

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное
образовательное учреждение высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

С. В. Рындина

**РАБОТА НАД ПРОЕКТОМ:
СОЗДАНИЕ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ**

Учебно-методическое пособие

ПЕНЗА 2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Пензенский государственный университет» (ПГУ)

С. В. Рындина

Работа над проектом:
создание мобильных приложений

Учебно-методическое пособие

Пенза
Издательство ПГУ
2023

УДК 004.738.5(075)+339(075)

P95

Р е ц е н з е н т

кандидат технических наук, доцент

А. А. Масленников

Рындина, Светлана Валентиновна.

P95

Работа над проектом: создание мобильных приложений : учеб.-метод. пособие / С. В. Рындина. – Пенза : Изд-во ПГУ, 2023. – 60 с.

Рассмотрена организация проектной работы со старшеклассниками на примере создания мобильных приложений без использования программного кода.

Издание подготовлено на кафедре «Цифровая экономика» для «Института непрерывного образования ПГУ», может использоваться как преподавателями инженерной школы ПГУ, так и самими обучающимися при выполнении проекта.

УДК 004.738.5(075)+339(075)

© Пензенский государственный
университет, 2023

Содержание

1. Планирование проектной деятельности со школьниками	4
2. Формирование проектной команды, бизнес-идея, целевая аудитория	9
3. Анализ предметной области	13
4. Разработка сценариев взаимодействия с приложением: регистрация и авторизация.....	14
5. Разработка сценариев взаимодействия с приложением: описание основных интерфейсных групп и дизайн базы данных	24
6. Интерфейсная группа: Рецепты.....	30
7. Интерфейсная группа: Дневник калорий	38
8. Интерфейсные группы: Профиль пользователя, Связь с поддержкой	40
9. Дополнительные материалы: создание чата	46
10. Проект «Разработка мобильного приложения»	58
Список литературы.....	59

1. Планирование проектной деятельности со школьниками

В рамках профориентационной работы Институт непрерывного образования ПГУ реализует проект «Политехническая школа ПГУ» (инженерные классы) для обучающихся 10 и 11 классов образовательных учреждений города Пензы и области. Двухгодичная программа включает работу над проектом под руководством одного из преподавателей.

Автор пособия со своей проектной командой обучающихся 11 классов сопровождает проект разработки мобильного приложения с помощью конструктора Adalo.

Проектная деятельность – это деятельность, максимально приближенная к будущей профессиональной деятельности в рамках выполнения трудовых функций квалифицированным сотрудником с технологическими компетенциями.

Инженерные классы, организованные ПГУ для старшеклассников, также позволяют заранее примерить на себя должностные функции будущей профессии, поработать в проектных группах не только со своими одноклассниками, но и неформально пообщаться со сверстниками из других образовательных учреждений, протестировать свою готовность сотрудничать для достижения общей цели.

Проектная деятельность развивает сразу две группы компетенций:

- soft skills («мягкие» навыки) это формирование компетенций, связанных с выстраиванием коммуникаций, в том числе и цифровых, формирование умений работать в команде, умений и навыков управления временем, навыков проведения презентаций, навыков продаж своего проекта аудитории и своих компетенций проектной команде, поддержка стремления к лидерству, личному развитию, самомотивации;

- hard skills («жесткие» навыки) – это профессиональные знания, умения, инструменты и компетенции, которые связаны с технической стороной деятельности в проекте.

Проект – это «временное предприятие, направленное на создание уникального продукта, услуги или результата» [1].

Базовые составляющие любого проекта – это его цель (планируемый результат), начало, завершение, сроки исполнения (продолжительность), необходимые ресурсы (исполнители, программное обеспечение, оборудование и т.п.).

Проектная деятельность всегда дает не только основной результат в виде конечного продукта, услуги, товара и т.п., но и дополнительные

эффекты в виде знаний, опыта, новых возможностей, полезных контактов и т.п.

Перед преподавателем, курирующим проект, стоит иная цель, чем перед проектной командой. Проектная команда стремится достичь цели своего проекта, получить проектный результат. Преподавателем в проектной работе ставятся обучающие цели и задачи: расширение кругозора участников проекта в области технологических решений, формирование необходимых в будущей профессиональной карьере навыков коммуникации, поиска компромисса, разделения обязанностей, принятия ответственности, а также умений использовать различные инструменты и программное обеспечение. Главной же целью проектной работы в инженерной школе является формирование более осознанного подхода к выбору будущей профессии со стороны обучающихся.

Проектная работа основана на управлении проектом, состоящим:

- из действий по согласованию ролей участников команды, распределенных на основании имеющихся навыков и знаний;
- процессов, которые участники реализуют;
- инструментов и методов, которые участники используют для достижения поставленной проектной цели и получения планируемого результата.

Под влиянием конкурентной рыночной среды, процессов цифровизации и цифровой трансформации, основанных на сквозных технологиях, автоматизации и т.п., навыки проектной работы и проектного управления становятся все более востребованными бизнесом. Главное преимущество проектов – в гибкой подстройке под изменения конъюнктуры рынка, экономической ситуации в целом.

При управлении проектом куратору необходимо учесть в тематическом планировании все этапы работы над проектом.

Первый этап управления проектами – инициация. Проект рассматривается в широком контексте: какова бизнес-идея проекта, есть ли у проекта перспектива, в том числе возможность реализации, какова потенциальная целевая аудитория, ценен ли для этой аудитории результат проекта. Принимается решение, стоит ли бизнес-концепт проекта требуемых для его реализации усилий и ресурсов.

На втором этапе происходит формирование плана проекта. Бизнес-концепт приобретает более глубокое содержание, происходит детализация и формализация сформулированных на предыдущем этапе идей. Определяются цели и задачи проекта, предметная область проекта, контрольные точки в процессе реализации, роли участников команды и зоны

их ответственности. Составляется дорожная карта проекта с промежуточными результатами, которые должны быть достигнуты. В учебном проекте обычно речь не идет о реальном бюджете, однако основы бюджетирования и управления проектными рисками применимы и в этом случае. Можно рассчитать трудозатраты участников проекта, определить ресурсы: оборудование, программное обеспечение и т.п., составить карту рисков проекта и проранжировать их.

Третий этап – собственно выполнение проекта, который реализуется совместно с четвертым – мониторингом хода реализации проекта. Все участники согласно дорожной карте проекта выполняют поставленные задачи. На регулярных встречах фиксируется процесс выполнения, отслеживаются возникающие сложности, перераспределяются ресурсы. При необходимости происходит возврат на предыдущий этап, вносятся изменения в план проекта.

На завершающем этапе готовится итоговый отчет по проекту, проводится оценка проекта (рефлексия – что получилось, что можно было сделать иначе, какие навыки и знания были приобретены в ходе работы над проектом). Вся документация по проекту сохраняется на будущее, чтобы при необходимости можно было воспользоваться наработками в новых проектах.

В проекте необходимо определить роль менеджера продукта/проекта. Его обязанности шире, чем обязанности участника проектной команды: он распределяет задания внутри команды, отслеживает прогресс, инициирует дополнительные встречи с участниками команды.

Куратор проекта – преподаватель, взаимодействует в рамках регулярных встреч со всей командой проекта, направляет, советует, консультирует, формирует необходимые для реализации проекта знания и навыки, получает на регулярной основе отчет о ходе реализации проекта от менеджера проекта. Также куратор отслеживает вовлеченность всей команды в выполнение проекта, справедливое распределение заданий между участниками, помогает решать коммуникационные проблемы менеджеру проекта.

В качестве начальной постановки бизнес-концепта для проектной работы был предложен проект разработки мобильного приложения для фитнес-индустрии.

Для проекта «Разработка прототипа фитнес-приложения» примерное тематическое планирование занятий приведено в табл. 1.

Тематическое планирование

Примерная дата	Тема занятия	Вид занятия	Результат
Сентябрь	Бизнес-идея проекта. Обсуждение возможных вариантов, выбор предметной области	Круглый стол	Слайды в презентацию: название проекта, роли участников, предметная область, целевая аудитория
Сентябрь	Анализ предметной области. Цифровые продукты: создание мобильных приложений	Лекция	Слайды в презентацию: анализ предметной области
Октябрь	Работа в конструкторе приложений: создание экранов регистрации и авторизации	Практико-ориентированное занятие	Разработка приложения
Октябрь	Разработка сценариев взаимодействия с приложением. Дизайн базы данных	Лекция	Слайды в презентацию: основной функционал приложения и дизайн базы данных
Ноябрь	Работа в конструкторе приложений: создание экранов со списками. Интерфейсная группа: рецепты	Практико-ориентированное занятие	Разработка приложения
Ноябрь	Работа в конструкторе приложений: создание экранов с формами. Интерфейсная группа: дневник калорий	Практико-ориентированное занятие	Разработка приложения
Декабрь	Работа в конструкторе приложений: навигация, добавление переключателей, app bar и tab bar	Практико-ориентированное занятие	Разработка приложения
Декабрь	Работа в конструкторе приложений. Интерфейсная группа: запросы в поддержку	Практико-ориентированное занятие	Разработка приложения
Январь	Работа в конструкторе приложений: настройка чата	Практико-ориентированное занятие	Разработка приложения

Окончание табл. 1

Примерная дата	Тема занятия	Вид занятия	Результат
Январь	Подготовка питча и итоговой презентации	Круглый стол	Итоговая презентация и речь
Февраль	Защита проекта	Итоговая аттестация	Представление проекта

Тематический план в любом случае будет претерпевать изменения, так как в данной теме возможна гибкая подстройка под интересы участников проекта. Куратор проекта предлагает как минимум одно направление и несколько посильных ответвлений.

2. Формирование проектной команды, бизнес-идея, целевая аудитория

Информация для куратора

На первой встрече необходимо заинтересовать участников проектной команды темой проекта. Обозначить, что предложенная куратором тема может быть скорректирована. Тема проекта обсуждается и согласовывается с членами проектной команды. Обсуждение не может быть формальным: оно должно содержать аргументацию, обосновывающую правильность выбора.

Также необходимо поставить перед участниками задачу распределения ролей в проектной команде. Так как результат проектной работы представляется на защите, то, кроме самого приложения, необходимо создать и презентацию, а также подготовить финальную речь.

Необходимо, чтобы участники проекта распределили роли и обязанности: кто будет менеджером проекта, отвечающим за его следование плану, своевременность выполнения работ по проекту и т.п., кто будет аналитиком проекта, выполняющим поиск и структурирование информации, отвечающим за наполнение презентации содержанием, кто будет дизайнером проекта, отвечающим за фирменный стиль приложения, подбор необходимого визуального сопровождения и его легальное использование, кто будет инженером данных, отвечающим за дизайн базы данных.

Важно, чтобы каждый участник проектной команды выбрал себе обязанности по душе, в которых он будет первым среди равных. Не только аналитик анализирует предметную область, а все участники проектной команды вносят свой вклад в анализ, но этим процессом руководит аналитик.

Это позволит каждому принять на себя руководящую роль в определенных процессах проекта и разделить ответственность за конечный результат проекта.

Оптимальная команда для реализации проекта по созданию приложения – 3–5 участников.

Информация для участников проекта

Цифровой продукт начинается с идеи, которая может быть верифицирована рынком еще на начальном этапе до привлечения серьезных инвестиций для ее реализации. Уверенность в том, что цифровой продукт, основанный на предложенной идее, будет финансово успешен и жизнеспособен, желательно подкрепить исследованиями рынка.

Одним из способов, позволяющим проверить отзывчивость рынка по отношению к предлагаемому продукту, является концепция MVP (minimum viable product, минимально жизнеспособный продукт).

Концепция описывает цель создания продукта: для какой целевой группы какую проблему будет решать продукт.

Шаг от идеи к продукту (даже в версии MVP) можно сделать на основе некоторой концепции, которая позволяет разработать структуру будущего продукта.

На рис. 2.1 представлен пример описания портрета пользователя (lifestyle – это образ жизни, под тематику подходит слишком многое: от советов по организации питания и описания спортивных тренировок до создания модных образов).

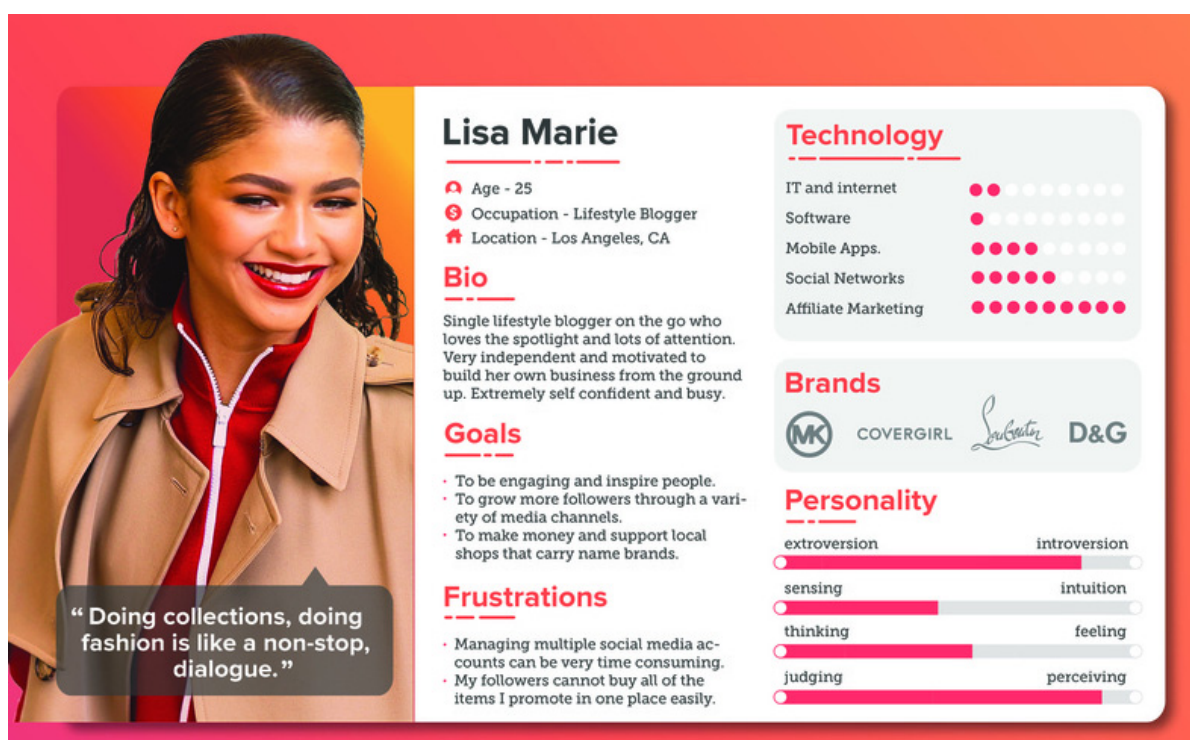


Рис. 2.1. Пример user person [2]

Самостоятельное написание кода для приложения даже при наличии базовых знаний – достаточно долгий процесс. Вносить изменения в созданный продукт, что называется «на лету», тоже не получится. А потратить деньги на разработку приложения сразу в коде могут позволить себе далеко не все компании. В некоторых случаях это даже нецелесообразно, так как нет твердых данных о платежеспособном спросе на продукт, о его ценности для пользователей.

No-code – это способ создания приложений с помощью специальных платформ, использующих методы drag and drop и визуального

моделирования вместо написания кода. При этом код все равно существует, он связан с визуальными элементами, с которыми взаимодействует разработчик в процессе создания приложения. Также подобными платформами может поддерживаться расширение функциональности создаваемых приложений с помощью программного кода.

Платформы для создания бизнес-приложений без кодирования или с его минимальным использованием – это хороший вариант, чтобы создать первую версию продукта и, что называется «в бою», определиться с целевой аудиторией продукта, сформировать функционал на основе обратной связи от пользователей, а также протестировать, есть ли деньги в этой нише, готова ли аудитория платить за продукт, есть ли потенциал роста.

Работа над бизнес-приложением включает следующие этапы:

- Формулирование гипотез для продукта и моделирование бизнес-процессов, которые будет обслуживать бизнес-приложение.
- Создание прототипа и проведение пользовательских исследований, глубинных интервью.
- Публикация продукта и получение обратной связи (feedback): уточнение целевой аудитории продукта и получение данных продаж (загрузок/установок/предзаказов).

Чтобы освоить подход к созданию приложений без использования кода (no-code), нужно понимать, как функционируют цифровые решения: бизнес-логика приложения, какие процессы пользователя это цифровое решение сопровождает и обслуживает, какие интерфейсы для организации взаимодействия должны быть реализованы.

Есть различные решения для доставки мобильных приложений пользователю.

PWA (Progressive Web Application) – технология, которая позволяет клиентам установить сайт на смартфон как приложение, она имитирует работу нативного приложения. Для реализации этого подхода есть специальные инструменты: Service Worker, Push Notifications, Application Shell и др., которые позволяют приложению, например, работать оффлайн при необходимости. Это приложения, которые устанавливаются на рабочий стол смартфона при помощи QR-кода или ссылки. Из-за комиссии и повышенных требований, которые выставляют Apple App Store и Google Play Store, многие переключаются на PWA.

Есть технология Web View, которая позволяет оборачивать приложения PWA тонким слоем нативного кода, использующего SDK (software development kit, комплект для разработки программного обеспечения) браузера для открытия веб-приложения. Такие приложения можно распространять через магазины приложений: Apple App Store и Google Play Store.

Нативное приложение создается с использованием инструментов разработки вендоров: Apple/Google и распространяется через магазины приложений: Apple App Store и Google Play Store.

Adalo – конструктор мобильных приложений. Компоненты: поля, кнопки, списки и изображения размещаются на холсте, т.е. на экране приложения, для них настраиваются действия, которые могут связывать элементы с коллекциями данных. Есть возможность собирать данные и отображать их по запросу.

Платформы no-code и low-code обладают определенными ограничениями, и важно понять, являются ли они препятствием для успешности задуманного приложения:

- ограничения по кастомизации дизайна: какими могут быть элементы экрана; например, можно ли сделать отдельные элементы фотографии кликабельными (в приложении для подбора образа – элементы одежды на фото, в приложении для подбора макияжа – части лица и т.п.);
- проект находится под управлением выбранной платформы: если что-то случится с платформой или платформа обнаружила нарушение каких-либо правил (платформы, законодательных норм и т.п.) – проект может быть потерян (отключен);
- безопасность и хранение данных: если происходит интеграция со сторонними сервисами, то могут возникнуть проблемы с доступом к данным, надежностью передачи данных, поддержанием их структуры, так как исходный код недоступен, то сложно проверить, насколько система защищена от возможных утечек конфиденциальной информации.

3. Анализ предметной области

Информация для участников проекта

Обзор конкурентов позволяет определить, какими конкурентными преимуществами должен обладать разрабатываемый продукт: какой функционал для подобного продукта является базовым, какие дополнительные возможности есть у конкурентов, возможно ли в проектируемом продукте реализовать те же функции.

Это возможность вырасти на чужих идеях, определить для разрабатываемого продукта, в чем будет его конкурентное преимущество: функциональность, коммуникации с иными целевыми сегментами, ценовое предложение и т.п.

Анализируя конкурентов и цифровые решения в выбранной индустрии, можно определить пользовательские истории и бизнес-требования.

Пользовательская точка зрения и пользовательские сценарии преобразуются в UX-дизайн (User eXperience, пользовательский опыт) продукта.

Все требования в продукте подразделяются на две основные категории: потребности (пользователей) и решения (функции), которые будут реализованы в продукте для удовлетворения потребностей.

Определившись с первоначальным набором функций приложения и его целевой аудиторией, можно приступать к разработке, помня, что создание приложения – итерационный, т.е. поэтапный, процесс. При этом иногда для более качественного результата мы не только движемся вперед, но и возвращаемся на предыдущие этапы, чтобы с новыми знаниями и новым видением критично посмотреть на первоначальные планы и реализации и скорректировать их.

4. Разработка сценариев взаимодействия с приложением: регистрация и авторизация

Adalo (<https://www.adalo.com/>) – платформа для создания мобильных приложений без использования программного кода.

Преимущества Adalo – наличие бесплатного плана использования (Explore Plan), который включает в себя:

- создание неограниченного количества приложений;
- возможность публикации в домене Adalo.

Регистрация в Adalo – это переход по кнопке Get Started (рис. 4.1). На аккаунт, который используется при регистрации, придет письмо для верификации адреса (рис. 4.2).

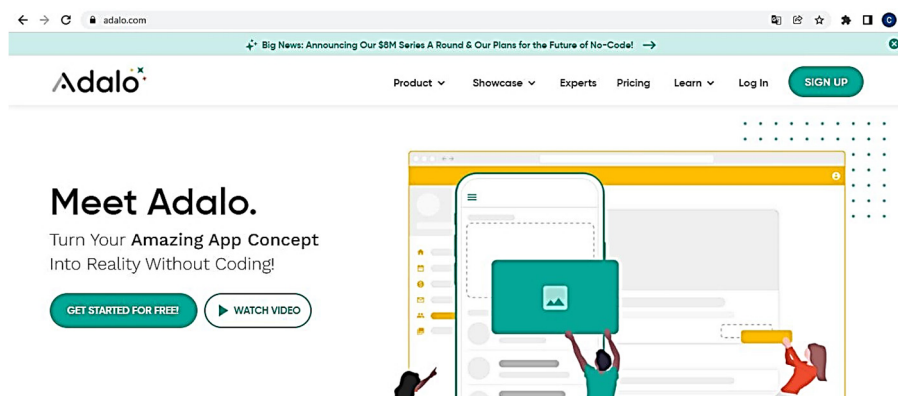


Рис. 4.1. Регистрация в сервисе Adalo

The image shows a 'Sign Up For Free!' form on the Adalo website. At the top of the form is the Adalo logo. The form contains three input fields: 'Email Address' with the value 'svetlanar004@gmail.com', 'Password' with masked characters '*****', and 'Full Name' with the value 'Svetlana Ryndina'. Below these fields is a checkbox labeled 'I have read & agree to Adalo's Terms of Use', which is checked. At the bottom of the form is a large yellow button labeled 'LET'S DO THIS!'.

Рис. 4.2. Авторизация в сервисе Adalo

После входа в Adalo предлагается создать приложение. Выбирается тип приложения: мобильное приложение или настольное веб-приложение (рис. 4.3); выбирается шаблон или пустой экран (рис. 4.4); выбирается цветовая схема для приложения (которую можно изменить в любое время, и компоненты, использующие предустановленные цвета, будут автоматически обновлены) и вводится название приложения (рис. 4.5).

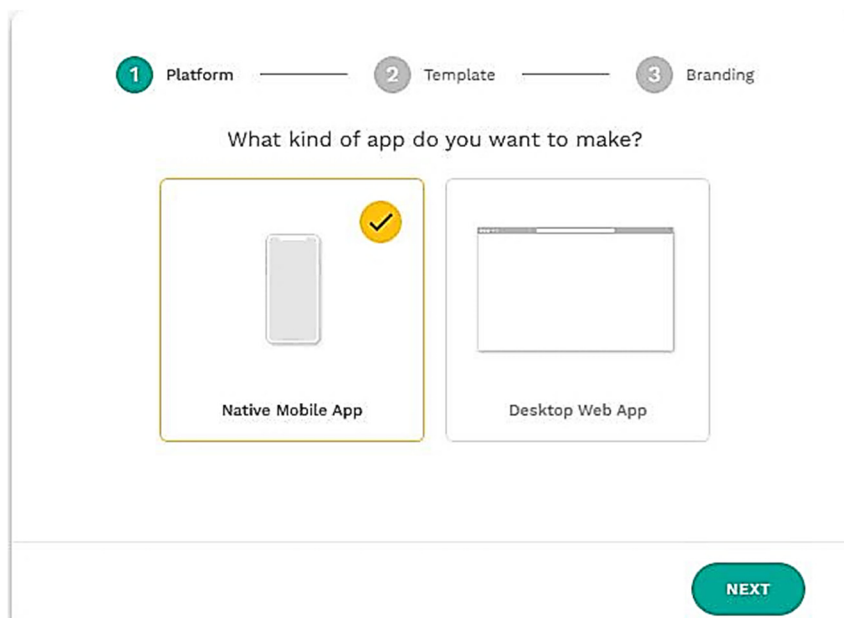


Рис. 4.3. Выбор типа приложения

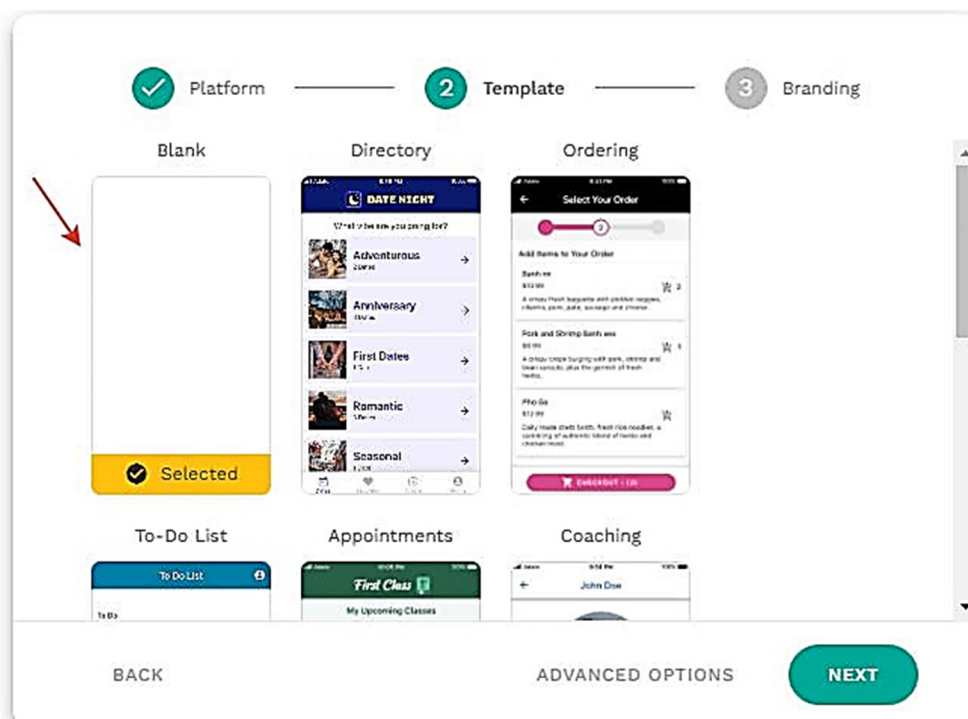


Рис. 4.4. Выбор шаблона для приложения

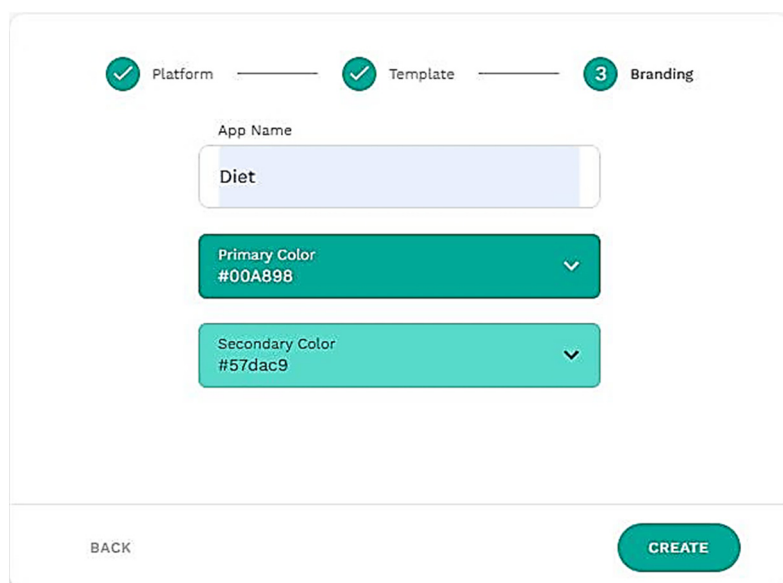


Рис. 4.5. Выбор цветовой схемы приложения и ввод его названия

После базовых настроек для нового приложения Adalo открывает холст (canvas) – рабочую область, в которой проектируются экраны приложения (рис. 4.6). По холсту можно перемещаться с помощью полосы прокрутки (вверх и вниз) или стрелок на клавиатуре (влево и вправо). Для изменения масштаба при нажатой клавише <Ctrl> используем вращение колесика мыши или нажатие клавиш <+>/<->.

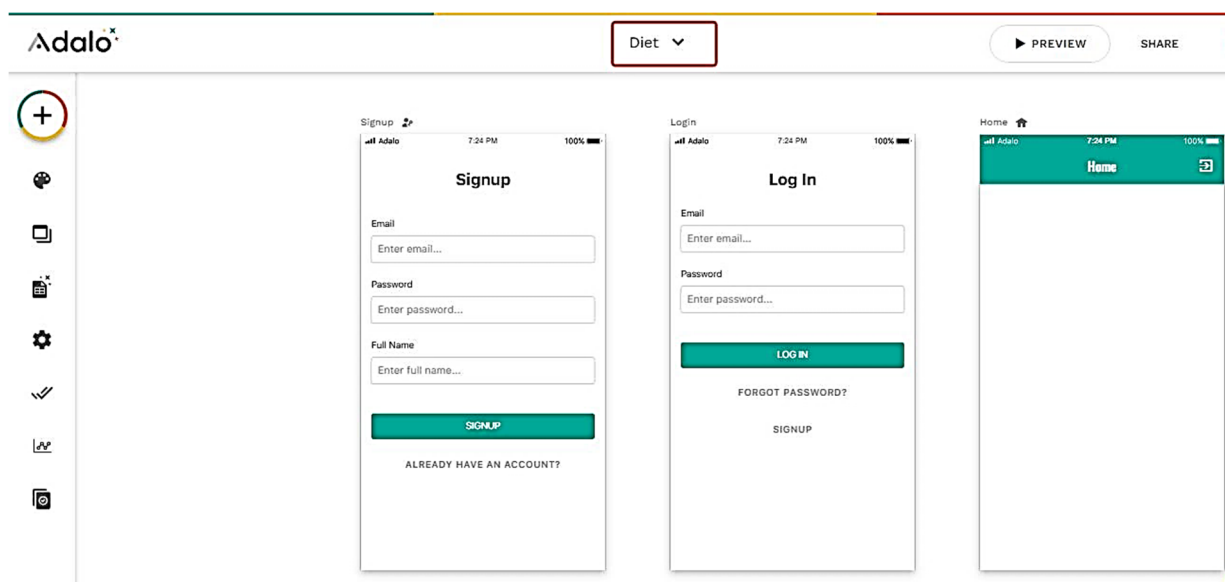


Рис. 4.6. Рабочая область для редактирования приложения

На холст можно добавлять новые экраны, на экраны можно добавлять компоненты (текст, кнопки и т.п.). Изначально приложение было создано с тремя дефолтными экранами (рис. 4.7): экран регистрации

(Signup) – это в текущий момент экран особого типа Welcome screen, т.е. с него начинается взаимодействие с приложением; главный экран приложения с особым статусом Home screen; экран авторизации Login, который является обычным экраном (Normal screen).

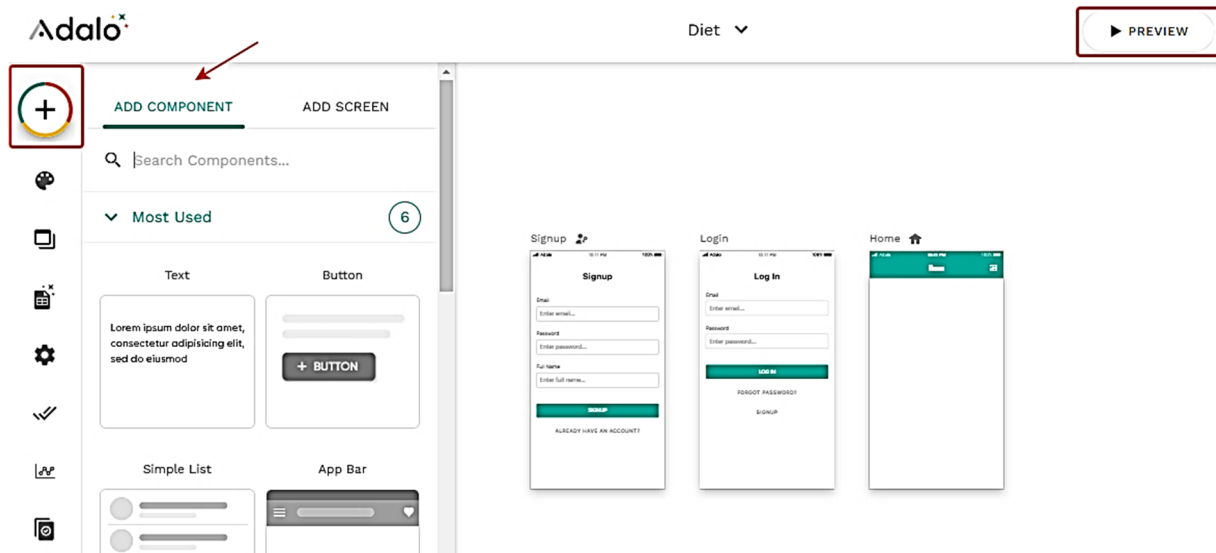


Рис. 4.7. Добавление компонентов на экран

Протестируем работу первых экранов приложения. Нажмем кнопку Preview и заполним форму регистрации (рис. 4.8).

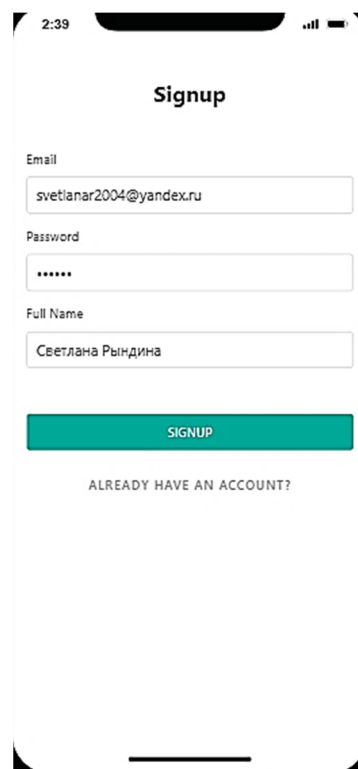


Рис. 4.8. Тестирование регистрации

После щелчка по кнопке регистрации данные формы сохранятся в базе данных приложения (Database Collections) и осуществится переход на главный экран приложения. В правом верхнем углу главного экрана Home приложения есть кнопка выхода из аккаунта, воспользуемся ей.

После выхода из аккаунта приложение возвращает нас на стартовый экран с регистрацией. Так как у нас уже есть аккаунт, то перейдем на экран авторизации с помощью ссылки ALREADY HAVE AN ACCOUNT. Пройдем авторизацию, введя пару email/password, и окажемся опять на главной странице приложения.

Рассмотрим, какие компоненты реализуют этот функционал. Выделим форму на экране Signup и перейдем в ее настройки. Как видим, форма связана с таблицей данных Users. То есть вначале создается таблица данных с атрибутами (полями), а только потом форма. С помощью формы этого экрана в таблицу Users добавляются данные новой регистрации (рис. 4.9).

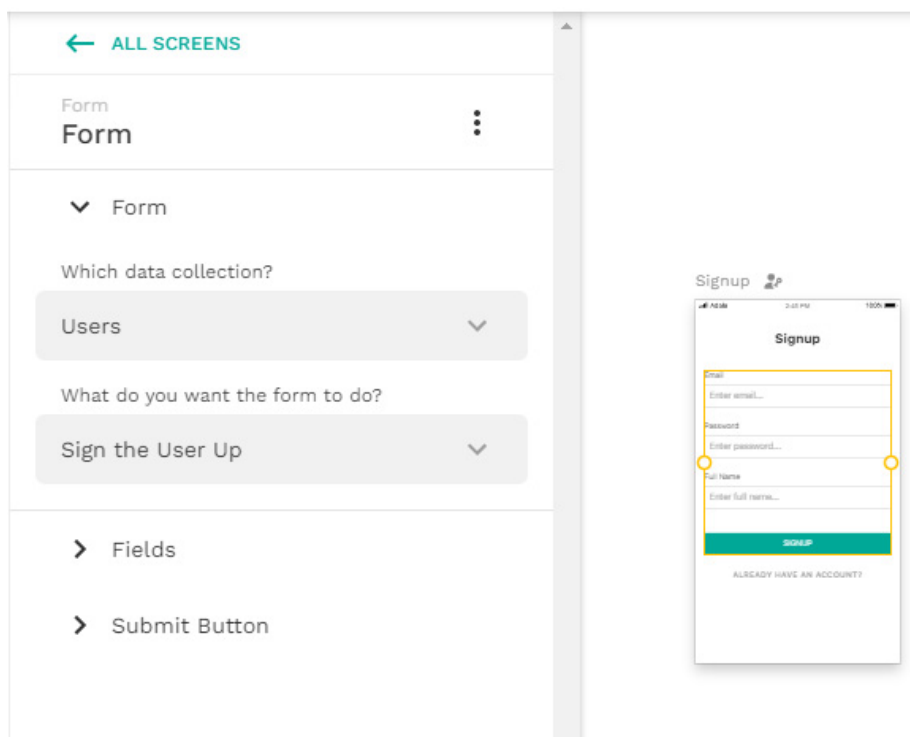


Рис. 4.9. Форма регистрации (настройки)

В форме регистрации отображаются три поля, которые есть в таблице Users (рис. 4.10). Количество полей, их состав и порядок можно менять. Однако включить в форму поле, которого нет в таблице Users, не получится.

Также на форме есть кнопка Signup, для которой настроены два действия: добавление записи в таблицу Users и переход на экран Home.

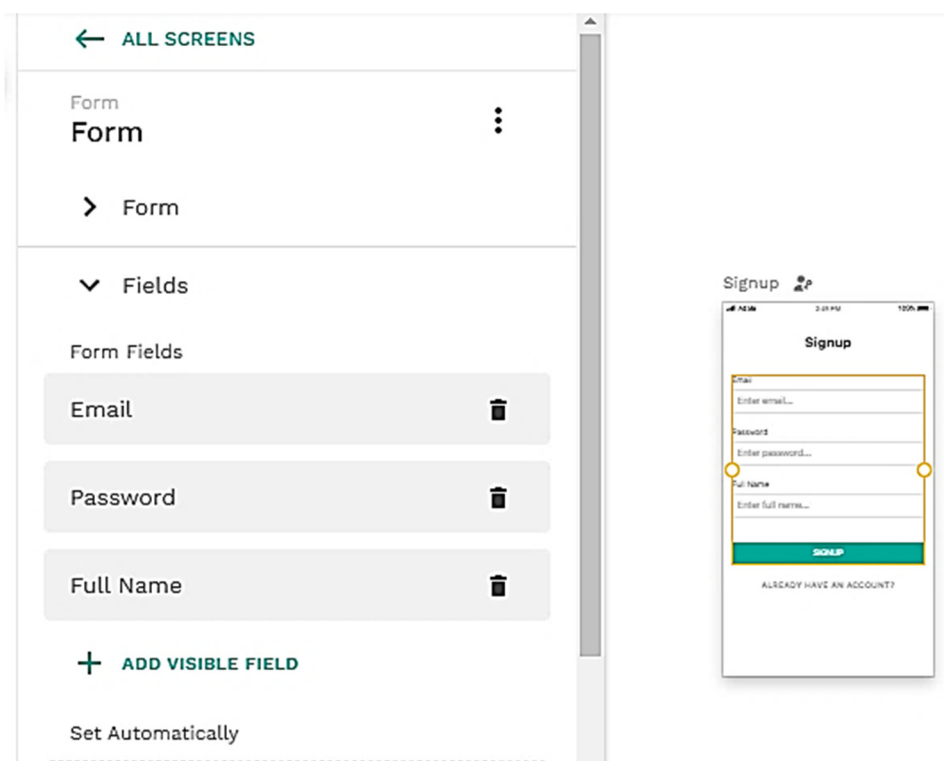


Рис. 4.10. Поля формы

Кроме формы с кнопкой, на экране есть еще дополнительный компонент – текстовая кнопка, для которой настроено действие – переход на экран Login (рис. 4.11)

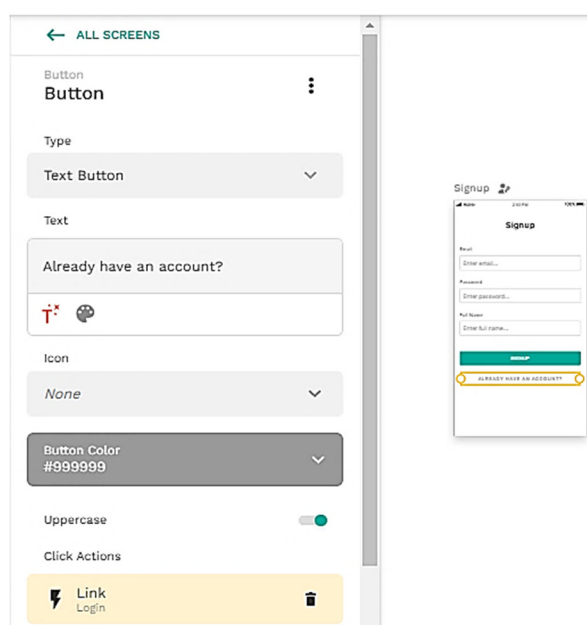


Рис. 4.11. Текстовая кнопка

С компонентами и их настройками на экране Login рекомендуется ознакомиться самостоятельно.

На экране Home есть единственный компонент App Bar, который состоит из нескольких элементов (рис. 4.12). Правая кнопка – Icon – имеет настроенные действия: Log Out, т.е. выход из аккаунта, и переход на экран Signup.

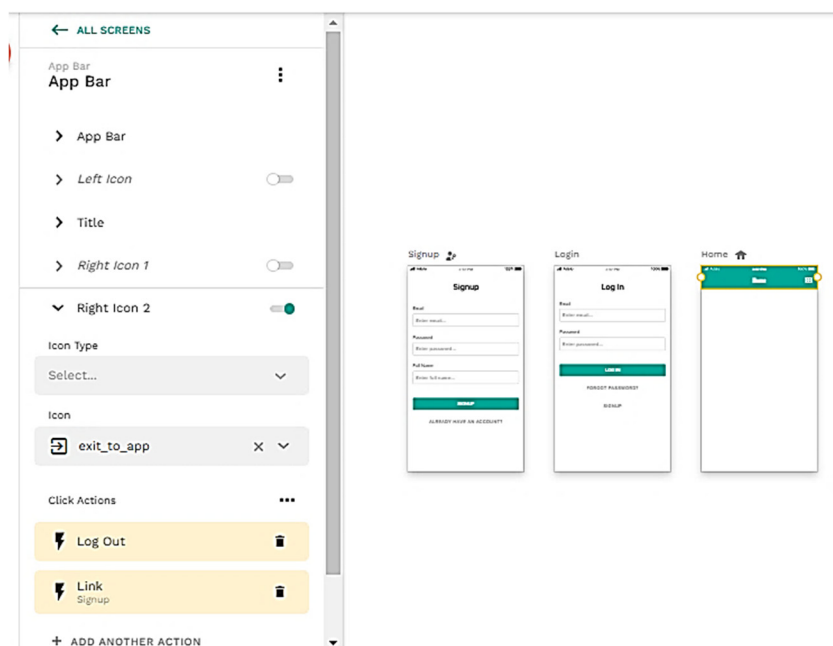


Рис. 4.12. Экран Home и настройки компонента App Bar

Также зайдем в раздел Database Collections и увидим, что там есть единственная преднастроенная таблица Users, в которой есть четыре поля/свойства и одна запись, которую мы внесли в таблицу при тестировании работы приложения (рис. 4.13).

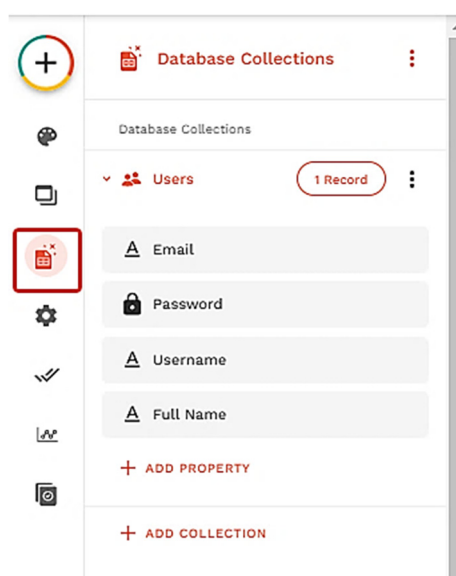


Рис. 4.13. Раздел Database Collections

Приветственный экран приложения создадим и настроим самостоятельно. На этом экране разместим текст приветствия, картинку и две кнопки: Войти и Регистрация.

Добавим новый пустой экран в приложение и назовем его Welcome (рис. 4.14). Переместим экран в начало ряда и изменим его тип с обычного экрана на Welcome screen (рис. 4.15). Значок человечка с ключом экран Signup потеряет, а только что созданный Welcome приобретет.

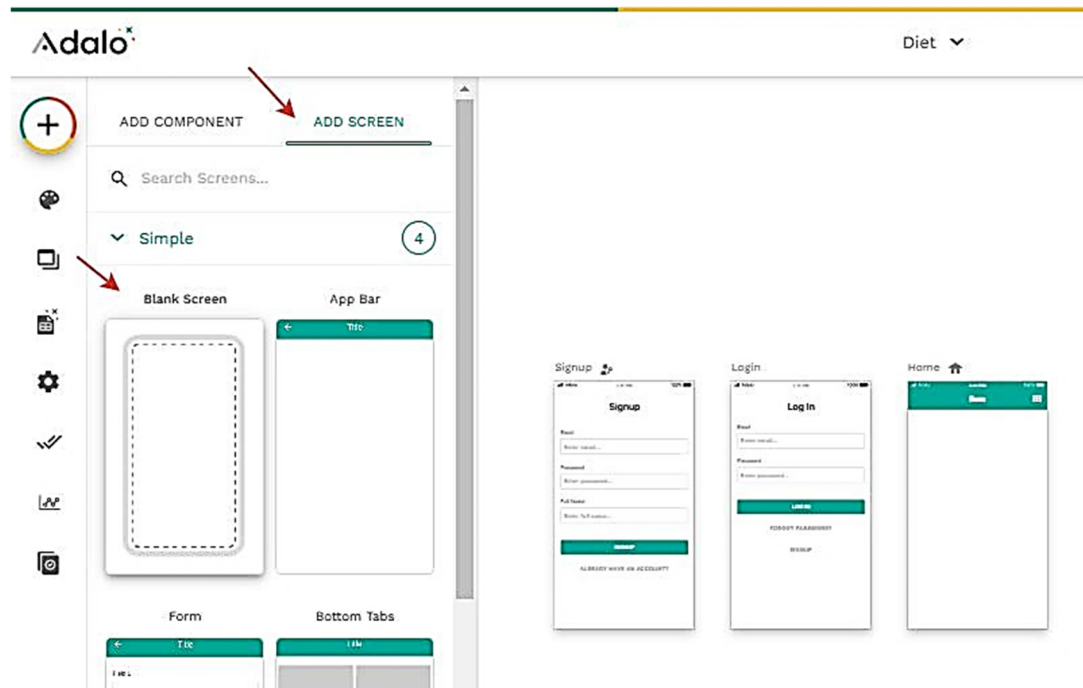


Рис. 4.14. Добавление экрана в приложение

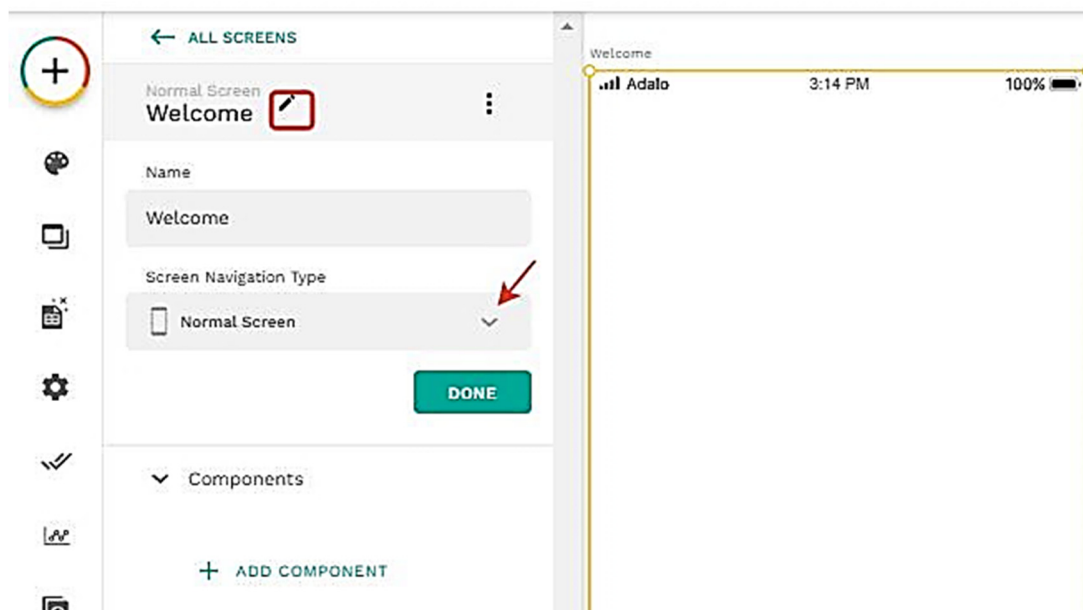


Рис. 4.15. Изменение типа экрана

Добавим на экран текстовый компонент и проведем его настройку (рис. 4.16). Значок палитры позволяет настроить форматирование вводимого текста.

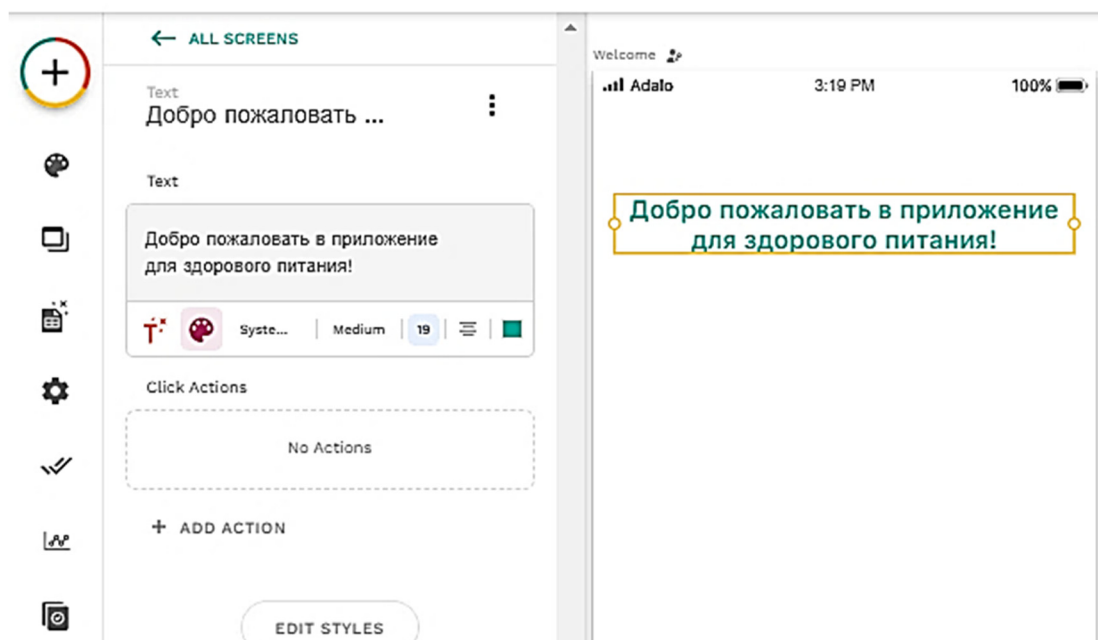


Рис. 4.16. Добавление и настройка текстового компонента

Добавим элемент Image (рис. 4.17) и первую текстовую кнопку Регистрация, для которой проведем настройку действия – переход на экран Signup (рис. 4.18).

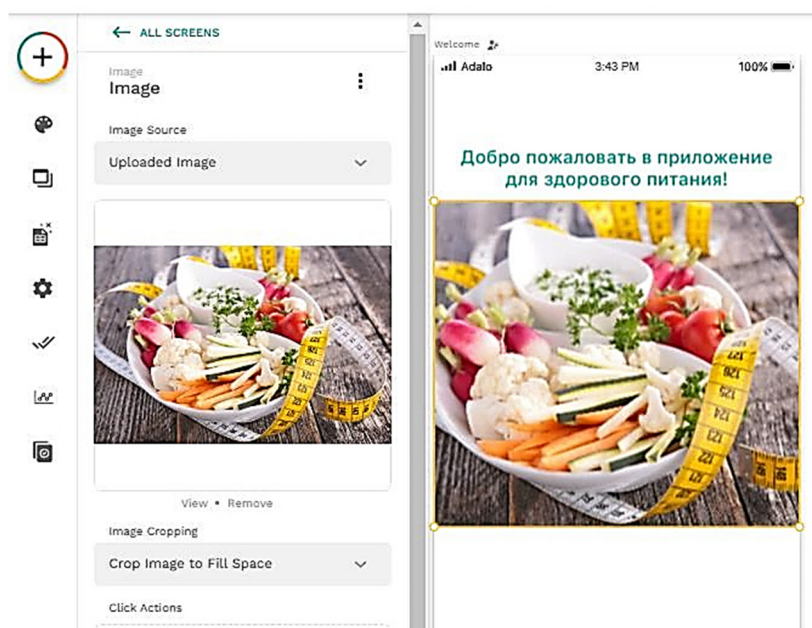


Рис. 4.17. Настройка изображения

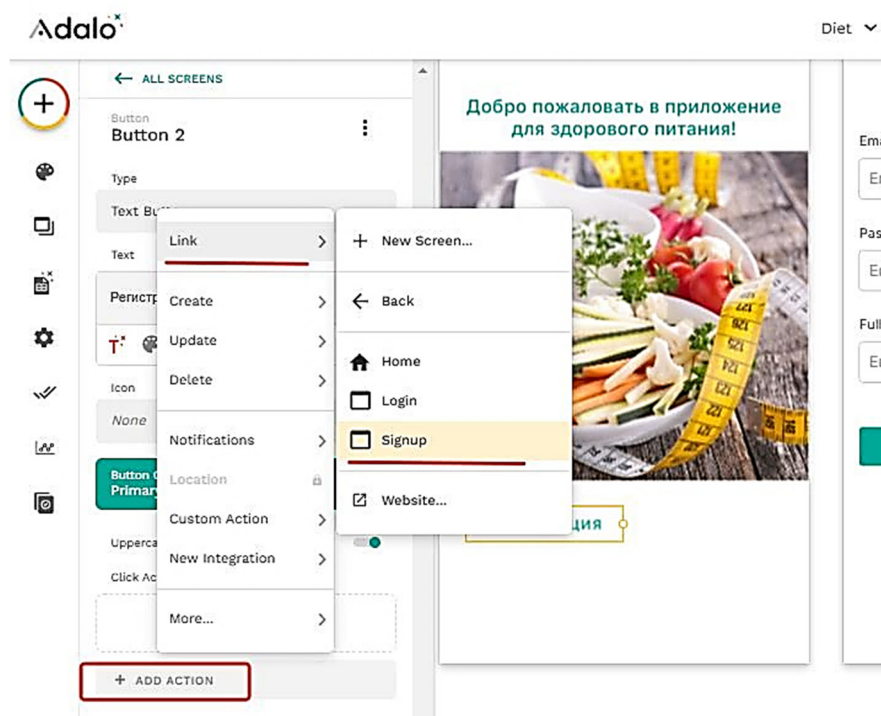


Рис. 4.18. Добавление действия для кнопки Регистрация

Добавим еще одну текстовую кнопку Войти и для нее настроим действие – переход на экран Login.

Если дефолтные экраны Login и Signup не устраивают, то можно кастомизировать (перенастроить) текстовые ссылки, пояснения и т.п. Например, так выглядит интерфейсная группа для входа/регистрации в приложении после донастройки (рис. 4.19).

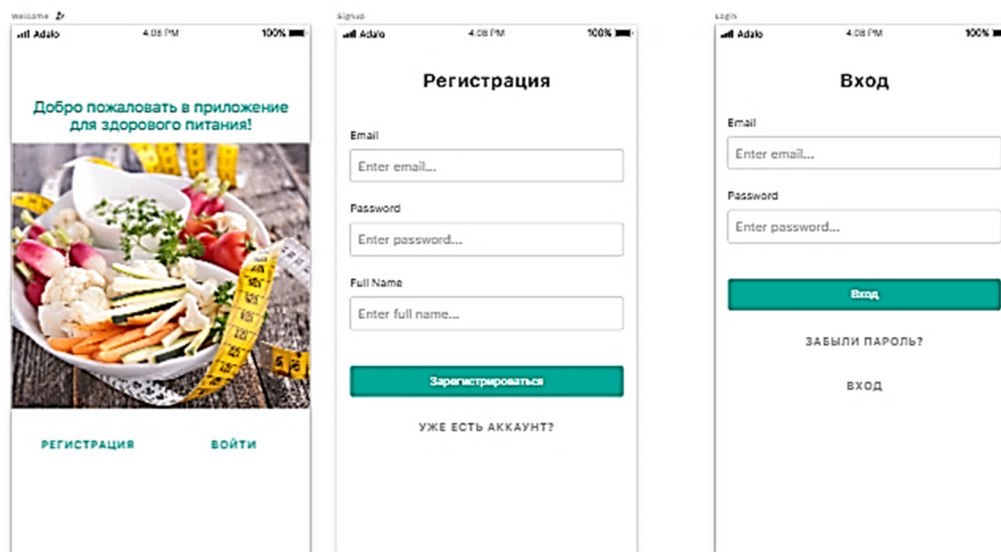


Рис. 4.19. Интерфейсная группа входа/регистрации

5. Разработка сценариев взаимодействия с приложением: описание основных интерфейсных групп и дизайн базы данных

База данных (Database) позволяет вводить данные для отображения пользователям приложения или для сбора данных, которые они вводят в приложение. Таблицы в базе данных, соответствующие определенным объектам/сущностям, называются в Adalo collection (коллекциями), а их атрибуты – это property (свойства, характеристики).

Приложение не может существовать без данных. Описывая интерфейсные группы, обязательно следует продумать и дизайн базы данных: какие сущности будут появляться в приложении и какие свойства этих сущностей важны в контексте сценариев этого приложения.

Интерфейсные группы (они же сценарии взаимодействия пользователя с приложением) для приложения «Диета» предложены следующие:

- дневник калорий: экран добавления записей в дневник, экран со списком записей дневника и экран просмотра конкретной записи из списка;
- профиль пользователя: экран личного кабинета пользователя, экран изменения информации в профиле;
- рецепты: экран выбора цели диеты, экран просмотра рецептов, отфильтрованных с учетом выбранной цели, экран любимых рецептов;
- поддержка: экран с формой обращения в поддержку, экран просмотра списка обращений и экран детализации конкретного обращения.

Для запланированных интерфейсных групп необходимо спроектировать базу данных приложения. База данных состоит из таблиц. В каждой таблице обязательно должно быть хотя бы одно поле. По умолчанию таблица создается с таким полем с дефолтным именем Name. Но при создании других полей и перемещении одного из них на первую позицию дефолтное поле можно удалить, также его можно переименовать, если тип данных поля устраивает. При добавлении поля/свойства в таблицу необходимо определиться с типом хранимых в поле данных. Наиболее общий тип данных поля – это текст. Тип данных созданного поля изменить нельзя. Можно только удалить поле и пересоздать его с новым типом заново.

Первая сущность – таблица Users – уже есть. Она обслуживает регистрацию/вход в приложение. Однако в этой таблице имеются не все поля, которые было бы логично заполнять пользователю приложения «Диета».

Дополним таблицу Users следующими полями (рис. 5.1):

- Photo; Weight; Height; Gender; Age.

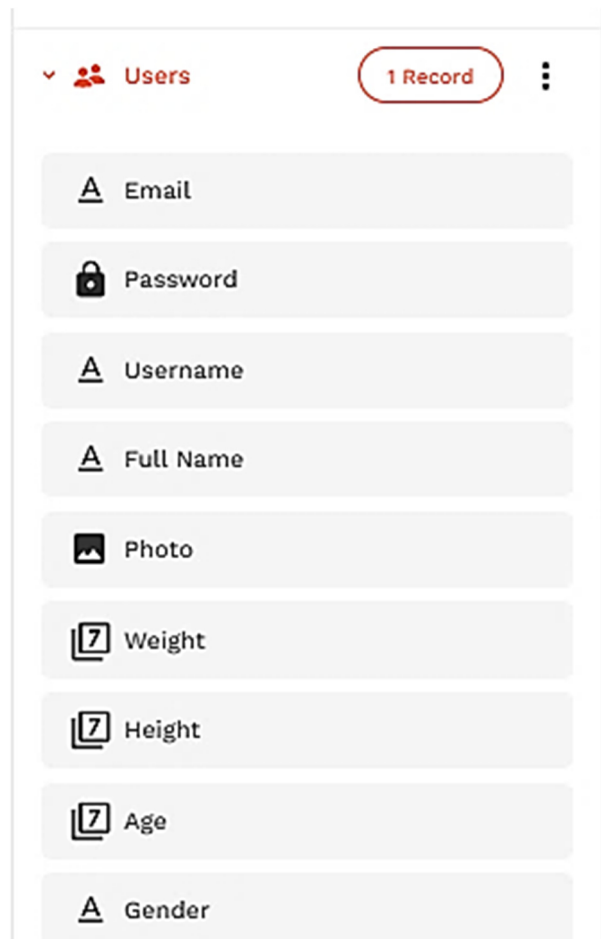


Рис. 5.1. Дополнительные поля/свойства в таблице Users

Новые таблицы, которые необходимо создать, и их свойства представлены на рис. 5.2–5.5.

Таблица Goals:

– Name_goal.

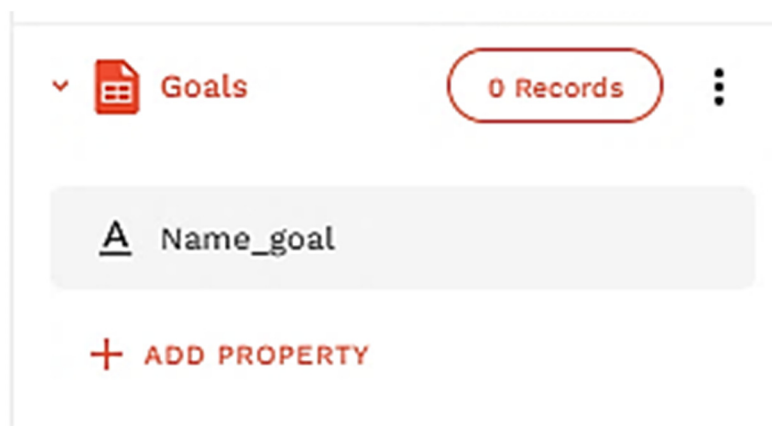


Рис. 5.2. Таблица Goals

Таблица Calories_diary:

- Date; Water; Calories.

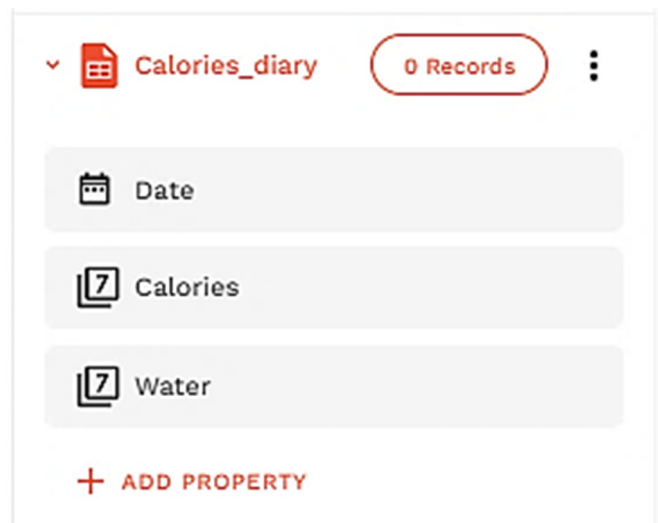


Рис. 5.3. Таблица Calories_diary

Таблица Recipes:

- Name; Photo; Calories_100g; Ingredients; Making.

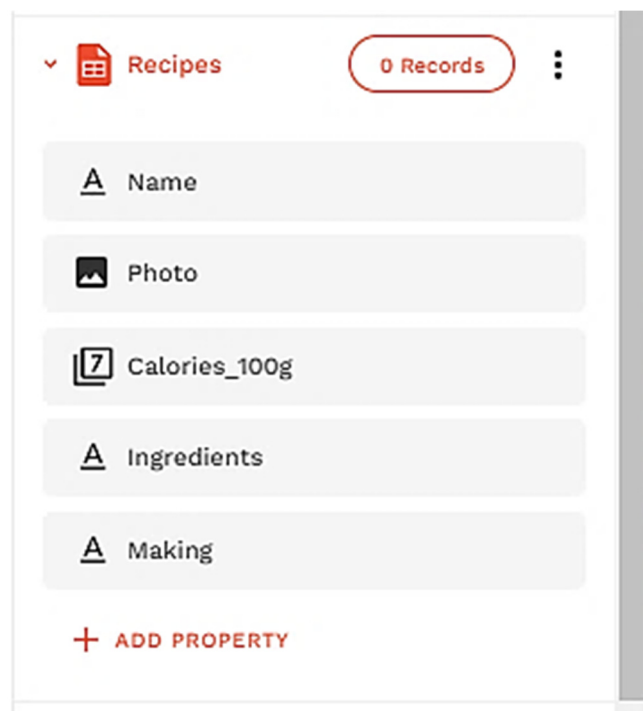


Рис. 5.4. Таблица Recipes

Таблица Request_support:

- Date; Text_request; Snapshot; Support_response.

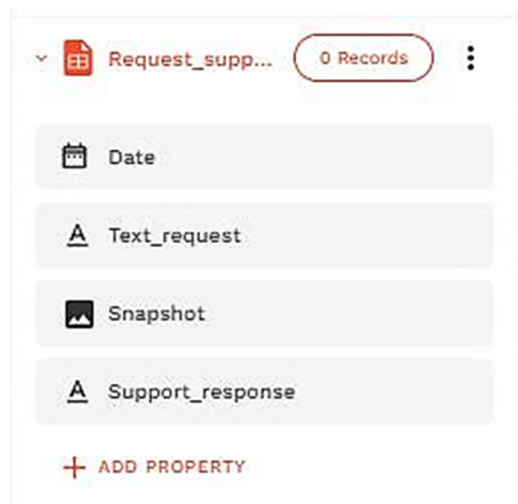


Рис. 5.5. Таблица Request_support

Перейдем к созданию и установлению связей между сущностями.

Таблицу Users необходимо связать с Goals, чтобы пользователь мог выбирать свою персональную цель для следования диете. В любой из двух связываемых таблиц, например Users, выбираем добавить Relationship (свойство) и связываемую таблицу (Goals) и устанавливаем связь (рис. 5.6).

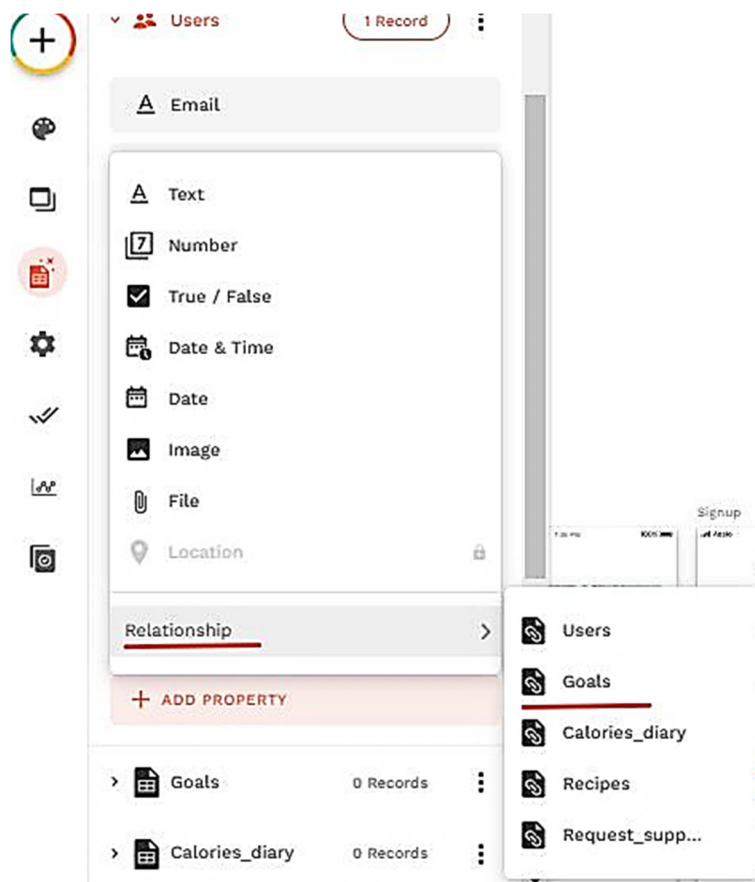


Рис. 5.6. Добавление поля Relationship

Вид связи определяется количеством участников с каждой стороны связи: так, у пользователя одна цель, но одну и ту же цель могут иметь многие пользователи (рис. 5.7).

How are Goals & Users related?

☐ G = Goals ☒ U = Users

```
graph TD; G1[G] --- U1[U]; G1 --- U2[U]; G1 --- U3[U];
```

A Goal can have multiple Users
A User belongs to one Goal

☒

```
graph TD; G1[G] --- U1[U]; G2[G] --- U1; G3[G] --- U1;
```

A Goal can only have one User
A User can have multiple Goals

☐

```
graph TD; G1[G] --- U1[U]; G1 --- U2[U]; G1 --- U3[U]; G2[G] --- U1; G2 --- U2; G2 --- U3; G3[G] --- U1; G3 --- U2; G3 --- U3;
```

A User can have multiple Goals
A Goal can have multiple Users

☐

CANCEL DONE

Рис. 5.7. Определение вида связи между таблицами

Поле связи Goals появится в таблице Users, но и поле связи Users появится в таблице Goals.

Users связываем с Calories_diary, у одного пользователя может быть много дневниковых записей (на каждую дату своя), а у записи в дневнике может быть только один пользователь.

Users необходимо связать и с Recipes, чтобы пользователь мог помещать рецепты в избранное, у одного пользователя может быть много любимых рецептов, а у рецепта может быть множество выбравших его пользователей.

Users связываем с Request_support, у одного пользователя может быть много обращений в поддержку, а у каждого конкретного обращения может быть только один отправивший его пользователь.

Также необходимо связать Recipes и Goals. Рецепт соответствует одной цели, а одной цели может отвечать множество рецептов.

Среди созданных таблиц будут те, которые заполняются разработчиками приложения: Goals, Recipes, а также те, которые заполняются пользователями с помощью экранных форм.

После установления связей внесем две цели (уменьшение веса, набор массы) в таблицу Goals. Для этого кликнем мышкой по числу записей рядом с таблицей и выберем в пустой таблице опцию – добавить (Add Goal). Затем аналогично в таблицу Recipes внесем три рецепта (рис. 5.8) так, чтобы рецепты отвечали разным целям [3].

← Recipe

Name: Салат с курицей и яичными блинчиками

Photo:  View Remove

Calories 100g: 81

Ingredients: Отварное филе – 280 г.
Калуста пекинская – 400 г.
Кукуруза – 1 банка.
Яйца – 3 шт.
Молоко – 50 мл.
Сметана 10 % – 80 г.
Чеснок – 2 дольки.
Зелень, соль.

Making: Из яиц и молока делаем болтушку, добавляем щепотку соли. Обжариваем яичные блинчики на сухой антипригарной сковороде с двух сторон. Пекинскую калусту шинкуем, филе разбираем на волокна. Остывшие блинчики поштучно сворачиваем в рулет, режем соломкой. Для заправки смешиваем нежирную сметану, чеснок, петрушку, подсаливаем по вкусу. Все ингредиенты смешиваем между собой.

Users: Add User...

Goal: Уменьшение веса

CANCEL SAVE

Рис. 5.8. Внесение в таблицу рецепта

6. Интерфейсная группа: Рецепты

Добавим в приложение три новых экрана: Recipes, Recipe_details и Favorite_recipes. Для новых экранов создадим AppBar с помощью нескольких компонентов:

- прямоугольник: Simple>Rectangle (и выберем брендовый цвет заливки);
- слева значок возврата на предыдущий экран Buttons>Icon (arrow_back) с действием возврата назад (рис. 6.1);
- справа значок выхода из приложения Buttons>Icon (exit_to_app) с действиями выхода из приложения (рис. 6.2) и возврата на стартовый экран;
- в центре текстовый блок с названием экрана.

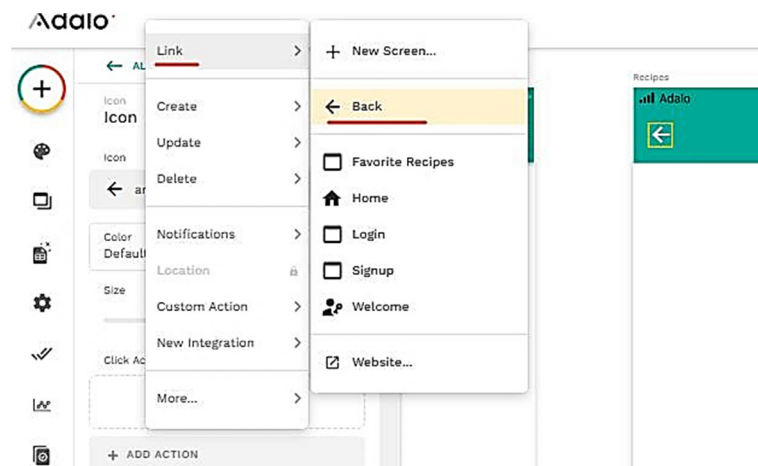


Рис. 6.1. Настройка действия для Icon

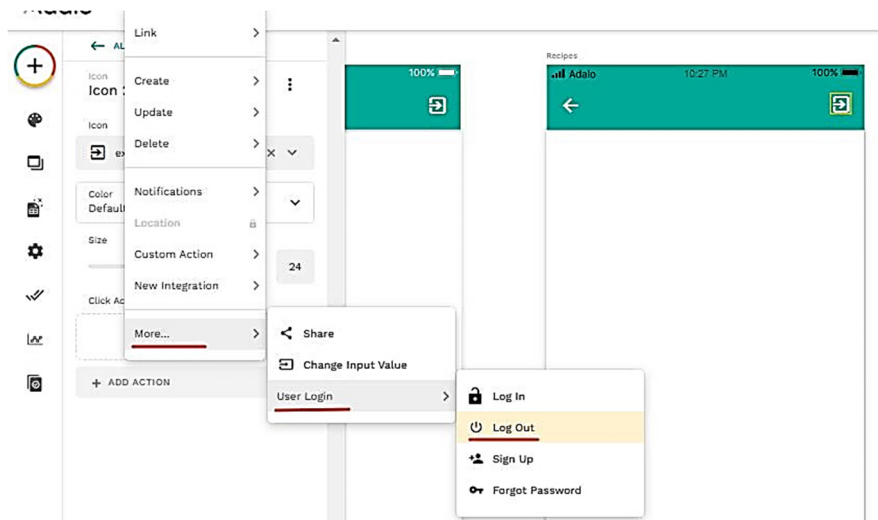


Рис. 6.2. Настройка действия для Icon 2

Выделим все четыре компонента и создадим из них группу (рис. 6.3), которую назовем AppBar (рис. 6.4). Теперь всю группу можно копировать и размещать на новых экранах как один элемент. При этом к каждому компоненту есть доступ и текст заголовка экрана можно легко изменить.

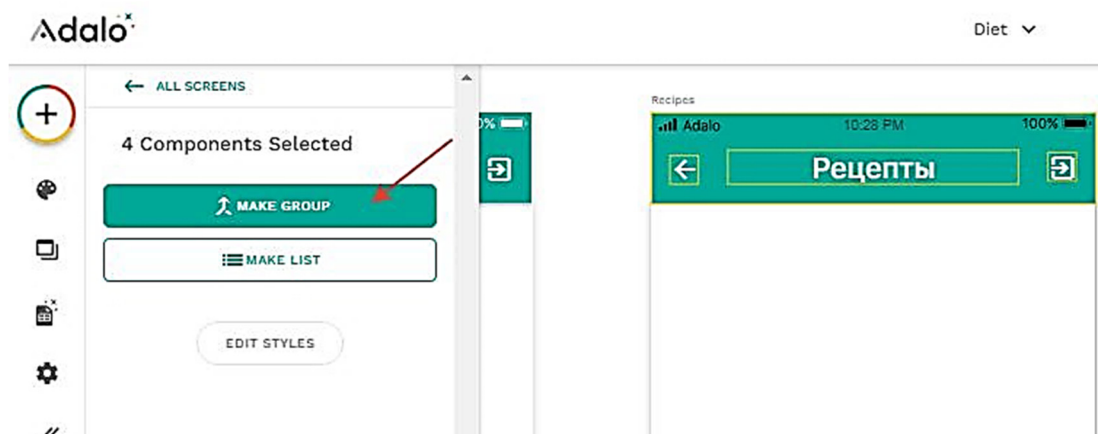


Рис. 6.3. Создание группы компонентов

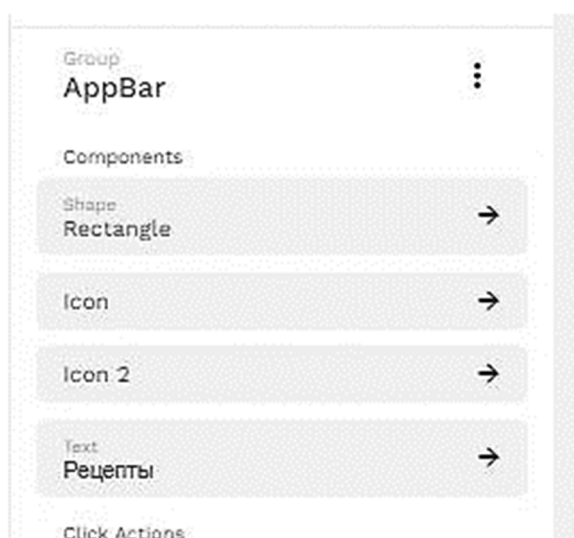


Рис. 6.4. Переименование группы компонентов

На экран Home добавим компонент Forms&Fields>Dropdown Menu со следующими настройками (рис. 6.5).

Также добавим на этот экран текстовую кнопку Рецепты с настройкой действия – переход на экран Recipes.

На экран Recipes добавим Simple list со следующими настройками: список основан на таблице Recipes, но для него добавлен пользовательский фильтр: текущие рецепты – те, которые отвечают выбранной в Dropdown Menu цели (рис. 6.6), т.е. Custom Filter: Current Recipe>Goal>Name_goal Is equal to Selected Goal>Name_goal.

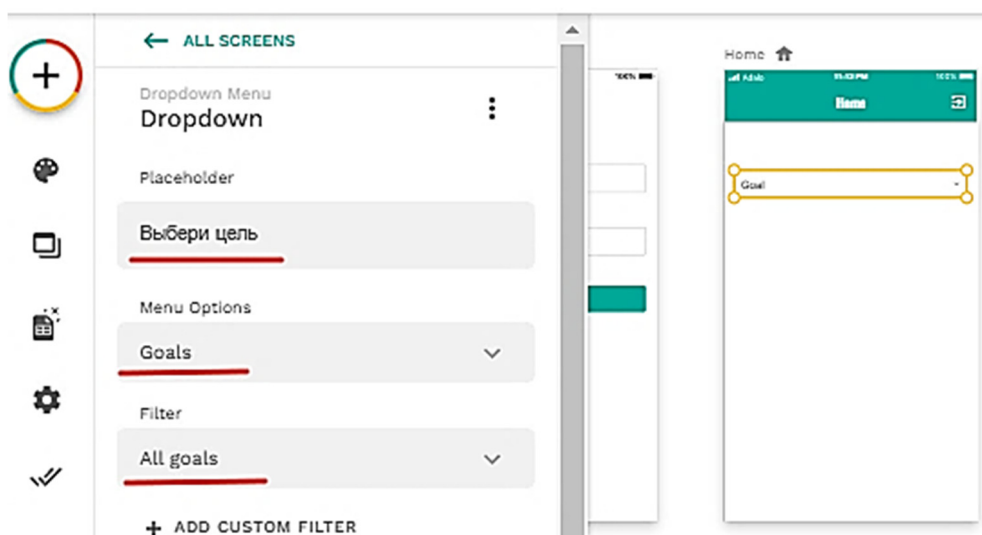


Рис. 6.5. Настройка компонента Dropdown

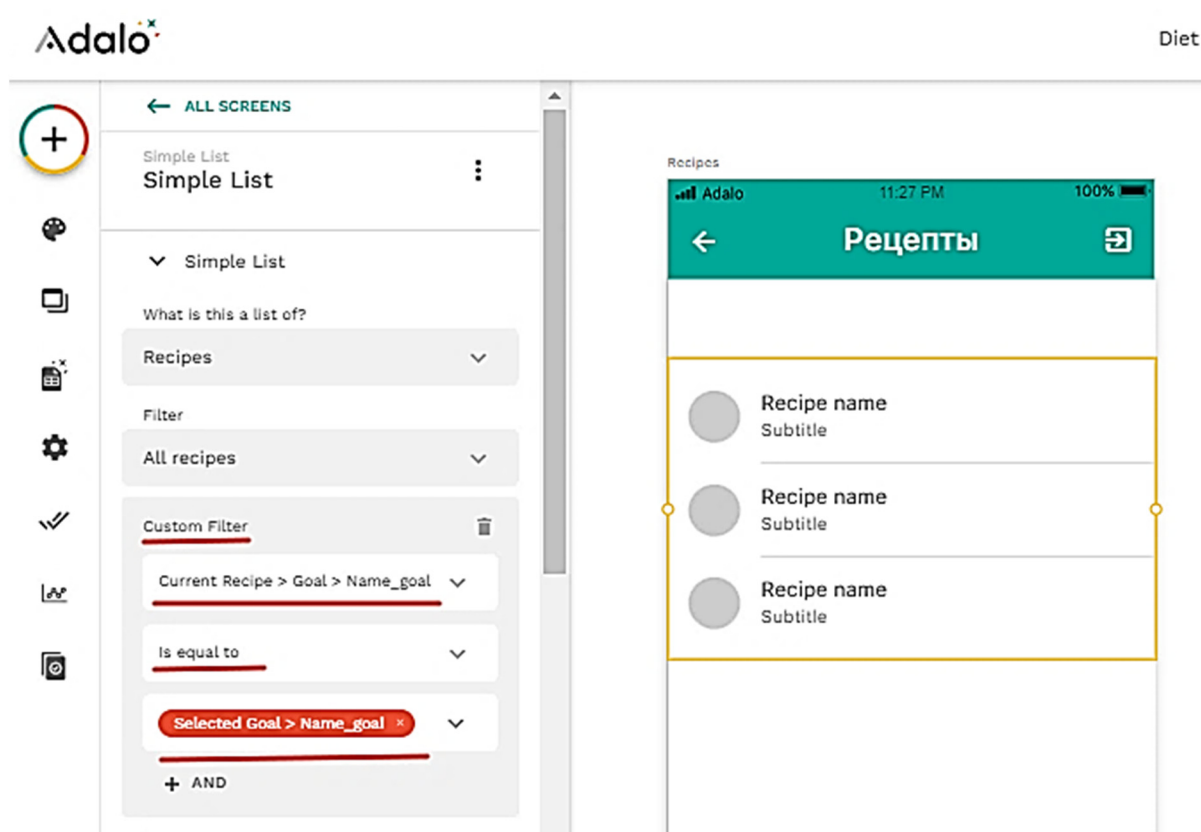


Рис. 6.6. Настройки для Simple list

С помощью полосы прокрутки спустимся в часть с настройками действия для списка и добавим переход на экран Recipe_details, а для отображения элементов списка отключим демонстрацию Subtitle.

Теперь при выборе любого рецепта в списке мы переходим на экран Recipe_details с его описанием, который пока пуст.

Для текстового компонента, размещенного на экране, доступна функция Magic Text, ссылающаяся на поле таблицы, из которого компонент забирает информацию (рис. 6.7). Это поле форматируется обычным образом.

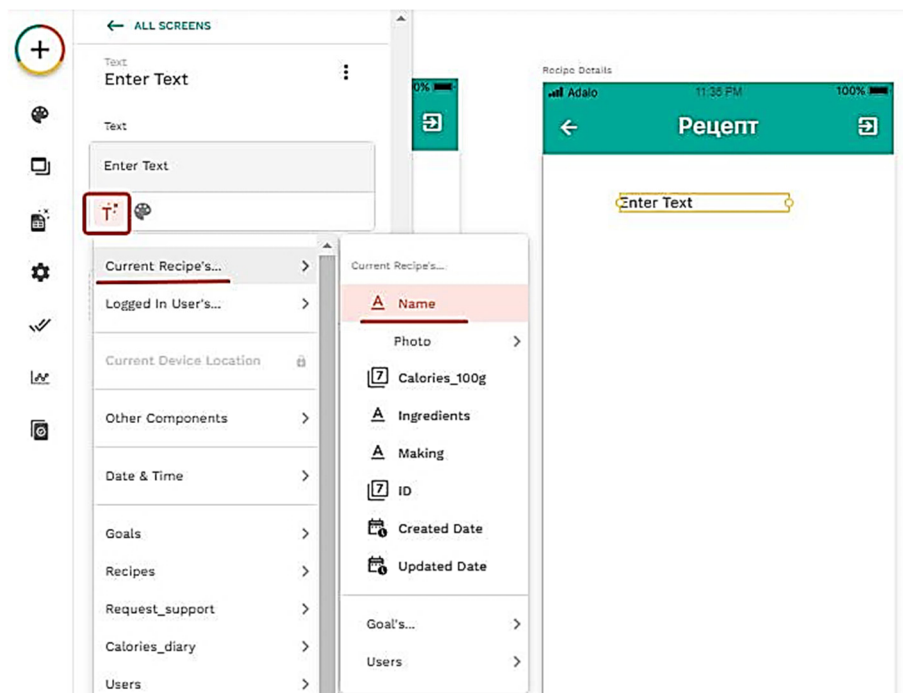


Рис. 6.7. Настройка Magic Text

Для размещенного на экране компонента Image также настроим связь с базой данных (рис. 6.8).

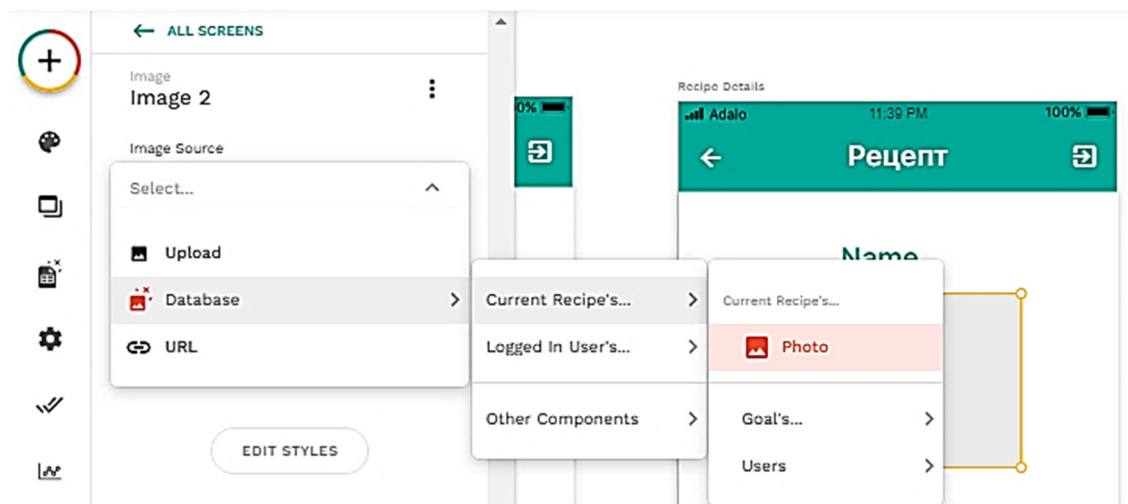


Рис. 6.8. Настройка компонента Image

Добавим все остальные поля на экран и протестируем отображение (рис. 6.9).

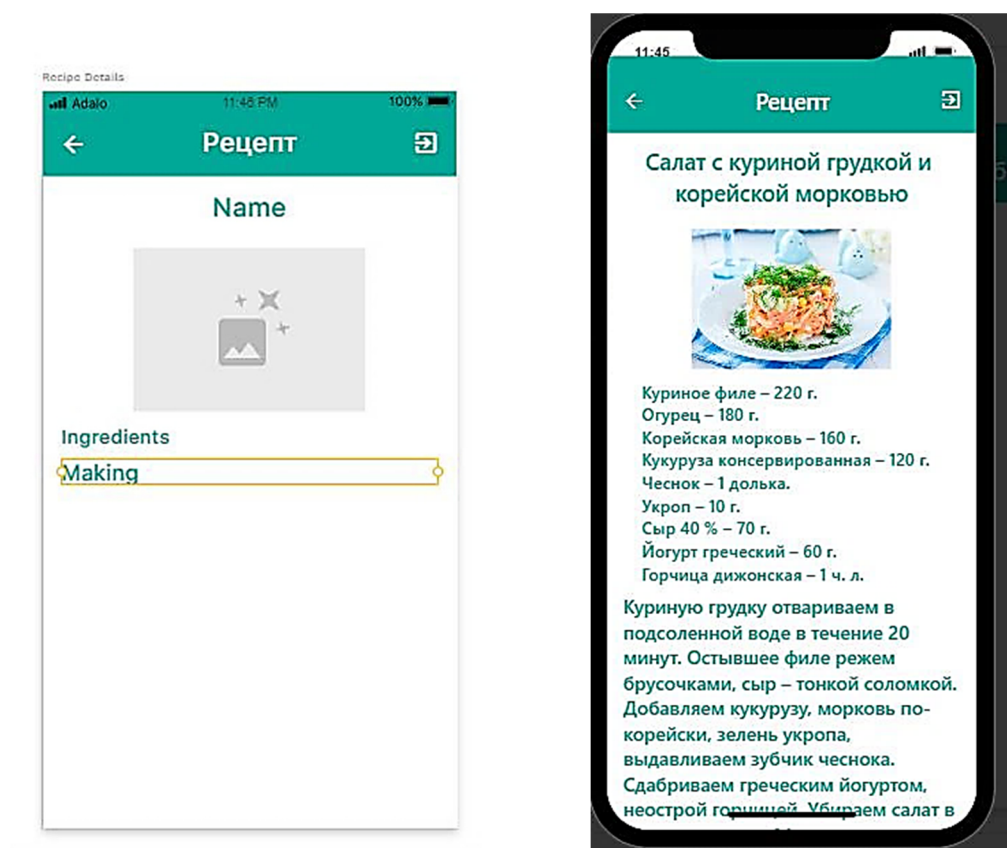


Рис. 6.9. Экран Recipe_details

Завершим настройку интерфейсной группы Рецепты добавлением на экран Recipe_details переключателя для избранного Buttons>Toggle (рис. 6.10).

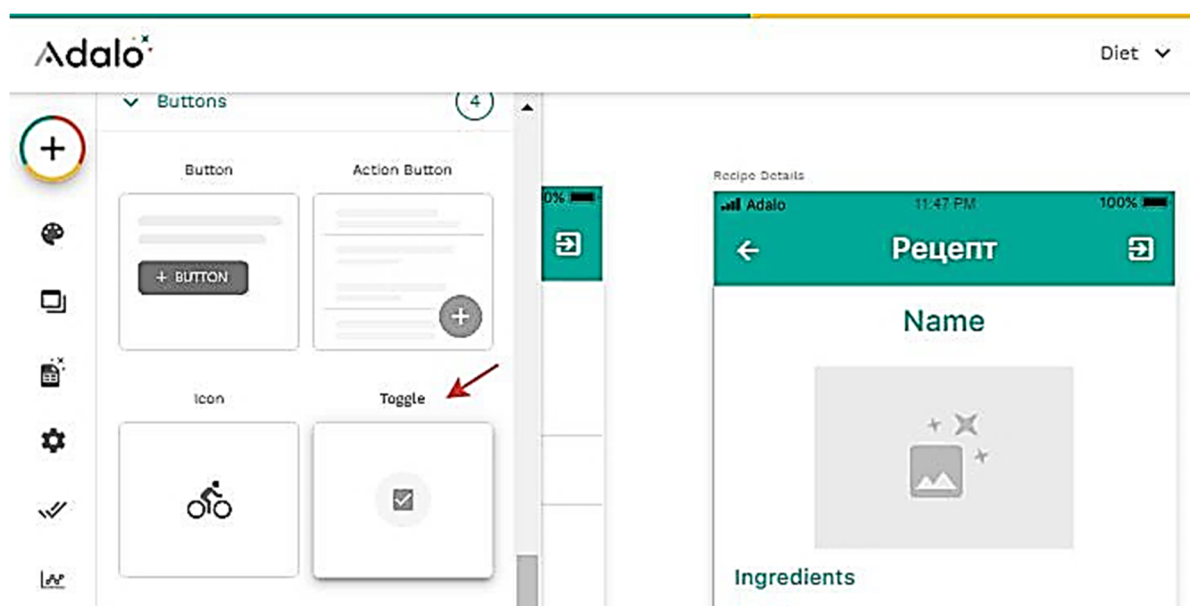


Рис. 6.10. Добавление переключателя

Этот переключатель должен быть связан с авторизованным пользователем и текущим рецептом, поэтому необходимо подключение к базе данных (рис. 6.11).

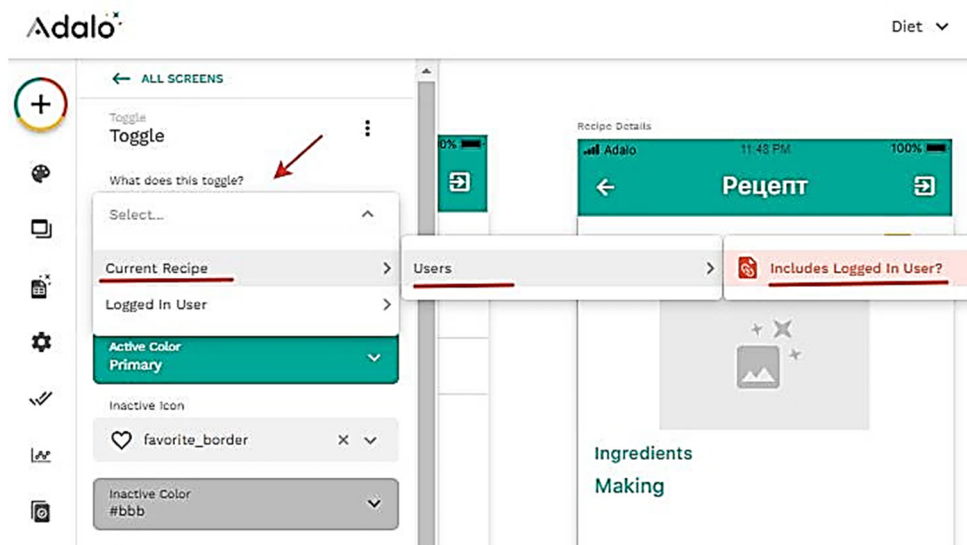


Рис. 6.11. Настройка связи с базой данных

Сам переключатель также нуждается в настройках (рис. 6.12).

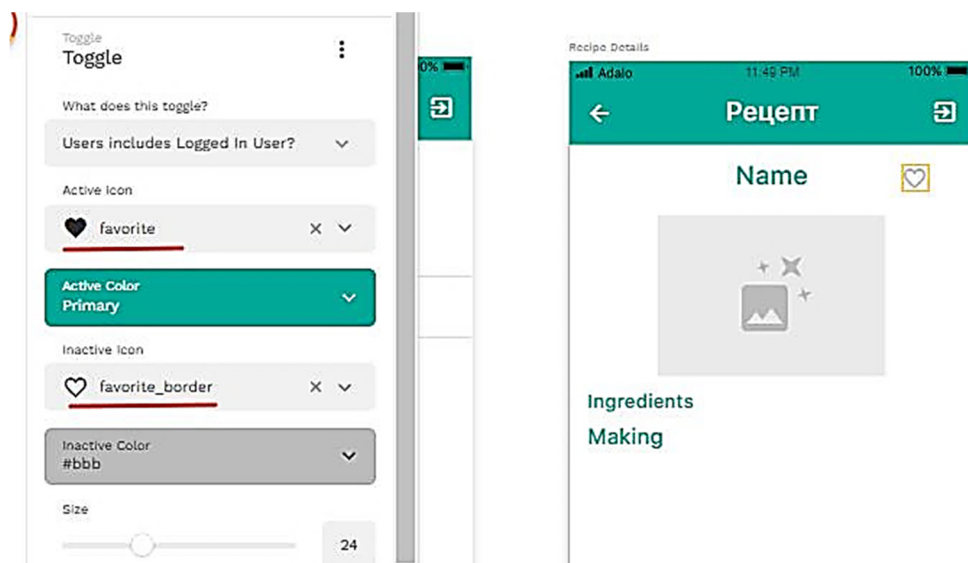


Рис. 6.12. Настройки для Toggle

Протестируем настроенную опцию в приложении и изменим переключатель Избранное для одного из рецептов.

Осталось поместить на экране Home кнопку Рецепты со значком сердца и настроить для нее действие – переход на экран Favorite_recipes (рис. 6.13), чтобы получить доступ к списку любимых рецептов.

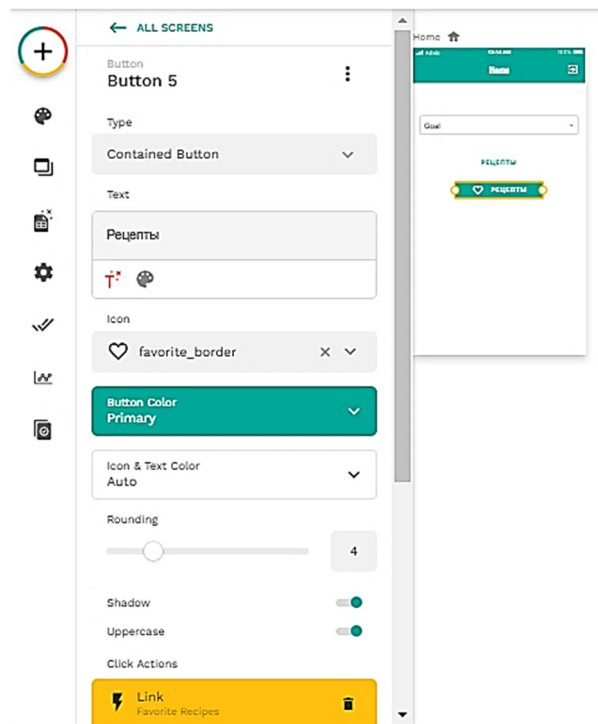


Рис. 6.13. Настройки для кнопки на экране Home

На экране Favorite_recipes разместим список Lists>Image list и настроим его связь с базой (рис. 6.14).

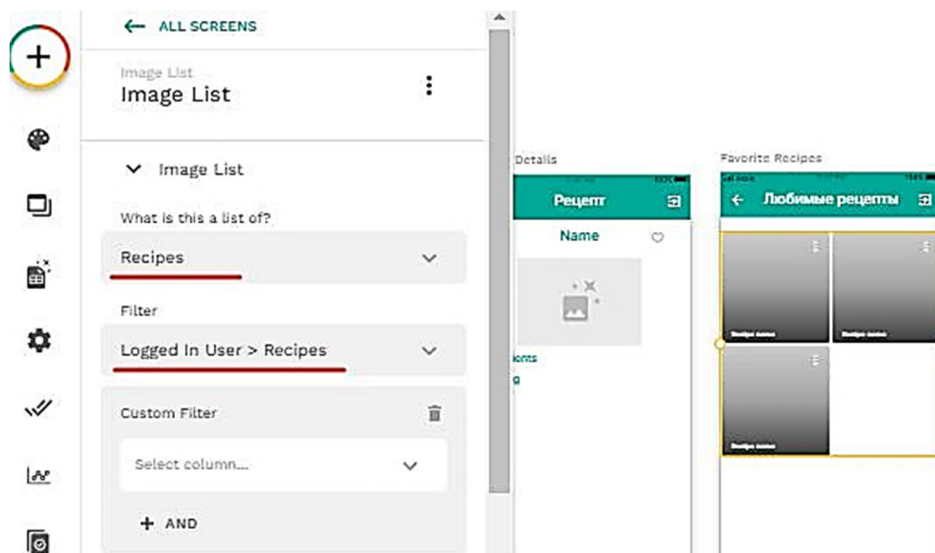


Рис. 6.14. Настройка связи с базой данных

Также протестируем, что при переходе с экрана Home на экран Favorite_recipes отображается тот рецепт, который был отмечен значком Избранное (рис. 6.15).



Рис. 6.15. Тестирование отметки Избранное для рецепта

Интерфейсная группа Рецепты настроена (рис. 6.16)

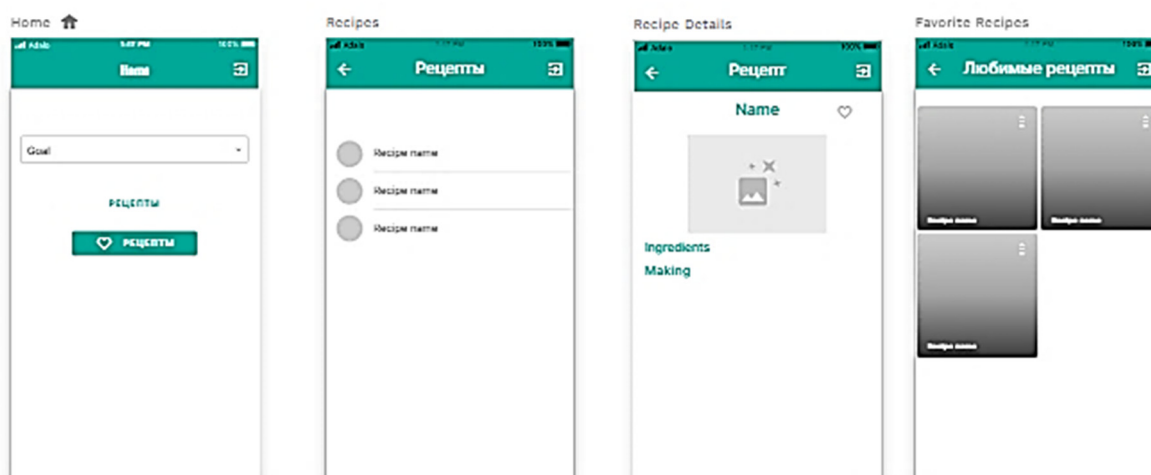


Рис. 6.16. Интерфейсная группа Рецепты

7. Интерфейсная группа: Дневник калорий

Добавим в приложение три новых экрана: `Calories_diary`, `Day_details` и `Add_day`.

На экране `Home` разместим текстовую кнопку Дневник калорий с настроенным переходом на экран `Calories_diary`. На экран `Calories_diary` добавим компонент Список (`Lists>Custom List`), связав его с соответствующей таблицей `Calories_diary` базы данных. В этом списке с помощью `Magic Text` настроим связь заголовка и подзаголовка с нужными полями таблицы (датой и числом потребленных калорий). Для списка также настроим действие – переход на экран `Day_details` для просмотра конкретной записи из списка. На экране `Calories_diary` нужна и текстовая кнопка Добавить запись для перехода на экран `Add_day`.

А для `Day_details` настроим с помощью `Magic Text` отображение конкретной записи из таблицы `Calories_diary`. Также добавим с помощью `Magic Text` отображение полного имени авторизованного пользователя из таблицы `Users`.

Интерфейсная группа Дневник калорий в первых двух экранах (рис. 7.1) похожа на ранее созданную интерфейсную группу Рецепты.

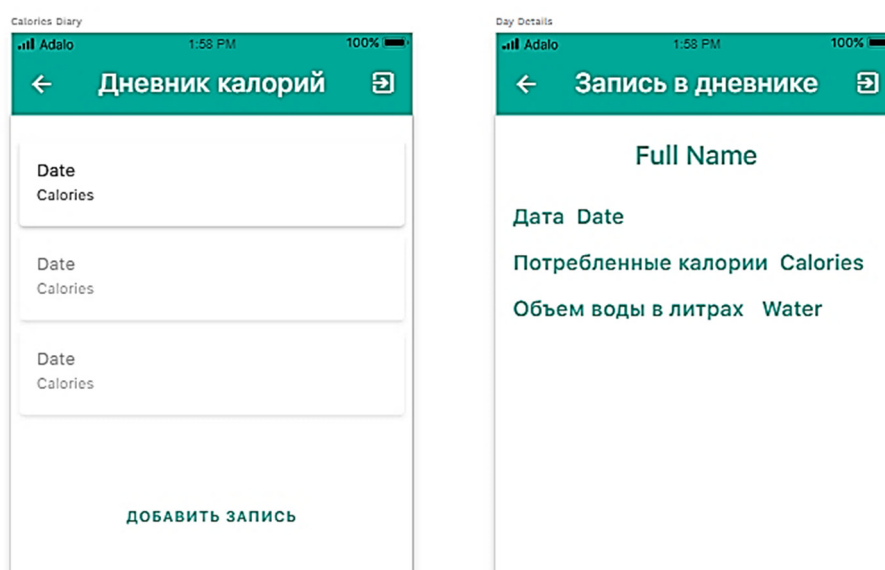


Рис. 7.1. Экраны `Calories_diary` и `Day_details`

Однако экран `Add_day` включает компонент формы, с которым ранее работали в интерфейсной группе входа/регистрации, но в уже в готовом виде.

На экран `Add_day` добавим форму и свяжем ее с таблицей базы данных, в которую должна добавляться новая запись (рис. 7.2).

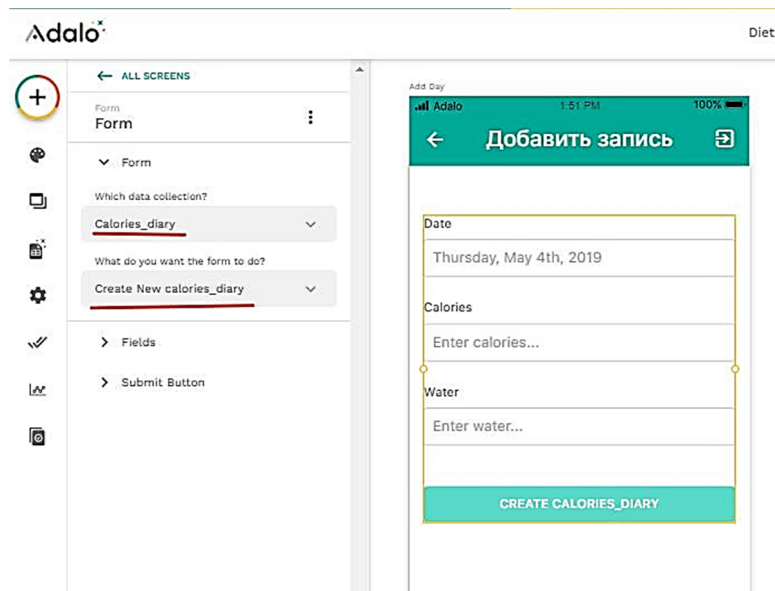


Рис. 7.2. Настройка формы

В настройках для кнопки уже есть действие – добавление новой записи в таблицу. Но добавим еще одно действие – возврат на экран `Calories_diary`. Протестируем добавление записи в дневник и ее отображение (рис. 7.3).

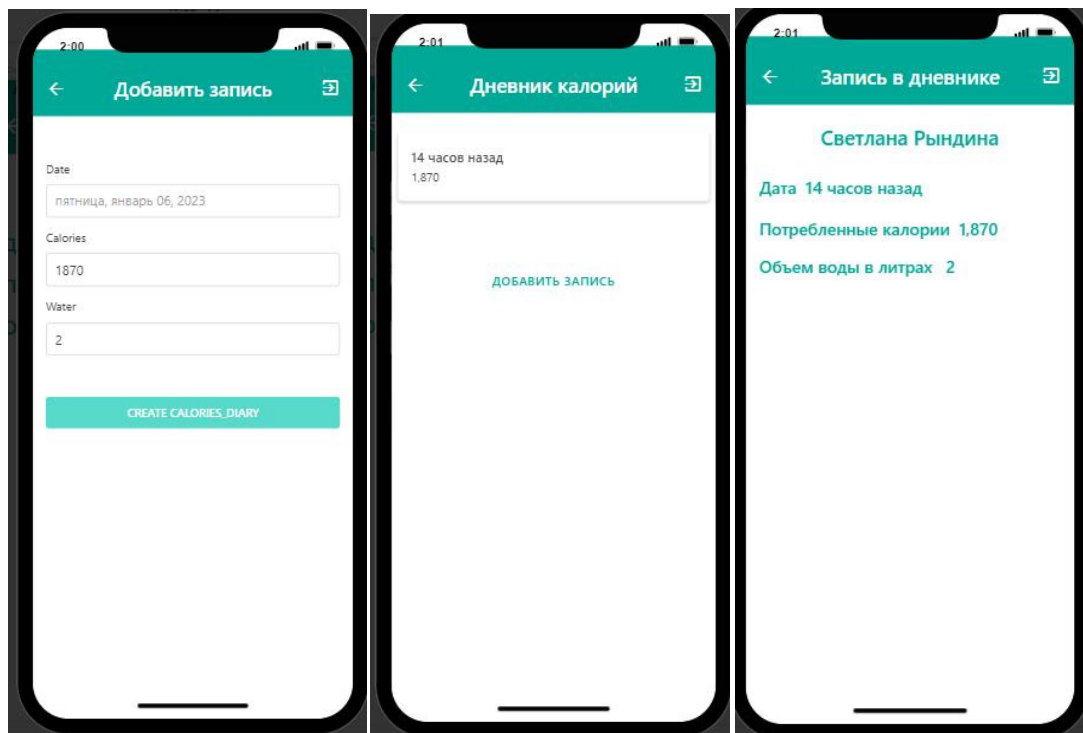


Рис. 7.3. Тестирование добавления записи в Дневник калорий

Работу с интерфейсной группой Дневник калорий на этом можно считать завершенной.

8. Интерфейсные группы: Профиль пользователя, Связь с поддержкой

Добавим в приложение три новых экрана: Profile, Edit_profile и Support.

На экране Profile с помощью текстовых компонентов Magic Text разместим информацию из таблицы Users об авторизованном пользователе, а также добавим кнопку Редактировать профиль для перехода на экран Edit_profile (рис. 8.1).

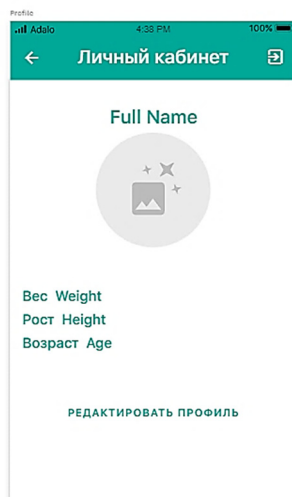


Рис. 8.1. Экран Profile

На экране Edit_profile разместим форму и свяжем ее с таблицей Users базы данных (рис. 8.2), настроим действие – обновление профиля. Все поля формы сделаем необязательными для заполнения (рис. 8.3).

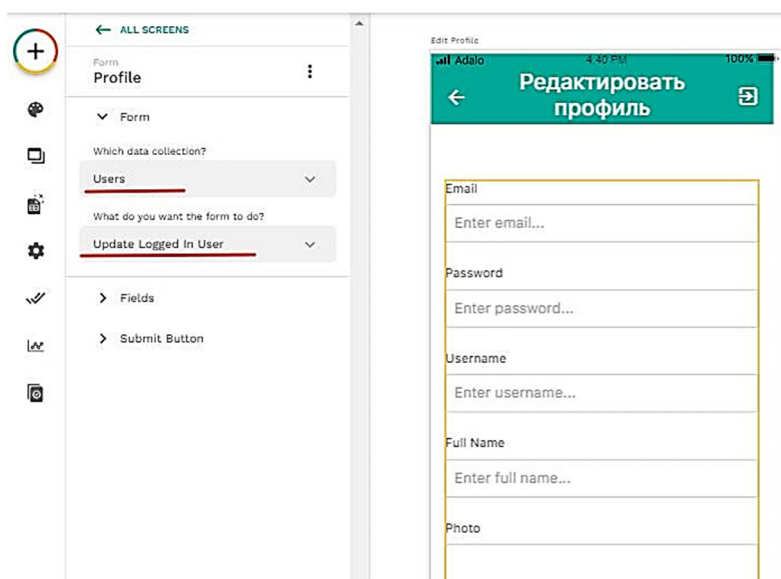


Рис. 8.2. Связь формы с таблицей данных

Form Profile

> Form

Fields

Form Fields

Email

Label
Email

Placeholder
Enter email...

Input Type
Single Line

Required Error Text
Not Required

DONE

Password

Username

Рис. 8.3. Настройки для полей формы

Увеличим размер экрана до размера формы. И для кнопки формы добавим еще одно действие – возврат на экран Profile (рис. 8.4).

← ALL SCREENS

Form Profile

> Form

> Fields

Submit Button

Text
UPDATE USER

Click Actions

Update User

Link Profile

+ ADD ANOTHER ACTION

EDIT STYLES

CHOOSE PHOTO

Weight
Enter weight...

Height
Enter height...

Age
Enter age...

Gender
Enter gender...

Goal
Select goal...

UPDATE USER

Рис. 8.4. Добавление действия кнопке

Хотя экран Support еще не настроен, но все основные интерфейсные группы в наличии. Пришло время создать меню навигации для перемещения по экранам приложения и добавить компонент Navigation>Tab Bar на экран Home. Всего на панели меню вкладок находится пять значков, вид которых можно менять, а также настраивать действия перехода на нужный экран. Один из значков соответствует текущей вкладке (рис. 8.5).

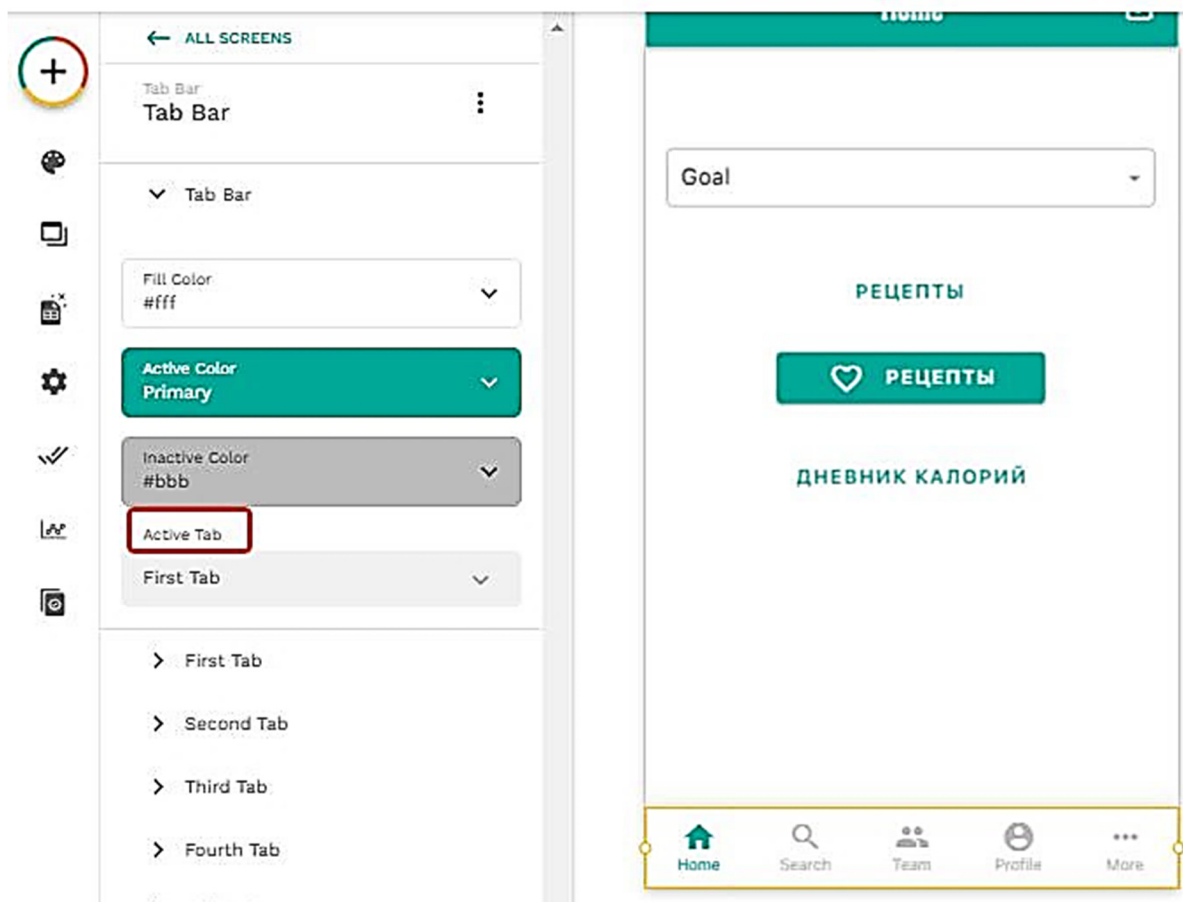


Рис. 8.5. Настройка Tab Bar

Значки Home и Profile оставим без изменений, как и их местоположение. Вместо Search будет значок Recipes, вместо Team – значок Diary, а More заменим на Support. Также для каждой иконки настроим действие – переход на соответствующую вкладку (рис. 8.6). В дальнейшем это меню будет скопировано на все основные экраны. При этом необходимо изменять Active Tab в соответствии с текущим экраном (рис. 8.7) и проверять, настроены ли действия перехода на всех неактивных значках.

Теперь настроим экран Support. На нем разместим список обращений в поддержку Lists>Custom List и кнопку перехода на экран создания обращения в поддержку (рис. 8.7).

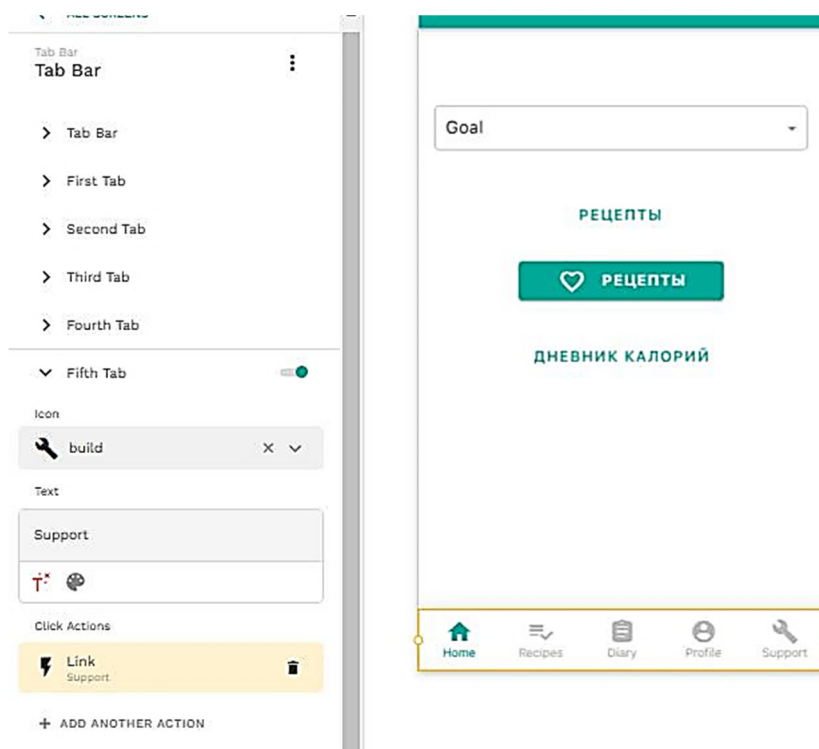


Рис. 8.6. Итоговый вид Tab Bar

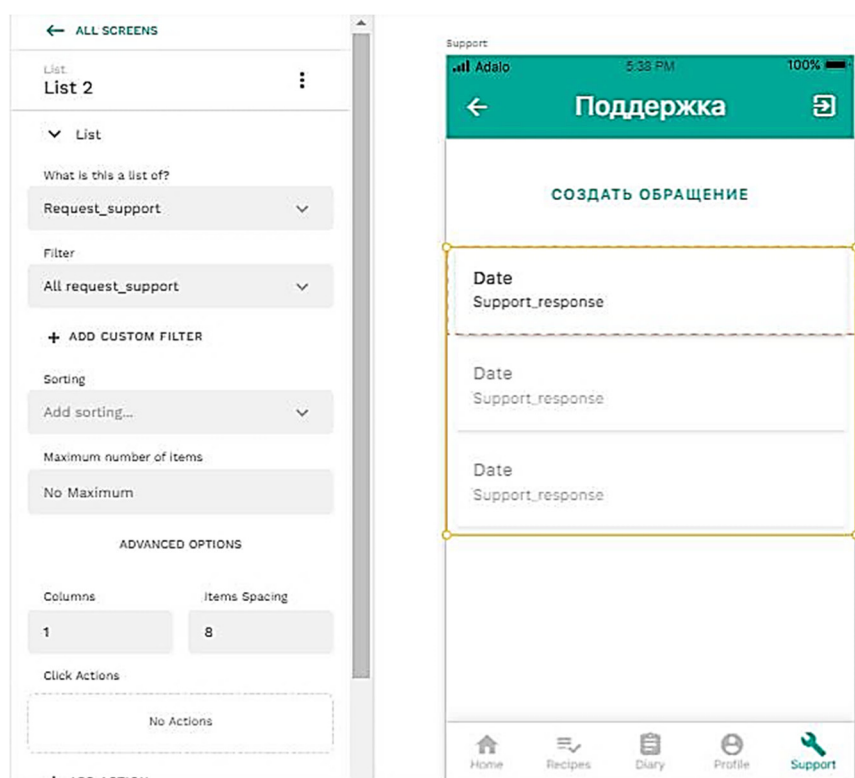


Рис. 8.7. Настройка списка обращений

Так как у нас появился список, то требуется создать экран Request_details для просмотра обращений с компонентами Magic Text (рис. 8.8)

и настроить в списке обращений действие – переход на экран Request_details.

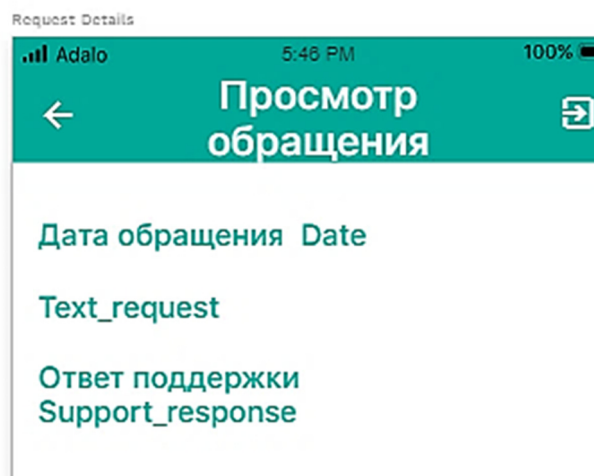


Рис. 8.8. Настройка экрана Request_details

Экран с формой обращения в поддержку Support_request также требуется создать. В форме не будет поля Support_response (рис. 8.9), так как ответ поддержки будет добавляться в таблицу сотрудником, а просмотреть ответ можно будет при переходе из списка обращений на экран Request_details. Поле Snapshot сделаем необязательным для заполнения.

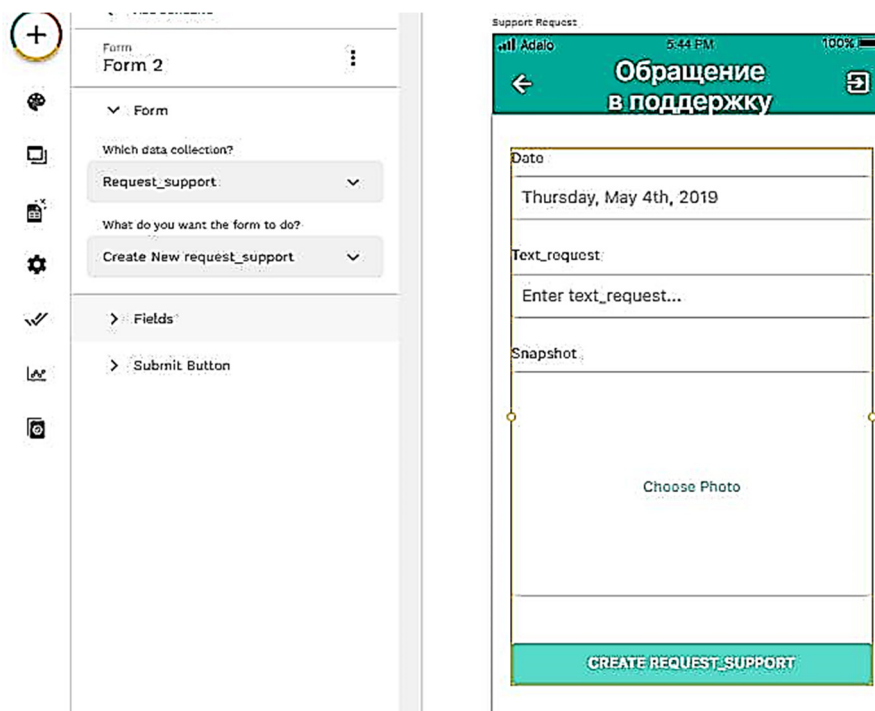


Рис. 8.9. Настройка формы

Таким образом, настроены две интерфейсные группы: Профиль пользователя (рис. 8.10) и Связь с поддержкой (рис. 8.11).

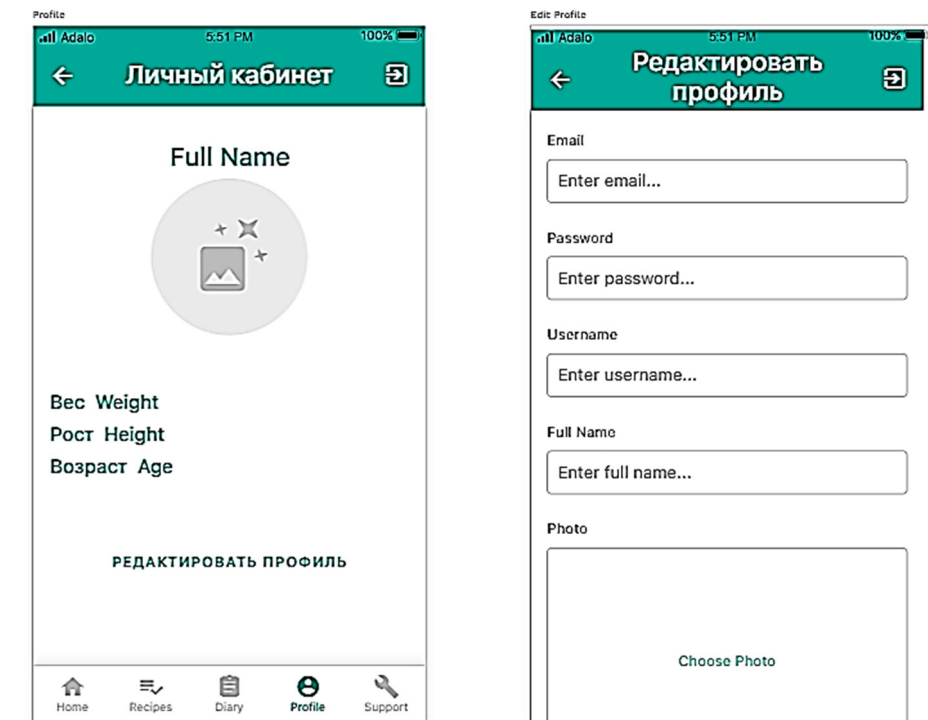


Рис. 8.10. Интерфейсная группа: Профиль пользователя

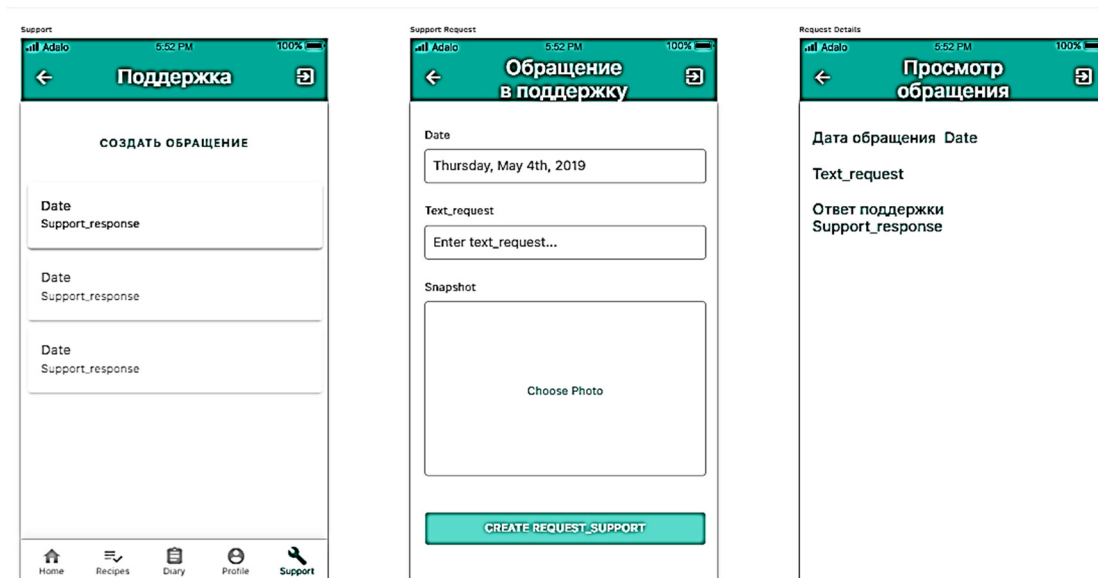


Рис. 8.11. Интерфейсная группа: Связь с поддержкой

9. Дополнительные материалы: создание чата

Для чата с поддержкой необходимо добавить в таблицу Users пользователя с Full Name support. Сделаем это обычным образом – через форму регистрации в приложении.

Также в базу данных необходимо добавить новые таблицы:

Таблицу Chatrooms:

- Name.

И таблицу Messages:

- Message_text.

Необходимо также добавить связи между таблицами. Для таблиц Users и Chatrooms – связь многие ко многим (более одного пользователя присоединяются к чату, и чатов у пользователей тоже может быть более одного). Users необходимо связать с Messages, где у сообщения может быть только один пользователь, но пользователю может принадлежать множество сообщений. Chatrooms связываем с Messages, где отдельные сообщения могут быть только в одном чате, но чат может состоять из множества сообщений.

Теперь добавим компонент Lists>Custom List на экран Support, который мы настроим на единственную запись в таблице Users со значением в поле Full Name, равным support (рис. 9.1).

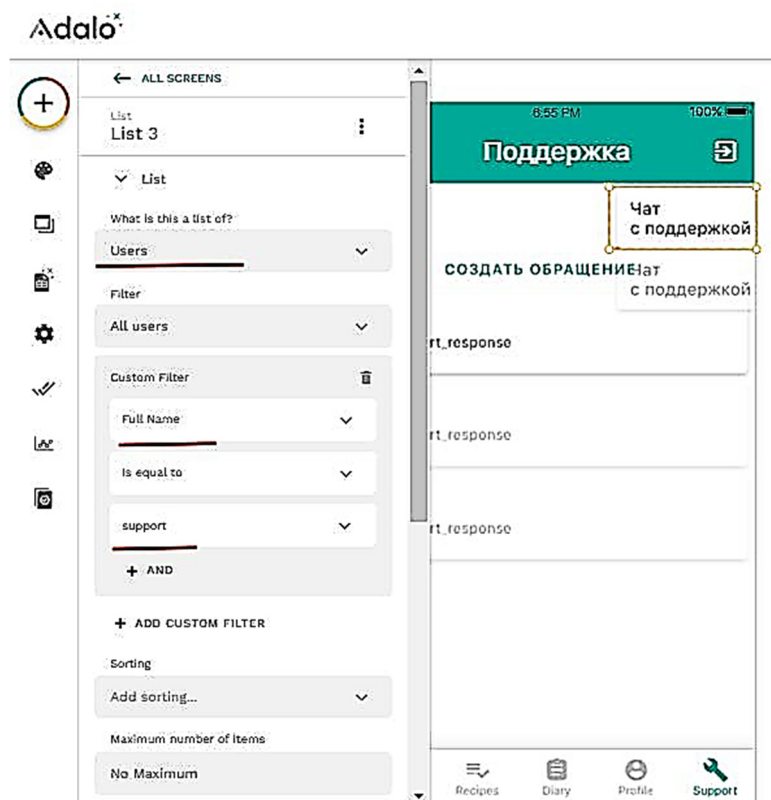


Рис. 9.1. Настройка списка с единственным значением в нем

Это необходимо, чтобы чат создавался только с текущим (и единственным благодаря фильтру) пользователем списка, которым является служба поддержки. Также добавим действие списку – переход на новый экран Chats. На этот экран добавим иконку, дающую возможность начать новый чат. Для этой иконки последовательно добавим четыре действия. Первое действие – Create>Chatroom, для нового чата необходимо задать обязательное поле Name, и это будет имя пользователя (рис. 9.2), с которым организуется чат (Current User – это единственный пользователь списка – support).

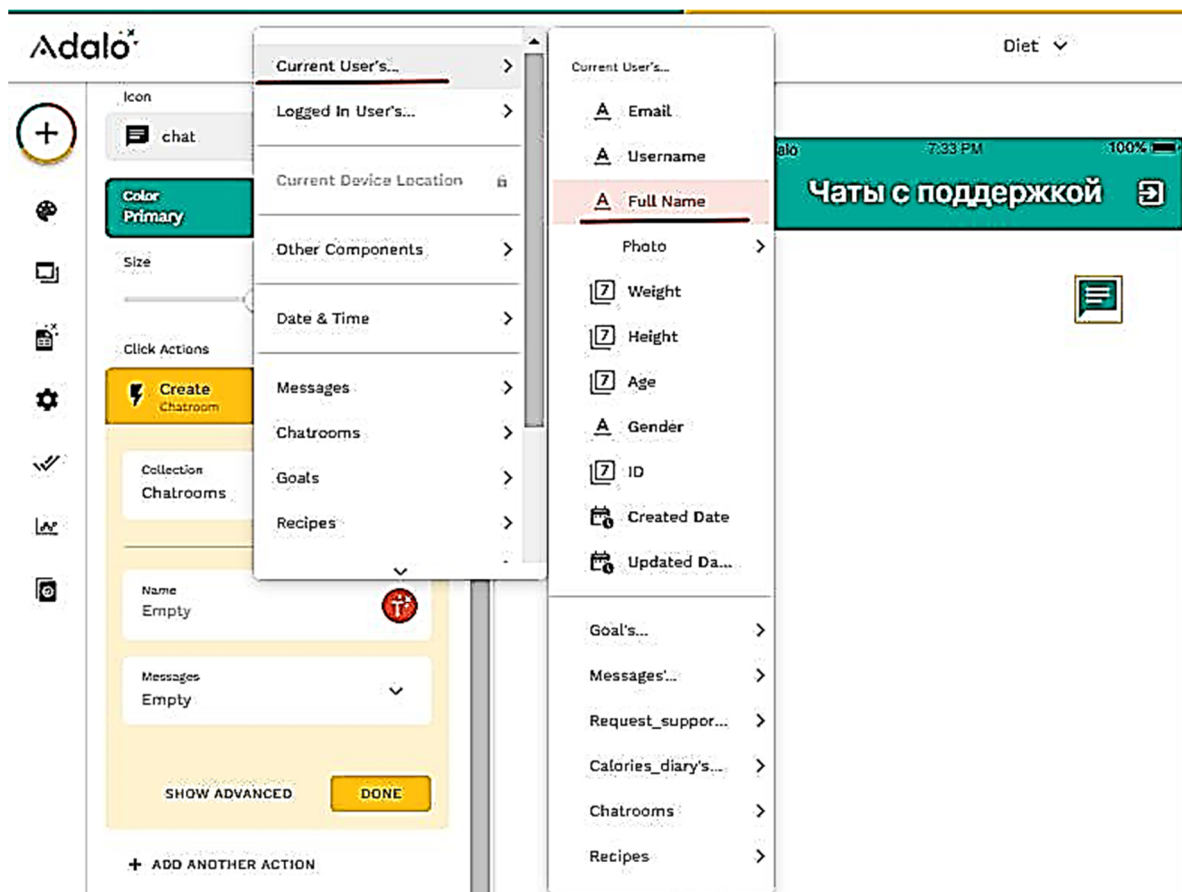


Рис. 9.2. Настройка действия Create

Во всех последующих действиях с Chatroom настройка поля Name – аналогичная.

Второе действие – Update>Chatroom (рис. 9.3), и здесь добавляется пользователь к чату (Current User, т.е. support).

Далее аналогично настраиваем действие Update>Chatroom (рис. 9.4), только добавляется пользователь к чату (Logged In User, т.е. авторизованный в данный момент пользователь).

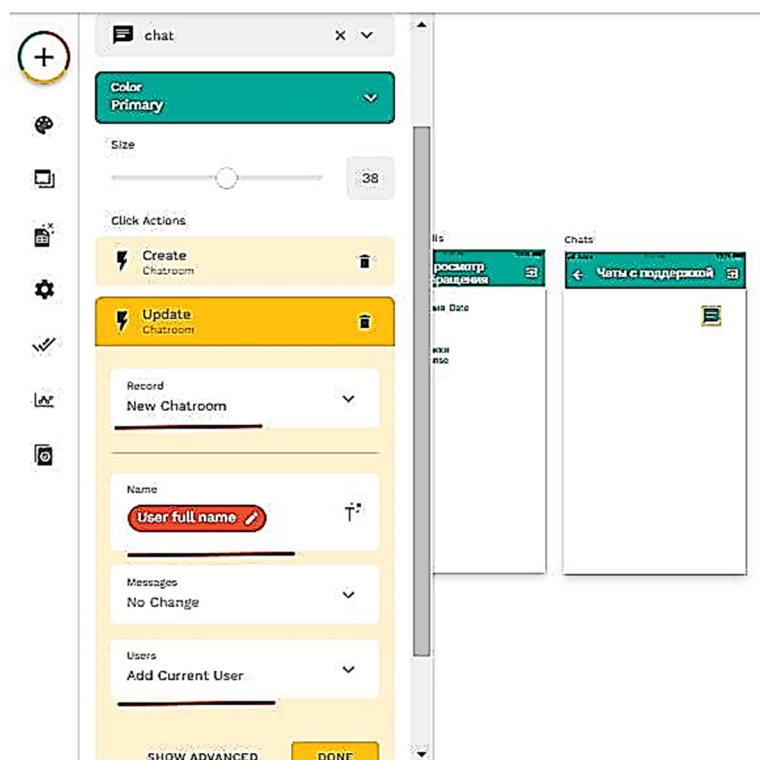


Рис. 9.3. Настройка действия Update для Current User

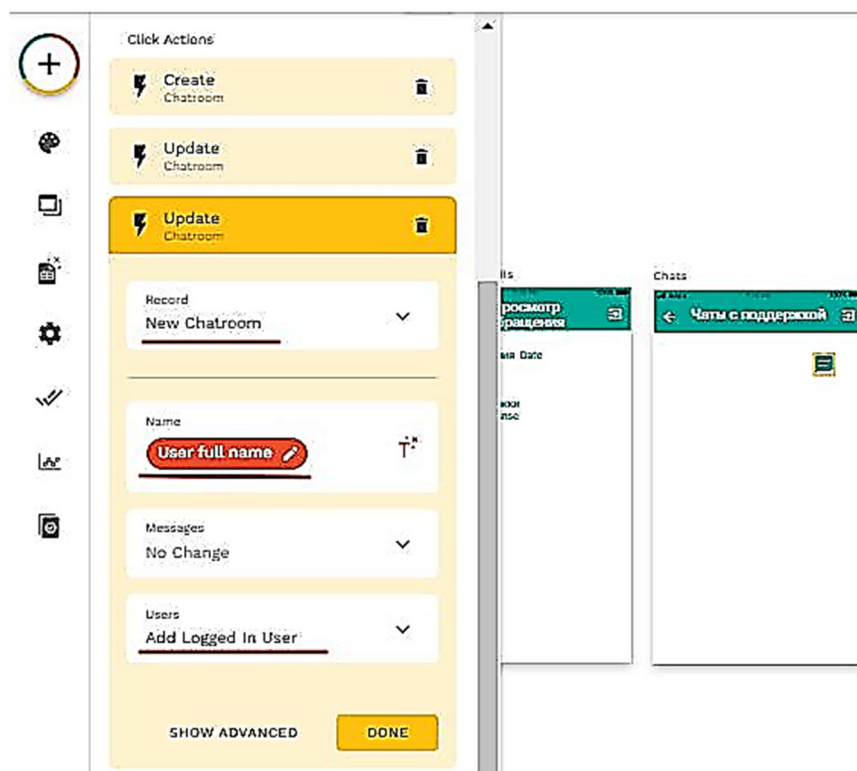


Рис. 9.4. Настройка действия Update для Logged In User

И последнее действие – переход на новый экран Chat. Однако вначале создадим этот экран (рис. 9.5).

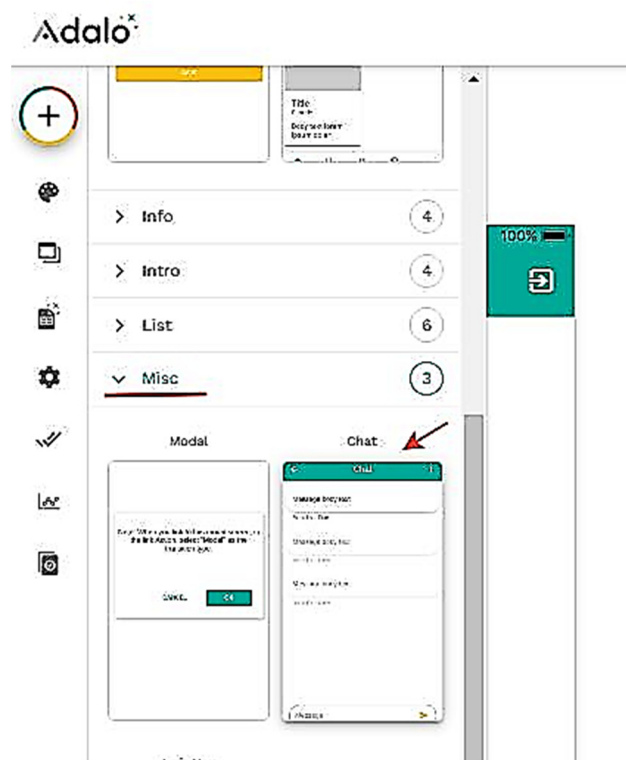


Рис. 9.5. Создание экрана Chat

После создания экрана добавим последнее действие кнопке (рис. 9.6).

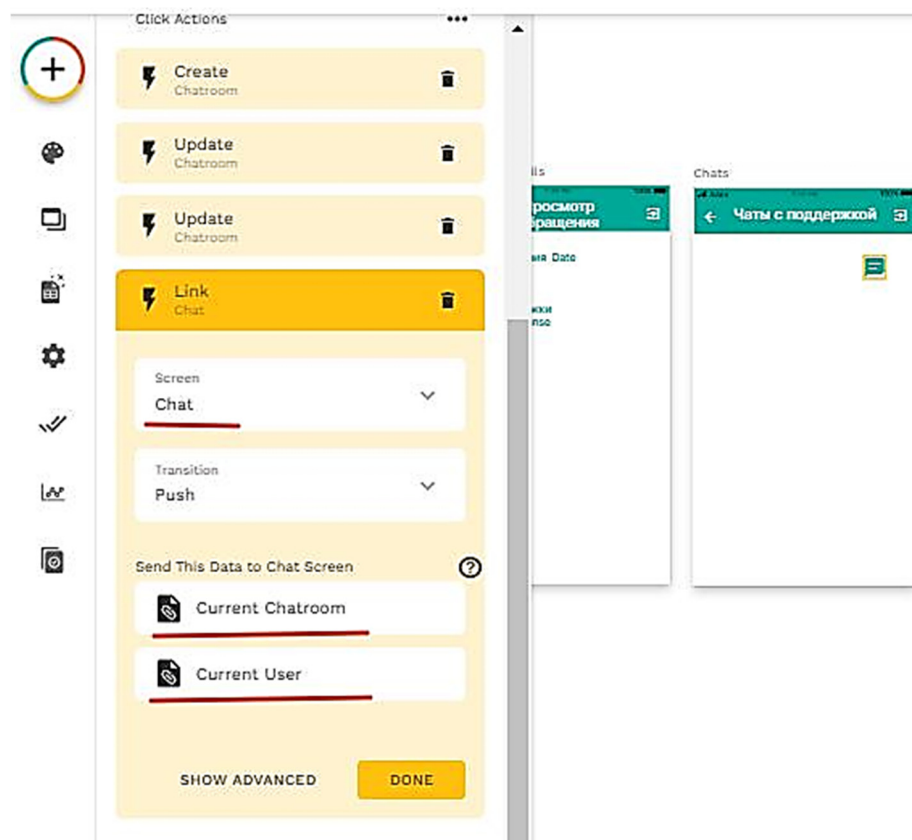


Рис. 9.6. Настройка перехода на экран Chat

Теперь перейдем к настройкам списка сообщений (рис. 9.7). Важно настроить опцию Auto Refresh, чтобы изменения в чате подхватывались на лету, а не требовали обновления экрана.

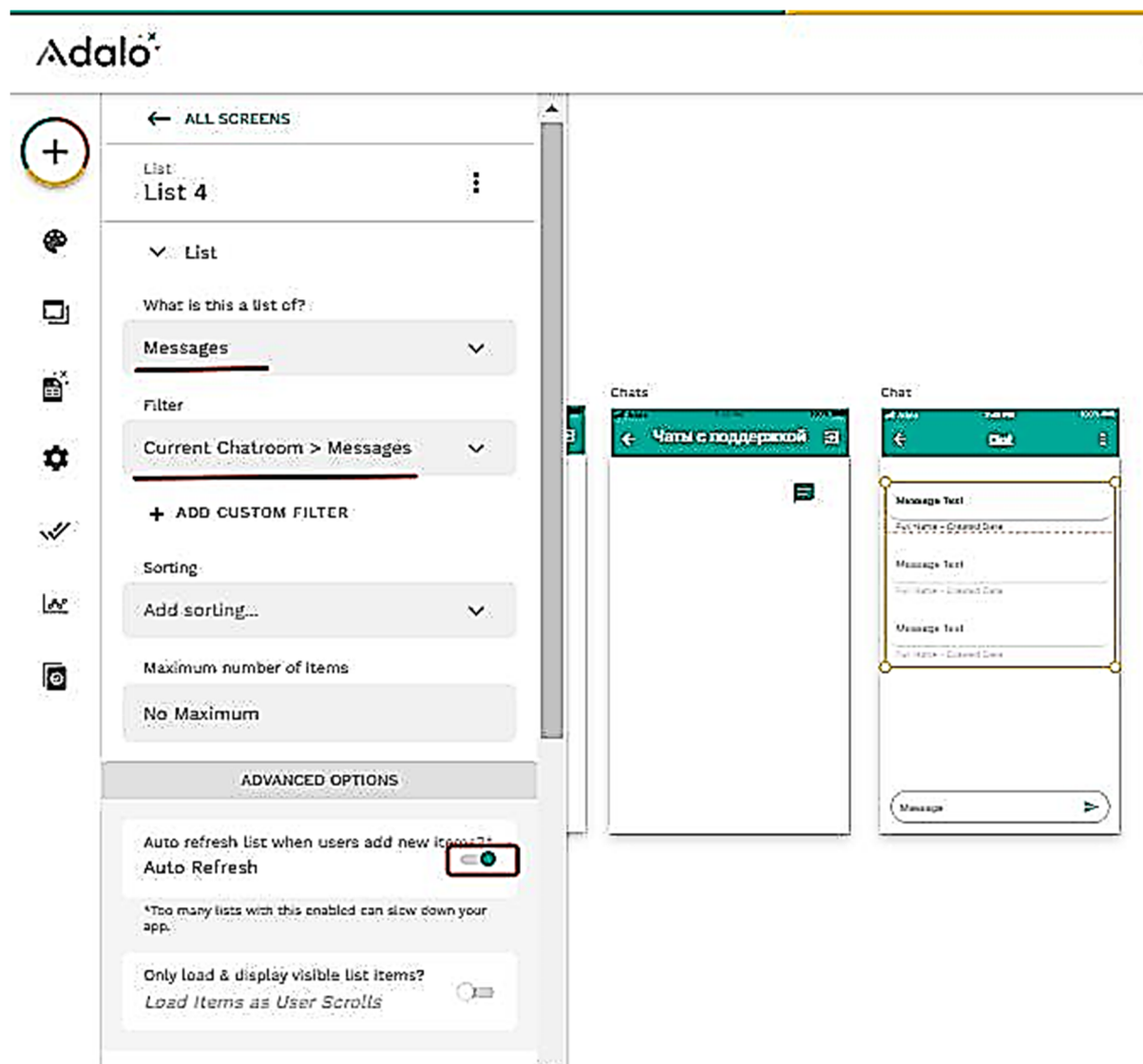


Рис. 9.7. Настройка списка сообщений

Проведем настройку одного сообщения списка и объединим три элемента сообщения (сам текст сообщения, дату создания и прямоугольник) в группу, которую назовем Message_text (рис. 9.8).

Скопируем эту группу ниже и разместим ее правее, это будут сообщения второго собеседника (рис. 9.9).

Изменим цвет прямоугольника для первого сообщения в списке на брендовый, а цвет текста сообщения на белый (чтобы сообщения от разных собеседников выделялись еще и цветом, а не только расположением относительно центра экрана), настроим для этих сообщений область видимости (рис. 9.10).

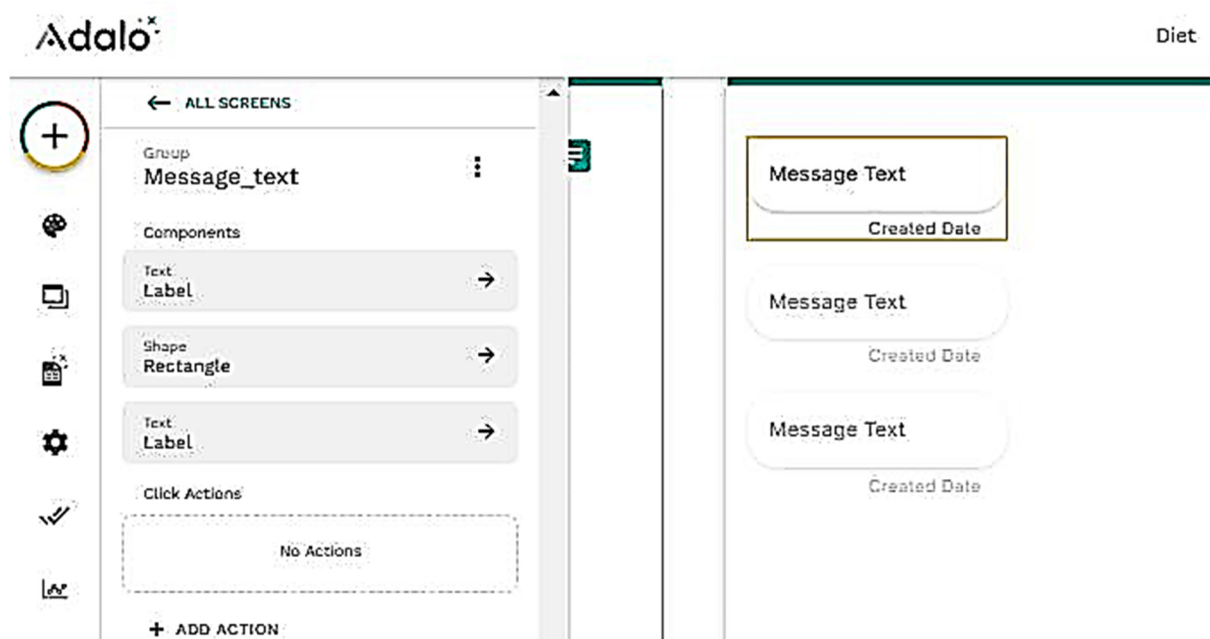


Рис. 9.8. Объединение элементов в группу

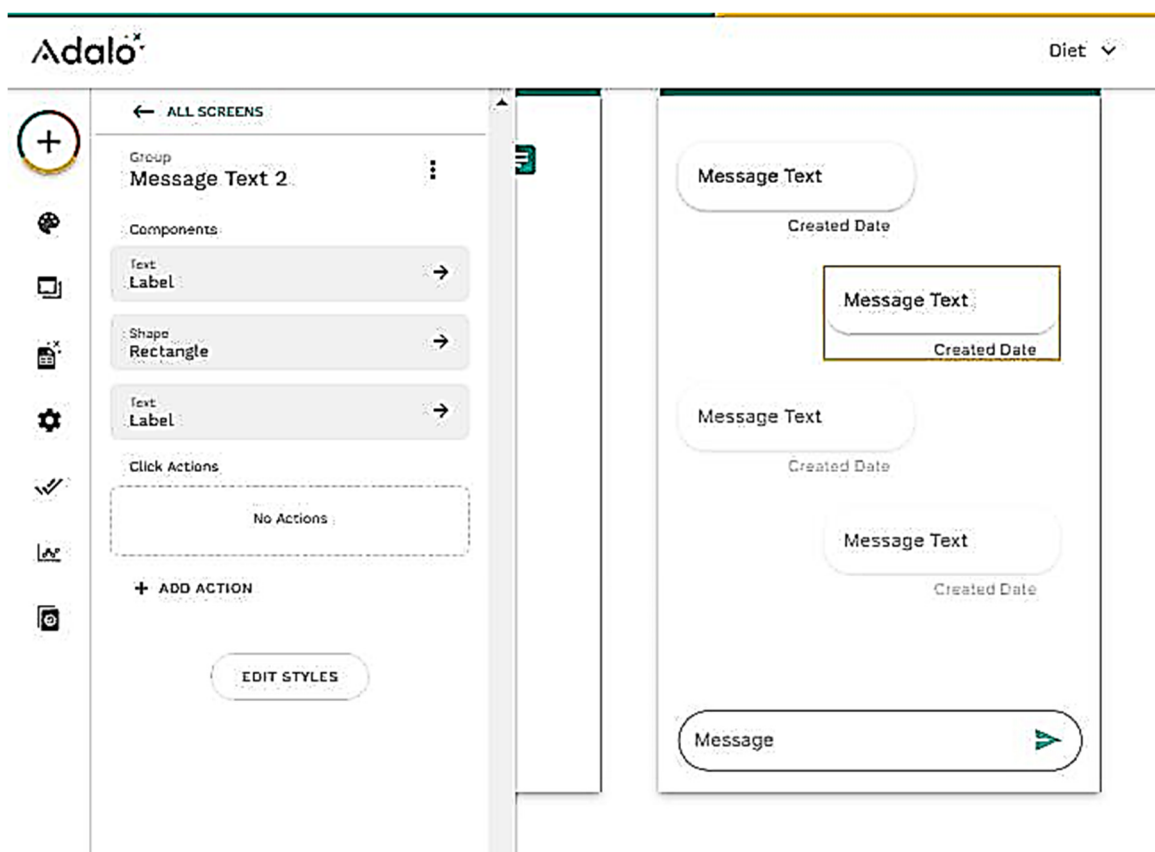


Рис. 9.9. Создание списка сообщений второго собеседника

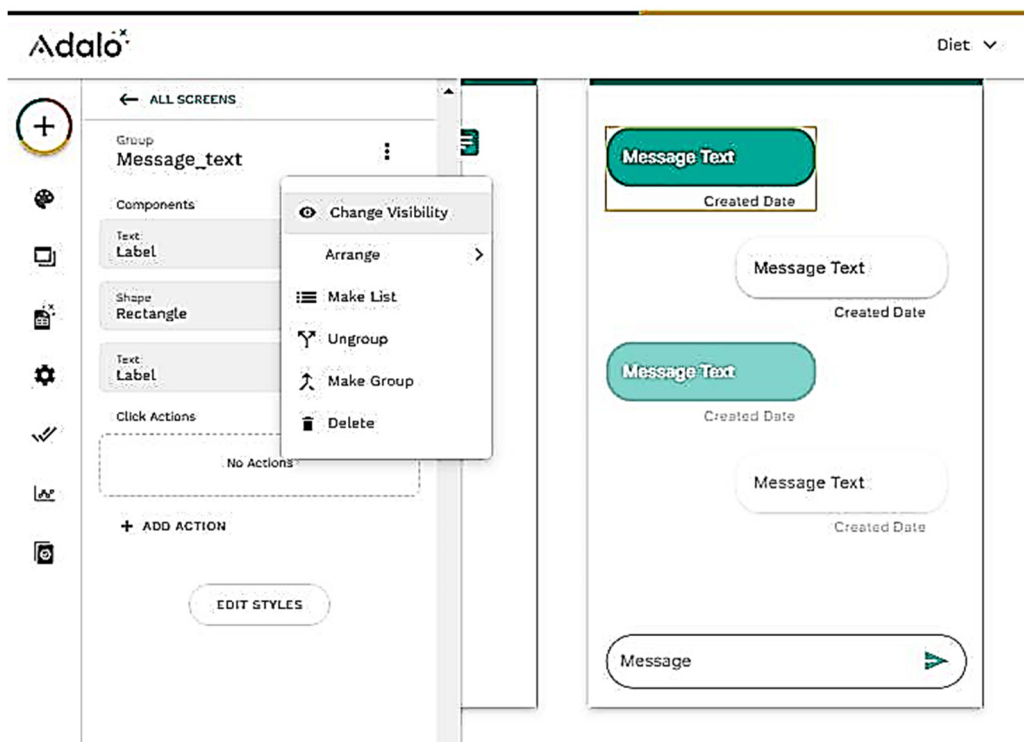


Рис. 9.10. Добавление настроек области видимости

В сообщениях поддержки мы настраиваем область видимости для поддержки (т.е. если email не равен авторизованному пользователю) (рис. 9.11). Иначе при отправке сообщения оно будет отображаться дважды – как отправленное и как полученное.

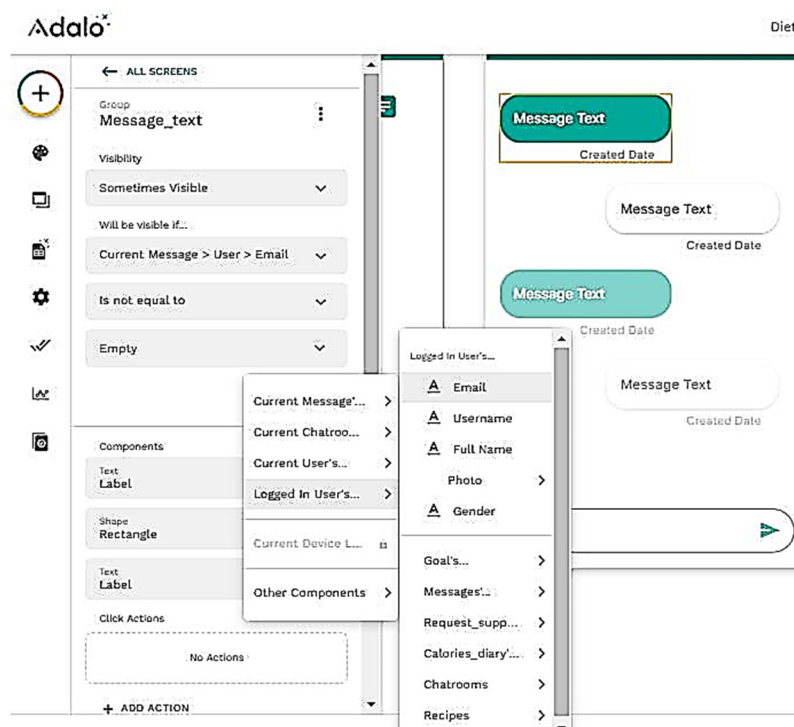


Рис. 9.11. Настройка области видимости для сообщений поддержки

А собственное сообщение, которое авторизованный пользователь набирает, имеет противоположные настройки видимости (т.е. если email равен авторизованному пользователю) (рис. 9.12).

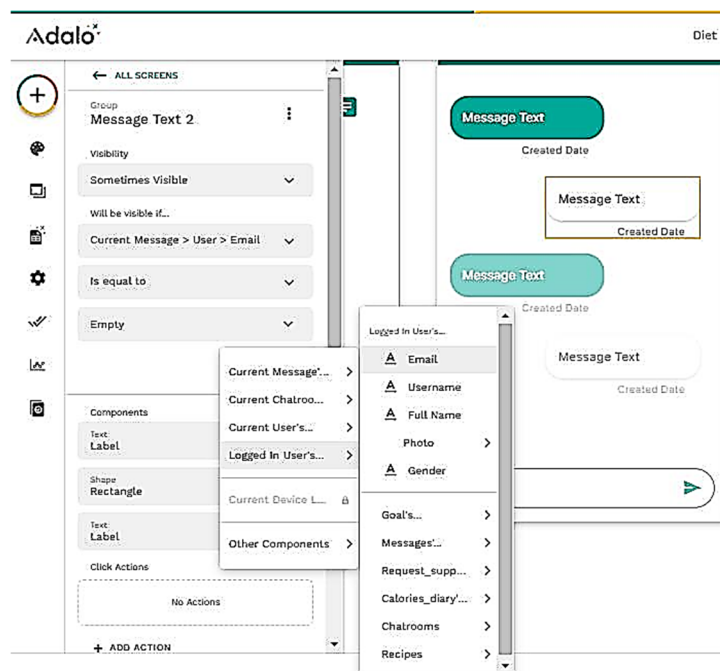


Рис. 9.12. Настройка области видимости для сообщений пользователя

Также настроим отправку сообщений в чат нажатием на кнопку «самолетик» в нижнем поле экрана. Настроим этой кнопке два действия. Первое – Create>Message (рис. 9.13, 9.14).

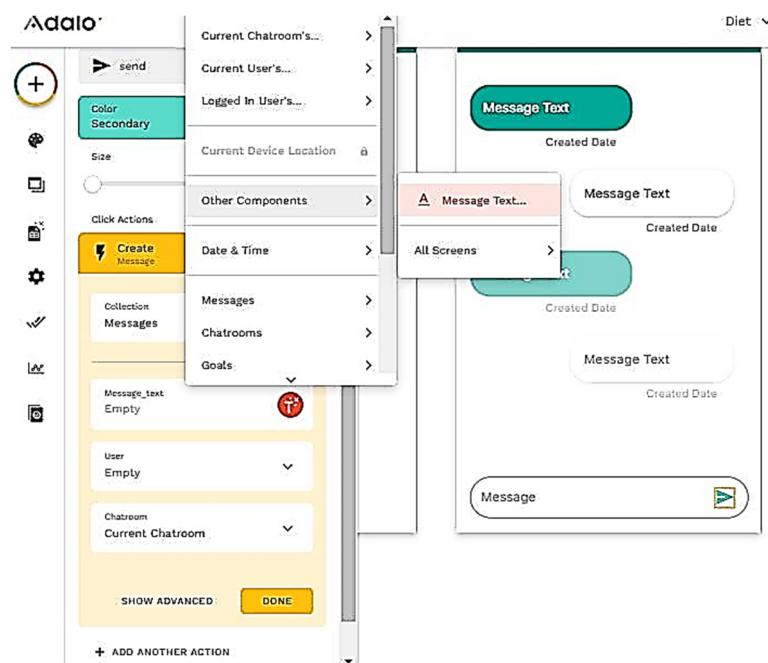


Рис. 9.13. Настройка значения поля Message_text

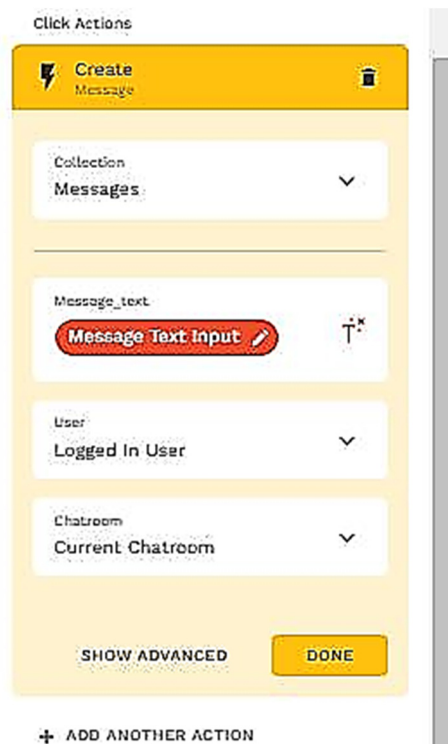


Рис. 9.14. Завершение настройки поля Message_text

Второе действие – Change Input Value (рис. 9.15), настройки для него показаны на рис. 9.16.

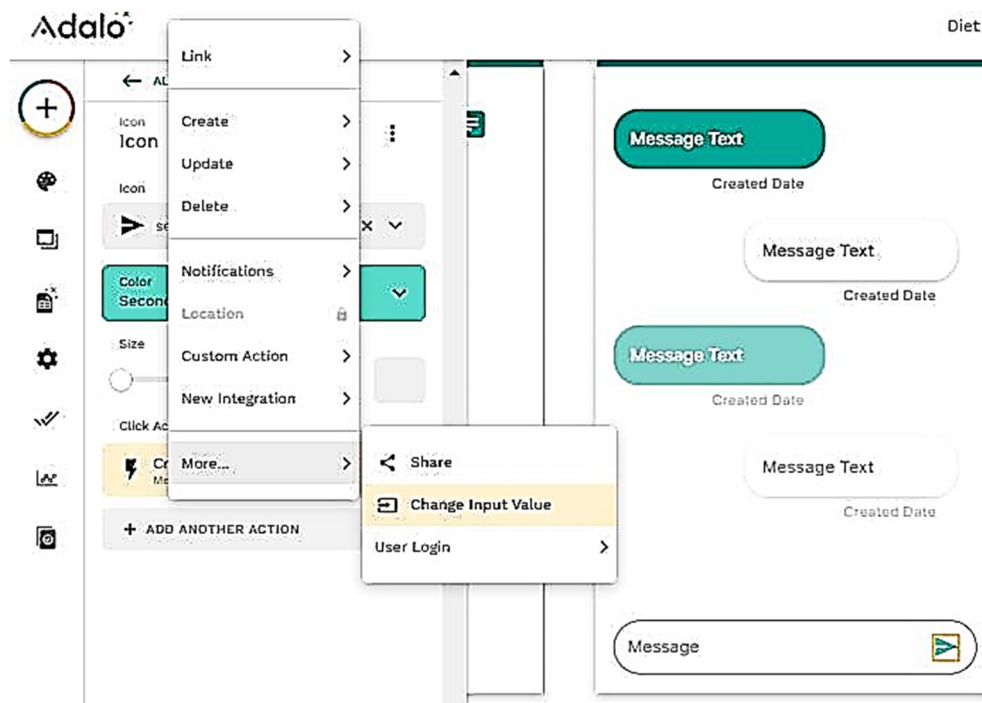


Рис. 9.15. Настройка обновления переменной для временного хранения сообщения Change Input Value

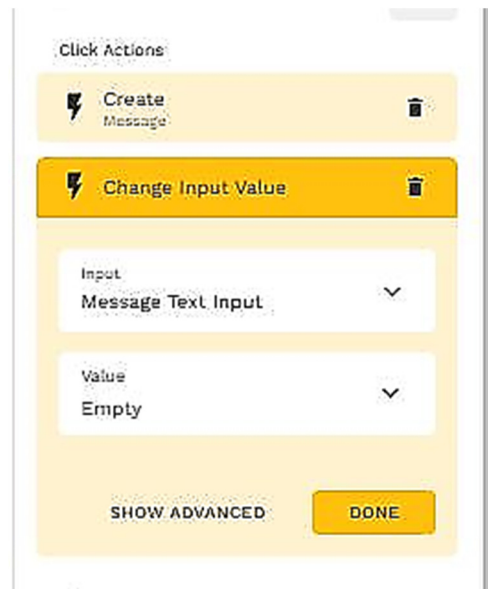


Рис. 9.16. Завершение настройки Change Input Value

Проверим корректность связи сообщений с таблицей данных для сообщений пользователя и сообщений поддержки (рис. 9.17). Добавим иконку в виде корзины к сообщениям авторизованного пользователя и настроим для нее действие – удаление текущего сообщения (рис. 9.18).

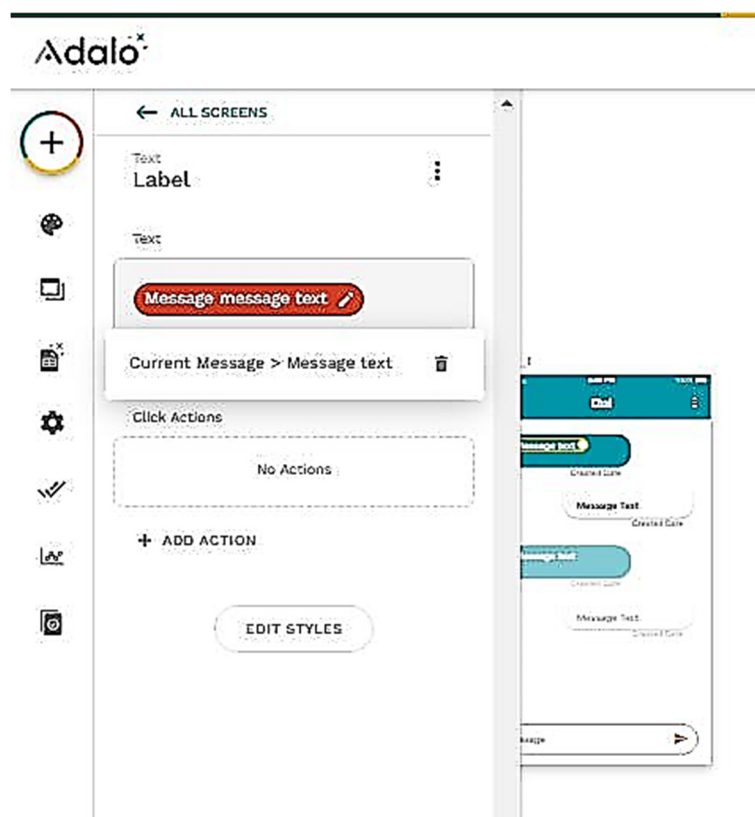


Рис. 9.17. Связь текста сообщения с полем в таблице Message

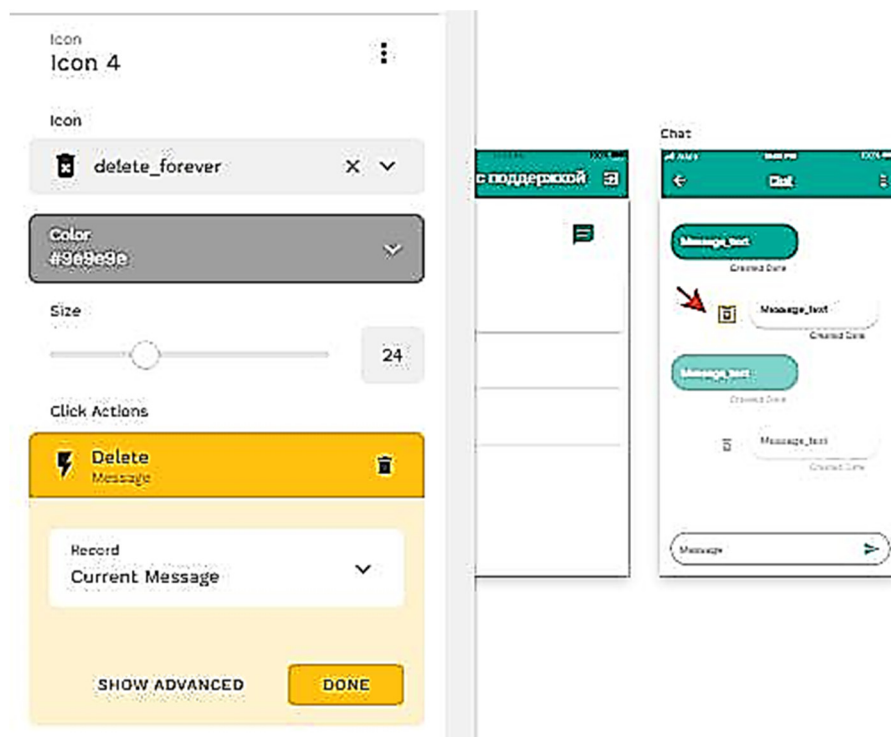


Рис. 9.18. Добавление значка корзины

Чтобы чат отображался корректно с новым элементом – корзиной, необходимо объединить группу этого сообщения и добавленную Icon в новую группу и настроить для новой группы область видимости как было показано на рис. 9.12. Также добавим список чатов Lists>Custom List на экран Chats (рис. 9.19). Это необходимо, чтобы пользователь Support мог присоединиться к созданному чату, а авторизованный пользователь мог вернуться в нужный чат.

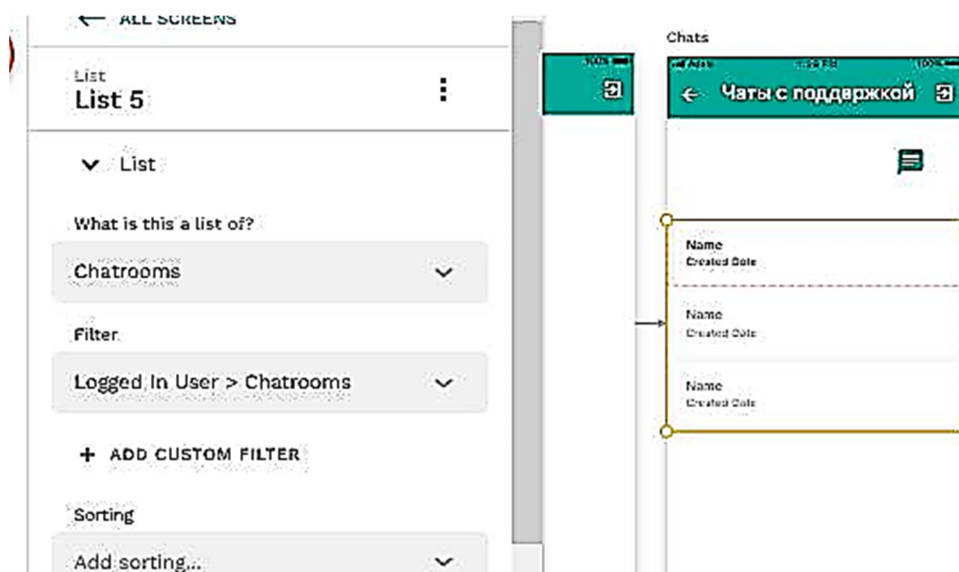


Рис. 9.19. Настройка списка чатов

Зайдем в двух разных браузерах от имени поддержки и от имени пользователя и протестируем работу чата (рис. 9.20).

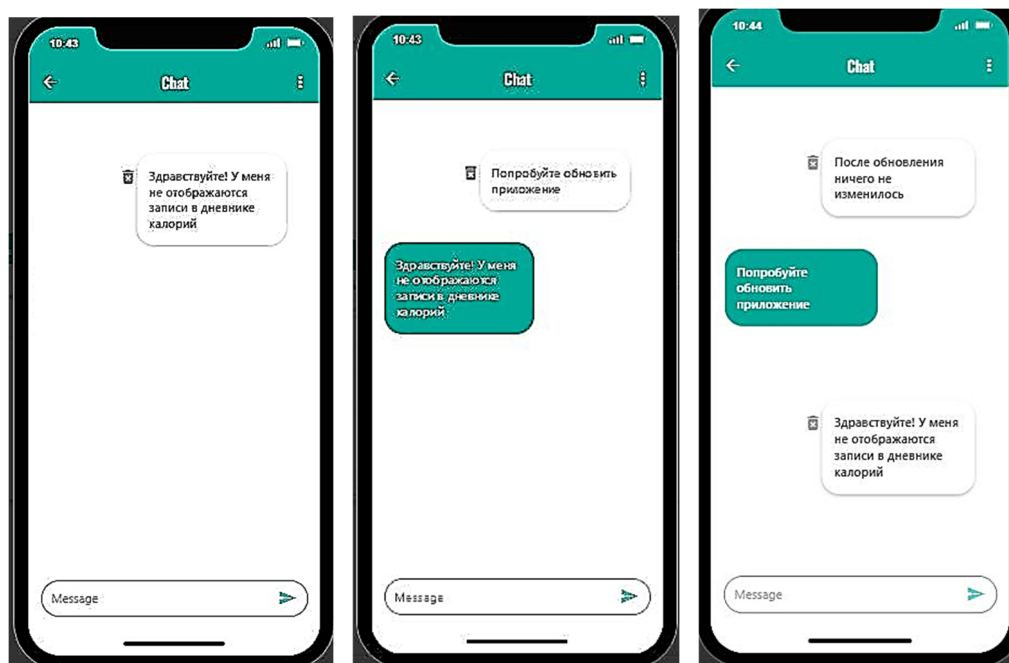


Рис. 9.20. Тестирование работы чата

10. Проект «Разработка мобильного приложения»

Цель работы: создать бизнес-приложение с использованием возможностей конструктора Adalo.

Формируемые знания, умения и навыки: научиться работать с инструментами no-code; знать, как от бизнес-идеи перейти к созданию цифрового продукта в виде мобильного приложения; уметь проектировать схему базы данных приложения, моделировать бизнес-процессы, которые обслуживает приложение, разрабатывать бизнес-логику и проектировать на ее основе экраны приложения с наполнением их компонентами и действиями.

Необходимо:

1. Предложить идею для бизнес-приложения, описать, какую проблему пользователя это приложение помогает решить, какую потребность закрывает.
2. Разработать схему базы данных приложения. Определить бизнес-логику приложения: карту путешествия пользователя, описать ее в виде последовательности экранов, продумать навигацию.
3. С использованием инструментов no-code создать мобильное приложение.
4. Провести тестирование приложения, исправить выявленные ошибки (например, в настройках компонентов, в навигации и т.п.).

Контрольные вопросы и задания

1. Из каких этапов состоит разработка приложения?
2. Из каких экранов состоит приложение? Какие компоненты добавлялись на экраны, какая функциональность за счет них обеспечивается в приложении?
3. Как организована навигация в приложении?
4. Какая схема базы данных у приложения?

Список литературы

1. Институт управления проектами. URL: <https://www.pmi.org/> (дата обращения: 27.12.2022).
2. UX User Persona. URL: <https://dribbble.com/shots/5846605-UX-User-Persona> (дата обращения: 27.12.2022).
3. ПП-салаты с курицей: 20 вкусных диетических рецептов. URL: <https://samors.ru/pp-salaty-s-kuricej-20-vkusnyh-dieticheskikh-receptov/?ysclid=lcjdgxl7hs498090709> (дата обращения: 27.12.2022).
4. Adalo Help. URL: <https://help.adalo.com/> (дата обращения: 27.12.2022).
5. Adalo Learn. URL: <https://www.adalo.com/learn> (дата обращения: 27.12.2022).

Учебное издание

Рындина Светлана Валентиновна

**Работа над проектом:
создание мобильных приложений**

Редактор *А. Г. Темникова*
Технический редактор *М. Б. Жучкова*
Компьютерная верстка *М. Б. Жучковой*

Подписано в печать 14.02.2023.
Формат 60×84¹/₁₆. Усл. печ. л. 3,49.
Тираж 5. Заказ № 17.

Издательство ПГУ.
440026, г. Пенза, ул. Красная, 40.
Тел.: (8412) 66-60-49, 66-67-77; e-mail: iic@pnzgu.ru

