

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное бюджетное
образовательное учреждение высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

С. В. Рындина

УПРАВЛЕНИЕ ЦИФРОВЫМ ПРОДУКТОМ: СОЗДАНИЕ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Учебно-методическое пособие

ПЕНЗА 2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

С. В. Рындина

Управление цифровым продуктом: создание мобильных приложений

Учебно-методическое пособие

Пенза
Издательство ПГУ
2022

УДК 004.738.5(075)+339(075)
Р95

Р е ц е н з е н т
кандидат технических наук,
доцент
А. А. Масленников

Рындина, Светлана Валентиновна.

Р95 Управление цифровым продуктом: создание мобильных приложений : учеб.-метод. пособие / С. В. Рындина. – Пенза : Изд-во ПГУ, 2022. – 92 с.

Рассмотрены платформы для создания мобильных приложений без использования программного кода. Материал соответствует программе дисциплины «Управление цифровым продуктом» и может быть использован при написании выпускной работы магистра.

Издание подготовлено на кафедре «Цифровая экономика» ПГУ и предназначено для обучающихся по магистерской программе «Цифровая экономика и бизнес-аналитика» направления 38.04.01 «Экономика».

УДК 004.738.5(075)+339(075)

© Пензенский государственный
университет, 2022

СОДЕРЖАНИЕ

Введение	4
1. Инструменты для проектирования и создания бизнес-приложений.....	5
2. Визуальный конструктор для создания мобильных приложений: сервис Glide	11
3. Визуальный конструктор для создания мобильных приложений: платформа Adalo	18
4. Разработка мобильного приложения для танцевально-спортивного клуба «Звездопад»	31
5. Визуальный конструктор для создания веб-приложений: Bubble	67
6. Лабораторная работа «Разработка мобильного приложения».....	88
Список литературы	90
Приложение. Настроенные экраны	91

Введение

В соответствии с учебным планом обучающиеся на втором курсе магистерской программы «Цифровая экономика и бизнес-аналитика» изучают дисциплину «Управление цифровым продуктом».

Целью учебно-методического пособия является знакомство студентов с no-code- и low-code-конструкторами для создания мобильных приложений, а также с процессами обслуживания и поддержки таких приложений, включающими: создание приложений, тестирование приложений, доработку по результатам анализа обратной связи.

В пособии рассматриваются возможности трех конструкторов для создания мобильных приложений без кода: Glide, Adalo и Bubble.

Благодаря материалам о танцевально-спортивном клубе «Звездопад», предоставленным одним из тренеров клуба Толмачевой Валерией, у автора пособия появилась возможность создать демонстрационный вариант мобильного приложения, не используя фейковые данные, стоковые фотографии и видео. Автор пособия выражает признательность Валерии Толмачевой, а также всем участникам клуба за предоставленные материалы.

1. Инструменты для проектирования и создания бизнес-приложений

Если вы разрабатываете приложение, выполняющее всего одну задачу, сценария будет достаточно, чтобы начать делать дизайн и даже разработку, а еще – заранее оценить, где пользователю будет неудобно, как он себя поведет на каждом этапе, что ему не пригодится.

Сценарий может быть оформлен в виде диалога, чтобы смоделировать сложность шагов/этапов взаимодействия, риски каждого шага, оценить протяженность взаимодействия пользователя с приложением во времени.

Обычно до начала разработки бизнес-приложения создается техническое задание (ТЗ). Главная задача, которую выполняет ТЗ, – описание проекта.

В самом базовом варианте техническое задание на бизнес-приложение может включать в себя следующие разделы:

- общие сведения: цель создания мобильного приложения, совместимость с ОС, масштабируемость (способность адаптироваться к изменению условий), возможность выполнения функций при отказе одного компонента или нескольких;
- сценарии использования: возможности неавторизованных и зарегистрированных пользователей (у зарегистрированных пользователей могут быть реализованы различные ролевые модели), функциональные схемы, моделирующие объединение простых функций приложения в группы, обмен данными, поддержка работы с другими приложениями (платежные, почтовые сервисы), администрирование ПО;
- список нефункциональных требований: безопасность, возможность сохранения отчетов об ошибках, производительность;
- проектирование функционала: экран загрузки, порядок регистрации (авторизации), меню, способы поиска нужной информации, уведомления для пользователей.

При создании ТЗ можно использовать шаблоны документов: Видение, Роли-кейсы, ТЗ, Описание прототипов. Каждый документ добавляет в описание будущего бизнес-приложения свой ракурс рассмотрения и уточняет требования к нему со стороны разных заинтересованных лиц.

Если необходимо запустить MVP (minimum viable product, минимально жизнеспособный продукт), то слишком подробное, деталь-

ное ТЗ нежелательно, так как оно все равно будет уточняться, дорабатываться в ходе эксплуатации MVP. В этом случае предпочтительны самые базовые варианты всех документов и дальнейшая динамичная коррекция их содержания по мере понимания того, как на продукт реагирует рынок.

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, но вы с ним не взаимодействуете, а работаете только с визуальными элементами. Подобными платформами может поддерживаться расширение функциональности создаваемых приложений с использованием 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-логику 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 или 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 другие, 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. Для разработки под Apple актуальны технологии: UIKit, iOS SDK, язык программирования Swift. Для Андроид – это Android SDK, языки программирования Java 8, Kotlin.

Более сложный и затратный в разработке вариант – кросс-платформенные приложения: гибридные, компилируемые в нативный код.

В пособии рассмотрим три платформы для быстрого создания мобильных приложений.

Glide – платформа для создания мобильных приложений без использования программного кода. В Glide приложения разрабатываются с помощью визуального интерфейса, объединенного с логикой электронных таблиц. Гибкость этого инструмента при создании мобильных приложений минимальная, но он максимально прост в освоении. Позволяет создать мобильные приложения PWA.

Adalo – конструктор мобильных приложений, которые можно публиковать в App Store и Google Play. 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.

Bubble – это платформа для создания веб-приложений, не требующая навыков программирования. Bubble – это инструмент all-in-one (все в одном). В нем есть визуальный редактор, базы данных, инструменты для реализации бизнес-логики и для работы с разными API. Bubble позволяет создавать полнофункциональные чаты, форумы, системы сбора и обработки заявок, таск-трекеры, маркетплейсы, CRM и дашборды. Bubble позволяет создавать адаптивные веб-приложения, которые подстраиваются под разрешения смартфонов. Мобильные приложения на Bubble можно публиковать в Apple App Store и Google Play Store с помощью специальных контейнеров.

Даже используя low-code- и no-code-инструменты разработки приложений, необходимо документировать процесс разработки. Причем чем сложнее инструмент, тем больший объем документирования должен поддерживаться.

Для Glide необходимо документировать список экранов, связи таблиц.

Для Adalo и Bubble документирование должно быть более подробным:

- общая архитектура решения, с какими сервисами будет интегрировано приложение;
- карта приложения с иерархической взаимосвязью всех экранов;
- база данных: сущности и их атрибуты (таблицы и описание полей в них), ключевые связи между сущностями;
- правила конфиденциальности;
- список задач (объединенных в user story или этапы разработки), которые выполнялись в процессе разработки приложения.

Все эти разделы динамически пополняются по мере продвижения к конечному этапу создания продукта. Основная сила инструментов no-code и low-code именно в скорости вывода продукта на рынок (в сокращении time to market) и быстром внесении изменений по результатам анализа обратной связи от клиентов: отзывов, обращений в службу поддержки, проведенных глубинных интервью. Поэтому копии созданных документов могут быть присоединены к новой версии продукта и модифицированы в соответствии с проведенными доработками продукта.

Использование инструментов no-code и low-code может быть временным решением. На малых масштабах или в начале проекта это позволяет рационально распределить ресурсы и выполнить необходимые исследования достаточно дешево.

Для цифровых продуктов на стадии идеи есть проблема, которая описывается как «Уловка-22» (т.е. ситуация, возникающая в результате логического парадокса между взаимоисключающими правилами и процедурами): для привлечения венчурных инвестиций желательно предъявить уже работающий продукт с доказанным Problem Solution Fit и подтвержденным Product Market Fit, а на это нужны деньги, для получения которых и привлекаются к перспективному проекту венчурные фонды и инвесторы.

Problem Solution Fit – подтверждение существования проблемы и того, что предлагаемое решение ей соответствует.

Product Market Fit – соответствие продукта ожиданиям целевой аудитории, подтверждение, что потребители готовы покупать продукт.

Так как тарифы у no-code- и low-code-инструментов зависят от количества активных пользователей (или проведенных транзакций), то при росте масштабов проекта оказывается выгоднее разработать клон своего проекта в коде. Но это та стадия развития цифрового сервиса, когда он уже зарабатывает, а не тратит, и речь идет об оптимизации процессов и сокращении издержек в успешном и прибыльном решении.

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.

Необходимо отметить, что заказная разработка в коде не преодолевает все указанные ограничения целиком и полностью: из-за

длительного цикла коммуникаций, согласований и связанных с изменениями дополнительных расходов не всегда можно получить идеальный результат, здесь также придется идти на определенные компромиссы по дизайну и функционалу, хостинг проекту будет предоставляться также при соблюдении определенных правил и норм, также нет гарантий, что сохранение данных и их конфиденциальность будут соответствовать самым высоким стандартам.

Рассмотрим, кому и для каких задач могут пригодиться no-code-и low-code-инструменты [1]:

- стартаперам: быстрое тестирование гипотез, быстрое тестирование новых продуктов;

- предпринимателям: создание бизнес-приложений для растущего и успешного оффлайн бизнеса, с помощью no-code-инструментов. Такие приложения легче поддерживать, интегрировать с рекламой, различной аналитикой, их можно протестировать на уже имеющейся клиентской базе. По отклику станет понятно, нужна ли внутренняя компания разработки или можно продолжать использовать построенное приложение;

- большим компаниям: разработка внутренних продуктов, приложений для сотрудников (например, трекеры задач для сотрудников, чек-листы для мониторинга, дашборды по аналитике и т.п.);

- дизайнерам: расширение своей роли в разработке приложений, не только прототипирование интерфейсов, но и создание готового продукта;

- менеджерам продуктов: развитие и улучшение тех навыков и компетенций, которые у конкретного product owner слабоваты. Pet-проекты (т.е. проекты, которые делаются для себя, в свободное от основной работы время), реализуемые менеджерами продуктов с помощью no-code-инструментов, выступают как тренажеры. Те, кто переходит в менеджмент из разработки, лучше понимают маркетинговую составляющую продукта на таком тренажере, а те, кто приходит в менеджмент из дизайна, продаж, продвижения – техническую сторону, т.е. бизнес-логику, архитектуру приложения. Такой проект для тренировки имеет потенциал и может развиваться в стартап и вполне успешный продукт.

2. Визуальный конструктор для создания мобильных приложений: сервис Glide

Glide (<https://www.glideapps.com/>) – платформа для создания мобильных приложений без использования программного кода. Сервис Glide разработали и запустили в 2018 г. бывшие сотрудники компании Microsoft Дэвид Сигел и Джейсон Смит.

Преимущество Glide, кроме низкого порога вхождения, – наличие бесплатного плана использования (версия Personal), который включает:

- до 100 мегабайт хранилища;
- до 500 строк информации;
- до 1000 изменений таблицы в течение месяца.

В интерфейсе создаваемых приложений присутствует реклама Glide.

Платформа позволяет создать приложение с нуля или используя готовый шаблон. В Glide можно создать небольшой интернет-магазин, социальную сеть, чат, приложение для отслеживания задач и многое другое.

Приложение наполняется контентом на основе данных таблиц, созданных либо в Google Sheets, либо в самом сервисе. Одновременно таблицы выступают и в роли базы данных приложения.

Все приложения Glide – это прогрессивные веб-приложения (progressive web app, PWA). PWA работает через браузер, но его можно установить на домашний экран смартфона. Интерфейс PWA доступен даже без интернета, но для доступа к данным все-таки понадобится сеть.

Регистрация в Glide – это внесение email и переход по кнопке Get Started (рис. 2.1). В качестве email используем свой аккаунт @gmail.com, на который придет письмо для верификации адреса.

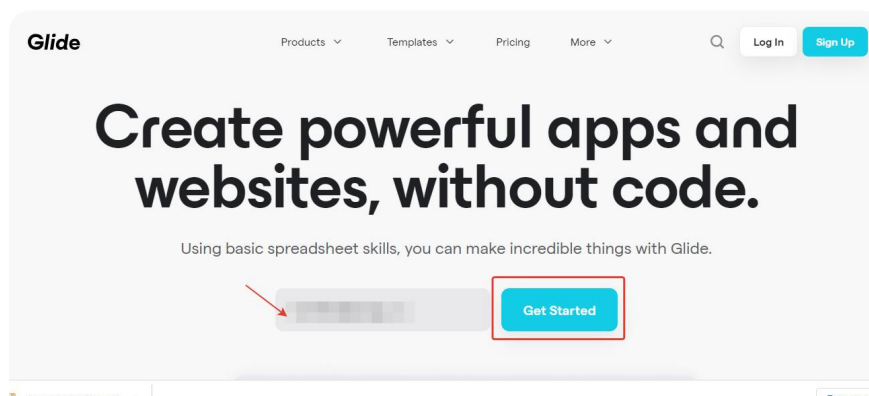


Рис. 2.1. Регистрация в сервисе Glide

Следующим будет окно входа в Glide, в котором опять выбираем аккаунт Google, т.е. кнопку Sign in with Google (рис. 2.2).

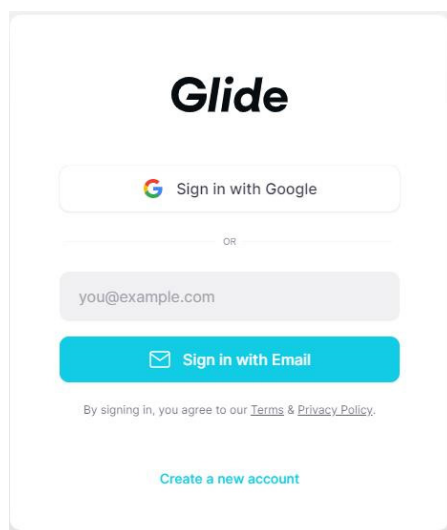


Рис. 2.2. Авторизация в сервисе Glide

После входа в приложение предлагается заполнить данные профиля. И можно приступить к созданию первого приложения [2]:

1. Создаем в файле Google Sheets (или в сервисе Glide) таблицы с данными приложения: каждый лист с таблицей – это отдельный экран. Таблица состоит из строки заголовков – название элементов экрана (свойств объектов), а значения в столбце – данные (рис. 2.3).

Рис. 2.3. Файл Google Sheets,
на основе которого создается приложение

2. В сервисе Glide создаем проект для разработки мобильного приложения, выбрав New project (рис. 2.4).

И, подключив (рис. 2.5, 2.6) ранее созданную таблицу (рис. 2.3), приступаем к настройке элементов вкладок.

3. Настраиваем вкладки приложения с помощью элемента Tabs (рис. 2.7).

Tabs – это сущности в таблице (Tab – отдельный лист). Tabs отображаются иконками внизу приложения и являются элементами

навигации. Их можно создавать прямо в приложении, дублировать, удалять, а также временно деактивировать, отключив видимость. Детально настраивать Tabs можно в рабочей области визуального редактора в правом окне. Можно задать название вкладки, указать конкретный источник данных в Google Sheets, присвоить иконку, задать стили в соответствии с назначением вкладки. Приложение отображается в виде эмуляции телефонного экрана и позволяет тестировать настроенные элементы.

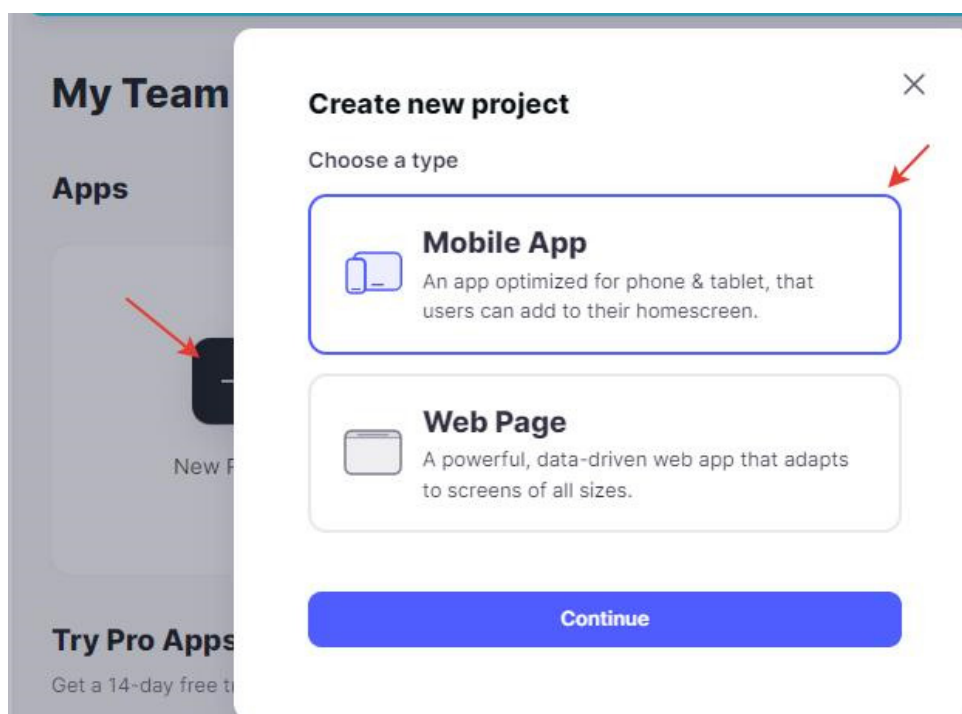


Рис. 2.4. Создание проекта

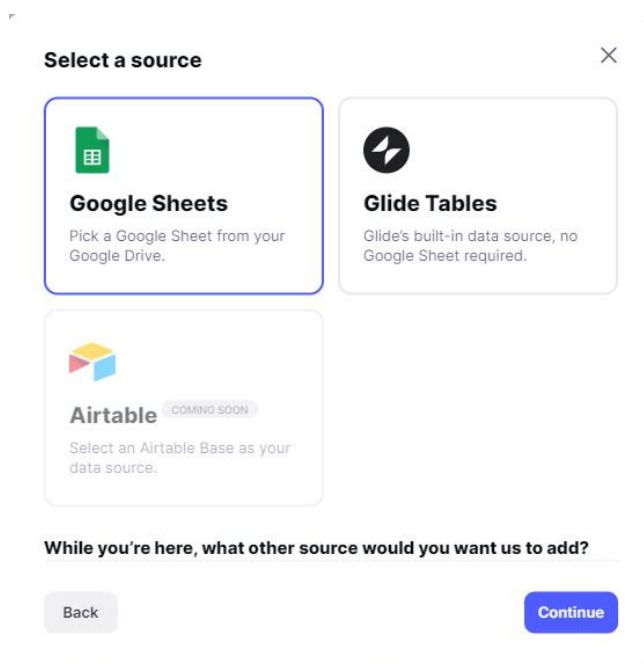


Рис. 2.5. Выбор типа источника данных для приложения

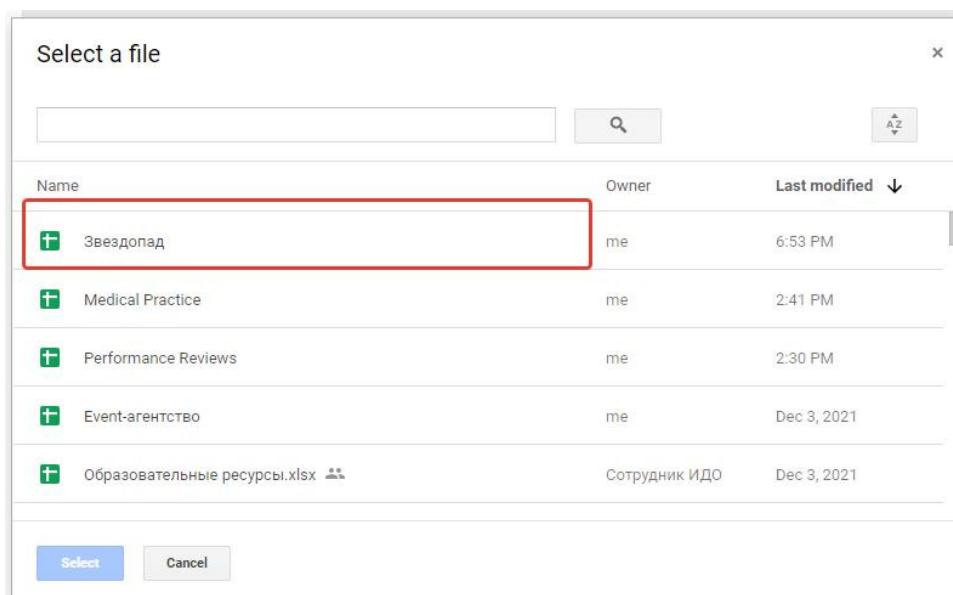


Рис. 2.6. Выбор файла с таблицами для приложения

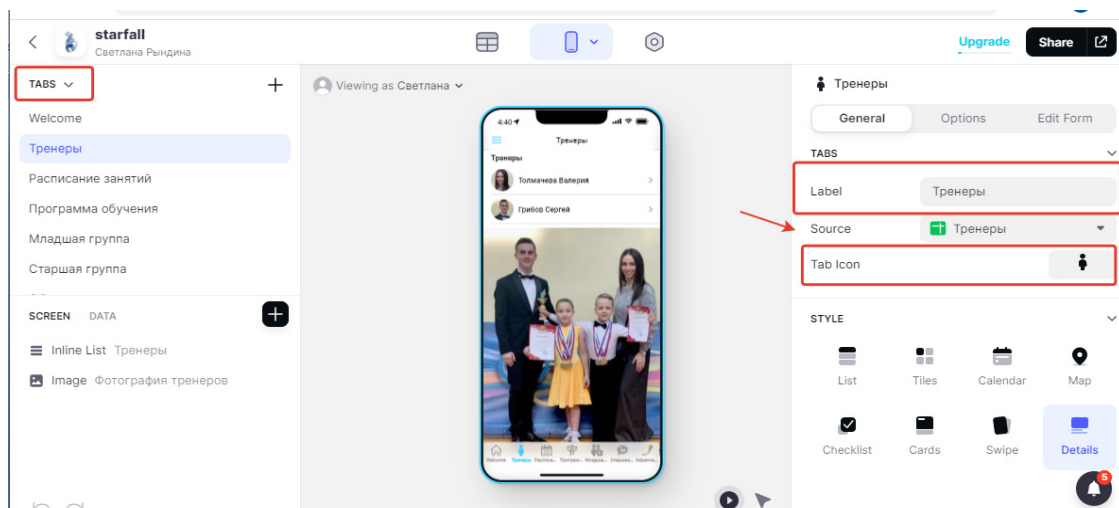


Рис. 2.7. Рабочая область визуального конструктора Glide

4. Для каждой вкладки настраиваем стиль. Он определяет не только внешний вид, но и функционал. Всего есть восемь стилей, точнее макетов (рис. 2.8):

- список (list) настраивает отображение данных страницы приложения в виде списка. Можно задать количество строк, изображений, изменить текстовые настройки;
- плитка (tiles) отображает редактируемую страницу в виде плитки. Можно настроить вид, размер, расстояние между плитками, добавить кнопки, теги и закладки и менять направление прокрутки плиток;
- календарь (calendar) подходит для отображения данных со свойствами даты и времени, например, таблицы событий, содержащей даты;

- карта (map) отображает адрес на странице в виде карты. Можно настроить два способа отображения: как статическое изображение на странице приложения или установить действие «просмотр адреса на карте», где карта открывается на устройстве;
- чек-лист (checklist) отображает макет страницы в виде контрольного списка с возможностью отмечать галочкой выполненные пункты;
- карточки (cards) представляют собой макет с одним или двумя столбцами, в котором элементы списка – это карточки с изображениями;
- свайп (swipe) – плавающие карточки, которые пользователи могут смахивать с экрана влево или вправо;
- детальная настройка (details) – макет для создания экрана с широким спектром компонентов для добавления.

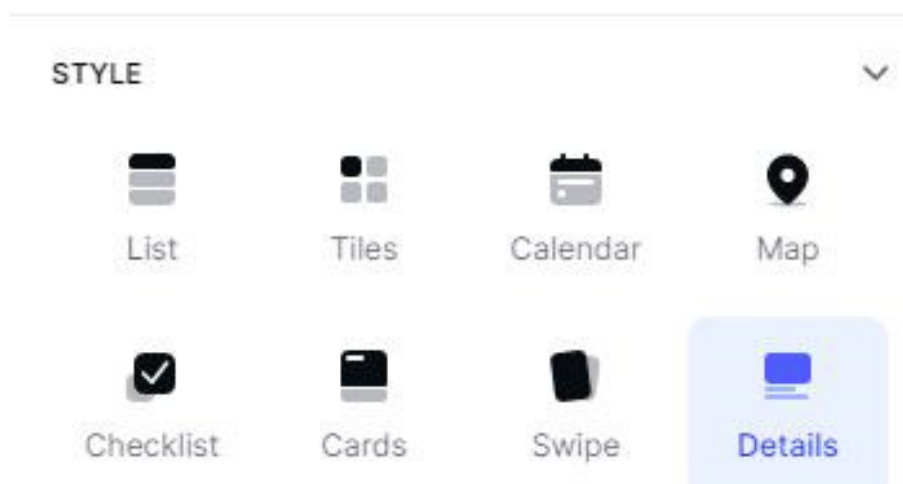


Рис. 2.8. Варианты стиля

Для добавления и настройки компонент в Glide есть следующие опции:

- Screen используется для детальной настройки, которая доступна в вариантах стиля Swipe и Details. Компоненты, которые можно добавить на редактируемый экран: текст, элементы дизайна, медиа-файлы, кнопки, списки, настраиваемые поля и др. (рис. 2.9);
- Data – опция, которая демонстрирует содержимое конкретной записи в таблице, а также позволяет добавлять новые позиции, функции, делать вычисления, задавать логику отношений в зависимости от заданных пользовательских сценариев.

5. Публикуем приложение. В бесплатной версии проект хостится на домене glideapp.io с QR-кодом для скачивания (рис. 2.10).

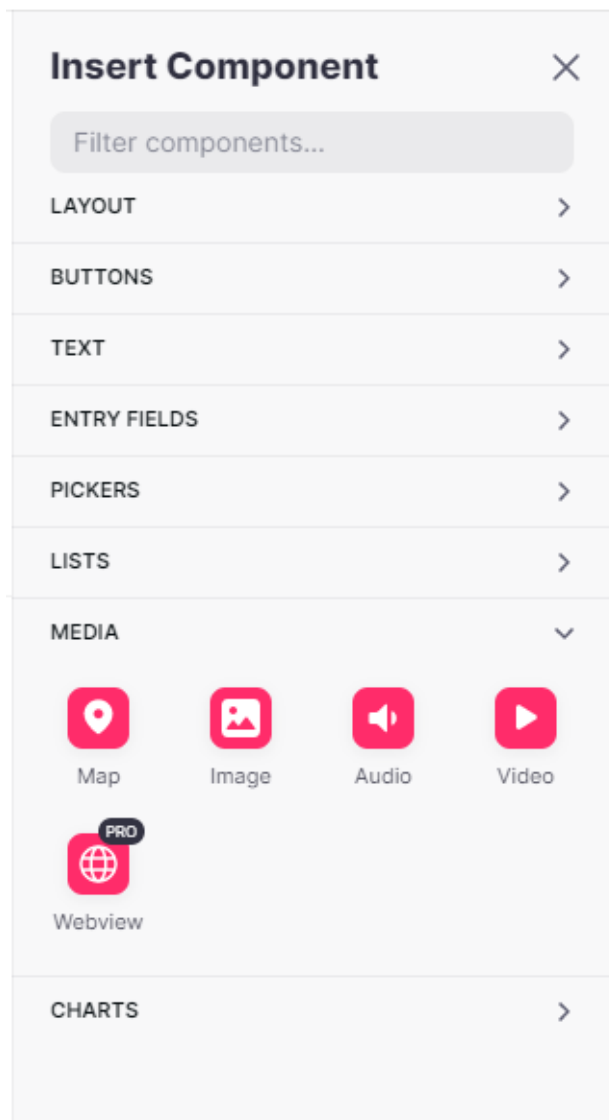


Рис. 2.9. Компоненты для добавления на экран

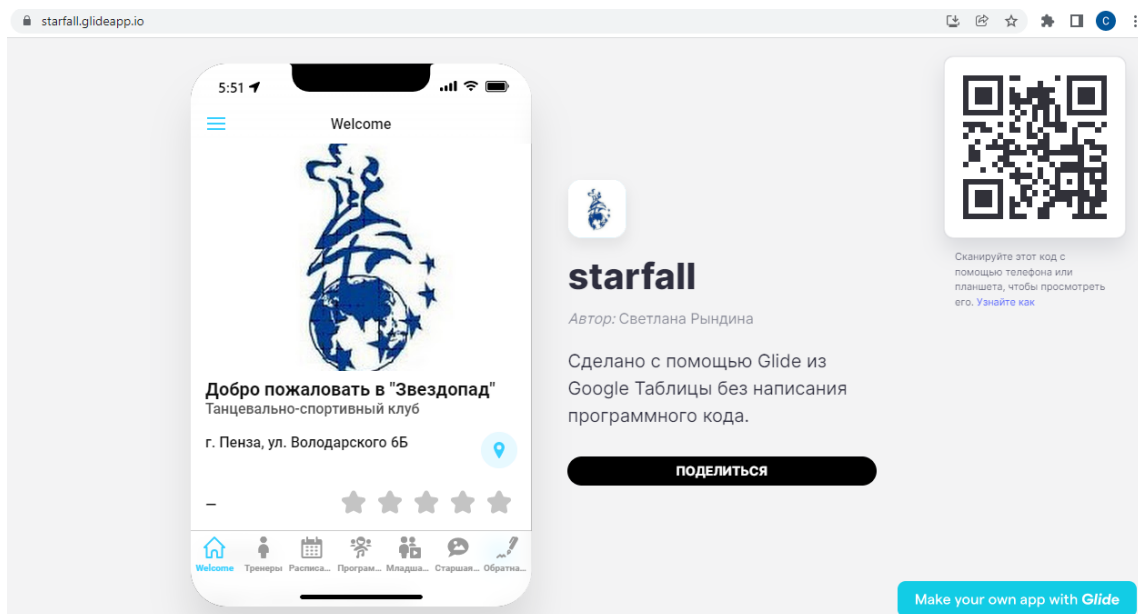


Рис. 2.10. Публикация приложения

При добавлении картинок и фотографий в приложение обратите внимание на их расширение.

Glide распознает файлы с расширением .jpg, .png, .gif, .svg.

Для значка приложения лучше использовать файл с расширением .svg. Значок приложения будет отображаться во многих местах и во многих размерах, а рисунок с расширением .svg будет масштабироваться и хорошо выглядеть при любом размере.

Более подробная информация о создании приложений в визуальном конструкторе Glide содержится в документации приложения [3].

3. Визуальный конструктор для создания мобильных приложений: платформа Adalo

Adalo (<https://www.adalo.com/>) – платформа для создания мобильных приложений без использования программного кода. Платформу основали в 2018 г. Эд Эдкин, Бен Хейфел и Джереми Блалок в США. Через год компания выиграла грант на 50 тыс. долл. от Arch Grants.

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

- создание неограниченного количества приложений;
- до 50 строк данных на одно приложение;
- возможность публикации в домене Adalo.

В интерфейсе создаваемых приложений присутствует реклама Adalo.

Для полноценной работы приложения такой объем сохраняемых записей в базе данных станет препятствием в процессе эксплуатации, но это не значит, что конкретно это ограничение невозможно обойти. Adalo поддерживает интеграцию со сторонними решениями для хранения данных, например, Airtable (<https://www.airtable.com/>) с бесплатным планом использования, включающим: неограниченное количество баз, работу до пяти пользователей (т.е. администраторов), конкретная база может обслуживать только одно приложение, одна интеграция, до 1200 записей на каждую базу и 2 ГБ вложений на базу. Для запуска бизнес-приложения это достаточно комфортные параметры.

Ценообразование на сервис представлено на официальной странице <https://www.adalo.com/pricing> (рис. 3.1, 3.2).

Регистрация в Adalo – это переход по кнопке Get Started (рис. 3.3). В качестве email используем свой аккаунт @gmail.com, на который придет письмо для верификации адреса (рис. 3.4).

После входа в Adalo предлагается создать приложение. Выбирается тип приложения: мобильное приложение или настольное веб-приложение (рис. 3.5); выбирается шаблон или пустой экран (рис. 3.6), выбирается цветовая схема для приложения (эти настройки можно изменить в любое время, и любые компоненты, использующие предустановленные цвета, будут автоматически обновлены) и вводится название приложения (рис. 3.7). Из последнего окна настройки можно

получить доступ к выбору дополнительных опций. Дополнительные параметры позволяют определить ввод данных / источник данных для вашего приложения, создать новую базу данных / настроить новый API или использовать базы данных с существующим приложением, доступным на панели мониторинга проекта.

Bring Your Idea to Life
Without Breaking the Bank

Free!	\$50/mo	\$200/mo
Explore Plan Perfect if you have an idea for an app and want to get started.	Pro Plan Perfect if you're launching the first version of your app.	Business Plan Perfect for your team to grow, iterate, and scale your app to reach everyone!
<ul style="list-style-type: none"> Unlimited Apps! Publish to an Adalo domain Adalo Branding Limited to 50 rows of data per app 	<ul style="list-style-type: none"> Unlimited Apps! Publish to Apple & Google Stores Publish to a Custom Web Domain Connect to External Data Collections Trigger Unlimited Custom Actions 5 GB of Data Storage 1 Seat (App Builder) 	<ul style="list-style-type: none"> Unlimited Apps! Publish to Apple & Google Stores Publish to a Custom Web Domain Connect to External Data Collections Trigger Unlimited Custom Actions 20 GB of Data Storage 5 Seats (App Builders) Dedicated Account Manager
GET STARTED	GET STARTED	GET STARTED

Рис. 3.1. Тарифы на использование платформы Adalo

Free!	\$50/mo	\$200/mo
Изучите план Идеально, если у вас есть идея для приложения и вы хотите начать.	Pro Plan Идеально, если вы запускаете первую версию своего приложения.	Бизнес-план Идеально подходит для вашей команды, чтобы расти, повторять и масштабировать ваше приложение, чтобы охватить всех!
<ul style="list-style-type: none"> Неограниченное количество приложений! Публикация в домене Adalo Адало Брендинг Ограничено до 50 строк данных на приложение. 	<ul style="list-style-type: none"> Неограниченное количество приложений! Публикация в магазинах Apple и Google Публикация в пользовательском веб-домене Подключение к внешним коллекциям данных Запускайте неограниченное количество настраиваемых действий 5 ГБ для хранения данных 1 место (разработчик приложений) 	<ul style="list-style-type: none"> Неограниченное количество приложений! Публикация в магазинах Apple и Google Публикация в пользовательском веб-домене Подключение к внешним коллекциям данных Запускайте неограниченное количество настраиваемых действий 20 ГБ хранилища данных 5 рабочих мест (разработчики приложений) Выделенный менеджер по работе с клиентами
НАЧАТЬ	НАЧАТЬ	

Рис. 3.2. Тарифы на использование платформы Adalo

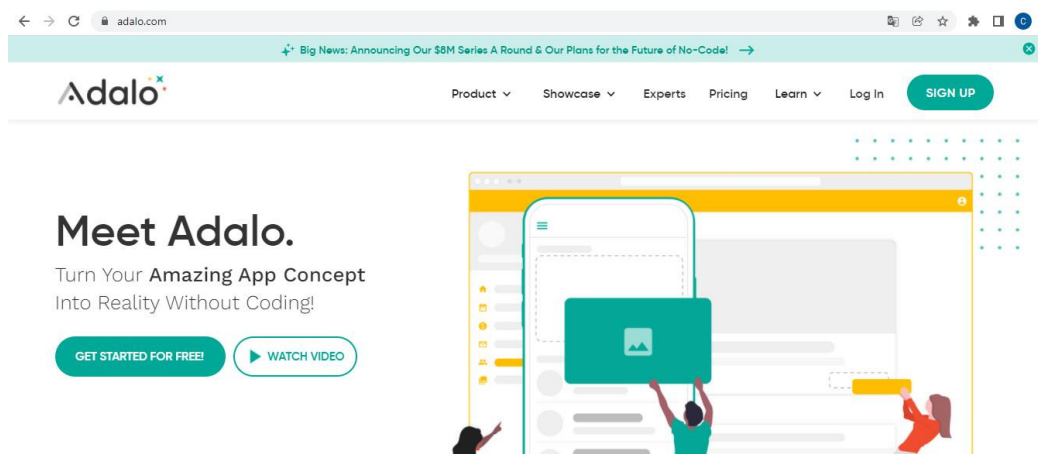


Рис. 3.3. Регистрация в сервисе Adalo

This is a screenshot of the 'Sign Up For Free!' form on the Adalo website. The form is centered and has a clean, modern design. It includes the Adalo logo at the top. The fields are: 'Email Address' (containing 'svetlanar004@gmail.com'), 'Password' (masked with dots), and 'Full Name' (containing '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. 3.4. Авторизация в сервисе Adalo

This screenshot shows the 'Platform' selection step of the Adalo onboarding process. At the top, there are three numbered steps: '1 Platform', '2 Template', and '3 Branding'. The main heading is 'What kind of app do you want to make?'. Below this, there are two large, light-gray rectangular buttons. The left button features a smartphone icon and is labeled 'Native Mobile App'. The right button features a desktop monitor icon and is labeled 'Desktop Web App'. A red arrow points from the left towards the 'Native Mobile App' button.

Рис. 3.5. Выбор типа приложения

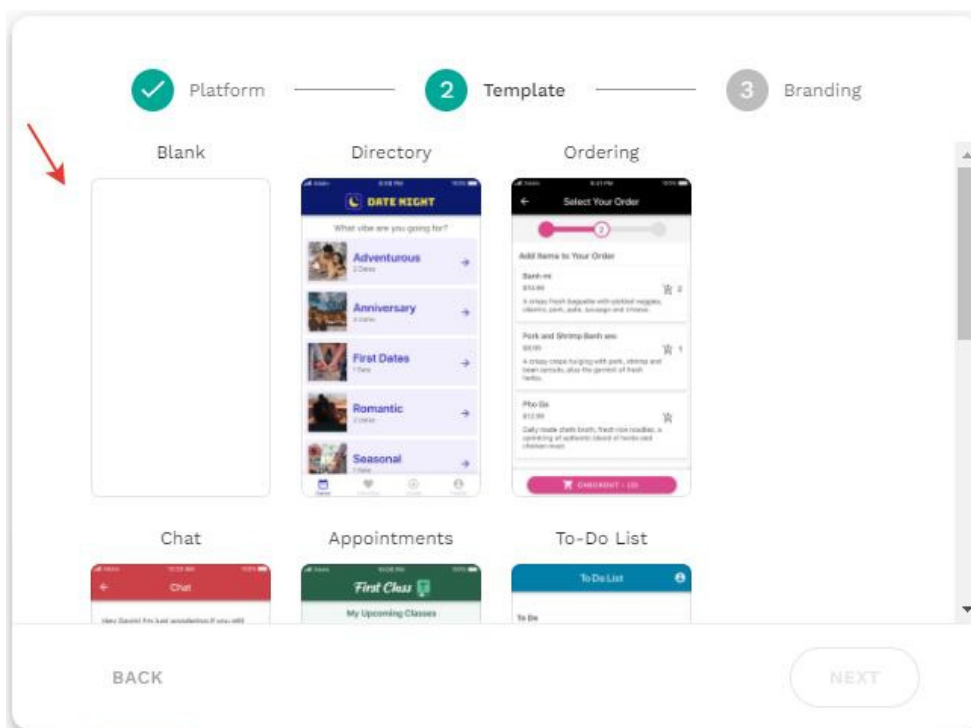


Рис. 3.6. Выбор шаблона для приложения

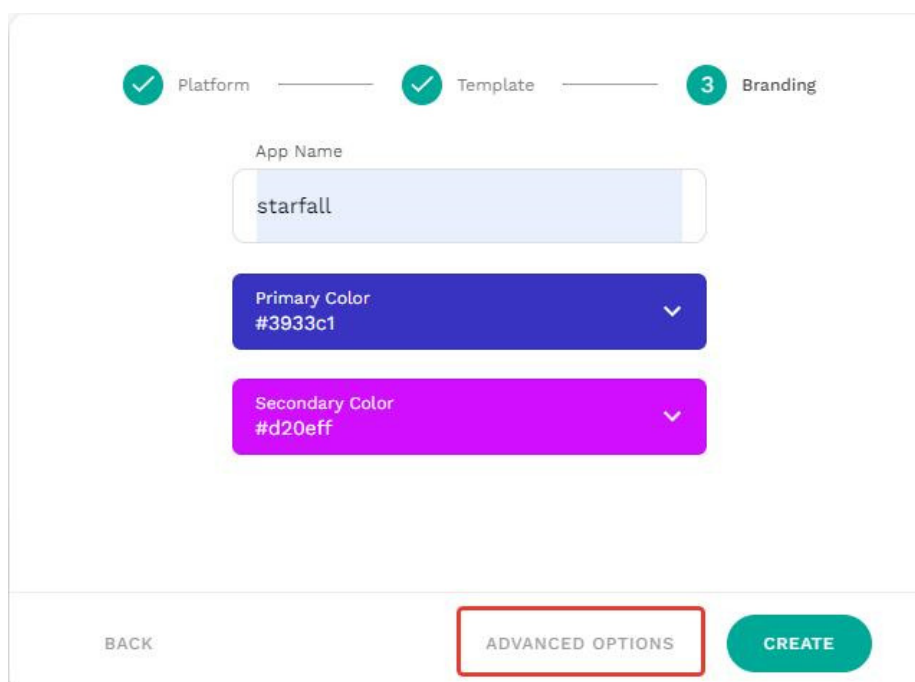


Рис. 3.7. Выбор цветовой схемы приложения
и ввод его названия

После базовых настроек для нового приложения Adalo открывает холст (canvas) – рабочую область, в которой проектируются экраны приложения (рис. 3.8). 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 +/-.

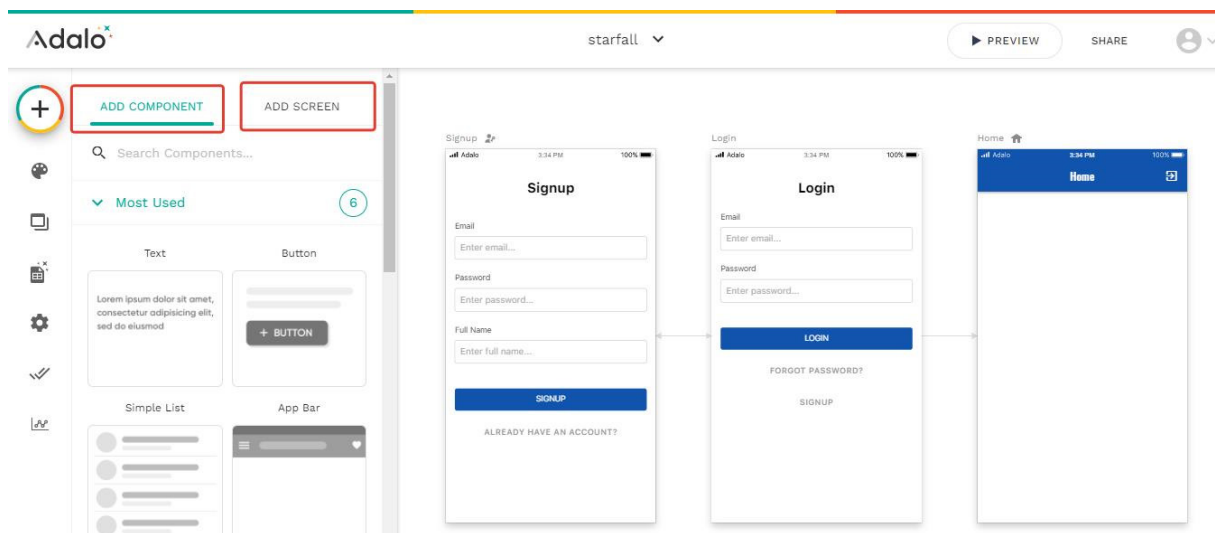


Рис. 3.8. Рабочая область для редактирования приложения

Разработка приложений в Adalo основана на трех основных концептах [4]:

1. Экраны и компоненты (Screens and Components) – строительные блоки интерфейса Adalo.
2. Действия (Actions) используются для определения и настройки того, что должно делать приложение, когда происходит нажатие на определенный компонент.
3. База данных (Database) позволяет вводить данные для отображения пользователям приложения или используется 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).

По умолчанию в новое приложение при создании включаются три шаблонных экрана с уже добавленными на них компонентами, реализующими их назначение: Home (главный экран), Signup (экран регистрации), экран Login (экран авторизации).

Левая панель инструментов содержит следующие опции, доступные по кнопкам:

1. Кнопка  позволяет добавить экраны (рис. 3.9) и компоненты (рис. 3.10). Вкладка ADD COMPONENT содержит инструменты для добавления компонентов пользовательского интерфейса, которые наиболее часто используются: навигация, списки, кнопки, простые формы, поля и формы, платежи, диаграммы и карта.

Три цвета, используемые в оформлении границы кнопки Добавить, отвечают за цветовую маркировку добавляемых элементов:

зеленым цветом окрашены визуальные элементы, красным – элементы базы данных, желтым – добавляемые действия.

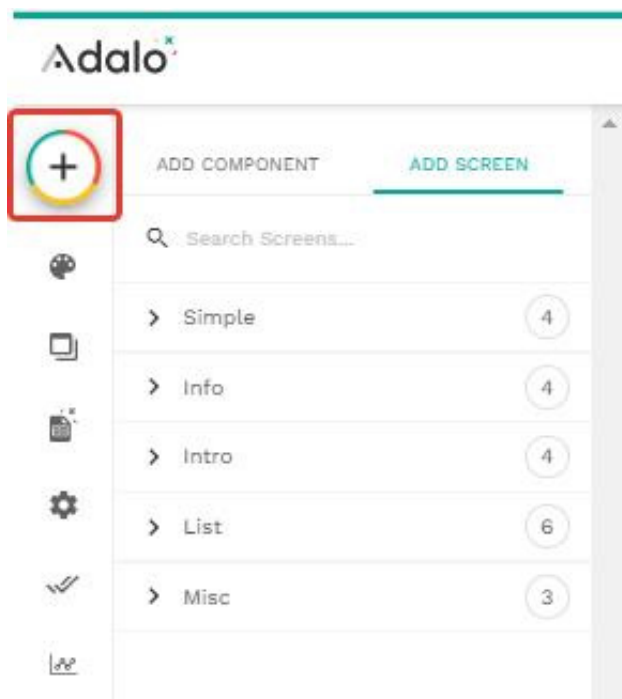


Рис. 3.9. Добавление экранов на холст

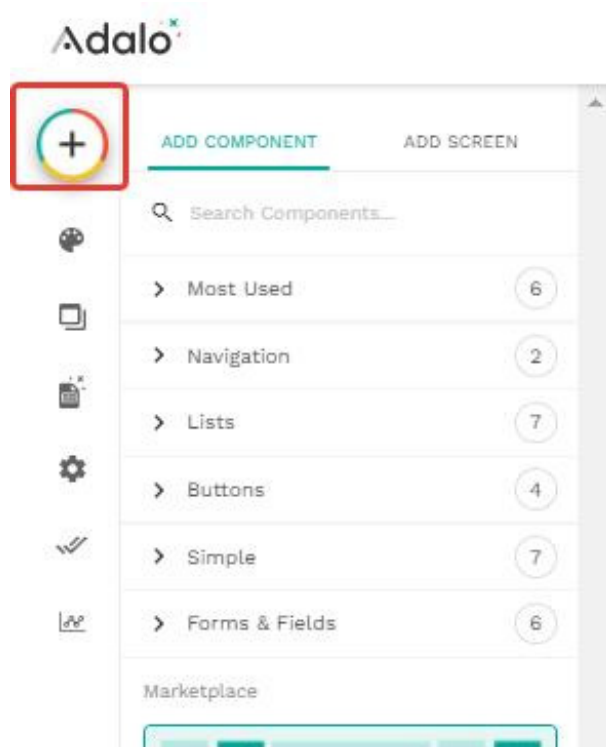


Рис. 3.10. Добавление компонентов на экран

2. Кнопка  отвечает за брендинг приложения и позволяет определить основные и второстепенные цвета для него (рис. 3.11).

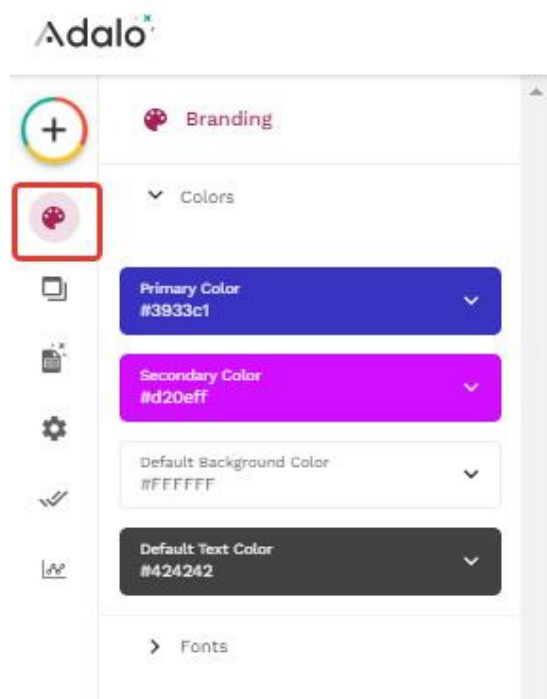


Рис. 3.11. Настройка цветовой схемы для приложения

Цвета основной и второстепенный были определены в момент создания приложения (рис. 3.7), но могут быть изменены в процессе разработки. В бесплатном плане использования выбор шрифтов крайне ограничен, но они хорошо масштабируются под различные разрешения и вполне эргономичны.

3. Кнопка  позволяет получить список всех экранов (рис. 3.12). В этой вкладке можно добавить экран на холст. При клике на экран в списке можно получить доступ ко всем компонентам этого экрана (рис. 3.13). Каждый элемент доступен для редактирования.

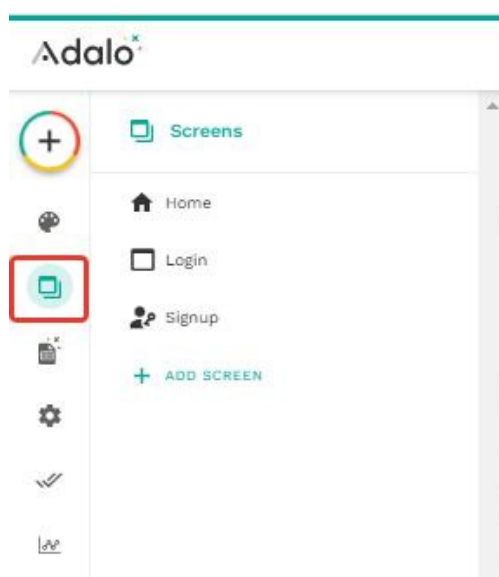


Рис. 3.12. Список экранов приложения

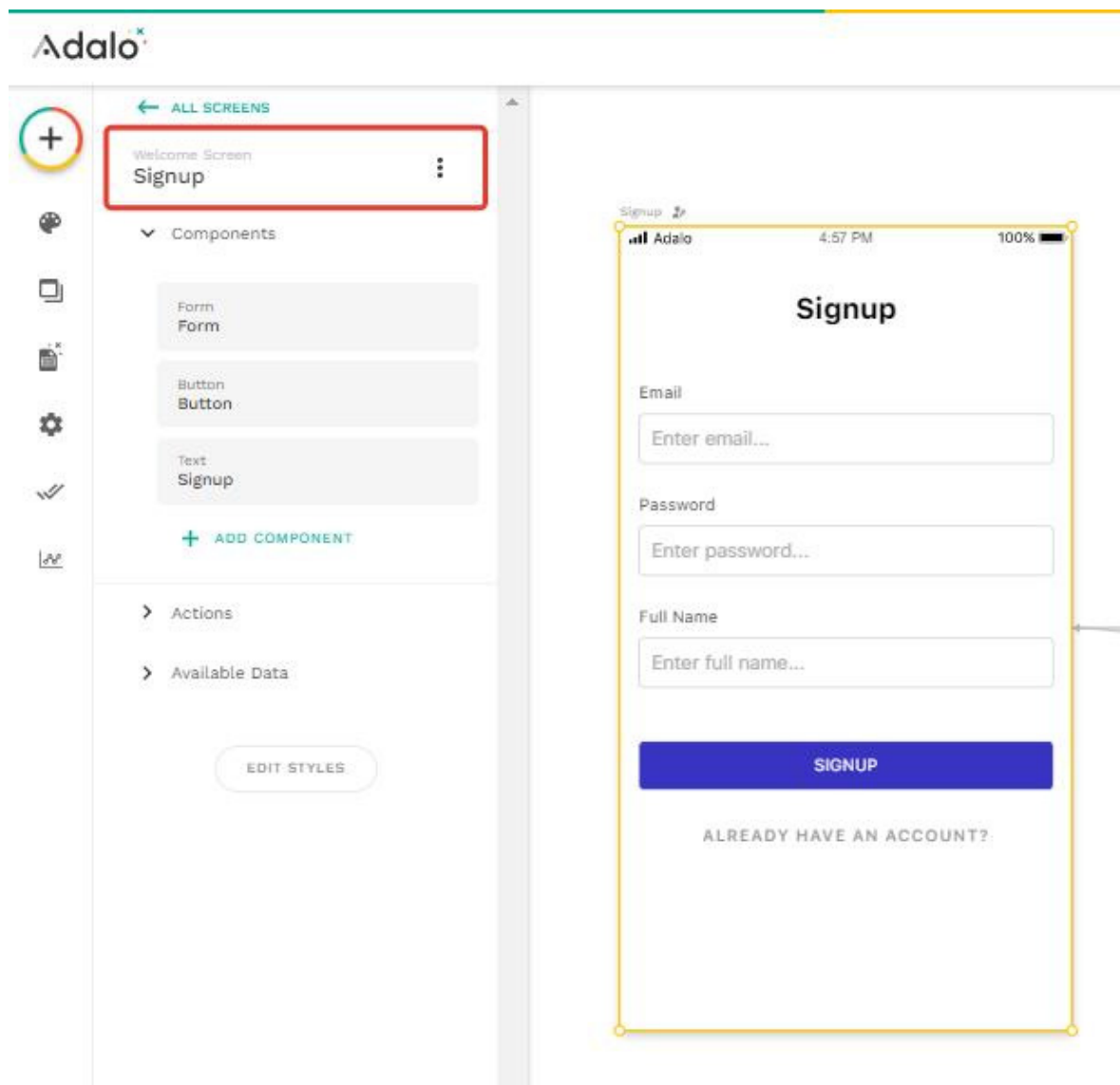


Рис. 3.13. Переход к компонентам экрана Signup, добавленного в приложение по умолчанию

4. Кнопка  открывает доступ к просмотру и обновлению базы данных (рис. 3.14). Красный цвет используется для обозначения элементов данных и не связан с ошибками. По умолчанию добавлена коллекция (таблица данных) Users и свойства (поля): Email, Password, Username, Full Name. Записей в этой коллекции пока нет.

5. Кнопка  позволяет определить/изменить базовые параметры приложения: название, его логотип, также есть возможность скопировать или удалить приложение (рис. 3.15). Для приложения был добавлен логотип (рис. 3.16). После выбора файла для логотипа необходимо кликнуть по кнопке Update app (обновить приложение), которая появится под добавленным изображением.

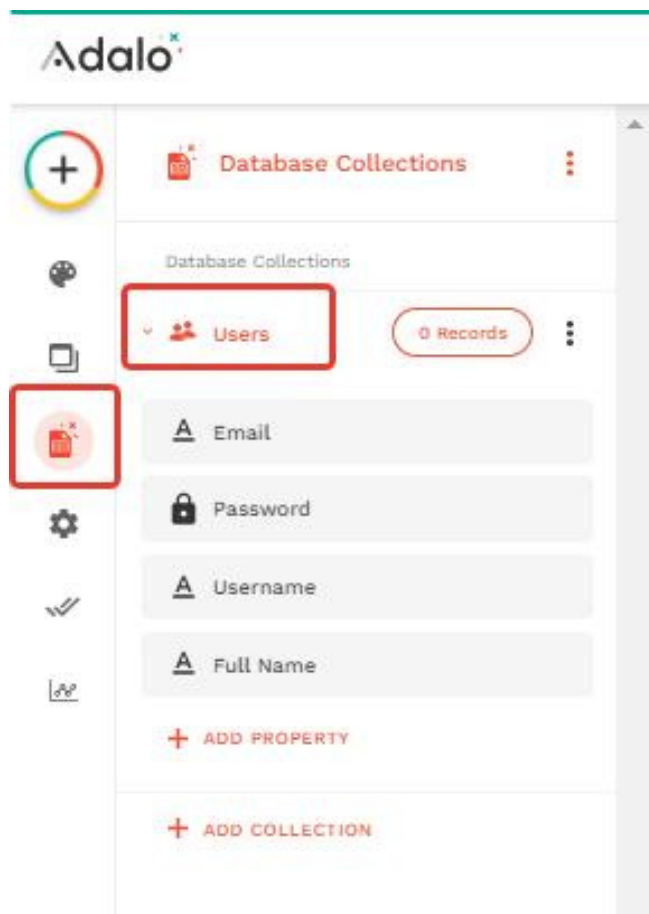


Рис. 3.14. Доступ к базе данных приложения

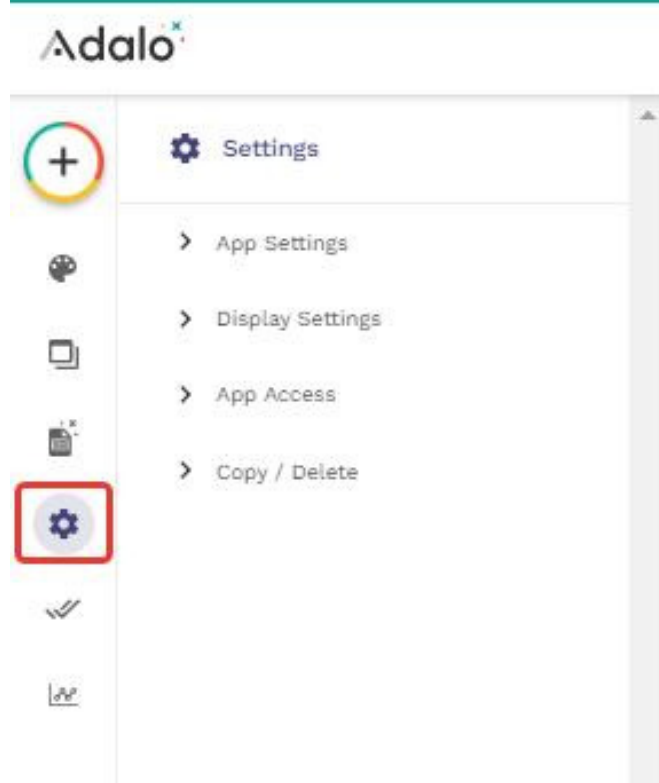


Рис. 3.15. Настройки для приложения

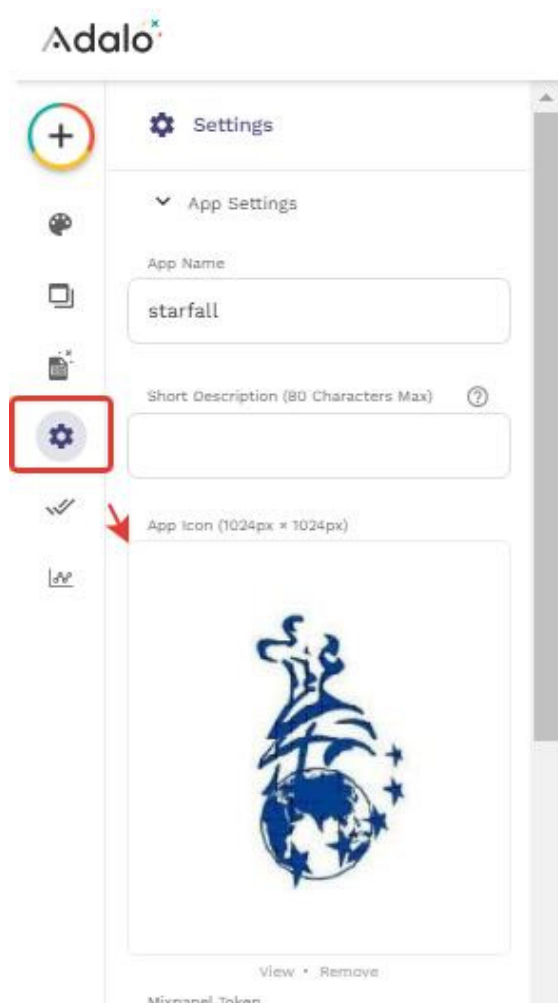


Рис. 3.16. Добавление логотипа

6. Кнопка  позволяет загрузить файл приложения в Apple App Store и Google Play Store, а для веб-приложений можно указать URL своего приложения (рис. 3.17), но эти функции будут доступны только на платных тарифах.

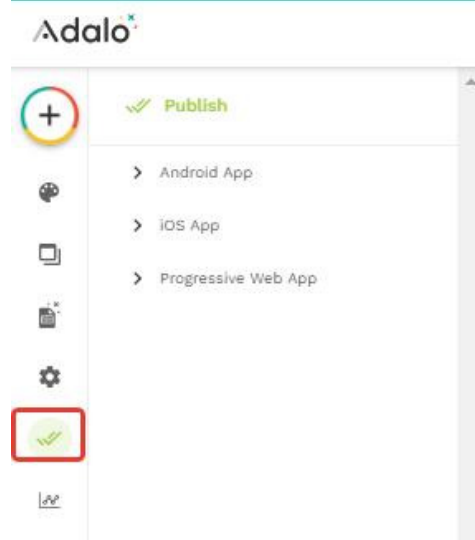


Рис. 3.17. Публикация приложения

7. Кнопка  позволяет строить интерактивные панели на основе данных использования приложения (рис. 3.18).

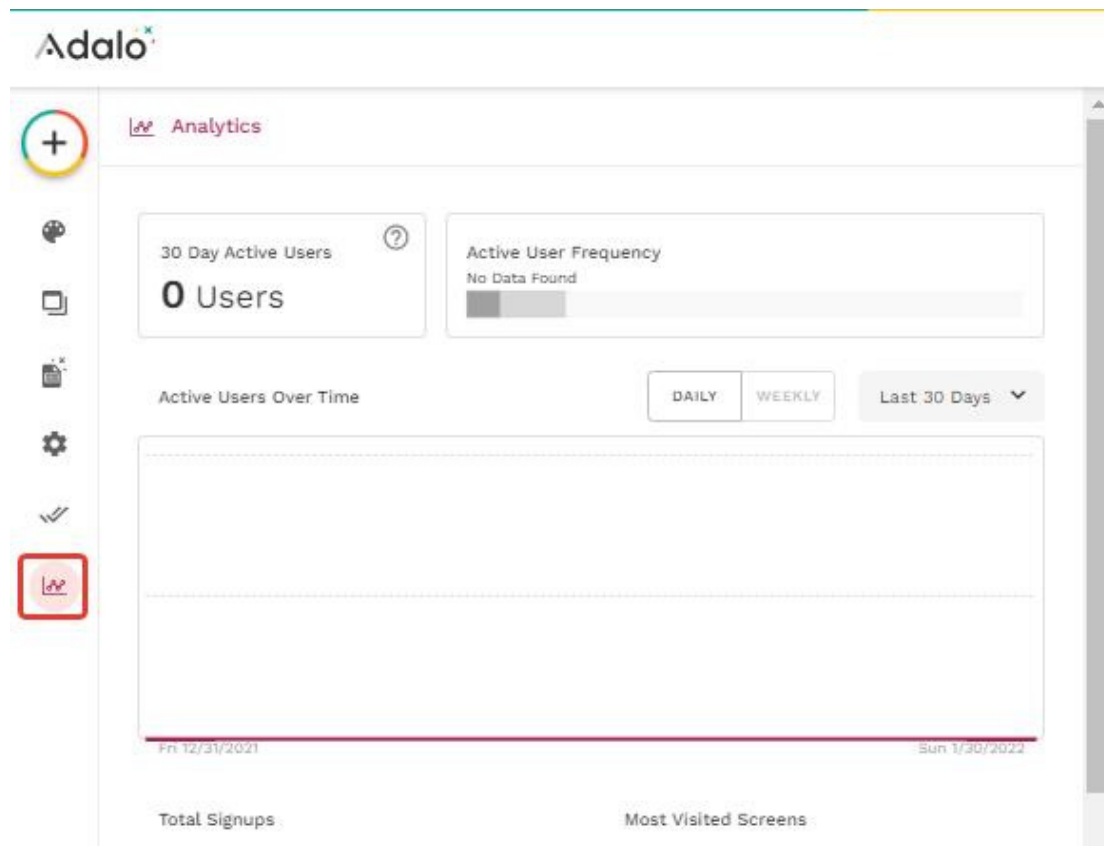


Рис. 3.18. Встроенная аналитика для приложения

В верхней панели справа доступны кнопки PREVIEW и SHARE (рис. 3.19).

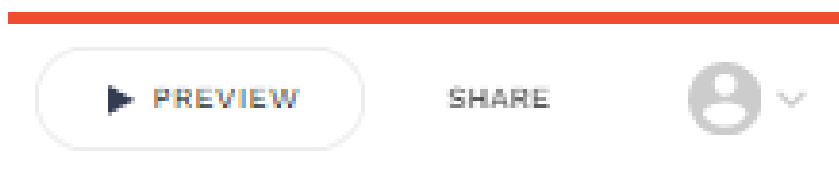


Рис. 3.19. Кнопки верхней панели инструментов

Опция Предварительный просмотр реализуется с помощью всплывающего окна (рис. 3.20). Можно проконтролировать, как будет выглядеть редактируемое приложение после внесения изменений. Также можно выбрать тип гаджета, для которого будет настроен предварительный показ.

Кнопка Поделиться предоставляет ссылку на приложение (<https://previewer.adalo.com/d6c02bfc-ca10-4586-8b14-1d436ce44d2b>), которую можно отправить пользователям, у которых нет учетной записи Adalo. Это позволяет получить доступ к приложению (рис. 3.21) для проведения тестирования, для получения обратной связи и т.п.

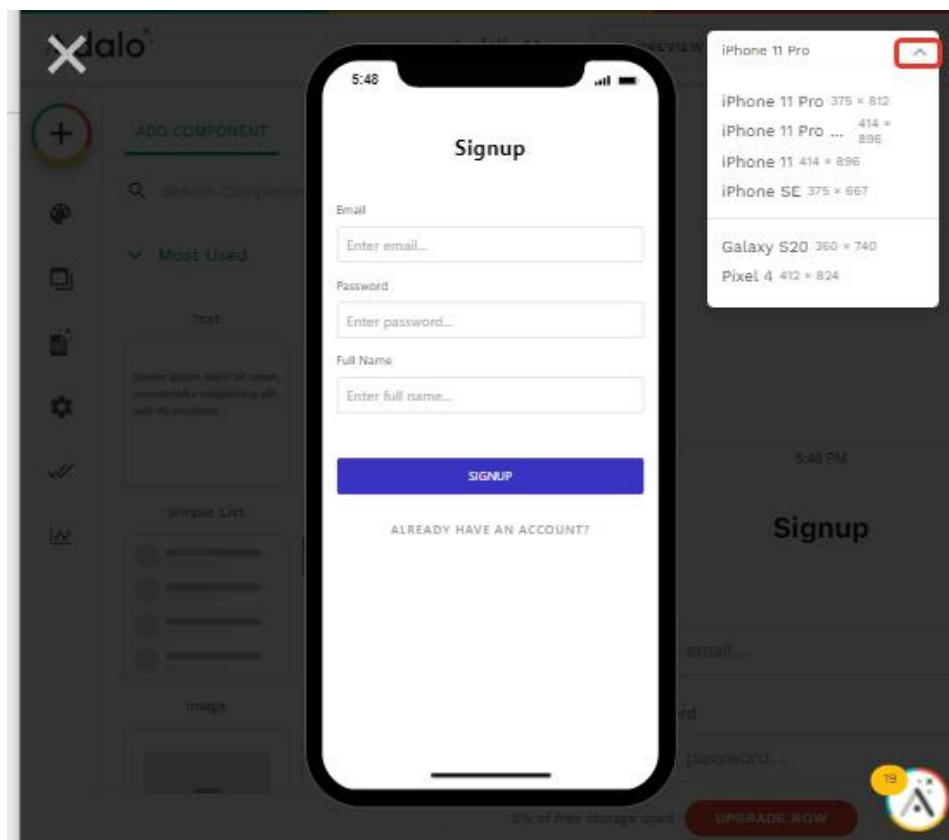


Рис. 3.20. Предварительный просмотр

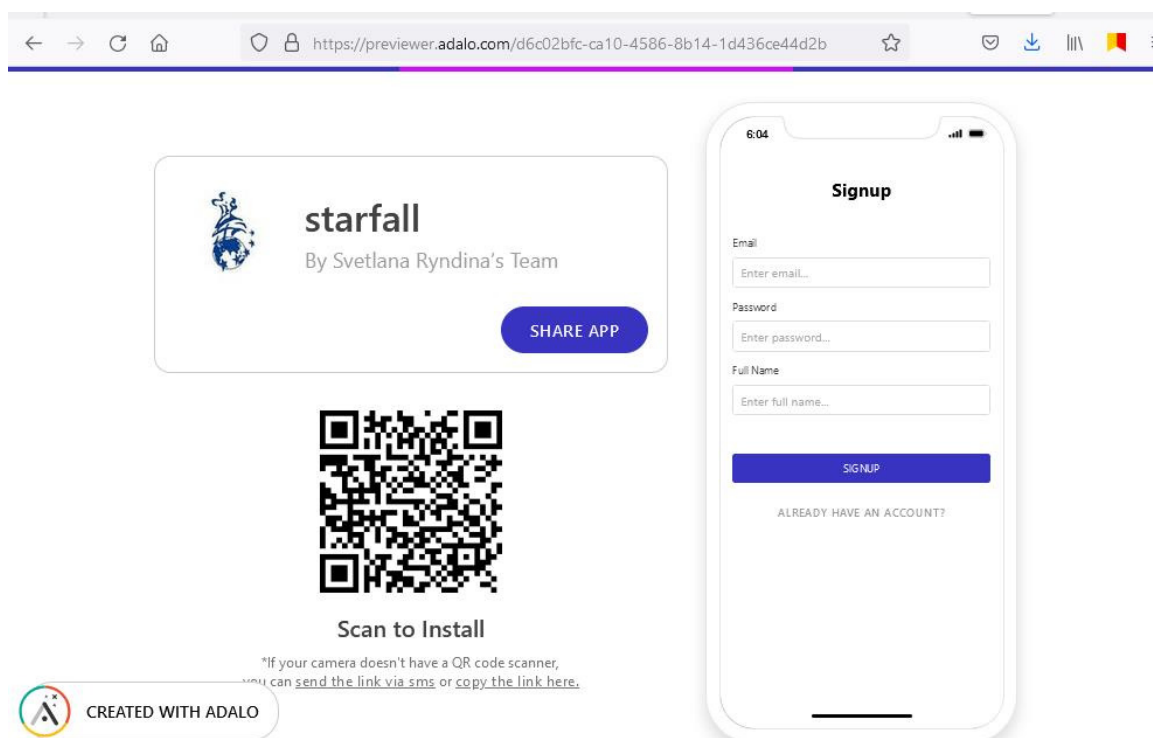


Рис. 3.21. Загрузка приложения в браузере без авторизации в Adalo

По центру в верхней панели располагается кнопка с названием приложения, это одновременно и переключатель между приложениями и возможность создать новое (рис. 3.22).

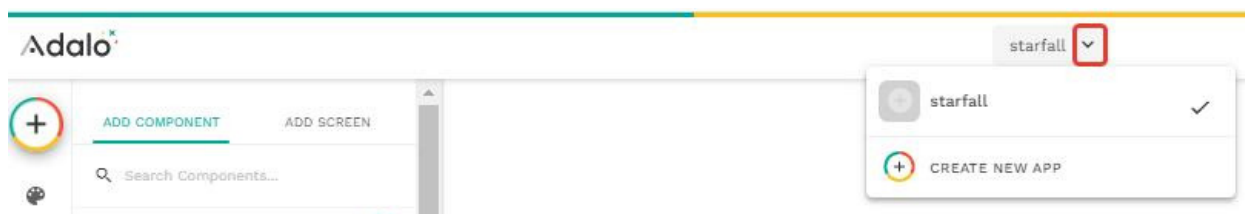


Рис. 3.22. Переключатель между приложениями

В настройках учетной записи (рис. 3.23) можно изменить данные профиля, тарифный план, настроить домены.

The screenshot shows the 'Profile' tab of the Adalo account settings. The page has three tabs: 'Profile', 'Team & Billing', and 'Domains'. The 'Profile' tab is active. It contains several input fields: 'Email' with the value 'svetlanar004@gmail.com', 'Name' with the value 'Svetlana Ryndina', and 'Password' with a 'CHANGE PASSWORD' button. Below these is a 'Developer Mode' section with a toggle switch labeled 'Enable Developer Mode' which is currently off. At the bottom right is an 'UPDATE ACCOUNT' button. A link 'Learn more at developers.adalo.com' is also present.

Рис. 3.23. Изменение настроек учетной записи

4. Разработка мобильного приложения для танцевально-спортивного клуба «Звездопад»

Для демонстрации возможностей сервиса Adalo в разработке мобильных приложений без написания кода рассмотрим создание мобильного приложения для танцевально-спортивного клуба «Звездопад». Это реально существующий клуб и весь контент предоставлен Валерией Толмачевой – одним из руководителей клуба – с согласия и разрешения остальных участников. Это очень ценно, когда даже в демонстрационном варианте используются не фейковые данные, и автор пособия очень признателен Валерии за предоставленные материалы.

Информация о работе клуба, об условиях вступления в клуб, о тренерах, о программах подготовки должна быть размещена на отдельных вкладках приложения. Кроме того, необходимо настроить страницы регистрации, авторизации, главную страницу приложения, которые по умолчанию добавлены в момент его создания (см. рис. 3.8).

На экране заставки Welcome будет размещена информация о клубе, об акционных предложениях.

Начнем с добавления нового экрана (рис. 4.1). Это будет пустой экран, на который постепенно добавим все необходимые компоненты.

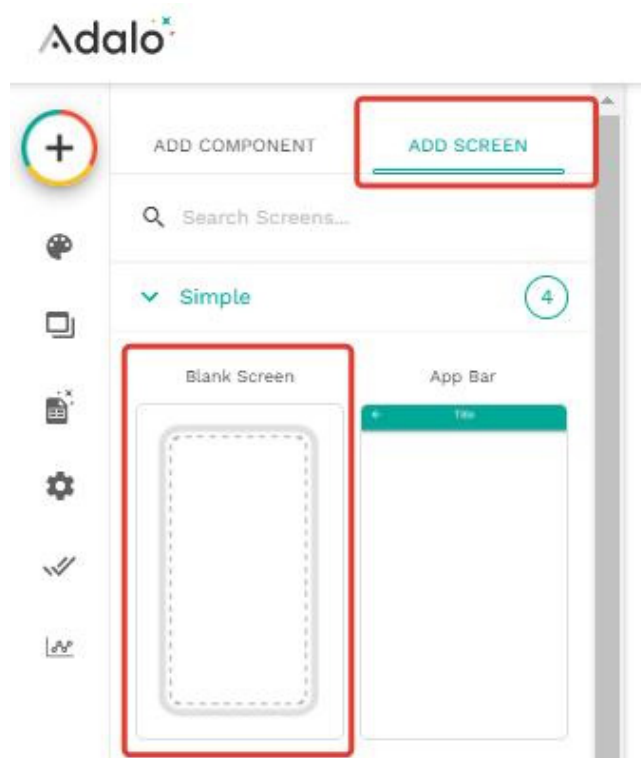


Рис. 4.1. Добавление на холст экрана

Экрану необходимо дать название (рис. 4.2). Это будет экран, который доступен всем неавторизованным пользователям при входе в приложение.

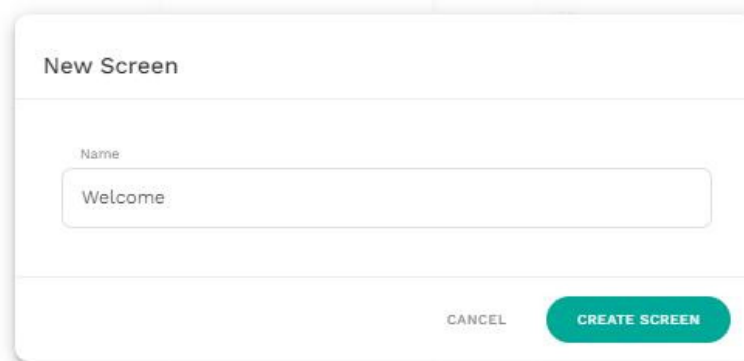


Рис. 4.2. Присвоение созданному экрану названия

Выделим созданный экран на холсте (можно кликнуть мышкой на его названии в левом верхнем углу над экраном) и перенесем его в позицию перед тремя экранами по умолчанию, так как заставка должна быть первым экраном на холсте. Настройку того, что загрузка приложения начинается с этого экрана, проведем позднее.

Добавим на экран компонент Image и выберем вариант доступа к ресурсу для добавления изображения Upload (загрузить), появится окно загрузки, в котором выберем файл логотипа (рис. 4.3).

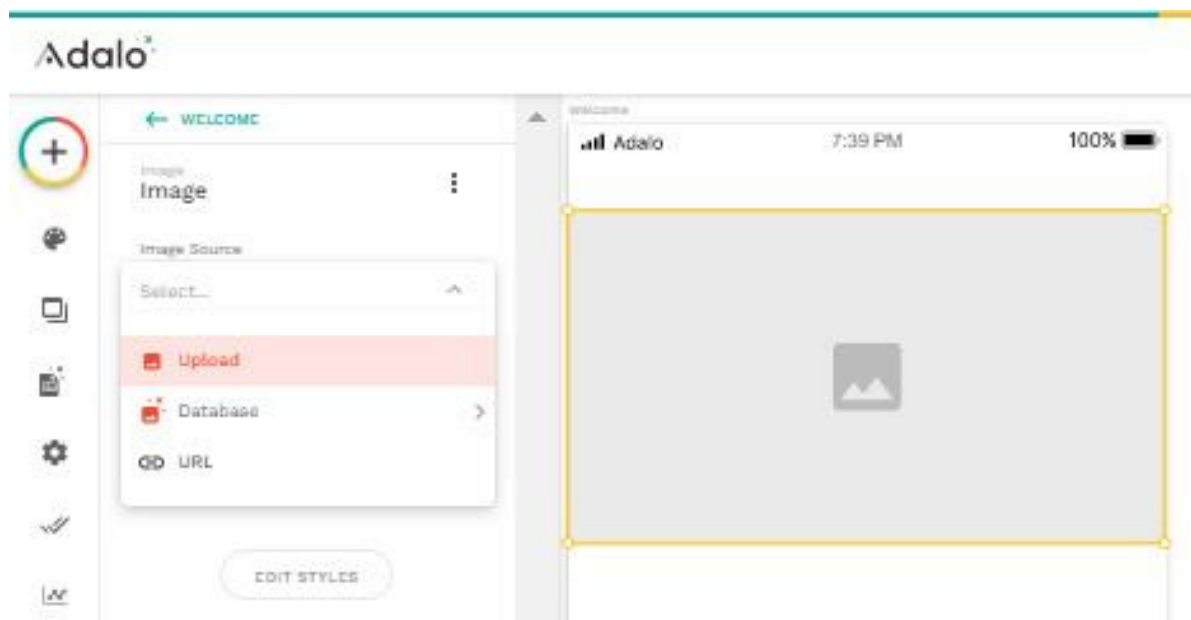


Рис. 4.3. Загрузка изображения

Добавим текстовый элемент. Название текстового элемента (если не задается отдельно) будет совпадать с самим текстом. Для форматирования текста нажмем кнопку «дизайн» и изменим выравнивание, размер текста, начертание (рис. 4.4).

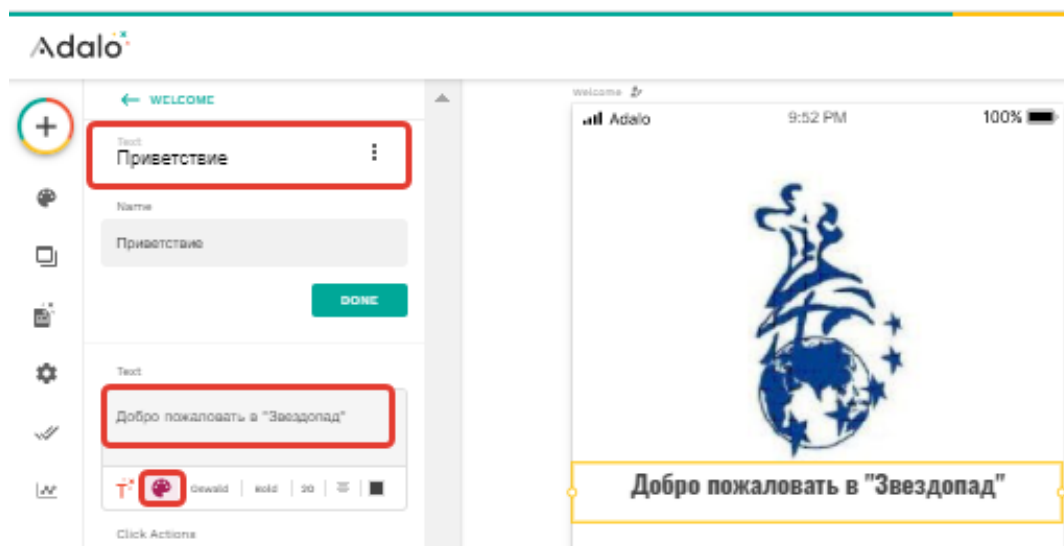


Рис. 4.4. Добавление и редактирование текстового компонента

Аналогично добавим еще два текстовых элемента. Для текстового элемента ВКонтакте проведем настройку действия (рис. 4.5). При клике мышкой на этом тексте должен осуществляться переход на ВКонтакте клуба.

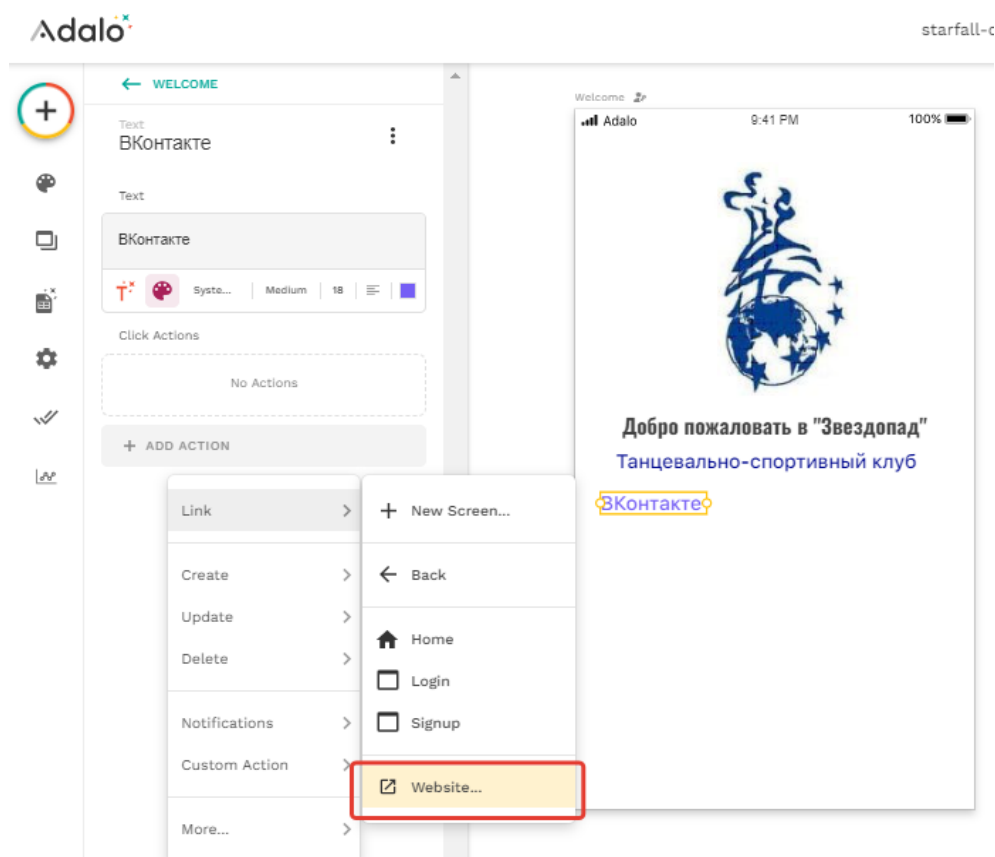


Рис. 4.5. Добавление действия к текстовому компоненту

В окне для ссылки (рис. 4.6) добавим URL: <https://vk.com/zvezdopadpnz>.

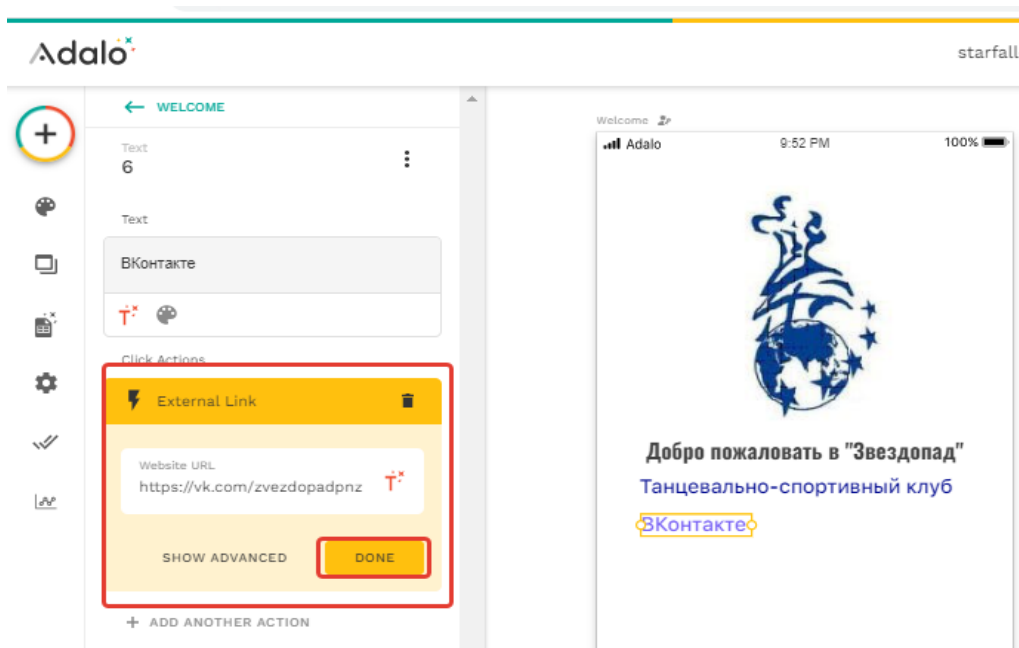


Рис. 4.6. Добавление ссылки

Добавим рядом с текстом кнопку. Уменьшим ее размер и настроим ее отображение следующим образом: добавим иконку с изображением (group_add), скругление кнопки сделаем равным 11, добавим к этому компоненту действие так, как это было сделано для текстового компонента (рис. 4.7).

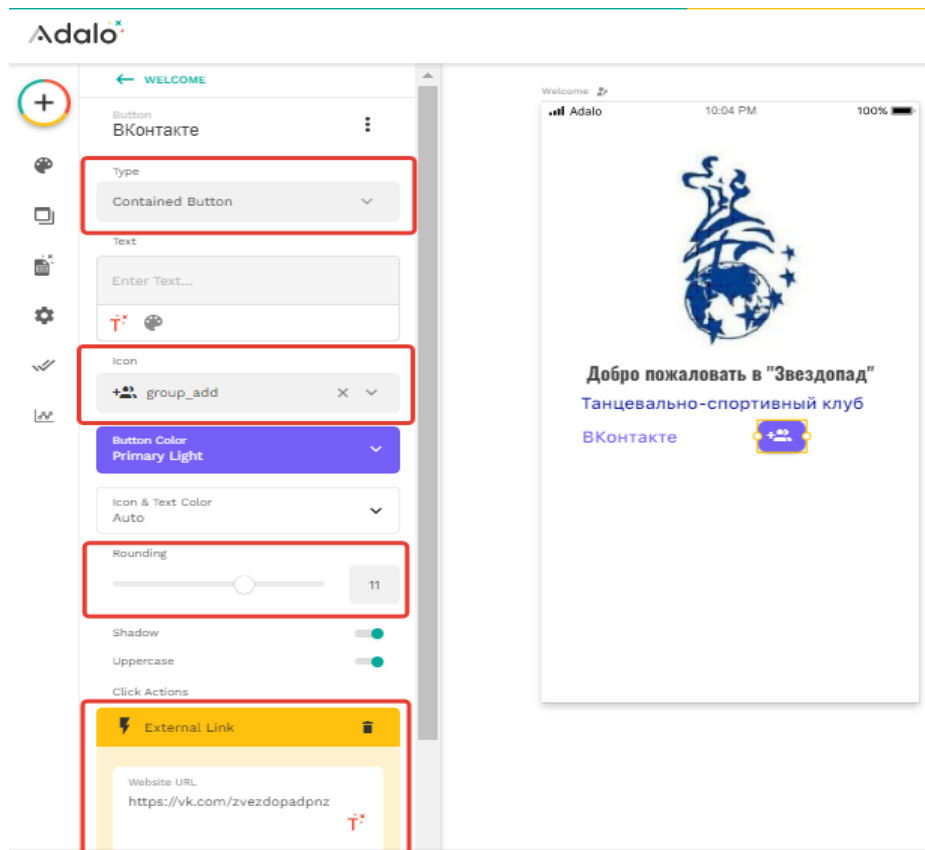


Рис. 4.7. Настройки для кнопки

Тип кнопки может быть разным, например, изменим его на Outlined Button и добавим еще два текстовых элемента на экран (рис. 4.8).

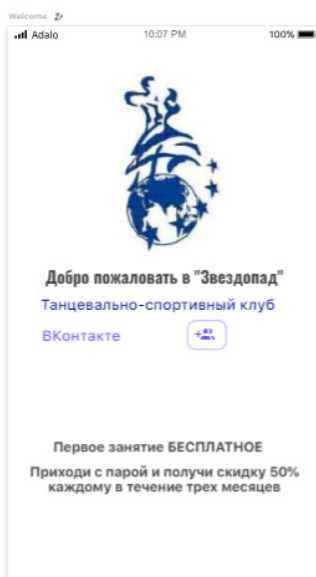


Рис. 4.8. Добавление новых элементов

Так как в окне заставки необходимо предусмотреть возможность авторизации и регистрации, добавим соответствующие кнопки. Первую из двух кнопок отформатируем, а вторую скопируем и изменим название и текст (рис. 4.9).

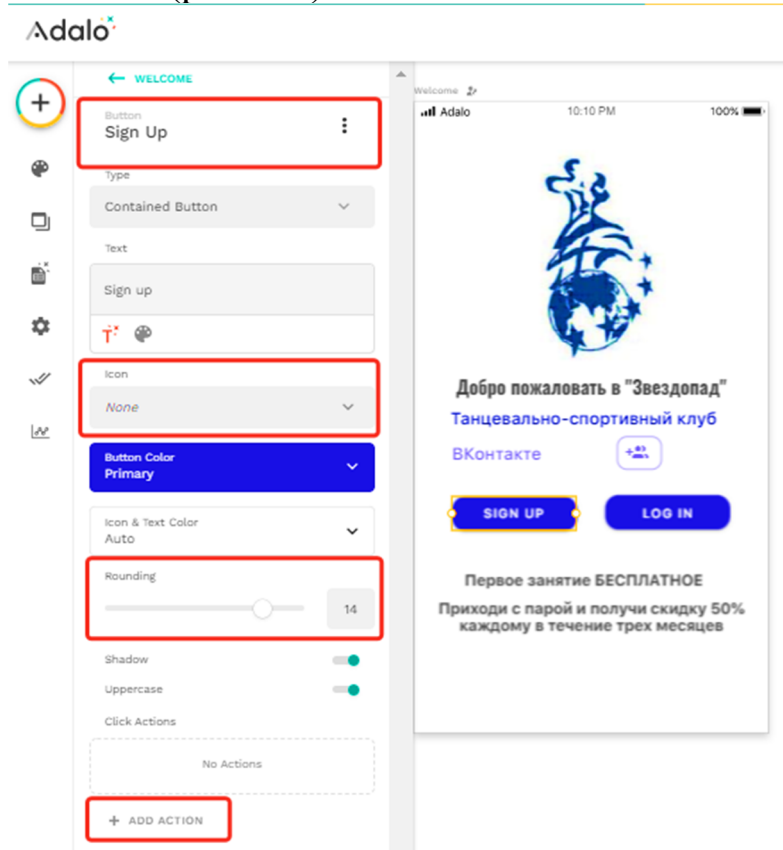


Рис. 4.9. Настройка кнопки

После этого настроим действие для каждой кнопки. Это тоже Link, но теперь ссылка должна быть связана с экраном, на который будем переходить по кнопке (рис. 4.10). Для каждой кнопки экран перехода будет свой. В появившемся окне настроек ссылки нажимаем кнопку Done (рис. 4.11).

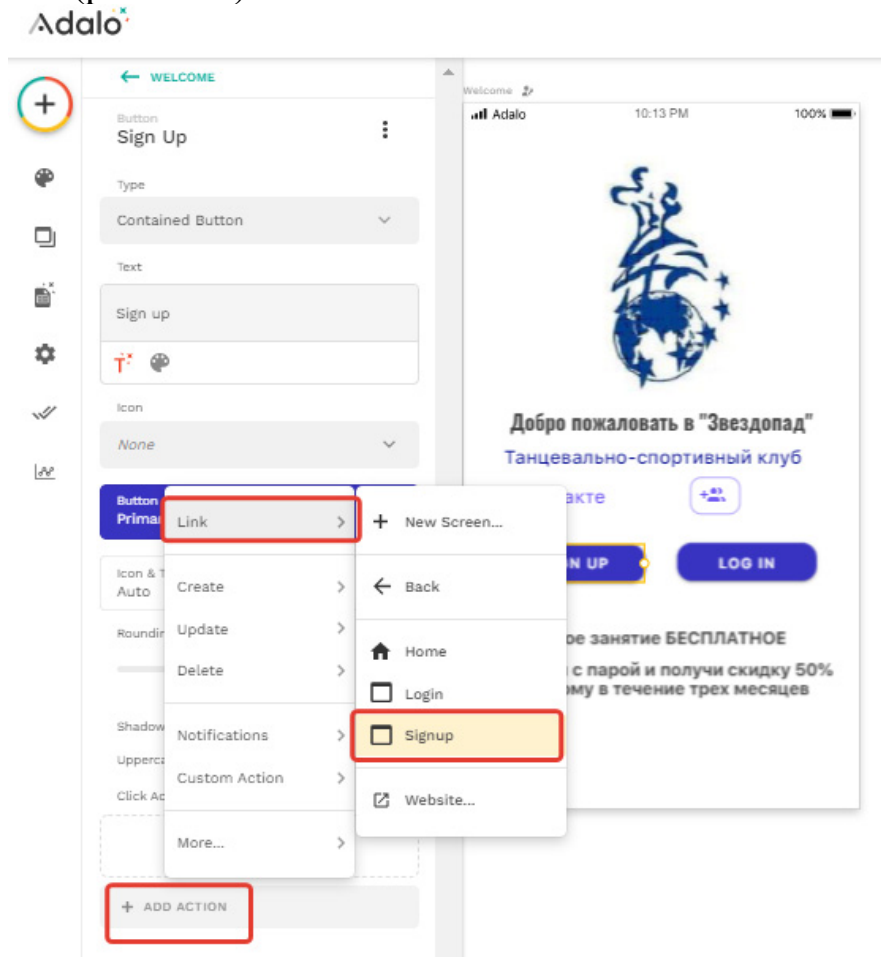


Рис. 4.10. Настройка действия для кнопки

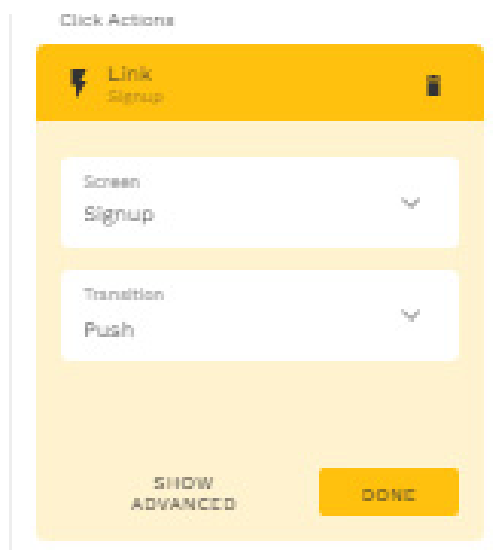


Рис. 4.11. Закрепление настроек

Выделим экран Welcome и сделаем его первым экраном после загрузки приложения (настройки очередности доступны после клика на название экрана в левой панели со списком всех компонентов). Для этого выберем тип Welcome Screen (рис. 4.12). Еще один экран с особым значением типа – это Home Screen. Этот экран был создан по умолчанию, и на него осуществляется переход после авторизации или регистрации.

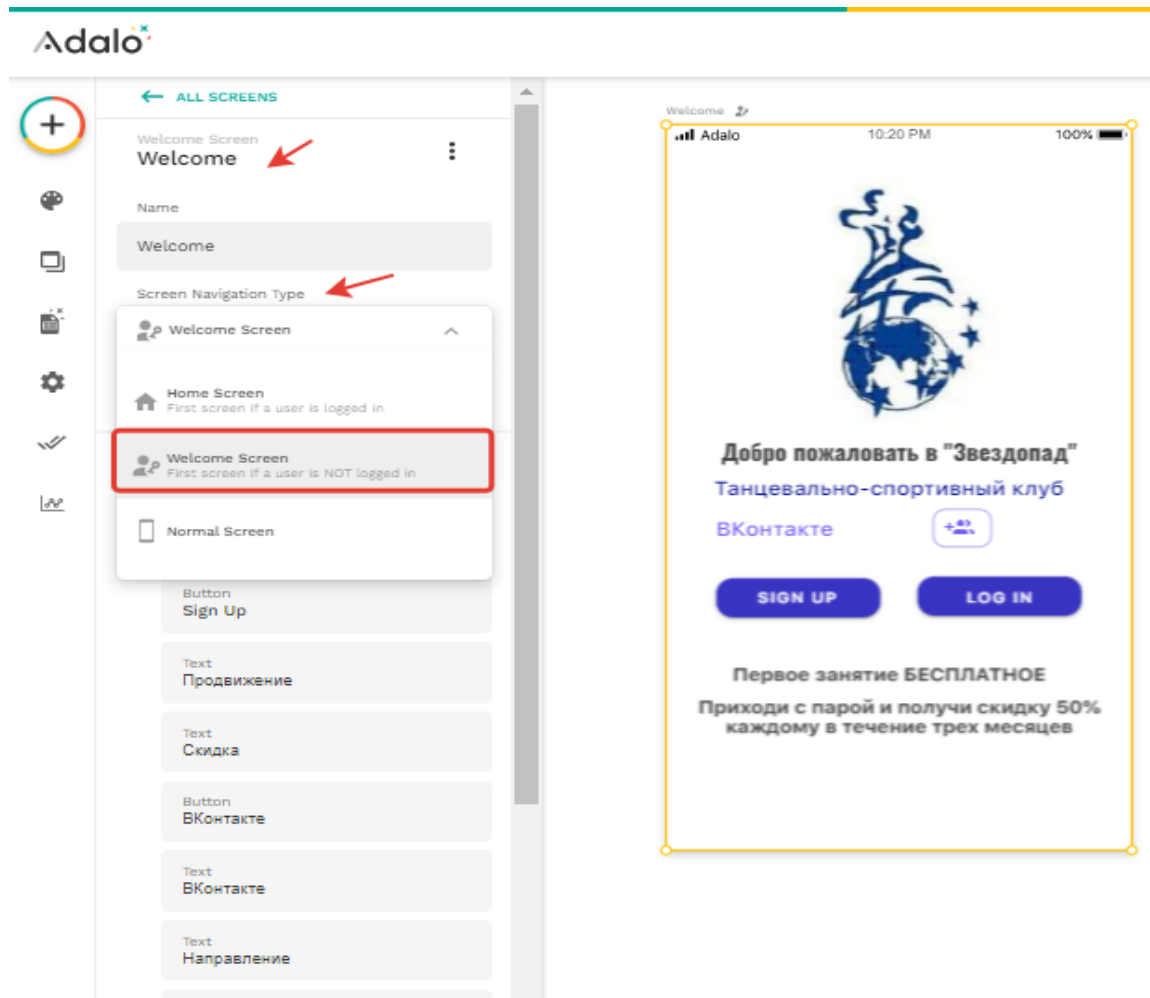


Рис. 4.12. Настройка очередности показа экранов

Для всех остальных экранов – тип Normal Screen, и на них должны быть настроены переходы, чтобы они были доступны в приложении.

Добавим на этот экран адрес местоположения клуба (г. Пенза, ул. Володарского 6Б) и свяжем с ним действие – переход по ссылке на карту (https://yandex.ru/maps/49/penza/house/ulitsa_volodarskogo_6b/YEwYdgZgTUcHqFtpfX15d3RqYQ==/panorama/?l=stv%2Csta&ll=45.010420%2C53.186895&panorama%5Bdirection%5D=290.600000%2C10.000000&panorama%5Bfull%5D=true&panorama%5Bpoint%5D=45.011443%2C53.186666&panorama%5Bspan%5D=122.690874%2C60.000000&sour

ce=wizgeo&tab=panorama&utm_medium=maps-desktop&utm_source=serp&z=16).

А вот кнопку добавим иначе, чем делали для перехода в группу ВКонтакте. Во вкладке Добавить компонент (левая панель инструментов), есть раздел Button, в котором больше вариантов для кнопок и есть кнопка Icon. Ее добавим рядом с адресом местоположения (рис. 4.13). Также настроим для нее действие (переход по ссылке на карту).

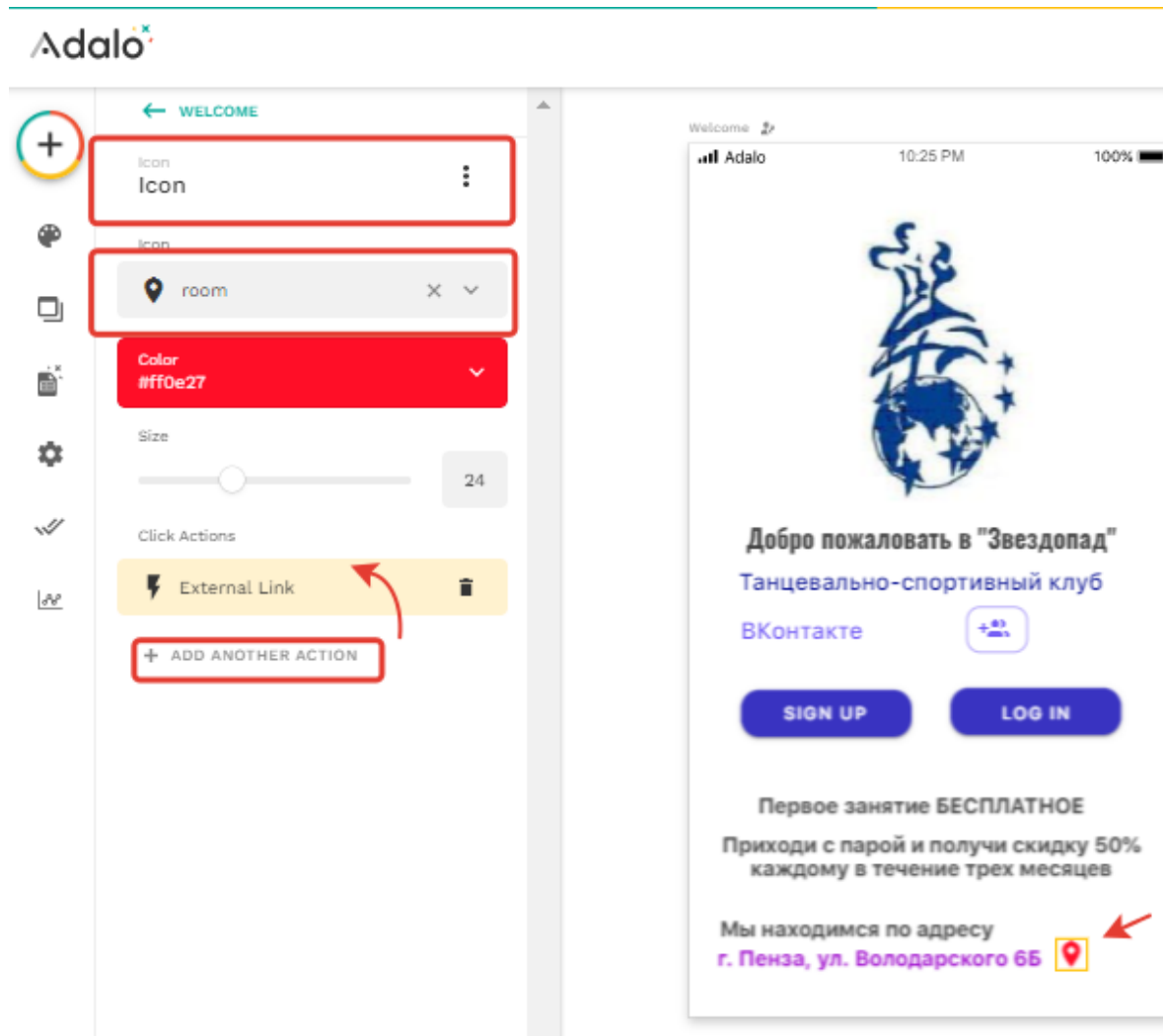


Рис. 4.13. Добавление компонента кнопка в виде Icon

В дальнейшем добавим элементы навигации на экраны:

- TabBar – панель вкладок в нижней части экрана, позволяющая быстро переключаться между разделами приложения;
- AppBar (Android) / NavBar (iOS) – панель инструментов в верхней части экрана, содержащая кнопки управления текущим экраном.

А пока протестируем промежуточный результат. Так как теперь этот экран первый, то именно он отобразится в предварительном просмотре (рис. 4.14). Тестируем, как срабатывают ссылки (текстовые и кнопочные).

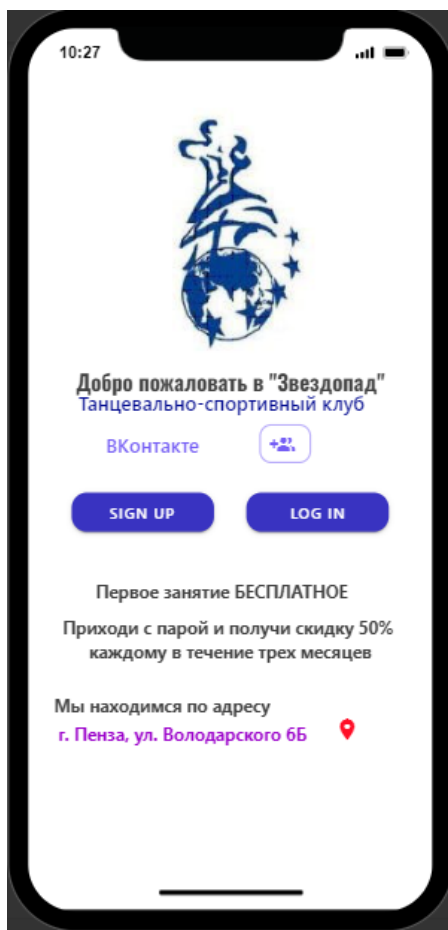


Рис. 4.14. Предварительный просмотр экрана заставки для приложения

Теперь перейдем к настройке трех экранов, которые были добавлены по умолчанию.

Начнем с экранов Login и Signup, войдем во все компоненты этих двух экранов, где используются словосочетания sign up (регистрация) и log in (вход), и перейдем от слитного написания к варианту с пробелом.

Компоненты, которые размещены на экранах входа и регистрации, стандартные, но вполне подходящие. Единственное, что хотелось бы добавить, – это название приложения в верхней части экрана и кнопку Назад, которая возвращает на предыдущий экран.

В разделе компонентов для добавления Simple есть Rectangle (прямоугольник), добавим его на экран Signup и отформатируем (рис. 4.15).

Далее добавляем два текстовых элемента с названием клуба и направлением его деятельности. Для названия клуба настраиваем действие: переход на домашнюю страницу.

Добавляем кнопку Icon в виде стрелки и связываем с ней действие «возвращение на предыдущий экран» (рис. 4.16).

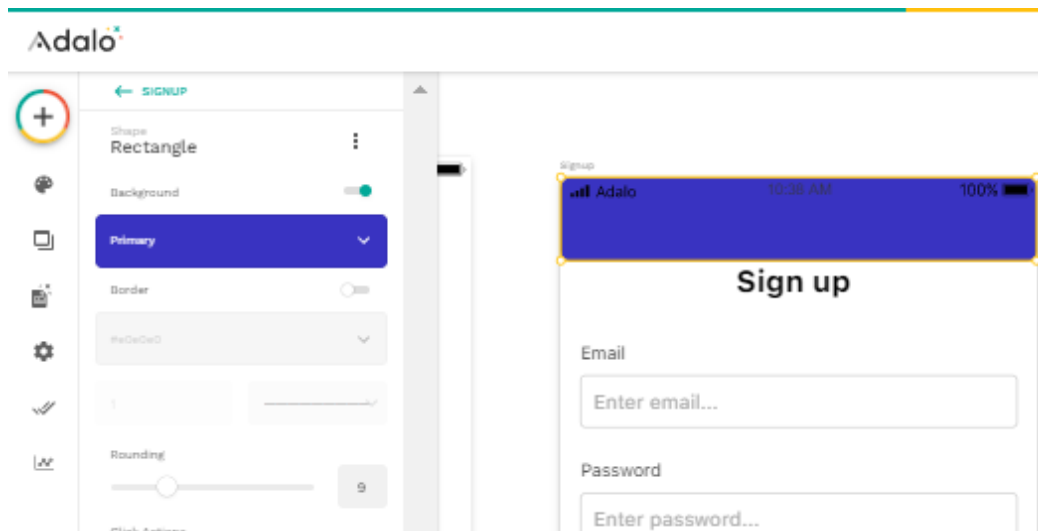


Рис. 4.15. Создание панели с названием приложения (шаг 1)

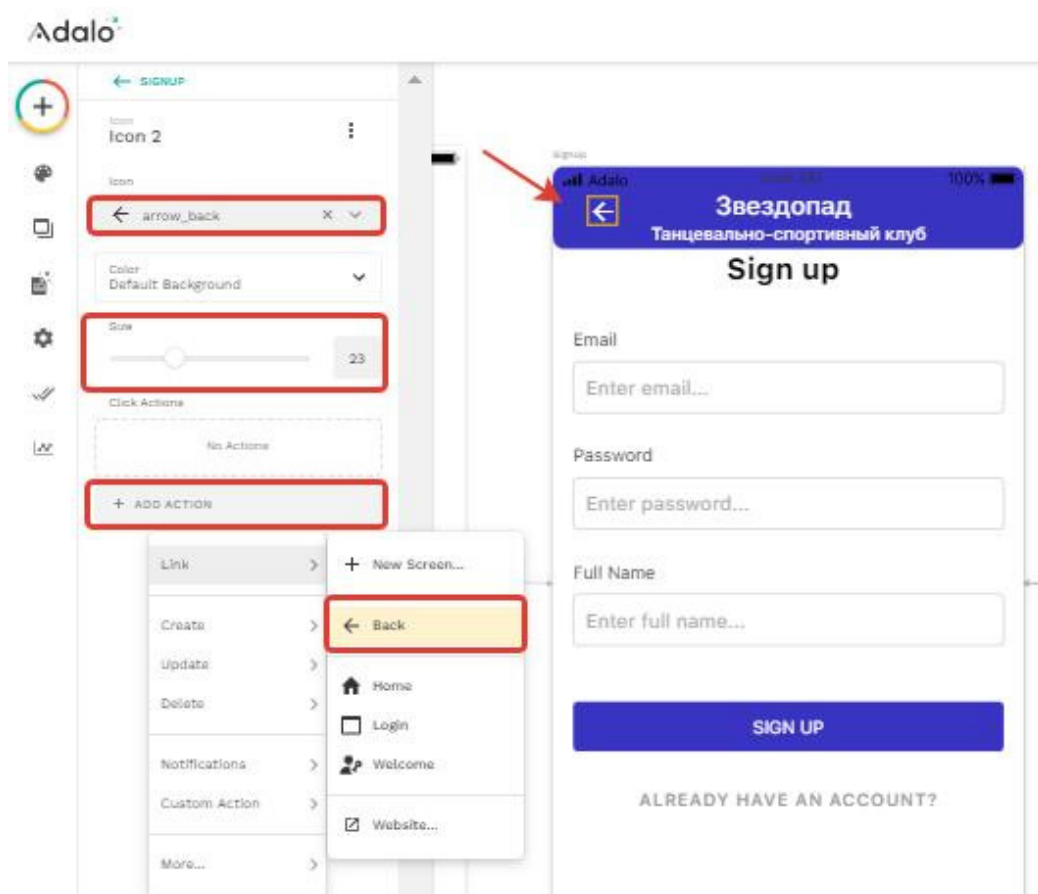


Рис. 4.16. Добавление кнопки на панель названия приложения

Так как настроенная панель подходит для размещения на других экранах (кроме заставки), то сгруппируем элементы панели. Выделим все элементы с помощью мыши, на экране появится меню для выполнения группировки компонентов (рис. 4.17). После группировки скопируем панель на два других экрана Login и Home (на экране Home предварительно удалим компоненты, добавленные по умолчанию).

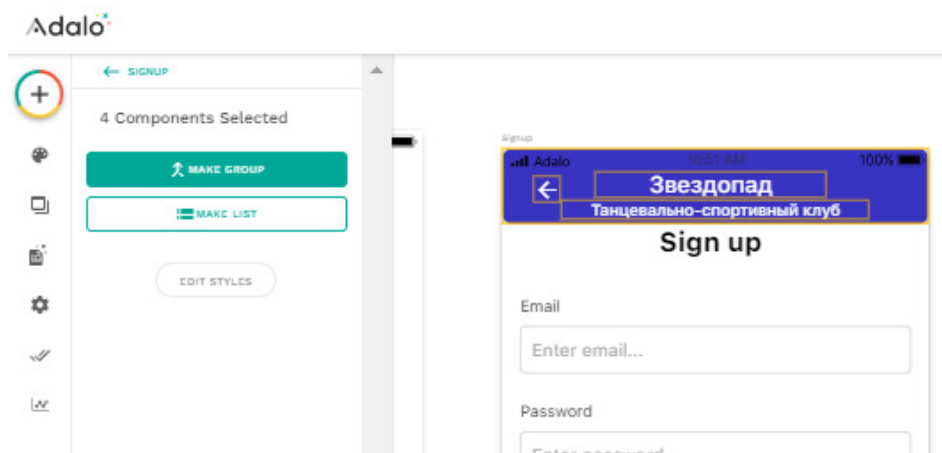


Рис. 4.17. Группировка компонентов

Протестируем работу приложения – теперь с экранов Login и Signup можно возвращаться на первый экран и, кликнув по названию клуба, перемещаться на пока малоинформативную домашнюю страницу.

Перейдем к настройке экрана Home. На этом экране уместна будет информация о тренерах клуба и программах, а также о расписании занятий. Добавим ее.

Начнем с информации о тренерах. Добавим текстовый элемент с названием списка и компонент из раздела Most Used Simple List. Теперь необходимо создать таблицу в базе данных, в которой будет храниться информация о тренерах. Для этого необходимо зайти на вкладку Database в левой панели инструментов и выбрать действие Add collection.

В диалоговом окне добавить и настроить все необходимые свойства/поля коллекции/таблицы Trainer: Name, Photo, Success (рис. 4.18), свойства в дальнейшем можно добавлять по мере необходимости, например, Achievement (рис. 4.19).

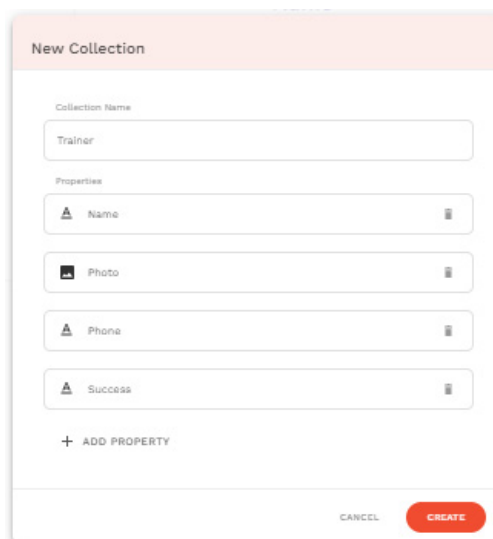


Рис. 4.18. Добавление таблицы данных

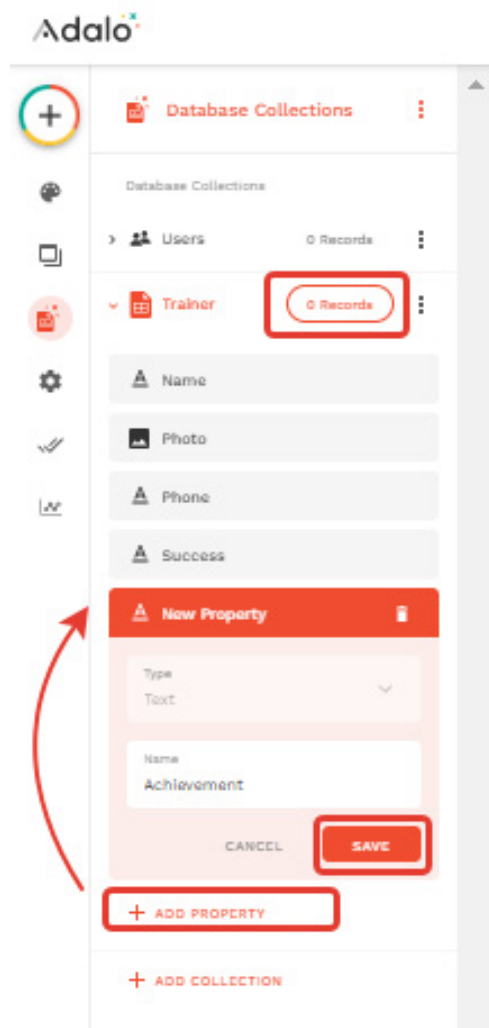


Рис. 4.19. Добавление свойства/поля в таблицу данных

Adalo позволяет заполнять таблицу данными не только через форму в приложении, но и непосредственно в редакторе. Кликнем на кнопку 0 Records рядом с Database Collections Trainer и перейдем к заполнению таблицы данными (рис. 4.20, 4.21). Инструмент, кроме добавления записей, позволяет также загрузить в таблицу данные из файла в формате .csv (стрелка вверх) и выгрузить данные таблицы (стрелка вниз в облаке), если это необходимо. После добавления всех нужных данных сохраняем введенные записи, кликнув на Done.

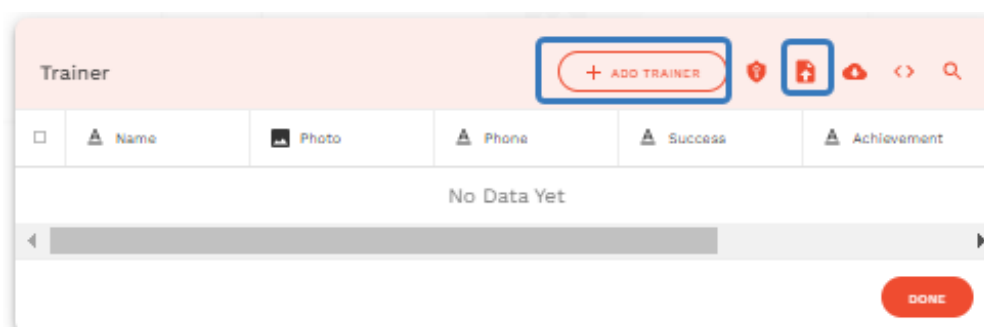


Рис. 4.20. Добавление записей

Trainer					
+ ADD TRAINER					
☐	Name	Photo	Phone	Success	Achievement
☐	Топалочкина Валерия		89093330665	Кандидат в мастера сп...	Трёхкратный чемпион ...
☐	Гребнев Сергей		89677024321	Кандидат в мастера сп...	Трёхкратный чемпион ...

Рис. 4.21. Добавление записей

Теперь можно настроить добавленный компонент список на экране Home (рис. 4.22). Выбираем из двух коллекций ту, что создали недавно и переходим к дальнейшей настройке списка.

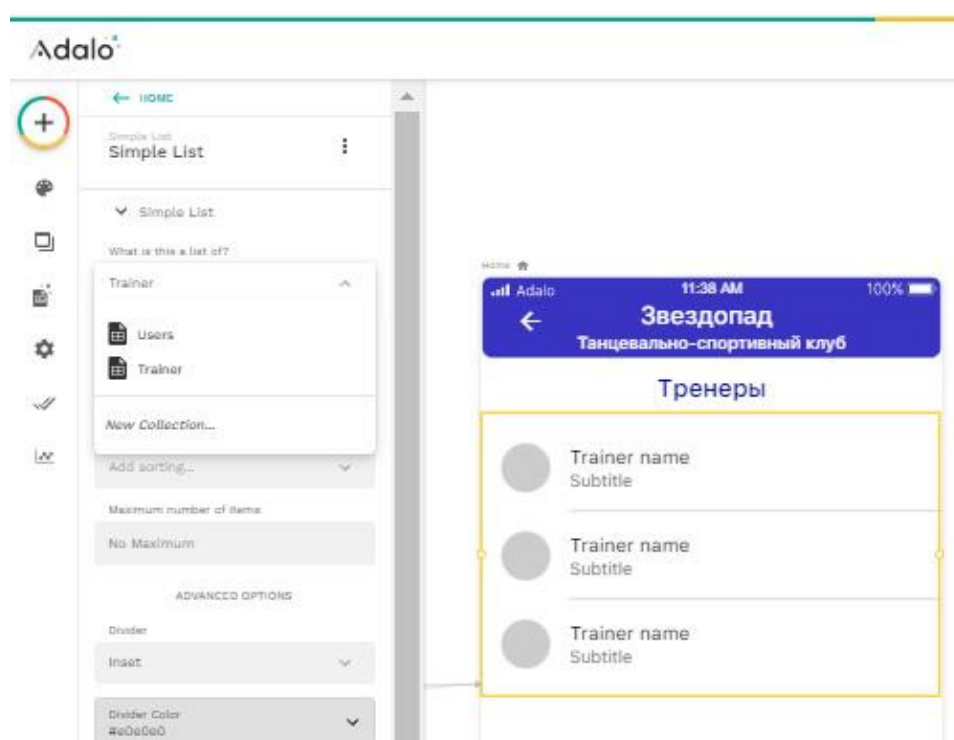


Рис. 4.22. Выбор таблицы данных для построения списка

Заголовок компонента список сразу установится верный, а вот настройкой всего остального для списка необходимо заняться самостоятельно. В качестве подзаголовка будет использоваться столбец Success. Если необходимо отобразить значение поля из таблицы данных в компоненте, используют опцию Magic Text (рис. 4.23), она позволяет выбрать необходимое поле из списка.

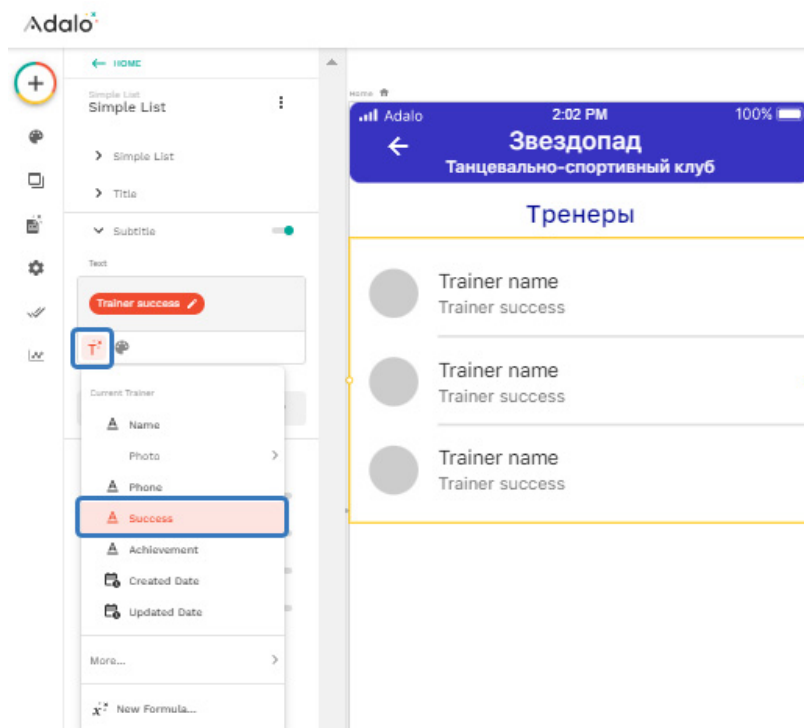


Рис. 4.23. Настройка Subtitle для списка

Также проверим, что изображение для аватара забирается из верного столбца таблицы (рис. 4.24).

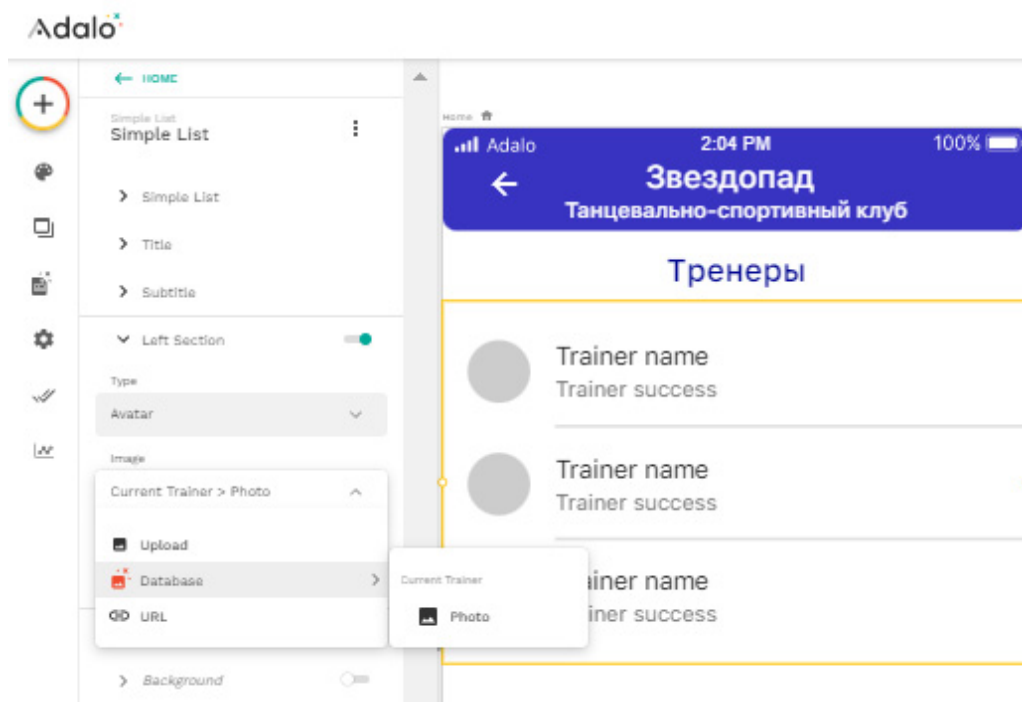


Рис. 4.24. Настройка левой секции списка

Мы получаем список тренеров, но хотелось бы на отдельном экране увидеть развернутую информацию о тренерах. Добавим экран Trainer, на который сразу скопируем верхнюю панель. Теперь добавим к списку тренеров на домашней странице действие – переход по ссылке на страницу Trainer (рис. 4.25).

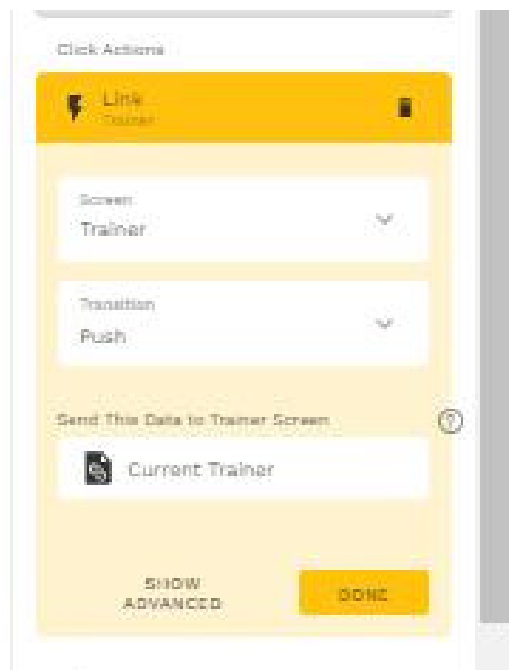


Рис. 4.25. Настройка действия для списка

Обратите внимание, что сразу подключается передача данных Current Trainer (текущий тренер, т.е. тот, кого мы выбрали из списка). Если действие не настроить, экран Trainer не будет знать, информацию о каком тренере нужно отображать; более того, эта коллекция вообще не будет доступна для выбора.

Переходим к настройке элементов экрана Trainer. Добавляем текстовый компонент и настраиваем для него Magic Text (рис. 4.26), связывая компонент с полем Trainer name, форматируем этот элемент обычным образом (размер текста, цвет, выравнивание).

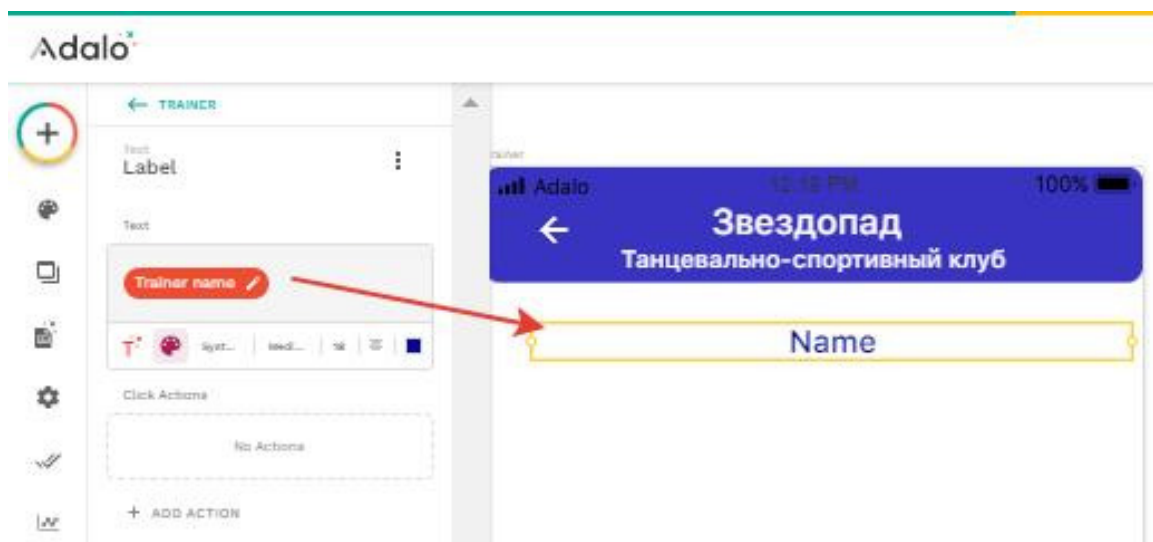


Рис. 4.26. Настройка текстового компонента

Добавляем компонент изображения и настраиваем Image Source: Trainer Photo (рис. 4.27).

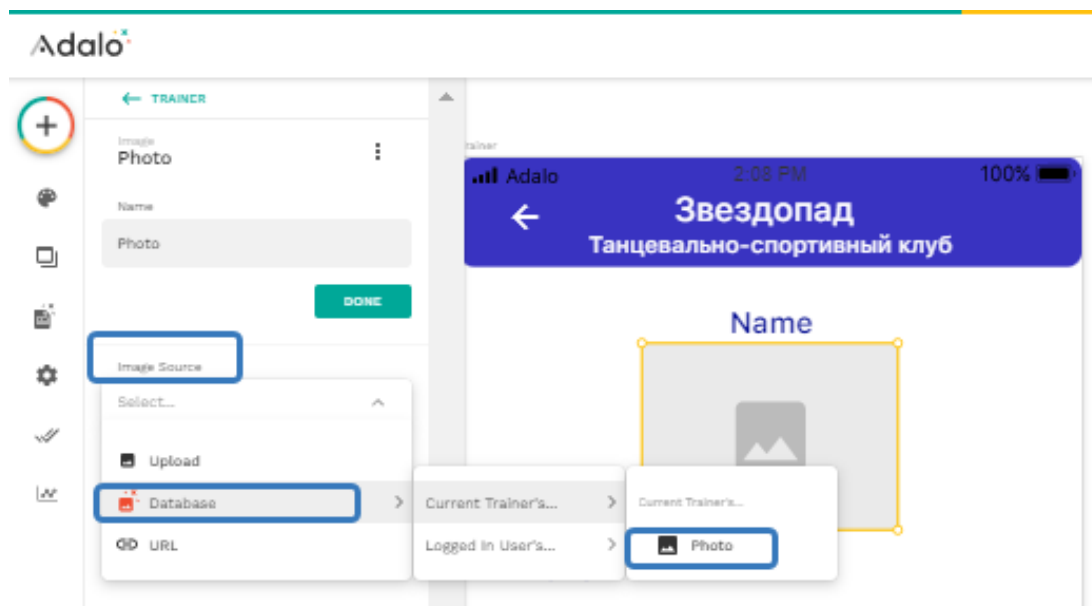


Рис. 4.27. Добавление фотографии на экран

Разместим еще три текстовых компонента на экране, продублировав Name и изменив ресурсы (поля), на которые ссылается Magic Text. Проведем редактирование стиля отображения фотографии: размер 100×100, скругление 50 (рис. 4.28).

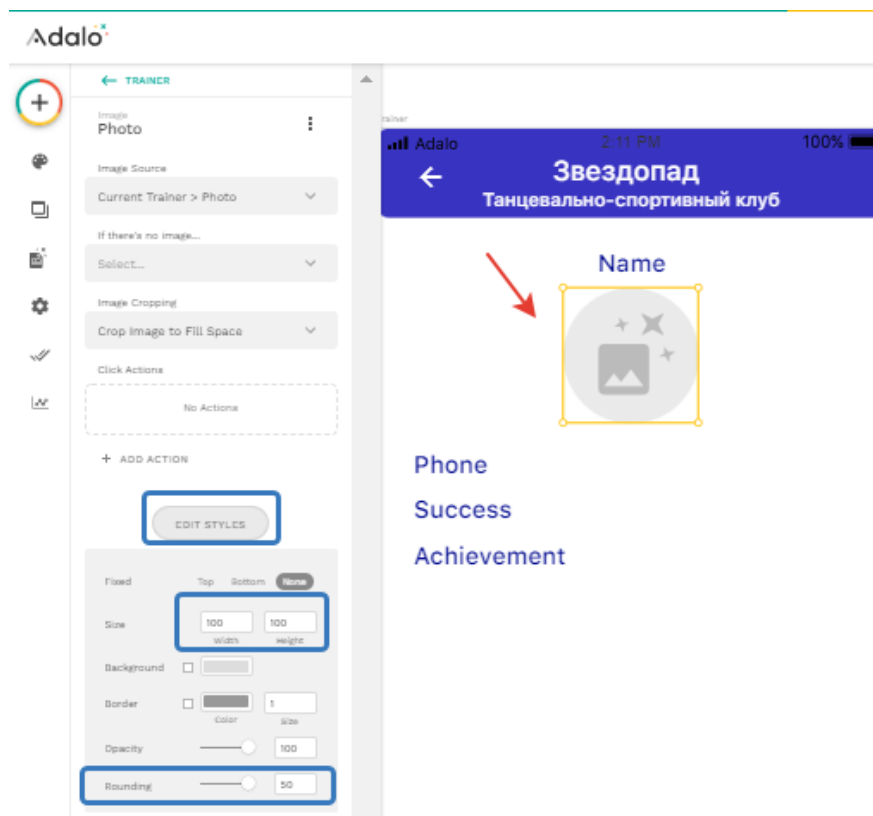


Рис. 4.28. Редактирование стиля изображения

Список всех элементов экрана доступен на вкладке экраны при переходе на конкретный экран в списке (рис. 4.29).

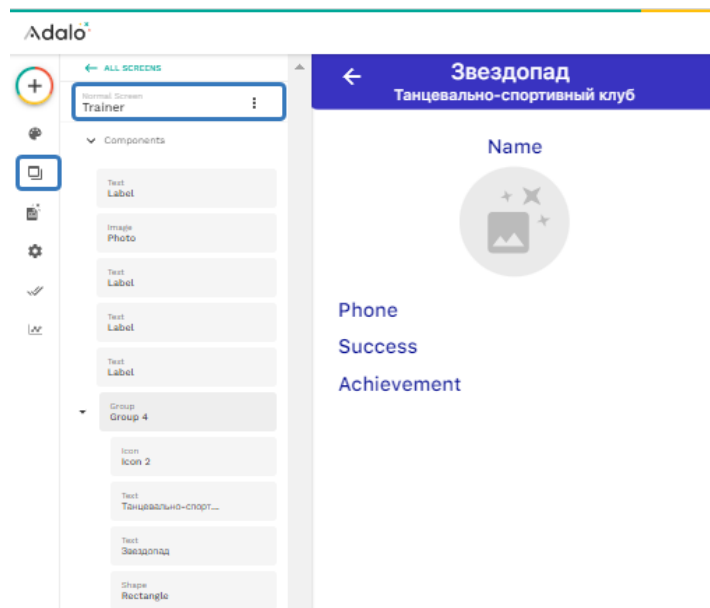


Рис. 4.29. Просмотр списка компонентов экрана

Протестируем с помощью предпросмотра, как выглядит список тренеров на экране Home (рис. 4.30) и страницы тренеров (рис. 4.31).

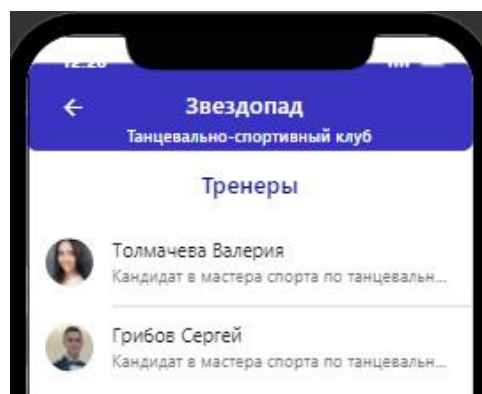


Рис. 4.30. Экран Home

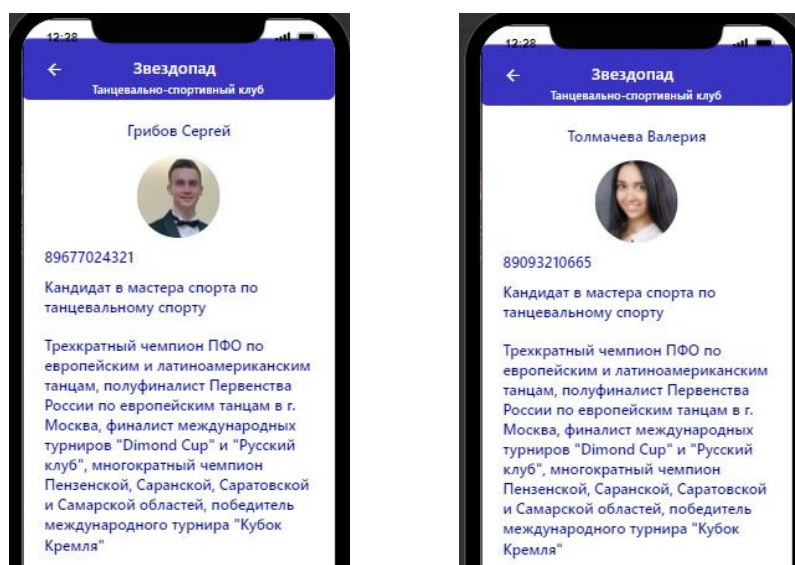


Рис. 4.31. Страницы тренеров

Добавим статичную текстовую информацию о том, какие программы танцев входят в подготовку участников клуба. И добавим новый список Card List, который находится в разделе Lists добавления компонентов (рис. 4.32).

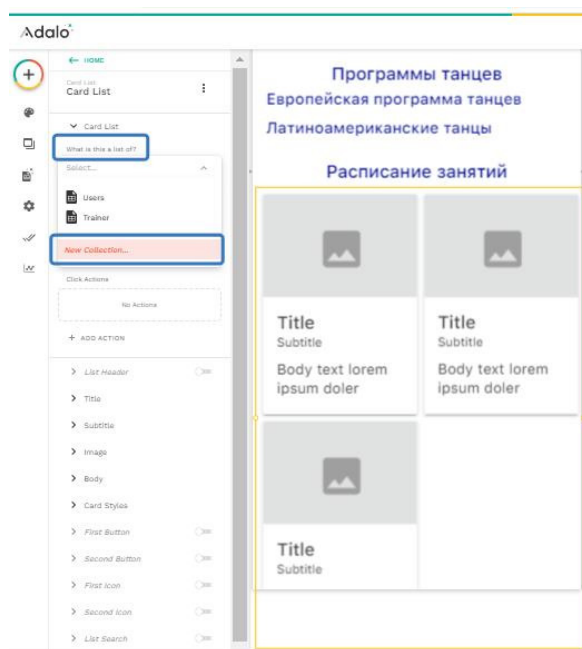


Рис. 4.32. Добавление списка расписания занятий

Для размещения списка нужно увеличить рабочую область экрана: выделим экран кликом по его названию и протянем вниз нижнюю границу экрана. Список должен отображать данные новой коллекции Schedule, которую создадим аналогично коллекции Trainer. В коллекцию Schedule добавим свойства: Group, Image, Day1, Day2, Day3 (рис. 4.33), после чего добавим в нее нужные данные (рис. 4.34).

Рис. 4.33. Создание новой коллекции

Schedule					
+ ADD SCHEDULE					
☐	Group	Image	Day1	Day2	Day3
☐	Старшая группа		Понедельник 18.00-19.00	Среда 18.00-19.00	Пятница 18.00-19.00
☐	Младшая группа		Вторник 18.00-19.00	Четверг 18.00-19.00	Суббота 18.00-19.00

Рис. 4.34. Добавление записей в новую коллекцию

У списка Card List есть заголовок. Настроим его (рис. 4.35), а дублирующий его текстовый компонент уберем.

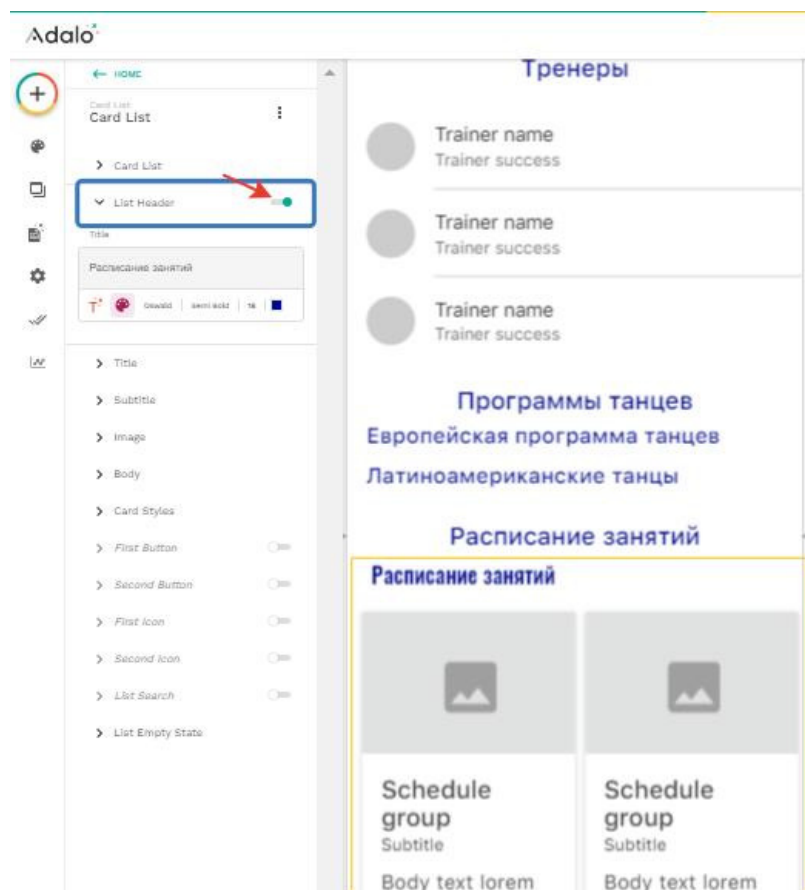


Рис. 4.35. Настройка заголовка для списка

Элемент Body отключим, а в элемент Subtitle добавим три компонента Magic Text: Schedule day1, Schedule day2, Schedule day3, добавляя после каждого элемента пробел с клавиатуры, чтобы текстовые значения не склеились при отображении (рис. 4.36).

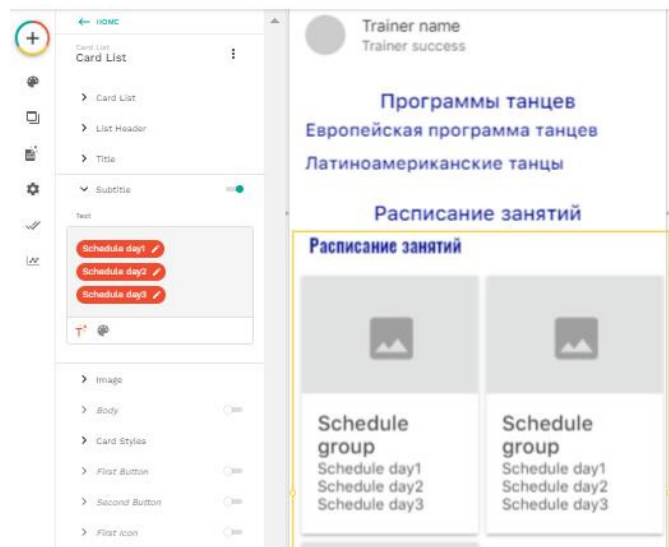


Рис. 4.36. Настройка списка

Протестируем полученный результат (рис. 4.37). Как видим, расписание выглядит неаккуратно.

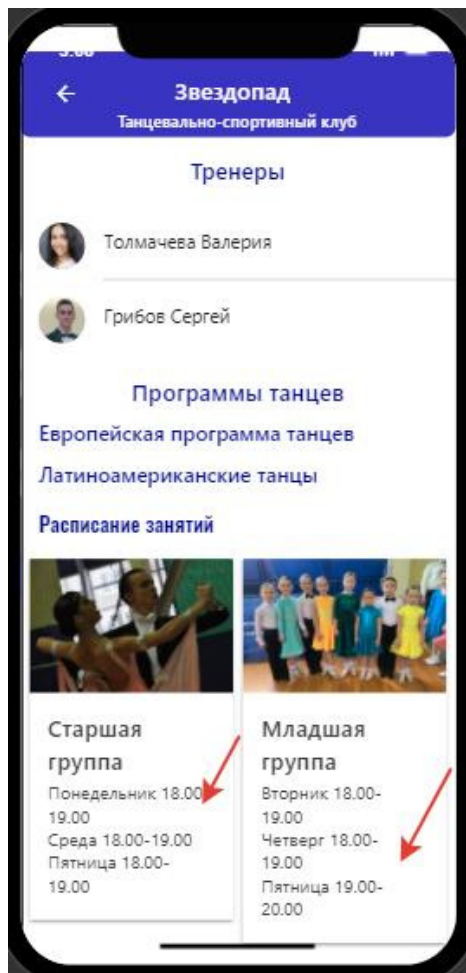


Рис. 4.37. Предварительный просмотр

Запись о дне и времени не помещается в строке, а перенос осуществляется не по пробелу между элементами внутри одной ячейки

таблицы. Эту проблему можно решить включением в коллекцию Schedule дополнительных свойств Time Day1, Time Day2, Time Day3 (рис. 4.38) и отредактировав записи в таблице (рис. 4.39).

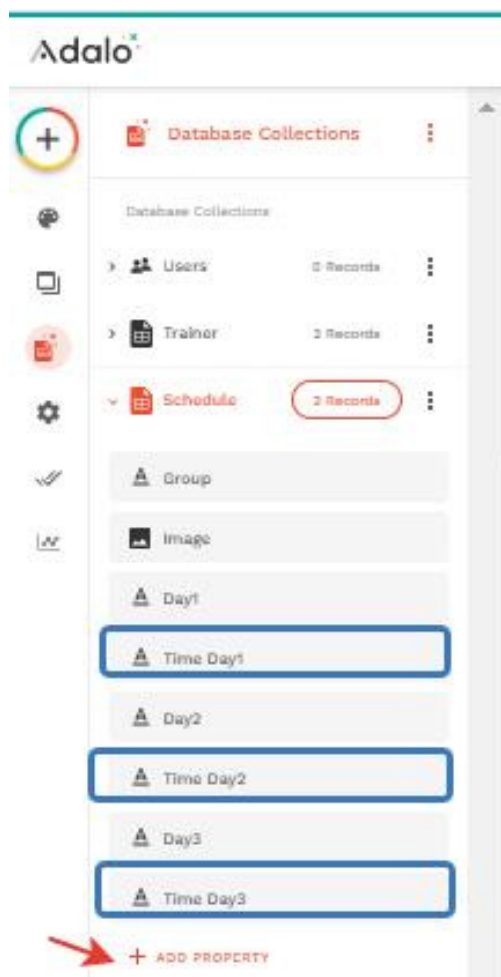


Рис. 4.38. Добавление новых свойств

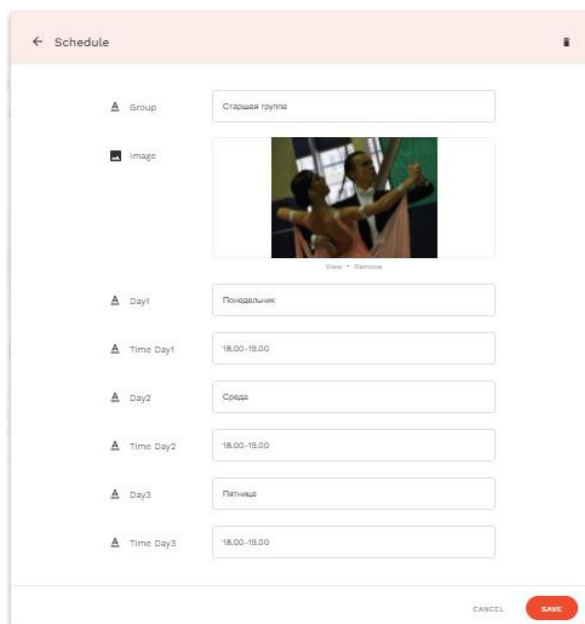


Рис. 4.39. Редактирование записей

Скорректируем настройки Subtitle, добавив новые элементы Magic Text: Schedule time day1, Schedule time day2, Schedule time day3, не забывая про пробелы (рис. 4.40).

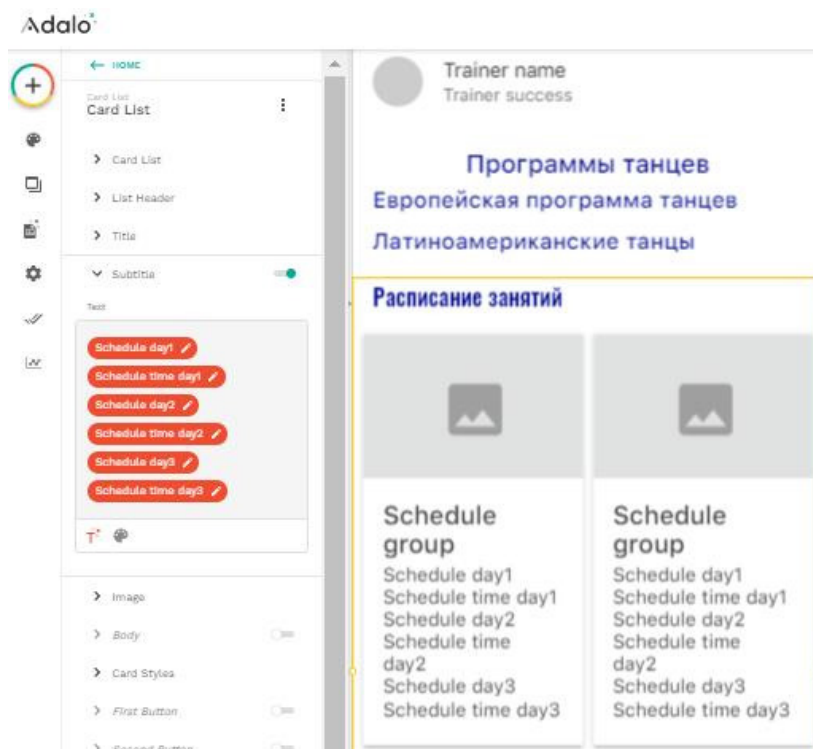


Рис. 4.40. Обновление настроек для списка
И протестируем результат (рис. 4.41).

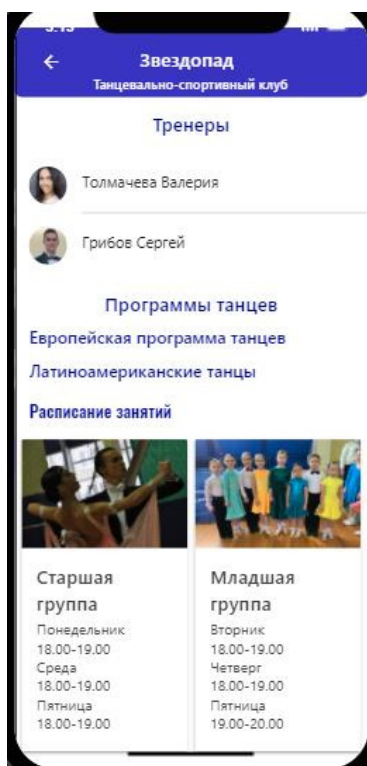


Рис. 4.41. Экран Home

Хотя вся информация поместилась на одном экране, тем не менее рассмотрим настройки, которые позволяют при прокрутке экрана оставлять на месте информационную панель или TabBar. Для этого необходимо кликнуть кнопку для группы компонентов Edit Styles и установить значение Fixed, например top (рис. 4.42).

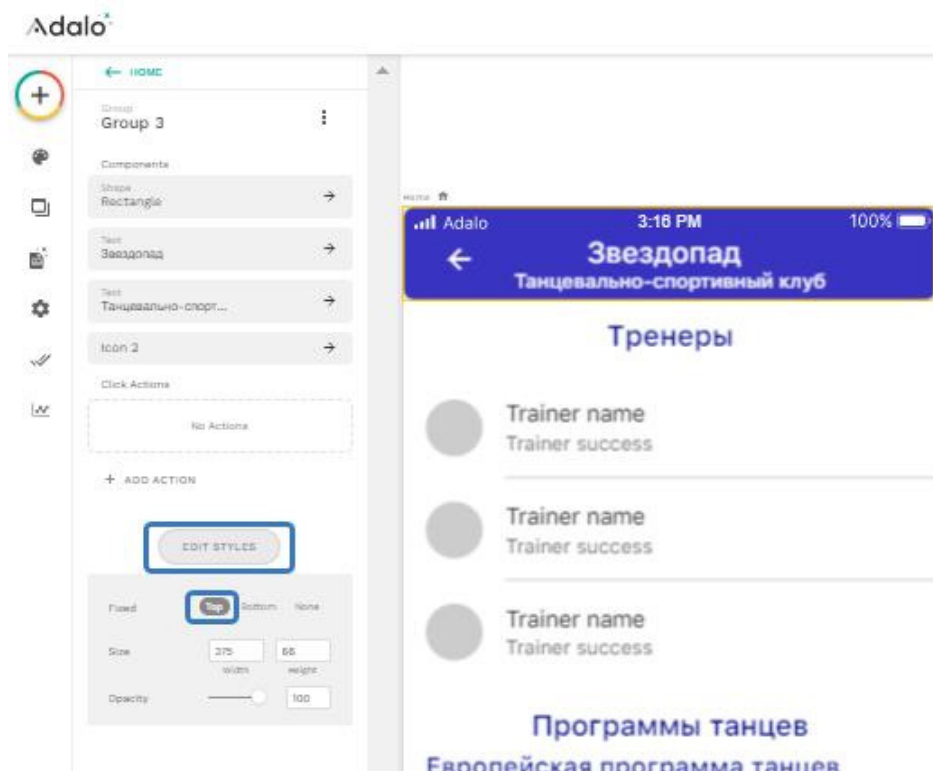


Рис. 4.42. Закрепление информационной панели

Добавим для зарегистрированных пользователей возможность создавать посты, которые будут отображаться на общем экране постов в приложении.

Нам понадобится экран, на котором будет список постов, отсортированных по дате добавления от недавних к более ранним, а также экран создания поста. Так как посты будут добавляться в приложение, то их необходимо хранить в базе данных. Следовательно, требуется добавить новую коллекцию Posts. Особенность добавления collection Posts заключается в наличии связи с collection Users, так как наши посты создают именно пользователи.

На вкладке Database добавим новую коллекцию Posts, а в нее добавим Property Relationship (связь) с коллекцией Users (рис. 4.43).

Свойство типа Relationship требует настройки типа связи между двумя таблицами или кардинальности, определяющий, сколько экземпляров сущности Users можно поставить в соответствие какому-то количеству экземпляров сущности Posts. Все варианты визуализируются в диалоговом окне (рис. 4.44). Так как один пользователь может

создать несколько постов, но каждый пост может принадлежать только одному пользователю, то выбираем первый вариант. Настроенное свойство необходимо сохранить (рис. 4.45). Особенность добавления такого свойства в коллекцию в том, что не только поле Users появится в таблице Posts, но и поле Posts автоматически присоединится к таблице Users (рис. 4.46).

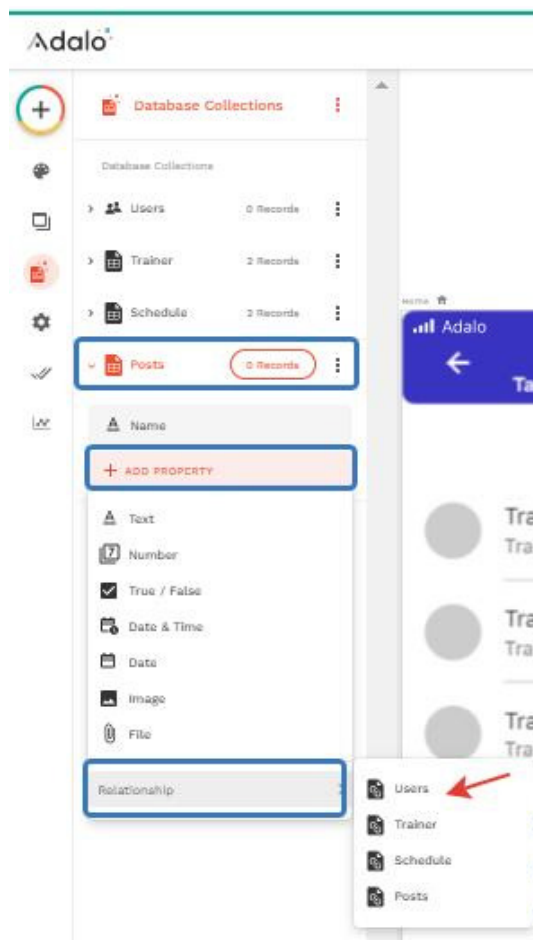


Рис. 4.43. Добавление новой коллекции

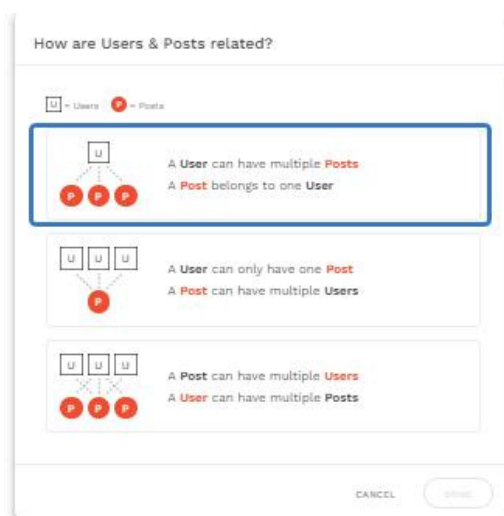


Рис. 4.44. Настройка кардинальности связи

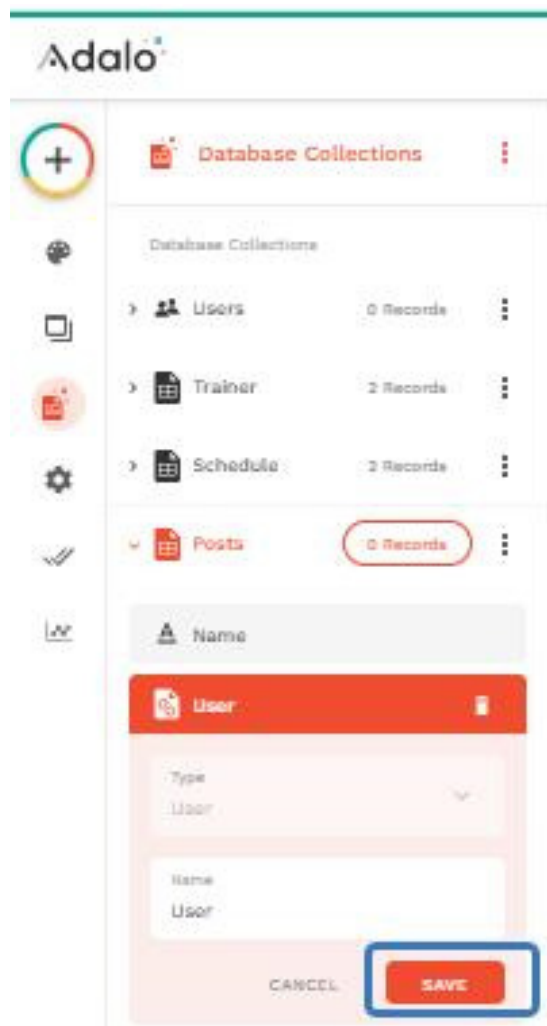


Рис. 4.45. Сохранение свойства

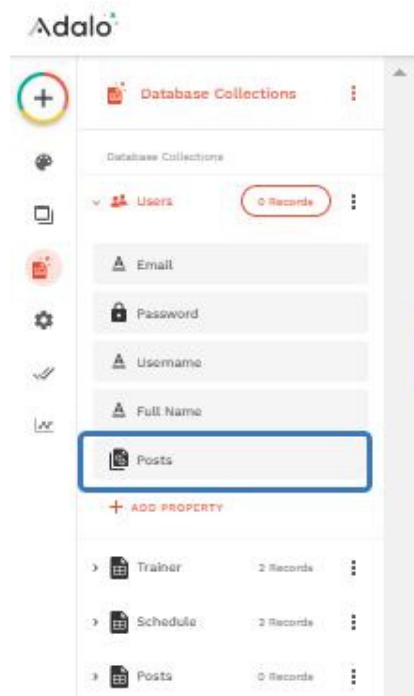


Рис. 4.46. Состав свойств коллекции Users

Изменим состав свойств коллекции Posts: Title, Image, Text (рис. 4.47).

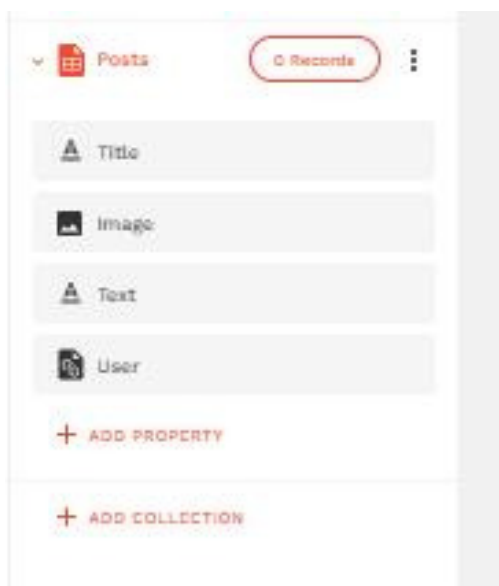


Рис. 4.47. Состав свойств коллекции Posts

На два добавленных на холст экрана: Posts и Add post добавим верхнюю панель и текстовый элемент – название экрана.

На экран добавления поста добавим компонент Form из раздела Most Used (рис. 4.48) и выберем коллекцию, с которой будут связаны поля формы.

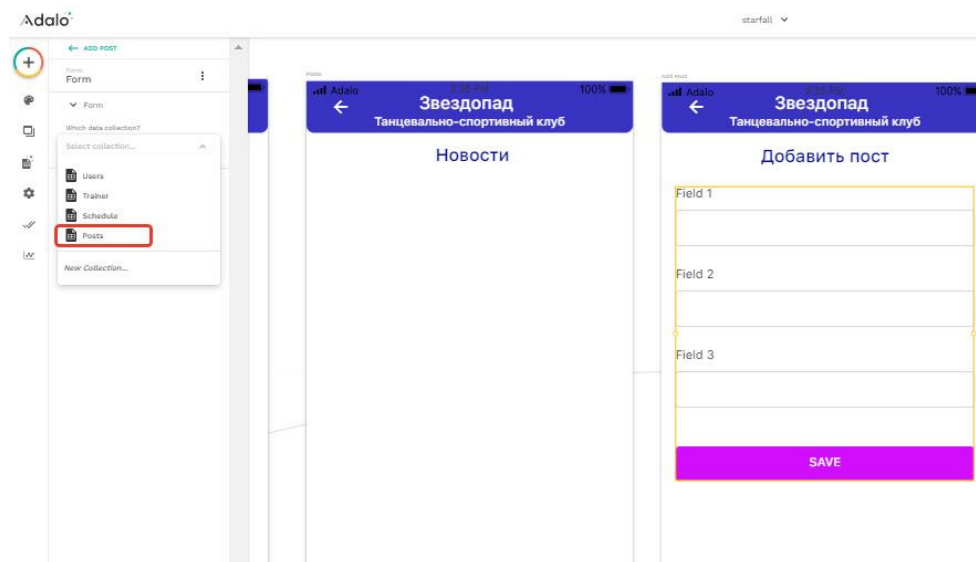


Рис. 4.48. Настройка формы

Вид формы после настройки связи с коллекцией будет вполне удовлетворительный. Единственное, добавим еще одно действие кнопке Create post. Пусть по клику этой кнопки не только создается новый пост, но и осуществляется переход на экран Posts, где отображаются все посты (рис. 4.49).

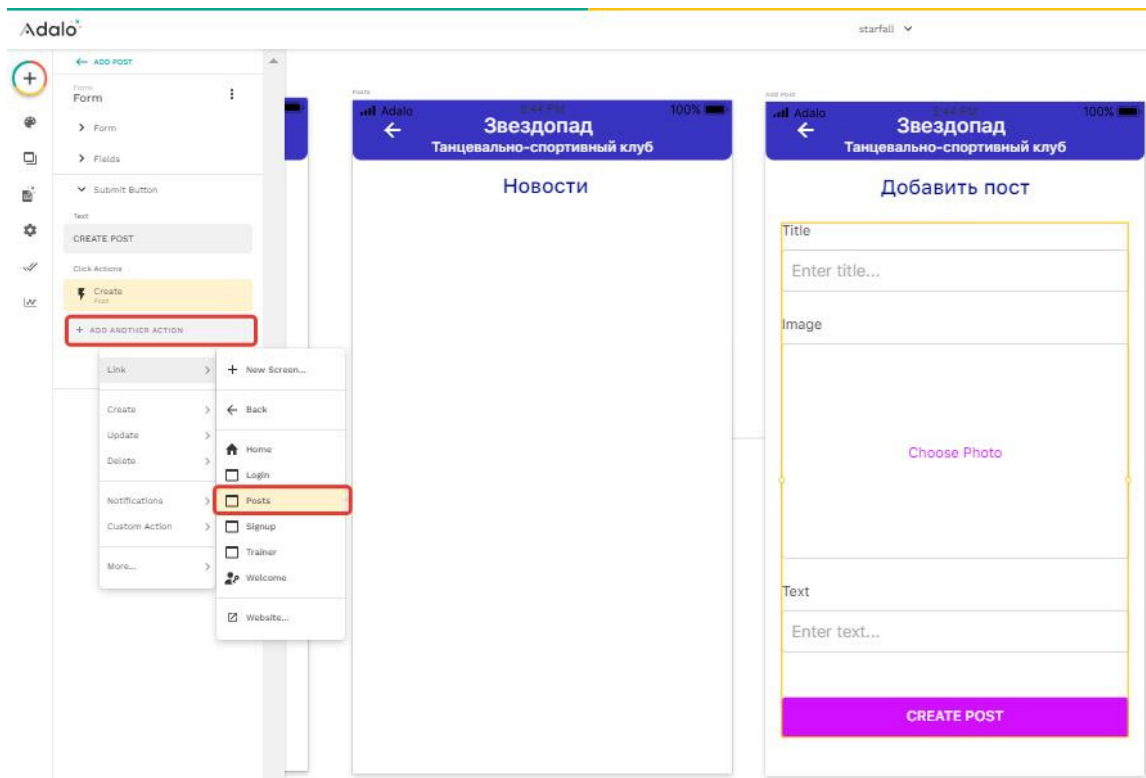


Рис. 4.49. Добавление еще одного действия кнопке

Добавим на экран Posts Image List (рис. 4.50) и настроим для него коллекцию Posts (рис. 4.51).

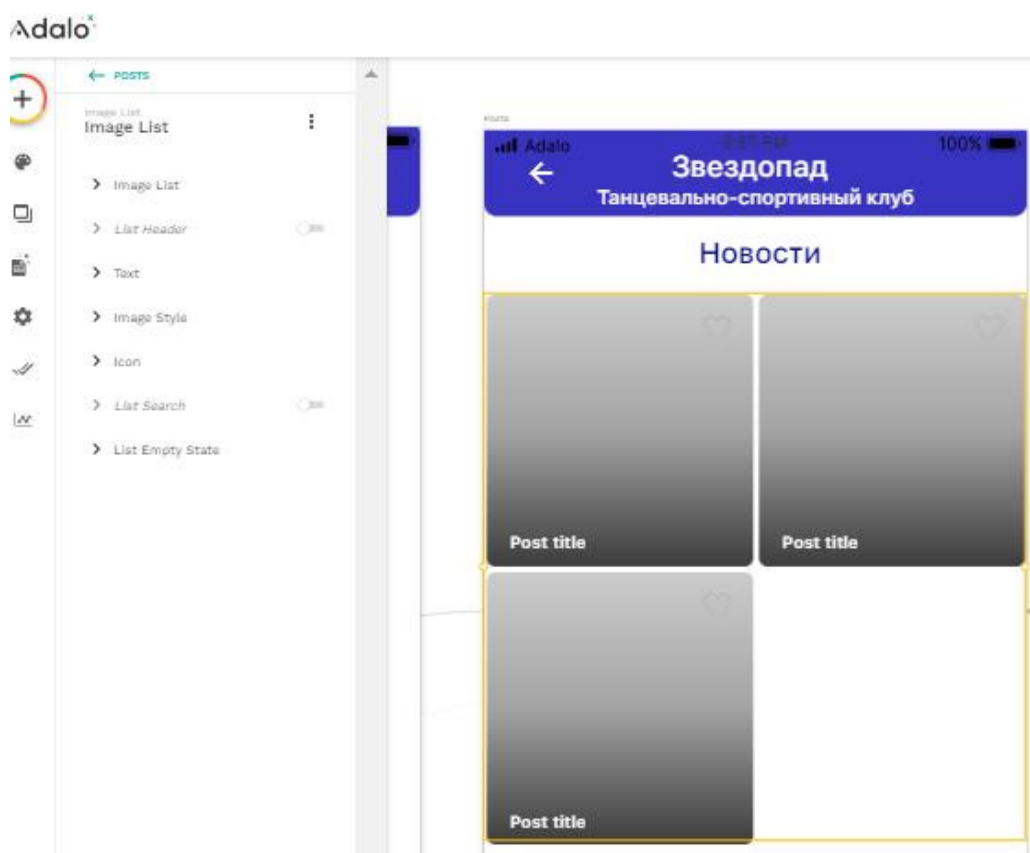


Рис. 4.50. Экран Posts

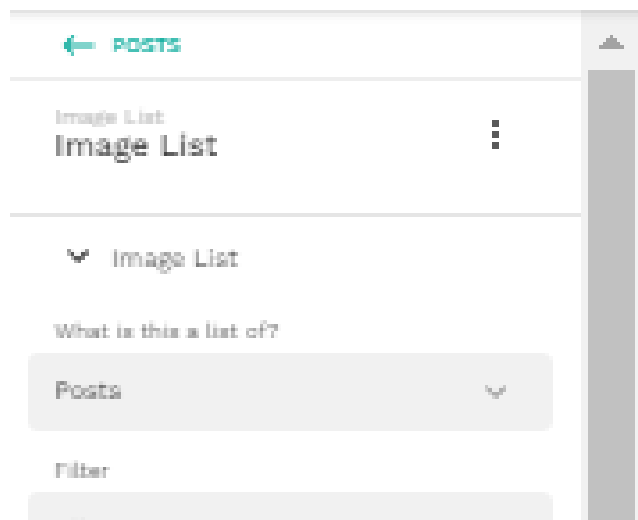


Рис. 4.51. Настройка коллекции для Image List

Настроим на отдельных изображениях в Image List в разделе Icon Toggle (переключатель) в виде сердечка (рис. 4.52).

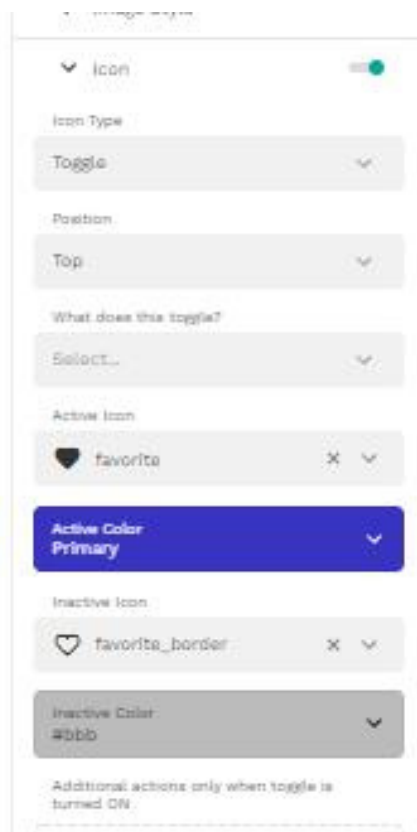


Рис. 4.52. Настройка переключателя для изображений

Добавим еще один экран Profile и займемся добавлением TabBar на экраны, так как на многие экраны можно попасть только из нижнего меню вкладок.

Добавим компонент TabBar из раздела Navigation на экран Home. Настроим каждую вкладку: изменим иконки, подписи под ними на подходящие, для всех вкладок (кроме первой Home) добавим действие —

переход по ссылке на соответствующий вкладке экран, отключим пятую вкладку, так как в нашем приложении в ней нет необходимости. Обратите внимание, что First Tab для экрана Home является Active Tab (для других экранов активные вкладки будут другие). Для фиксации положения нижнего меню на экране (чтобы при прокрутке оно оставалось доступным внизу экрана) нажмем на три точки рядом с компонентом во вкладке его настройки (такое меню для элемента носит название more...) и выберем опцию Make Group (рис. 4.53). Зафиксируем положение группы на экране в позиции Bottom (рис. 4.54).

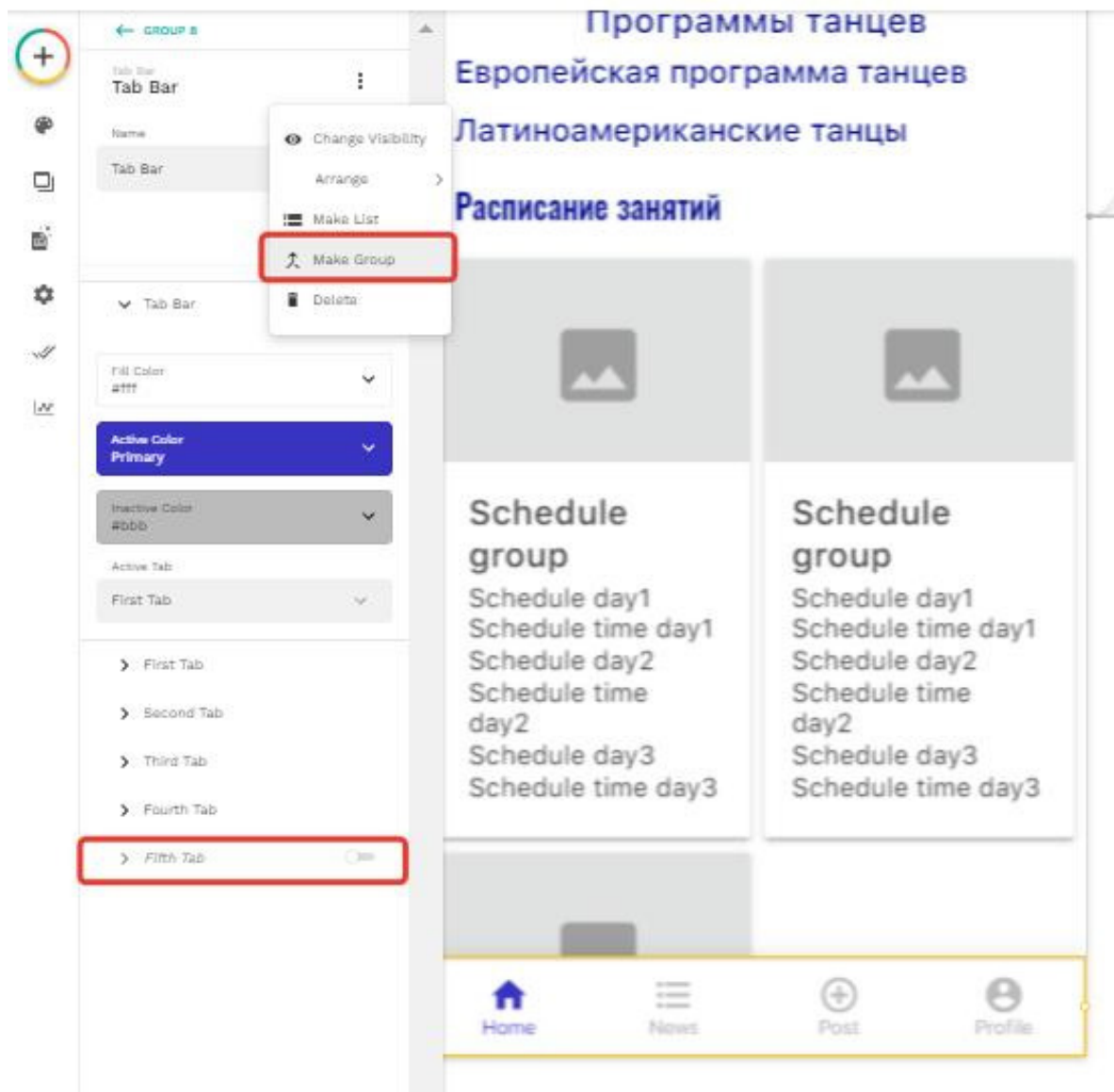


Рис. 4.53. Переход к настройкам для группы

Скопируем TabBar на экраны Posts, Add post, Profile и изменим его настройки: установим значение Active Tab на соответствующей этому экрану вкладке и добавим действие кнопке First Tab. Настройки для TabBar на экране Posts представлены на рис. 4.55.

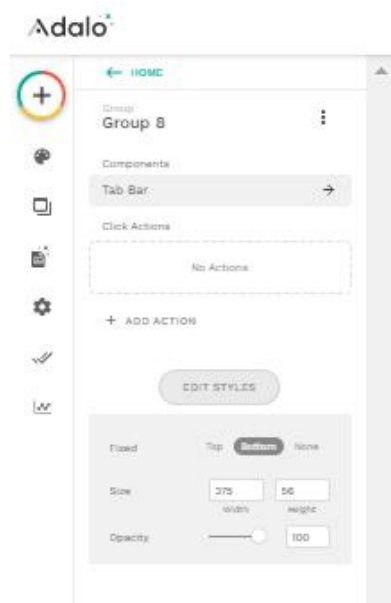


Рис. 4.54. Fixed Bottom

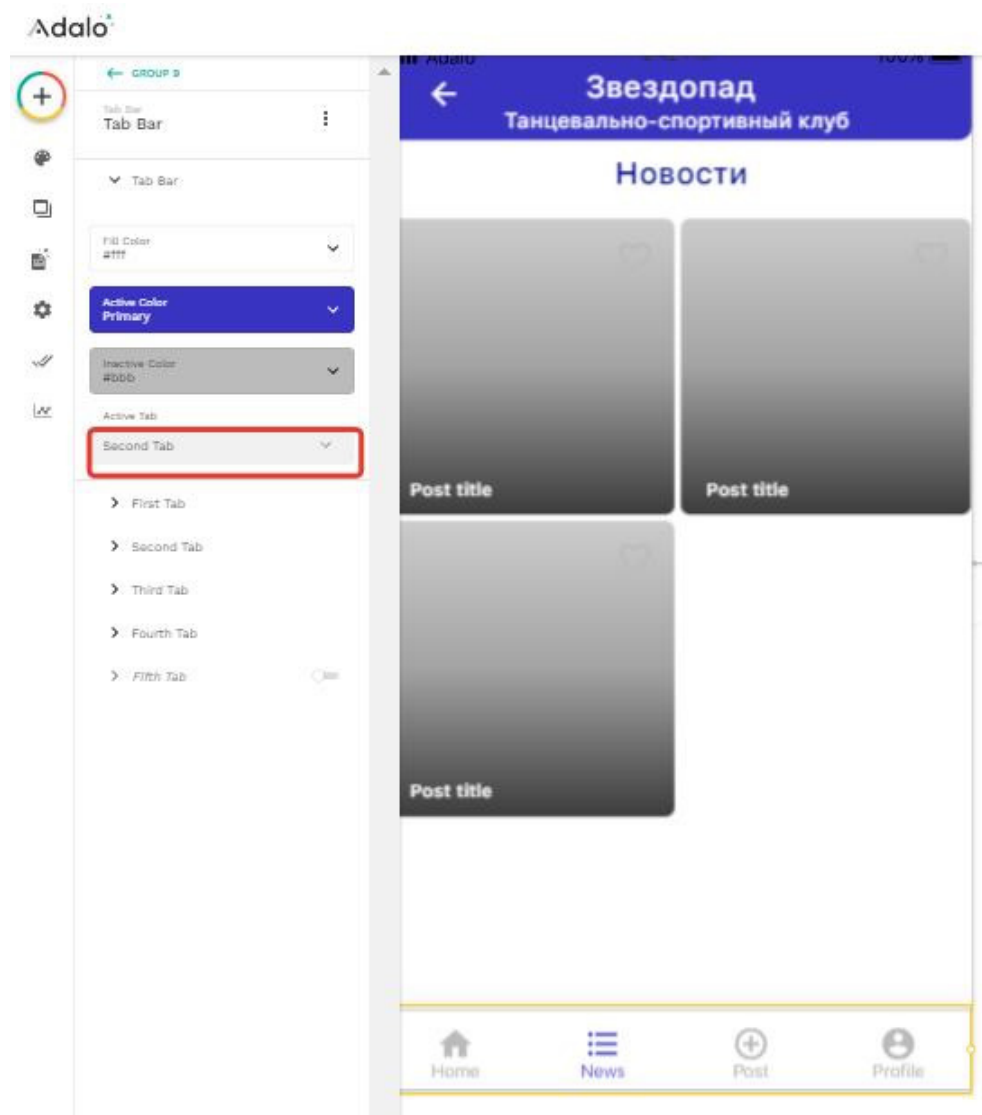


Рис. 4.55. Настройки TabBar на экране Posts

Проведем настройку экрана Profile. Информация, которую сообщают о себе пользователи при регистрации, довольно скудная. Форма регистрации и должна содержать только самые необходимые данные, чтобы не отпугнуть пользователя масштабом работы. А вот в личном кабинете можно предоставить возможность заполнить дополнительные сведения о себе. И прежде всего для обеспечения этого нужно дополнить коллекцию Users новыми свойствами: Phone, Photo, About, Birthday (рис. 4.56).

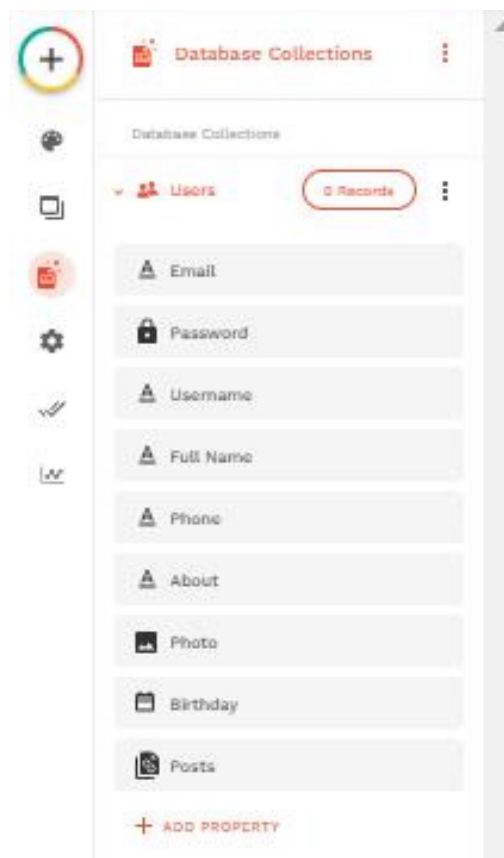


Рис. 4.56. Добавление свойств в коллекцию Users на вкладке Database

Добавим на экран текстовый компонент Log Out, текстовый компонент с Magic Text: Users Full Name (для отображения полного имени зарегистрированного пользователя) и изображение, которое настроим согласно рис. 4.57. В качестве Image Source выбираем фотографию зарегистрированного пользователя, а для настроек размеров изображения раскрываем дополнительные опции, кликнув на Edit Style (размер 100×100, скругление 50).

Также добавим на экран кнопку Edit Profile, а для настройки действия, связанного с этой кнопкой, создадим еще один экран Edit Profile (в этом окне разместим в дальнейшем форму редактирования профиля, а пока добавим верхнюю панель и текстовый компонент). Действие, добавленное кнопке Edit Profile, – переход на одноименный экран.

Добавим на экран Profile список постов, которые создал пользователь (рис. 4.58). Для такого же типа списка, который был размещен на экране Posts: Image List, мы не только настроим связь с коллекцией Post, но и добавим фильтр: посты зарегистрированного пользователя. Теперь в личном кабинете пользователя будут отображаться только его посты. Убираем Subtitle и проверяем настройку изображений в списке.

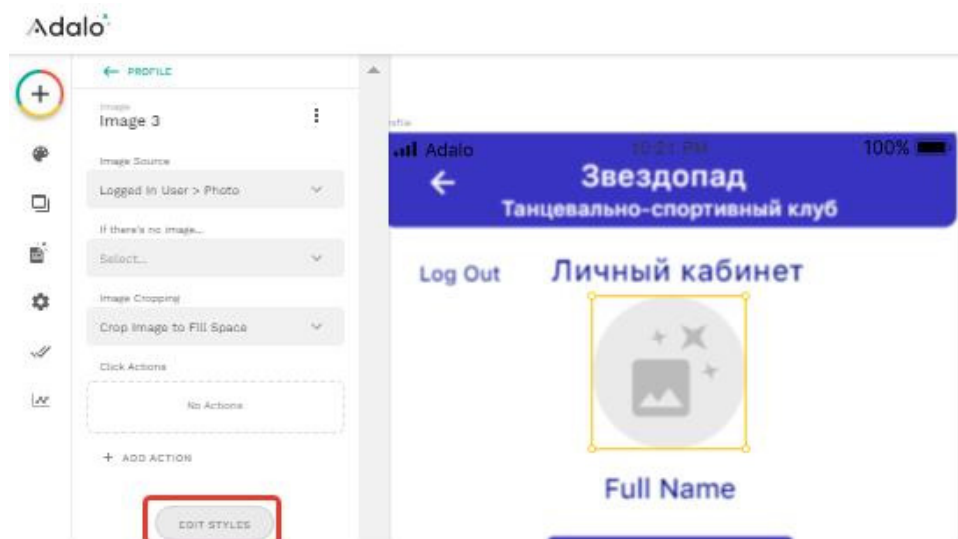


Рис. 4.57. Добавление элементов на экран Profile

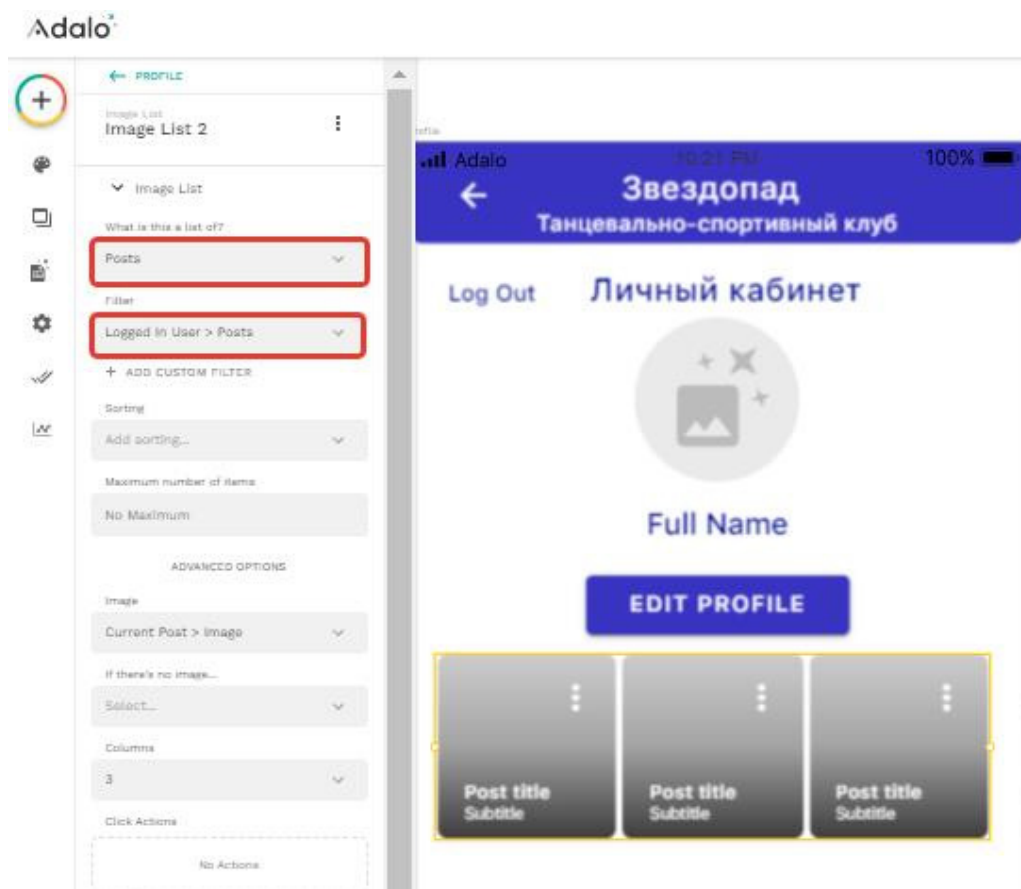


Рис. 4.58. Добавление на экран списка постов

Настраиваем действие для текстового элемента Log Out, чтобы пользователь мог покинуть приложение (рис. 4.59). Цепочка переходов: More...>User Login> Log Out.

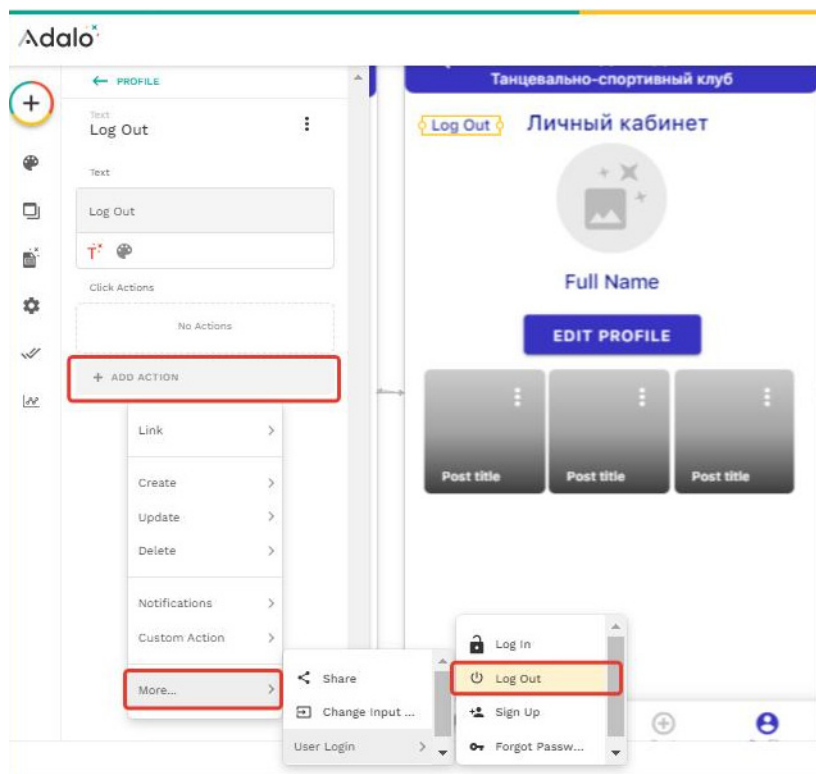


Рис. 4.59. Настройка действия для текстового элемента Log Out

Теперь настроим экран Edit Profile. На этот экран необходимо добавить из раздела Most Used компонент Form и провести его настройку (рис. 4.60).

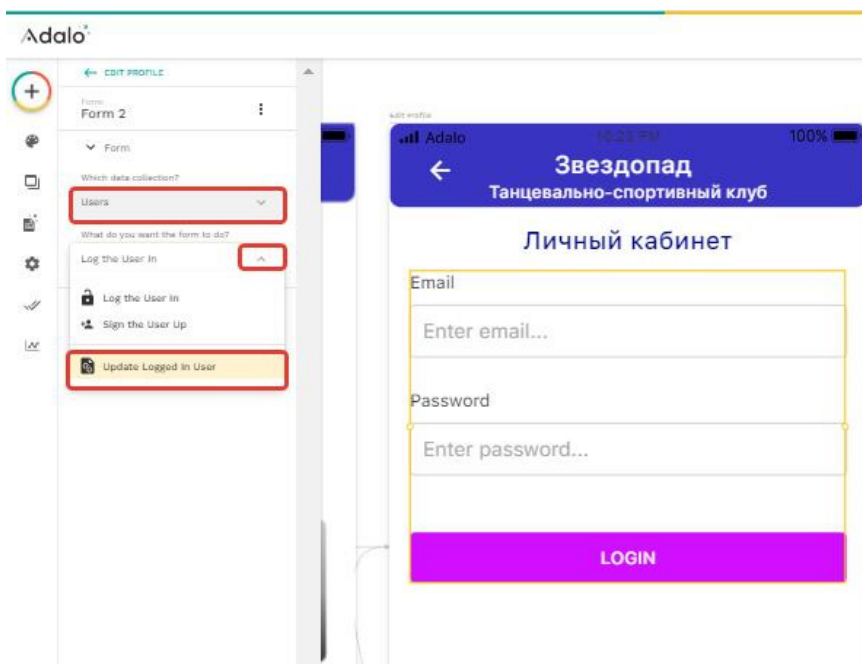


Рис. 4.60. Настройка формы

После выбора значения Update Logged in User в форму будут добавлены все поля из коллекции Users. Так как вся форма не помещается на экране – выделим экран и, подцепив мышкой нижнюю границу, увеличим его. Далее перейдем к настройке полей Form Fields (рис. 4.61).

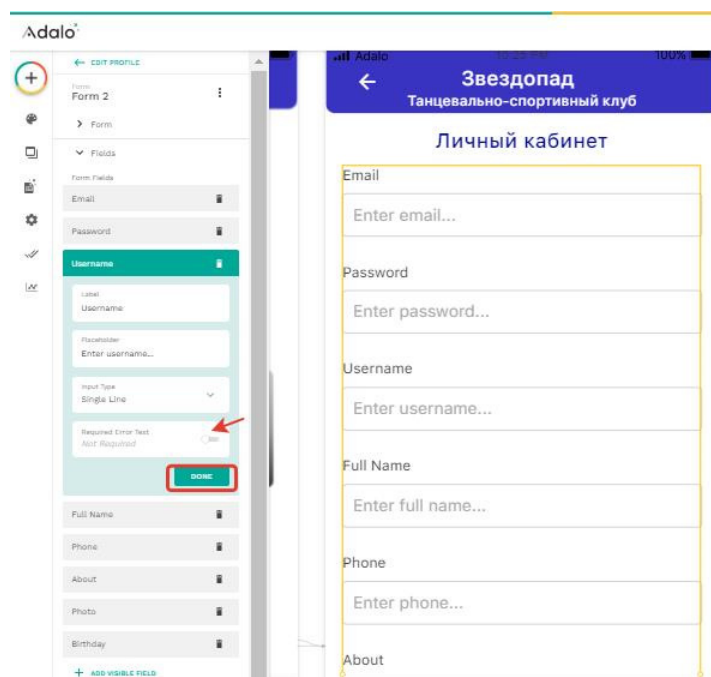


Рис. 4.61. Настройка формы

Кроме трех полей, которые были заполнены при регистрации, все остальные поля сделаем необязательными для заполнения (переключатель должен быть в неактивном положении для Required Enter...).

Также добавим для кнопки еще одно действие: переход на экран Profile после обновления профиля (рис. 4.62).

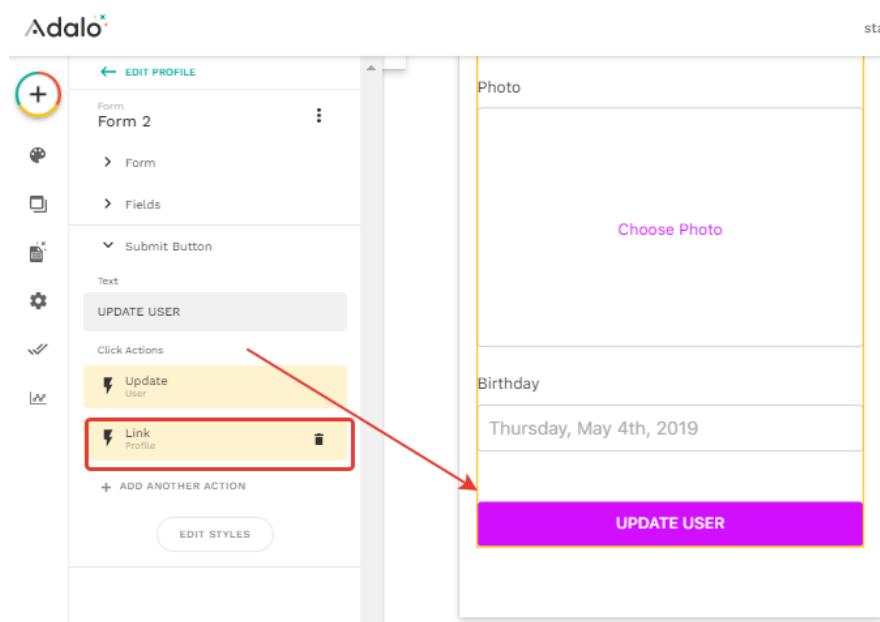


Рис. 4.62. Добавление действия для кнопки

В нашем приложении пока не сохраняются значения переключателей сердечек. Добавим в коллекцию Posts еще одну связь с коллекцией Users (рис. 4.63).

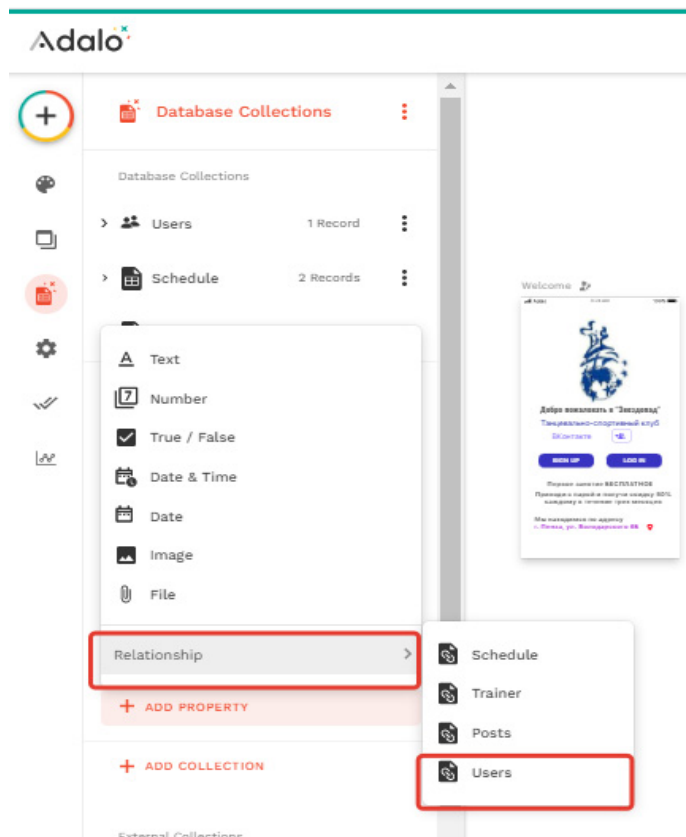


Рис. 4.63. Добавление свойства в коллекцию Posts

Так как у одного поста могут быть сердечки от многих пользователей, а один пользователь может добавить сердечки ко многим постам, то тип связи – многие ко многим (рис. 4.63).

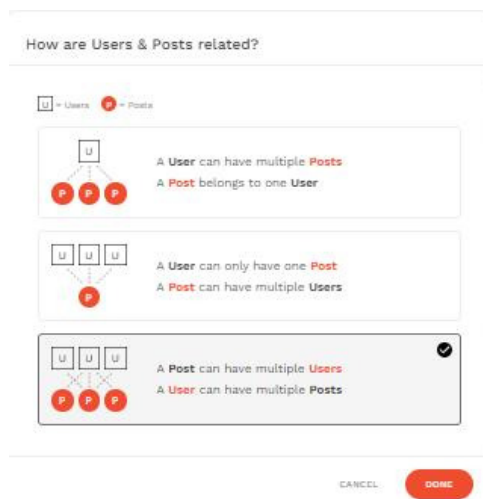


Рис. 4.63. Выбор типа связи

Добавленное в коллекцию Posts свойство назовем Liking users, а свойство Posts, которое добавилось в коллекцию Users, переименуем

в Liked posts. И настроим для переключателя Icon в Image List на экране Posts свойство What does this toggle? (рис. 4.64).

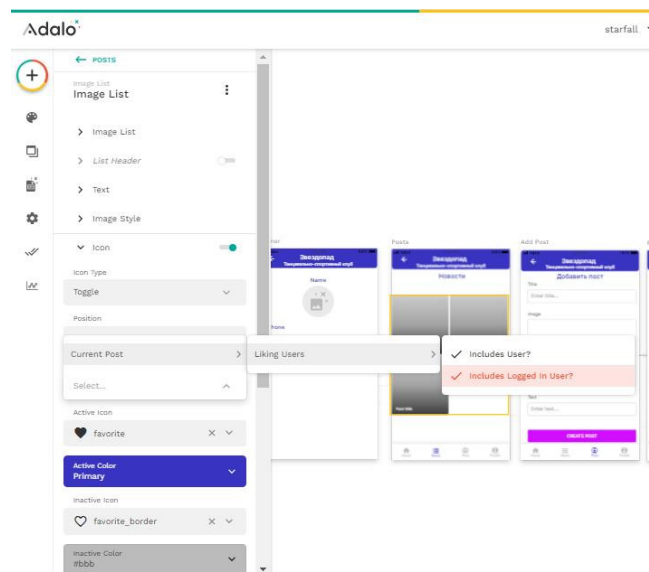


Рис. 4.64. Настройка сохранения состояния для переключателя сердечко

Все настроенные экраны представлены в приложении.

Протестируем созданное приложение. Зарегистрируемся, выложим несколько постов, обновим профиль и выйдем из приложения (рис. 4.65).

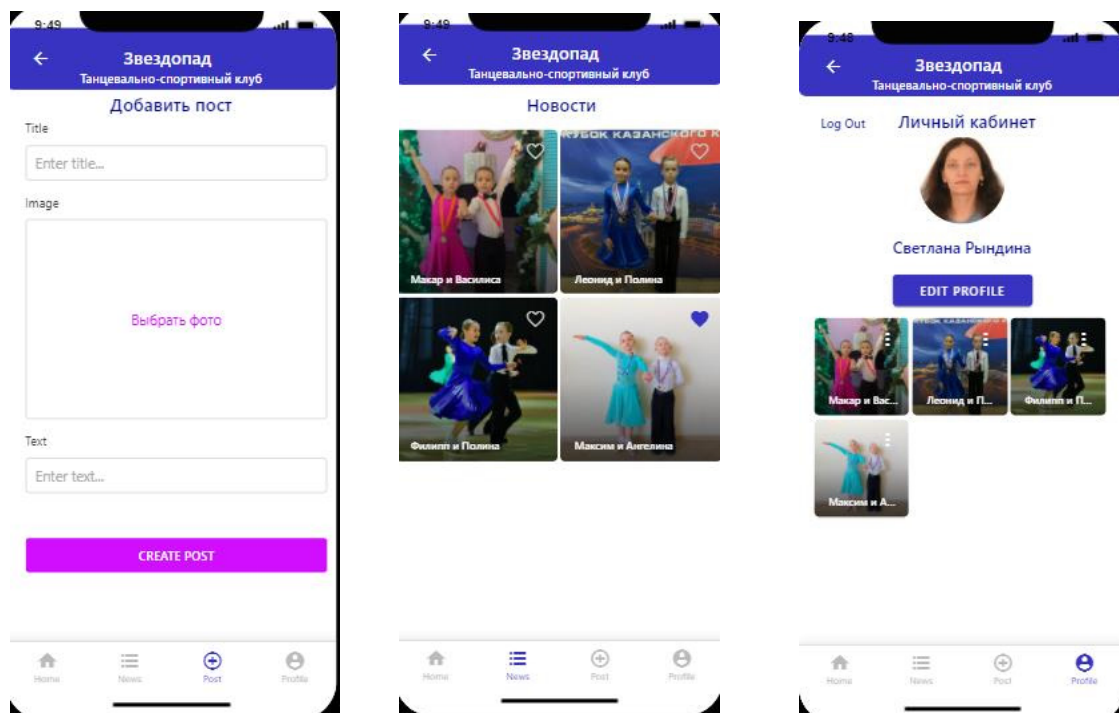


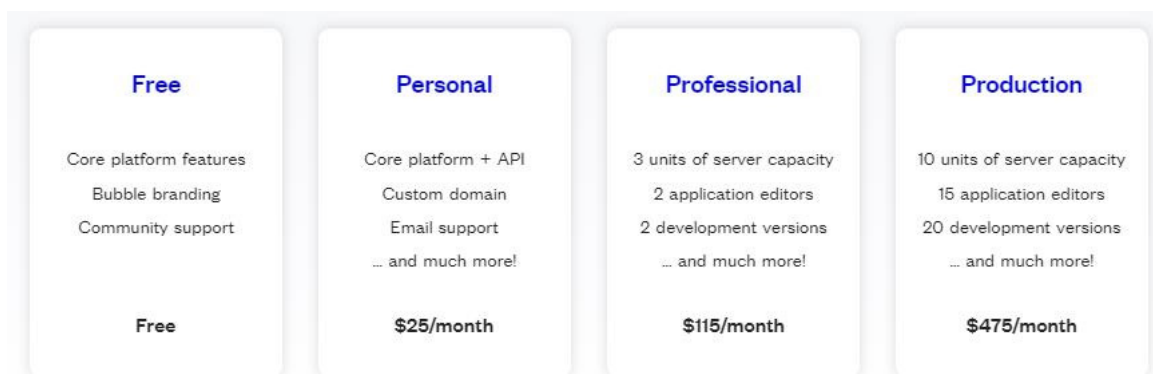
Рис. 4.65. Тестирование работоспособности приложения

Для дальнейшего совершенствования навыков разработки приложений в конструкторе Adalo можно воспользоваться соответствующими ресурсами [4, 5].

5. Визуальный конструктор для создания веб-приложений: Bubble

Bubble (<https://bubble.io/>) – платформа для создания веб-приложений без использования программного кода. Bubble.io – продукт компании, основанной в 2012 г. Эммануэлем Страшновым и Джошем Хаасом. Семь лет Bubble.io развивалась соучредителями на собственные средства как стратап, а с 2019 г. началось привлечение средств инвесторов для дальнейшего развития продукта (посевной раунд – шесть с половиной миллионов долларов в 2019 г., раунд А – сто миллионов долларов в 2021 г.). Стандартный подход no-code-инструментов – сосредоточиться на какой-то отдельной функциональности: лендинг, рассылки, аналитика и т.п. Сила конструктора Bubble в его многофункциональности и гибкости.

Однако для целей обучения этот продукт не так хорош, как Glide и Adalo, так как его бесплатный план использования позволяет только изучить базовые возможности платформы и создать свое приложение. А использовать созданное приложение по назначению можно лишь на платных тарифах (рис. 5.1).



Free	Personal	Professional	Production
Core platform features Bubble branding Community support	Core platform + API Custom domain Email support ... and much more!	3 units of server capacity 2 application editors 2 development versions ... and much more!	10 units of server capacity 15 application editors 20 development versions ... and much more!
Free	\$25/month	\$115/month	\$475/month

Рис. 5.1. Тарифная линейка Bubble

Приложение Bubble может быть запущено только на платформе Bubble; экспорт приложения в виде кода невозможен. Если уйти с платформы Bubble, то придется заново создавать логику приложения.

Регистрация в Bubble – это переход по кнопке Get Started for free (рис. 5.2). Обратите внимание, что веб-приложение конструктора Bubble создано с использованием конструктора Bubble (о чем информирует плашка в правом нижнем углу). При наличии аккаунта в Google регистрация происходит в один клик (рис. 5.3).

После принятия пользователем всех необходимых соглашений он попадает на стартовую страницу, на которой создатели предусмотрели обширную обучающую программу. Так как продукт достаточно

непрост в освоении, особенно для тех, кто раньше не сталкивался с разработкой. В самом конце этого списка располагается заветная кнопка Create a new app (рис. 5.4).

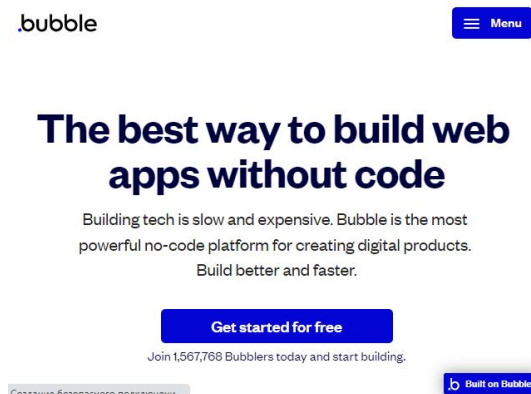


Рис. 5.2. Начало работы в Bubble

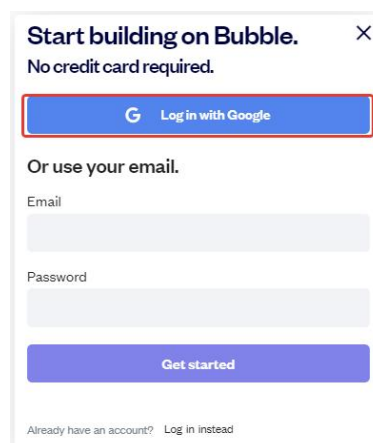


Рис. 5.3. Регистрация с аккаунтом @gmail.com

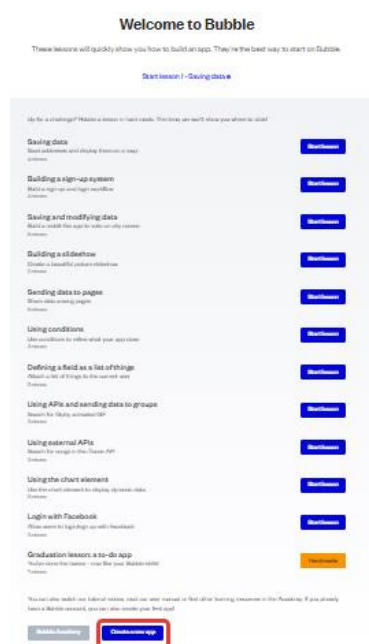
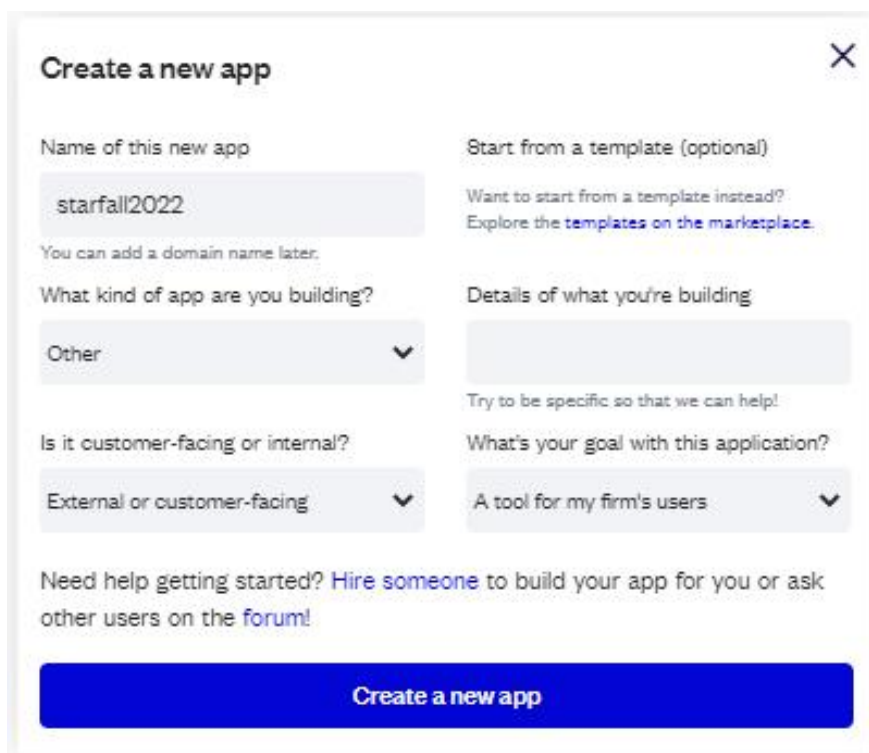


Рис. 5.4. Создание нового приложения

Заполняем краткую информацию о приложении (рис. 5.5). Название приложения должно быть уникальным и не может совпадать с названиями ранее созданных приложений (поэтому имя у приложения starfall2022, а не просто starfall).



The screenshot shows a 'Create a new app' form with the following fields and options:

- Name of this new app:** starfall2022
- Start from a template (optional):** Want to start from a template instead? Explore the [templates on the marketplace](#).
- What kind of app are you building?:** Other
- Is it customer-facing or internal?:** External or customer-facing
- Details of what you're building:** Try to be specific so that we can help! What's your goal with this application? A tool for my firm's users
- Need help getting started?:** Hire someone to build your app for you or ask other users on the forum!
- Create a new app** button

Рис. 5.5. Информация о приложении

Начинаем работу с чистого листа и закрываем ассистента (рис. 5.6).

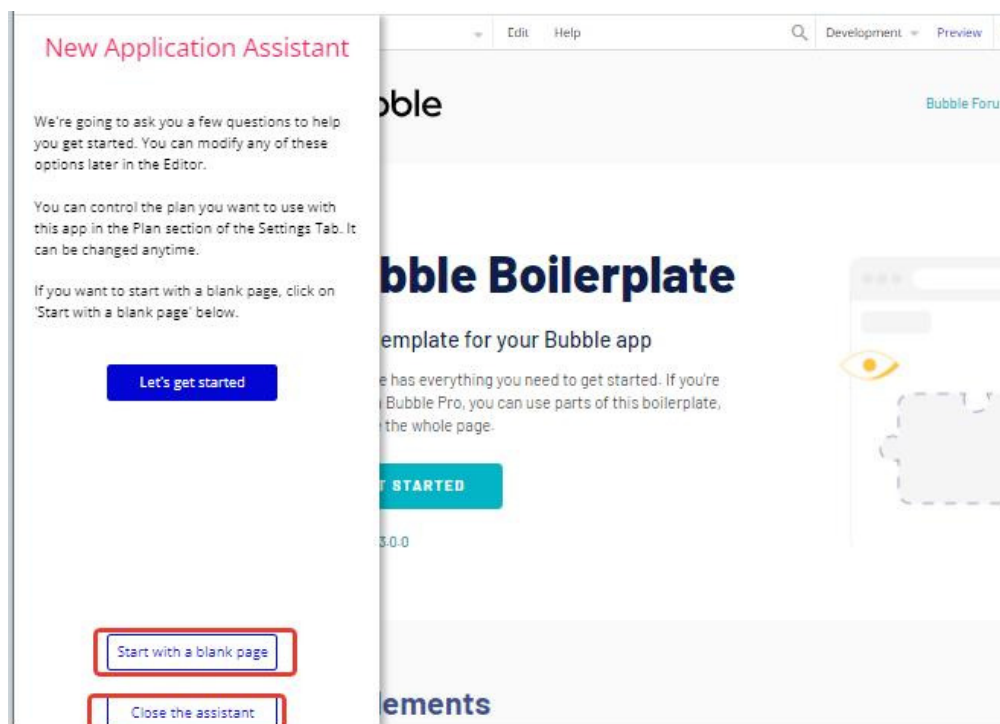


Рис. 5.6. Добавление первой страницы в приложение

Для Bubble корректнее говорить, создаем страницы, а не экраны, так как это все же инструмент, прежде всего предназначенный для веб-приложений, и только во вторую очередь для создания PWA и нативных приложений (такая функция в настройках страниц имеется, правда в стадии бета-тестирования).

Дважды кликнув мышкой в рабочем пространстве за пределами страницы – появится редактор свойств элемента, в данном случае страницы. Доступ к редактору свойств элемента можно получить, вызвав кликом правой клавиши мыши контекстное меню и выбрав в нем опцию Edit. Редактор свойств есть не только у элементов, но и у действий, а также событий. В редакторе свойств можно ввести заголовок стартовой страницы (по традиции для веб-приложений она называется index, что отражается в левом верхнем углу для переключателя между страницами и для добавления новых), настроить ее размеры. Для веб-приложений можно создать различные версии страницы для отображения в мобильном приложении, а если установить галочку рядом с позицией This page is a native app, то появится окно, предупреждающее, что для нативных приложений размер экрана нужно установить 640×320, и эта опция пока в стадии бета-тестирования.

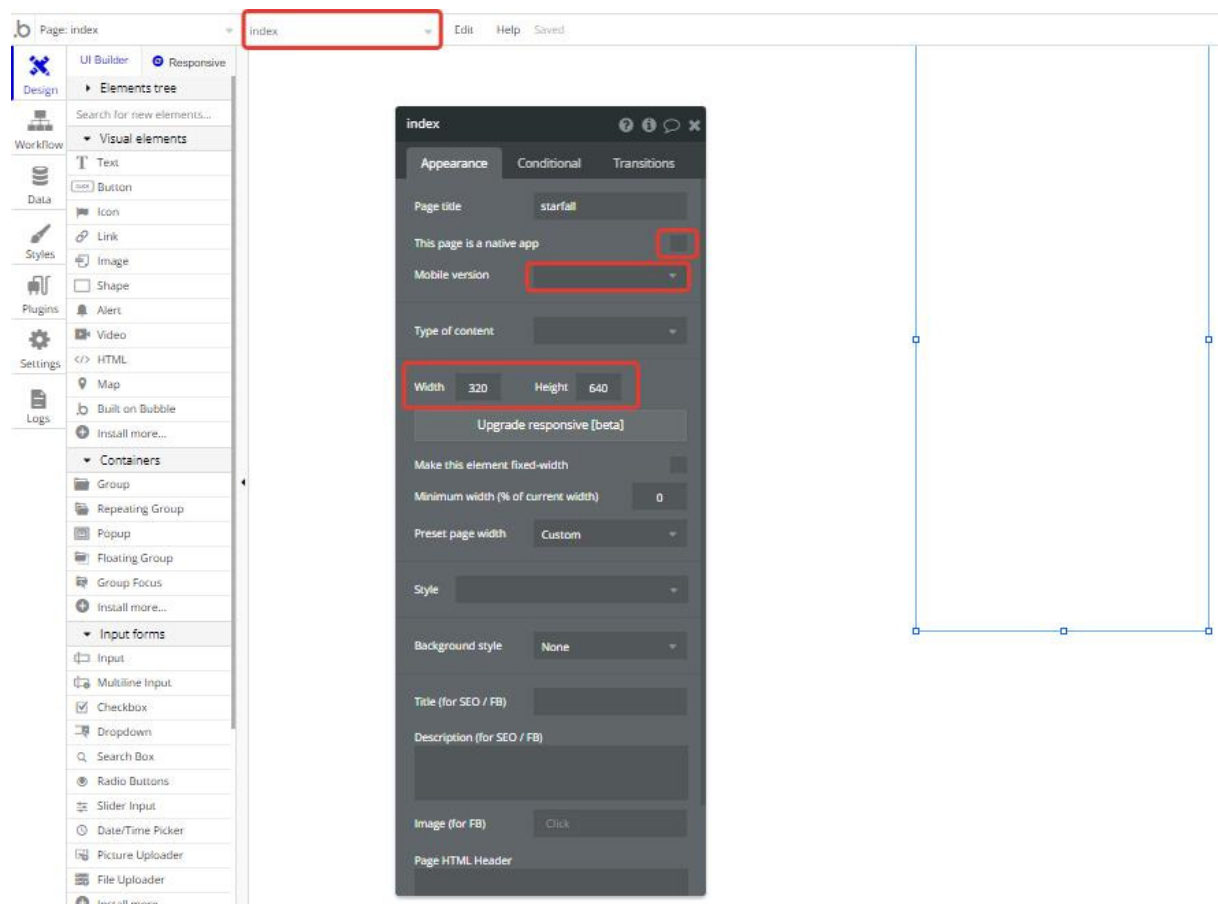


Рис. 5.7. Настройка первой страницы

Bubble автоматически сохраняет приложение в режиме реального времени. Иногда уведомление о сохранении может стать красным, в случае нестабильности соединения, тогда изменения могут быть потеряны и до восстановления соединения работу с приложением нужно временно приостановить.

Создание приложения – это итеративный процесс. Необходимо спроектировать последовательность экранов, продумать структуру данных и навигацию. Для создания новой страницы можно обратиться к меню Edit и даже использовать опцию, клонирующую все настройки текущей страницы, либо раскрыть меню рядом с именем страницы и выбрать опцию Add new page (рис. 5.8).

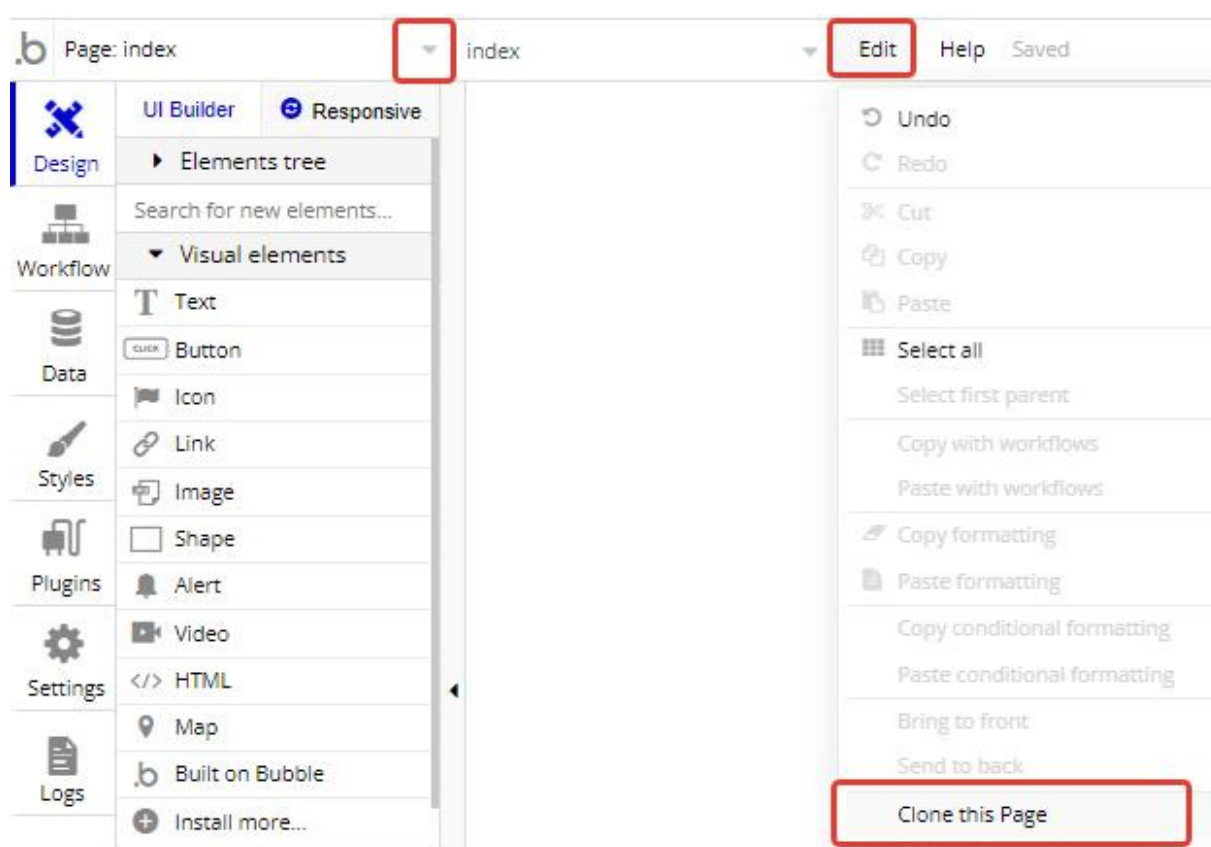


Рис. 5.8. Добавление новой страницы

На экране разместим элемент Image (загрузим файл логотипа), создадим замещающий картинку текст (logo), также для любого добавляемого элемента можно настроить рабочий процесс, т.е. действия (рис. 5.9).

Далее добавим текстовый элемент (рис. 5.10).

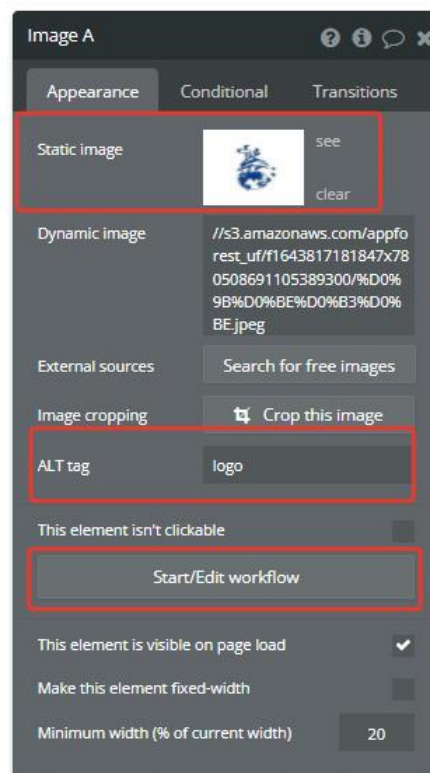


Рис. 5.9. Добавление изображения

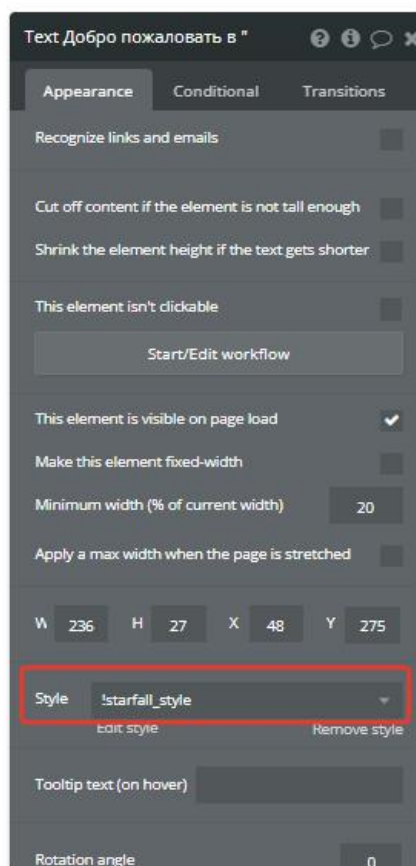


Рис. 5.10. Добавление текстового элемента

Стиль этого элемента можно выбрать из списка, а можно создать свой – уникальный (для быстрого поиска в списке стилей при настройке лучше в названии начинать со знаков, например с «!»).

При настройке стиля активизируется вкладка левой панели навигации Style (рис. 5.11).

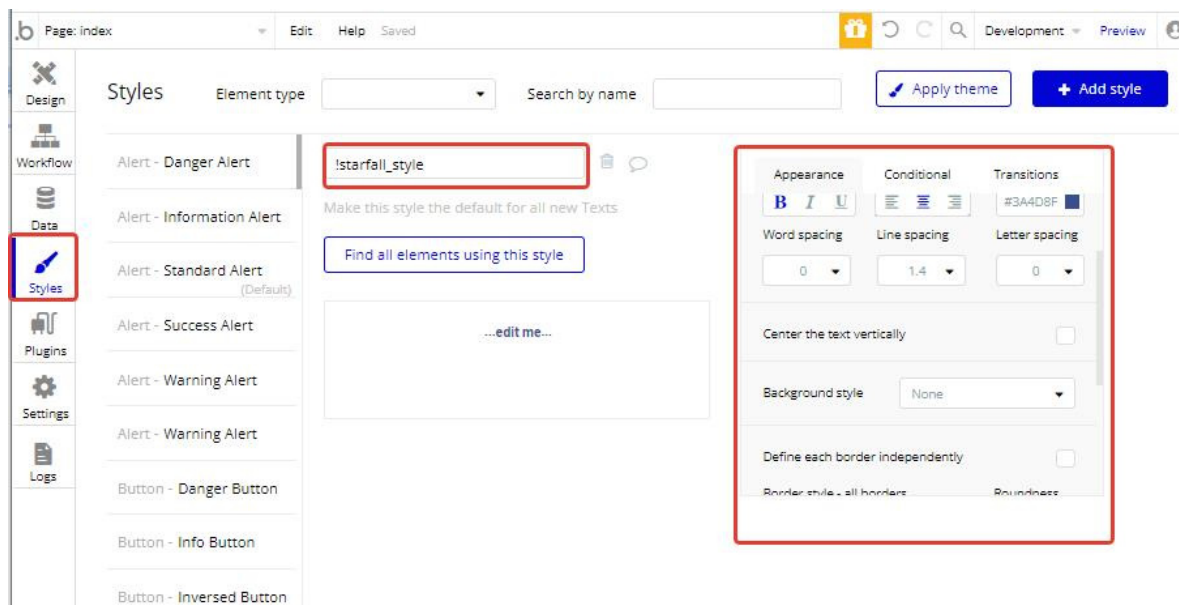


Рис. 5.11. Настройка созданного стиля

Все изменения сохраняются при внесении, поэтому, чтобы увидеть результат, нужно вернуться на вкладку Design.

Добавим клон страницы index и назовем ее sign-up (рис. 5.12).

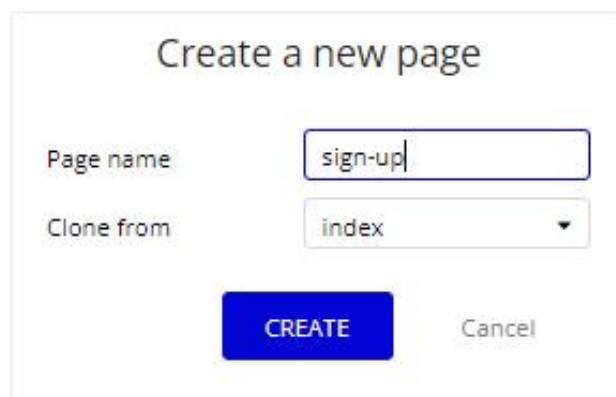


Рис. 5.12. Добавление страницы

Вернемся на страницу index и добавим элемент дизайна Button (рис. 5.13).

Для кнопки нужно настроить действие – переход на страницу регистрации, которую только что создали. Кликаем на кнопке Start/Edit workflow и перемещаемся на вкладку настройки действия (рис. 5.14). Нужное действие содержится в разделе Navigation и называется Go to page...

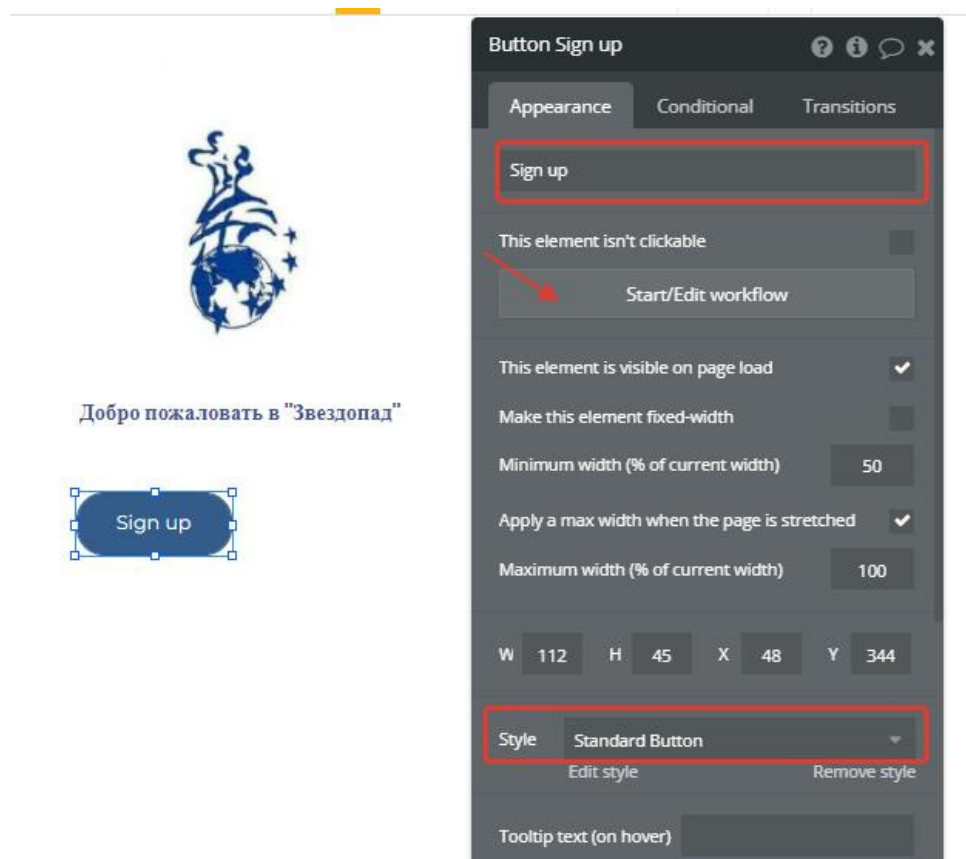


Рис. 5.13. Добавление кнопки

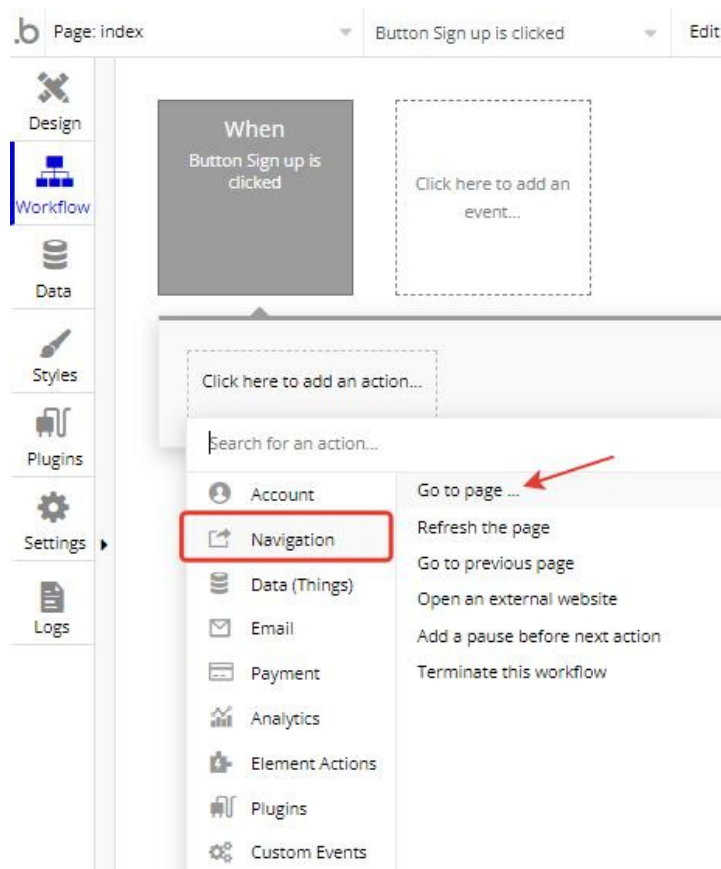


Рис. 5.14. Настройка действия

Результат настройки действия представлен на рис. 5.15. Возвращаемся на вкладку Design и переходим к настройке страницы sign-up.

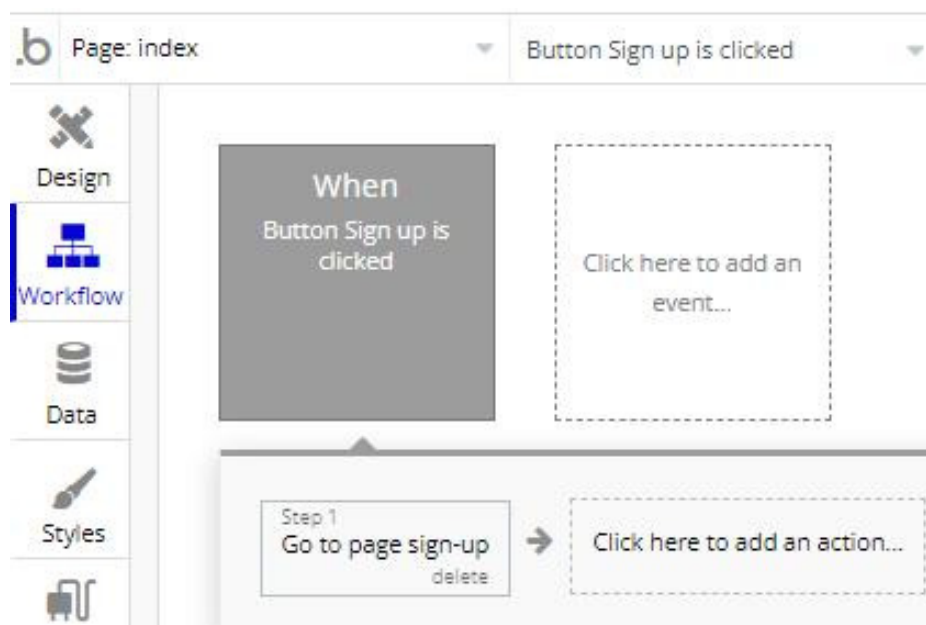


Рис. 5.15. Результат настройки действия

На странице sign-up удаляем логотип, редактируем текст и добавляем поле ввода (рис. 5.16). Все ранее добавляемые элементы были из раздела Visual elements, а поле ввода Input – в разделе Input forms.

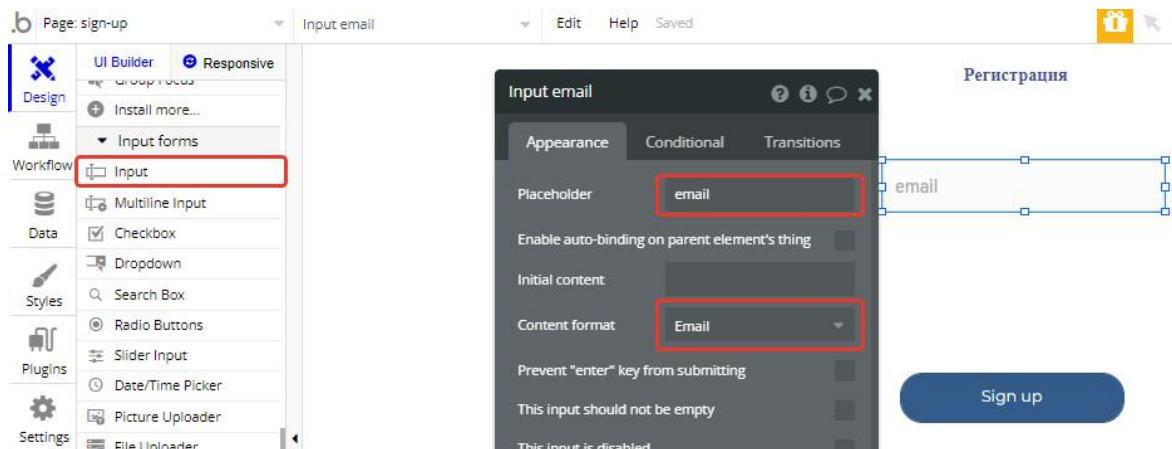


Рис. 5.16. Добавление поля ввода

Важно выбрать email в опции Content format, чтобы происходила проверка шаблона ввода для email. И поставить галочку в поле This input should not be entry, чтобы поле стало обязательным для заполнения. Настройки для всех трех полей ввода формы регистрации представлены на рис. 5.17–5.19.

Проведем настройку рабочего процесса для кнопки Sign up. На странице sign-up эта кнопка отвечает за сохранение данных из формы ввода (рис. 5.20).

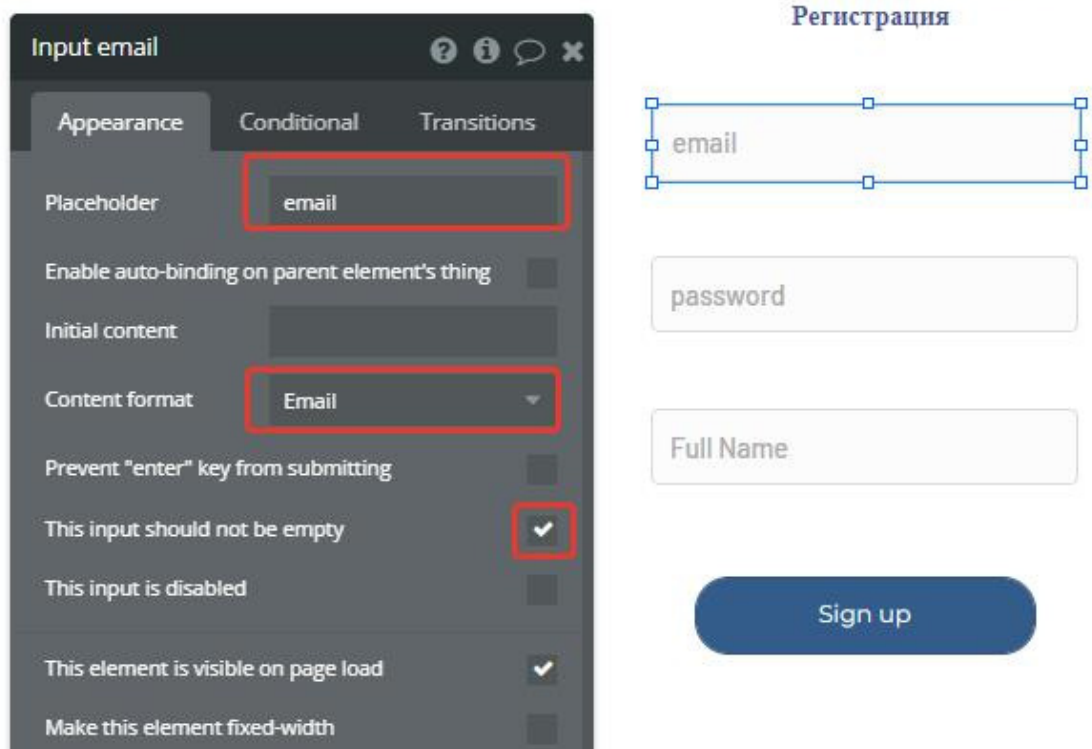


Рис. 5.17. Настройка поля ввода email

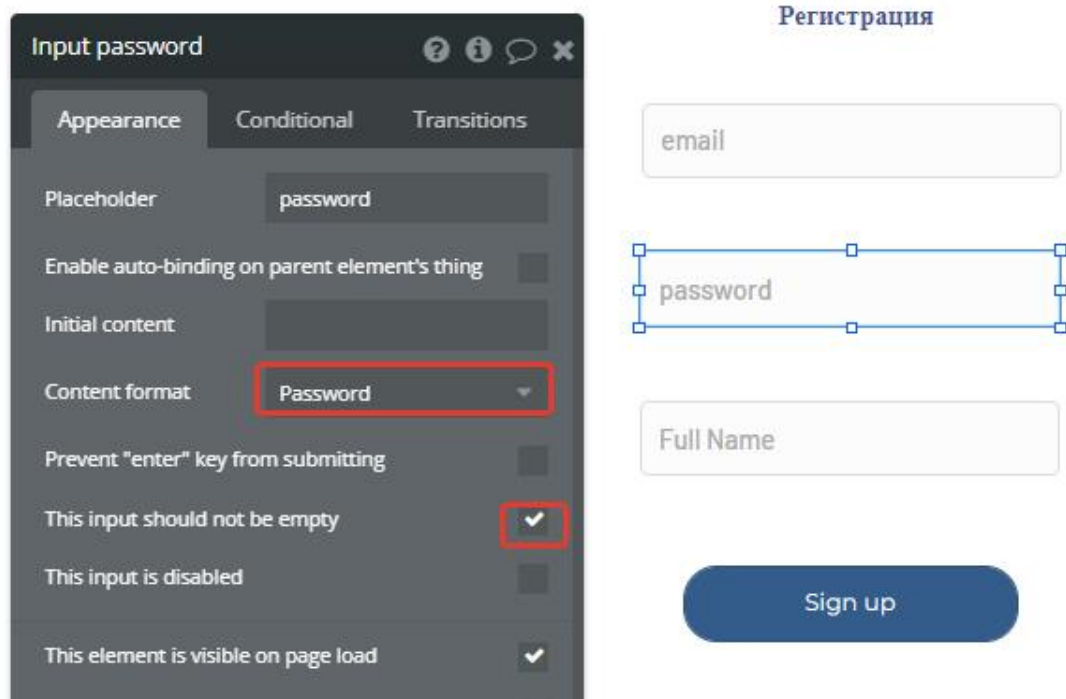


Рис. 5.18. Настройка поля ввода password

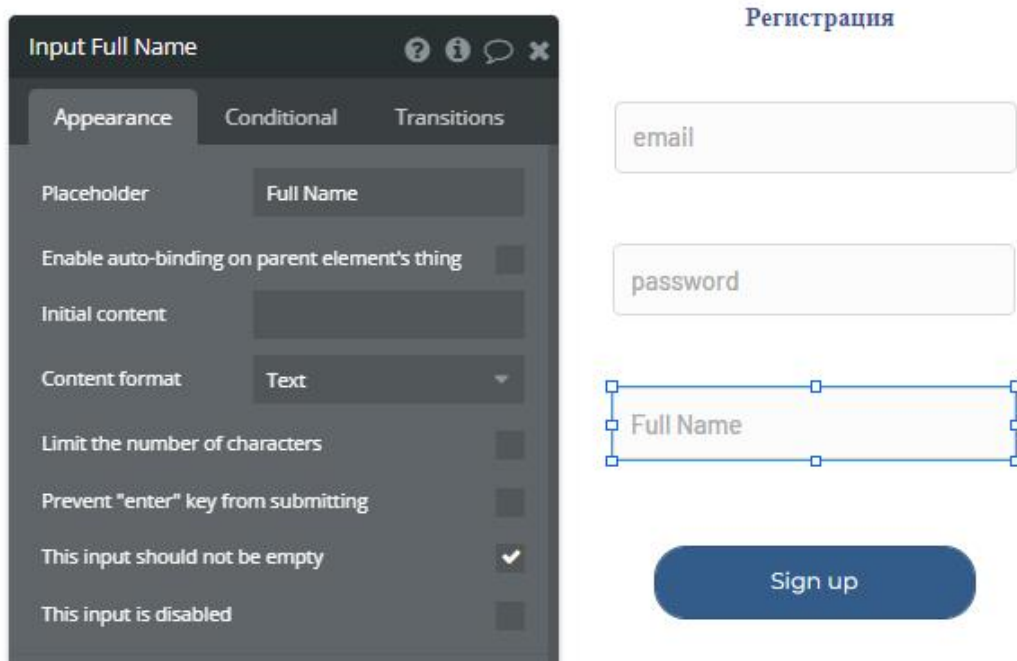


Рис. 5.19. Настройка поля ввода Full Name

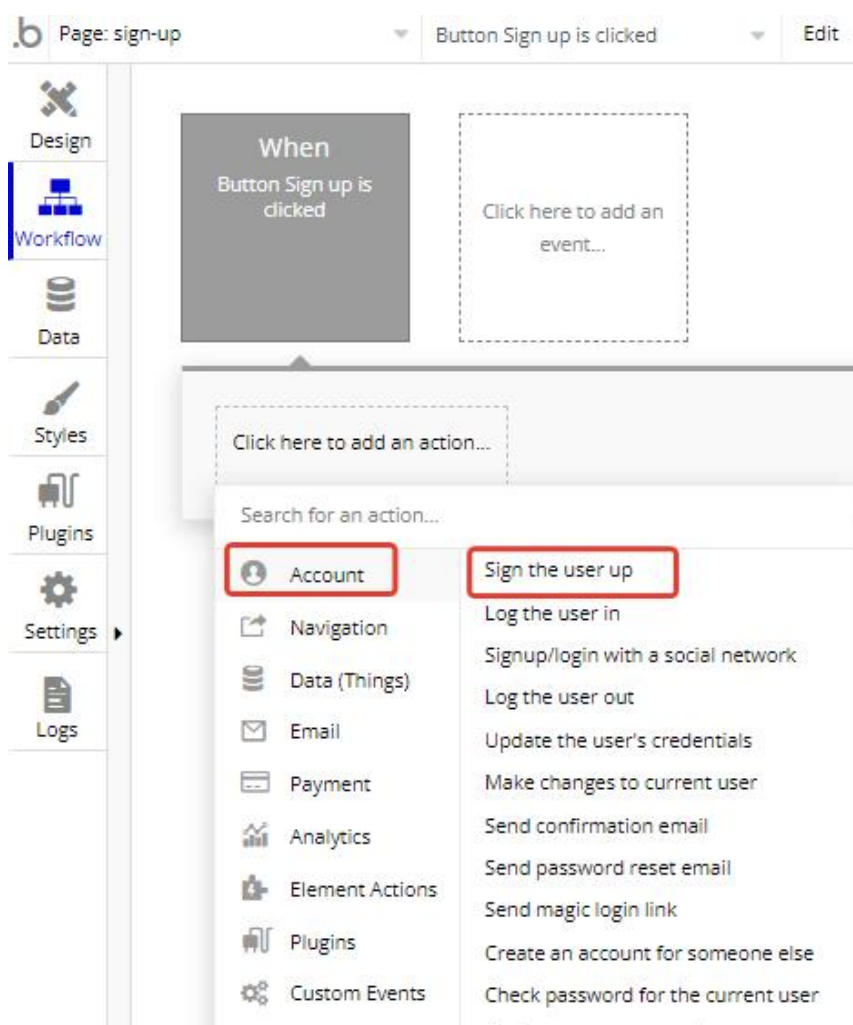


Рис. 5.20. Настройка действия для кнопки

В окне настроек для действия в полях Email и Password по кнопкам Click настраиваем, что это будут введенные значения, которые нужно сохранить в переменной (рис. 5.21).

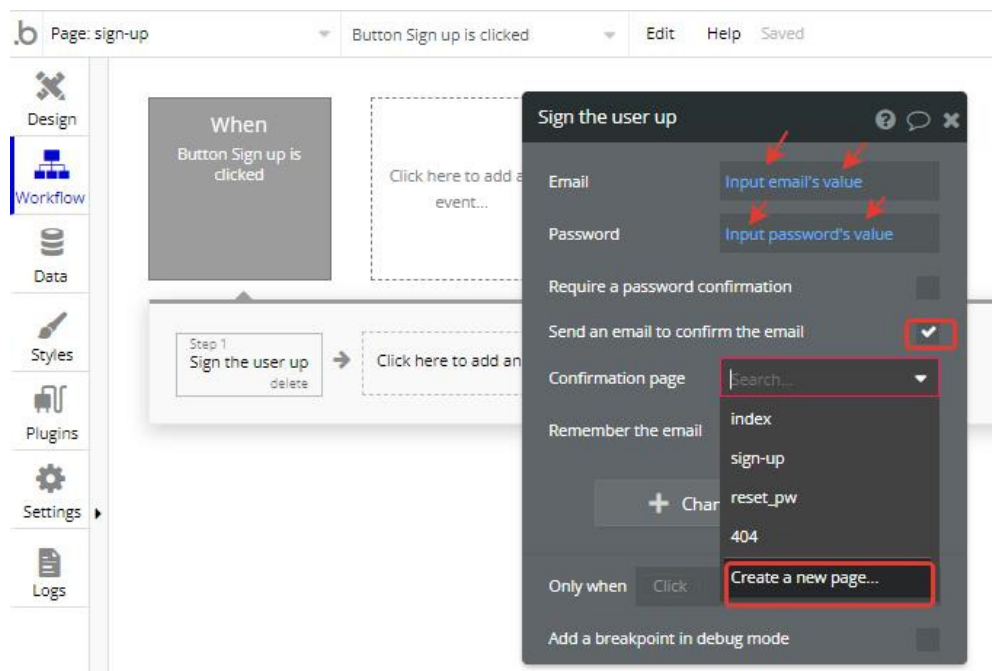


Рис. 5.21. Настройки для действия

Если выставить галочку на опции Send an email to confirm the email, то регистрация будет происходить с подтверждением по почте. После подтверждения необходимо перенаправлять на новую страницу, которую создадим сразу из редактора свойств и назовем home.

На новой странице разместим пока только текст заголовка «Звездопад», чтобы понимать, что это новая страница. Перейдем в предпросмотр и зарегистрируем пользователя с настоящим email, чтобы можно было выполнить подтверждение.

После регистрации вернемся в конструктор и зайдем на вкладку Data, отобразим всех пользователей (рис. 5.22).

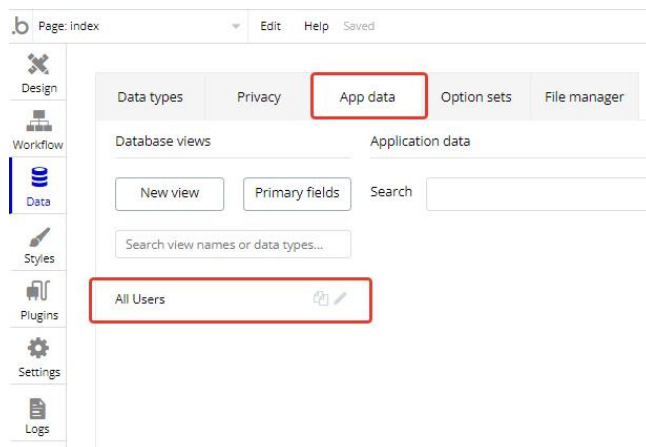


Рис. 5.22. Отображение таблицы пользователей

При необходимости для редактирования записи используем кнопку слева от нее (рис. 5.23).

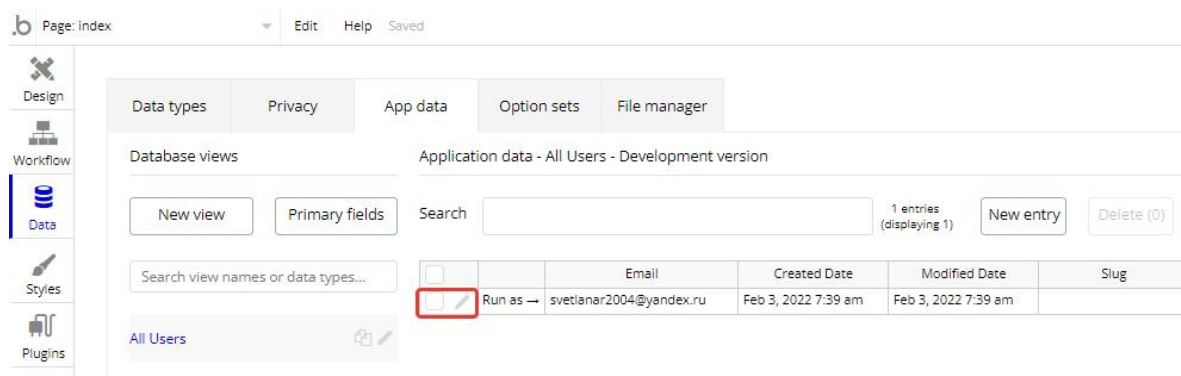


Рис. 5.23. Редактирование записи

Создадим в таблице новое поле, которое есть в форме регистрации (рис. 5.24, 5.25).

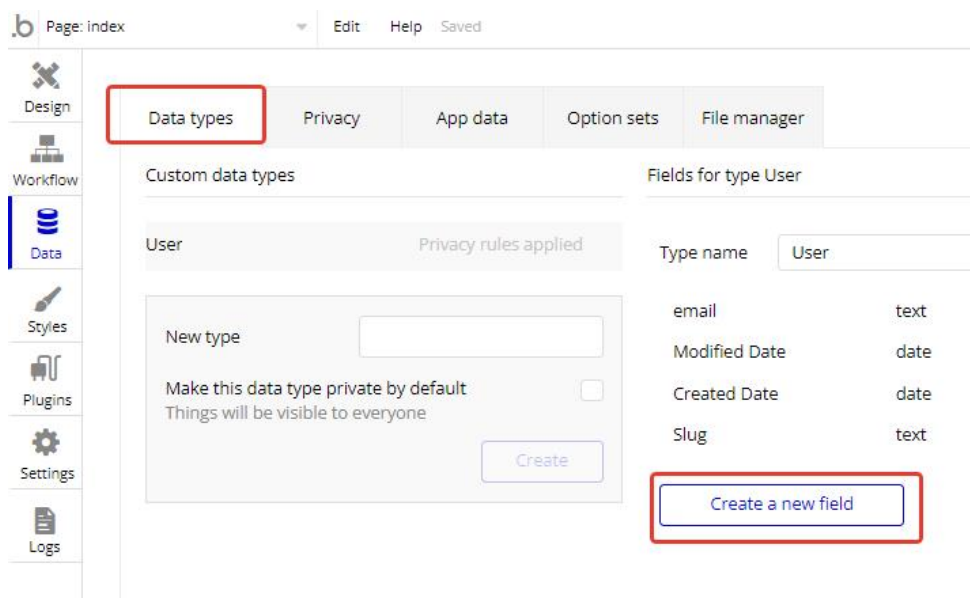


Рис. 5.24. Добавление в таблицу поля

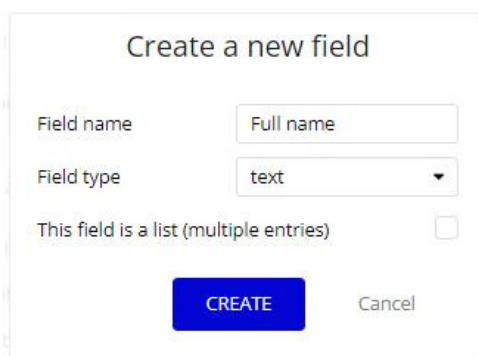


Рис. 5.25. Добавление в таблицу поля

Так как с текущим пользователем возможны только два действия – удаление записи в таблице в конструкторе или ее модификация,

просто изменим запись в конструкторе, добавив значение в поле Full name (рис. 5.26).

Modify an existing database entry					
Type of thing	User				
Full name	Рындина Светлана	Email	svetlanar2004@yandex.ru	Created Date	Feb 3, 2022 7:39 am
Modified Date	Feb 3, 2022 7:53 am	Slug		Unique id	1643863165337x5749579458700649
SAVE				Cancel	

Рис. 5.26. Изменение записи в таблице

Теперь вернемся в настройки Workflow для кнопки Sign up на странице sign-up. Уберем подтверждение по почте и добавим только что созданное поле Full name (рис. 5.27).

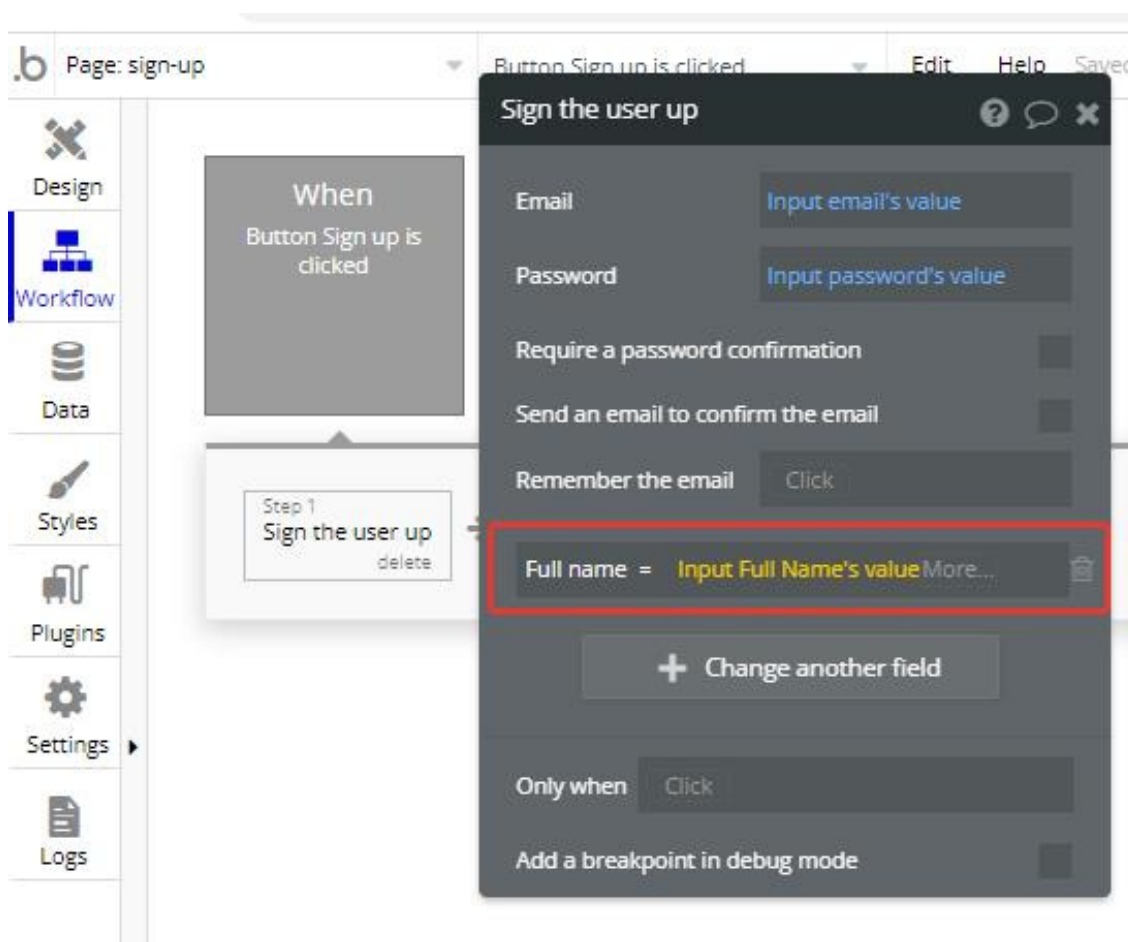


Рис. 5.27. Изменение настроек для действия

Теперь введенные значения в этом поле при регистрации будут сохраняться. Раньше это было невозможно из-за отсутствия в таблице данных такого поля.

Также добавим еще одно действие – переход на станицу home (рис. 5.28).

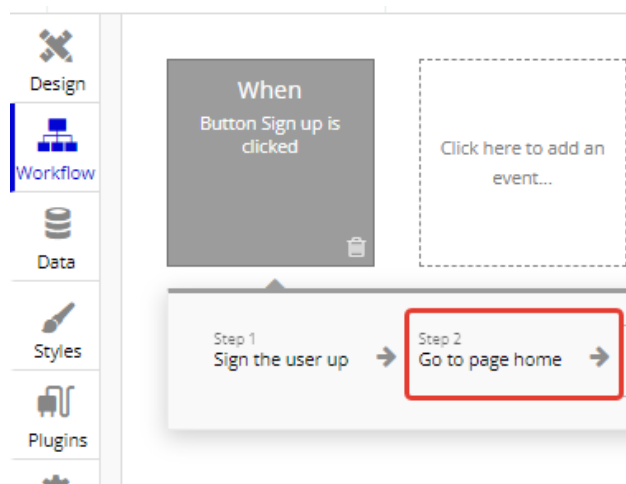


Рис. 5.28. Добавление второго действия

Настроим на странице home отображение имени пользователя. Для этого выберем всплывающую опцию Insert dynamic data (рис. 5.29).

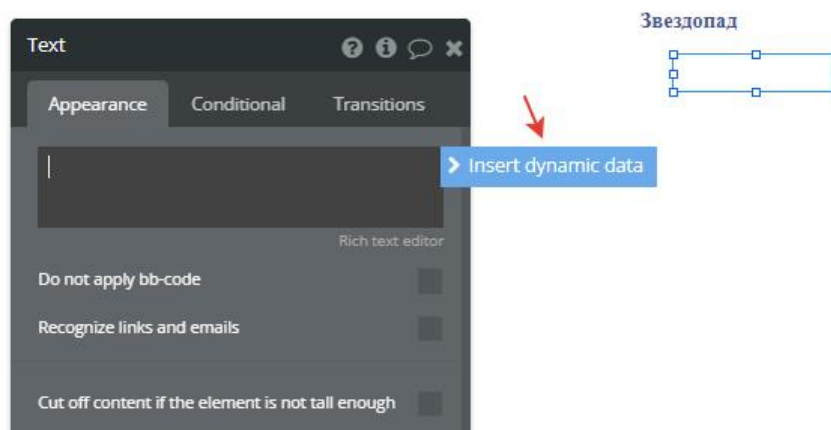


Рис. 5.29. Настройка отображения динамических данных

В появившемся списке выберем Current User (рис. 5.30), а затем 's Full name (рис. 5.31).

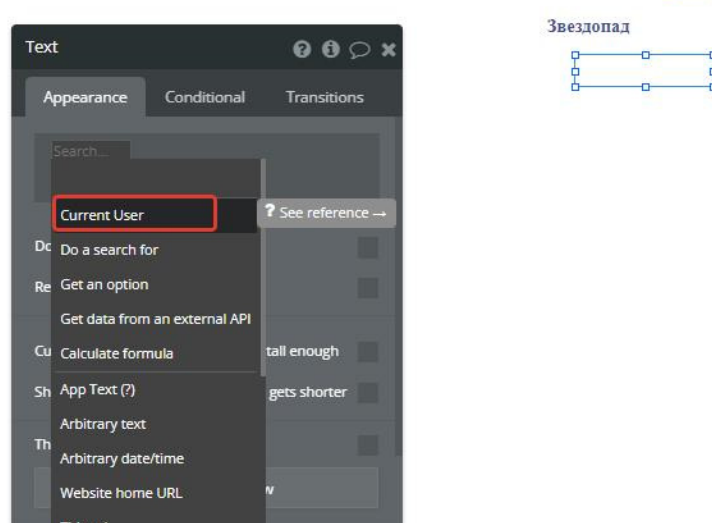


Рис. 5.30. Настройка отображения динамических данных

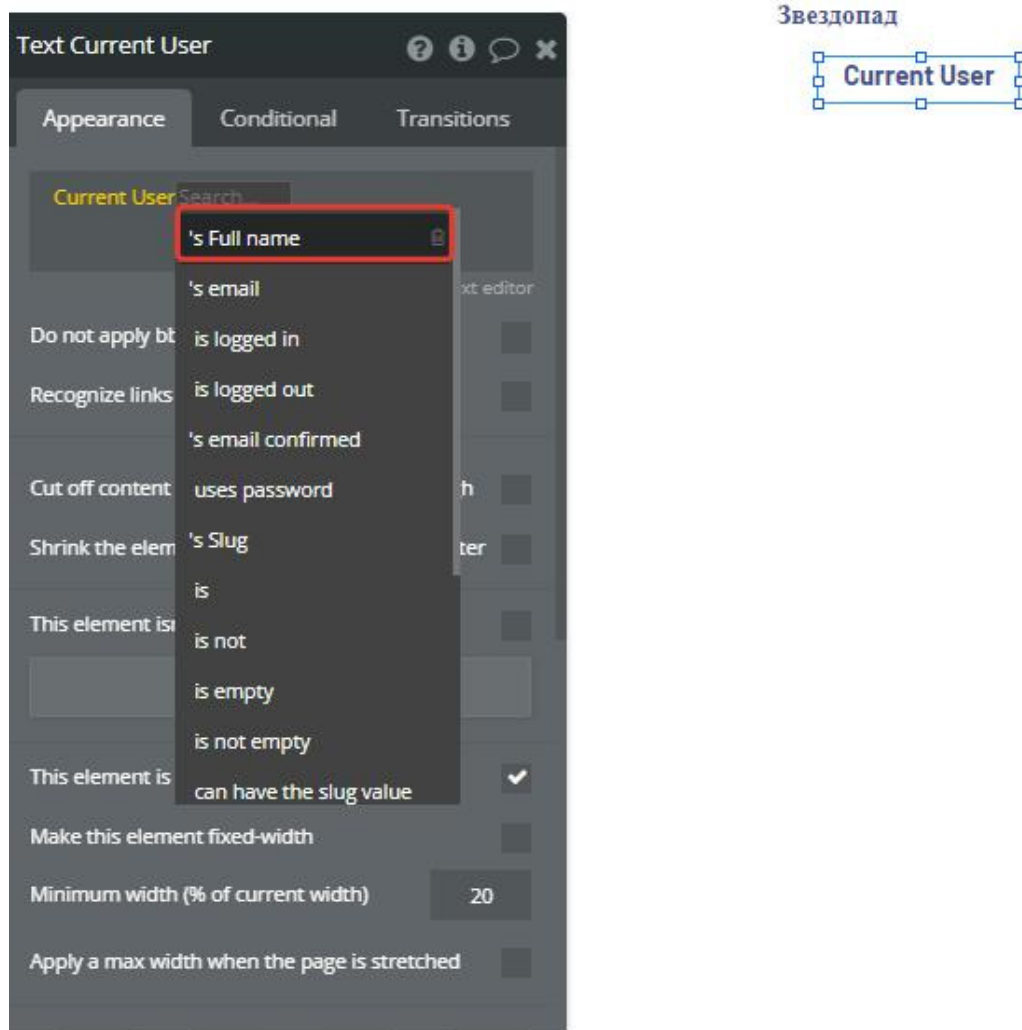


Рис. 5.31. Настройка отображения динамических данных

Также добавим на домашнюю страницу кнопка выхода из аккаунта (рис. 5.32).

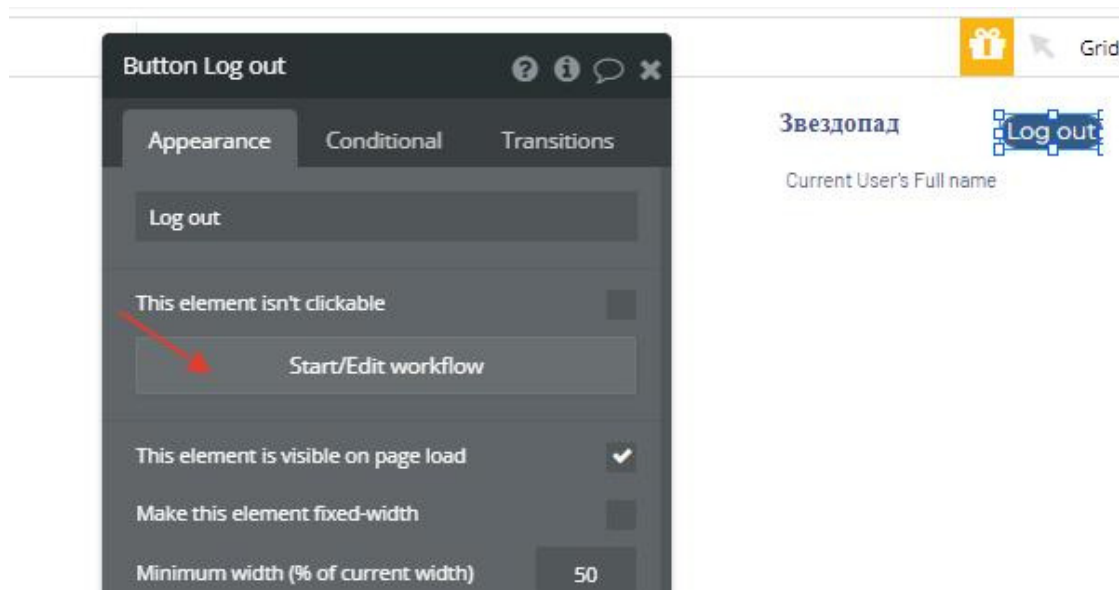


Рис. 5.32. Настройка кнопки Log out

И настроим для нее действие (рис. 5.33).

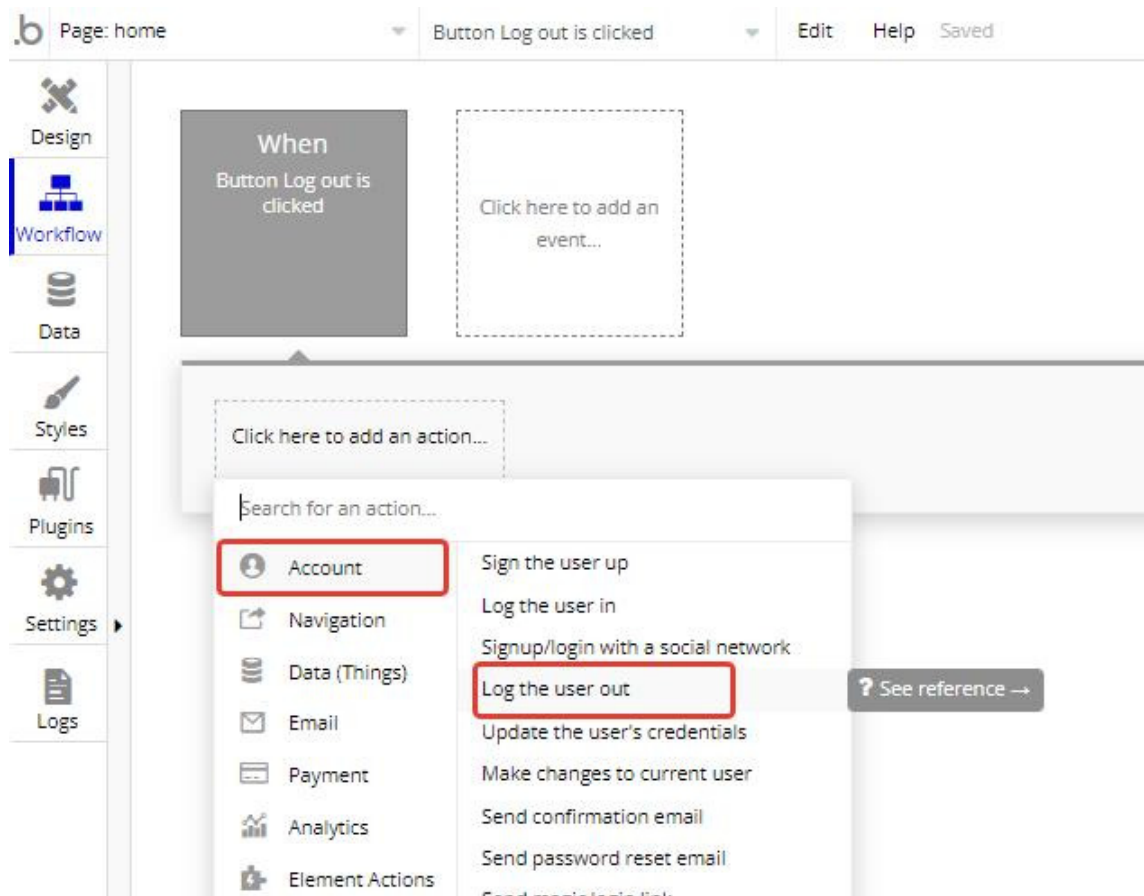


Рис. 5.33. Настройка действия для кнопки

Создадим клон страницы sign-up и назовем его log-in.

Изменим состав элементов страницы и текст согласно рис. 5.34.

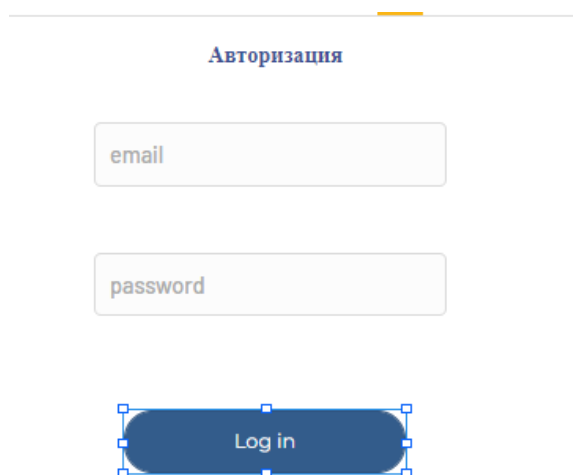


Рис. 5.34. Страница log-in

Настроим Workflow для кнопки Log in на странице log-in. Вначале настроим проверку значений введенных данных в состоянии кнопки: Only when Click в настройках When Button Log in is Clicked (рис. 5.35).

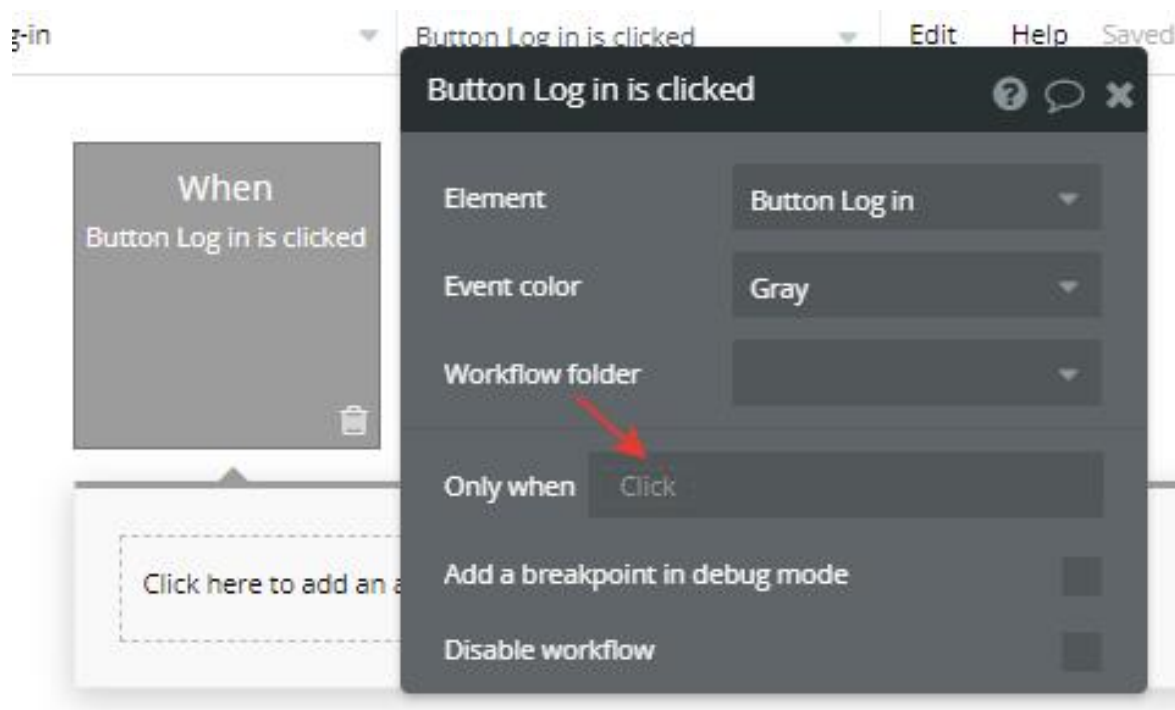


Рис. 5.35. Настройка проверки условий при клике на кнопку

Проверим наличие вводимого email в базе данных (рис. 5.36, 5.37).

Для email, хранящегося в базе, проверяем совпадение с Input email's value (см. рис. 5.37).

Введем условие, что количество совпадений имеющегося в базе email и введенного email должно быть больше нуля, т.е. :count>0, где показатель и знак сравнения выбираются из списка, а 0 вводится с клавиатуры (рис. 5.38, 5.39).

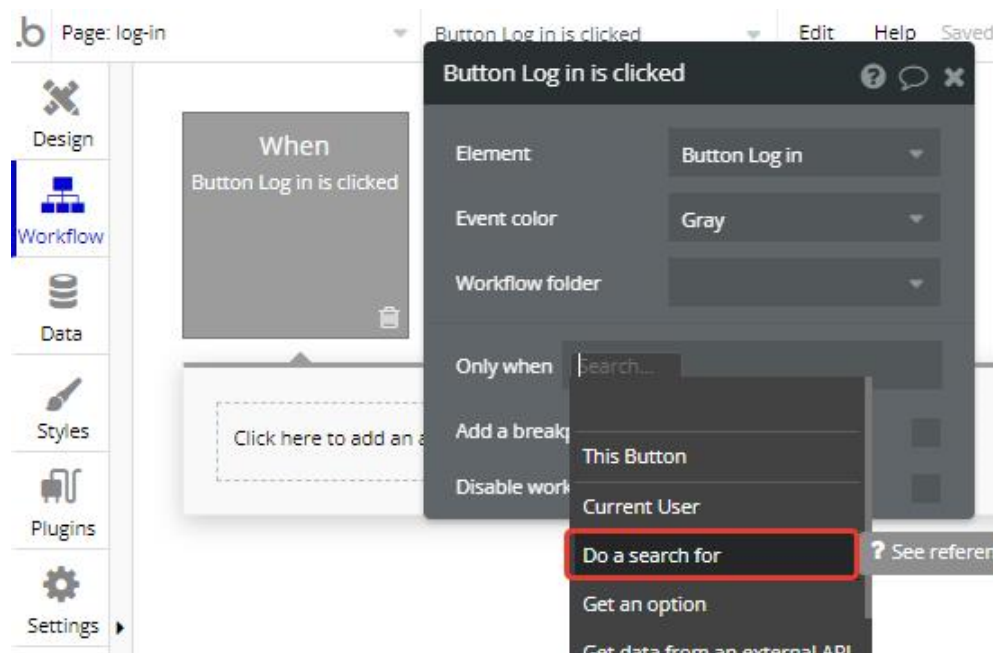


Рис. 5.36. Выполнение поиска в базе данных вводимого значения

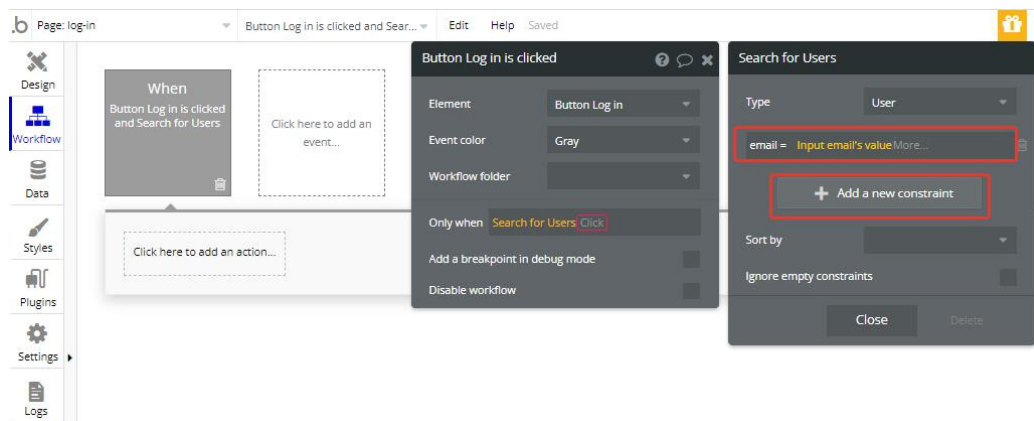


Рис. 5.37. Выполнение поиска в базе данных вводимого значения

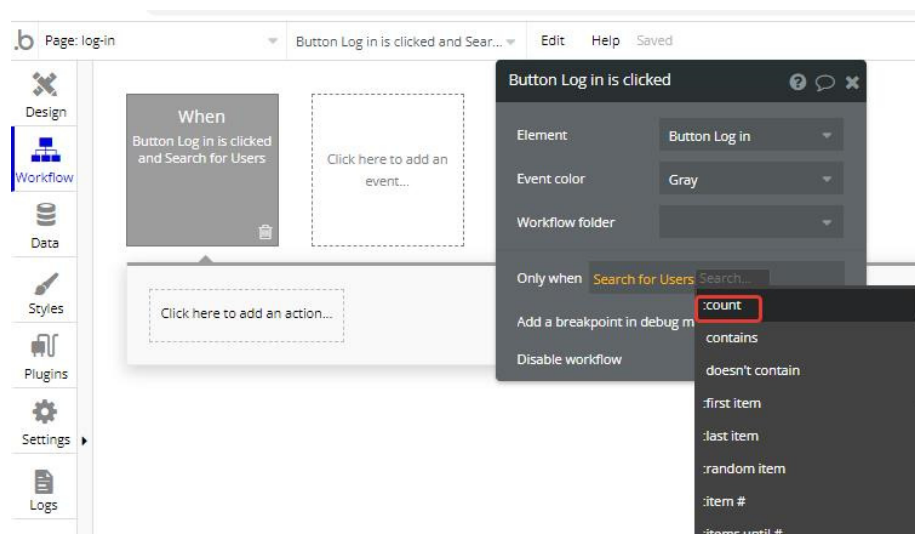


Рис. 5.38. Ввод условия

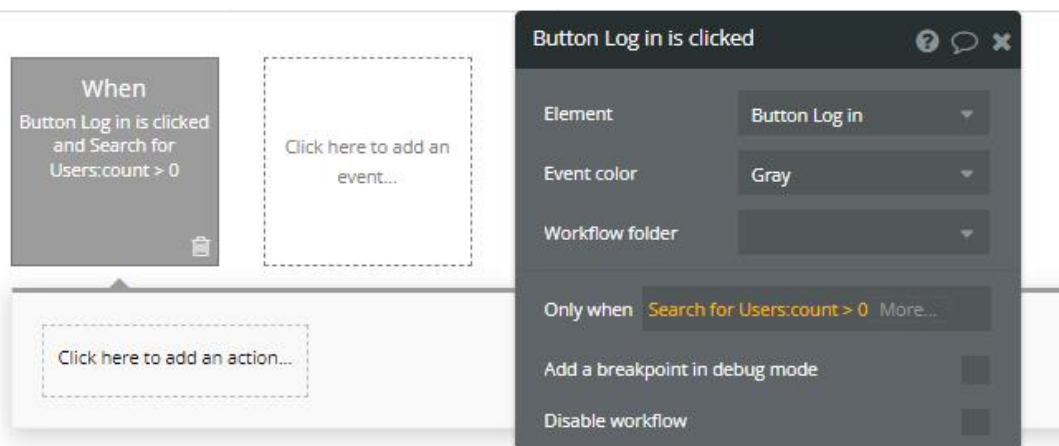


Рис. 5.39. Ввод условия

Настроим действие для кнопки, которое будет выполняться после успешной проверки условия: Account Log the user in (рис. 5.40, 5.41).

Для этого действия, как и для авторизации (Sign up), должны быть соотнесены email и password с введенными значениями: Input email's value и Input password's value (рис. 5.41).

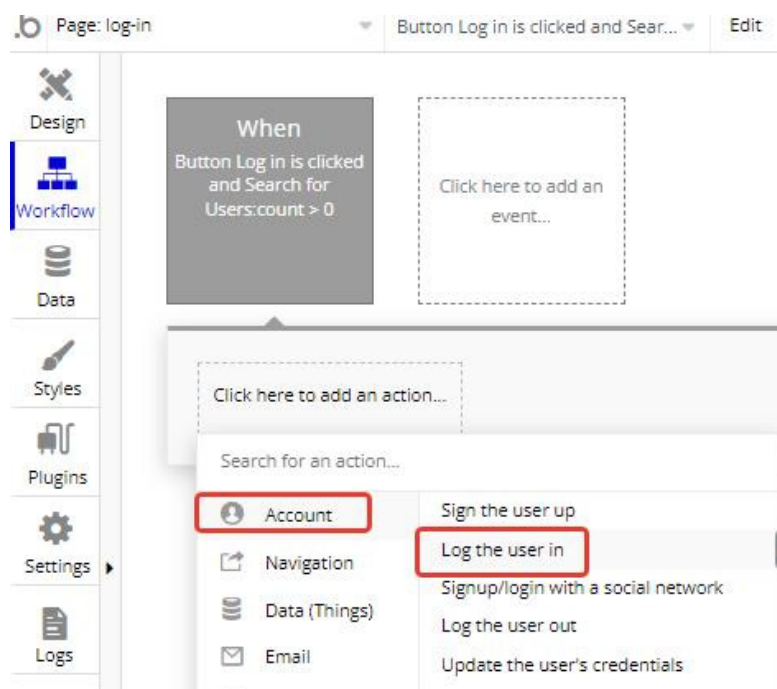


Рис. 5.40. Настройка действия

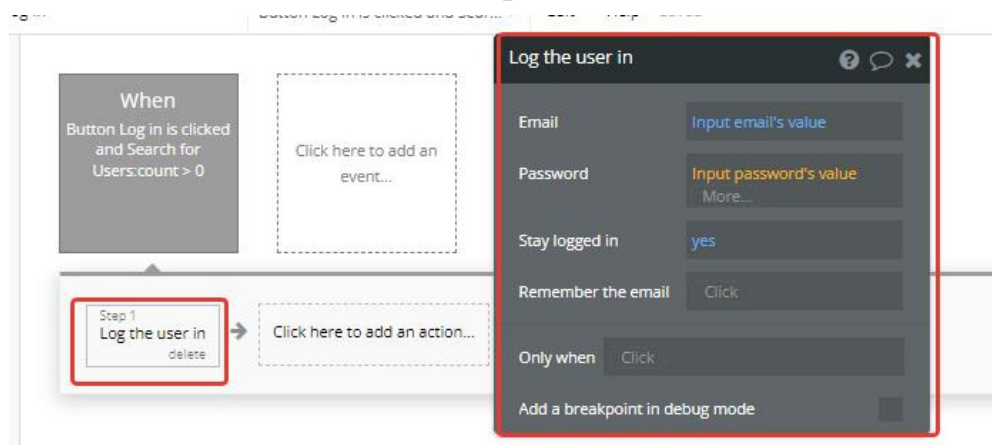


Рис. 5.41. Настройка действия

Добавим еще одно действие: в случае успешной авторизации должен осуществляться переход на домашнюю страницу (рис. 5.42).

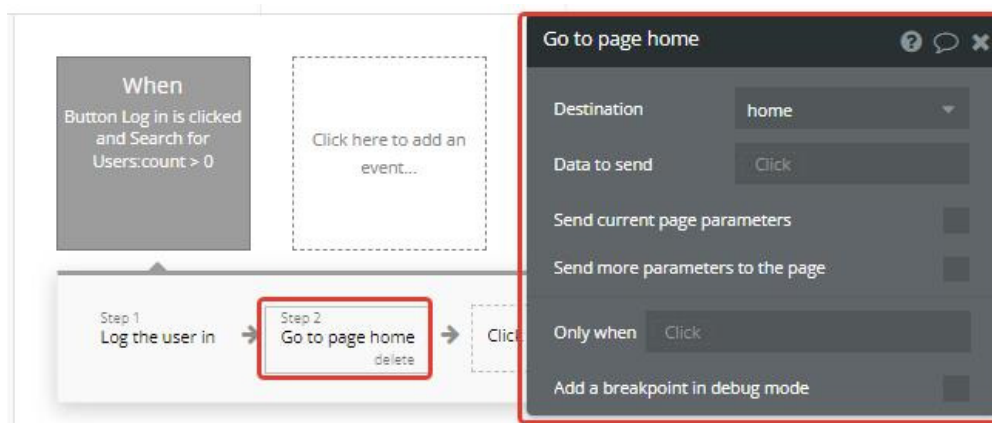


Рис. 5.42. Добавление еще одного действия

Теперь выполним копирование и вставку настроенного состояния кнопки и проведем модификацию: изменим условие, которое будет проверяться при нажатии на <1 (рис. 5.43), удалим действие авторизации, а переход настроим на страницу sign-up (рис. 5.44).

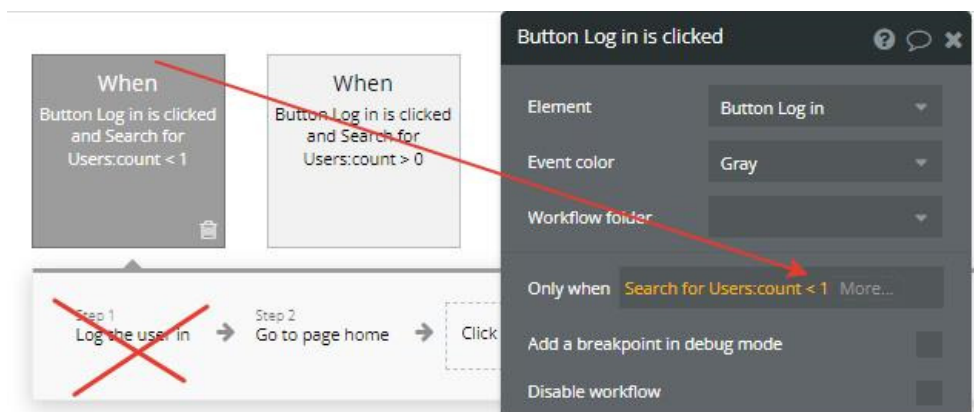


Рис. 5.43. Настройка проверки условия

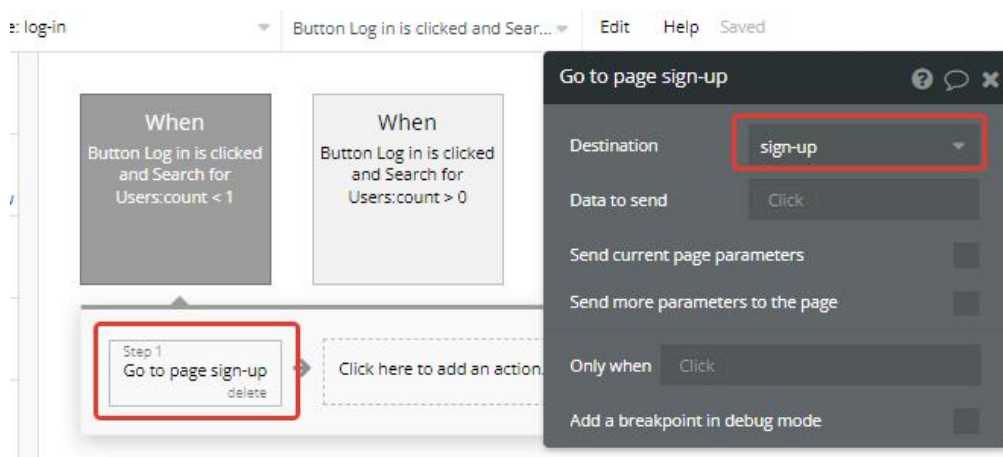


Рис. 5.44. Настройка действия

Проведем тестирование настроенных экранов в режиме предварительного просмотра (рис. 5.45).

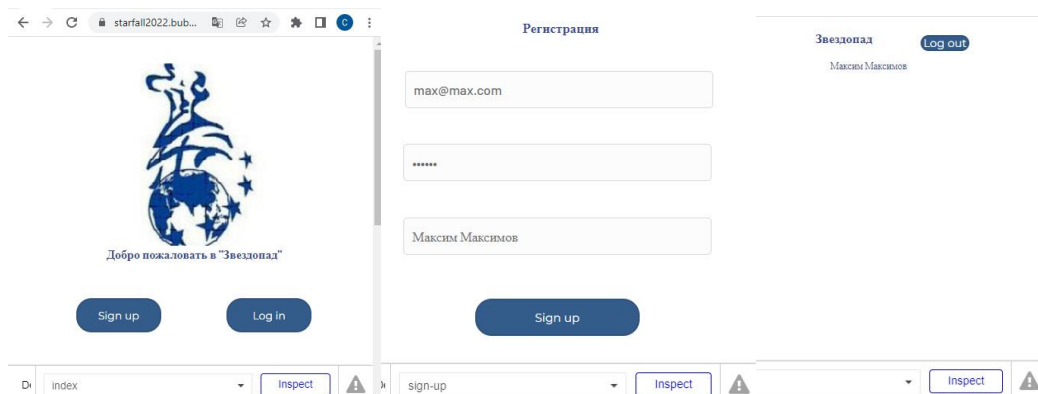


Рис. 5.45. Регистрация

Дальнейшее изучение возможностей инструмента возможно с помощью обучающих материалов [6–8].

6. Лабораторная работа

«Разработка мобильного приложения»

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

Формируемые знания, умения и навыки: научиться работать с no-code-инструментами; 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. Провести мини-исследование и выяснить, как эта проблема решается конкурентными продуктами. Замечательно, если разрабатываемый сервис будет иметь уникальную потребительскую ценность, но и создание клона какого-либо существующего приложения для достижения учебных целей возможно.

2. Разработать схему базы данных приложения. Определить бизнес-логику приложения: карту путешествия пользователя, описать ее в виде последовательности экранов, продумать навигацию.

3. С использованием no-code-инструментов создать мобильное приложение (можно использовать стоковые фотографии и видео; если используется контент, принадлежащий реально существующему бизнесу, то у его владельцев необходимо получить разрешение).

4. Провести тестирование приложения, исправить выявленные ошибки (например, в настройках компонентов, в навигации и т.п.).

5. Провести интервью с первыми пользователями, собрать от них фидбек и проанализировать его. По результатам анализа предложить как минимум одну идею по доработке приложения с точки зрения функциональности. А также исправить технические ошибки, выявленные во время пользовательского тестирования.

6. Предложить модель монетизации приложения.

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?
5. Что удалось реализовать, а что не получилось из-за ограничений используемого визуального конструктора?
6. Какую модель монетизации целесообразнее использовать для приложения? Почему?
7. Как видите дальнейшее развитие приложения?

Задание для самостоятельной работы

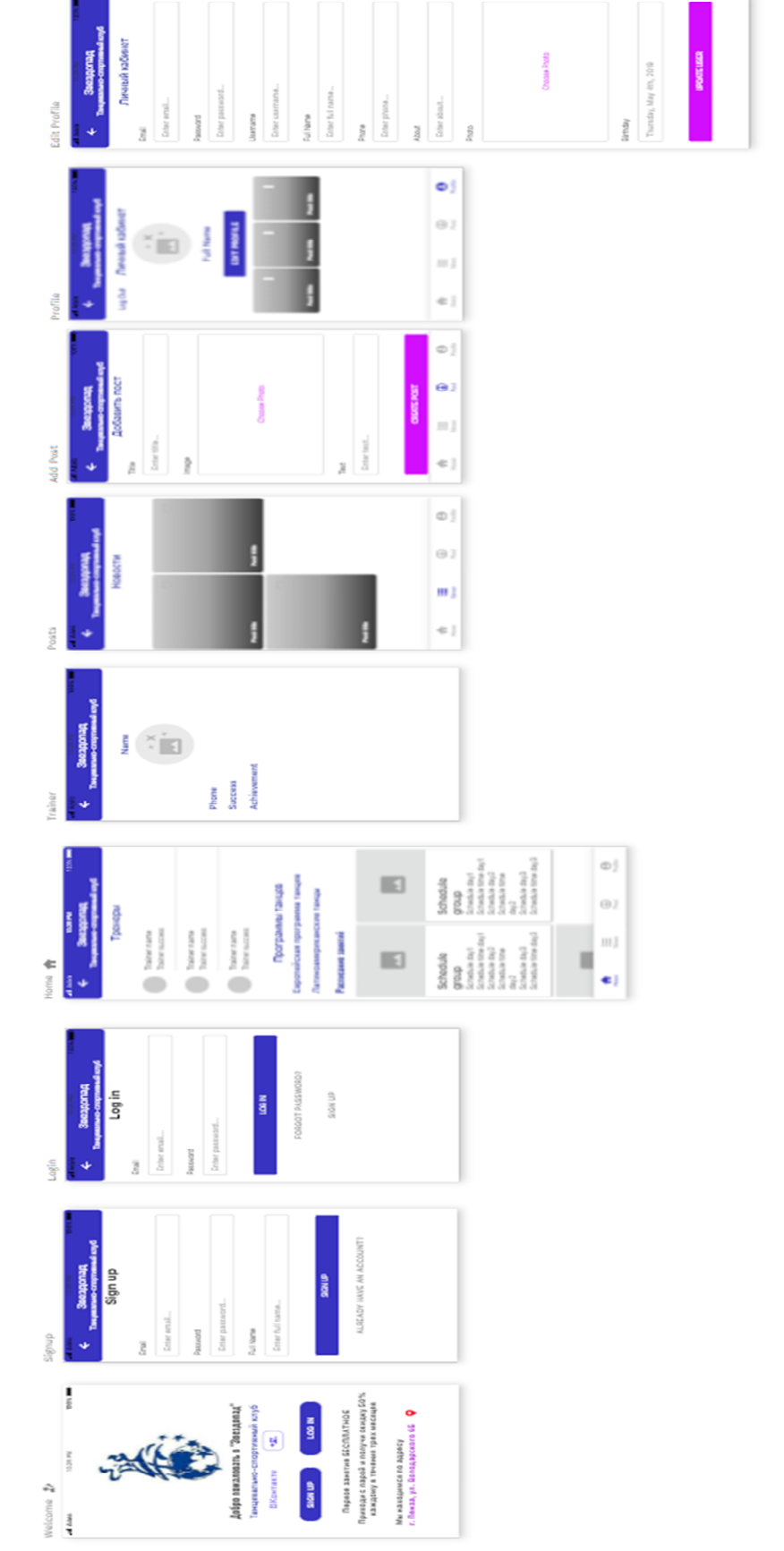
Описать свой опыт по созданию приложения в виде статьи или эссе: какие задачи планировалось решить, как происходила работа над приложением, с какими трудностями столкнулись и как они были преодолены, насколько довольны полученным результатом, какое видение продукта у вас сейчас, что нужно было сделать иначе.

Список литературы

1. NoCode революция, или как разработать MVP за неделю. URL: <https://vc.ru/life/155005-nocode-revolyuciya-ili-kak-razrabotat-mvp-za-nedelyu> (дата обращения: 27.12.2021).
2. Рындина С. В. Интеллектуальные информационные системы и технологии: smart-устройства и мобильные приложения : учеб.-метод. пособие. Пенза : Изд-во ПГУ, 2022.
3. Glide Library. URL: <https://docs.glideapps.com/all/> (дата обращения: 27.12.2021).
4. Adalo Help. URL: <https://help.adalo.com/> (дата обращения: 27.12.2021).
5. Adalo Learn. URL: <https://www.adalo.com/learn> (дата обращения: 27.12.2021).
6. Academy Bubble. URL: <https://bubble.io/academy> (дата обращения: 27.12.2021).
7. Manual Bubble. URL: <https://manual.bubble.io/> (дата обращения: 27.12.2021).
8. Мануал по Bubble. URL: <https://prostokosaka.gitbook.io/bubble/> (дата обращения: 27.12.2021).

Приложение

Настроенные экраны



Учебное издание

Рындина Светлана Валентиновна

**Управление цифровым продуктом:
создание мобильных приложений**

Редактор *В. В. Устинская*
Технический редактор *М. Б. Жучкова*
Компьютерная верстка *М. Б. Жучковой*

Подписано в печать 30.03.2022.
Формат 60×84¹/₁₆. Усл. печ. л. 5,35.
Тираж 7. Заказ № 179.

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

