

тестового проекта

Занятие 4

ИТ-программное решение для бизнеса

Независимый дизайнер тестового проекта: Рамин Мохаммаддуст

Независимый валидатор тестового проекта: Афшин Дехгани

Введение

На этом занятии вы выступите в роли разработчика мобильного приложения. Вы создадите приложение для заказчиков Belle Croissant Lyonnais. Данное приложение позволит заказчикам просматривать товары, заказывать и отслеживать свои заказы.

Основное внимание уделяется тому, чтобы сделать приложение простым в использовании и приятным на вид. Оно также должно плавно подключаться к системе, созданной в ходе Занятия 3.

На этом занятии вы проверите свои навыки по следующим направлениям:

- **Разработка мобильных приложений:** создание приложений, хорошо работающих на смартфонах.
- **Интеграция с API:** Подключение приложения к существующей системе.
- **Разработка пользовательского интерфейса:** Создание простого и приятного в использовании приложения.
- **Решение проблем:** Поиск и устранение неполадок по мере создания приложения.

Содержание

Данный учебный пакет содержит следующие материалы:

1. **Session Instructions (PDF):** Подробные инструкции с описанием задач, которые необходимо выполнить, и ожидаемых результатов для этого занятия.
2. **Common Folder:** Эта папка содержит дополнительные ресурсы, такие как логотип, значки, руководство по стилю и другие элементы дизайна Belle Croissant Lyonnais, которые могут быть использованы при разработке приложения.
3. **API Documentation (PDF):** Документация по API продукта/заказчика/заказа, содержащая подробную информацию о конечных точках, форматах запросов/ответов и аутентификации.

Описание проекта и задач

На этом занятии вы усовершенствуете мобильное приложение Belle Croissant Lyonnais, добавив в него новые функции для индивидуальных заказов, управления пользователями и обработки заказов.

Методические рекомендации:

1. **Простота в использовании:** Сделайте приложение простым и понятным для заказчиков.
2. **Привлекательный вид:** Для оформления приложения следуйте руководству по стилю Belle Croissant Lyonnais.
3. **Корректная работа:** Убедитесь, что все части приложения работают корректно и без ошибок.
4. **Безопасность:** Защитите информацию о пользователях и сделайте приложение безопасным в использовании.
5. **Своевременность:** Выполните все задания в установленный срок.

Технические особенности:

1. **Интеграция с API:** При работе с внутренними API следуйте рекомендациям, приведенным в руководстве "Web API Design: The Missing Link".
2. **Проверка данных:** Перед отправкой в API убедитесь, что введенные пользователем данные верны и полны.
3. **Обработка ошибок:** Отображение четких сообщений пользователю при возникновении каких-либо проблем.

Дополнительные факторы:

- Приложение должно работать плавно и быстро на предоставленном мобильном устройстве.
- Используйте четкие надписи и инструкции для всех элементов UI.
- Организуйте информацию таким образом, чтобы она была понятна пользователям.
- Убедитесь, что приложение доступно для пользователей с ограниченными возможностями.

Инструкции для участника

4.1 Разработка базы данных для управления пользователями

Цель:

Разработайте нормализованную схему базы данных для поддержки функций управления пользователями Belle Croissant Lyonnais в их мобильном приложении. Схема должна эффективно хранить пользовательские данные и управлять ими, включая учетные данные для проверки подлинности, информацию профиля, адреса и подписки на списки рассылки.

Задача:

Проанализируйте приведенные ниже примеры пользовательских данных и создайте диаграмму Entity-Relationship Diagram (ERD), которая может соответствовать этим данным и любым будущим потребностям в расширении:

```
User 1:
  Адрес электронной почты: marie.dupont@example.com
  Пароль: MyPassword1
  Имя: Мари
  Фамилия: Дюпон
  Номер телефона: +33 6 12 34 56 78
  Подписка на рассылку: Да,
  Фотография профиля: /images/marie_dupont.jpg
  Секретный вопрос: Девичья фамилия матери?
  Безопасный ответ: (хэшированный)
  Адреса доставки:
    Дом: 123 Rue de la République, Lyon, 69002, France Место
    работы: 456 Avenue Jean Jaurès, Lyon, 69003, France (предпочтительно)
  Предпочтительный способ доставки: Доставка
```

User 2:
Адрес электронной почты: john.smith@example.com
Пароль: 12John33
Имя: Джон
Фамилия: Смит
Номер телефона: +33 7 89 01 23 45
Подписка на рассылку: Нет
Фото профиля: /images/john_smith.jpg
Секретный вопрос: Кличка первого питомца?
Надежный ответ: (хэширован)
Адреса доставки:
Адрес дома: 789 Rue Garibaldi, Lyon, 69007, France
Предпочтительный способ доставки: Самовывоз.

Результаты:

- **Имя файла:** Session4_UserDB_ERD.pdf
- **Тип файла:** Отчет в формате PDF, содержащий:
 - Снимок экрана с реализованной базой данных в Microsoft SQL Server Management Studio (SSMS).

Дополнительные примечания:

База данных должна иметь возможность хранить несколько адресов доставки для каждого пользователя, при этом один из них должен быть помечен как предпочтительный адрес.

- База данных должна отслеживать, подписался ли пользователь на рассылку.
- База данных должна надежно хранить пароли пользователей и ответы на секретные вопросы.
- В базе данных должен храниться путь к файлу или URL-адрес изображения профиля пользователя.
- В базе данных должен храниться предпочтительный способ доставки, выбранный пользователем (самовывоз или доставка по почте).
- База данных должна быть спроектирована по принципам масштабируемости и гибкости, что позволяет в будущем вносить дополнения или изменения в модель данных.

4.2 Разработка API для управления пользователями

Цель:

Внедрить внутренний API для системы управления пользователями Belle Croissant Lyonnais, придерживаясь схемы базы данных, разработанной в задаче 4.1, и интегрируясь с основным API продукта/заказа.

Задача:

Реализуйте следующие конечные точки RESTful API:

- **Идентификация:**
 - **POST /api/register:** Создание новой учетной записи пользователя.
 - **POST /api/login:** Аутентификация пользователя.
 - **POST /api/forgot-password:** Инициирование сброса пароля (отправьте секретный вопрос).
 - **POST /api/reset-password:** Сброс пароля после проверки ответа системы безопасности.

- **Управление профилем:**
 - **GET /api/profile:** Получение профиля пользователя.
 - **PUT /api/profile:** Обновление профиля (включая загрузку изображений).
 - **GET /api/addresses:** Получение адресов пользователей.
 - **POST /api/addresses:** Добавление нового адреса.
 - **PUT /api/addresses/{id}:** Обновление существующего адреса по идентификатору.
 - **DELETE /api/addresses/{id}:** Удаление адреса по идентификатору.
- **Подписка:**
 - **POST /api/subscribe:** Подписка на список рассылки.
 - **POST /api/unsubscribe:** Отписка от рассылки.
- **Соответствие требованиям заказчиков:**
 - **GET /api/customers:** Получение данных о заказчиках из основного API Belle Croissant Lyonnais на основе предоставленного электронного письма.

Результаты:

- **Запущенный API:** Развернутый API, доступный в конкурентной среде (например, localhost).
- **Session4_UserAPI_Endpoints.txt :** Текстовый файл со всеми реализованными конечными точками API, HTTP-методами, обязательными/необязательными параметрами и примерами полезной нагрузки запроса/ответа в формате JSON.

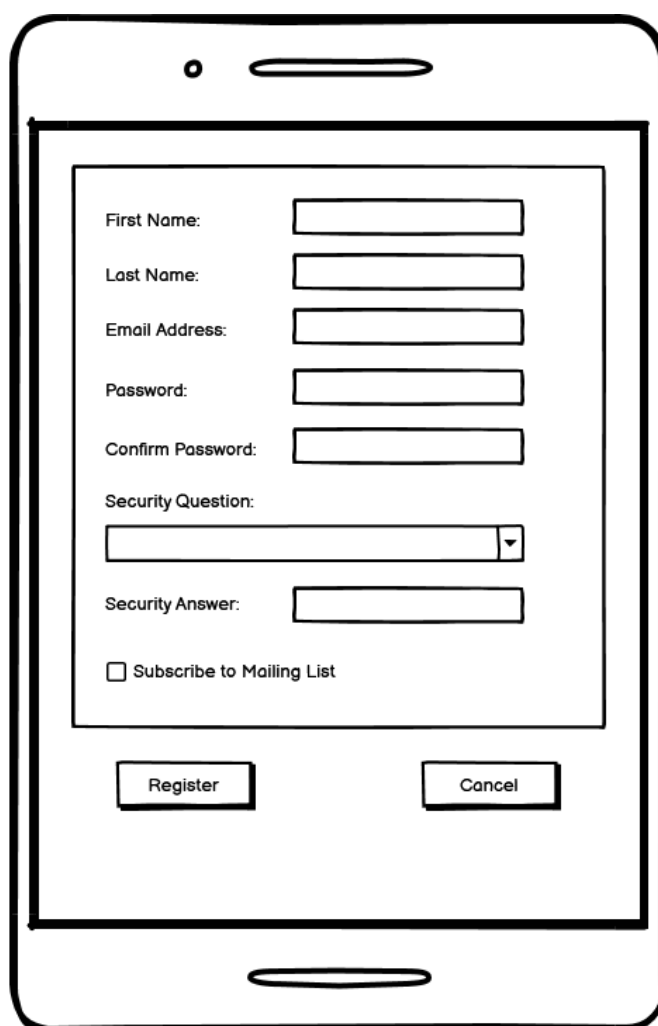
Дополнительные примечания:

- В качестве руководства используйте документ "Web API Design Best Practices.pdf".
- Предположим, что мобильное приложение будет обрабатывать выбор и отображение изображений, а API нужно только сохранять и извлекать URL-адреса изображений или пути к ним.

4.3 UI экрана регистрации

Цель:

Разработать и внедрить удобный экран регистрации для мобильного приложения Belle Croissant Lyonnais. Этот экран должен позволить новым заказчикам создавать учетные записи и подписываться на рассылку.



First Name:

Last Name:

Email Address:

Password:

Confirm Password:

Security Question:

Security Answer:

☐ Subscribe to Mailing List

Register Cancel

Задача:

1. Разработайте и реализуйте элементы пользовательского интерфейса, как показано на рисунке.

- **Поля ввода:**
 - Имя
 - Фамилия
 - Адрес электронной почты
 - Пароль
 - Подтвердить пароль
 - Секретный вопрос (выпадающее меню с заранее определенными параметрами, полученными из API)
 - Безопасный ответ
- **Флажок:**
 - "Subscribe to Mailing List"
- **Кнопка:**

- "Register" и "Cancel"

2. Реализация функциональности:

- Проверьте правильность введенных пользователем данных (убедитесь, что все обязательные поля заполнены, формат электронной почты правильный, пароли совпадают и т.д.).
- Отправьте запрос на регистрацию в конечную точку API /api/register.
- Отображение соответствующих сообщений об ошибках при неправильном вводе данных или сбоях регистрации.
- После успешной регистрации пользователь автоматически войдет в систему и перейдет к экрану профиля.

Результаты:

- Интегрируйте экран регистрации в мобильное приложение Belle Croissant Lyonnais.

4.4 UI экрана входа в систему

Цель:

Разработать и внедрить удобный экран входа в мобильное приложение Belle Croissant Lyonnais, обеспечивающий надлежащее взаимодействие с API управления пользователями и безопасную аутентификацию.

Задача:

1. Реализация UI:

- Разработайте и реализуйте экран входа в систему, как показано на схеме.
- Включите следующие элементы:
 - Поле для ввода адреса электронной почты с проверкой подлинности
 - Поле для ввода пароля с проверкой подлинности
 - Кнопка "Login"
 - Кнопка "Create Account" (для перехода к экрану регистрации)
 - Ссылка "Forgot Password?" (для перехода к экрану сброса пароля)

2. Функциональность:

- Проверьте правильность введенных пользователем данных (убедитесь, что указаны адрес электронной почты и пароль).
- Отправьте запрос на вход в систему в конечную точку API /api/login.
- Если вход в систему прошел успешно:
 - Надежно храните полученный токен JWT для будущих запросов к API.
 - Перейдите к экрану профиля.
- Если войти в систему не удалось, выведите на экран соответствующее сообщение об ошибке пользователя (например, "Invalid credentials").

Результаты:

- Интегрируйте экран входа в систему в мобильное приложение Belle Croissant Lyonnais.
- **Создание тестового пользователя:** Создайте новую учетную запись пользователя в своей базе данных управления пользователями со следующими учетными данными:
 - **Имя пользователя (Email):** testuser@belle-croissant.com
 - **Пароль:** BCLpass.WSL24

4.5 UI экрана "Forgot Password"

Цель:

Разработать и внедрить удобный экран "Forgot Password" для мобильного приложения Belle Croissant Lyonnais.

Задача:

1. **Разработайте и реализуйте элементы пользовательского интерфейса:** Создайте следующие элементы UI на экране "Forgot Password":
 - **Ввод адреса электронной почты:** Текстовое поле для ввода адреса электронной почты с подтверждением (убедитесь, что адрес электронной почты указан и правильно отформатирован).
 - **Отображение секретного вопроса:** Область отображения секретного вопроса, выбранного из API, отображается после правильного ввода адреса электронной почты (при потере фокуса).
 - **Ввод секретного ответа:** Текстовое поле для ввода ответа на секретный вопрос.
 - **Кнопки:**
 - Кнопка "Verify Answer" позволяет сверить ответ с сохраненным значением, используя конечную точку `/api/reset-password`.
 - Нажмите кнопку "Cancel", чтобы вернуться к экрану входа в систему.
 - **Ввод нового пароля (условный):**
 - Если ответ на вопрос о безопасности правильный, отобразите поля ввода для ввода и подтверждения нового пароля.
 - Нажмите кнопку "Reset Password", чтобы обновить пароль, используя конечную точку `/api/profile`.
2. **Реализация функциональности:**
 - **Отправка запроса на сброс:** При нажатии кнопки "Send Reset Link" отправьте запрос на сброс пароля на конечную точку API `/api/forgot-password` с указанным адресом электронной почты.
 - **Отображение секретного вопроса:** После успешного выполнения запроса отобразите секретный вопрос, полученный из API.
 - **Подтверждение ответа:** После нажатия кнопки "Verify Answer" отправьте электронное письмо и ответ на конечную точку `/api/reset-password` для проверки.

- **Сброс пароля:** Если ответ правильный, включите поля ввода нового пароля и разрешите пользователю установить новый пароль.
- **Обработка ошибок:** Отображение соответствующих сообщений об ошибках при неверном вводе, неправильных ответах на вопросы безопасности или ошибках API.

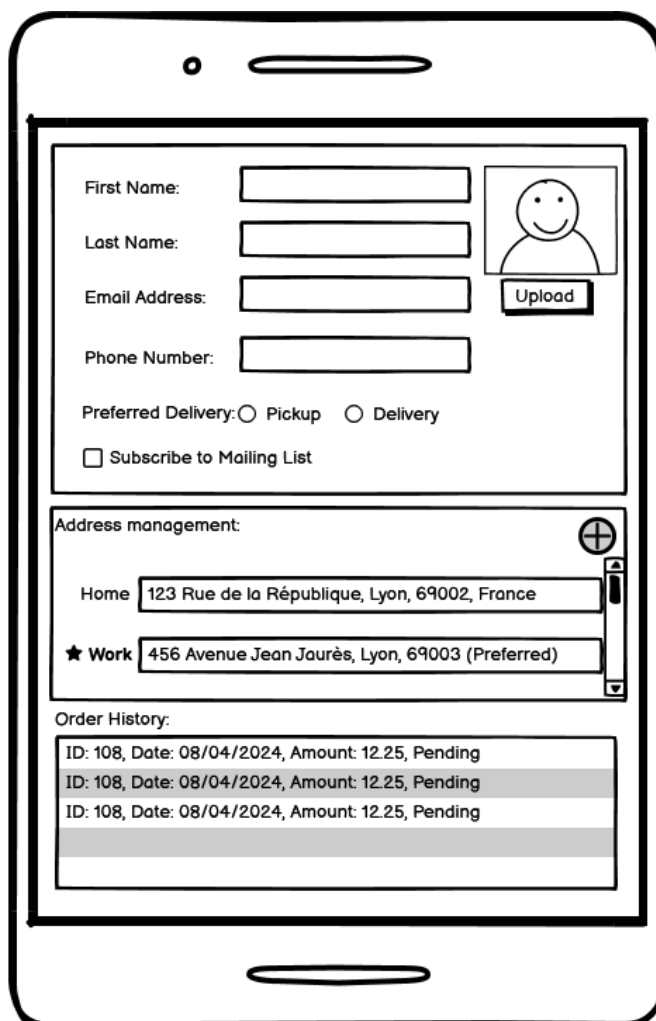
Результаты:

- Интегрируйте экран "Forgot Password" в мобильное приложение Belle Croissant Lyonnais.
- **Создание тестового пользователя:** Создайте новую учетную запись пользователя в своей базе данных управления пользователями со следующими учетными данными:
 - **Электронная почта:** testuser2@belle-croissant.com
 - **Пароль:** BCLpass.WSL24
 - **Секретный вопрос:** "What is your mother's maiden name?"
 - **Секретный ответ:** "Smith".

4.6 UI экрана профиля пользователя

Цель:

Разработка и внедрение экрана профиля пользователя для мобильного приложения Belle Croissant Lyonnais, позволяющего пользователям просматривать и редактировать информацию о своем профиле, управлять адресами, а также просматривать историю своих заказов.



Задача:

1. Отображение профиля и встроенное редактирование:

- **Отображаемые поля (доступные для редактирования):**
 - Имя
 - Фамилия
 - Адрес электронной почты
 - Номер телефона
 - Фотография профиля (с возможностью загрузки / изменения)
 - Подписка на список рассылки (переключатель)
 - Предпочитаемый способ доставки (переключатели: Pickup или Delivery)
- **Встроенное редактирование:**
 - Разрешите пользователям нажимать на поле, чтобы редактировать его непосредственно на экране профиля.

- При редактировании визуально укажите, что поле находится в режиме редактирования (например, измените цвет фона, покажите кнопки "Save" и "Cancel").
- Проверяйте вводимые пользователем данные (например, проверьте формат электронной почты).
- Если проверка завершится неудачно, отобразите сообщение об ошибке рядом с полем и не сохраняйте его до тех пор, пока оно не будет исправлено.
- Сохраняйте изменения в API управления пользователями (/api/profile) только в том случае, если пользователь явно нажимает "Save".
- Обеспечьте визуальную обратную связь (например, сообщение об успешном завершении) после успешного сохранения.
- Используйте кнопку "Cancel", чтобы отменить изменения.

2. Управление адресами:

- **Отображение:**
 - Список сохраненных адресов с их метками и индикатором "Preferred" (если применимо).
 - Возможность добавить новый адрес.
- **Встроенное редактирование:**
 - Разрешите встроенное редактирование адресов, как полей профиля.
 - Проверьте правильность ввода адреса (например, убедитесь, что указан правильный почтовый индекс).
 - Используйте API управления пользователями (/api/addresses, /api/addresses/{id}) для добавления, обновления и удаления адресов.
 - Отметьте адрес как "Preferred", используя API.

3. История заказов:

- **Отображение:**
 - Список прошлых заказов, включая:
 - Идентификатор заказа
 - Дата заказа
 - Общая сумма
 - Статус заказа

Результаты:

- Интегрируйте экран профиля пользователя, функции управления адресами и истории заказов в мобильное приложение Belle Croissant Lyonais.