

Приложение Shoppingmall

Тема

Создание мобильного приложения для посетителей крупного торгового центра.

Легенда

В вашу компанию обратился заказчик — владелец торгово-развлекательного центра в одном из городов-миллионников России. Ему необходимо мобильное приложение для посетителей, которое должно:

- 1) помогать пользователям ориентироваться в товарах и услугах ТРЦ;
- 2) мотивировать возвращаться в ТРЦ благодаря программе лояльности.

В результате вы создадите полностью готовое к релизу приложение, которое обеспечивает выполнение бизнес-целей заказчика и имеет хорошую графическую реализацию.

ТЗ на приложение для ТРЦ

Обязательная часть

Первый запуск приложения

Работа с API:

- При первом старте приложения (то есть если нет сохранённого `mobileDeviceId`) выполняется [запрос](#). Передаются параметры:
 - `install_app` — время запуска приложения в формате `timestamp`;
 - `os` = «iOS»;
 - `version_os` — версия iOS текущего устройства;
 - `model` — модель устройства.
- Полученный в ответе параметр `mobileDeviceId` нужно сохранить и использовать при всех следующих запросах.

Онбординг

Онбординг отображается у всех новых пользователей при первом запуске приложения.

Стандартная работа онбординга:

- свайпнуть (перейти на следующий экран),
- нажать кнопку «Пропустить».

[Макеты](#)

Слайд с запросом уведомлений

Этот слайд можно пролистать. На нём есть две кнопки: «Разрешить» и «Позже».

- Кнопка «Позже» просто пролистывает слайд.
- При нажатии на кнопку «Разрешить» вызывается системный алерт с запросом на отправку уведомлений. При подтверждении уведомлений («Да» в алерте) включаются уведомления (см. подробнее в «Меню» → «Экран настроек»).

Макеты

Домашний экран (раздел таббара)

Домашний экран состоит из блоков, которые содержат карточки с информацией из других разделов приложения. В админке должна быть возможность выбирать, какой контент попадает в эти карточки: будет подтягиваться изображение и текстовая информация (что именно, определится в дизайне).

Карточки располагаются в слайдерах.

- По тапу на карточку происходит переход на экран соответствующего объекта.
- По тапу на кнопку «Все» — переход на экран соответствующего раздела.

Если для блока в админке не выбрано ни одного объекта, он не отображается.

Макеты

Первый блок — главный слайдер

Сюда могут попадать карточки из разделов «Новости», «Мероприятия».

Работа с API:

- Вызов [метода](#) с параметром `on_main=true`.

Второй блок — «Новые предложения»

Сюда могут попадать новые предложения из программы лояльности.

Работа с API:

- Вызов [метода](#) с параметром `on_home=true`.

Третий блок — «Полезная информация»

Сюда могут попадать карточки из раздела «Услуги» (в каталоге).

Работа с API:

- Вызов [метода](#) с параметрами `slug=services` и `on_home=true`.

Четвертый блок — «Мероприятия»

Сюда могут попадать карточки из раздела «Мероприятия».

Работа с API:

- Вызов [метода](#) с параметром on_home=true.

Каталог (раздел таббара)

Основной экран

Содержит список категорий.

В кнопке каждой категории будет название категории и изображение или иконка.

[Макеты](#)

Данные экрана

Данные для экрана хранятся в коде приложения.

Для перехода на нужный экран категорий используется соответствие кнопок и slug:

- Магазины — shops.
- Еда — food.
- Развлечения — entertainment.
- Услуги — services.
- Фитнес — sport.
- Кино — cinema.

Экран категории

Поиск работает нативно. Ищет объект (магазин, кафе) по названию. Должен искать объект как на оригинальном языке (англ.), так и на кириллице. Если по запросу ничего не найдено — должно быть предусмотрено сообщение об этом.

Фильтр сортирует объекты по подкатегориям (поле объекта в админке).

Витрина табличного типа. Содержит:

- лого объекта;
- название.

[Макеты](#)

Работа с API:

- Список формируется с помощью [метода](#) с параметром slug, полученным на предыдущем экране.
- С помощью [метода](#) с параметром slug формируются данные о наличии дополнительных фильтров и ID категории.

- Список подкатегорий формируется с помощью [метода](#) с параметром categoryId, полученным из предыдущего запроса.

Экран объекта

Состоит из трёх блоков. В первом блоке отображаются:

- логотип объекта;
- этаж, на котором объект находится;
- ссылка на сайт объекта.

В блоке «Предложения за баллы» выводятся предложения этого объекта. Если ни одного предложения нет, блок не отображается.

Третий блок — текстовый (описание объекта).

[Макеты](#)

Работа с API:

- Данные для экрана формируются [запросом](#).
- Данные для списка предложений формируются [запросом](#).

Бонус (раздел таббара)

Программа лояльности для авторизованных пользователей.

Основной экран

Содержит список подразделов.

В кнопке будет название подраздела и иконка или изображение (определится дизайном).

[Макеты](#)

Экран каталога подарков

Витрина предложений, которые можно приобрести за баллы.

В карточке предложения отображаются:

- изображение;
- «стоимость» предложения;
- название объекта;
- название предложения;
- дисклеймер.

[Макеты](#)

Работа с API:

- Данные для экрана формируются [запросом](#).

Экран предложения

Отображается вся информация о предложении:

- изображение;
- название объекта;
- название предложения;
- дисклеймер;
- описание;
- условия получения.

При нажатии на крестик экран предложения закрывается → происходит возврат на экран, с которого его открывали.

На кнопке «Приобрести» указана «стоимость» предложения.

[Макеты](#)

Работа с API:

- Данные для экрана формируются [запросом](#).

Меню (раздел таббара)

Основной экран

Содержит список подразделов меню в табличном представлении.

[Макеты](#)

Экран настроек

Блок «Уведомления»

В этом блоке включаются и отключаются уведомления. Для пользователей, которые включили уведомления (в онбординге или позже), должны быть активированы уведомления «Мероприятия в ТРЦ».

[Макеты](#)

Экран новостей и акций

Лента с карточками новостей и акций. Карточки новостей и акций немного различаются.

В карточке новости выводятся:

- изображение новости;
- название новости.

В карточке акции выводятся:

- изображение акции;
- название акции;
- дата, до которой действует акция.

Макеты

Работа с API:

- Данные для экрана формируются [запросом](#).

Экран новости

Отображается вся информация о новости:

- изображение;
- название новости;
- дата публикации;
- описание.

При нажатии на крестик экран новости закрывается → происходит возврат на экран, с которого её открывали.

Макеты

Работа с API:

- Данные для экрана формируются [запросом](#).

Экран мероприятий

Лента с карточками мероприятий.

Макеты

Работа с API:

- Данные для экрана формируются [запросом](#).

В карточке мероприятия выводятся:

- изображение;
- стикер (если пользователь зарегистрирован на это мероприятие);
- название мероприятия;
- дата проведения.

Макеты

Работа с API:

- Данные для экрана формируются [запросом](#).

Экран «Как добраться»

Вебвью со страницей [«Яндекс Карты»](#).

[Макеты](#)

Экран «О приложении»

Состоит из двух блоков:

- Оценка приложения.
- Ссылки на документацию.

Блок оценки приложения

Кнопка «Оценить» становится активной только после выбора определённого количества звёзд.

- Если оценка от 1 до 4 (включительно), по тапу на кнопку открывается экран для написания отзыва.
- Если оценка — 5, по тапу на кнопку открывается модальное окно с оценкой приложения в форме оценки курса Skillbox.

После оценки приложения (любой) пользователь возвращается на экран «О приложении», где блок оценки у этого пользователя пропадает.

[Макеты](#)

Работа с API:

- Данные для экрана формируются [запросом](#).

Экран для написания отзыва

Стандартная форма для отзыва. Отображается текстовое поле для написания отзыва. Текстовое поле активно, и сразу отображается клавиатура. При нажатии на кнопку «Отправить» (на клавиатуре) данные из формы отправляются на ваш имейл.

[Макеты](#)

Работа с API:

- Данные для экрана формируются [запросом](#).

Дополнительная часть

Последний слайд онбординга

- Кнопка «Пропустить» отправляет на домашний экран.
- Кнопка «Регистрация» — на экран регистрации.
- Кнопка «У меня уже есть аккаунт» — на экран авторизации.

[Макеты](#)

Работа с API:

- Полученный push-токен отправляется на бэкенд [запросом](#).

Регистрация

Общая информация

Пользователь может зарегистрироваться либо при первом использовании приложения после онбординга, либо позже в любое время — в настройках. Для неавторизованных пользователей в настройках будут доступны кнопки «Войти» и «Зарегистрироваться». Некоторые разделы приложения должны быть доступны только авторизованным пользователям. Для неавторизованных на таких экранах будут заглушки (или алерты о необходимости зарегистрироваться).

За регистрацию будут начисляться баллы (подробнее о баллах — в разделе «Экран „Получите баллы“»).

- Регистрация должна выполняться по номеру телефона. При регистрации его необходимо верифицировать кодом из СМС (то есть аккаунт будет привязан к номеру).
- Авторизация выполняется с помощью ввода номера телефона, после чего придёт СМС-сообщение с кодом, который, в свою очередь, нужно ввести в приложении.

Экран для введения номера телефона при регистрации

Сразу отображается цифровая клавиатура. Проверяем поле для номера телефона на правильность ввода. Кнопка «Получить код» становится активной при вводе полного номера. При нажатии на кнопку отправляем СМС с кодом верификации и переходим на экран для введения кода (в тестовой версии коды не отправляются и могут быть просмотрены в админке).

[Макеты](#)

Работа с API:

- После ввода номера телефона вызывается [запрос](#).

Экран для введения кода из СМС

Сразу отображается цифровая клавиатура. Должна быть предусмотрена возможность повторной отправки кода (кнопка «Отправить повторно» появляется спустя 15 секунд после загрузки экрана).

После введения кода происходит автоматическая проверка.

- Если код введён неверно, показываем алерт (см. экран в прототипе).
- Если верно — загружаем следующий экран.

[Макеты](#)

Работа с API:

- Для проверки полученного кода вызывается [запрос](#). В него передаётся номер телефона, введённый код и текущий `mobileDeviceId`.

Экран для введения данных профиля

Поле для ввода имени и фамилии сразу активно, и отображается буквенная клавиатура. Для поля, в которое необходимо ввести имейл, отображается соответствующая клавиатура (с @).

Оба поля проверяем на правильность ввода. Если есть ошибки, выделяем поля и показываем сообщение о необходимости проверки введённых данных. Кнопка «Сохранить» становится активной только после правильного ввода данных. После сохранения данных профиля начисляем пользователю баллы и показываем экран для начисления баллов.

[Макеты](#)

Работа с API:

- Для отправки введённых данных используется [запрос](#).

Экран для начисления баллов

Отображаются:

- количество начисленных баллов;
- кнопка, которая закрывает этот экран.

[Макеты](#)

Работа с API:

- Для отправки введённых данных используется [запрос](#).

- После успешной отправки выполняется [запрос](#). В ответе приходит начисленное количество баллов, которое показываем [на экране](#).

Экран для введения промокода

В приложении должна быть реферальная программа.

У каждого зарегистрированного пользователя есть реферальный код. Если новый пользователь вводит этот код, обоим пользователям начисляются баллы. Экран для введения промокода доступен один раз — при регистрации после сохранения данных пользователя.

На экране есть возможность его закрыть («Нет промокода»). При этом происходит переход на следующий экран — экран опроса.

Кнопка «Готово» становится активной только после введения кода полностью. После нажатия кнопки код валидируется.

- Если такого кода нет, показываем алерт.
- Если есть — начисляем баллы обоим пользователям (тому, кто вводит код, показываем экран для начисления баллов).

[Макеты](#)

Работа с API:

- Для отправки введенных данных используется [запрос](#), и передается поле `referral_code`.

Экран опроса

После введения промокода или закрытия экрана для его введения новым пользователям предлагается пройти опрос. Однако его можно пропустить (при этом происходит переход на домашний экран).

В текущем опросе два вопроса и два поля для ввода. Оба поля обязательны для заполнения. Клавиатура появляется после нажатия на кнопку «Получить баллы» или после тапа на поле.

- У поля ввода даты рождения — системный `date picker`.
- У второго поля ввода информации о наличии детей — системный `option picker`.

После заполнения обоих полей начисляем пользователю баллы, показываем экран для начисления баллов. При закрытии экрана для начисления баллов происходит переход на домашний экран.

Данные о факте прохождения опроса и ответы на вопросы сохраняются в профиле пользователя в админке.

[Макеты](#)

Работа с API:

- Для отправки введенных данных используется [запрос](#). Передаются только поля date_birth и have_kids.
- После успешной отправки выполняется [запрос](#). В ответе приходит численное количество баллов, которое показываем [на экране](#).

Авторизация

Всё, что есть в приложении, доступно как авторизованным, так и неавторизованным пользователям. Но чтобы воспользоваться функциональностью (например, программой лояльности), нужно авторизоваться (см. все экраны в прототипе).

Пользователь может авторизоваться либо при первом использовании приложения после онбординга, либо позже — в настройках. Для неавторизованных пользователей в настройках будут доступны кнопки «Войти» и «Зарегистрироваться» (см. макет). Авторизация происходит по номеру телефона и коду, отправляемому в СМС на этот номер.

Экран для введения номера телефона при авторизации

Сразу отображается цифровая клавиатура. Проверяем поле номера телефона на правильность ввода. Кнопка «Получить код» становится активной при вводе полного номера.

После нажатия кнопки «Получить код» проверяем номер телефона на наличие такого аккаунта.

- Если аккаунта ещё нет, показываем соответствующий алерт.
- Если есть — отправляем СМС с кодом для входа и переходим на экран для введения кода.

[Макеты](#)

Работа с API:

- После ввода номера телефона вызывается [запрос](#).

Экран для введения кода из СМС

Сразу отображается цифровая клавиатура. Предусмотрена возможность повторной отправки кода (кнопка «Отправить повторно» появляется спустя 15 секунд после загрузки экрана).

После введения кода происходит автоматическая проверка.

- Если код введен неверно, показываем алерт.
- Если верно — переходим на домашний экран.

[Макеты](#)

Работа с API:

- Для проверки полученного кода вызывается [запрос](#). В него передаётся номер телефона, введённый код и текущий mobileDeviceId.

Экран настроек

Блок «Данные профиля»

Отображается введенная пользователем при регистрации информация. Если пользователь не авторизован, отображаются кнопки «Вход» и «Регистрация».

Данные можно редактировать.

- Имя и имейл после редактирования можно просто сохранить.
- Номер телефона (если пользователь сохраняет новый) необходимо верифицировать (процесс как при регистрации).

При редактировании и сохранении новых данных записываем их в профиль пользователя в админке (номер телефона переписываем только после удачной верификации).

[Макеты](#)

Работа с API:

- Для отправки введённых данных используется [запрос](#).
- Для получения данных для отображения используется [запрос](#).

Кнопка «Выйти из профиля»

При нажатии на кнопку происходит выход из профиля (дополнительно показываем алерт с просьбой подтвердить выход).

[Макеты](#)

Работа с API:

- Для выхода из учётной записи используется [запрос](#).

Бонус (раздел таббара)

Это программа лояльности для авторизованных пользователей.

Основной экран

Этот раздел доступен только авторизованным пользователям. Если пользователь не авторизован — показываем алерт (см. экран в прототипе).

Содержит список подразделов.

В кнопке будет название подраздела и иконка или изображение (определится дизайном).

Макеты

Экран «Получите баллы»

У каждого зарегистрированного пользователя есть уникальный реферальный код. Если новый пользователь вводит этот код, обоим начисляются баллы. Должна быть возможность скопировать и расшарить код (нативное расшаривание).

Работа с API:

- Для получения данных для отображения используется [запрос](#) и поле `referral_code`.

Макеты

Экран каталога подарков

Чтобы воспользоваться предложением, необходимо тапнуть на кнопку «Воспользоваться». При этом:

- Вместо описания и условий получения предложения показываем условия подтверждения.
- Кнопка «Воспользоваться» меняется на «Предложение получено».

По тапу на кнопке «Предложение получено» показываем алерт. При подтверждении получения предложения («Да» в алерте):

- Удаляется предложение из раздела «Мои предложения» этого пользователя.
- Не важно, из какого раздела пользователь теперь будет переходить в это предложение, — везде должна быть кнопка «Приобрести».
- Добавляется в админке запись об использовании предложения пользователем.

Макеты

Работа с API:

- Данные для экрана формируются [запросом](#).

Экран «Мои подарки»

На этом экране отображаются приобретённые пользователем подарки, которые можно использовать. Если у пользователя нет ни одного подарка, должно быть предусмотрено сообщение об этом.

Макеты

Работа с API:

- Данные для экрана формируются [запросом](#).

Экран истории транзакций

На этом экране отображается история транзакций:

- Начисление бонусов по реферальной программе (название реферальной программы, количество начисленных бонусов).
- Списание бонусов при приобретении предложений (название предложения, объект, количество списанных бонусов).
- Списание бонусов при регистрации на мероприятие (название мероприятия, количество списанных бонусов).

Транзакции разнесены по блокам — датам.

Макеты

Работа с API:

- Данные для экрана формируются [запросом](#).

Требования к приложению

- Приложение использует Swift/UIKit или SwiftUI. Выбирайте технологию осознанно, оценивая свой текущий уровень владения ею. Это сильно увеличит шансы на успешное выполнение задания.
- Код оформлен в виде репозитория, работа над проектом ведётся в виде последовательных коммитов с разработкой текущего приложения.
- Обработываются все необходимые состояния экранов: ошибка, лоадинг, «пусто».
- Приложение умеет адекватно реагировать на пропажу Сети: показывает соответствующие уведомления, не «крашится».
- У приложения есть иконка и корректное название.
- Приложение поддерживает минимум два языка: русский и английский.
- В приложении реализован UITabBarController для главных экранов.
- На экранах со списками реализована возможность pull-to-refresh — перезагрузки данных.
- Минимальная поддерживаемая версия iOS — 15.
- Куратор может попросить изменить дизайн в рамках текущего технического задания — так бывает в реальной разработке, когда бизнес начинает принимать работу у программистов.

Дополнительные материалы

- Для работы можете использовать [макет в Figma](#).
- [Общее описание API](#).

Критерии оценки

- Нет багов и вылетов в приложении.
- Выполнены все требования к приложению.
- В приложении реализован функционал, описанный в сценариях использования.
- Приложение имеет завершённый дизайн, соответствующий предоставленным эскизам.
- Каждый экран в приложении реализован с помощью паттерна проектирования.
- Технический слой работы с сетью и базами данных абстрагирован от UI с помощью протоколов.
- Навигация в приложении корректно работает, используются UINavigationController и UITabBarController.