ТЕХНИЧЕСКОЕ ЗАДАНИЕ
на разработку мобильного приложения.

I. НАЗНАЧЕНИЕ РАЗРАБОТКИ

1. Наименование объекта	Разработка мобильного, кроссплатформенного приложения для устройств на базе операционных систем Android и iOS.
2. Назначение разработки	Платформа позволяющая вести поиск мастеров для выполнения работ по ремонту и отделке помещений.
3. Перечень услуг и их объем	Разработать, протестировать, сдать заказчику исходники приложения, исходники backend'а, файлы для размещения приложения в Google Play и AppStore. Предоставить помощь в запуске проекта. (Поднятие реальных серверов, размещение приложения в магазинах)
4. Требования к оказанию услуг	Определяются в соответствии с пунктами технического задания: Пункт I «Назначение разработки», Пункт II «Применяемые условные обозначения основных объектов и сокращений» Пункт III «Технические условия» Пункт IV «ОПИСАНИЕ МОДЕЛЕЙ ДАННЫХ» Пункт V «Сервер API; Описание входных/выходных данных» Пункт VI «Описание ролей пользователя, статусов заказа и прав доступа.» Пункт VII «Описание структуры и интерфейса приложения» Пункт VIII «Макеты экранов»
5. Требования к безопасности оказания услуг и услугам, оказанным в полном объеме	При разработке мобильного приложения необходимо предусмотреть: - удобство, простоту и лаконичность пользовательского интерфейса - передачу данных между мобильным приложением и сервером через HTTPS - защиту информации на стороне сервера от предоставления лицам, которые не имеют прав на ее просмотр - непосредственный доступ к серверу только через логин и пароль
6. Требования к последующей технической поддержке	Исполнитель осуществляет бесплатную техническую поддержку мобильного приложения на протяжении двух месяцев. В случае возникновения каких-либо технических сбоев, он обязуется устранить их в сроки не более 1 недели.

Исполнитель_____ Заказчик_____

II. ПРИМЕНЯЕМЫЕ УСЛОВНЫЕ ОБОЗНАЧЕНИЯ ОСНОВНЫХ ОБЪЕКТОВ И СОКРАЩЕНИЙ

- JSON - (JavaScript Object Notation) текстовый формат обмена данными, основанный на JavaScript.
- API (Application Programming Interface) интерфейс взаимодействия между клиентской частью системы и сервером.
- Мастер – пользователь приложения, осуществляющий поиск заказов по ремонту и отделке помещений.
- Заказчик – пользователь приложения, размещающий заказы по ремонту и отделке помещений.
- Push-уведомления - один из способов распространения информации (контента) через сеть Интернет, данные поступают от поставщика к пользователю на основе установленных параметров.
- Книжный формат – вид верстки экрана, при котором устройство находится в вертикальном положении.

III. ТЕХНИЧЕСКИЕ УСЛОВИЯ.

1. Общие требования.

Приложение должно соответствовать требованиям площадок Google Play и App Store.

Обязательная реализация push-уведомлений с сохранением всех оповещений и возможности просмотра в интерфейсе программы.

Отправка оповещений реализуется при помощи сервиса Firebase Cloud Message.

Обмен данными с серверами компании должен происходить с помощью защищенного протокола передачи данных HTTPS в формате JSON по API, используя данные авторизации.

Обмен данными с сервером должен происходить только при непосредственном обращении к этим данным. Для возможности снижения нагрузки на канал обмена данными.

После загрузки и публикации новой версии приложения в магазин приложений, где оно размещено, должно быть предусмотрено автоматическое обновление приложения на устройствах пользователей.

Серверная часть приложения, включая базу данных, размещаются на виртуальном выделенном сервере на операционной системе Linux.

2. Требования к интерфейсу приложения

Пользовательский интерфейс приложения должен быть простым и интуитивно понятным. Элементы пользовательского интерфейса должны обеспечить простую навигацию пользователя в приложении, а также его информированность в любой момент времени.

Интерфейс Приложения должен быть представлен на русском языке.

Приложение предоставляет разные возможности взаимодействия для Мастера и Заказчика.

Интерфейс должен быть реализован в соответствии с дизайном макетов экранов, представленным в пункте VIII «Макеты экранов» данного технического задания. Исполнитель оставляет за собой право вносить небольшие изменения в интерфейс для более эффективного использования возможностей платформ Android и iOS. Все изменения, вносимые со стороны исполнителя, должны быть заранее согласованы с заказчиком в устной или письменной форме.

3.Технические требования для мобильного приложения

Для разработки мобильного приложения должны использоваться SDK Flutter с открытым исходным кодом и язык программирования Dart.

Приложение должно стабильно работать на мобильных устройствах на базе операционных систем:

- Android версий от 5 до 10 включительно
- iOS версий от 9 до 14 не включая

Верстка приложения является адаптивной рассчитанной на устройства с экранами от 4 до 6.5 дюймов и осуществляется в книжном формате, при повороте устройства в горизонтальное положение изменения верстки не предусмотрены.

4. Требования к предоставляемой информации

Для разработки приложения требуется регистрация следующих аккаунтов на сторонних сервисах:

- Аккаунт на хостинге. Для аренды виртуального сервера и размещения на нем серверной части приложения
- Аккаунт в google. Для использования функция сервиса Firebase
- Аккаунт в Вконтакте. Для использования их API предоставления информации о городах
- Аккаунт Apple. Для размещения приложения в App Store
- Аккаунт разработчика Google Play. Для размещения приложения в Play Market
- Аккаунт Robokassa. Для проведения платежных операций

Для регистрации вышеуказанных учетных записей заказчик предоставляет необходимую информацию

Исполнитель_____ Заказчик_____

IV. ОПИСАНИЕ МОДЕЛЕЙ ДАННЫХ

1. Пользователи (users)

Field	Type	Null	Default	Unique	Desc
id	varchar(100)	NO		YES	Идентификатор пользователя, генерируется Google Firebase.
email	varchar(150)	NO		YES	Электронный адрес пользователя, получается из объекта пользователя Google Firebase.
firstname	varchar(100)	NO			Имя
lastname	varchar(100)		NULL		Фамилия
middlename	varchar(100)		NULL		Отчество
image	varchar(255)		NULL		Имя файла изображения пользователя
region_id	integer	NO			Идентификатор региона из КЛАДР
city	varchar(100)	NO			Город
address	varchar(255)		NULL		Адрес
role_id	integer		0		Идентификатор роли, где 01 – заказчик 10 – мастер 11 – заказчик и мастер
experience	real		0		Стаж, лет
created_at	timestamp	NO	NOW		Дата создания, присваивается БД при создании
updated_at	timestamp	NO	NOW		Дата изменения, присваивается БД при изменении
deleted	boolean	NO	0		Флаг, удалена ли запись

2. Виды работ (specs)

Field	Type	Null	Default	Unique	Desc
id	integer	NO	autoincrement	YES	Идентификатор, присваивается БД при создании
name	varchar(50)	NO		YES	Название вида работ
deleted	boolean	NO	0		Флаг, удалена ли запись

3. Заказы (orders)

Field	Type	Null	Default	Unique	Desc

id	integer	NO	autoincrement	YES	Идентификатор, присваивается БД при создании
name	varchar(100)	NO			Название заказа
created_at	timestamp	NO	NOW		Дата создания заказа, присваивается БД при создании
updated_at	timestamp	NO	NOW		Дата изменения, присваивается БД при изменении
started_at	timestamp		NULL		Дата предполагаемого начала работ по заказу
real_started_at	timestamp		NULL		Дата фактического начала работ по заказу
ended_at	timestamp		NULL		Дата завершения работ по заказу
status_id	integer		0		Идентификатор статуса
caption	varchar(500)	NO			Описание
client_id	integer	NO			Идентификатор заказчика
guide_price	real		0		Ориентировочная цена, рублей
end_price	real		NULL		Конечная цена, рублей
deleted	boolean	NO	0		Флаг, удалена ли запись

4. Отзывы (reviews)

Field	Type	Null	Default	Unique	Desc
id	integer	NO	autoincrement	YES	Идентификатор, присваивается БД при создании
order_id	integer	NO			Идентификатор заказа, на который оставлен отзыв
rating	int	NO	NULL		Оценка работы, от 1 до 5.
comment	varchar(150)		NULL		Комментарий
created_at	timestamp		NOW		Дата создания отзыва, присваивается БД при создании
updated_at	timestamp	NO	NOW		Дата изменения, присваивается БД при изменении
deleted	boolean	NO	0		Флаг, удалена ли запись

5. Предложения (offers)

Field	Type	Null	Default	Unique	Desc
id	integer	NO	autoincrement	YES	Идентификатор, присваивается БД при создании
created_at	timestamp	NO	NOW		Дата создания заказа
updated_at	timestamp	NO	NOW		Дата изменения, присваивается БД при изменении
order_id	integer	NO			Идентификатор заказа, к которому подали предложение
master_id	integer		NULL		Идентификатор исполнителя

message	varchar(500)	NO			Сообщение предложения
selected	boolean	NO	false		Флаг выбора заказчиком отклика
deleted	boolean	NO	0		Флаг, удалена ли запись

6. Уведомления (notifications)

Field	Type	Null	Default	Unique	Desc
id	integer	NO	autoincrement	YES	Идентификатор, присваивается БД при создании
message	varchar(500)	NO			Текст уведомления
user_id	integer	NO			Идентификатор пользователя, к которому прикреплено уведомление
created_at	timestamp	NO	NOW		Дата получения уведомления
deleted	boolean	NO	0		Флаг, удалено ли уведомление

7. Связка видов работ и пользователей (works_users)

Field	Type	Null	Default	Unique	Desc
user_id	integer	NO			Идентификатор пользователя
spec_id	integer	NO			Идентификатор вида работ
created_at	timestamp	NO	NOW		Дата создания связки
updated_at	timestamp	NO	NOW		Дата изменения связки

8. Связка видов работ и заказов (works_orders)

Field	Type	Null	Default	Unique	Desc
order_id	integer	NO			Идентификатор заказа
spec_id	integer	NO			Идентификатор вида работ
created_at	timestamp	NO	NOW		Дата создания связки
updated_at	timestamp	NO	NOW		Дата изменения связки

9. Статусы заказа (statuses)

Field	Type	Null	Default	Unique	Desc
id	integer	NO	autoincrement	YES	Идентификатор статуса
name	varchar(50)	NO			Название статуса

10. Устройства, привязанные к пользователю (devices)

Field	Type	Null	Default	Unique	Desc

id	integer	NO	autoincrement	YES	Идентификатор устройства
user_id	varchar(100)	NO			Идентификатор пользователя
token	varchar(300)	NO			Токен устройства
created_at	timestamp	NO	NOW		Дата добавления устройства

V. СЕРВЕР API; ОПИСАНИЕ ВХОДНЫХ/ВЫХОДНЫХ ДАННЫХ

Регистрация пользователя

POST (/api/{version}/users/)

Вход:

{

“token_auth”: - Firebase ID token (string)

“token_FCM”: - FCM token (string)

“user”: - информация о пользователе. Содержит поля:

{

email - электронный адрес пользователя, номер телефона или Apple Id. Извлекается из объекта пользователя Google Firebase и зависит от типа авторизации(string)

firstname - имя пользователя (string)

lastname - фамилия пользователя (string)

middlename - отчество пользователя (string)

image - изображение пользователя, кодированное в base64

region_id - id региона (integer)

city – название города (string)

address - почтовый адрес пользователя (string)

role_id - идентификатор роли (integer)

experience - стаж работы, лет (real)

specs - **массив** id видов работ (array int)

}

}

Выход:

error_num - номер ошибки (integer)

error_message - текст ошибки (string)

Обновление данных пользователя

PUT (/api/{version}/users/)

Вход:

token_auth - Firebase ID token (string)

user - информация о пользователе. Содержит поля:

email - электронный адрес пользователя, номер телефона или Apple Id. Извлекается из объекта пользователя Google Firebase и зависит от типа авторизации(string)

firstname - имя пользователя (string)

lastname - фамилия пользователя (string)

middlename - отчество пользователя (string)

image - изображение пользователя, закодированное в base64

region_id - id региона (integer)

city – название города (string)

address - почтовый адрес пользователя (string)

role_id - идентификатор роли (integer)

experience - стаж работы, лет (real)

specs: **массив** id видов работ (array int)

Выход:

error_num - номер ошибки (integer)

error_message - текст ошибки (string)

Получение данных пользователя

GET (/api/{version}/users/my/)

Вход:

{“token_auth”: } - Firebase ID token (string)

Выход:

user - информация о пользователе. Содержит поля:

email - электронный адрес пользователя, номер телефона или Apple Id. Извлекается из объекта пользователя Google Firebase и зависит от типа авторизации(string)

firstname - имя пользователя (string)

lastname - фамилия пользователя (string)

middlename - отчество пользователя (string)

image - ссылка на изображение в профиле пользователя (string)

region_id - id региона (integer)

city – название города (string)

address - почтовый адрес пользователя (string)

role_id - идентификатор роли (integer)

experience - стаж работы, лет (real)

specs: **массив** видов работ (array spec)

notifications: **массив** уведомлений (array notification)

error_num - номер ошибки (integer)

error_message - текст ошибки (string)

Просмотр чужого профиля пользователя

GET (/api/{version}/users/other/)

Вход:

token_auth - Firebase ID token (string)

id - идентификатор пользователя (integer)

Выход:

user - информация о пользователе. Содержит поля:

image - ссылка на фотографию профиля (string)

firstname - имя (string)

lastname - фамилия (string)

middlename - отчество (string)

experience - стаж работы (real)

city - город проживания (string)

specs - **массив**, содержащий названия и id видов работ (array)

error_num - номер ошибки (integer)

error_message - текст ошибки (string)

Создание нового заказа

POST (/api/{version}/orders/)

Вход:

token_auth - Firebase ID token (string)

order - информация о заказе. Содержит в себе поля:

name - название заказа (string)

caption - описание заказа (string)

started_at - время предполагаемого начала работ (timestamp)

region_id - идентификатор города, где будет выполняться заказ (integer)

city - название города, где будет выполняться заказ (string)

address - адрес проведения работ (string)

specs - **массив**, содержащий названия и id видов работ (array)

guide_price - оценочная стоимость заказа (real)

Выход:

error_num - номер ошибки (integer)

error_message - текст ошибки (string)

Изменение заказа

PUT (/api/{version}/orders/)

Вход:

token_auth - Firebase ID token (string)

order - информация о заказе. Содержит в себе поля:

id - идентификатор заказа (integer)

name - название заказа (string)

caption - описание заказа (string)

region_id - идентификатор города, где будет выполняться заказ (integer)
city - название города, где будет выполняться заказ (string)
address - адрес проведения работ (string)
status_id - состояние заказа (integer)
guide_price - оценочная стоимость заказа (real)
started_at - время предполагаемого начала работ (timestamp)
specs - **массив**, содержащий названия и id видов работ (array)

Выход:

error_num - номер ошибки (integer)
error_message - текст ошибки (string)

Удаление заказа

PUT (/api/{version}/orders/delete/)

Вход:

token_auth - Firebase ID token (string)
order_id - идентификатор заказа (integer)

Выход:

error_num - номер ошибки (integer)
error_message - текст ошибки (string)

Отказ от работы(мастером, заказ в подтверждении)

PUT (/api/{version}/orders/giveup/)

Вход:

token_auth - Firebase ID token (string)
order_id - id отклика (string)

Выход:

error_num - номер ошибки (integer)
error_message - текст ошибки (string)

Отмена выбора мастера (только на этапе “подтверждение”)

PUT (/api/{version}/orders/cancel/)

Вход:

token_auth - Firebase ID token (string)
order_id - id заказа (string)

Выход:

error_num - номер ошибки (integer)
error_message - текст ошибки (string)

Начало работы по заказу (для мастера)

PUT (/api/{version}/orders/start/)

Вход:

token_auth - Firebase ID token (string)
order_id - id отклика (string)

Выход:

error_num - номер ошибки (integer)
error_message - текст ошибки (string)

Завершение работы по заказу

PUT (/api/{version}/orders/end/)

Вход:

token_auth - Firebase ID token (string)
order_id - id заказа (string)

Выход:

error_num - номер ошибки (integer)
error_message - текст ошибки (string)

Получение списка заказов пользователя (мои заказы)

GET (/api/{version}/orders/my/)

Вход:

token_auth - Firebase ID token (string)
filters - поля для фильтрации заказов. Структура:
city_id - id города (integer)
date - минимальная дата заказа (timestamp)
guide_price - минимальная стоимость заказа (real)
page - номер страницы с результатами запроса (необходимо для пагинации) (integer)
specs - **массив** id видов работ (array int)
sort_field - поле, по которому будет осуществлена сортировка (string)
sort_order - порядок сортировки, где 1 - по возрастанию, 0 - по убыванию (boolean)

Выход:

orders - массив заказов, где один заказ состоит из полей:
id - идентификатор заказа (integer)
role - флаг, определяющий на связь между заказом и пользователем, где 1 - пользователь выполняет этот заказ, а 0 - пользователь разместил этот заказ (boolean)
name - название заказа (string)
caption - описание заказа (string)
status_id - состояние заказа (integer)
guide_price - оценочная стоимость заказа (real)
started_at - время предполагаемого начала работ (timestamp)
specs - **массив**, содержащий названия и id видов работ (array)

Получение списка заказов другого пользователя

GET (/api/{version}/orders/history)

Вход:

token_auth - Firebase ID token (string)

filters - поля для фильтрации заказов. Структура:

city_id - id города (integer)

date - минимальная дата заказа (timestamp)

price - минимальная стоимость заказа (real)

page - номер страницы с результатами запроса (необходимо для пагинации) (integer)

specs - **массив** id видов работ (array int)

sort_field - поле, по которому будет осуществлена сортировка (string)

sort_order - порядок сортировки, где 1 - по возрастанию, 0 - по убыванию (boolean)

Выход:

orders - **массив** заказов, где один заказ состоит из полей:

id - идентификатор заказа (integer)

name - название заказа (string)

caption - описание заказа (string)

status_id - состояние заказа (integer)

guide_price - оценочная стоимость заказа (real)

started_at - время предполагаемого начала работ (timestamp)

specs - **массив**, содержащий названия и id видов работ (array)

error_num - номер ошибки (integer)

error_message - текст ошибки (string)

Просмотр деталей заказа

GET (/api/{version}/orders/)

Вход:

token_auth - Firebase ID token (string)

order_id - id заказа, данные которого нужно получить

Выход:

id - идентификатор заказа (integer)

name - название заказа (string)

caption - описание заказа (string)

address - адрес проведения работ (string)

status - состояние заказа (status)

guide_price - оценочная стоимость заказа (real)

started_at - время предполагаемого начала работ (timestamp)

client - информация о пользователе. Содержит поля:

id - идентификатор пользователя (string)

image - ссылка на фотографию профиля (string)

firstname - имя (string)
lastname - фамилия (string)
city - название города (string)
specs - **массив**, содержащий названия и id видов работ (array)
offers - **массив**, содержащий отклики мастеров. Каждый отклик имеет структуру:
id - идентификатор отклика (integer)
message - текст отклика мастера (string)
master_info - информация о мастере, который выполняет заказ:
image - ссылка на фотографию профиля (string)
firstname - имя (string)
lastname - фамилия (string)
city - название города (string)
rating - рейтинг мастера от 0 до 5 (real)
specs - **массив**, содержащий названия и id видов работ (array)
reviews - оценка работы мастера, которая состоит из полей:
rating - рейтинг мастера от 0 до 5 (real)
message - текст отзыва заказчика о мастере (string)

Получение списка найденных заказов (поиск)

GET (/api/{version}/orders/search/)
Вход:
token_auth - Firebase ID token (string)
filters - поля для фильтрации заказов. Структура:
city_id - id города (integer)
date - минимальная дата заказа (timestamp) -
price - минимальная стоимость заказа (real)
page - номер страницы с результатами запроса (необходимо для пагинации) (integer)
specs - **массив** id видов работ (array int)
sort_field - поле, по которому будет осуществлена сортировка (string)
sort_order - порядок сортировки, где 1 - по возрастанию, 0 - по убыванию (boolean)
Выход:
orders - **массив** заказов, где один заказ состоит из полей:
id - идентификатор заказа (integer)
name - название заказа (string)
caption - описание заказа (string)
status_id - состояние заказа (integer)
guide_price - оценочная стоимость заказа (real)
started_at - время предполагаемого начала работ (timestamp)
specs - **массив**, содержащий названия и id видов работ (array)

Количество завершенных заказов

GET (/api/{version}/orders/finished/)

Вход:

token_auth - Firebase ID token (string)

id - идентификатор пользователя (string)

Выход:

finished_orders - количество завершенных заказов (integer)

Получение списка видов работ

GET (/api/{version}/specs/)

Вход:

Выход:

specs - массив видов работ, где каждый элемент состоит из:

id - идентификатор вида работ

name - название вида работ

error_num - номер ошибки (integer)

error_message - текст ошибки (string)

Размещение отзыва о работе мастера

POST (/api/{version}/reviews/)

Вход:

token_auth - Firebase ID token (string)

order_id - id заказа (integer)

review:

message - текстовый отзыв о работе мастера (string)

rating - оценка работы мастера заказчиком от 0 до 5 (real)

Выход:

error_num - номер ошибки (integer)

error_message - текст ошибки (string)

Редактирование сообщения в отзыве о работе мастера

PUT (/api/{version}/reviews/)

Вход:

token_auth - Firebase ID token (string)

review - отзыв, который необходимо изменить:

id - id отзыва (integer)

message - измененный текстовый отзыв о работе мастера (string)

rating - null (изменению не подлежит)

Выход:

error_num - номер ошибки (integer)
error_message - текст ошибки (string)

Рейтинг мастера

GET (/api/{version}/reviews/rating/)
Вход:
token_auth - Firebase ID token (string)
id - идентификатор пользователя (string)
Выход:
rating - рейтинг мастера от 0 до 5 (real)

Оставить отклик на заказ

POST (/api/{version}/offers/)
Вход:
token_auth - Firebase ID token (string)
order_id - идентификатор заказа (integer)
message - текстовое предложение заказчику (string)
Выход:
error_num - номер ошибки (integer)
error_message - текст ошибки (string)

Назначить мастера на заказ

PUT (/api/{version}/offers/select/)
Вход:
token_auth - Firebase ID token (string)
offer_id - id отклика (string)
Выход:
error_num - номер ошибки (integer)
error_message - текст ошибки (string)

Отменить отклик на заказ

PUT (/api/{version}/offers/delete/)
Вход:
token_auth - Firebase ID token (string)
order_id - идентификатор заказа (integer)
Выход:
error_num - номер ошибки (integer)
error_message - текст ошибки (string)

Детальный просмотр отклика мастера

Исполнитель_____ Заказчик_____

GET (/api/{version}/offers/)

Вход:

token_auth - Firebase ID token (string)

offer_id - идентификатор отклика (integer)

Выход:

master - информация о мастере, который выполняет заказ:

id - идентификатор мастера (string)

image - ссылка на фотографию профиля (string)

firstname - имя (string)

lastname - фамилия (string)

middlename - отчество (string)

rating - рейтинг мастера от 0 до 5 (real)

message - текст отклика мастера (string)

Получение истории уведомлений

GET (/api/{version}/notifications/history/)

Вход:

token_auth - Firebase ID token (string)

Выход:

notifications - **массив** уведомлений, где каждый имеет поля:

message - текст уведомления (string)

date - дата создания уведомлений (timestamp)

error_num - номер ошибки (integer)

error_message - текст ошибки (string)

Оплата комиссии приложению мастером?

GET (/api/{version}/pay/)

Вход:

token_auth - Firebase ID token (string)

Выход:

link - ссылка на робокассу (string)

error_num - номер ошибки (integer)

error_message - текст ошибки (string)

Получение списка всех городов

Используется стороннее API vk.com

GET (https://api.vk.com/method/database.getCities?country_id=1&q={поисковый запрос}&need_all=0&count={количество городов в ответе}&v=5.103&access_token={сервисный код})

Вход:

Выход:

response - ответ, возвращаемый сервером vk. Содержит в себе:

items - массив городов. Каждый элемент содержит:

id - идентификатор города (integer)

name - название города (string)

area - район, где расположен город (string)

region - область, где расположен город (string)

error - сообщение об ошибке. Содержит в себе:

error_code - код ошибки (string)

error_msg - текст ошибки (string)

VI. ОПИСАНИЕ РОЛЕЙ ПОЛЬЗОВАТЕЛЯ, СТАТУСОВ ЗАКАЗА И ПРАВ ДОСТУПА.

Статусы

У каждого заказа, помимо его основной информации: названия, ориентировочной стоимости, даты исполнения, описания, есть специальный показатель – статус. В зависимости от статуса может изменяться доступный набор функций, отображаемые поля, права доступа к полям записей.

1. Поиск мастера. Статус присваивается заказу сразу после создания.

Заказы с данным статусом отображаются:

- На экране «Поиск» у мастеров и заказчиков (заказы с прочими статусами не доступны через поиск);
- На экране «Мои заказы» у заказчиков и мастеров, если мастер откликнулся на заказ;
- В истории заказов в профиле у мастера (который оставил отклик), при просмотре профиля другим пользователем;
- В истории заказов в профиле у заказчика, при просмотре профиля другим пользователем.

2. Подтверждение. Статус присваивается заказу после того, как заказчик выбрал исполнителя для своего заказа. При этом отклик мастера помечается, как выбранный. Данный заказ отображается:
   - У выбранного мастера на экране «Мои заказы»;
   - На экране «Мои заказы» у заказчиков;
   - В истории заказов в профиле у мастера (который оставил отклик), при просмотре профиля другим пользователем;
   - В истории заказов в профиле у заказчика, при просмотре профиля другим пользователем.
3. В работе. Данный статус присваивается заказу после того, как мастер подтвердил согласие выполнить заказ. При этом к заказу добавится дата фактического начала работ. Данный заказ отображается:
   - У мастера-исполнителя на экране «Мои заказы»;
   - На экране «Мои заказы» у заказчиков;
   - В истории заказов в профиле у мастера (который выполняет работу), при просмотре профиля другим пользователем;
   - В истории заказов в профиле у заказчика, при просмотре профиля другим пользователем.

4. Выполнено. Данный статус присваивается заказу после того, как мастер подтвердит завершение работ и проставит заказу финальную стоимость. При этом к заказу добавится дата завершения работ. Данный заказ отображается:
   - У мастера-исполнителя профиле;
   - На экране «Мои заказы» у заказчиков;
   - В истории заказов в профиле у мастера (который выполнил заказ), при просмотре профиля другим пользователем;
   - В истории заказов в профиле у заказчика, при просмотре профиля другим пользователем.
5. Оценка. Данный статус присваивается заказу после выполнения оплаты комиссии со стороны мастера. Данный заказ отображается:
   - У мастера-исполнителя профиле;
   - На экране «Мои заказы»;
   - В истории заказов в профиле у мастера (который выполнил заказ), при просмотре профиля другим пользователем;
   - В истории заказов в профиле у заказчика, при просмотре профиля другим пользователем.
6. В архиве. Данный статус присваивается заказу после того, как заказчик проставит оценку заказу. При этом к заказу до. Данный заказ отображается:
   - У мастера-исполнителя профиле;
   - У заказчика в профиле, в истории заказов;
   - В истории заказов в профиле у мастера (который оставил отклик), при просмотре профиля другим пользователем;
   - В истории заказов в профиле у заказчика, при просмотре профиля другим пользователем.

Действия пользователей в зависимости от их роли и статуса заказа

Роль/Статус	Поиск мастера	Подтверждение	В работе	Выполнено	Оценка	В архиве / Удалено
Заказчик	Удаление заказа	Звонок мастеру	Звонок мастеру	Звонок мастеру	Отзыв	Просмотреть профиль мастера
	Редактирование	Отмена выбранного мастера	Просмотреть профиль мастера	Просмотреть профиль мастера	Звонок мастеру	Просмотреть профиль заказчика
	Звонок мастеру	Просмотреть профиль мастера	Просмотреть профиль заказчика	Просмотреть профиль заказчика	Просмотреть профиль мастера	
	Выбор мастера	Просмотреть профиль заказчика			Просмотреть профиль заказчика	
	Просмотреть профиль мастера					
	Просмотреть профиль заказчика					

Исполнитель_____ Заказчик_____

Мастер	Отклик	Начало выполнения	Закончить выполнение	Оплата заказа	Звонок заказчику	Просмотреть профиль мастера
	Отмена отклика	Звонок заказчику	Звонок заказчику	Звонок заказчику	Просмотреть профиль мастера	Просмотреть профиль заказчика
	Просмотреть профиль заказчика	Отказ от заказа	Просмотреть профиль мастера	Просмотреть профиль мастера	Просмотреть профиль заказчика	
		Просмотреть профиль мастера	Просмотреть профиль заказчика	Просмотреть профиль заказчика		
		Просмотреть профиль заказчика				

Просмотр - просмотреть информацию по заказу может любой пользователь.
Редактирование - возможность изменять название, описание, город, адрес, дату начала работ, оценочную стоимость работ и виды работ у заказа.
Удаление заказа - возможность удалить заказ.
Звонок мастеру - возможность позвонить мастеру, вызов функционала телефона или, если он не доступен – отображение номера телефона мастера.
Звонок заказчику - возможность позвонить заказчику, вызов функционала телефона или, если он не доступен – отображение номера телефона заказчика.
Выбор мастера - выбор мастера, который будет выполнять заказ из списка соискателей.
Отказ от заказа - отказ от выполнения работы по заказу, удаление своего отклика из списка соискателей.
Просмотреть профиль мастера - возможность просмотреть подробную информацию о профиле мастера.
Просмотреть профиль заказчика - возможность просмотреть подробную информацию о профиле заказчика.
Начало выполнения - начать выполнять работу по заказу, ввод даты фактического начала работ.
Отклик - оставить предложение к заказу и добавить себя в список соискателей.
Отмена отклика - удаление своего отклика из списка соискателей.
Отмена выбранного мастера - отмена назначения мастера на заказа, возврат заказа в статус «Поиск мастера».
Закончить выполнение - завершение мастером работ по заказу.
Оплата заказа - возможность перевести на счет приложения определенную сумму по завершению выполнения работ, используя несколько способов оплаты.
Отзыв – возможность выставить заказчиком оценку выполненным работам и оставить краткий текстовый отзыв по их завершению.

Права доступа пользователей на чтение данных о заказе в зависимости от его статуса.

Наименования поля	Поиск мастера	Подтверждение	В работе	Выполнено	Оценка	В архиве / Удалено
Описание работы	+	+	+	+	+	+

Исполнитель_____ Заказчик_____

Информация о заказчике	+	+	+	+	+	+
Статус заказа	+	+	+	+	+	+
Виды работ	+	+	+	+	+	+
Номер заказа	+	+	+	+	+	+
Город	+	+	+	+	+	+
Адрес объекта	+	+	+	+	+	+
Оценочная стоимость	+	+	+			
Конечная стоимость				+	+	+
Мастер		+	+	+	+	+
Отзыв о мастере						+
Дата начала работ			+	+	+	+
Дата окончания работ				+	+	+
Откликнувшиеся мастера	+					

Права доступа пользователей на запись данных заказа в зависимости от его статуса и роли пользователя.

Наименования поля	Поиск мастера	Подтверждение	В работе	Выполнено	Оценка	В архиве / Удалено
Описание работы	Заказчик	Заказчик				
Дополнительная информация	Заказчик	Заказчик				
Виды работ	Заказчик	Заказчик				
Город	Заказчик	Заказчик				
Адрес	Заказчик	Заказчик				
Оценочная стоимость	Заказчик	Заказчик				
Конечная стоимость				Мастер		
Мастер	Заказчик	Заказчик				
Отзыв о мастере					Заказчик	
Дата начала работ		Мастер				
Дата окончания работ				Система		
Откликнувшиеся мастера	Мастер					

Периодические задачи

На сервере должны периодически выполняться следующие задачи:

- Если пользователь создал заказа и с момента создания заказа прошло уже 3 недели, а мастер так и не был назначен на выполнение заказа, то ему отправляется уведомления, что через неделю этот заказ будет удален.

Исполнитель_____ Заказчик_____

- Если пользователь создал заказа и с момента создания заказа прошло уже 4 недели, а мастер так и не был назначен на выполнение заказа, то ему отправляется уведомления, что заказ был удален и осуществляется процедура удаления заказа.
- Ежедневное выполнение бэкапа данных из базы.

Кроме того должна быть возможность запускать эти скрипты вручную по необходимости.

VII. ОПИСАНИЕ СТРУКТУРЫ И ИНТЕРФЕЙСА ПРИЛОЖЕНИЯ.

Экран загрузки

Экран загрузки отображается сразу же после запуска приложения. Во время отображения этого экрана выполняется проверка на наличие в системе авторизованного пользователя. (сохраненный объект firebaseUser). Если такой пользователь имеется, то приложение производит проверку его роли и осуществляет переход к соответствующему главному экрану. Если авторизированных пользователей не имеется, то приложение осуществляется переход к экрану авторизации.

Экран авторизации

Экран авторизации включает себя логотип приложения и, в зависимости от операционной системы устройства и ее версии, одну из следующих групп элементов для авторизации пользователя:

1) кнопку «Войти через Google» - для устройств на базе ОС Android
2) кнопку «Войти через Apple ID» - для устройств на базе iOS версии 13 и выше
3) поле для ввода номера телефона и кнопку «Войти» - для устройств на базе iOS ниже 13 версии

Авторизация проходит при помощи стороннего сервиса Google Firebase. Кроме того, на экране авторизации, в нижней части экрана находится ссылка на текст с политикой конфиденциальности (наличие такого текста, является обязательным условием для публикации приложения).
После нажатия на кнопку «Войти» происходит вызов стандартного, для соответствующей платформы, диалога с выбором аккаунта для авторизации. После выбора одного из представленных аккаунтов (если их несколько на устройстве) производится проверка на наличие пользователя в базе.
Если указанный пользователь в базе обнаружен, то приложение производит проверку его роли и осуществляет переход к соответствующему главному экрану.
Если указанный пользователь в базе не обнаружен, то приложение производит переход к экрану регистрации.

Экран регистрации и редактирования пользователя

Экран анкеты представляет из себя форму из нескольких полей:

- Фамилия (обязательное, строка, 128 символов)
- Имя (обязательное, строка, 128 символов)
- Отчество (обязательное, строка, 128 символов)
- Город (обязательное, выбор из справочника, выбор из городов и населенных пунктов России, используя стороннее api)

- Фото (опциональное, кнопка, открывающая галерею, ограничение 10Мб)
- Телефон (обязательное, паттерн числа и знаки, 32 символа) (Можно указать несколько значений, с возможностью выбрать по умолчанию, проверка на уникальность)
- Адрес (строка, 1024 символов)

Дополнительно на экране имеются два флага «Я- мастер..» и «Я- заказчик..». Отметив первый флаг пользователь получает для заполнения еще два поля:

- Стаж (обязательно, число с плавающей точкой, не больше 100 лет)
- Виды работ (перечисляемое поле, выбор из справочника, обязательно должен быть отмечен хотя бы один вид работ)

Кроме формы на экране имеется ссылка на пользовательское соглашение, чекбокс для принятия данного соглашения и кнопка «Продолжить». Без заполнения обязательных полей и отметки чекбокса – кнопка «Продолжить» не доступна.
Если пользователь впервые пользуется приложением, то ему показывается диалоговое окно с предложением просмотреть инструкции (Предоставляется заказчиком в виде PNG файлов) по размещению заказа и отклику на него.

Экран удаленного аккаунта

Экран появляется после успешной авторизации, если аккаунт пользователя был удален и содержит в себе фирменный логотип компании, сообщение об удалении аккаунта, кнопки связи с технической поддержкой и выходом из приложения

Главный экран

Данный экран доступен для зарегистрированных пользователей. Внизу экрана находится навигационное меню, позволяющее переключаться между четырьмя вкладками:

- Поиск
- Мои заказы
- Профиль
- Настройки

Вкладка «Поиск»

На вкладке «Поиск» отображается список заказов, соответствующих заданному фильтру. При запуске по умолчанию выставлен только фильтр по региону пользователя. Каждый заказ представлен отдельной карточкой на которой указана:

- Название заказа
- Дата начала работ (ориентировочная или фактическая)
- Оценочная или фактическая стоимость
- Краткое описание заказа (не более 4 строк)
- Виды работ (не более 6 в сокращенном варианте)
- Статус заказа.

Для более тонкой фильтрации в специальном окне можно задать дополнительные критерии для отсева не интересующих заказов.
Нажав на любую карточку заказа пользователь перейдет на экран просмотра подробной информации по заказу.

Экран фильтров

Экран необходим для фильтрации заказов по каким-либо признакам. Содержит в себе:

- Город
- Дату, начиная с которой, необходимо отображать заказы
- Виды работ
- Стоимость, от который необходимо показывать заказы
- Сортировка (поле, по которому нужно сортировать заказы)
- Порядок, в котором нужно сортировать заказы (по возрастанию или по убыванию)

Вкладка «Настройки»
Находясь на вкладке «Настройки» пользователь может:

- Включить/отключить возможность получения уведомлений от приложения.
- Перейти к экрану просмотра истории уведомлений
- Позвонить в службу поддержки
- Просмотреть инструкцию по размещению заказа
- Просмотреть инструкцию по отклику на заказ

Вкладка «Мои заказы»

На вкладке «Мои заказы» отображается список заказов, либо размещенных текущим пользователем (если пользователь заказчик) либо тех заказов, в которых пользователь является исполнителем (если пользователь мастер). Каждый заказ представлен отдельной карточкой на которой указана:

- Название заказа
- Дата начала работ (ориентировочная или фактическая)
- Оценочная или фактическая стоимость
- Краткое описание заказа (не более 4 строк)
- Виды работ (не более 6 в сокращенном варианте)
- Статус заказа.

Заказы можно фильтровать, задав критерии для фильтрации на специальном экране.
Нажав на любую карточку заказа пользователь перейдет на экран просмотра подробной информации по заказу.
Если пользователь является заказчиком, то из этой вкладки он может создавать и размещать свои заказы.

Экран «Новый заказ»
На вкладке «Новый заказ» находится форма, которую заполняет заказчик для создания нового заказа:

- Название заказа

- Описание работы (многострочное текстовое поле, 5120 символов*)*
- Город, в котором требуется провести работы (обязательное, выбор из справочника, выбор из городов и населенных пунктов)
- Адрес, по которому требуется провести работы (строка)
- Дата начала работ
- Оценочная стоимость
- Виды работ (перечисляемое поле, выбор из справочника)

Кроме формы на экране имеется кнопка «Разместить».

Экран «Редактирование заказа»

На экране редактирования заказа находится форма, которую заполняет заказчик для изменения информации о заказе:

- Название заказа
- Описание работы (многострочное текстовое поле, 5120 символов*)*
- Город, в котором требуется провести работы (обязательное, выбор из справочника, выбор из городов и населенных пунктов)
- Адрес, по которому требуется провести работы (строка)
- Дата начала работ
- Оценочная стоимость
- Виды работ (перечисляемое поле, выбор из справочника)

Кроме формы на экране имеется кнопка «Изменить».

Вкладка «Профиль»

На вкладке профиль пользователь может посмотреть информацию по своему профилю:

- Имя
- Фамилию
- Отчество
- Город
- Фото
- Телефон
- Адрес

Если пользователь является мастером, то в этом же окне отображаются

- Стаж
- Средний рейтинг мастера, вычисляемый из выставленных оценок к выполненным работам
- Виды работ, выполняемые мастером

Здесь же пользователь может осуществить редактирование своего профиля, а также выход из него, нажав на одну из отведенных для этого кнопок.
Осуществив выход, сохраненный в локальной памяти объект с данными пользователя будет выгружен, а пользователь будет снова перенаправлен на экран авторизации.

Исполнитель_____ Заказчик_____

Перейдя в режим редактирования профиля пользователю будет представлена форма из тех же полей, что и на экране регистрации.
Кроме формы на экране имеется кнопка «Сохранить».

Экран заказа
На экране подробностей о заказе указана следующая информация о заказе:

- Название заказа
- Состояние заказа
- Номер заказа
- Информация о заказчике (фото, имя, фамилия)
- Город
- Адрес выполнения заказа
- Виды работ
- Дата заказа
- Оценочная стоимость
- Описание заказа.

Если пользователь является заказчиком, а заказ находится в стадии поиска исполнителя, то в этом же окне отображаются предложения мастеров по выполнению заказов. Каждое из них – отдельная карточка, на которой отображаются контактные данные и рейтинг мастера. При нажатии на карточку открывается экран подробного просмотра предложения от мастера.

Если пользователь является заказчиком, а заказ находится в стадии выполнения работ, то в этом же окне будет отображена кнопка «Связаться с мастером» (по нажатию осуществляет переход в стандартное для платформы приложение «Телефон» с уже вбитым номером телефона в номеронабирателе)

Если пользователь является заказчиком, а заказ находится в стадии оценки, то в этом окне будет отображена кнопка «Оценить работу», вызывающая диалог для выставления оценки заказу.

Если пользователь является мастером, а заказ находится в стадии поиска исполнителя, то на экране отображается кнопка «Откликнуться», по нажатию на которую появляется небольшой диалог с предложением ввести дополнительное примечание к заявке и кнопкой отправить, по нажатию на которую отправляется заявка на выполнение заказа. Кнопка при этом становится неактивной или исчезает (дважды откликнуться нельзя).

Если пользователь является мастером, а заказ находится в стадии ожидания, то вместо кнопки «Откликнуться» здесь будет показана кнопка «Начать работу», по нажатию на которую будет осуществлен переход заказа в стадию «В работе».

Во время выполнения заказа на этом экране вместо кнопки «Начать работу» появляется кнопка «Завершить работу». После нажатия на нее мастеру предлагается в специальном окне выставить конечную стоимость и дату завершения заказа.

После того, как пользователь оставит отзыв к заказу этот отзыв также будет отображаться на этом экране.

Дополнительные функции, доступные из экрана заказа, зависящие от состояния, отображены в таблице “**Действия пользователей в зависимости от их роли и статуса заказа**” раздела VI текущего технического задания.

Окно «Просмотр профиля»

В этом окне пользователь просматривает подробную информацию о другом пользователе, являясь гостем. Окно представляет из себя:

1) Карточку с основной информацией о пользователе, включающую в себя: имя, фамилию, отчество, фото и рейтинг (если просматривается профиль мастера)
2) Информационную панель из двух вкладок:
    - “О пользователе”, где размещена информация о городе и количестве завершенных заказов. Если просматривается профиль мастера, то добавляются еще поля со стажем и видами проводимых работ.
    - “История заказов”, где размещены карточки с заказами, которые пользователь размещал, выполнял или выполняет в текущий момент. Нажав на любую карточку, пользователь перейдет на подробную информацию о заказе.

Прочее

Помимо перечисленных экранов, интерфейс включает несколько диалоговых окон и экранов. Они, как правило не содержат в себе данных, но необходимы для выбора чего-либо из списка, уточнения информации и подтверждения действий.
Среди них:

1. Экран выбора города (используется при регистрации пользователя и при задании фильтра заказов)
2. Экран выбора видов работ (используется при регистрации пользователя и при задании фильтра заказов)
3. Экран выбора даты (используется задании различного рода дат, в том числе для фильтрации)
4. Экран истории уведомлений (показывает список уведомлений)
5. Экран подробностей об отклике (отображает краткую информацию о мастере и показывает его предложение для заказа)
6. Диалоговое окно завершения заказа. В нем мастер проставляет итоговую стоимость заказа и тем самым подтверждает завершение работ и изменение статуса заказа на «Завершено»

Уведомления

Каждый раз при смене статуса заказа пользователя отправляются push-уведомления.

1. Ожидание подтверждения. Данный статус присваивается заказу после того, как заказчик выбрал исполнителя для своего заказа. При этом исполнителю отправляется push-уведомление «На ваш заказ "<Название заказа и номер>" откликнулся мастер»
2. Подтверждение. Данный статус присваивается заказу после того, как заказчик выбрал мастера для исполнения заказа. Мастеру при этом приходит уведомление «В заказе "<Название заказа и номер>" вы выбраны мастером, подтвердите выполнение работ»
3. В работе. Данный статус присваивается заказу после того, как мастер подтвердил согласие выполнить заказ. При этом заказчику отправляется push-уведомление «Ваш заказ "<Название заказа и номер>" принял в работу <Имя мастера>»

4. Оценка. Данный статус присваивается заказу после того, как мастер подтвердит завершение работ, проставит финальную стоимость и оплатит пошлину владельцу приложения. При этом заказчику отправляется push-уведомление «Оцените работу мастера по заказу "<Название заказа и номер>"»
5. В архиве. Данный статус присваивается заказу после того, как заказчик проставит оценку заказу. При этом исполнителю отправляется push-уведомление «Вы получили оценку по заказу "<Название заказа и номер>"»

Все уведомления после получения сохраняются в специально отведенной для этого таблице в БД. В дальнейшем к этому списку можно получить доступ внутри приложения. При клике на уведомлении внутри приложения должен осуществляться переход к окну заказа, к которому оно принадлежит.

Оплата

В приложении предусмотрена возможность оплаты мастером комиссии владельцу приложения после получения оплаты за выполненные работы. Пока мастер не выполнит оплату комиссии – заказчик не может выставить оценку его работе, а следовательно мастер не получит повышение рейтинга и не сможет взять новые заказы, существует ограничение на мастера в 3 заказа «В работе».

Оплата будет осуществляться с помощью подключения стороннего сервиса (например ROBOKASSA), который должен поддерживать несколько способов оплаты: Карта, Яндекс деньги, WebMoney, Kiwi и т.п.

Процедура оплаты осуществляется следующим образом:

1) После нажатия кнопки «Оплатить» в заказе, мастер переходит на специальный экран. Экран представляется собой компонент WebView в котором открывается страница платежного сервиса. На страницу передаются параметры, необходимые для осуществляется платежа.
2) Пользователь выбирает удобный ему способ оплаты и вписывает свои реквизиты, после чего подтверждает осуществление платежа
3) В случае успеха завершения процедуры оплаты комиссии, заказу присваивается статус «Оценка» и заказчику становится доступна функция оценки заказа