**Содержание**

[Введение 4](#_Toc121915657)

[1 Теоретическая часть 8](#_Toc121915658)

[1.1 Разработка системы требований 8](#_Toc121915659)

[1.2 UML – проектирование 10](#_Toc121915660)

[1.3 Выбор языка и среды программирования 12](#_Toc121915661)

[2 Практическая часть 14](#_Toc121915662)

[2.1 Разработка ER-диаграммы 14](#_Toc121915663)

[2.2 Карта приложения 18](#_Toc121915664)

[2.3 Тестирование программного обеспечения 20](#_Toc121915665)

[Заключение 22](#_Toc121915666)

[Список использованных источников 26](#_Toc121915667)

Введение

Интернет имеет большой коммерческий потенциал и возможность перевода своего бизнеса на качественно новый уровень, поэтому любая серьезная компания в наше время имеет собственный сайт. Веб-студия — это компания, которая занимается разработкой и проектированием сайтов. Это всегда группа людей, где каждый отвечает за свою часть работы. Работа веб-студий актуальна в наше время, но помимо частого посещения страниц в браузерах, пользователи проводят еще больше времени в мобильных приложениях и телефон у них всегда под рукой. Поэтому для быстрой связи с пользователем актуально создание мобильного приложения. Таким образом, актуальность создания web-студии обуславливается необходимостью объединения разработчиков для более комфортной и организованной работы и сокращением времени на разработку при увеличении ее качества, сам сайт для web-студии станет ее лицом, поможет заявить о себе и продвинуть ее в сети Интернет, зарекомендовать себя на рынке и эффективнее выстроить работу, а мобильное приложение позволит установить удобное взаимодействие и общение между заказчиком и разработчиками путем реализации чата.

Проблема работы – как составить оптимальную модель программного продукта, позволяющего автоматизировать общение между заказчиком и исполнителями.

Объект исследования – программные продукты для автоматизации ИС Web-студии

Предмет исследования – разрабатываемый программный продукт.

Цель курсовой работы – спроектировать мобильное приложение для Web-студии, разработать часть функционала и протестировать разработанные функции.

Для реализации постеленной цели необходимо решить ряд задач:

1. Составить систему требований к разрабатываемому ПО.
2. Выявить основных лиц, взаимодействующих с приложением.
3. Составить диаграмму прецедентов.
4. Описать процесс взаимодействия пользователей с информационной системой.
5. Выбрать среду разработки и обосновать этот выбор.
6. Спроектировать базу данных с помощью диаграммы «сущность - связь»
7. Выявить основной функционал
8. Разработать алгоритм работы программного продукта по средствам диаграммы деятельности
9. Разработать тестовые сценарии для проверки работоспособности реализованного функционала.
10. Составить сопутствующую техническую документацию

В ходе выполнения курсовой работой необходимо подробно изучить работу Web-студий, разработать алгоритм работы таких функций программного обеспечения как:

* авторизация;
* вывод страницы сайта с процессом выполнения заказа внутри приложения;
* разработка чата между заказчиком и разработчиками.

Проанализировать различные средства автоматизации и выбрать наиболее подходящие по таким параметрам как знание языка разработки, удобство среды, рациональность выполнения. Структурировать систему требований, выявить пред причины разработки мобильное приложение Web-студии, составить систему ограничений, рассматривающей программный продукт с разных сторон. Проанализировать ситуации, при которых в работе программного продукта возникают исключительные ситуации. Выявить основных действующих лиц и описать их поведение в системе с помощью диаграммы вариантов использования. Описать процесс взаимодействия пользователей системы между собой и с информационной системой.

Описать необходимость использования языка SQL для реализации базы данных, системы управления базами данных PhpMyAdmin, языка программирования Java и среды разработки Android Studio для реализации подсистемы разрабатываемой информационной системы.

Для хранения такой информации как данные пользователей, информация о проектах и хранения заявок проектируется и разрабатывается база данных. Для этого выделяются необходимые сущности, указываются их атрибуты и связи между ними. На основе схемы хранения данных будет составлен словарь данных. На сервере устанавливается база данных и с помощью выбранной СУБД происходит обработка информации в ней. Сервер БД должен иметь минимальные технические характеристики:

* Процессор: 1 ядерный процессор;
* Тактовая частота: 3,2 ГГц или выше;
* Оперативная память: 1 ГБ или выше;
* Дисковое пространство: рабочие - RAID10 не менее 4 дисков (SCSI или SAS) суммарным объемом не менее 200 Гб;
* Для хранения архивных копий - RAID1 2 диска (SCSI или SAS) суммарным объемом не менее 500 Гб в расчете на 1 год работы в системе;
* Диск для горячей замены (Hot Spare).

После выгрузки схемы хранения данный на выделенный сервер составляется отчет о каждой созданной таблице с помощью специальных запросов.

Разработка интуитивно понятного интерфейса пользователя является не менее важным направлением работы. Поскольку интерфейс пользователя – это способ и средства взаимодействия пользователя с программами.

Для полноценной разработки пользовательского интерфейса разрабатывается карта приложения, которая отражает перечень и взаимодействие всех основных форм и доступного на них функционала.

После разработки всего функционала его необходимо протестировать. Для тестирования выбирается метод черного ящика. Метод черного ящика заключается в том, что система изучается не как совокупность взаимодействующих элементов, а как нечто целое (неделимое), взаимодействующее со средой на своих входах и выходах.

Помимо проверки работоспособности программного продукта со стороны пользователя его необходимо проверить на логику работы, для этого составятся Unit тесты, которые проверят работу транзакционных операция, например, вывод информации о проекте, о его статуе, изменении статуса и связи проектов с пользователями.

Для работы пользователей разработается программный документ руководство пользователя, содержащее алгоритм работы с мобильным приложением Web-студии, технические характеристики и всевозможные ошибки работы.

1 Теоретическая часть

1.1 Разработка системы требований

Для составления системы требованием необходимо четко определить проблему для дальнейшего ее решения.

1. Структурирование проблемы

|  |  |
| --- | --- |
| Элемент | Описание |
| Проблема | Как составить оптимальную модель программного продукта, позволяющего автоматизировать общение между заказчиком и исполнителями. |
| Воздействует на что (кого) и  результатом чего является | Проблема оказывает влияние на Заказчиков и Исполнителей.  От качества реализации зависит репутация компании, впечатление Заказчиков от качества выполнения работы и, следовательно, доходы компании. |
| Выигрыш от решения может  состоять в следующем | Предлагаемое решение: Создание мобильного приложения, где Заказчик сможет наблюдать за этапами выполнения заказа, а также для связи с Разработчиками путем чата. Преимущества от данного решения:  - повышения доверия со стороны заказчика;  - повышение репутации компании за счет внедрения дополнительных технологий;  - удобство использования отдельного приложения для моментальной связи с разработчиками;  - удобство просмотра процесса выполнения заказа, без обращения к компании напрямую и без посещения сайта; |

Таким образом необходимо составить оптимальную модель программного продукта, позволяющего автоматизировать общение между заказчиком и исполнителями.

Определение корневых причин, лежащих в основе проблемы:

1. Обращение в поддержку занимает слишком много времени;
2. Частые звонки по телефону для уточнения информации подходят не всем клиентам;
3. Частое обращение к сайту с телефона неудобно и занимает слишком много времени;
4. Использование социальных сетей для обращения в поддержку подходит как не всем клиентам, так и не всем сотрудникам.

Экономический покупатель системы – Руководитель Web-студии “Webex.

Пользователями системы являются Клиенты и Разработчики компании. Помимо них, система может оказать косвенное влияние на потенциальных клиентов компании, повышая репутацию web-студии.

После того, как система будет разработана, представлена и развернута, проводится тестирование от сотрудников компании, после которого система оценивается. Других внешних или внутренних пользователей системы, чьи потребности следует учесть нет.

После проведения тестирования и сдачи системы, ее сопровождением займутся сотрудники компании-заказчика. Управление системой передается администратору. Система имеет связь с Базой Данных, откуда и получает информацию. Взаимодействие проводится между веб-сайтом и мобильном приложением.

Ограничения для системы:

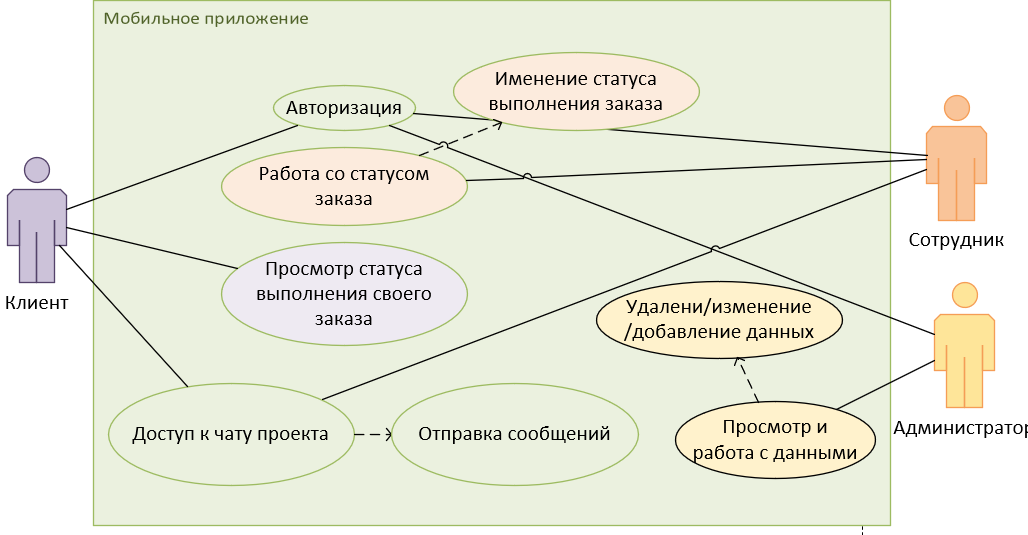
* обеспечение совместимости между мобильным приложением и web-сайтом;
* соответствуя требованиям безопасности, необходимо шифровать пароли в Базе Данных;
* разработка продукта в рамках законодательства о «Защите персональных данных» (Федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ).

1.2 UML – проектирование

Диаграмма прецедентов — это наиболее общее представление функционального назначения системы. На ней применяются два типа основных сущностей: варианты использования и действующие лица, между которыми устанавливаются следующие основные типы отношений:

* ассоциация между действующим лицом и вариантом использования;
* обобщение между действующими лицами;
* обобщение между вариантами использования;
* зависимости (различных типов) между вариантами использования.

Диаграмма прецедентов представлена на рисунке 1:

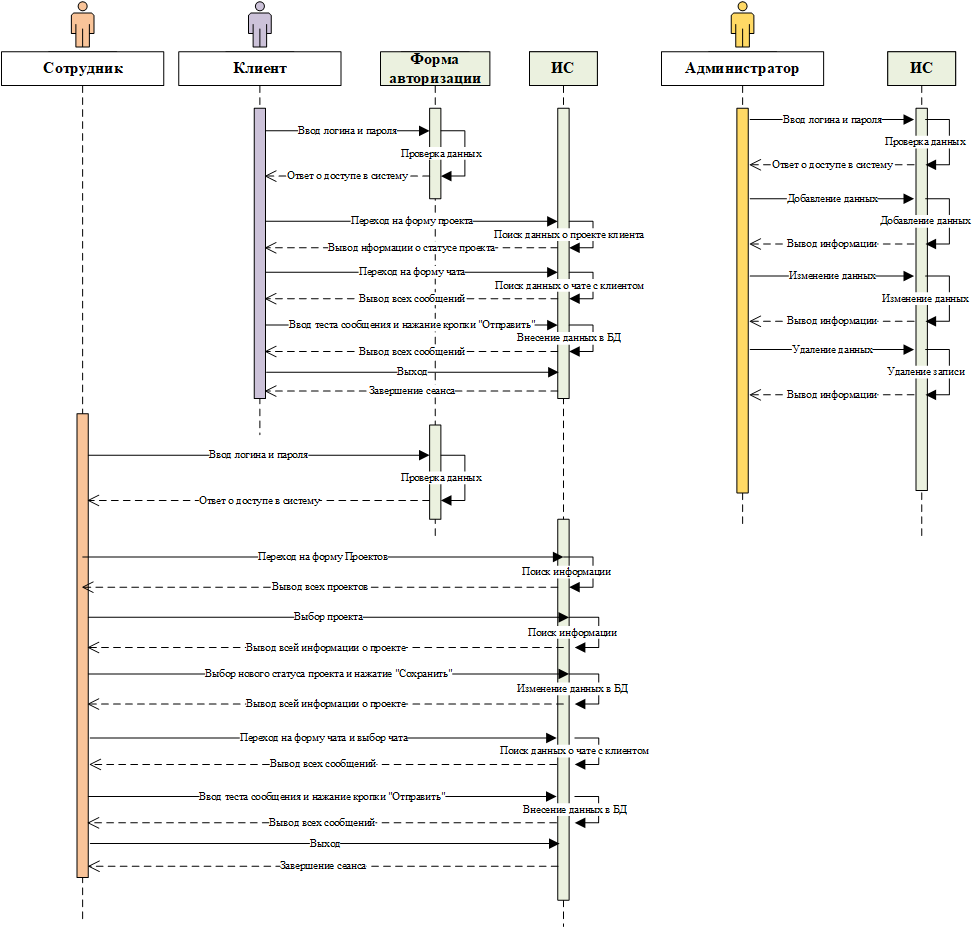


1. Дигамма прецедентов

Основными действующими лицами являются Клиенты и Сотрудники, между которыми происходит коммуникация в чате, что является основной функцией приложения. Дополнительно Клиенты могут просматривать информацию о своих проектов, а Сотрудники влиять на эту информацию. В качестве контролирующего лица выступает Администратор, в чьем вмешательстве как правильно нет необходимости, но в крайнем случае он сможет узнать необходимую информацию или изменить ее.

Диаграмма последовательности — диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления. Используется в языке UML. По сути, это запись протокола конкретного сеанса работы системы (или фрагмента такого протокола). В объектно-ориентированном программировании самым существенным во время выполнения является посылка сообщений взаимодействующими объектами. Именно последовательность посылки сообщений отображается на данной диаграмме.

Диаграмма последовательностей системы представлена на рисунке 2:



1. Дигамма последовательностей системы

Для доступа в систему под любым пользователем необходима авторизация. Регистрация всех пользователей производится руководителем на отдаленном сервере, так как регистрация клиентов происходит только после заключения с ними контракта и после начала работы над их проектом, сотрудников регистрирует руководство, так что в самостоятельной регистрации нет практичного смысла. После авторизации пользователям доступен ряд функций, показанных на диаграмме.

1.3 Выбор языка и среды программирования

Для программной реализации модуля представлены следующие языки программирования: C++, Java, C#, Kotlin и такие среды разработки, как Apache NetBeans IDE, IDE Eclipse, Android Studio.

В качестве языка программирования выбран язык Java. Это строго типизированный объектно-ориентированный язык программирования общего назначения. Отличительной особенностью Java в сравнении с другими языками программирования общего назначения является обеспечение высокой продуктивности программирования, нежели производительность работы приложения или эффективность использования им памяти. Огромное преимущество Java заключается в том, что на этом языке можно создавать приложения, способные работать на различных платформах. Также для Java написано много фреймворков и библиотек, руководств и инструкций, что упрощает изучение и использование языка.

Для выбора среды разработки необходимо понимать, что приложение будет создаваться для мобильных телефонов на базе андроид, относительно немногие разработчики используют NetBeans специально для платформы Android, следовательно, много информации и библиотек для подобного рода разработки будет проблематично найти.

Среды разработки Android Studio и Eclipse имеют как схожести, так и различия и оба хорошо подходят для мобильной разработки, но для реализации всех задач разрабатываемого приложения наилучшим вариантом является Android Studio, так как в ней лучшая поддержка новых функций.

Android Studio - интегрированная среда разработки для работы с платформой Android, отличающаяся удобством графического интерфейса и средствами отладки. Также основными особенностями Android Studio является возможность вёрстки в реальном времени, доступно множество вариантов размеров и разрешений экранов. Присутствует раздел справки, что намного облегчает работу в среде разработки. Встроены инструменты улучшения качества приложений, доступно средство взаимодействия с бета-тестерами и много другое.

Главные возможности среды разработки:

* включены все «интеллектуальные» возможности по редактированию кода - авто дополнение, рефакторинг и анализ кода;
* есть возможность визуального просмотра будущего приложения;
* сборка проекта работает быстрее, чем в альтернативных средах разработки;
* удобный конструктор интерфейсов;
* встроенный SDK Manager;
* удобное и интуитивно понятное логирование проекта;
* указанные цвета и рисунки, использованные в layout'e отображаются на границе в виде небольших превью, которые легко помогают понять какой конкретно ресурс используется;
* управление локализацией и переводами;
* удобное взаимодействие с СУБД PhpMyAdmin;
* предпросмотр приложений на разных разрешениях, языках, и версиях API во встроенном эмуляторе.

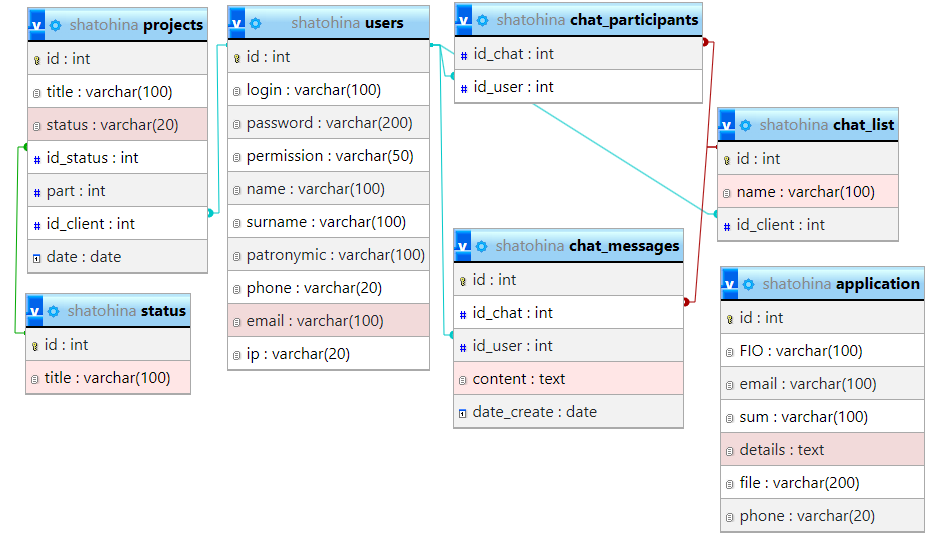
Таким образом разработка будет производится в среде разработке Android Studio на языке Java.

2 Практическая часть

2.1 Разработка ER-диаграммы

ER-диаграмма — это разновидность блок-схемы, где показано, как разные «сущности» связаны между собой внутри системы. ER-диаграммы чаще всего применяются для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса. ER-диаграммы полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные линии, для отображения сущностей, их атрибутов и связей. Эти диаграммы устроены по тому же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи — глаголов.

На рисунке 3 представлена ER-диаграмма базы данных информационной системы «Web-студия WEBEX»



1. ER-диаграмма БД ИС «WEBEX»

На ER-диаграмме представлены основные таблицы ИС. Разрабатываемое мобильное приложение обращается ко всем таблицам, кроме таблицы application.

Словарь данных представлен на таблицах 2-6 и структура таблиц, соответственно, на рисунках 4-9

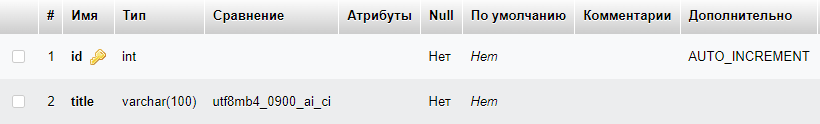
1. «Проекты»

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| **projects** | | | |
| Первичный | id | Да | Номер проекта |
|  | title | Да | Наименование проекта |
|  | status | Да | Статус выполнения |
|  | part | Да | Этап выполнения |
| Внешний | id\_client | Да | Внешний ключ к таблице «Пользователи» |
|  | date | Да | Дата создания |



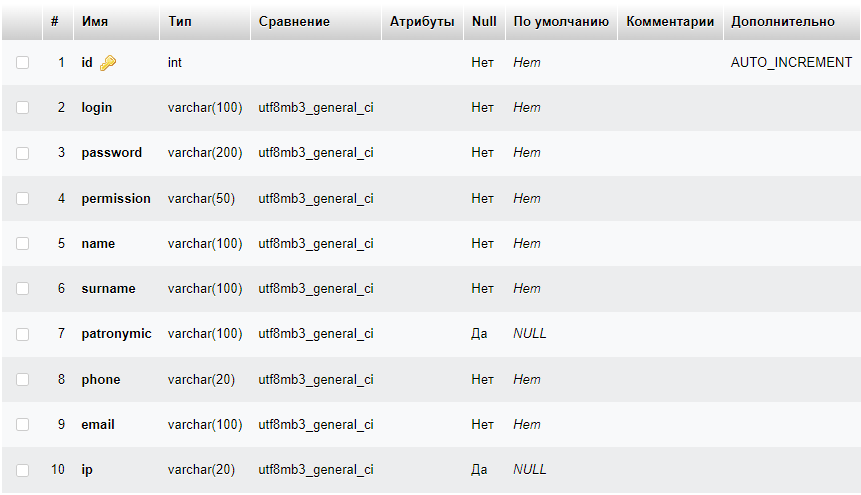
1. Структура таблицы «Проекты»
2. «Список статусов проекта»

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| **status** | | | |
| Первичный | id | Да | Номер варианта |
|  | title | Да | Наименование варианта |



1. Структура таблицы «Список статусов проекта»
2. «Пользователи»

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| **users** | | | |
| Первичный | id | Да | Номер пользователя |
|  | login | Да | Логин |
|  | password | Да | Пароль |
|  | permission | Да | Права доступа |
|  | name | Да | Имя |
|  | surname | Да | Фамилия |
|  | patronymic | Нет | Отчество |
|  | phone | Да | Номер телефона |
|  | email | Да | Почта |
|  | ip | Да | Ip-адрес |



1. Структура таблицы «Пользователи»
2. «Список чатов»

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| **chat\_list** | | | |
| Первичный | id | Да | Номер чата |
|  | name | Да | Наименование чата |
| Внешний | id\_client | Да | Внешний ключ к таблице «Пользователи» |



1. Структура таблицы «Список чатов»
2. «Участники чата»

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| **chat\_participants** | | | |
| Внешний | id\_chat | Да | Внешний ключ к таблице «Список чатов» |
| Внешний | id\_user | Да | Внешний ключ к таблице «Пользователи» |



1. Структура таблицы «Участники чата»
2. «Сообщения»

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| **chat\_messages** | | | |
| Первичный | id | Да | Номер сообщения |
| Внешний | id\_chat | Да | Внешний ключ к таблице «Список чатов» |
| Внешний | id\_user | Да | Внешний ключ к таблице «Пользователи» |
|  | content | Да | Текст собщения |
|  | date\_create | Да | Дата создания |



1. Структура таблицы «Сообщения»

Таблица application содержит в себе необработанные заявки потенциальных клиентов, это информация не используется в мобильном приложении.

2.2 Карта приложения

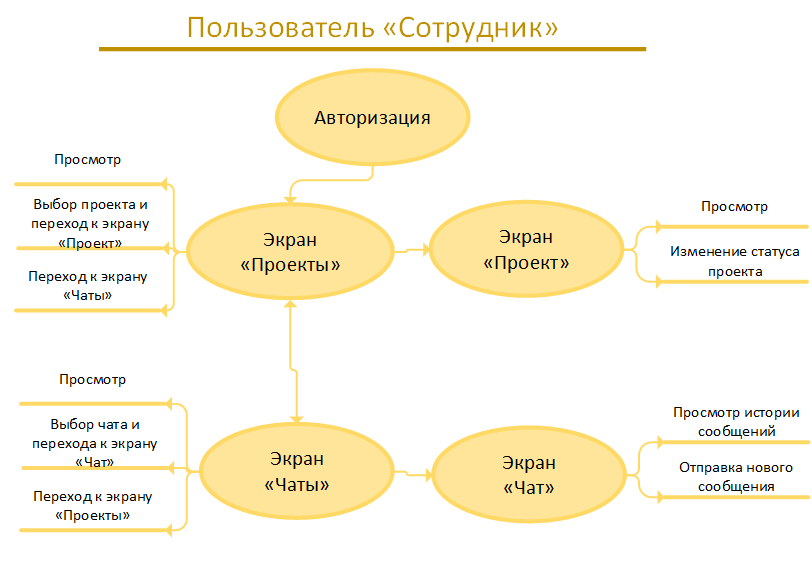
Карта приложения – это карта, показывающая логику приложения в удобном графическом формате, на ней показаны основные окна приложений и функции, доступные на них.

Карта приложения пользовательской части приложения для пользователя «Клиент» приведена на рисунке 10



1. Карта приложения для Клиента

Карта приложения пользовательской части приложения для пользователя «Сотрудник» приведена на рисунке 11



1. Карта приложения для Сотрудника

Некоторые формы имеют единый вид, но в зависимости от пользователя, зашедшего на него, функционал может меняться.

2.3 Тестирование программного обеспечения

При тестировании методом «белого ящика» (англ. white-box testing, также говорят — прозрачного ящика) разработчик теста имеет доступ к исходному коду и может писать код, который связан с библиотеками тестируемого ПО. Это типично для юнит-тестирования (англ. unit testing), при котором тестируются только отдельные части системы. Оно обеспечивает то, что компоненты конструкции работоспособны и устойчивы до определенной степени. Тестовый сценарий описан в таблице 8

1. Тестирование методом белого ящика, функция авторизации

|  |  |  |  |
| --- | --- | --- | --- |
| Тест | Ожидаемый  результат | Фактический  результат | Результат  тестирования |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Тестирование методом белого ящика, функция авторизации

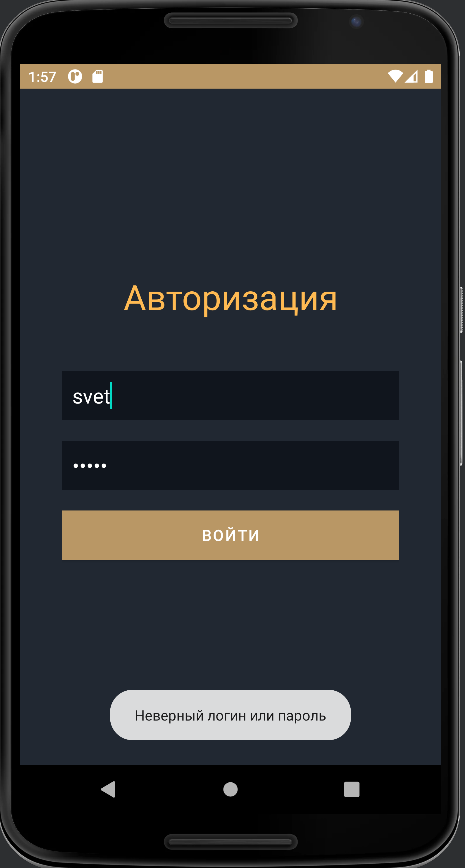
|  |  |  |  |
| --- | --- | --- | --- |
| Тест | Ожидаемый  результат | Фактический  результат | Результат  тестирования |
| Проверка авторизации | Авторизация под определенной ролью | Авторизация под определенной ролью | Успешная авторизация |
| Проверка передачи текста в сообщение | Добавление записи в БД | Добавление записи в БД | Текст в сообщение передаётся |
| Проверка смены статуса проекта | Статус проекта был изменен в БД | Статус проекта был изменен в БД | Успешная смена проекта |

При тестировании методом «черного ящика» (англ. black-box testing) тестировщик имеет доступ к ПО только через те же интерфейсы, что и заказчик или пользователь, либо через внешние интерфейсы, позволяющие другому компьютеру либо другому процессу подключиться к системе для тестирования. Например, тестирующий модуль может виртуально нажимать клавиши или кнопки мыши в тестируемой программе с помощью механизма взаимодействия процессов с уверенностью в том, что эти события вызывают тот же отклик, что и реальные нажатия клавиш и кнопок мыши. Тестовый сценарий приведен в таблице 9

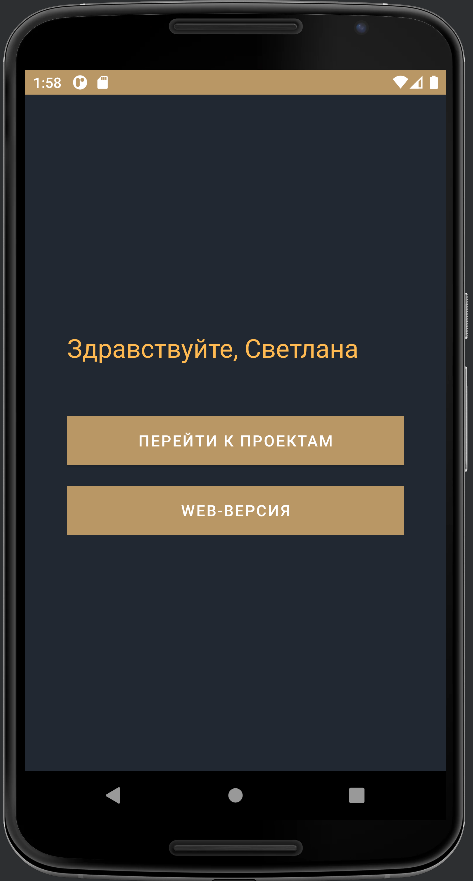
1. Тестирование методом черного ящика

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Поле названия карточки | Ожидаемый результат | Что проверяется |
| 1 | Ввод верных данных | Авторизация | Вход в систему |
| 2 | Пустое значение или неверные данные | Вывод сообщения | Правильность ввода |
| 3 | Смена статуса проекта | Вывод измененного статуса | Изменение данных |
| 4 | Отправка сообщения | Вывод отправленного сообщения | Добавление сообщения |

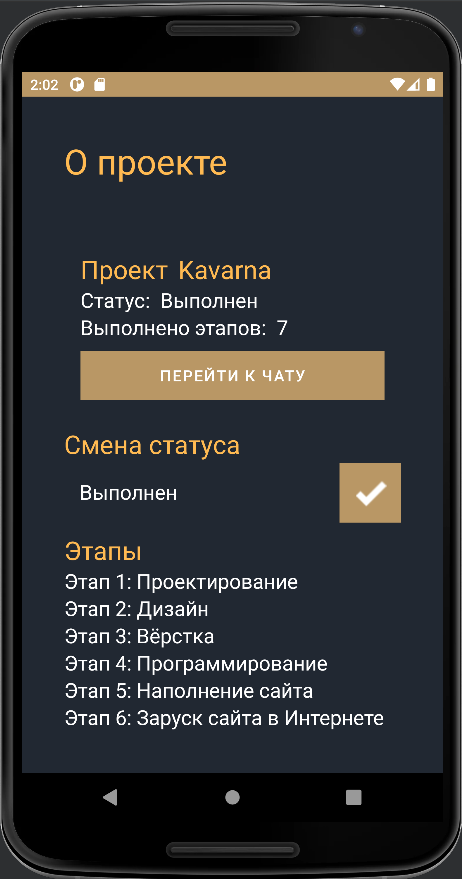
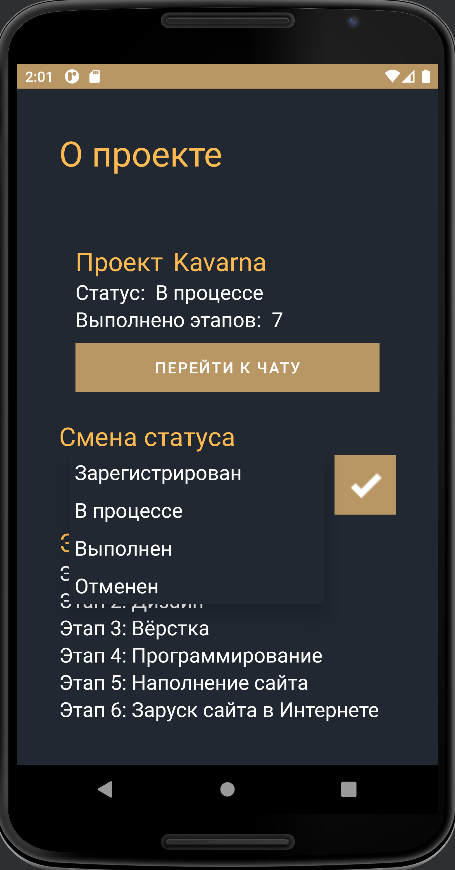
Ниже на рисунках 12-15 приведено тестирование функции составления слова



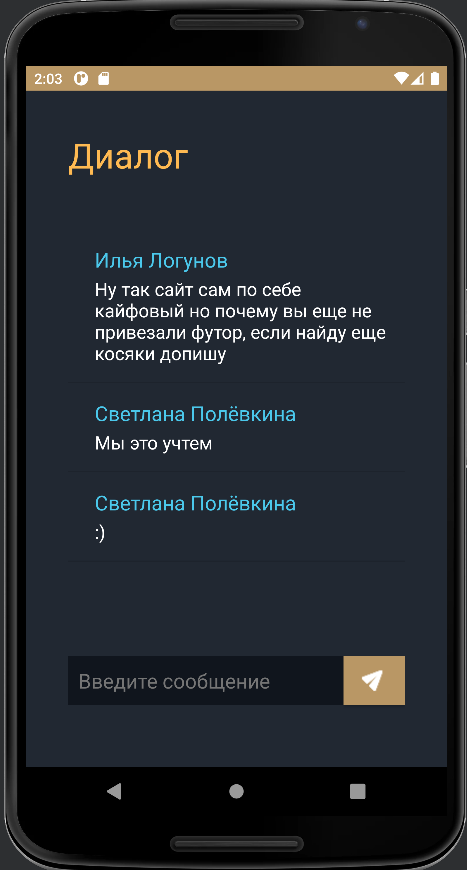
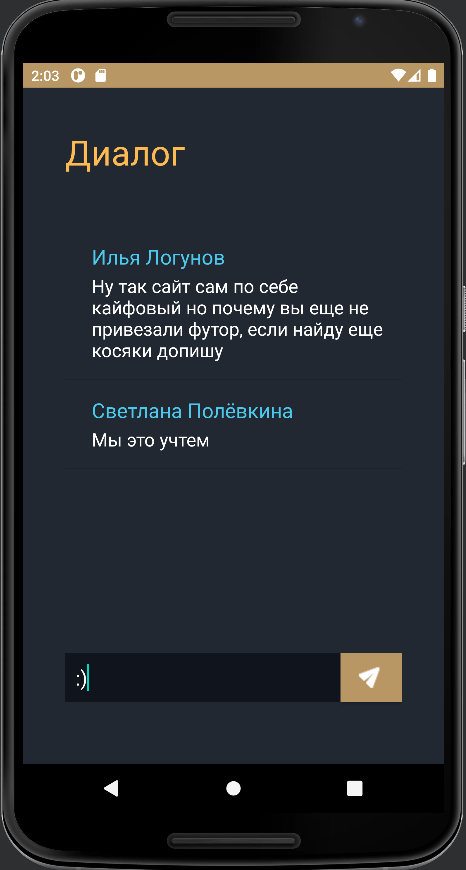
1. Ввод неверных данных



1. Ввод верных данных



1. Смена статуса



1. Отправка сообщения

Заключение

Целью курсовой работы являлась проектирование мобильного приложения для Web-студии, разработка его функционала и дальнейшее тестирование. Для поставленной цели выполнены следующие задачи:

1) Составлена система требований к разрабатываемому ПО.

2) Выявлены основные лица, взаимодействующих с приложением.

3) Составлена диаграмма прецедентов.

4) Описан процесс взаимодействия пользователей с информационной системой.

5) Обоснованно выбрана среда разработки.

6) Спроектирована база данных с помощью диаграммы «сущность - связь»

7) Выявлен основной функционал

8) Разработан алгоритм работы программного продукта по средствам диаграммы деятельности

9) Разработан тестовый сценарий для проверки работоспособности реализованного функционала.

10) Составлена сопутствующая техническая документация

Был разработан функционал для 2 ролей внутри системы:

* Клиент;
* Сотрудник.

В удаленной базе данных веб-сайта, к которой подключается мобильное приложение, существуют еще две роли:

* Руководитель;
* Администратор.

Внутри мобильного приложения достаточно использование двух ролей, поэтому любой руководитель или администратор смогут попасть в мобильное приложение под своими данными, но их функционал не будет отличаться от функционала Сотрудников.

Функционал Клиента:

* Просмотр статуса выполнения своего заказа;
* Доступ к чату;
* Возможность написания сообщения и получения ответа.

Функционал Сотрудника:

* Просмотр всех заказов;
* Просмотр статуса выполнения всех заказов;
* Смена статуса проектов;
* Доступ к чату;
* Возможность написания сообщения и получения ответа.

Перед началом любой разработки, необходимо осуществлять сбор информации. Сбор информации – важнейшая часть любого проекта вне зависимости от его масштабов. Он необходим для понимания и удовлетворения потребностей клиентов. Процесс сбора требований включает в себя определение и документирование требований клиентов, пользователей, заинтересованных сторон и т.д., связанных с проектом. Эти знания будут использоваться для разработки различных решений: продуктов, услуг, программного обеспечения и т.д. В состав требований в работе входила авторизация пользователей, разработка учебных карточек и чтение статей.

По составленным требованиям в ходе работы разработано мобильное приложение для IT компании «WEBEX», которая является Web-студией и осуществляет ряд работ по созданию сайтов своим клиентам.

В состав разработанного функционала входит авторизация (регистрация пользователей производится сотрудниками самостоятельно через веб-сайт). Все сотрудники могут просматривать проекты, менять их статус, а также общаться с клиентом путем чата. Клиенты имеют доступ только к своему проекту, могут просматривать его статус и общаться с сотрудниками компании путем чата.

Для лучшего учета всех функций, спроектированы диаграммы, в которых учтены требования к программному продукту и описаны взаимодействия пользователей.

В качестве языка программирования выбран язык Java. Это строго типизированный объектно-ориентированный язык программирования общего назначения. Отличительной особенностью Java в сравнении с другими языками программирования общего назначения является обеспечение высокой продуктивности программирования, нежели производительность работы приложения или эффективность использования им памяти.

Идеальной средой разработки для реализации всех задач разрабатываемого приложения является Android Studio.

Для хранения данных спроектирована база данных при помощи диаграммы «сущность связь», т.е. интерфейс, позволяющий двум независимым компонентам программного обеспечения обмениваться информацией, а также составлен словарь данных, который позволил закрыть пробелы возникшие при проектировании диаграммы, и подвести в необходимой версии базы данных, для хранения нужной информации правильного функционирования веб-сайта.

Для отображения алгоритма работы веб-ресурса разработаны диаграммы деятельности, а именно диаграммы использования и диаграммы последовательности. Проектирование UML играет важную роль в разработке программного обеспечения и веб-сайтов, а также в системах, не связанных с ИТ, во многих отраслях, поскольку он дает возможность визуально показать поведение и структуру системы или процесса. UML помогает продемонстрировать возможные ошибки в структурах приложений, поведении системы и других бизнес-процессах.

На проверку работоспособности разработаны тестовые сценарии методом черного и белого ящика, а также проведено Unit-тестирование. Тестирование проходили функции регистрации, авторизации, функции сотрудников и прав доступа к страницам и функционалу. При тестировании были выявлены недочеты и ошибки, большинство из которых были решены, самые сложные и затруднительные оставлены на дальнейшую разработку.

Цель тестирования — проверка соответствия предъявляемым требованиям, обеспечение уверенности в качестве разработки, поиск очевидных ошибок в продукте, которые должны быть выявлены до того, как их обнаружат пользователи программы. Поэтому очень важно перед выходом приложения в эксплуатацию к реальному пользователю провести качественное тестирование со многих сторон как со стороны кодовой части, соответствующий методу белого ящика, так и со стороны пользовательского интерфейса, методом черного ящика.

Для эксплуатации мобильного приложения, разработано руководство пользователя, в которым расписан основной функционал для поддержания деятельности ресурса, выделено посредством чего осуществляется доступ и что для этого необходимо, а также выделены основные сообщения пользователю.

Разработанный продукт, помимо внедрения в систему компании WEBEX, в дальнейшем может быть использован в качестве шаблона для аналогичных систем, а также в качестве примера системы и изучения мобильной разработки студентами.

Преимущество использования системы для самостоятельного просмотра статуса выполнения проекта Клиентами на самом деле является удобной. Клиентам не нужно постоянно звонить в компанию, попадая на перерывы и ожидая ответа, а сотрудникам не придется постоянно быть на линии, и они смогут выполнять свою работу, не беспокоясь об этом. Учитывая, что не все веб-студии работают с офиса, и чаще всего предпочитают удаленную работу, то назначать чей-то личный телефон – телефоном приема бесконечных звонков, не лучшая идея.

Потенциальный клиент может оставить заявку и дождаться звонка или сообщения на телефон/почту, после заключения договора он получает данные для доступа к ресурсу, где в любой момент может узнать, на какой стадии находится его проект. А в случае возникновения вопросов, он сможет оставить сообщение и получить на него ответ, что упрощает обращение к компании и позволяет не вступать лишний раз в прямую коммуникацию.

Список использованных источников

1. Губко, А. А. Электрооборудование и электроснабжение горных предприятий: учеб.пособие / А. А. Губко, Е. А. Губко. — Новосибирск :Академиздат, 2017. - 531 с.
2. Суворин, А. В. Монтаж и эксплуатация электрооборудования систем электроснабжения : учеб.пособие / А. В. Суворин. — Красноярск : СФУ, 2018. — 400 с. — URL: https://e.lanbook.com/book/117768 (дата обращения: 26.02.2021).
3. Amit Phaltankar. Amitph. Документация для работы с Java Persistence. [Электронный ресурс] – URL: https://www.amitph.com/java-persistence-api-guide/ (дата обращения: 04.08.2022).
4. Baeldung. Документация по работе с чтением файлов. [Электронный ресурс] – URL: https://www.baeldung.com/reading-file-in-java (дата обращения: 27.07.2022).
5. Datanucleus. Документация для работы с Java Persistence. [Электронный ресурс] — URL: https://www.datanucleus.org/products/accessplatform\_5\_1/jpa/query.html#jpql (дата обращения: 11.08.2022)
6. Educba. Документация для работы с Java Persistence. [Электронный ресурс] – URL: https://www.educba.com/jpql/ (дата обращения: 11.08.2022)
7. FlatLaf. Документация для работы с технологиями FlatLaf. [Электронный ресурс] – URL: https://www.formdev.com/flatlaf/ (дата обращения: 04.08.2022)
8. Java2s. Документация по работа с Java Swing компонентами. [Электронный ресурс] – URL: Lhttp://www.java2s.com/Tutorials/Java/Swing\_How\_to/JFrame/Change\_the\_color\_of\_titlebar\_in\_JFrame.htm (дата обращения: 04.08.2022)
9. MigLayout. Документация для работы с Java Layout Manager. [Электронный ресурс] – URL: https://www.miglayout.com/ (дата обращения: 03.08.2022)
10. Rukovodstvo. Документация по работе с файлами. [Электронный ресурс] – URL: https://rukovodstvo.net/posts/id\_746/?ysclid=laz3t698yd116993750#filewriter (дата обращения: 26.07.2022)
11. Wayword Notes. Использование Java Swing Animation. [Электронный ресурс] – URL: https://ernienotes.wordpress.com/2010/12/08/using-java-swing-animation/ (дата обращения: 03.08.2022)
12. Канакова, С.Г. Методические указания по выполнению курсовой работы по МДК.01.01 Разработка программных модулей: Методические указания / С.Г. Канакова. - Прокопьевск, 2022. – 30 с. (дата обращения: 12.08.2022)

Приложение А

(Обязательное)

Руководство пользователя

Государственное бюджетное профессиональное образовательное учреждение

Прокопьевский горнотехнический техникум им. В.П. Романова

Руководство пользователя

Разработка модулей информационной системы веб-студии

Версия: 1.0

КР.РП.01З

Разработал: Шатохина В.А.

2022

Программный продукт представляет собой мобильное приложение (далее приложение), под наименованием «Webex», в честь названия компании, которой оно принадлежит.

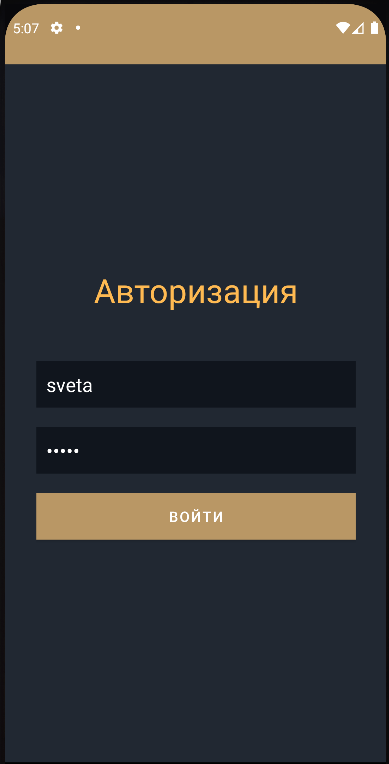
Приложение автоматизирует процесс доступа Заказчиков к информации о его проекте, а также позволяет получить персональный чат для связи с сотрудниками компании. Таким образом для доступа к информации о своем заказа, клиенту не нужно будет звонить в компанию, всю основную информацию по проекту он сможет получить в любую минуту, и при необходимости связаться с компанией.

Для использования приложение, необходимо скачать его на телефон с и запустить:



1. Иконка скаченного приложения

После установки, оно появится на устройстве. Его необходимо запустить. После запуска откроется страница авторизации. Данные предоставляются клиентам компании. Их необходимо ввести в соответствующие поля и нажать кнопку «войти», форма авторизации представлена на рисунке 2



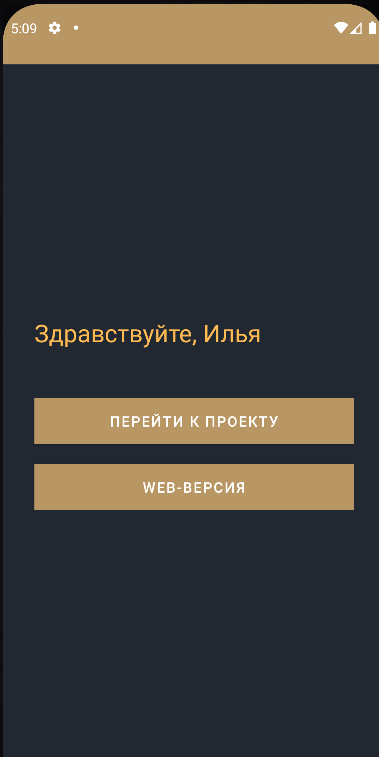
1. Форма «Авторизации»

Пользователь «Клиент»

После авторизации пользователь окажется на странице личного кабинета, где может совершить два действия:

Кнопка «Перейти к проекту» - открывает форму «О проекте».

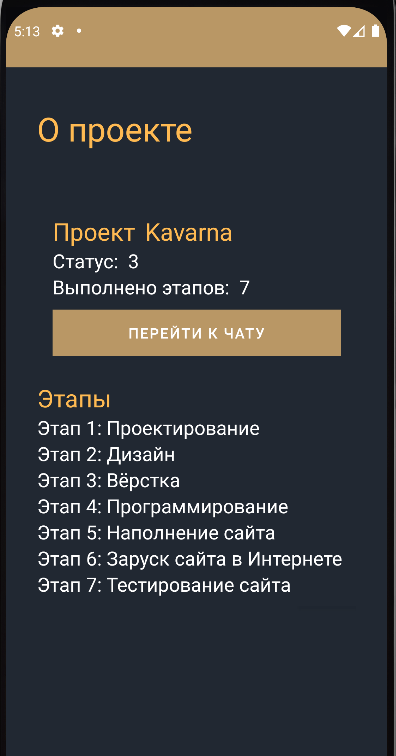
Кнопка «Web-версия» - открывает сайт компании внутри приложения.



1. Форма «Личный кабинет»

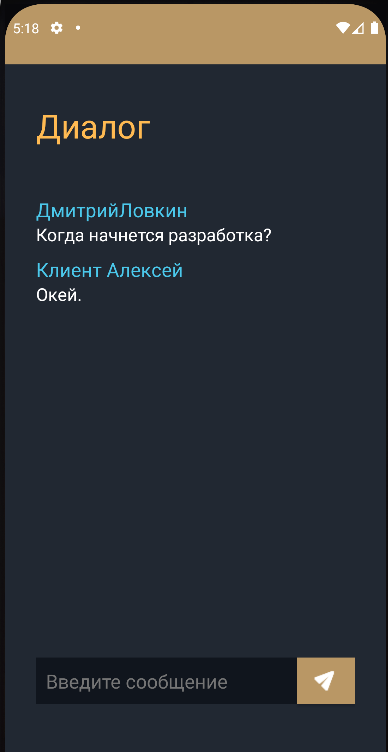
На форме о проекте отображено его название, статус выполняемого заказа и этап, на котором он находится. Для справки ниже описаны этапы разработки проекта.

Кнопка «Перейти к чату» - открывает форму «Чат» для связи с сотрудниками компании.



1. Форма «О проекте»

На форме «Чат» отображаются все сообщения, а также внизу есть поле для написания нового сообщения. После его заполнения необходимо нажать на кнопку .



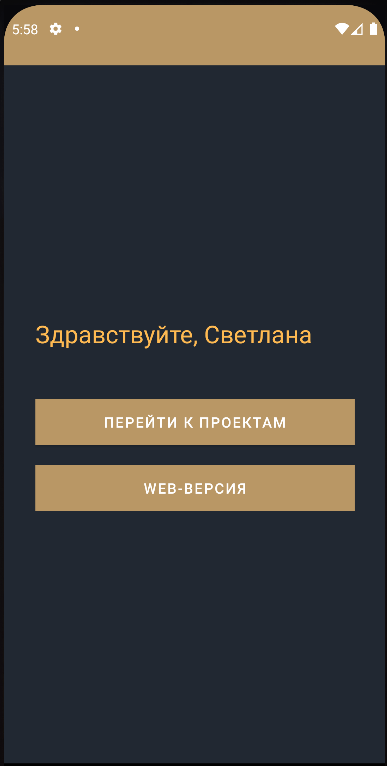
1. Форма «Чат»

Пользователь «Сотрудник»

После авторизации пользователь окажется на странице личного кабинета, где можно совершить два действия:

Кнопка «Перейти к проектам» - открывает форму «Проекты».

Кнопка «Web-версия» - открывает сайт компании внутри приложения.



1. Форма «Личный кабинет»

На форме «Проекты» отображен список всех проектов.

Кнопка «Просмотр» - открывает форму «О проекте».

Кнопка «Перейти к чату» - открывает форму «Чат» для связи с сотрудниками компании.

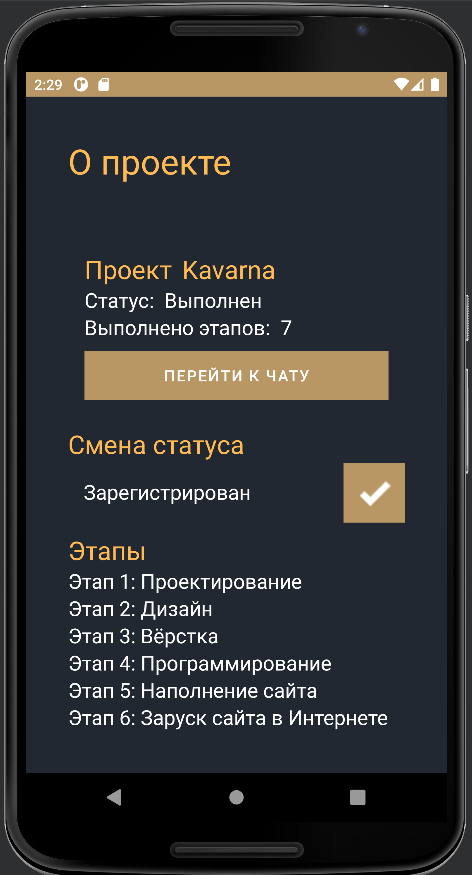


1. Форма «Проекты»

На форме о проекте отображено его название, статус выполняемого заказа и этап, на котором он находится. Для справки ниже описаны этапы разработки проекта.

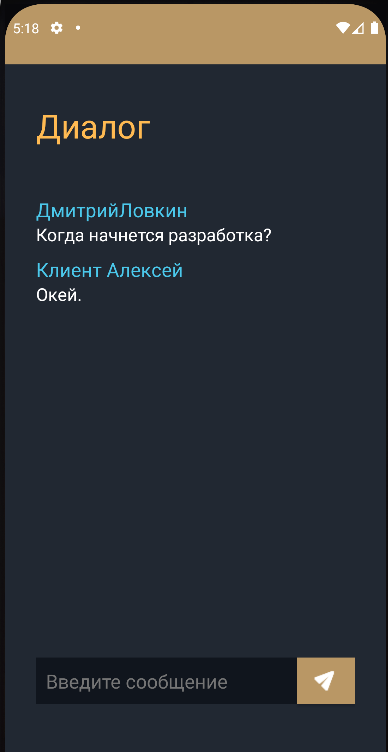
Кнопка «Перейти к чату» - открывает форму «Чат» для связи с клиентом компании.

В блоке «Смена статуса» при выборе статуса и нажатие кнопки происходит смена статуса.



1. Форма «О проекте»

На форме «Чат» отображаются все сообщения, а также внизу есть поле для написания нового сообщения. После его заполнения необходимо нажать на кнопку .



1. Форма «Чат»