Строка(256)	Имя пользователя(логин) автора контекста
Description	Строка(1000)	Описание контекста
ParentContextId	Число	Идентификатор родительского контекста на сервере. Если принимает значение NULL – контекст находится на верхнем уровне иерархии.
SortOrder	Число	Порядок сортировки контекста внутри заданной ветки. При построении дерева контекстов записи сортируются внутри ветки по этому полю. Значения должны быть от 0 по порядку.
Mode	Число(2 байта)	Не используется в текущей версии
Deleted	Булевый	Флаг, показывающий, что контекст удален
DeepLevel	Число(2 байта)	Глубина контекста в дереве. На верхнем уровне будет 0, далее по нарастающей
CompanyId	Число	Идентификатор связанной компании
OriginalId	Число	Идентификатор оригинального контекста на сервере, копией которого является этот контекст (применяется, если это контекст относится к компании и был создан другим

		пользователем компании)
IsCompany	Булевый	Флаг, является ли контекст «компанией»
IsSection	Булевый	Флаг, является ли контекст «отделом»
SectionType	Число(1 байт)	Не используется в текущей версии

Для поддержки логики работы на мобильном устройстве и синхронизации данных необходимо добавить следующие поля:

LocalContextId – идентификатор контекста пользователя на мобильном устройстве. При создании контекста, который еще не синхронизирован с сервером, контексты следует идентифицировать именно по этому полю. Соответственно, контексты, у которых есть LocalContextId но нет ContextId не существуют на сервере и если такой контекст был удален – его не следует синхронизировать.

LocalParentContextId – идентификатор родительского контекста на мобильном устройстве. Служит для создания древовидной структуры для контекстов, которые не синхронизированы с сервером.

LocalCreateDateTime – дата и время создания сущности на мобильном устройстве. Используется, чтобы выстроить правильный порядок синхронизации объектов.

LocalChangeDateTime – дата и время последнего изменения контекста на мобильном устройстве. Должна храниться в московском часовом поясе (UTC +4)

2.3 Контакт

2.4.1 Основная сущность контакта

Контакты используются в органайзере для нескольких функций:

1. Записная книжка, где хранится список контактов, с которыми работает пользователь
2. Группировка задач по контакту, чтобы видеть список задач по конкретному человеку или нескольким людям
3. Через контакты происходит постановка задач людям, с которыми вы работаете в рамках компаний в тайм-мастере.

Поля контакта, хранимые на сервере, представлены на рисунке 2.4.1.

Contacts				
Имя столбца	Тип данных	Длина	Разрешить значения NULL	
 ContactId	int	4	☐	
UserName	nvarchar(...	256	☐	
ContactNa...	nvarchar(...	500	☐	
Description	nvarchar(...	2000	☒	
Photo	nvarchar(...	256	☒	
SortOrder	int	4	☒	
Deleted	bit	1	☒	
TmProfileId	int	4	☒	
BirthDay	tinyint	1	☒	
BirthMonth	tinyint	1	☒	
BirthYear	smallint	2	☒	
CorpStatus	tinyint	1	☒	
			☐	

Рис. 2.4.1 – поля сущности «контакт» на сервере

Пояснения к полям приведены ниже в таблице 2.4.1

Таблица 2.4.1 – Описание полей для контакта на сервере

Название поля	Тип данных	Пояснение
ContactId	Число	Уникальный идентификатор контакта на сервере
UserName	Строка(256)	Имя пользователя (логин) владельца контакта
ContactName	Строка(500)	Наименование контакта (обязательное поле)
Description	Строка(2000)	Описание контакта
Photo	Строка(256)	Ссылка на фото контакта. Задается как относительный адрес для сервера. То есть например /upload/contactphotos/lap1n_5504.jpg Соответственно, для получения картинки необходимо перейти по ссылке http://time-master.ru/upload/contactphotos/lap1n_5504.jpg?t=23082012110954
SortOrder	Число	Порядок сортировки контакта в общем списке. Для каждого контакта должно быть свое уникальное число, начиная с 0 по порядку
Deleted	Булевое	Флаг, удален ли контакт 1 – удален 0 – не удален
TmProfileId	Число	ИД профиля пользователя, с которым связан контакт. Если задано – это корпоративный контакт и ему можно ставить задачи.
BirthDay	Число(1 байт)	Дата рождения контакта
BirthMonth	Число(1 байт)	Месяц рождения контакта
BirthYear	Число(2 байт)	Год рождения контакта

	байта)	
CorpStatus	Число(1 байт)	Статус контакта для совместной работы. Принимает одно из следующих значений: 0 – личный контакт 1 – приглашенный для совместной работы пользователь 2 – корпоративный контакт 3 – контакт, который был корпоративным, но удален из компании.

Для поддержки логики работы на мобильном устройстве и синхронизации данных необходимо добавить следующие поля:

LocalContactId – идентификатор контакта пользователя на мобильном устройстве. При создании контакта, который еще не синхронизирован с сервером, контакт следует идентифицировать именно по этому полю. Соответственно, контакты, у которых есть LocalContactId но нет ContactId не существуют на сервере и если такой контакт был удален – его не следует синхронизировать.

LocalPhotoPath – путь к фото контакта на мобильном устройстве. Фото контакта сохраняется на мобильном устройстве при первой загрузке информации о контакте. В последующем фото может быть изменено и тогда оно будет сначала загружено на сервер, потом скачано оттуда обработанным т.е. процесс получения фото с сервера повторится.

LocalCreateDateTime – дата и время создания сущности на мобильном устройстве. Используется, чтобы выстроить правильный порядок синхронизации объектов.

LocalChangeDateTime – дата и время последнего изменения контакта на мобильном устройстве. Должна храниться в московском часовом поясе (UTC +4).

2.4.2 Контактные данные

Для контакта может быть задан некоторый набор контактных данных – телефон, адрес электронной почты и т.п.

Каждая запись контактного данного характеризуется типом, который определяет, что это за запись. Для большинства типов контактных данных название определяется именно по типу, но в некоторых случаях пользователь может задать собственный тип данных, указав его название.

Поля контактных данных, хранимые на сервере, представлены на рисунке 2.4.2.

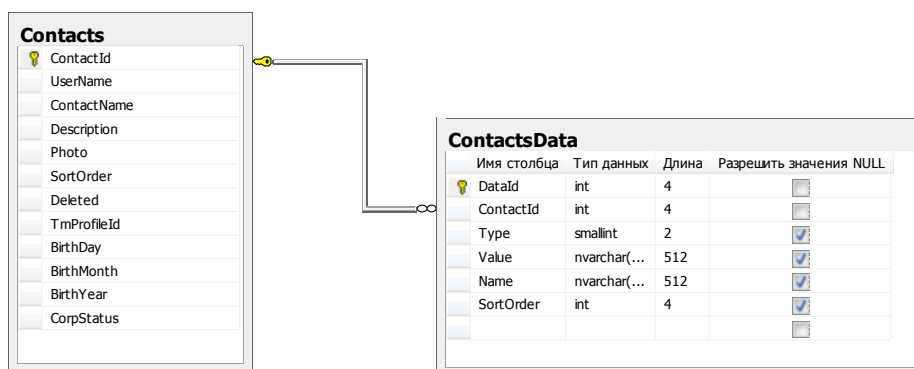


Рис. 2.4.2 – поля сущности «контакт» на сервере

Пояснения к полям приведены ниже в таблице 2.4.2

Таблица 2.4.2 – Описание полей для данных контакта на сервере

Название поля	Тип данных	Пояснение
DataId	Число	Уникальный идентификатор контактного данного на сервере
ContactId	Число	Идентификатор контакта, к которому относятся контактные данные
Type	Число(2 байта)	Тип контактного данного. Принимает одно из следующих значений: 0 – мобильный телефон 5 – рабочий телефон 10 - Домашний телефон 15 - Мобильный телефон 2 20 - Рабочий телефон 2 25 - Домашний телефон 2 30 - Факс 35 - Рабочий факс 50 - Email 55 - Личный Email 60 - Рабочий Email 2 65 - Рабочий Email 70 - Личный Email 2 100 - ICQ 105 - QIP 110 - SKYPE 115 - Mail Agent 120 - MSN 125 - Jabber 130 - ВКонтакте 135 - Live Journal 140 - Yandex Online 145 - Google Talk 150 - Facebook 155 - Twitter 160 - Личный сайт 165 - Личный сайт 2 250 - Адрес 255 - Адрес 2 260 - Юридический адрес 261 - Должность 265 – Компания 200 -Собственное поле 1 201 -Собственное поле 2 202 -Собственное поле 3 203 -Собственное поле 4 204 -Собственное поле 5 205 -Собственное поле 6
Value	Строка(512)	Значение контактного данного
Name	Строка(512)	Имя контактного данного, дополнительно заданного пользователем (напр. Номер паспорта)

SortOrder	Число	Порядок сортировки при выводе списка контактных данных
-----------	-------	--

Для поддержки логики работы на мобильном устройстве и синхронизации данных необходимо добавить следующие поля:

LocalContactDataId – идентификатор записи контактного данного на мобильном устройстве.

LocalContactId – идентификатор контакта пользователя на мобильном устройстве.

LocalCreateDateTime – дата и время создания сущности на мобильном устройстве. Используется, чтобы выстроить правильный порядок синхронизации объектов.

LocalChangeDateTime – дата и время последнего изменения контакта на мобильном устройстве. Должна храниться в московском часовом поясе (UTC +4).

Deleted – флаг удаления на мобильном устройстве. Требуется для синхронизации удаления контактных данных.

2.4 Файл

Файлы представляют собой загружаемые в органайзер пользователем файлы. Сущность помимо самого файла содержит его название, описание, тип контента и служебную информацию о размере, хеш-сумме и т.п.

Данная сущность в мобильной версии будет больше всего отличаться от серверной версии, так как практически будет использоваться два хранилища файлов – локальное и серверное.

Для файла ведется история изменений, причем файл последней версии доступен локально, а все остальные версии файла загружаются с сервера по запросу. Для записей в истории изменений остается тоже имя файла, что и для оригинального. Также для файлов из истории изменений применяются те же правила доступа, что и для оригинального файла, если файл был расшарен другим пользователем.

При синхронизации с сервера происходит только загрузка списка файлов. Сами файлы загружаются при непосредственном обращении к указанному файлу. Загрузка файлов, удаление и обновление файлов на сервере наоборот происходит сразу.

Поля для файлов, хранимые на сервере, представлены на рисунке:

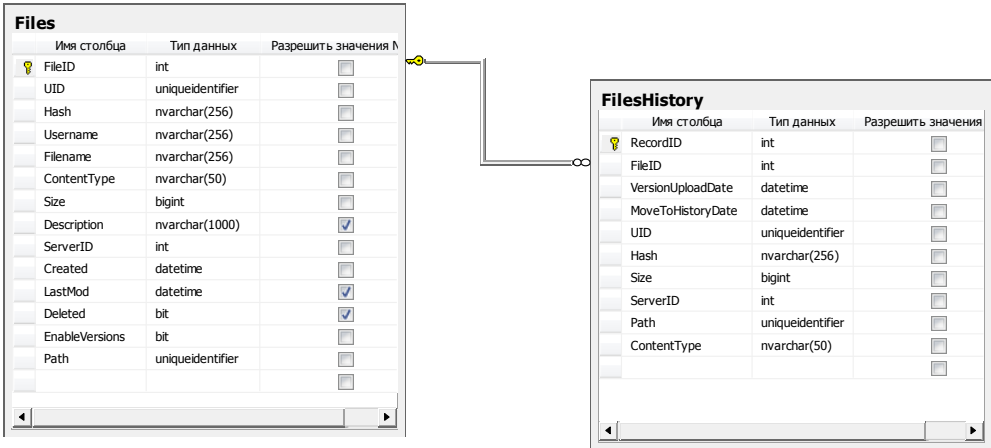


Рисунок 2.4 – поля сущности «файл» на сервере

Пояснения к полям приведены в таблицах 2.4.1 и 2.4.2.

Таблица 2.4.1 – Описание полей сущности «Файл» на сервере

Название поля	Тип данных	Пояснение
FileID	Число	Идентификатор файла на сервере
UID	Уникальный идентификатор	Уникальный идентификатор, присвоенный файлу для обращения к нему на сервере
Hash	Строка(256)	Хеш-сумма файла
Username	Строка(256)	Имя владельца файла
Filename	Строка(256)	Название файла
ContentType	Строка(50)	Тип контента
Size	Число (8байт)	Размер файла
Description	Строка(1000)	Описание файла
ServerID	Число	Идентификатор сервера, на котором хранится файл. В мобильной версии этот идентификатор никак не используется
Created	Дата и время	Дата создания файла по московскому времени
LastMod	Дата и время	Дата последнего изменения файла по московскому времени
Deleted	Булевый	Флаг удаления файла 1 – удален 0 – не удален
EnableVersions	Булевый	Флаг, показывающий, должна ли храниться история изменений файла 1 – должна храниться 0 – не должна
Path	Уникальный идентификатор	Уникальный идентификатор пути к файлу на сервере

Помимо этих полей, для поддержания работы мобильной версии должны быть добавлены поля:

LocalFileID – уникальный идентификатор файла на мобильном устройстве.

LocalCreateDateTime – дата и время создания сущности на мобильном устройстве. Используется, чтобы выстроить правильный порядок синхронизации объектов.

LocalChangeDateTime – дата и время последнего изменения записи по московскому времени.

LocalPath – путь к файлу на мобильном устройстве.

LocalFileHash – хеш-сумма файла на мобильном устройстве.

SharedBy – идентификатор пользователя, который предоставил доступ к файлу

Checksum – контрольная сумма для операций с файлом

FileServerUrl – путь к серверу, на котором физически хранится файл.

Таблица 2.4.2 – Описание полей сущности «История изменений файла» на сервере

Название поля	Тип данных	Пояснение
RecordID	Число	Уникальный идентификатор версии файла на сервере
FileID	Число	Уникальный идентификатор файла на сервере
VersionUploadDate	Дата и время	Дата и время загрузки этой версии файла
MoveToHistoryDate	Дата и время	Дата и время перемещения версии файла в историю
UID	Уникальный идентификатор	Уникальный идентификатор, присвоенный файлу для обращения к нему на сервере
Hash	Строка(256)	Хеш-сумма файла
Size	Число(16байт)	Размер файла
ServerID	Число	Идентификатор сервера, на котором хранится файл. В мобильной версии этот идентификатор никак не используется
Path	Уникальный идентификатор	Уникальный идентификатор пути к файлу на сервере
ContentType	Строка(50)	Тип контента

Помимо этих полей, для поддержания работы мобильной версии должны быть добавлены поля:

LocalRecordID – уникальный идентификатор версии файла на мобильном устройстве

LocalFileID – уникальный идентификатор файла на мобильном устройстве.

LocalCreateDateTime – дата и время создания сущности на мобильном устройстве. Используется, чтобы выстроить правильный порядок синхронизации объектов.

LocalChangeDateTime – дата и время последнего изменения записи по московскому времени.

LocalPath – путь к файлу на мобильном устройстве.

LocalFileHash – хеш-сумма файла на мобильном устройстве.

Checksum – контрольная сумма для операций с файлом

FileServerUrl – путь к серверу, на котором физически хранится файл.

2.5 Рабочая группа

В данном разделе рассмотрены сущности для поддержания корпоративных функций органайзера.

В настоящее время корпоративные функции включают в себя

- возможность ставить задачи пользователям (сущность «рабочая группа»)
- возможность обсуждать задачи в рамках рабочей группы (сущность «сообщение»)

Сущности построены следующим образом: для задачи, которую один пользователь хочет поставить другому, создается рабочая группа. Другие пользователи, которые назначаются работать с задачей, являются участниками группы. Принимая задачу, пользователь создает у себя ее копию с указанием на оригинальную задачу (поле OriginalTaskId) и устанавливает свой статус в группе как работающего над задачей.

Рабочие группы и связанные с ними сущности изображены на рисунке 2.5.1.



Рисунок 2.5.1 – Рабочие группы и их связи с другими сущностями

Пояснения к полям приведены ниже в таблице 2.5.1

Таблица 2.5.1 – Описание полей для данных сущности Рабочая группа

Название поля	Тип данных	Пояснение
GroupId	Число	Уникальный идентификатор группы на сервере
OwnerId	Число	Идентификатор профиля владельца группы (того, кто делегировал задачу, создав таким образом группу)
State	Число(1 байт)	Состояние группы. Принимает одно из следующих значений: 0 – создана 1 – в работе 2 – завершена

		3 – закрыта 4 – отменена 5 – нет участников 10 – удалена
Name	Строка(256)	Название группы
TaskId	Число	Идентификатор задачи на сервере, на основе которой была создана группа.

Для рабочих групп в мобильной версии должны также храниться следующие параметры:

LocalTaskID – идентификатор задачи к которой была создана группа на мобильном устройстве.

LocalWorkgroupID – уникальный идентификатор группы на мобильном устройстве.

LocalCreateDateTime – дата и время создания сущности на мобильном устройстве. Используется, чтобы выстроить правильный порядок синхронизации объектов.

LocalChangeDateTime – дата и время последнего изменения записи по московскому времени.

Пояснения к полям приведены ниже в таблице 2.5.2

Таблица 2.5.2 – Описание полей для данных сущности Члены рабочей группы

Название поля	Тип данных	Пояснение
MemberId	Число	Уникальный идентификатор участника группы на сервере
GroupId	Число	Уникальный идентификатор связанной группы на сервере
UserId	Число	Уникальный идентификатор профиля участника группы на сервере
UserInCompanyId	Число	Уникальный идентификатор пользователя в компании (сущность UsersInCompanies, таблица 2.1.4)
State	Число(1байт)	Статус участника группы. Принимает одно из следующих значений: 0 – приглашен 1 – работает 2 – отклонил приглашение 10 - удален
Role	Число(1байт)	Роль пользователя в группе 0 – владелец группы 1 – участник группы
JoinDate	Дата и время	Дата и время присоединения к группе по московскому времени
LastReadMsg	Число	Идентификатор последнего прочитанного сообщения об обновлениях в группе

При добавлении участников в группу на мобильном устройстве используется серверный идентификатор пользователя в системе.

Необходимо также добавить в мобильной версии следующие поля для сущности:

LocalMemberID – уникальный идентификатор участника группы на мобильном устройстве

LocalWorkgroupID – уникальный идентификатор группы на мобильном устройстве.

LocalCreateDateTime – дата и время создания сущности на мобильном устройстве. Используется, чтобы выстроить правильный порядок синхронизации объектов.

LocalChangeDateTime – дата и время последнего изменения записи по московскому времени.

Пояснения к полям приведены ниже в таблице 2.5.3

Таблица 2.5.3 – Описание полей для данных сущности Сообщение

Название поля	Тип данных	Пояснение
MessageId	Число	Уникальный идентификатор сообщения на сервере
WorkgroupId	Число	Уникальный идентификатор рабочей группы на сервере
FromUser	Число	Уникальный идентификатора профиля пользователя – автора сообщения
ToUser	Число	Уникальный идентификатор профиля пользователя, которому адресуется сообщение. В настоящий момент не используется.
Type	Число(1байт)	Тип сообщений 0 – личное (сейчас не используется) 1 – сообщение в группу
Date	Дата и время	Дата и время создания сообщения по Москве
Status	Число(1байт)	Статус сообщения 0 – сообщение отправлено
Text	Текст(до 5000 символов)	Текст сообщения
DiscussionId	Число	Уникальный идентификатор обсуждения. Не используется в текущей версии.

При наличии подключения к интернету сообщения в обсуждениях задач должны отправляться мгновенно. При отсутствии подключения – они накапливаются и отправляются с указанием фактической даты отправки.

Для поддержки синхронизации в мобильной версии необходимо добавить к сущности следующие поля:

LocalMessageID – уникальный идентификатор сообщения на мобильном устройстве

LocalWorkgroupID – уникальный идентификатор группы на мобильном устройстве.

LocalCreateDateTime – дата и время создания сущности на мобильном устройстве. Используется, чтобы выстроить правильный порядок синхронизации объектов.

LocalChangeDateTime – дата и время последнего изменения записи по московскому времени.

2.6 Дневник

Дневник позволяет записывать, чем пользователь занимался в течение дня для дальнейшего учета времени. При выполнении задач из списка задач пользователя дневник заполняется автоматически.

Поля для записей дневника, хранимые на сервере, представлены на рисунке 2.6

DiaryRecords			
	Имя столбца	Тип данных	Разрешить значения NULL
	 RecordId	int	☐
	Username	nvarchar(...	☐
	Date	date	☐
	StartTime	time(7)	☒
	EndTime	time(7)	☒
	TaskId	int	☒
	IsWasteTime	bit	☒
	WasCompleted	bit	☒
	RecordText	nvarchar(...	☐
	FullText	ntext	☒
			☐

Рисунок 2.6 – поля сущности «запись дневника»

Пояснения к полям приведены в таблицах 2.6.

Таблица 2.4.1 – Описание полей сущности «Файл» на сервере

Название поля	Тип данных	Пояснение
RecordId	Число	Уникальный идентификатор записи в журнале на сервере
Username	Строка(256)	Имя пользователя (логин) владельца записи
Date	Дата	Дата записываемого события по московскому времени(+4)
StartTime	Время	Время начала записываемого события
EndTime	Время	Время окончания записываемого события
TaskId	Число	Идентификатор связанной задачи на сервере
IsWasteTime	Булевый	Флаг – является ли указанный промежуток пустой тратой времени 1 – является 0 – не является
WasCompleted	Булевый	Флаг – была ли указанная задача завершена в данный промежуток времени 1- была завершена 0 – не была завершена
RecordText	Строка(2000)	Текст записи в дневник
FullText	Строка(5000)	Текст с комментариями

Для поддержки механизмов синхронизации в мобильной версии необходимо будет добавить следующие поля

LocalRecordId – идентификатор записи дневника пользователя на мобильном устройстве. При создании записи, которая еще не синхронизирована с сервером, запись следует идентифицировать именно по этому полю. Соответственно, записи, у которых есть LocalRecordId но нет RecordId не существуют на сервере и если такая запись была удалена – её не следует синхронизировать.

Deleted – флаг удаления для записей дневника на мобильном устройстве. Должен заполняться значением «1» если запись, которая существует на сервере, удалена, но еще не синхронизирована.

LocalCreateDateTime – дата и время создания сущности на мобильном устройстве. Используется, чтобы выстроить правильный порядок синхронизации объектов.

LocalChangeDateTime – дата и время последнего изменения записи дневника на мобильном устройстве. Должна храниться в московском часовом поясе (UTC +4).

2.8 Отношения между сущностями

Для связей между сущностями задача-контекст, контекст-контакт, задача-контакт, файл-задача, файл-контекст и файл-контакт на сервере используются вспомогательные сущности, которые облегчают поиск связей.

Все 6 сущностей имеют одинаковую структуру. Рассмотрим ее на примере задача-контекст. (рисунок 2.8).

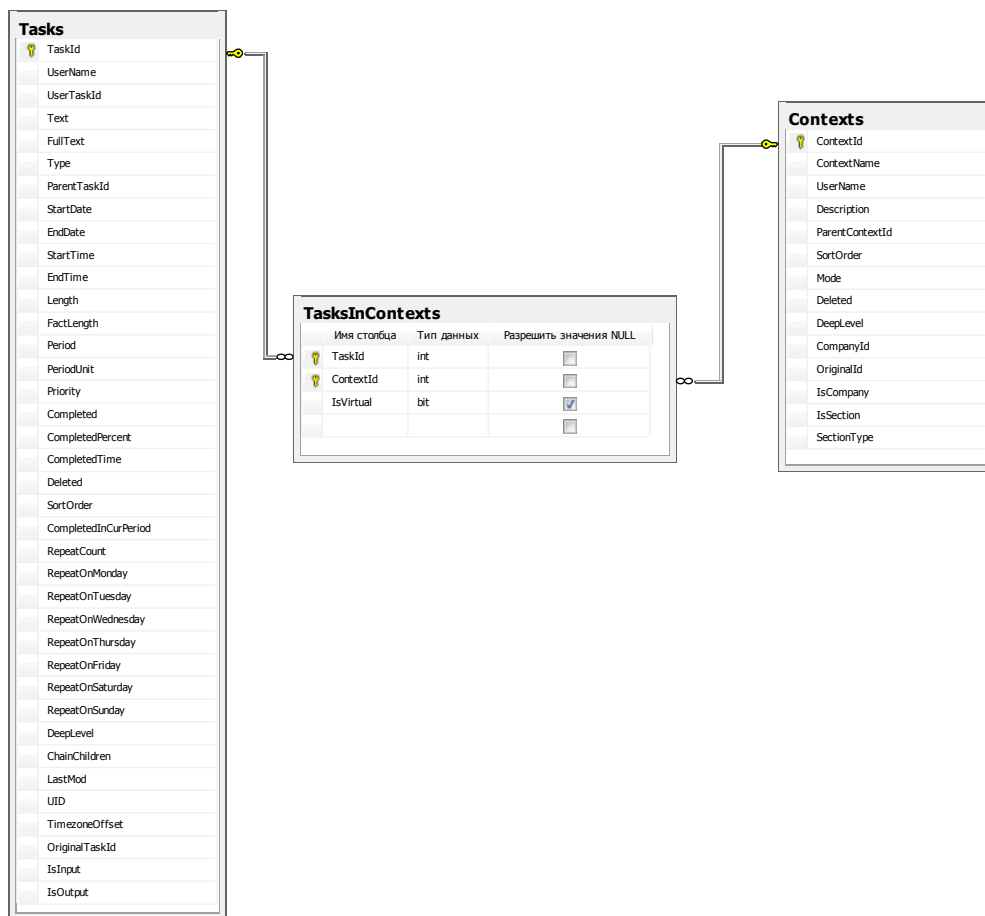


Рисунок 2.8 – отношение между задачей и контекстом.

Структура содержит следующие поля:

- Идентификатор первой сущности в связке (в примере TaskID) – число
- Идентификатор второй сущности в связке (в примере ContextID) – число
- Флаг для поддержки «виртуальной связи» - IsVirtual – булевый.

Флаг виртуальных связей используется для того, чтобы определять сущности, которые связаны не напрямую, а за счет иерархии.

Например:

- если контакт относится к задаче, он относится ко всем ее подзадачам (для подзадач будет помечен как виртуальный). То есть для задач направление движения сверху вниз по глубине иерархии.

- если контакт относится к контексту – он относится к его родительским контекстам (для родительских контекстов будет помечен как виртуальный). То есть для контекстов направление движения снизу вверх по иерархии.

Для файлов и контактов иерархия не предусмотрена, поэтому для связи файл-контакт флаг поддержки виртуальной связи можно игнорировать.

Примеры использования и дополнительное описание можно посмотреть по ссылке http://time-master.ru/organizer/help/green_tasks/

Непосредственно сами связи не синхронизируются отдельно, они появляются при синхронизации основных сущностей. Поэтому, **для поддержки синхронизации для создаваемых в мобильном приложении сущностей следует использовать локальные идентификаторы сущностей**, такие как LocalTaskId, LocalContextId и т.п.

При синхронизации по локальным идентификаторам должны быть вычислены серверные, которые и будут использованы при обмене информацией.

3. Описания бизнес-процессов и алгоритмов в программе

В разделе описаны основные процессы в системе, алгоритмы этих процессов и правила валидирования (проверки на корректность) данных, вводимых пользователем.

3.1 Начальная загрузка приложения

Прототип стартового экрана приведен в разделе 4.5.

В случае если запуск приложения осуществляется впервые:

1. Происходит создание структуры данных в соответствии с последней версией приложения т.е. создается структура данных от первого релиза и последовательно выполняются все скрипты, обновляющие структуру. В это время пользователю показывается экран с сообщением «подождите, идет подготовка к первому запуску приложения».
2. Пользователю отображается 2 вкладки – вход в систему и регистрация (см. разделы 3.2 и 3.3 соответственно).

В случае если приложение уже запускалось ранее:

1. При необходимости (если это первый запуск после обновления) последовательно выполняются все скрипты, обновляющие структуру данных. В это время пользователю показывается экран «подождите, идет установка обновлений».
2. Приложение проверяет, был ли выполнен вход в органайзер:
 - а. Если ранее пользователем был выполнен вход в органайзер, он переходит к разделу «Рабочий день».
 - б. Если же вход в органайзер не был выполнен, пользователю отображаются 2 вкладки – вход в систему и регистрация (см. разделы 3.2 и 3.3 соответственно).

3.2 Вход в систему

Прототип экрана для входа в систему приведен в разделе 4.6.

Вход в систему осуществляется при вводе логина (или адреса почты) и пароля. Оба поля являются обязательными.

Если при нажатии на кнопку «Вход» не заполнено какое-то из обязательных полей, это поле или поля подсвечиваются красным цветом.

Если оба поля заполнены, после нажатия на кнопку «Вход» происходит следующее:

Базовый сценарий действий:

1. Приложение отправляет HTTP запрос по основному адресу: <http://time-master.ru/api/mobile/auth.ashx> или же адресу сервера синхронизации, если он указан, методом POST с параметрами: login = указанный логин, password = указанный пароль.
2. Приложение получает ответ в формате **JSON**.
3. Приложение проверяет, что результат запроса (поле Success) имеет значение true.

Примечание [L2]: Согласовать формат взаимодействия

4. Приложение сохраняет параметр `AuthToken`, полученный в результате запроса, и использует полученное значение при последующих запросах к серверу, передавая Cookie с именем `LMUser` и значением `AuthToken`.
5. Пользователь переходит в раздел «Рабочий день».
6. Выполнение сценария завершено.

Примечание [L3]: Как лучше передавать – в cookie или может быть заголовком запроса? Как это реализуется в мобильных приложениях?

Альтернативный сценарий 1: если на шаге 2 базового сценария сервер вернул ошибку или не удалось получить ответ на запрос

1. Пользователю отображается сообщение об ошибке: «Произошла ошибка при попытке входа в органайзер. Проверьте, что Вы подключены к интернету, и повторите попытку».
2. Выполнение сценария завершено.

Альтернативный сценарий 2: если на шаге 3 базового сценария результат поля `Success` имеет значение `false`

1. Приложение отображает текст ошибки, передаваемый пользователю в поле `ErrorMessage`.
2. Выполнение сценария завершено.

3.3 Создание пользователя (регистрация)

Прототип экрана для входа в систему приведен в разделе 4.7.

Пользователю отображается форма, где предлагается ввести следующие поля (все поля обязательные для заполнения)

- Адрес электронной почты – должен быть в формате адреса EMAIL описанного в соответствующей документации
- Пароль. Не менее 6 символов длиной.
- Повторение пароля, должно совпадать с паролем.

При нарушении правил ввода данных поле, правило для которого нарушено, подсвечивается красным цветом и отображается соответствующее сообщение (например, пароль должен быть не менее 6 символов).

После нажатия на кнопку «Регистрация» происходит следующее:

Базовый сценарий действий:

1. Приложение отправляет HTTP запрос по адресу: <http://time-master.ru/api/mobile/register.ashx> методом POST с параметрами: `login` = указанный адрес электронной почты, `password` = указанный пароль, `password2` = повтор пароля.
2. Приложение получает ответ в формате `JSON`.
3. Приложение проверяет, что результат запроса (поле `Success`) имеет значение `true`.
4. Приложение сохраняет параметр `AuthToken`, полученный в результате запроса, и использует полученное значение при последующих запросах к серверу, передавая Cookie с именем `LMUser` и значением `AuthToken`.
5. Пользователь переходит в раздел «Рабочий день».
6. Выполнение сценария завершено.

Примечание [L4]: JSON или XML?

Примечание [L5]: Может быть проще сразу сохранить куки? Как лучше?

Альтернативный сценарий 1: если на шаге 2 базового сценария сервер вернул ошибку или не удалось получить ответ на запрос

1. Пользователю отображается сообщение об ошибке: «Произошла ошибка при попытке регистрации в органайзере. Проверьте, что Вы подключены к интернету, и повторите попытку».
2. Выполнение сценария завершено.

Альтернативный сценарий 2: если на шаге 3 базового сценария результат поля Success имеет значение false

1. Приложение отображает текст ошибки, передаваемый пользователю в поле ErrorMessage.
2. Выполнение сценария завершено.

3.4 Создание задачи

Задача может быть создана из разных элементов интерфейса:

- Быстрое добавление задачи, когда вводится только текст
- Форма для создания и редактирования задач (4.17)
- Добавление задач выделением области на календаре (4.9, 4.10, 4.11)

В любом случае в одной условной транзакции к базе данных должна быть создана задача со всеми зависящими от нее сущностями – связями с контекстами, контактами, файлами, напоминаниями, рабочими группами и т.п.

Все проверки на корректность полей задачи и связанных сущностей должны быть выполнены до сохранения данных в БД. При не прохождении хотя бы одной проверки сценарий создания задачи прерывается и пользователю отображается сообщение об ошибке.

Базовый сценарий создания задачи:

1. Приложение проверяет, что пользователь авторизован
2. Приложение проверяет, что пользователь может создавать задачи (должен быть премиум-пользователем или же иметь менее 30 активных задач)
3. Приложение проверяет корректность заполнения названия задачи – название должно быть заполнено
4. Если указана родительская задача, приложение проверяет, что такая задача существует и активна
5. Если для задачи указан тип «задана» и для нее заданы дата или время, приложение проверяет, что дата и время заданы корректно:
 - a. Дата начала задачи не должна быть больше даты и времени окончания
 - b. Время для задачи должно быть в пределах от 00:00 до 23:59
 - c. Не возникает ситуации, когда задано время, но не задана дата
6. Если для задачи указан тип «событие», приложение проверяет, что корректно указаны временные параметры события:
 - a. Обязательно должна быть указана дата начала события (для повторяющихся это дата первого повторения)

- b. Если указана дата окончания события, она должна быть больше даты начала
 - c. Если указаны время начала или окончания события, они должны быть в пределах от 00:00 до 23:59
 - d. Если указан период повторения для события, длительность события должна быть меньше этого периода, чтобы события не наезжали друг на друга
- 7. Приложение проверяет, что все связанные с задачей контакты существуют
- 8. Приложение проверяет, что все связанные с задачей контексты существуют
- 9. Если для задачи заданы напоминания, для каждого из них проверяется корректность напоминания
 - a. Проверяется, что задан текст напоминания
 - b. Проверяется, что напоминание не задано в прошлом
 - c. СМС напоминание помещается в 5 СМС (т.е. длина сообщения меньше 67*5 символов кириллицы или 153*5 для латиницы)
- 10. Если для задачи заданы рабочие группы, происходит проверка корректности каждой из создаваемых рабочих групп
 - a. Пользователь может создавать рабочие группы т.е. либо имеет «премиум» учетную запись, либо участвует в не более чем 5 рабочих группах
 - b. Для группы задан хотя бы один участник, помимо создателя группы
- 11. Для задачи рассчитываются уникальный идентификатор LocalTaskId, записывается часовой пояс в зависимости от настроек устройства (отклонение от UTC0 в минутах) и устанавливаются значения даты и времени создания задачи и даты и времени последнего изменения
- 12. Происходит сохранение информации о задаче в базу данных
 - a. Помимо основных связей с контекстами, контактами и файлами создаются также виртуальные связи.
- 13. Приложение обновляет интерфейс в зависимости от того, откуда создается задача. Подробнее см. описание основных разделов органайзера.

Примечание [L6]: Не факт что нужно для мобильной версии, может быть проще так считать.

Альтернативный сценарий 1: при проверке параметров задачи или связанных с ней сущностей (шаги 1-12) произошла ошибка

- 1. Пользователю отображается сообщение об ошибке
- 2. Выполнение сценария завершено

Альтернативный сценарий 2: при сохранении информации о задаче в базу данных произошла ошибка

- 1. Приложение откатывает все изменения, сохраненные в базу данных в рамках сценария
- 2. Пользователю отображается сообщение об ошибке
- 3. Выполнение сценария завершено

3.5 Редактирование задачи

Редактирование задачи аналогично созданию задачи, за некоторыми исключениями, которые указаны в данном разделе:

- 1. Следует проверять, что задача существует и относится к учетной записи, которая на данный момент активна (под который был выполнен вход)

2. Если было изменено количество часов, потраченных на задачу, или же был изменен процент выполнения задачи – происходит запись о событии в дневник (см. сценарий автоматического ведения дневника).
3. Следует проверять, сдвинулась ли задача по времени. Если задача сдвинулась по времени, следует на аналогичный временной промежуток сдвинуть напоминания, заданные относительно начала задачи. Если напоминание задано на точное время и оно уже прошло – напоминание следует сделать неактивным.
4. Если для задачи установлен процент завершения, равным 100, следует перейти к сценарию завершения задачи.
5. Если из рабочей группы удалены все участники, следует закрыть рабочую группу
6. Для задачи должна быть сохранена дата последнего обновления

В остальном действия при обновлении задачи аналогичны созданию, только вместо добавления добавление информации в базу данных происходит ее обновление по локальному идентификатору задачи, события, рабочей группы и т.п.

3.6 Завершение задачи и «пролистывание» события

К завершению задачи приводит установка галочки напротив задачи, повторное нажатие на иконку завершившегося события напротив события или же задание процента выполнения задачи или события равным 100.

К «пролистыванию» события приводит нажатие на значок повторений напротив события. В этом случае все повторы события включая выбранный, должны считаться завершенными и следует отобразить дату следующего наступления события или же показать, что повторы завершены.

Базовый сценарий завершения задачи:

1. Приложение проверяет, что задача существует и относится к учетной записи, которая на данный момент активна (под которой был выполнен вход)
2. Если задача относится к типу «задача», происходит автоматическая запись события в дневник пользователя (см. сценарий автоматического заполнения дневника)
3. Если это пролистывание события – событию устанавливается дата последнего завершения
4. Если это завершение задачи или события – задача или событие помечается завершенной
 - а. Также завершенным должны быть отмечены все подзадачи, которые не были завершены или удалены ранее
 - б. Для завершенных подзадач записи в дневник не делаются
5. Изменения сохраняются в базе данных мобильного приложения.

Альтернативный сценарий 1: при проверке параметров задачи (шаг 1) произошла ошибка

1. Пользователю отображается сообщение об ошибке
2. Выполнение сценария завершено

Альтернативный сценарий 2: при сохранении информации задачи в базу данных произошла ошибка

1. Приложение откатывает все изменения, сохраненные в базу данных в рамках сценария
2. Пользователю отображается сообщение об ошибке

Примечание [L7]: Нужно добавить сценарий

Примечание [L8]: Нужно добавить сценарий

3. Выполнение сценария завершено

3.7 Открыть завершенную ранее задачу

Открытие завершенной ранее задачи происходит из списка завершенных задач. После открытия у задачи меняется флаг Completed и она возвращается в конец списка текущих задач.

Если у задачи имеются также завершенные родительские задачи – они должны открываться по цепочке «вверх» до того момента, пока не дойдет до открытой задачи или же до самого верха иерархии. Таким образом, должна быть исключена ситуация, когда задача остается «подвешенной в воздухе».

При повторном открытии задачи никаких дополнительных проверок не производится.

3.8 Удаление задачи

Удаление задачи аналогично завершению задачи с той разницей, что задачу нельзя потом восстановить.

Аналогично завершению задачи при удалении у задачи удаляются все подзадачи. Причем, если они ранее были завершены – они также удаляются.

При удалении задачи для нее ставится в базе данных флаг Deleted = 1

Если задача не синхронизировалась до удаления с сервером – она не участвует в дальнейшей последующих синхронизациях.

3.9 Создание контекста

Контекст может быть создан из разных элементов интерфейса:

- Быстрое добавление контекста, когда вводится только текст
- Форма для создания и редактирования контекстов (4.18)
- Автоматическое создание контекста при вступлении в компанию

В любом случае в одной условной транзакции к базе данных должен быть создан контекст со всеми зависящими от него сущностями – связями с контактами, задачами и файлами.

Все проверки на корректность полей контекста и связанных с ним сущностей должны быть выполнены до сохранения данных в БД. При не прохождении хотя бы одной проверки сценарий создания контекста прерывается и пользователю отображается сообщение об ошибке.

Базовый сценарий создания контекста:

1. Приложение проверяет, что пользователь авторизован
2. Приложение проверяет, что пользователь может создавать контексты (должен быть премиум-пользователем или же иметь менее 10 активных контекстов) (проверка не происходит, если это автоматическое создание контекста внутри компании).

3. Приложение проверяет корректность заполнения названия контекста – название должно быть заполнено
4. Если указан родительский контекст, приложение проверяет, что такой контекст существует и не удален
5. Приложение проверяет, что связанные с контекстом сущности – задачи, контакты, файлы существуют и не удалены
6. Для контекста рассчитывается уникальный идентификатор LocalContextId и прописываются дата и время создания и последнего изменения текущим моментом
7. Происходит сохранение информации о контексте в базу данных
 - а. Сохраняются также виртуальные связанные сущности – файлы, задачи, контакты
8. Приложение обновляет интерфейс в зависимости от того, откуда создается контекст. Подробнее см. описание основных разделов органайзера.

Альтернативный сценарий 1: при проверке параметров контекста или связанных с ней сущностей (шаги 1-12) произошла ошибка

1. Пользователю отображается сообщение об ошибке
2. Выполнение сценария завершено

Альтернативный сценарий 2: при сохранении информации о контексте в базу данных произошла ошибка

1. Приложение откатывает все изменения, сохраненные в базу данных в рамках сценария
2. Пользователю отображается сообщение об ошибке
3. Выполнение сценария завершено

3.10 Редактирование контекста

Редактирование контекста аналогично созданию контекста, за некоторыми исключениями, которые указаны в данном разделе:

1. Следует проверять, что контекст существует и относится к учетной записи, которая на данный момент активна (под который был выполнен вход)
2. Для контекста должна быть сохранена дата последнего обновления

3.11 Удаление контекста

После удаления контекст исчезает из списка активных контекстов и его невозможно восстановить.

При удалении контекста должны быть удалены все его дочерние контексты.

При удалении контексту устанавливается флаг Deleted = 1.

Если контекст ранее не синхронизировался с сервисом – он не участвует в дальнейшей синхронизации.

Контекст невозможно удалить, если он корпоративный. Для удаления корпоративного контекста необходимо предварительно выйти из компании, к которой относится данный контекст.

3.12 Создание контакта

Контакт может быть создан из разных элементов интерфейса:

- Быстрое добавление контакта, когда вводится только его имя (название)
- Форма для создания и редактирования контактов (4.19)
- Автоматическое создание контактов при вступлении в компанию
- Добавление нового сотрудника в компанию
- Добавление сотрудника компании в контакты

В любом случае в одной условной транзакции к базе данных должен быть создан контакт со всеми зависящими от него сущностями – связями с контекстами, задачами и файлами.

Все проверки на корректность полей контакта и связанных с ним сущностей должны быть выполнены до сохранения данных в БД. При не прохождении хотя бы одной проверки сценарий создания контакта прерывается и пользователю отображается сообщение об ошибке.

Базовый сценарий создания контакта:

1. Приложение проверяет, что пользователь авторизован
2. Приложение проверяет, что пользователь может создавать контакты (должен быть премиум-пользователем или же иметь менее 20 активных контактов) (проверка не происходит, если это автоматическое создание контекста внутри компании).
3. Приложение проверяет корректность заполнения названия контакта – название должно быть заполнено
4. Приложение проверяет, что указанные контактные данные имеют верный формат и не повторяются
5. Приложение проверяет, что связанные с контактом сущности – контексты, контакты, файлы существуют и не удалены
6. Если для контакта указано фото и оно существует на мобильном устройстве – оно сохраняется в специальную папку приложения, а для контакта прописывается путь к этой фотографии в поле LocalPhotoPath
7. Для контакта рассчитывается уникальный идентификатор LocalContactId и указывается время создания и последнего изменения текущим моментом
8. Происходит сохранение информации о контакте в базу данных
 - а. Сохраняются также виртуальные связанные сущности – файлы, задачи, контексты
9. Приложение обновляет интерфейс в зависимости от того, откуда создается контакт. Подробнее см. описание основных разделов органайзера.

Альтернативный сценарий 1: при проверке параметров контакта или связанных с ним сущностей (шаги 1-6) произошла ошибка

1. Пользователю отображается сообщение об ошибке
2. Выполнение сценария завершено

Альтернативный сценарий 2: при сохранении информации о контакте в базу данных произошла ошибка

1. Приложение откатывает все изменения, сохраненные в базу данных в рамках сценария

2. Пользователю отображается сообщение об ошибке
3. Выполнение сценария завершено

3.13 Редактирование контакта

Редактирование контакта аналогично созданию контекста, за некоторыми исключениями, которые указаны в данном разделе:

1. Следует проверять, что контакт существует и относится к учетной записи, которая на данный момент активна (под который был выполнен вход)
2. Для контакта должна быть сохранена дата последнего обновления
3. Контактные данные контакта, для которых не указано значение, должны быть удалены
4. Если для контакта меняется фотография, она должна быть скопирована в специальную папку, в которой работает мобильное приложение, а старая фотография должна быть удалена.

3.14 Удаление контакта

После удаления контакт исчезает из списка активных контактов и его невозможно восстановить.

При удалении контакту устанавливается флаг Deleted = 1.

Если контакт ранее не синхронизировался с сервисом – он не участвует в дальнейшей синхронизации.

Фотография, которая была сохранена для контакта в специальной папке мобильного приложения, должна быть удалена.

3.15 Создание записи дневника

Раздел рассматривает не автоматическое создание записи дневника, т.е. запись, которую вводит пользователь через интерфейс.

В любом случае, в одной условной транзакции к базе данных должна быть создана запись дневника.

Все проверки на корректность полей записи дневника должны быть выполнены до сохранения данных в БД. При не прохождении хотя бы одной проверки сценарий создания записи дневника прерывается и пользователю отображается сообщение об ошибке.

Базовый сценарий создания записи денника:

1. Приложение проверяет, что пользователь авторизован
2. Приложение проверяет корректность заполнения названия записи дневника – название должно быть заполнено
3. Приложение проверяет, что указана дата записи дневника

4. Если для записи задано время начала и окончания, приложение проверяет, что время начала не больше времени окончания и что времена находятся в пределах от 00:00 до 24:00
5. Если для записи дневника указана связанная задача, приложение проверяет, что
6. Для записи дневника рассчитывается уникальный идентификатор LocalRecordId и прописываются дата и время создания и последнего изменения текущим моментом
7. Происходит сохранение информации о записи дневника в базу данных
8. Приложение обновляет интерфейс в зависимости от того, откуда создается запись дневника. Подробнее см. описание основных разделов органайзера.

Альтернативный сценарий 1: при проверке параметров записи дневника или связанных с ней сущностей (шаги 1-5) произошла ошибка

1. Пользователю отображается сообщение об ошибке
2. Выполнение сценария завершено

Альтернативный сценарий 2: при сохранении информации о записи дневника в базу данных произошла ошибка

1. Приложение откатывает все изменения, сохраненные в базу данных в рамках сценария
2. Пользователю отображается сообщение об ошибке
3. Выполнение сценария завершено

3.16 Редактирование записи дневника

Редактирование записи дневника аналогично созданию записи, за исключением того, что при редактировании должна быть обновлена дата и время последнего изменения сущности.

3.17 Удаление записи дневника

После удаления запись дневника исчезает из списка записей и её невозможно восстановить.

При удалении записи устанавливается флаг Deleted = 1.

Если запись ранее не синхронизировалась с сервисом – она удаляется окончательно.

3.18 Создание рабочей группы

Рабочая группа создается только с созданием или редактированием задачи.

В данной версии для одной задачи может существовать только одна активная рабочая группа.

При создании рабочей группы происходит следующее:

1. Если к задаче не относится один из участников рабочей группы, он должен быть добавлен к связанным с группой контактам.

2. Если задача, связанная с рабочей группой, не относится ни к одному корпоративному контексту, задача должна быть связана с теми корпоративными контекстами, к которым относятся участники группы.

3.19 Комментирование задачи (отправка сообщения внутри группы)

Для задач, у которых существуют рабочие группы, доступно комментирование (отправка сообщений).

При наличии подключения к интернету сообщения в обсуждениях задач должны отправляться мгновенно. При отсутствии подключения – они накапливаются и отправляются с указанием фактической даты отправки.

Для сообщения должны быть сделаны следующие проверки:

1. Автор сообщения является одним из участников комментируемой группы
2. Текст сообщения заполнен
3. Группа, к которой относится сообщение, должна существовать

3.20 Добавление файла

Перед добавлением файла должен быть произведен поиск файлов с таким же именем и при наличии совпадения должно отобразиться предложение заменить существующий файл. Если пользователь согласится – происходит операция редактирования файла (3.21).

При добавлении файла в органайзер, файл должен быть скопирован в специальную папку мобильного приложения. Фактическое имя файла при этом **меняется на уникальный идентификатор**. В органайзер же сохраняется начальное имя файла и оно потом используется при отображении списка файлов.

Информация о файле сохраняется вместе со всеми связанными сущностями – задачами, контекстами, контактами в одной транзакции к базе данных.

3.21 Редактирование файла

Редактирование файла аналогично созданию файла, за исключением того, что:

1. При редактировании должна быть обновлена дата и время последнего изменения файла.
2. Если для файла не включен режим ведения истории изменений, старая версия файла удаляется.

3.22 Удаление файла

Файл должен быть удален из файловой системы мобильного устройства, а для записи о файле в базе данных должен быть установлен флаг Deleted=1.

Если запись о файле ранее не синхронизировалась с сервером, она не участвует в дальнейших синхронизациях.

3.23 Синхронизация данных

Синхронизацией данных называется процесс, при котором данные мобильной версии и версии на сервере приходят к одному состоянию.

Актуальность данных определяется последней датой изменения данных, при этом серверная версия при прочих равных имеет больший приоритет.

Синхронизация осуществляется в несколько этапов, которые описаны ниже:

1. Выполняется проверка, что синхронизация возможна для данного пользователя
 - a. Пользователь авторизован и существует на сервере
 - b. Пользователь имеет премиум-версию и не попадает в ограничение на создаваемые сущности, т.е. при синхронизации не создаются задачи более 30, контексты более 10, контакты более 20
 - c. Проверяется, что размер файлов для загрузки вместе с существующими файлами не превосходит 100 Мб и 1 Гб для стандартной и премиум версии соответственно
2. Получается «дельта» с сервера с указанного момента времени (дата последней синхронизации).
3. Формируется общий список изменений, включающий локальные изменения и изменения, сделанные на сервере
4. Составленный на шаге 3 список проверяется на наличие конфликтов
5. По каждому элементу списка выполняется запрос к серверу синхронизации для получения данных или же их обновления. Результаты запросов сохраняются в БД мобильного приложения.

3.24.1 Порядок синхронизации сущностей

Синхронизация мобильного приложения должна осуществляться с сервером условными сессиями. Во время сессий происходит ряд запросов к серверу и при ошибках в них сессия должна быть остановлена и в ряде случаев пользователю должно отобразиться сообщение об ошибке (случаи будут разобраны ниже).

Синхронизация устроена таким образом, что при её прерывании на определенной операции, чаще всего она будет продолжена с того же места, где началась.

Шаги, которые должно выполнить приложение для синхронизации, приведены ниже:

1. Приложение проверяет дату последнего обновления данных профиля. Если данные были обновлены более 5 минут назад, у сервера запрашиваются последние данные профиля и сохраняются в базу данных приложения.
2. Приложение получает список объектов, которые были изменены после последней даты синхронизации. При этом объекты сортируются по дате последнего изменения.

Примечание [L9]: Сопоставить с 3.23.1 как более поздним разделом

Примечание [L10]: Подробно расписать составление списка

3. Если пользователь не является пользователем премиум-версии, выполняются проверки для количества синхронизируемых сущностей:
- Суммарное количество задач пользователя не должно превышать 30
 - Суммарное количество контактов пользователя не должно превышать 20
 - Суммарное количество контекстов пользователя не должно превышать 10
 - Суммарное количество входящих и исходящих задач не должно превышать 5

Если какое-то условие не выполняется, пользователю отображается сообщение о том, что для синхронизации требуется обновить органайзер до премиум-версии.

4. К серверу синхронизации выполняется запрос с указанием даты последней синхронизации данных на мобильном устройстве. В ответ должен быть получен список сущностей, которые были изменены на сервере за указанное время с указанием типа изменения
- Сущность создана – в этом случае приложение должно получить последнюю версию сущности
 - Сущность обновлена – в этом случае приложение должно получить последнюю версию сущности
 - Сущность удалена – в этом случае приложение должно удалить указанную сущность
5. Списки, сформированные на шагах 2 и 4, сопоставляются. Происходит разрешение конфликтов: сущности в мобильном приложении, ожидающие синхронизации, которые были удалены на сервере, удаляются из списка.
6. По каждому элементу списка выполняется соответствующий запрос к серверу на создание, удаление или обновление сущности.

Примечание [L11]: Добавить адрес для получения списка синхронизации и список сущностей

Примечание [L12]: Написать подробнее про разрешение конфликтов

Примечание [L13]: Добавить описания интерфейсов для обмена данными для каждой сущности

3.24.2 Фоновая синхронизация и синхронизация по требованию пользователя

Синхронизация должна работать в двух режимах – фоновая синхронизация (если активна в настройках, происходит каждые несколько минут в зависимости от заданных в настройках значений) и синхронизация по требованию.

Принципиально эти синхронизации не отличаются.

3.26 Процедура обновления приложения

Разрабатываемые приложения должны поддерживать стандартные механизмы обновления (через Android Market для Android и т.п.). При этом все данные пользователя, которые ранее были введены в органайзер, должны сохраняться.

При первом запуске приложения после обновления происходит проверка, были ли обновления в структуре данных приложения. Если обновления были, должны быть выполнены специальные скрипты, которые должны входить в обновление и которые обновят структуру данных до актуального состояния, ничего при этом не удалив.

При синхронизации приложения в случае, если структура данных мобильной версии не соответствует структуре данных на сервере (т.е. приложение не было своевременно обновлено)

должна либо высвечиваться информация о том, что требуется обновление приложения (если обновление критическое) или же разница в структуре должна игнорироваться.

Таким образом, при синхронизации следует передавать версию мобильного приложения, под которым она происходит.

4. Описания интерфейсов

Раздел содержит описания интерфейсов в виде прототипов, принципы их работы и требования по реализации интерфейсов для различных мобильных платформ.

4.1 Общая информация и требования к интерфейсам

4.1.1 Фирменный стиль

Основные элементы управления разрабатываемых приложений должны соответствовать оформлению сайта <http://time-master.ru> кроме тех случаев, когда это противоречит гайдлайнам платформы, под которую ведется разработка (см. 4.1.3).

Примеры элементов фирменного стиля:

- Шрифт Tahoma
- Цвет основного фона – голубой
- Цвет выделений – оранжевый
- Кнопки – в обычном состоянии – голубой фон, серая рамка с закруглением. При нажатии – золотистый фон
- Реакция элемента при выделении – красный цвет текста (см. аналоги со ссылками на сайте).
- Иконки – по возможности использовать оригинальные иконки органайзера или создавать собственные на их основе и по их образцу.

4.1.2 Поддержка телефонов и планшетов

Приложения должны быть разработаны таким образом, чтобы поддерживать телефоны (маленький экран, горизонтальный и вертикальный вид) и планшеты (большой экран, горизонтальный и вертикальный вид).

Приложения должны поддерживать функцию поворота экрана, не зависимо от того, используется ли телефон или планшет.

При этом для планшета в горизонтальном положении слева должно находиться меню для навигации внутри органайзера, а для телефона должна существовать кнопка, вызывающая такое меню (место кнопки определяется гайдлайнами).

4.1.3 Гайдлайны

При дизайне интерфейсов следует руководствоваться официальной документацией по разработке интерфейсов для мобильной платформы:

- http://developer.android.com/guide/practices/ui_guidelines/index.html - для Android
- <http://developer.apple.com/library/ios/#documentation/userexperience/conceptual/mobilehig/Introduction/Introduction.html> - IOS

- [http://msdn.microsoft.com/en-us/library/hh202915\(v=VS.92\).aspx](http://msdn.microsoft.com/en-us/library/hh202915(v=VS.92).aspx) – Windows Phone 8

4.1.4 Скорость работы и время отклика

Разрабатываемое приложение должно максимально быстро откликаться на любые действия пользователя. Если действие предполагает долгое время и не может выполняться в фоновом режиме, должен отображаться прогресс-бар.

4.2 Специфические требования для Android

Поддерживаемые версии операционной системы

Приложение должно работать на платформе Android начиная с версии 2.1 и выше.

Выбор версии основан на распространенности различных версий OS Android

<http://developer.android.com/about/dashboards/index.html>

Минимальное поддерживаемое разрешение экрана - mdpi.

В случае наличия на мобильном устройстве кнопок для функций «меню», «назад» и т.п. – кнопки должны работать корректно.

Для использования возможностей более поздних версий Android, их необходимо безопасно подключать как описано в документации

<http://developer.android.com/google/play/publishing/multiple-apks.html#ApiLevelOptions>

Разрешения, требуемые для установки приложения

- Доступ в интернет
- Список может быть дополнен

Примечание [L14]: Дополнение списка в соответствии с документацией

Действия пользователя, поддерживаемые интерфейсом

(<http://developer.android.com/design/patterns/gestures.html>)

- Нажатие однократное (touch, «тач»)
- Длинное нажатие (long press, «пресс»)
- Листание (swipe, «свайп»)
- Перетягивание (drag, «драг»)
- Двойное нажатие (double touch, «дабл-тач»)
- Растянуть (pinch open)
- Уменьшить (pinch close)

Термины, используемые при описании интерфейсов для Android

- Navigation Bar – область в верхней части экрана - содержит кнопку «назад», иконку приложения и название текущего экрана.
<http://developer.android.com/design/patterns/navigation.html>
- Action Bar – элементы находящиеся на экране, которые содержат важнейшие элементы управления для данного экрана.
Пример: добавить задачу, синхронизировать данные и т.п.
<http://developer.android.com/design/patterns/actionbar.html>
Допускается разделение Action Bar на части для экранов различных размеров и скрытие части кнопок под элемент «Action overflow» (по ссылке приведены таблицы с рекомендациями – сколько кнопок допустимо для различных разрешений экрана).
- Contextual Action Bar (CAB) – частный случай ActionBar, когда стандартные элементы навигации заменяются на допустим «сохранить и отмена». Появляется после определенных действий пользователя на экране. Описан в том же разделе гайдлайнов, что и Action Bar а также <http://developer.android.com/design/patterns/selection.html>.
- Drawers – специальный термин для меню, которое выезжает слева, сдвигая основной экран, и дает возможность навигации по другим разделам приложения.
Может добавлять в Action Bar, если требуется сложная навигация между экранами
- Multi-pane Layouts – экраны, содержимое которых зависит от ориентации дисплея устройства.
<http://developer.android.com/design/patterns/multi-pane-layouts.html>
К примеру для рабочего дня слева может отображаться список дел на день, а справа ToDo для горизонтально ориентированного планшета, а для телефона это будут 2 отдельных экрана с переключением по принципу вкладок.
- Acknowledging – уведомление. Например «Синхронизация завершена».
См. <http://developer.android.com/design/patterns/confirming-acknowledging.html>
http://developer.android.com/design/media/confirm_ack_acknowledge.png
- Settings – настройки приложения. Оформляются по правилам, приведенным в <http://developer.android.com/design/patterns/settings.html>

Переходы между экранами

При переходе «вглубь» должна быть анимация перехода справа-налево. При возврате назад – слева направо.

<http://developer.android.com/design/patterns/swipe-views.html>

Доступность стандартных функций телефона

При наличии в интерфейсах адресов электронной почты, номеров телефонов и т.п. – при нажатии на них должен совершаться звонок, отправляться электронное письмо и т.д. соответственно.

Такие элементы будут отдельно указаны при детальном описании экранов.

Дополнительные требования

При дизайне и разработке интерфейсов недопустимо использовать элементы оформления специфические для других платформ (см. <http://developer.android.com/design/patterns/pure-android.html>). В частности:

- Недопустимо использовать элементы гайдлайнов других мобильных платформ (например http://developer.android.com/design/media/migrating_ios_dialers.png – неверно, т.к. андроид разрешает размещать вкладки только сверху)
- Недопустимо использовать стили элементов других мобильных платформ.
- Недопустимо использовать типичные для других платформ иконки (http://developer.android.com/design/media/migrating_icons.png - для андройда только верхняя строчка)

4.3 Специфические требования для IOS

Версия IOS

Особенности гайдлайнов

Примечание [L15]: Интерфейс пока не отрисовывался, гайдлайны не анализировались

4.4 Специфические требования для Windows Phone 8

Версия Windows Phone

Особенности гайдлайнов

Примечание [L16]: Интерфейс не отрисовывался, гайдлайны не анализировались

4.5 Стартовый экран

Стартовый экран отображается при установке скриптов, обновляющих приложение и при первом запуске приложения, когда выполняются функции, создающие структуру данных.

На стартовом экране должен отображаться логотип и надпись TimeMaster, выполненная в расцветках, аналогичных используемым на сайте. Снизу более мелким шрифтом должна отображаться надпись «подождите, идет загрузка».

https://www.dropbox.com/sh/k3xagjcsbvcqk02/LqbRZZX2H5#f:time_master_1-splash%2B1%2B2%2B3-11.jpg

4.6 Вход в систему

После стартового экрана происходит проверка, был ли выполнен ранее вход в систему.

Если вход не был выполнен, отображается интерфейс, дающий возможность войти в свою учетную запись или же зарегистрироваться на сайте.

При этом переключение между входом и регистрацией должно осуществляться максимально удобно (пример – «вкладки» для Android с возможностью листания влево-вправо или же кнопка снизу – «зарегистрироваться»).

Форма для входа содержит текстовые поля – логин и пароль и кнопку «войти».

Если пользователь нажимает на кнопку «войти» не заполнив одно из полей должно появляться сообщение о том, что оба поля обязательны для заполнения.

Также должно быть предусмотрено сообщение об ошибке на случай, если пользователь ввел неверные логин-пароль или нет соединения с сервером.

Если все поля заполнены корректно, отображается экран загрузки (4.8) и приложения пытается авторизоваться на сервере (3.2).

При успешной авторизации отображается экран загрузки, происходит синхронизация данных и пользователю отображается экран «Рабочий день».

Необходимо также предусмотреть возможность входа в систему с указанием сервера синхронизации. В этом случае должны появиться дополнительные поля: адрес сервера и использовать ли HTTPS. Заполнение адреса сервера является обязательным.

Схематически интерфейс для входа изображен на рисунке 4.6 в папке «Прототипы»

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_1-splash%2B1%2B2%2B3-02.jpg

Примечание [L17]: Тут и далее пока ссылки до окончательной компоновки ТЗ

4.7 Регистрация

При регистрации от пользователя требуется ввести:

- Адрес электронной почты
- Пароль
- Подтверждение пароля

После этого пользователь нажимает на кнопку «Зарегистрироваться» и происходит запрос к серверу (отображается экран загрузки (см. 4.8) с сообщением, что идет создание учетной записи).

Перед отправкой запроса на регистрацию приложение должно выполнить необходимые проверки: все поля заполнены, электронная почта указана в правильном формате, пароль и подтверждение пароля совпадают. Если одно из этих условий не выполнено или в ходе регистрации или же в ходе выполнения запроса произошла ошибка – должно отобразиться сообщение об ошибке.

При успешной регистрации происходит синхронизация данных с сервером (фактически, будет получена только информация о профиле, т.к. учетная запись только что создана) и пользователю отображается экран «Рабочий день».

Схематически интерфейс для регистрации изображен на рисунке 4.7 в папке «прототипы»

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_1-splash%2B1%2B2%2B3-03.jpg

4.8 Экран загрузки

Отображается при входе в систему, выполнении различных длительных действиях, синхронизации и т.п.

При отображении экрана остальная часть становится серой и неактивной, а по середине отображается значок загрузки (анимированный) с сообщением о том, что идет загрузка.

В некоторых случаях (например, загрузка файла) отображается также прогресс бар с указанием, на сколько процентов выполнена операция.

Схематически интерфейс для экрана загрузки изображен на рисунке 4.8 в папке «прототипы».

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_1-splash%2B1%2B2%2B3-04.jpg

4.9 Настройки

Раздел настроек должен быть выполнен стандартно в зависимости от платформы и включать в себя следующие разделы:

- Частота синхронизации – как часто приложение будет пытаться синхронизироваться с сервером.
- Частота запросов последних сообщений от коллег – как часто приложение будет запрашивать у сервера информацию о количестве новых задач, сообщений и уведомлений.
- Использовать HTTPS при обмене данными с сервером
- Напоминать о просроченных задачах
- Напоминать о необходимости заходить в органайзер
- Текущий часовой пояс
- Начало и окончание рабочего дня
- Использовать ли HTTPS для передачи данных

Если позволяют гайдлайны, настройки должны быть разделены по группам - синхронизация, настройки приложения, часовой пояс и т.п.

4.10 Меню

Меню позволяет переходить между основными разделами органайзера и переходить к последним входящим сообщениям, уведомлениям и входящим задачам для корпоративных пользователей.

Меню должно содержать следующие пункты:

- Рабочий день
- Календарь
- Задачи
- Контексты

- Контакты
- Файлы
- Дневник
- Новые сообщение (только для корпоративных профилей)

Каждый из перечисленных выше пунктов меню – считается основным разделом органайзера и кнопка «назад» на телефоне для такого раздела означает выход из приложения.

Соответственно слева-сверху для этих разделов должна отображаться кнопка, показывающая меню.

Изначально меню не отображается, а появляется при нажатии на кнопку «меню».

При этом допустимо сдвигать основной экран правее (например, android) или показывать меню поверх остальных элементов (какие-то другие платформы).

Схематично меню изображено на рисунке 4.10.1 и 4.10.2 в папке «прототипы».

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_1-splash%2B1%2B2%2B3-06.jpg

4.11 Переходы между основными разделами органайзера

В этом разделе приведены диаграммы переходов между экранами для основных разделов органайзера.

Переходы для раздела «Рабочий день» приведены на рисунке 4.11.1

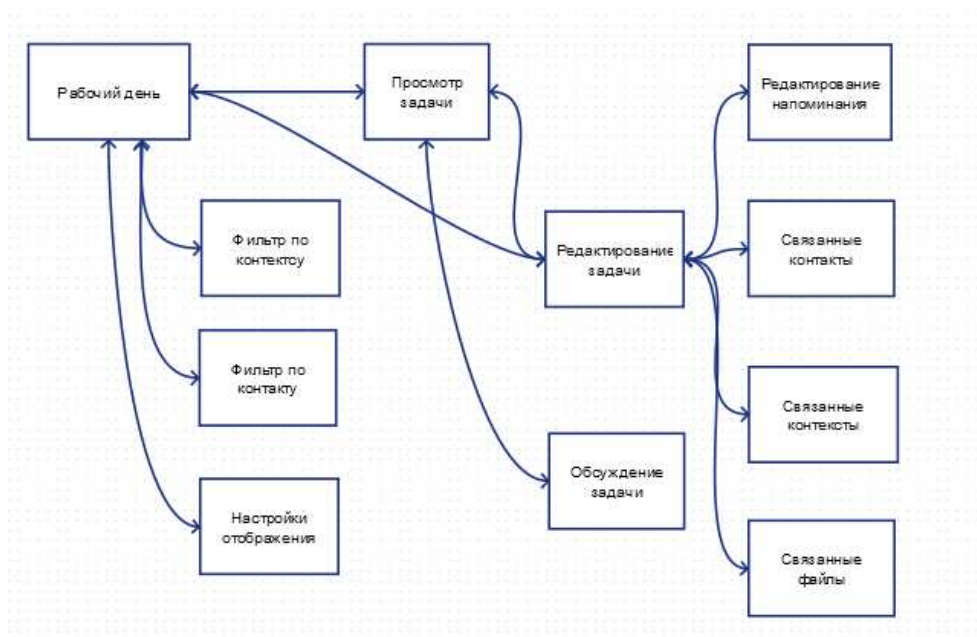


Рисунок 4.11.1 – Переходы для раздела «Рабочий день»

Переходы для раздела «Календарь» приведены на рисунке 4.11.2

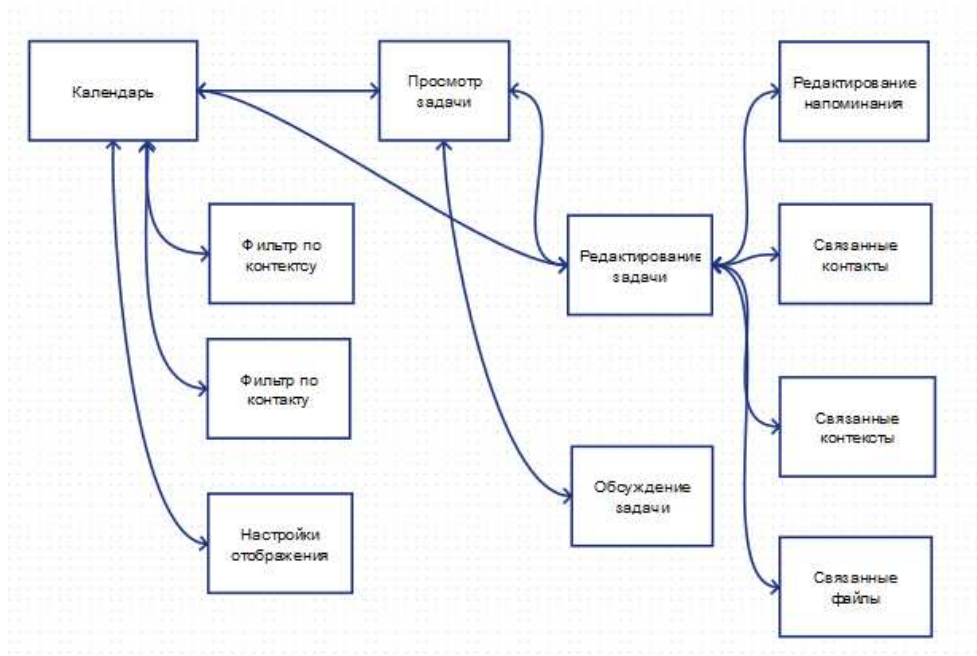


Рисунок 4.11.2 – Переходы для раздела «Календарь»

Переходы для раздела «Задачи» приведены на рисунке 4.11.3

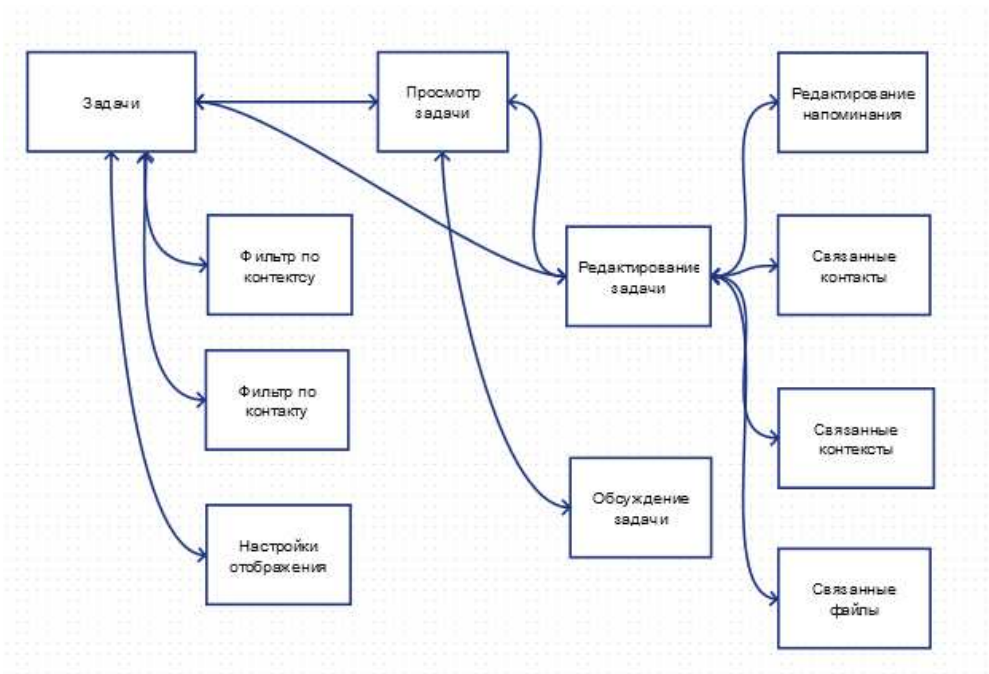


Рисунок 4.11.3 – Переходы для раздела «Задачи»

Переходы для раздела «Контексты» приведены на рисунке 4.11.4

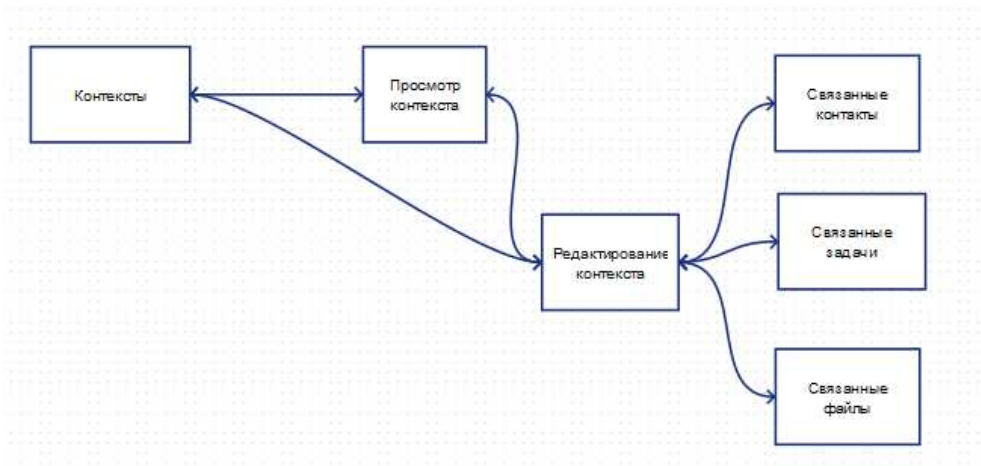


Рисунок 4.11.4 – Переходы для раздела «Контексты»

Переходы для раздела «Контакты» приведены на рисунке 4.11.5

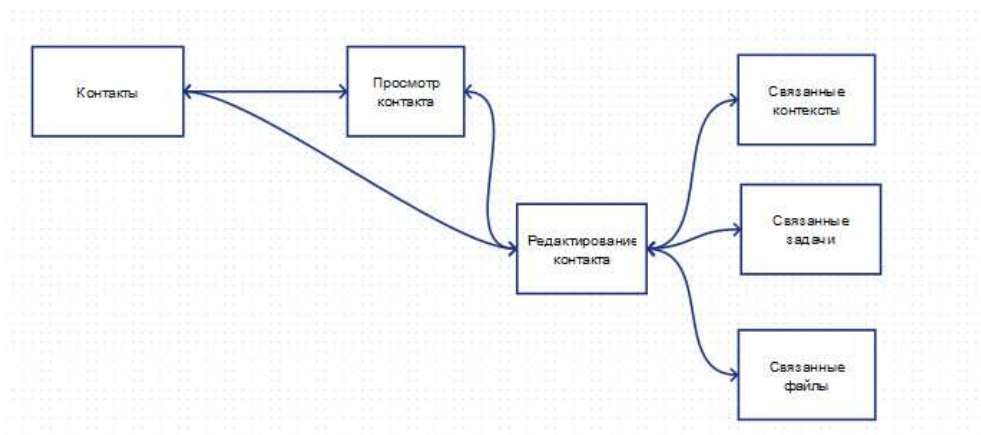


Рисунок 4.11.5 – Переходы для раздела «Контакты»

Переходы для раздела «Файлы» приведены на рисунке 4.11.6

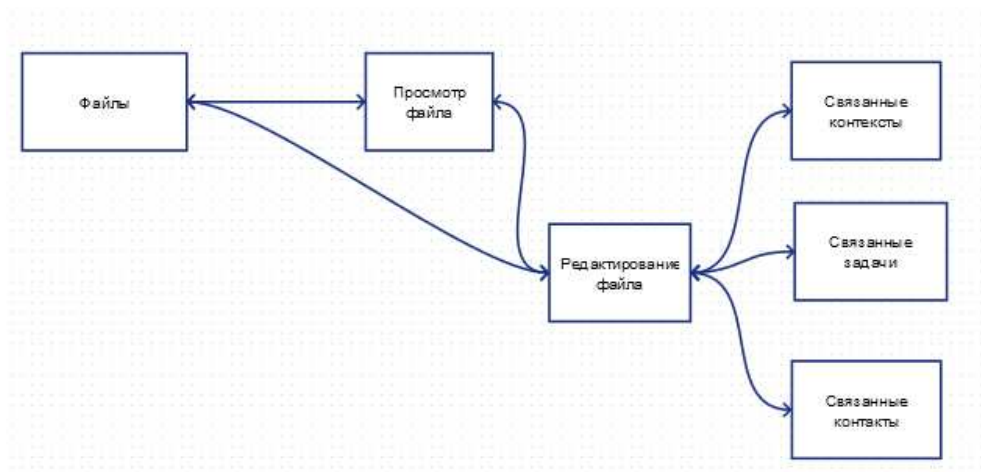


Рисунок 4.11.6 – Переходы для раздела «Файлы»

Переходы для раздела «Дневник» приведены на рисунке 4.11.7

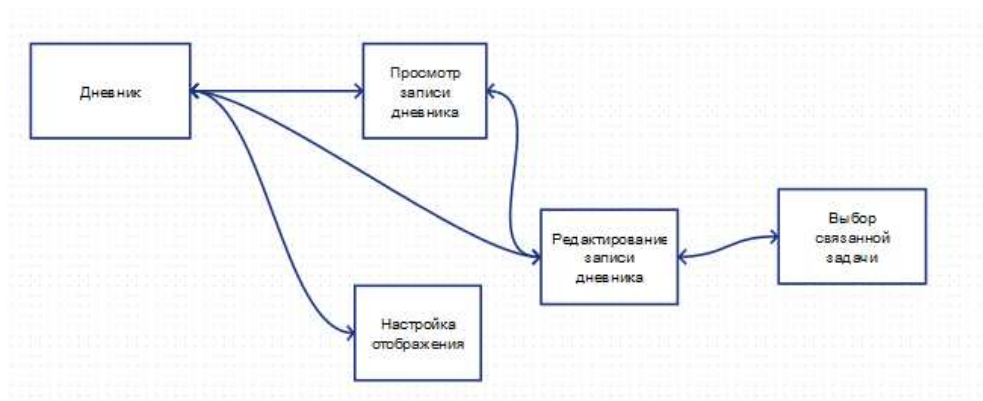


Рисунок 4.11.7 – Переходы для раздела «Дневник»

4.12 Рабочий день

При создании раздела следует ориентироваться на функционал аналогичной страницы веб-интерфейса органайзера <http://time-master.ru/organizer/work/> (Рисунок 4.10).

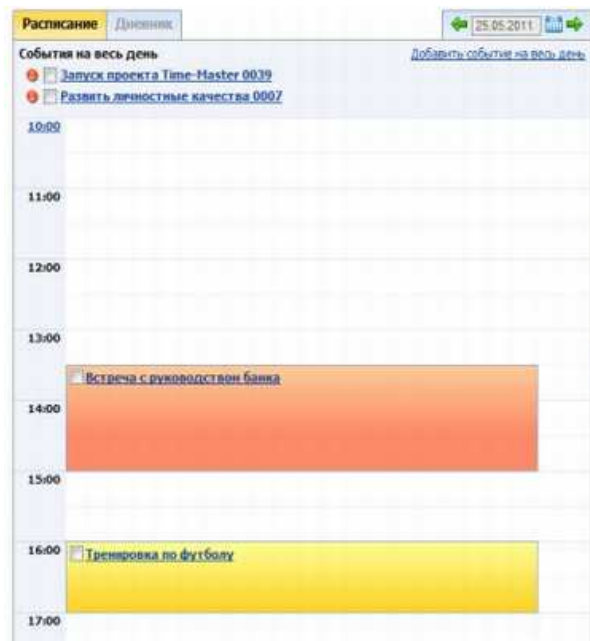


Рисунок 4.10 – Рабочий день на сайте Time-Master.ru

Должны быть предусмотрены следующие функции:

- Переход между датам (вперед, назад, выбор даты из календаря)

- Отображение списка задач и событий на целый день сверху
- Визуальное отображение расписания
- Кнопка для добавления задачи
- Кнопка для фильтрации данных
- Отображение дедлайнов (см. http://time-master.ru/img/c/help/h_wd17.jpg)
- Должна учитываться ситуация, когда несколько задач и событий заданы на одно время (см. http://time-master.ru/img/c/help/h_wd15.jpg)
- Должна учитываться ситуация, когда текст задачи может не влезать в прямоугольник или надпись у дедлайна (т.е. при нажатии на задачу должно появляться облачко с текстом).

Для версии для планшета рабочий день может быть объединен с разделом ToDo при горизонтальном просмотре. При этом задачи из ToDo должны перетаскиваться на календарь.

Прототип «рабочего дня» приведен на рисунке 4.12 в папке «прототипы».

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_1-splash%2B1%2B2%2B3-05.jpg

4.13 Неделя

При создании раздела следует ориентироваться на функционал аналогичной страницы веб-интерфейса органайзера <http://time-master.ru/organizer/calendar/week.aspx> и <http://time-master.ru/mobile/calendar/week.aspx> (Рисунок 4.11).



Рисунок 4.11 – Неделя на сайте Time-Master.ru

Должны быть предусмотрены следующие функции:

- Навигация между неделями
- Выбор недели в календаре
- Выбор количества дней, отображаемых в неделю
- Отображение задач на всю неделю сверху (см. http://time-master.ru/img/c/help/h2_k6.jpg - задача «прохождение тренинга по продажам»)
- По каждому дню отображение задач на весь день

- Отображение задач в расписании в течение дня
- Кнопка для добавления задачи
- Кнопка для фильтрации данных
- Отображение дедлайнов
- Должна учитываться ситуация, когда несколько задач заданы на одно и тоже время
- Должна учитываться ситуация, когда текст задачи может не влезать в прямоугольник или надпись у дедлайна

Для устройств с маленьким экраном можно рассмотреть аналог <http://time-master.ru/mobile/calendar/week.aspx>

Прототип календаря на неделю приведен на рисунке 4.13 в папке «прототипы».

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_1-splash%2B1%2B2%2B3-08.jpg

4.14 Месяц

Для месячного представления должны поддерживаться следующие функции (см. <http://time-master.ru/mobile/calendar/>):

- Выбор месяца для отображения
- Список задач и событий на весь месяц
- Отображение задач на целый день в календаре визуально и количества задач по ходу дня там же в виде цифр
- Кнопка для добавления задачи
- Кнопка для фильтрации по контекстам и контактам
- Должна учитываться ситуация, когда задачи или события попадают на одно и тоже время

Прототип календаря на месяц приведен на рисунке 4.14 в папке «прототипы».

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_1-splash%2B1%2B2%2B3-09.jpg

4.15 ToDo

Отображает список ближайших действий – задач, у которых нет подзадач. События в ToDo не отображаются. В качестве примера можно ориентироваться на <http://time-master.ru/mobile/work/todo.aspx> и <http://time-master.ru/organizer/work/> правую часть страницы.

- Список ближайших задач
- Сортировка задач по приоритету, дате или порядковому номеру
- Кнопка для фильтрации по контекстам и контактам
- Кнопка для добавления задачи
- Перебрасывание задач с ToDo на расписание

- При построении должно учитываться свойство задач «Выполнять подзадачи последовательно». В этом случае при наличии у родительской задачи с таким свойством к примеру 3х подзадач, отображается только первая из них
- Список должен пополняться по мере завершения подзадач родительскими задачами

Прототип раздела «ToDo» приведен на рисунке 4.15 в папке «прототипы».

https://www.dropbox.com/sh/k3xagicsbvcgk02/LqbRZZX2H5#f:time_master_1-splash%2B1%2B2%2B3-10.jpg

4.16 Дерево задач

Дерево задач отображает древовидное представление задач. Пользователю отображается несколько уровней дерева: каждый последующий уровень сдвинут относительно предыдущего до определенного предела (приблизительно 4 уровня для экрана телефона, 6 для экрана планшета). После этого следующий уровень просто выделяется новым цветом.

В качестве примера можно ориентироваться на страницы <http://time-master.ru/mobile/tasks/> и <http://time-master.ru/mobile/tasks/>

Раздел позволяет:

- Просматривать дерево задач с определенной максимальной глубиной
- Выделять задачу
- Добавлять задачу на верхний уровень дерева или как подзадачу к выделенной задаче
- Переходить к просмотру и редактированию задач
- Фильтровать дерево задач по контакту или контексту

Прототип раздела «Дерево задач» приведен на рисунке 4.16 в папке «прототипы».

https://www.dropbox.com/sh/k3xagicsbvcgk02/LqbRZZX2H5#f:time_master_2-4-01.jpg

4.17 Контакты

Список контактов должен отображать существующие в органайзере контакты с фотографиями.

В качестве примера можно ориентироваться на страницы <http://time-master.ru/organizer/contacts/> и <http://time-master.ru/mobile/contacts/>

Раздел позволяет:

- Просматривать список контактов
- Сортировать контакты по порядку или по алфавиту
- Добавлять и редактировать контакты
- Просматривать информацию о контактах
- Фильтровать контакты по контекстам

Прототип раздела «Контакты» представлен на рисунке 4.17 в папке «прототипы».

Примечание [L18]: Обсуждаемо.

Варианты:

- 1) Уменьшать расстояние между уровнями в зависимости от того, сколько их
- 2) Давать возможность скрола вправо-влево
- 3) С какого-то уровня перестраивать дерево заново, начиная с этого уровня.

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_5-10-04.jpg

4.18 Контексты

Раздел позволяет смотреть список контекстов в виде дерева.

Контексты отображаются в несколько уровней аналогично разделу «Задачи» - т.е. в зависимости от устройства после какого-то уровня дерево перестает сдвигаться влево, а начинает выделяться цветом.

В качестве примера можно ориентироваться на страницы <http://time-master.ru/organizer/contexts/> и <http://time-master.ru/mobile/contexts/>

Раздел позволяет:

- Просматривать список контекстов
- Добавлять и редактировать контексты
- Просматривать информацию о контекстах

Прототип раздела «Контексты» представлен на рисунке 4.18 в папке «прототипы».

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_5-10-01.jpg

4.19 Файлы

Раздел позволяет смотреть список файлов, которые пользователь загрузил в органайзер.

Файлы отображаются списком и отсортированы по алфавиту. Отдельно отображаются личные файлы и файлы, к которым пользователю предоставили доступ.

Раздел позволяет:

- Просматривать список файлов
- Просматривать информацию о файлах
- Добавлять и редактировать файлы
- Фильтровать файлы по контекстам и контактам

Прототип раздела «Файлы» представлен на рисунке 4.19 в папке «прототипы».

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_5-10-09.jpg

4.20 Дневник

Дневник позволяет просматривать записи дневника пользователя по дням.

В качестве примера можно ориентироваться на разделы <http://time-master.ru/organizer/diary/> и <http://time-master.ru/mobile/diary/>

Записи отображаются в виде списка, причем элементы списка могут быть нескольких цветов:

- зеленый, если задача была завершена
- красный, если время было потрачено впустую
- желтый, если задача была завершена, но пользователь считает время потраченным впустую
- белый, если пользователь не делал специальных пометок

Прототип раздела «Дневник» представлен на рисунке 4.20 в папке «прототипы».

4.21 Список завершенных задач

Список завершённых задач – второй таб в разделе «Задачи».

При создании раздела следует ориентироваться на аналогичный функционал сайта <http://time-master.ru/organizer/tasks/completed.aspx> и его описание <http://time-master.ru/organizer/help/tasks/#p6>

Внешний вид списка аналогичен дереву задач, но чекбокс напротив каждой из задач изначально установлен. При нажатии на него чекбокс переходит в состояние «не установлен», причем, если задача имеет родительские задачи, чекбокс сбрасывается для всех задач вверх по иерархии.

Процесс открытия ранее завершенной задачи описан в 3.7.

Для завершенных задач можно посмотреть свойства аналогично просмотру свойств открытой задачи, но изменить свойства для завершенных задач нельзя – её требуется предварительно открыть.

Для совместных задач из раздела завершенных задач возможно продолжение обсуждения задачи.

4.22 Просмотр задачи

Просмотр задачи состоит из нескольких табов, которые переключаются свайпом влево или вправо.

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_2-4-03.jpg

Для открытых задач отображается кнопка «Завершить».

Для завершенных – «Открыть повторно».

4.23 Редактирование задачи

Редактирование состоит из нескольких табов.

При нажатии «Сохранить» сохраняются все изменения во всех табах.

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_2-4-05.jpg

4.24 Просмотр контекста

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_5-10-02.jpg

4.25 Редактирование контекста

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_5-10-03.jpg

4.26 Просмотр контакта

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_5-10-05.jpg

4.27 Редактирование контакта

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_5-10-06.jpg

4.28 Просмотр напоминания

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_2-4-10.jpg

4.29 Редактирование напоминания

4.30 Просмотр файла

4.31 Редактирование файла

4.32 Просмотр записи дневника

4.33 Редактирование записи дневника

4.34 Обсуждение задачи

https://www.dropbox.com/sh/k3xagicsbvcqk02/LqbRZZX2H5#f:time_master_2-4-12.jpg

5. Настойки приложения

В раздел включены настройки, которые должны быть доступны для пользователя в приложении с пояснением, как они работают и что от них зависит.

- Частота синхронизации – как часто приложение будет пытаться синхронизироваться с сервером.
- Частота запросов последних сообщений от коллег – как часто приложение будет запрашивать у сервера информацию о количестве новых задач, сообщений и уведомлений.
- Использовать HTTPS при обмене данными с сервером
- Напоминать о просроченных задачах
- Напоминать о необходимости заходить в органайзер

-

6. Структуры входящих и исходящих сообщений

В разделе будут описаны или даны ссылки на описания всех интерфейсов для взаимодействия с сервером.

7. Этапы разработки

7.1 Первый этап

Разработка интерфейсов на основе заглушек.

- Должны быть реализованы основные классы бизнес-логики на основе библиотеки LifeManager.Objects
- Должны быть реализованы все экраны приложения с тестовыми данными
- Должны быть реализованы переходы между экранами
- Должны быть реализованы элементы интерфейса экранов и их анимация при необходимости
- Текст из ресурсов

7.2 Второй этап

Разработка бизнес-логики

- База данных приложения и связанные с ней операции типа обновления
- Бизнес-логика
- Привязка интерфейса к бизнес-логике

7.3 Третий этап

Синхронизация приложения с ТМ

- Разработка процедуры входа
- Настройки пользователя
- Синхронизация объектов
- Загрузка объектов, которые подтягиваются онлайн

7.4 Четвертый этап

Специфические для мобильного устройства функции

- Уведомления
- Взаимодействие с телефонной книжкой
- Голосовой ввод для задач, распознавание текста с фото и т.п.

-

8. Требования к разработке, документированию и оформлению кода

8.1 Требования к оформлению кода

8.2 Правила разработки