

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ОТЧЕТ

Лабораторная работа №2
по курсу: «Конструирование ПО»
на тему: «**Структурный подход к программированию.**
Стадия «Эскизный проект»»

Выполнил:
ст. гр. ИПО-14а
Гончарук С.И.

Цель работы: научиться создавать формальные модели и на их основе определять спецификации разрабатываемого программного обеспечения.

Содержание

| | |
|--|----|
| 1. ОБЩИЕ ПОЛОЖЕНИЯ..... | 3 |
| 1.1. Наименование системы | 3 |
| 1.1.2. Краткое наименование системы | 3 |
| 1.2. Основания для проведения работ..... | 3 |
| 1.3. Наименование организаций – Заказчика и Разработчика | 3 |
| 1.3.1. Заказчик..... | 3 |
| 1.3.2. Разработчик..... | 3 |
| 1.4. Цели, назначение и область использования системы | 3 |
| 1.5. Нормативные ссылки..... | 4 |
| 1.6. Очередность создания системы..... | 4 |
| 2. ОСНОВНЫЕ ТЕХНИЧЕСКИЕ РЕШЕНИЯ..... | 6 |
| 2.1. Решения по структуре системы, подсистем, средствам и способам связи для информационного обмена между компонентами системы | 6 |
| 2.1.1. Логическая и компонентная архитектура системы..... | 6 |
| 2.1.2. Функциональная структура системы | 6 |
| 2.2. Решения по режимам функционирования, диагностированию работы системы..... | 7 |
| 2.3. Решения по персоналу и режимам его работы | 9 |
| 2.4. Сведения об обеспечении заданных в техническом задании потребительских характеристик системы, определяющих ее качество | 9 |
| 2.5. Состав функций, комплексов задач, реализуемых системой | 12 |
| 2.5.1. Подсистема сбора, обработки и загрузки данных..... | 12 |
| 2.5.1.1. Функция «Создания новой записи в записной книге»..... | 12 |
| 2.6. Решения по составу информации, объему, способам ее организации, видам машинных носителей, входным и выходным документам и сообщениям, последовательности обработки информации и другим компонентам..... | 12 |
| 2.6.1. Описание информационной базы | 12 |
| 2.6.2. Решения по пользовательскому интерфейсу | 13 |
| 2.7. Методы и средства разработки..... | 19 |
| 3. СЛОВАРЬ ТЕРМИНОВ..... | 20 |

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Наименование системы

1.1.1. Полное наименование системы

Полное наименование - «Система управления личным временем».

1.1.2. Краткое наименование системы

Краткое наименование - «Расширенный органайзер».

1.2. Основания для проведения работ

Основанием для разработки является документ:

Контракт №27/33381 на выполнение подготовительных работ по созданию системы «Органайзер».

Стороны, утвердившие документ – ГБУЗ ДонНТУ от 08.02.2017 и ОАО “Put-in”.

1.3. Наименование организаций – Заказчика и Разработчика

1.3.1. Заказчик

Заказчик: ОАО “Put-in”

Адрес фактический: г. Москва, Красная площадь – 1.

Телефон / Факс: +7 (495) 2222222

1.3.2. Разработчик

Разработчик: ЗАО “学生”

Адрес фактический: г. Владивосток, Симачева – 32.

Телефон / Факс: +7 (495) 3333333

1.4. Цели, назначение и область использования системы

Функционал программы должен позволять создавать, редактировать и удалять заметки, расписание, отметки на карте, приглашать друзей на мероприятия.

Эксплуатационное назначение – управление личным временем и повседневными делами.

Потребность в разработке нового решения при существовании готовых продуктов, которые частично реализуют требования, описанные настоящим техническим заданием, обусловлено тем, что обычно предполагаемый к реализации функционал содержится одновременно в нескольких программных продуктах. В большинстве вариантов интерфейс подобного ПО не продуман и сделан «на скорую руку».

Для решения проблем, описанных выше требуется создать общую систему, которая содержала бы все эффективные инструменты для управления своим временем и делами в приятной графической оболочке.

1.5. Нормативные ссылки

При эскизном проектировании использовались следующие нормативно-технические документы:

1. Техническое задание на создание системы управления личным временем «Расширенный органайзер».
2. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы.

1.6. Очередность создания системы

Ниже представлена предполагаемая очередность создания системы:

- Производится разработка модели хранилища данных.
- Согласовываются форматы и структуры обмена данными с системами-источниками.
- Проектируются процессы сбора данных в область временного хранения данных.
- Проектируются процессы загрузки данных в область постоянного хранения данных.
- Проектируются типовые отчеты.
- Разрабатывается схема организации доступа пользователей.

- Разрабатывается схема взаимодействия пользователя с пользовательским интерфейсом.
- Производится настройка активного сетевого оборудования.
- Производится настройка аппаратно-технической части: установка и настройка серверов, подключение к сетевому активному оборудованию, настройка сетевых параметров и т.п.
- Разрабатывается план установки серверного программного обеспечения.
- Производится установка серверного программного обеспечения.
- Реализация графического интерфейса.
- Реализация части ПО, отвечающей за хранение пользовательских данных (База данных).
- Реализация записной книги.
- Реализация карты и событий на карте.
- Реализация системы пользователей.
- Реализация личных сообщений, добавления в друзья, приглашений на события.
- Реализация подсистемы пользовательских настроек.
- Тестирование графического интерфейса.
- Тестирование базы данных.
- Тестирования механизмов взаимодействия пользователей в пределах от отправки сообщений, до добавления в друзья и приглашений на события.
- Тестирование подсистемы пользовательских настроек.
- Оптимизация запросов к базе данных.

2. ОСНОВНЫЕ ТЕХНИЧЕСКИЕ РЕШЕНИЯ

2.1. Решения по структуре системы, подсистем, средствам и способам связи для информационного обмена между компонентами системы

2.1.1. Логическая и компонентная архитектура системы

Перечень используемых для создания системы «Органайзер» программных средств приведен ниже:

- СУБД (MySQL, 6.0.6)

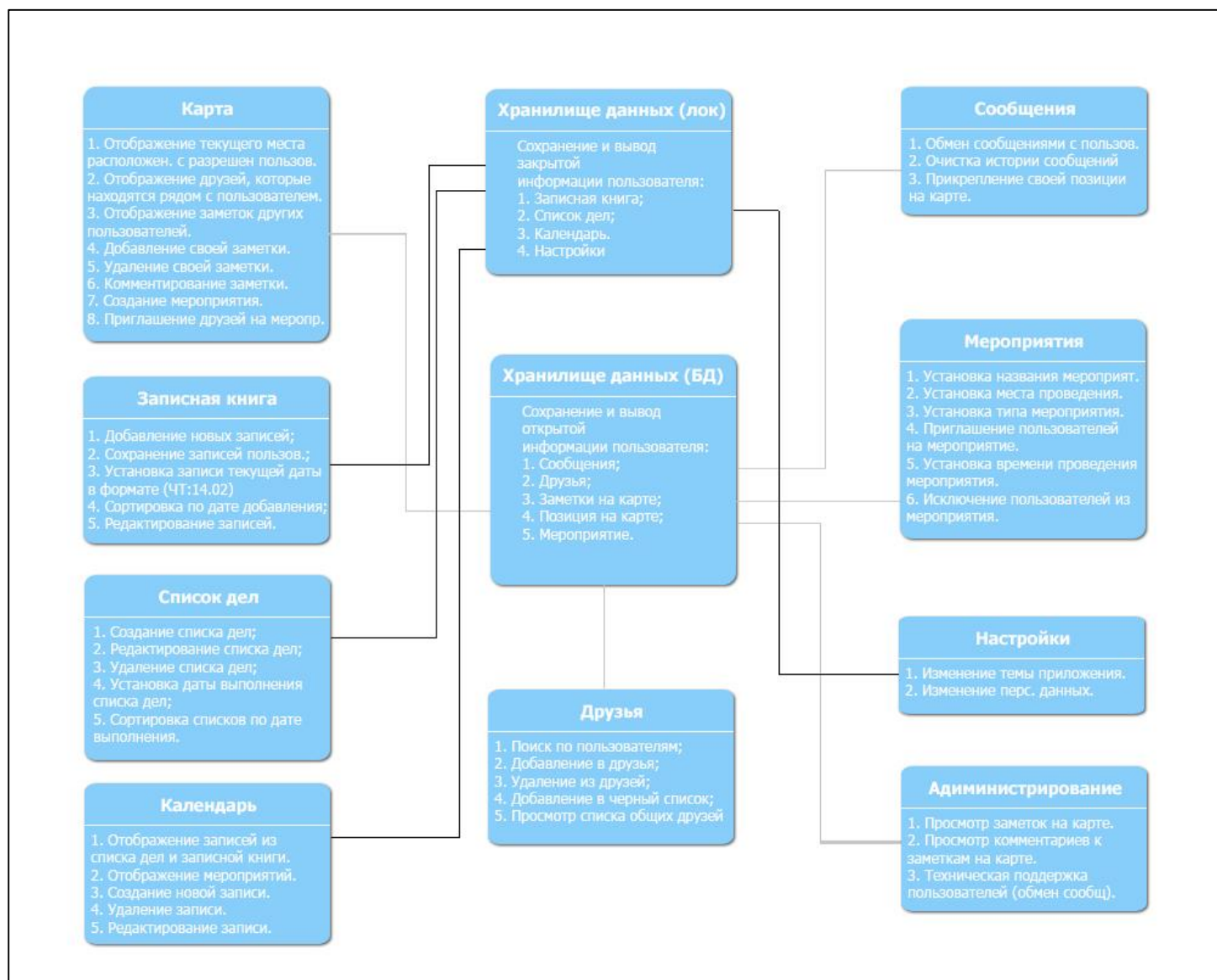
2.1.2. Функциональная структура системы

Функциональная структура системы состоит из таких подсистем:

- Хранилище данных (БД) – отвечает за хранение открытых данных пользователя;
- Хранилище данных (Локальное) – отвечает за хранения закрытых данных пользователя (Разграничение существует для того, чтобы обезопасить конфиденциальные данные пользователей);
- Карта – отвечает за взаимодействие пользователей с интерактивной картой;
- Друзья – отвечает за возможность пользователей кооперироваться друг с другом;
- Мероприятия – отвечает за возможность посещения друзьями одного и того же мероприятия;
- Список дел – отвечает за возможность пользователю составлять свои списки дел;
- Календарь – отвечает за отображения на календаре всех списков дел, записей из записной книги и всех мероприятий, в которых отмечен пользователь;
- Сообщения – отвечает за возможность обмена сообщениями между пользователями;
- Записная книга – отвечает за возможность ведения личных записей пользователя;

- Настройки – отвечает за возможность изменения темы приложения и пользовательских данных;
- Администрирование – отвечает за возможность модерирования пользовательских заметок на карте администраторами приложения.

Функциональная структура системы изображена в виде подсистем и их взаимодействия на рисунке 1.



2.2. Решения по режимам функционирования, диагностированию работы системы

Предлагается следующая реализация решений по режимам функционирования системы:

- Основной режим, в котором все подсистемы выполняют свои основные функции.
- Профилактический режим, в котором одна или все подсистемы не выполняют своих функций. В данный режим работы система переходит в следующих случаях: возникновение необходимости модернизации аппаратно-программного комплекса; возникновение необходимости проведения технического обслуживания; выход из строя аппаратно-программного комплекса, вызванный выходом из строя элементов аппаратной или программной базы; выход из строя сети передачи данных и другие аварийные ситуации.

В основном режиме функционирования система обеспечивает:

- работу пользователей в режиме – 24 часа в день, 7 дней в неделю (24x7);
- выполнение своих функций – сбор, обработка и загрузка данных; хранение данных, предоставление отчетности по показателям.

В профилактическом режиме система обеспечивает возможность проведения следующих работ:

- техническое обслуживание;
- модернизацию аппаратно-программного комплекса;
- устранение аварийных ситуаций.

Общее время проведения профилактических работ не должно превышать 5% от общего времени работы системы в основном режиме (10 часов в месяц).

Принимается предварительное решение о том, что для обеспечения высокой надежности функционирования как системы в целом, так и ее отдельных компонентов необходимо проводить регулярное диагностирование состояния компонентов.

В таблице ниже представлены средства диагностики по подсистемам.

| Подсистема | Средства диагностирования |
|---------------------------------|---|
| Подсистема хранения данных (БД) | DB Manager – диагностика и настройка и конфигурация одной или более БД. |

Подсистема хранения данных (БД):

- администратор подсистемы должен каждый день контролировать работоспособность серверной части (БД) т.к. данная подсистема является критичной для работоспособности системы в целом;
- администратор подсистемы должен каждый день проводить анализ базы данных на наличие ошибок и предупреждений, возникающих при ее работе.

2.3. Решения по персоналу и режимам его работы

На основании пункта «Требования к численности персонала» технического задания приведены соответствующие решения по численности, квалификации и функциям персонала создаваемой системы, режимам работы персонала.

Сформирована таблица с возможными вариантами привязки ролей пользователей и администраторов системы к организационной структуре Заказчика.

| Роль | Подразделение |
|---|---------------------------------------|
| Конечный пользователь | - |
| Администратор подсистемы хранения данных. | Департамент информационных технологий |

2.4. Сведения об обеспечении заданных в техническом задании потребительских характеристик системы, определяющих ее качество

Далее приводится таблица трассировки требований, заданных в техническом задании, и описанных проектных решений (методы их реализации).

| Требование | Метод реализации |
|---|---|
| Диагностирование системы | Реализуется путем определения перечня работ по диагностированию подсистем. |
| Сохранение работоспособности системы в различных вероятных условиях | Реализуется путем разработки процедур резервного копирования, подготовки персонала, использования современных методов разработки и проверенных на |

| | |
|--|--|
| | <p>практике стандартных программных средств.</p> <p>На объекте автоматизации обязательно ведение журналов инцидентов в электронной форме.</p> |
| <p>Программная система должна предоставлять пользователю графический пользовательский интерфейс для управления своими заметками и делами.</p> | <p>Реализуется путем создания одностраничного приложения (Angular.js). Одним из больших преимуществ одностраничных приложений является загрузка всего приложения 1 раз, а затем после действий пользователя происходит обновление просматриваемой зоны просмотра. Навигация выполняется при помощи бургер – меню. Так как такой метод экономит пространство, что важно для мобильных приложений, где пространство просмотра достаточно ограничено.</p> |
| <p>Система должна предоставлять пользователю возможность создавать, удалять, редактировать заметки, список дел, расписание, календарные планы.</p> | <p>Реализуется путем создания отдельных модулей под эти задачи, которые позволят работать с текстовой и графической информацией. Для календаря не будет использоваться стандартный, а будет создан новый объект.</p> |
| <p>Программная система должна предоставлять возможность пользователю привязывать запись к определённой геолокации. С дальнейшим</p> | <p>Реализуется путем API – интерфейса Google Maps. Будет создан слой поверх стандартной карты, где будут располагаться метки. Так же будет</p> |

| | |
|---|--|
| нанесением этой позиции на общую карту. | использован стандартный метод API для получения текущего расположения устройства. API Google Maps достаточно эффективен для таких целей, так как поддерживает много методов триангуляции сигнала. |
| Система должна разделять данные пользователя на открытые и закрытые во избежание взлома определённых данных пользователя. | Реализуется путем разделения и расположения двух типов данных на двух разных хранилищах: открытые – сервер БД, закрытые – на устройстве пользователя (Local storage). |
| Система должна предоставлять возможность создавать пользователю свой профиль и задавать ключевые данные. | Реализуется путем добавления в базу данных таблиц, отвечающих за профиль. Поля «Имя», «Фамилия», «Текущая геолокация» (Последняя геолокация) и т.д. |
| Система должна предоставлять возможность добавлять друзей, удалять друзей, помещать пользователей в черный список. | Реализуется путем добавления в базу данных таблиц, отвечающих за друзей. С созданием поля – «Статус» 0 – заявка не отправлена 1 – заявка отправлена 2 – заявка принята 3 – черный список |
| Система должна предоставлять возможность создания мероприятий с привязкой к месту, возможность приглашать на мероприятия друзей, редактировать мероприятия: название, время, тип, место, участники. | Реализуется путем создания таблиц в базе данных отвечающих за мероприятия. Предполагаются поля: название, время, тип, место, участники. |

2.5. Состав функций, комплексов задач, реализуемых системой

2.5.1. Подсистема сбора, обработки и загрузки данных

2.5.1.1. Функция «Создания новой записи в записной книге»

Описание возможного сценария для последующей реализации задачи «Создания новой записи в записной книге» приведено в таблице.

| Подзадача | Действие |
|--|---|
| Создания новой записи в записной книге | <ul style="list-style-type: none"> - Пользователь открывает меню тапом по кнопке меню. - Выбирает пункт меню «Записная книга», тем самым обращаясь к этому модулю. - Модуль предоставляет средства для ведения записной книги. - Пользователь тапом по знаку «+» открывает экран создания новой записи. - Пользователь вносит текстовые данные. - В качестве даты берется текущая дата (День:Месяц). - При удержании тапа на записи пользователь может отредактировать или удалить запись. - По тапу на треугольнике в заголовке пользователь может сортировать список по возрастанию даты или по убыванию. |

2.6. Решения по составу информации, объему, способам ее организации, видам машинных носителей, входным и выходным документам и сообщениям, последовательности обработки информации и другим компонентам

2.6.1. Описание информационной базы

В табличном виде приведен перечень и описание предметных областей модели данных хранилища данных.

| Предметная область | Описание |
|--------------------------|---|
| Пользовательские профили | <p>В данной области возможен анализ пользовательских профилей.</p> <p>Например, из данной области можно получить информацию на запросы следующего характера:</p> <ul style="list-style-type: none"> - Общую статистику по профилям - Месторасположение пользователя (страна, город) - прочее |

После чего в табличной форме приводится наименование и описание каждой сущности предметной области модели данных.

| Сущность модели данных | Описание сущности |
|------------------------|---|
| Профиль | <p>Пользовательские данные.</p> <p>**Пример**</p> <p>Гай Монтэг</p> <p>Россия, Владивосток</p> <p>24 года</p> <p>Сегодня свободен от дел (Статус)</p> <p>Последнее месторасположение – Донецкая областная Филармония</p> <p>Друзей – 79</p> <p>Общих друзей – 22</p> |

2.6.2. Решения по пользовательскому интерфейсу

Из пункта «Требования к численности персонала» технического задания берется список ролей администраторов системы

Макеты интерфейса программного продукта для администраторских ролей.

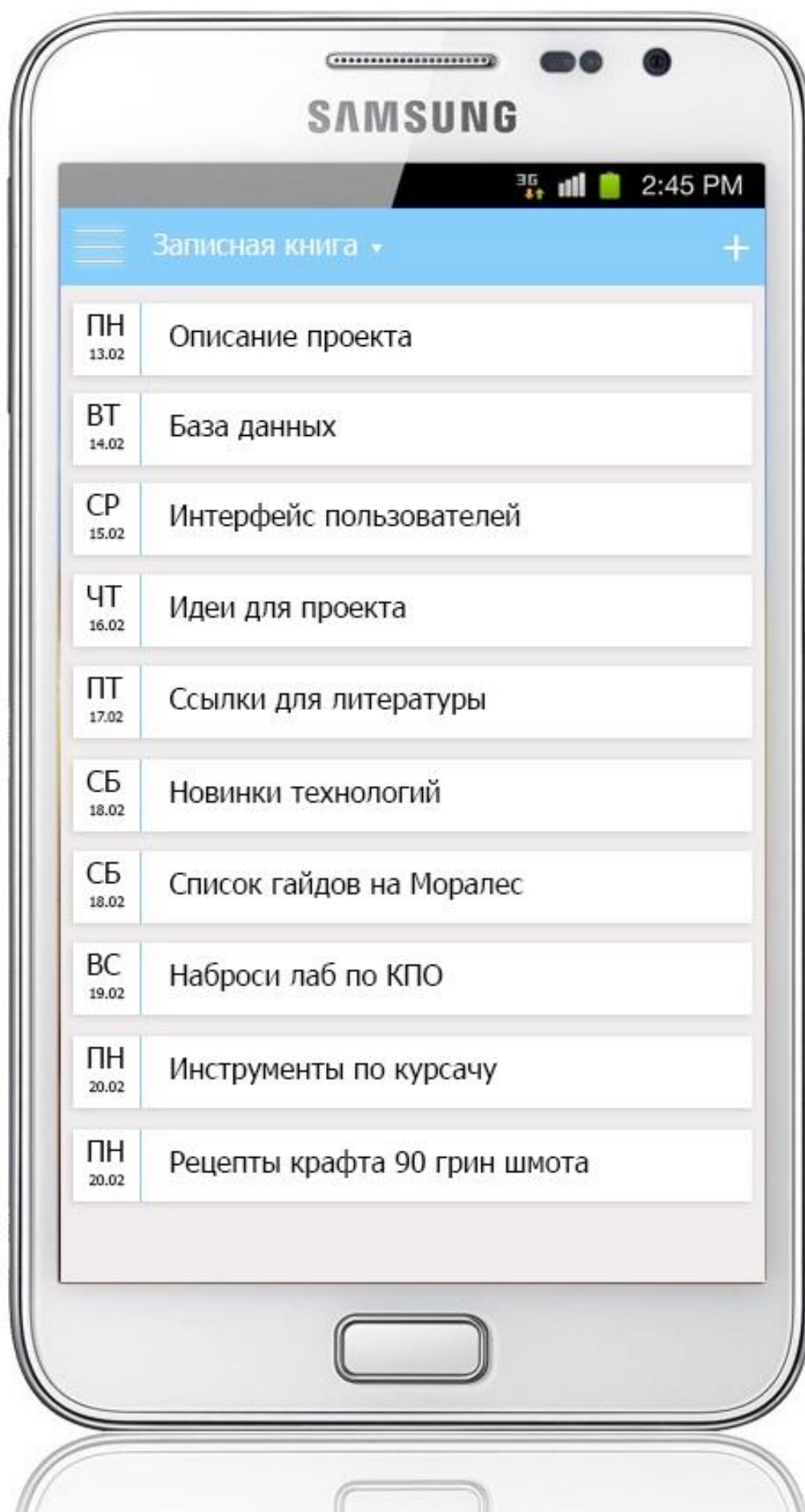
Макеты интерфейса программного продукта для конечного пользователя.

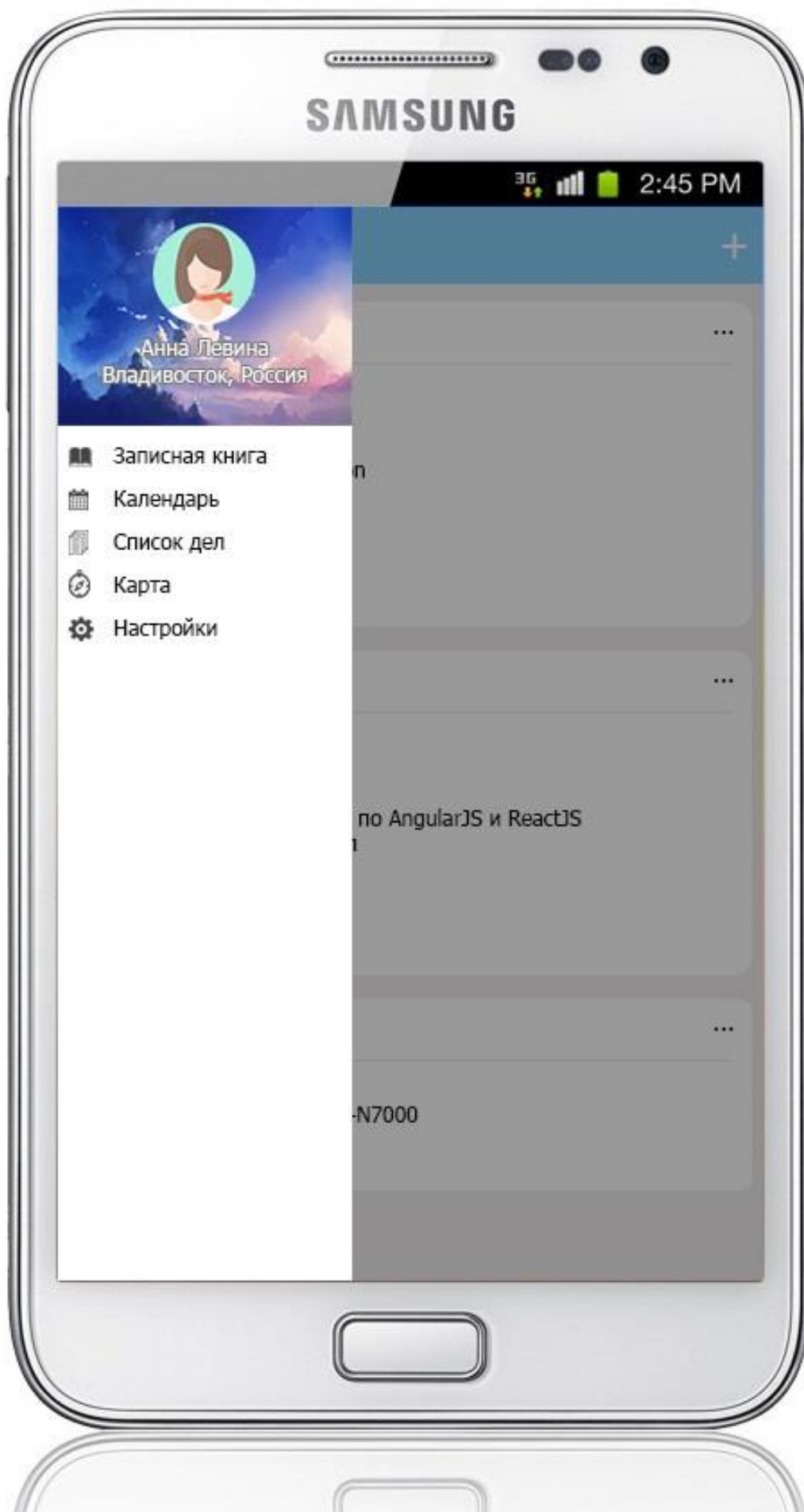
Пример экранной формы для ведения записной книги:

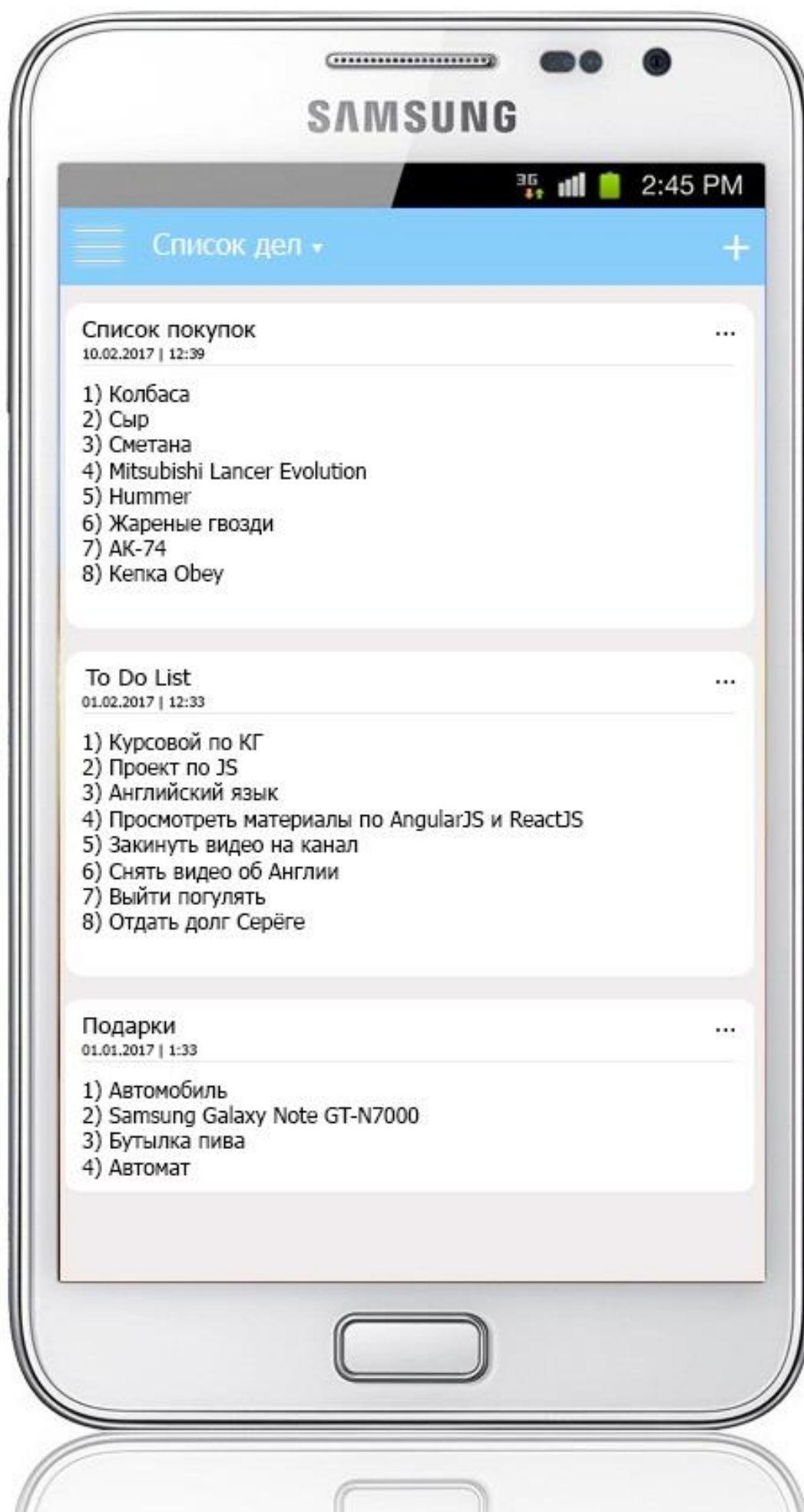
- 1 – кнопка – меню, содержащее список доступных инструментов.
- 2 – название раздела.
- 3 – сортировка записей по дате (убыванию или возрастанию).
- 4 – кнопка добавления новой записи.
- 5 – запись, в которой указывается дата добавления в формате (День недели, день, месяц).

Пример экранной формы меню:

- 1 – фотография текущего пользователя (изображение, установленное пользователем в качестве изображения профиля).
- 2 – Фамилия и имя текущего пользователя.
- 3 – Местоположение текущего пользователя (город и страна).
- 4 – пункты меню.









2.7. Методы и средства разработки

В качестве основного языка программирования будет использоваться JavaScript, так как будет разрабатываться гибридное приложение, которое будет работать не только на мобильной платформе, но также существует возможность разместить разработанное ПО в сети интернет в виде сайта, не задействуя больших усилий.

В качестве программных средств будут использоваться:

Brackets – удобный и расширяемый дополнениями редактор кода, в первую очередь направленный на frontend.

MySQL Workbench – удобное средство управления базами данных с графической оболочкой для быстрой разработки и сопровождения баз данных.

В качестве серверной операционной системы предполагается использование систем семейства Windows (XP SP3 и выше) или Linux (11 версии и выше).

Для работы с базой данных будет использоваться SQL стандарта СУБД MySQL.

Для макетирования и раскадровок будет использоваться программа Adobe Photoshop CS6. Для разработки пользовательского интерфейса будет использоваться фреймворк Angular.js. Так как разрабатывается гибридное приложение с одностраничным интерфейсом, а Angular.js – является лучшим JS фреймворком для этой цели.

Для разработки гибридного (мобильного) приложения будет использоваться фреймворк Ionic. Как один из самых производительных и документируемых фреймворков этой категории.

Моделирование разрабатываемого ПО и описание процессов разработки будет реализовываться посредством стандарта UML и программного обеспечения Rational Rose.

В качестве метода разработки ПО будет использован «Итерационный» так как разрабатываемое ПО достаточно малообъемное и будет разрабатываться в команде разработчиков состоящей из 3 человек.

3. СЛОВАРЬ ТЕРМИНОВ

- СУБД (Система управления базами данных) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

- База данных (БД) — это организованная структура, предназначенная для хранения информации. Обычно БД представляются в виде совокупности взаимосвязанных файлов или таблиц, предназначенных для решения конкретной задачи. С понятием БД тесно связано понятие системы управления базой данных (СУБД).

- ПО – это совокупность программ, позволяющих осуществить на вычислительном устройстве автоматизированную обработку информации.

- SQL – structured query language — «язык структурированных запросов» — формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД).;

- Итерационный метод разработки – это выполнение работ параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы. Проект при этом подходе в каждой фазе развития проходит повторяющийся цикл PDCA: Планирование — Реализация — Проверка — Оценка (англ. plan-do-check-act cycle).

- Макетирование – графический план будущего приложения.

- Раскадровка – последовательность рисунков, служащая вспомогательным средством при создании дизайна программы.

- API Google Maps – набор готовых классов, процедур, функций, структур и констант, предоставляемых картографическим приложением для использования во внешних средах.