

ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ _____

ПРЕПОДАВАТЕЛЬ

доцент, кандидат тех. наук				Бржезовский А. В.
должность, уч. степень, звание		подпись, дата		инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

ПРОГРА

Вариант 5

по курсу: МЕТОДЫ И СРЕДСТВА ПРОЕКТИРОВАНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4128			Воробьев В.А.
			подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

СОДЕРЖАНИЕ

1	Постановка задачи	3
1.1	Задание	3
2	Выполнение работы	4
2.1	Возможности бизнеса	4
2.2	Бизнес-цели	4
2.3	Критерии успеха	4
2.4	Положение о концепции проекта	5
2.5	Бизнес-риски	5
2.6	Предположения и зависимости	5
2.7	Требования к системе	6
2.7.1	Определение требований пользователей	6
2.7.1.1	Определение списка пользователей	6
2.7.2	Сценарии пользования	6
2.7.3	Ошибочные сценарии	6
2.7.4	Альтернативные сценарии	7
2.7.5	Функциональные требования	8
2.8	Модель базы данных	9
2.8.1	Процедуры	11
2.8.2	Триггеры	11
2.8.3	Индексы	12
2.9	Описание работы интерфейса	12
	ЗАКЛЮЧЕНИЕ	22
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	23
	ПРИЛОЖЕНИЕ	24

1 Постановка задачи

Цель работы: изучить принципы использования индексации данных.

1.1 Задание

Целью данной курсовой работы является разработка автоматизированной информационной системы для организации юмористических мероприятий. Основная задача разработки данной системы заключается в создании платформы, которая позволит площадкам удобно создавать мероприятия, приглашать артистов и управлять процессами организации, а артистам – находить подходящие мероприятия, просматривать их информацию и записываться на участие. Система призвана упростить взаимодействие между организаторами и артистами, сделав процесс планирования и участия в мероприятиях более прозрачным и удобным. Перед началом разработки необходимо определить перечень правил и требований, описать будущих пользователей системы и их роли, а также детализировать алгоритмы их взаимодействия с системой.

2 Выполнение работы

Для выполнения работы был выбран бекенд фреймворк Ktor и фронтенд фреймворк Flutter.

2.1 Возможности бизнеса

Создание приложения для управления конференциями открывает значительные возможности для улучшения взаимодействия между организаторами и участниками. Система должна упростить процесс регистрации, предоставлять актуальную информацию о мероприятиях и обеспечивать удобный канал связи между всеми заинтересованными сторонами.

2.2 Бизнес-цели

Основные задачи, которые система призвана решить:

- Неэффективность процесса регистрации на конференции: отсутствие унифицированного подхода затрудняет процесс для участников и организаторов.
- Нестабильная коммуникация: это приводит к недопониманиям и задержкам в обмене важной информацией.
- Недостаток доступа к актуальной информации: затрудняет планирование и снижает удовлетворённость участников.

Для решения этих проблем установлены следующие бизнес-цели:

- **БЦ-1:** Унифицировать процесс регистрации на конференции, обеспечивая простую и интуитивно понятную навигацию.
- **БЦ-2:** Улучшить качество и скорость взаимодействия с пользователями через надежные и удобные каналы связи.
- **БЦ-3:** Предоставлять доступ к полной и актуальной информации о мероприятиях, помогая пользователям принимать быстрые и обоснованные решения.

2.3 Критерии успеха

Для оценки успешности внедрения приложения определены следующие критерии:

- **КУ-1:** Более 100 созданных конференций в день.
- **КУ-2:** Сохранение рейтинга приложения выше 4.5 на основных платформах (App Store, Google Play) как показатель высокой удовлетво-

ренности пользователей.

Эти цели и критерии направляют развитие приложения, способствуя улучшению взаимодействий на конференциях и повышая удовлетворенность всех участников процесса, от организаторов до посетителей.

2.4 Положение о концепции проекта

Для организаторов конференций и участников, желающих принять участие в мероприятиях, предлагаемая система «Конференции» является информационной системой, которая обеспечит единую платформу взаимодействия. Это позволит сторонам своевременно получать необходимую информацию друг о друге и значительно упростит процесс регистрации и участия. Система будет включать профили участников и организаторов с важной информацией, включая наличие свободных мест и программу мероприятий. В отличие от существующих каналов, наш продукт специализирован для этой сферы и предоставляет все необходимые инструменты для удобства пользователей конференций.

2.5 Бизнес-риски

- **БР-1:** Организаторы могут не захотеть менять привычные каналы взаимодействия (вероятность – 0.3, ущерб – 7).
- **БР-2:** Возможные сбои в работе приложения могут сорвать ключевые мероприятия и подорвать доверие пользователей (вероятность – 0.1, ущерб – 9).

2.6 Предположения и зависимости

- **Пр-1:** Организаторы и участники будут иметь доступ к системам с соответствующим интерфейсом для взаимодействия и регистрации.
- **Пр-2:** Несколько представителей организаторов будут иметь доступ к системе для повышения скорости отклика и эффективного общения с участниками.
- **Зав-1:** Система должна быть независимой от конкретных устройств, на которых она установлена.
- **Зав-2:** В зависимости от того, кто пользуется системой, её функции и интерфейс должны адаптироваться соответственно, обеспечивая персонализированный опыт для организаторов и участников.

2.7 Требования к системе

2.7.1 Определение требований пользователей

2.7.1.1 Определение списка пользователей

Таблица 2.1 - Список ролей

Роль	Описание
Участник	Целью участника является поиск и регистрация на интересные ему конференции. Участник может просматривать список мероприятий, их расписания и профили организаторов. Он также имеет возможность зарегистрироваться на конференции, отменить регистрацию при необходимости и редактировать свой личный профиль.
Организатор конференции	Организатор отвечает за создание и управление конференцией. Его цель – организовать успешные мероприятия и привлечь участников. Организатор может добавлять и редактировать мероприятия, управлять списком участников и взаимодействовать с ними посредством уведомлений. Он также может редактировать профиль своей конференции.

2.7.2 Сценарии пользования

Конечно, давайте представим ошибочные и альтернативные сценарии в указанном формате:

2.7.3 Ошибочные сценарии

1. Проблемы с загрузкой списка конференций

- 1.a) Приложение пытается загрузить список конференций.
- 1.b) Если интернет-соединение отсутствует, приложение отображает сообщение: “Не удалось загрузить конференции:

Проверьте интернет-соединение и попробуйте снова”.

- 1.c) Если произошла серверная ошибка, приложение выдает сообщение: “Ошибка сервера: Попробуйте повторить чуть позже”.

2. Сбой при записи на конференцию

- 2.a) Участник нажимает кнопку «Записаться» на странице конференции.
- 2.b) Если запись не удалась из-за проблемы с сервером, приложение отображает сообщение: “Запись не выполнена: Попробуйте снова через некоторое время”.
- 2.c) Если произошла другая ошибка, пользователь видит сообщение: “Произошла ошибка: Проверьте подключение и статус конференции”.

3. Попытка записи на уже записанную конференцию

- 3.a) Участник нажимает кнопку «Записаться» на конференции, на которую уже зарегистрировался.
- 3.b) Приложение отображает уведомление: “Вы уже зарегистрированы на эту конференцию”.

2.7.4 Альтернативные сценарии

1. Отмена записи на конференцию

- 1.a) Участник желает отменить свою регистрацию на конференцию.
- 1.b) Участник возвращается на экран конференции и нажимает «Отменить запись».
- 1.c) Приложение отображает подтверждение отмены: “Ваша запись отменена”, и обновляет статус конференции.

2. Фильтрация списка конференций

- 2.a) Участник хочет увидеть только конференции, проводимые в определенные даты.
- 2.b) Участник задает фильтр по дате через соответствующий интерфейс.
- 2.c) Приложение обновляет и показывает отфильтрованный список конференций.

2.7.5 Функциональные требования

Таблица 2.2 - Функциональные требования

Функциональное требование	Шаги и описание
ПросмотрСпискаКонференций	Участник открывает приложение и видит список доступных конференций.
ПросмотрИнформацииОконференции	Участник нажимает на конференцию из списка.
.ПросмотрДеталей	Приложение отображает подробную информацию о конференции: название, дата, время, местоположение и описание.
ЗаписьНаКонференцию	Участник нажимает кнопку «Записаться» на странице с информацией о конференции.
.ПодтверждениеЗаписи	Приложение подтверждает успешную запись и обновляет статус конференции для участника.
ОтменаЗаписиНаКонференцию	Участник возвращается на страницу конференции и нажимает «Отменить запись».
.ПодтверждениеОтмены	Приложение подтверждает отмену и обновляет статус конференции для участника.
ФильтрацияСпискаКонференций	Участник выбирает фильтр по дате через интерфейс.
.ОбновлениеСписка	Приложение обновляет и показывает отфильтрованный список конференций.

Функциональное требование	Шаги и описание
ОшибкаЗагрузкиСписка	При возникновении ошибки, приложение отображает сообщение об ошибке и предлагает повторить через некоторое время.
ОшибкаЗаписиНаКонференцию	Если запись не удалась, приложение отображает сообщение об ошибке и предлагает попробовать повторить операцию позже.
ПроверкаПовторнойЗаписи	Участник пытается повторно записаться на конференцию, на которую уже зарегистрировался.
.ОтображениеПредупреждения	Приложение отображает уведомление: “Вы уже зарегистрированы на эту конференцию”.

2.8 Модель базы данных

Таблица 2.3 - Словарь данных

Таблица	Поле	Тип данных	Описание
Users	username	VARCHAR(255)	Уникальное имя пользо- вателя
	password_hash	VARCHAR(255)	Хэш пароля пользо- вателя
	token	VARCHAR(255)	Токен для сессии пользо- вателя

Таблица	Поле	Тип данных	Описание
Cities	id	INT	Уникальный идентификатор города
	name	VARCHAR(255)	Название города
Topics	id	INT	Уникальный идентификатор темы
	name	VARCHAR(255)	Название темы
Conf	id	INT	Уникальный идентификатор конференции
	name	VARCHAR(255)	Название конференции
	city_id	INT	Ссылка на город
	topic_id	INT	Ссылка на тему
	author_username	VARCHAR(255)	Имя пользователя автора
	start_date	DATE	Дата начала конференции
	end_date	DATE	Дата окончания конференции

Таблица	Поле	Тип данных	Описание
	description	TEXT	Описание конференции
	contacts	TEXT	Контактная информация
	image_url	VARCHAR(255)	URL изображения

2.8.1 Процедуры

Процедура RegisterNewUser: Использует входные параметры new_username, new_password_hash и new_token для регистрации нового пользователя в системе конференций. Процедура сначала проверяет, существует ли уже пользователь с указанным именем пользователя, извлекая текущий токен из таблицы Users. Если пользователь уже зарегистрирован, процедура возвращает существующий токен в выходной параметр resulting_token, указывая на то, что пользователь уже существует. В случае, если пользователь с таким именем не найден, процедура добавляет новую запись в таблицу Users с предоставленными именем пользователя, хешем пароля и токеном. После успешной регистрации возвращает новый токен через выходной параметр resulting_token, подтверждая успешное создание учетной записи и регистрацию пользователя.

2.8.2 Триггеры

Триггер after_conference_insert: автоматически выполняется после добавления новой записи в таблицу Conferences. Этот триггер предназначен для автоматического добавления автора конференции в список участников этой же конференции. После успешного выполнения INSERT-запроса в таблицу Conferences, триггер вставляет новую запись в таблицу ConferenceAttendance, используя имя пользователя автора конференции (NEW.author_username) и идентификатор конференции (NEW.id).

2.8.3 Индексы

1. **Индекс `idx_users_token`:** Создан для оптимизации поиска по столбцу `token` в таблице `Users`, что ускоряет операции проверки аутентификации.
2. **Индекс `idx_conferences_city`:** Улучшает производительность запросов по столбцу `city_id` в таблице `Conferences`, позволяя быстрее фильтровать конференции по местоположению.
3. **Индекс `idx_conferences_topic`:** Ускоряет операции поиска и фильтрации по столбцу `topic_id` в таблице `Conferences`, упрощая работу с темами конференций.
4. **Составной индекс `idx_users_username_email`:** Обеспечивает быстрое выполнение запросов, которые фильтруют данные по `username` и `email` одновременно в таблице `Users`.
5. **Составной индекс `idx_conferences_date_location`:** Улучшает производительность запросов, фильтрующих конференции по `event_date` и `city_id`, оптимизируя сортировку по дате и расположению в таблице `Conferences`.

2.9 Описание работы интерфейса

На этом экране пользователь видит форму для входа в систему. Она содержит поля для ввода имени пользователя и пароля, а также кнопку для подтверждения входа. Этот интерфейс обеспечивает безопасную авторизацию, после которой пользователи получают доступ к функциональности приложения, включая просмотр и управление конференциями.

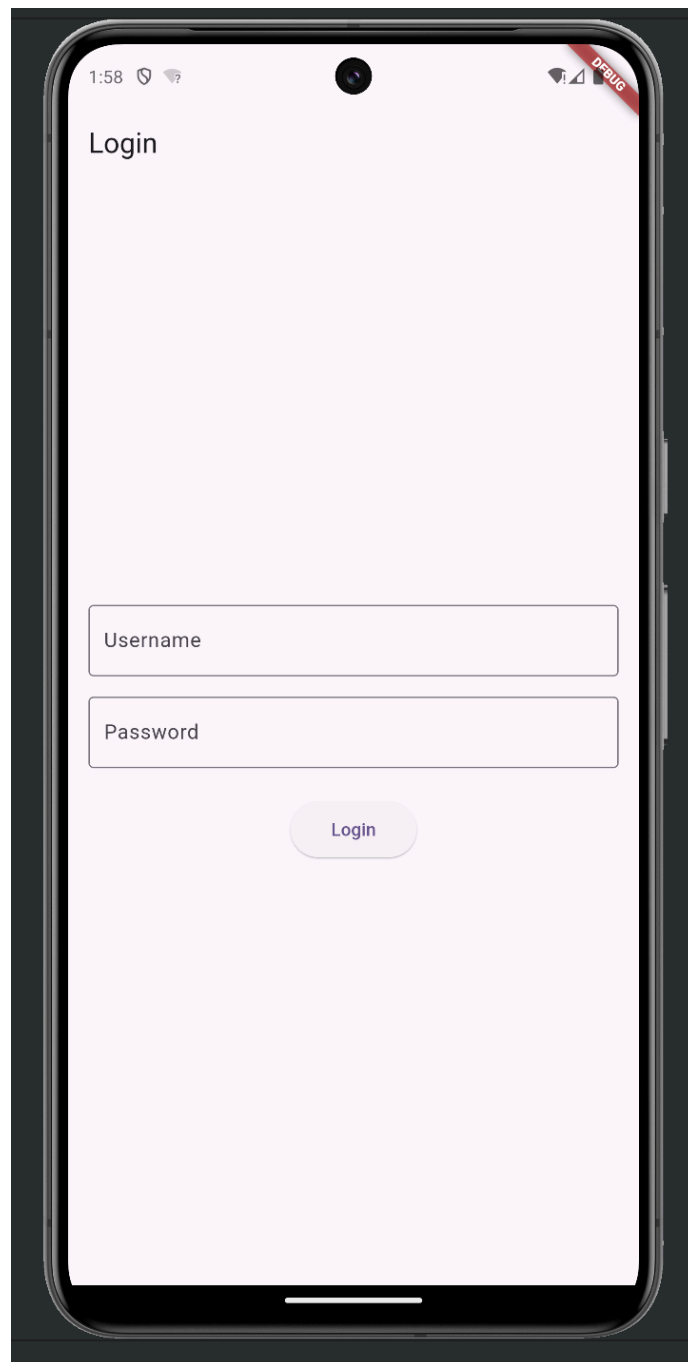


Рисунок 2.1 - Форма логина

Этот экран представляет собой основной список всех доступных конференций, доступных для пользователя. Здесь можно увидеть основную информацию по каждой конференции, включая название, дату и тему. Экран также предоставляет возможность быстрого доступа к детальной информации о каждой конференции посредством простого нажатия на соответствующую строку в списке.

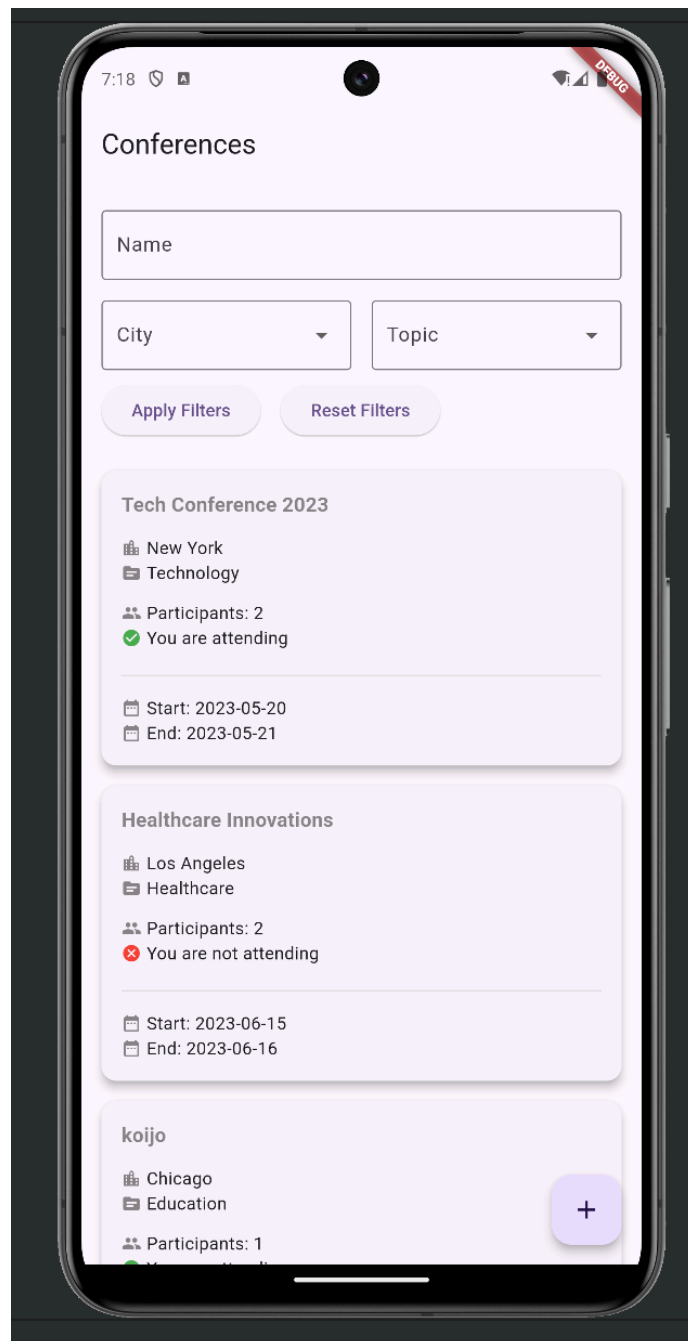


Рисунок 2.2 - Список конференций

На данном экране пользователи видят полную информацию о выбранной конференции. Блоки данных содержат детали, такие как даты проведения, местоположение, описание и контакты организаторов. Интерфейс также предлагает операции редактирования и удаления конференции, если они разрешены для текущего пользователя.

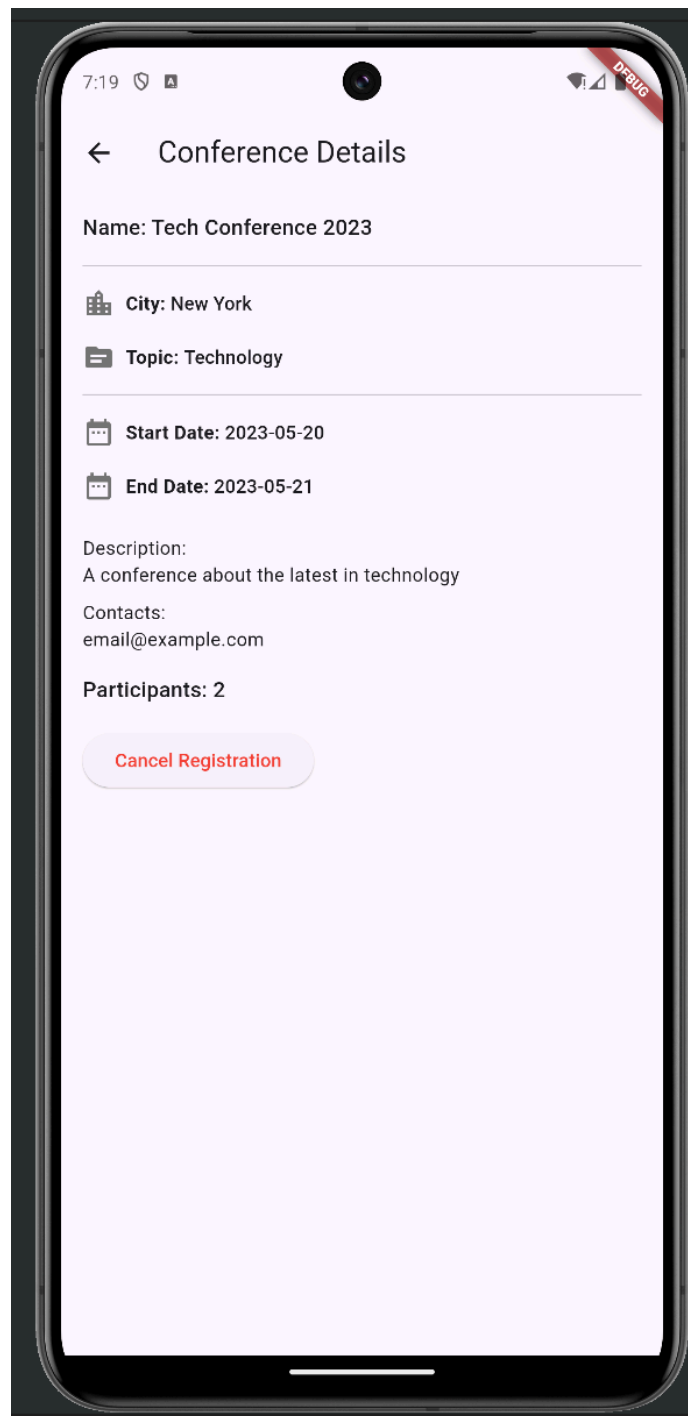


Рисунок 2.3 - Информация о конференции

Этот экран демонстрирует окно настройки фильтрации списка конференций. Пользователь может выбрать параметры фильтрации, такие как даты, тема или город, чтобы сузить отображаемый список конференций. Этот функционал помогает пользователям быстро находить и получать доступ к наиболее релевантным для них конференциям.

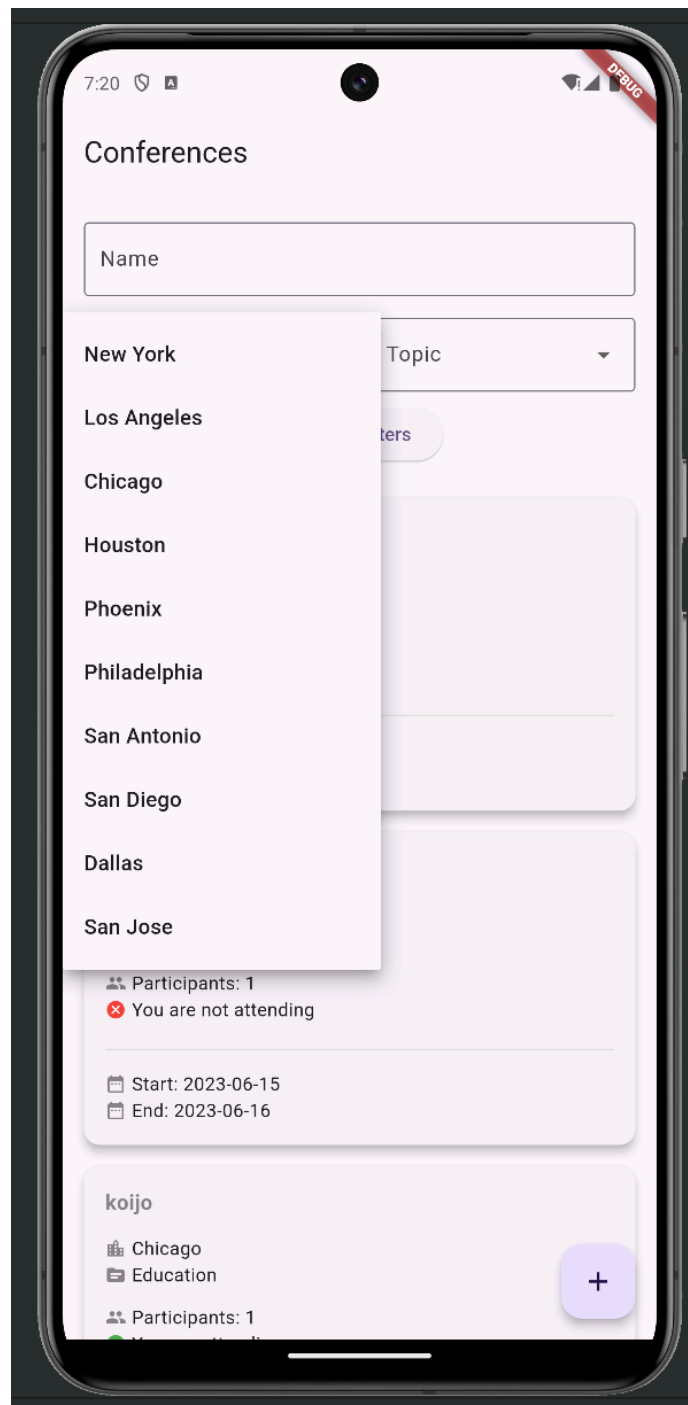


Рисунок 2.4 - Настройка фильтрации

После настройки фильтров пользователь возвращается на основной экран со списком конференций, где теперь отображаются только те конференции, которые соответствуют установленным критериям. Это подчеркивает удобство использования и персонализацию отображаемого контента.

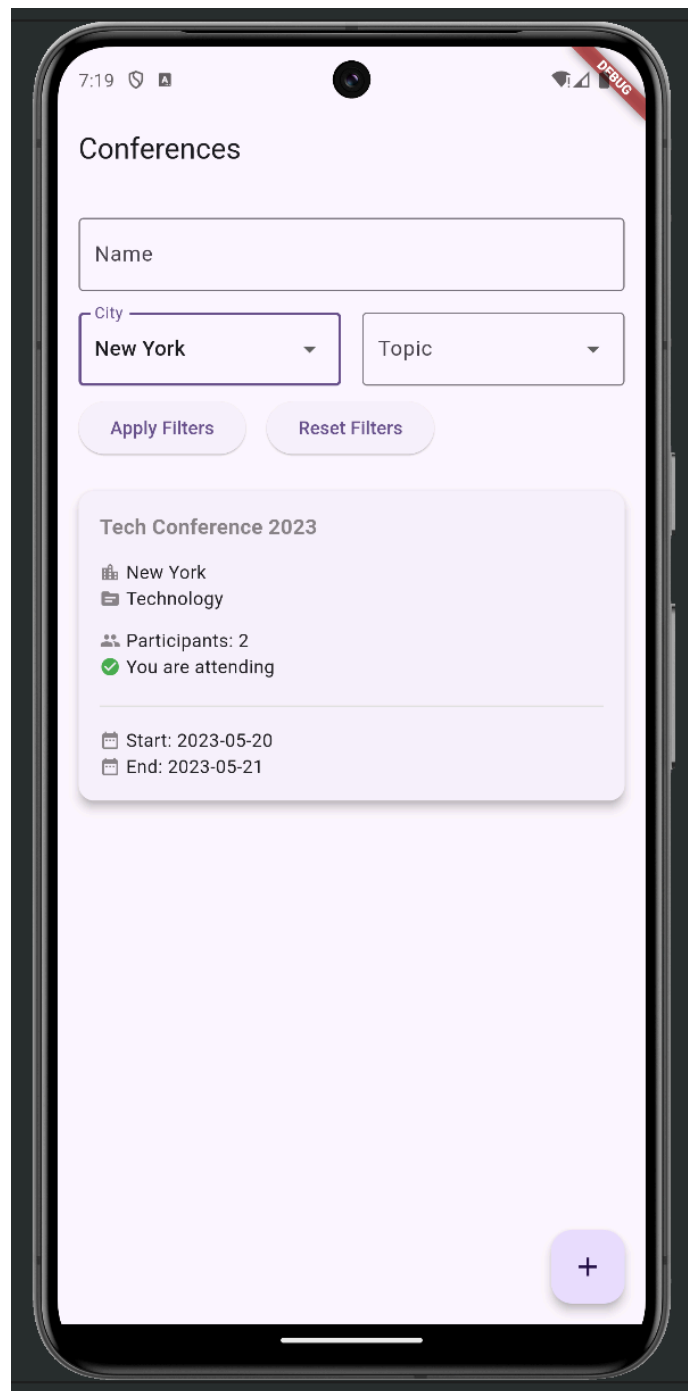


Рисунок 2.5 - Применение фильтрации

Этот скриншот показывает диалоговое окно, появляющееся в ситуации, когда происходят ошибки, например, при авторизации или подключении к базе данных. Диалог содержит сообщение об ошибке и инструкции по устранению проблемы, а также кнопку для закрытия окна. Такое окно позволяет пользователям быть в курсе возможных проблем и предпринимать соответствующие действия для их решения.

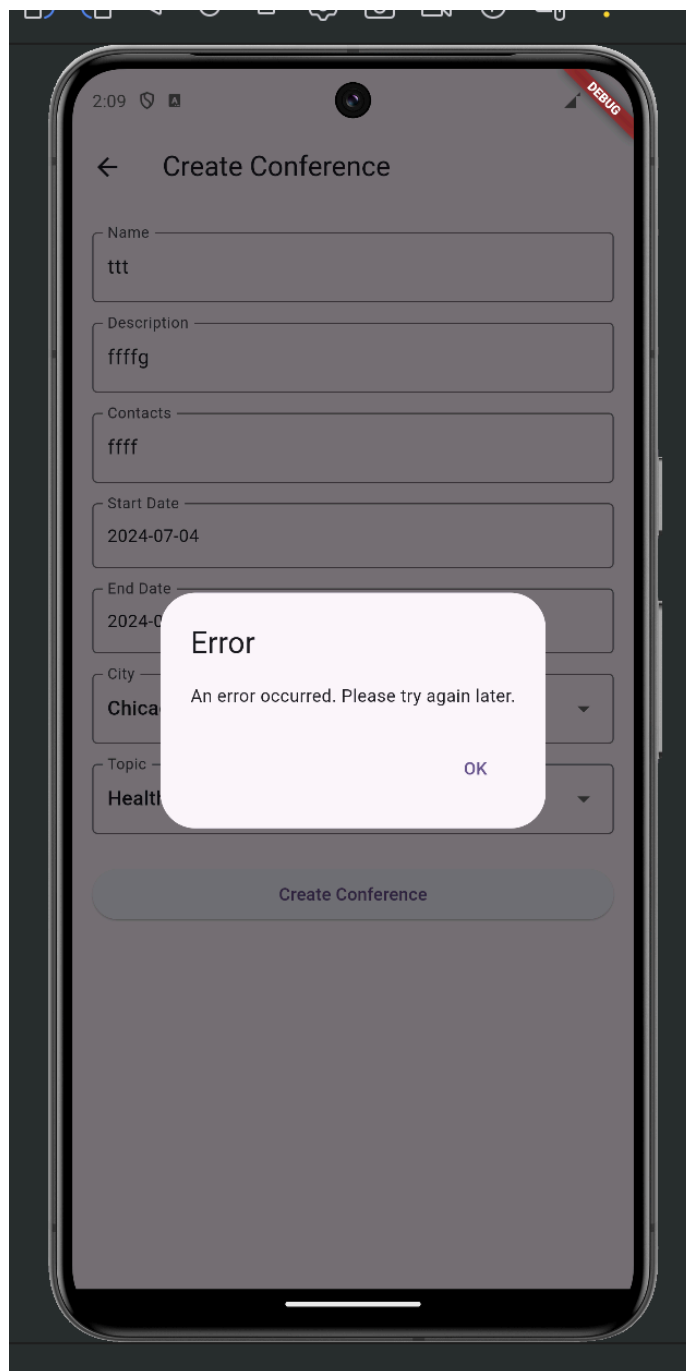


Рисунок 2.6 - Диалог об ошибке

Экран создания конференции разработан с целью облегчения и упрощения процесса планирования новых мероприятий. Пользователь может ввести основную информацию о конференции, включая название, описание, дату и время проведения, а также место. Такой интерфейс способствует быстрой и удобной настройке параметров мероприятия, обеспечивая интуитивно понятный процесс, который минимизирует время на создание событий и снижает вероятность ошибок.

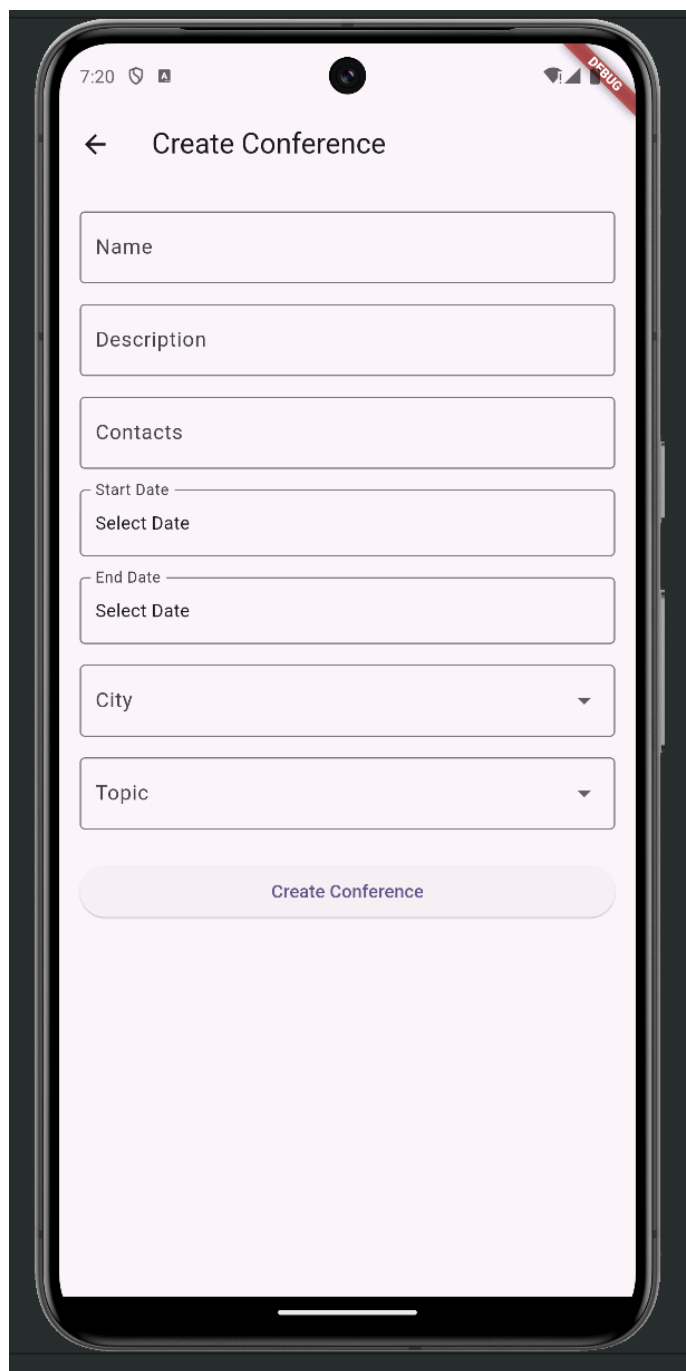


Рисунок 2.7 - Экран создания конференции

Приложение также поддерживает веб-версию, что позволяет пользователям взаимодействовать с системой через браузер. Это обеспечивает большую гибкость, независимо от используемой операционной системы или устройства. Такая функциональность предоставляет участникам и организаторам мероприятий возможность легко получать доступ к данным, управлять регистрациями и обмениваться информацией, находясь в любом месте с подключением к интернету.

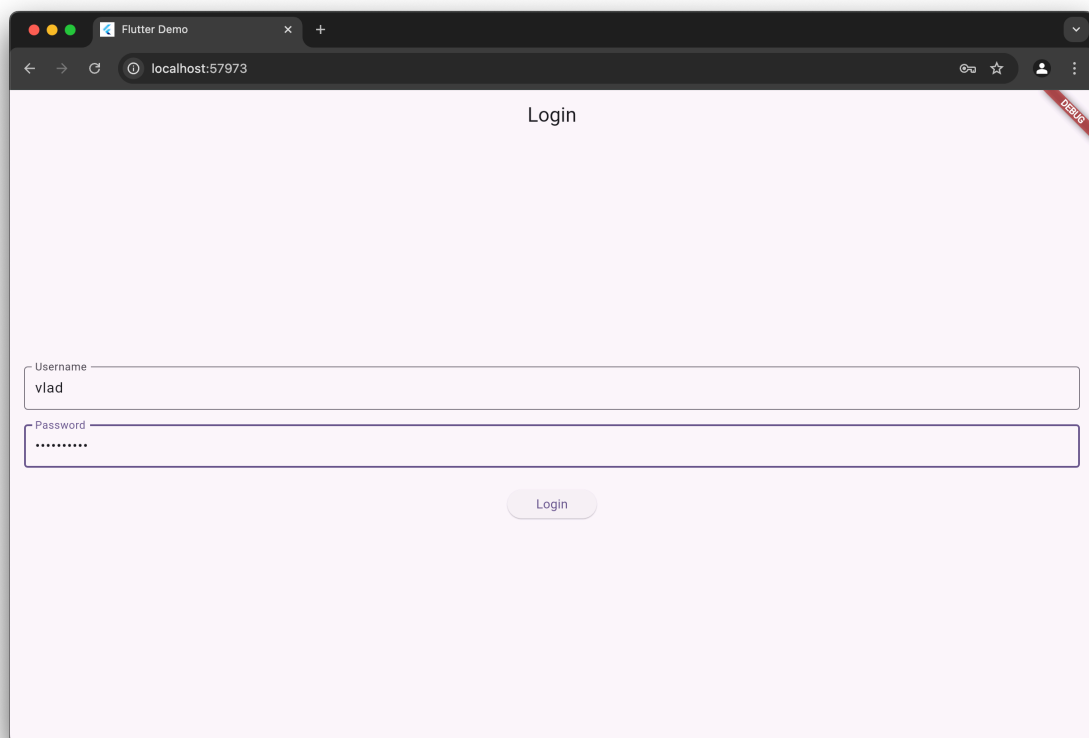


Рисунок 2.8 - Экран логина в веб-версии

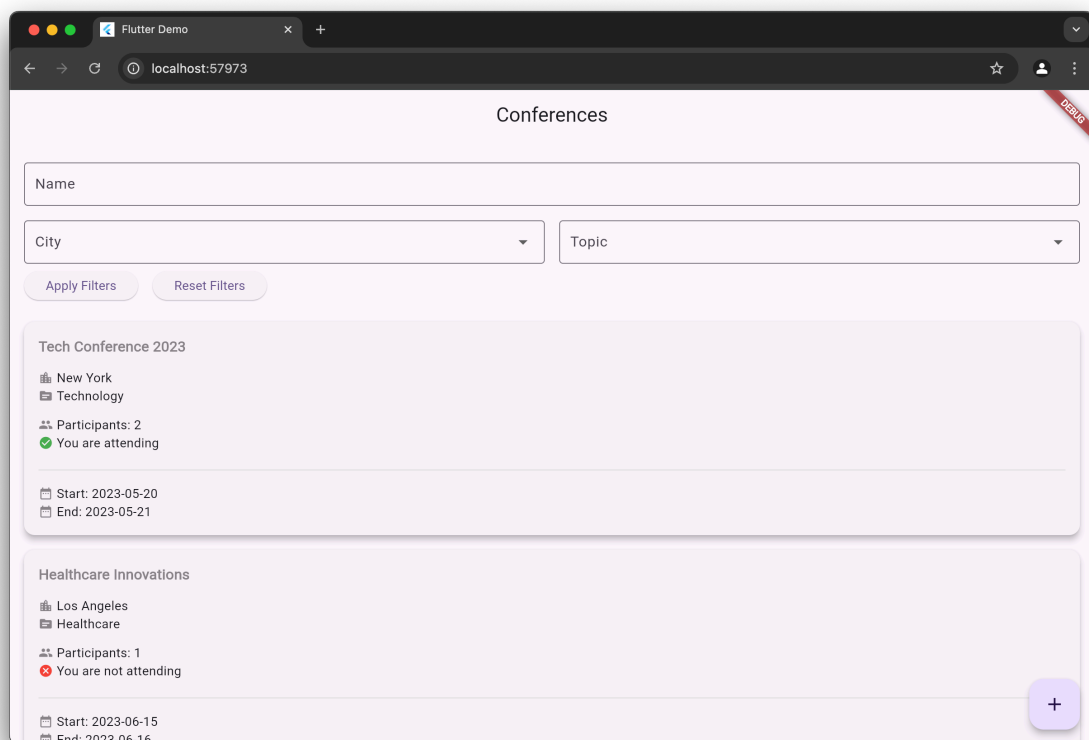


Рисунок 2.9 - Список конференций в веб-версии

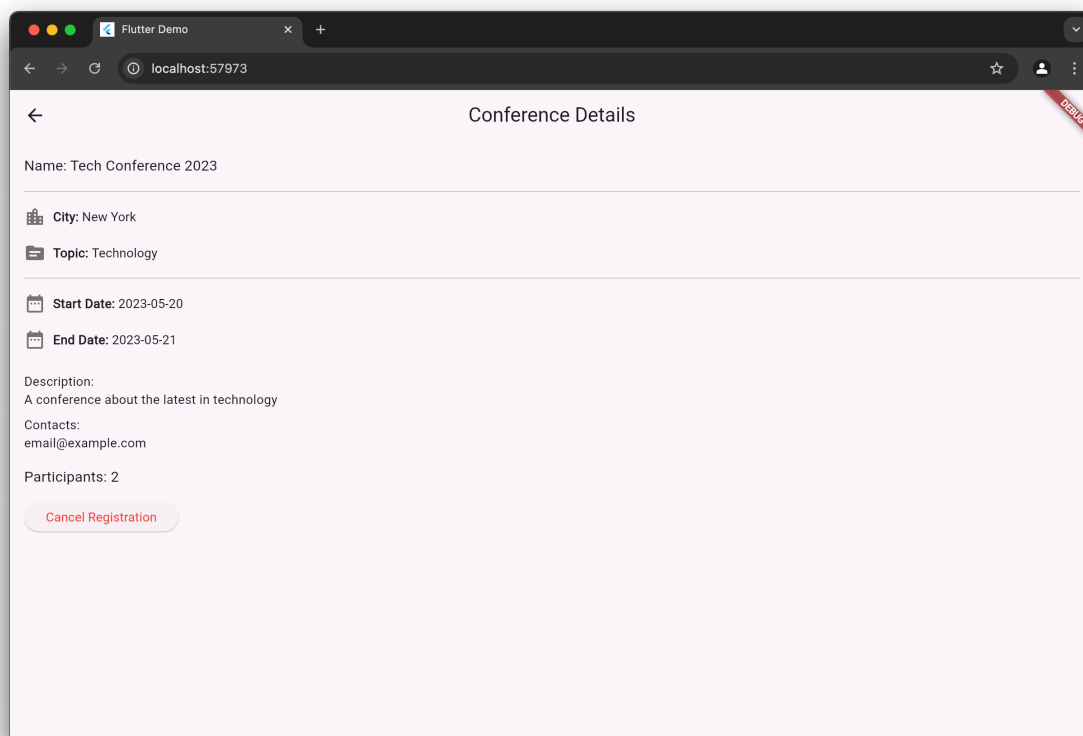


Рисунок 2.10 - Информация о конференции в веб-версии

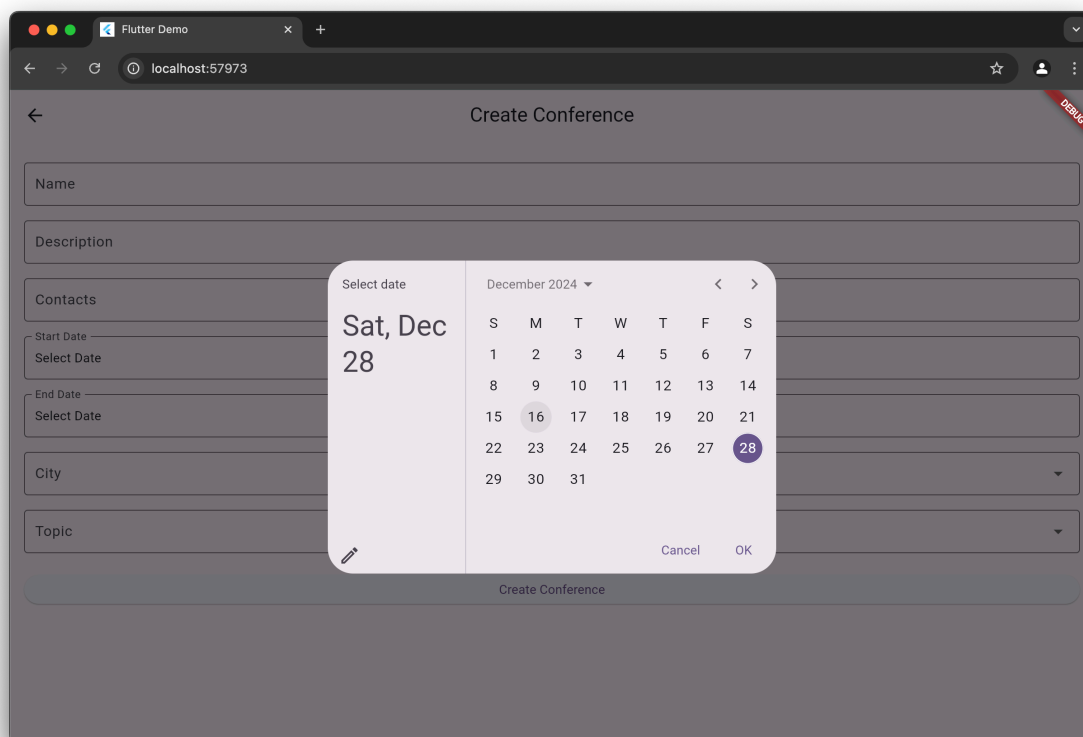


Рисунок 2.11 - Экран создания конференции в веб-версии

ЗАКЛЮЧЕНИЕ

В процессе курсовой работы была разработана автоматизированная информационная система для управления конференциями, используя фреймворки Ktor и Flutter. Основной целью проекта стало создание современного и функционального программного продукта, который обеспечивает удобство не только в организации мероприятий, но и в регистрации участников на них. Роли пользователей были четко определены, а также проработаны функциональные и технические требования системы. Использование Ktor для серверной части и Flutter для разработки кросс-платформенных мобильных приложений позволило создать надежную и эффективную систему, готовую к практическому применению и обеспечивающую поиск и фильтрацию конференций.

Итоговый код доступен в Приложении и на GitHub (URI - https://github.com/vladcto/suai-labs/tree/main/7_semester/МСПИСИТ/34).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Грегори Т. Чистая архитектура с Flutter: Подход, готовый к производству / Т. Грегори, Издано самостоятельно, 2020.
2. Мюллер Л. Производительность кроссплатформенной мобильной разработки: Xamarin против Flutter // Международный журнал инженерии программного обеспечения и знаний. 2021. № 2 (31). С. 345–360.
3. Нурбеков А. Ktor для серверного Kotlin: Разработка асинхронных микросервисов с использованием Kotlin / А. Нурбеков, Нью-Йорк: Apress, 2021.
4. Фриман А. Flutter для начинающих: Введение в разработку кроссплатформенных мобильных приложений с использованием Flutter и Dart 2 / А. Фриман, Бирмингем: Packt Publishing, 2019.

ПРИЛОЖЕНИЕ

```
1 Application.kt
2 package suai.vladcto
3
4 import io.ktor.serialization.kotlinx.json.json
5 import io.ktor.server.application.Application
6 import io.ktor.server.application.install
7 import io.ktor.server.engine.embeddedServer
8 import io.ktor.server.netty.Netty
9 import io.ktor.server.plugins.contentnegotiation.
    ContentNegotiation
10 import kotlinx.serialization.json.Json
11 import suai.vladcto.routing.configureRouting
12
13 fun main(args: Array<String>) {
14     embeddedServer(
15         Netty,
16         port = 8080,
17         host = "127.0.0.1",
18         module = Application::module
19     ).start(wait = true)
20 }
21
22 fun Application.module() {
23     install(ContentNegotiation) {
24         json(
25             Json {
26                 ignoreUnknownKeys = true
27                 isLenient = true
28                 prettyPrint = true
29             }
30         )
31     }
32     configureRouting()
33 }
34
35 DatabaseFactory.kt
36 package suai.vladcto.db
37
38 import java.sql.Connection
39 import java.sql.DriverManager
```



```

40
41 object DatabaseFactory {
42     private var connection: Connection? = null
43
44     private fun connect() {
45         connection = DriverManager.getConnection(
46             "jdbc:mysql://localhost:3306/conference",
47             "root",
48             ""
49         )
50     }
51
52     fun getConnection(): Connection {
53         val connection = this.connection
54         if (connection == null || connection.isClosed) {
55             connect()
56         }
57         return this.connection!!
58     }
59 }
60 Routing.kt
61 package suai.vladcto.routing
62
63 import io.ktor.http.HttpStatusCode
64 import io.ktor.server.application.Application
65 import io.ktor.server.request.receive
66 import io.ktor.server.response.respond
67 import io.ktor.server.routing.get
68 import io.ktor.server.routing.post
69 import io.ktor.server.routing.routing
70 import suai.vladcto.services.auth.AuthService
71 import suai.vladcto.services.city.CityService
72 import suai.vladcto.services.conference.ConferenceRequest
73 import suai.vladcto.services.conference.ConferenceService
74 import suai.vladcto.services.conference.
75     ConferenceUpdateRequest
76 import suai.vladcto.services.topic.TopicService
77
78 fun Application.configureRouting() {
79     val authService = AuthService()
80     val topicService = TopicService()

```

```

80     val cityService = CityService()
81     val conferenceService = ConferenceService(
82         topicService,
83         cityService,
84     )
85
86     routing {
87         post("/authorize") {
88             val params = call.request.queryParameters
89             val username = params["name"]
90             val password = params["password"]
91             if (username.isNullOrBlank() || password.
92                 isNullOrBlank()) {
93                 call.respond(HttpStatusCode.BadRequest, "
94                     Invalid user data")
95                 return@post
96             }
97
98             val result = authService.registerUser(username,
99                 password)
100
101             result
102                 .onSuccess { token ->
103                     call.respond(HttpStatusCode.Created,
104                         mapOf("token" to token))
105                 }
106                 .onFailure { error ->
107                     call.respond(HttpStatusCode.Conflict,
108                         error.message ?: "Registration failed
109                             ")
110                 }
111
112             }
113
114         get("/conferences_search_options") {
115             val searchOptions = conferenceService.
116                 getSearchOptions()
117             call.respond(HttpStatusCode.OK, searchOptions)
118         }
119
120         get("/conferences_list") {
121             val name = call.request.queryParameters["name"]

```

```

114         val cityId = call.request.queryParameters["cityId"]?.toIntOrNull()
115         val topicId = call.request.queryParameters["topicId"]?.toIntOrNull()
116         val startDate = call.request.queryParameters["startDate"]
117         val endDate = call.request.queryParameters["endDate"]
118         val description = call.request.queryParameters["description"]
119         val contacts = call.request.queryParameters["contacts"]
120         val imageUrl = call.request.queryParameters["imageUrl"]
121
122         val filterRequest = ConferenceRequest(
123             name = name ?: "",
124             cityId = cityId ?: 0,
125             topicId = topicId ?: 0,
126             startDate = startDate ?: "",
127             endDate = endDate ?: "",
128             description = description ?: "",
129             contacts = contacts ?: "",
130             imageUrl = imageUrl ?: ""
131         )
132
133         val conferences = conferenceService.
            getConferences(filterRequest)
134
135         call.respond(HttpStatusCode.OK, conferences)
136     }
137
138     postAuthorized(authService, "/create_conference") {
139         username ->
140         val conferenceRequest = call.receive<
141             ConferenceRequest>()
142
143         if (conferenceRequest.name.isBlank() ||
144             conferenceRequest.cityId <= 0 ||
145             conferenceRequest.topicId <= 0 ||
146             conferenceRequest.startDate.isBlank() ||

```

```

conferenceRequest.endDate.isBlank()
143     ) {
144         call.respond(HttpStatusCode.BadRequest, "
            Missing or invalid conference data")
145         return@postAuthorized
146     }
147
148     val result = conferenceService.createConference(
149         name = conferenceRequest.name,
150         cityId = conferenceRequest.cityId,
151         topicId = conferenceRequest.topicId,
152         authorUsername = username,
153         startDate = conferenceRequest.startDate,
154         endDate = conferenceRequest.endDate,
155         description = conferenceRequest.description,
156         contacts = conferenceRequest.contacts,
157         imageUrl = conferenceRequest.imageUrl,
158     )
159
160     result.onSuccess {
161         call.respond(
162             HttpStatusCode.Created,
163             "Conference created successfully with ID:
                $it"
164         )
165     }.onFailure { error ->
166         call.respond(
167             HttpStatusCode.Conflict,
168             error.message ?: "Error creating
                conference"
169         )
170     }
171 }
172
173 getAuthorized(authService, "/conference/{id}") {
    username ->
174     val conferenceId = call.parameters["id"]?.
        toIntOrNull()
175     if (conferenceId == null) {
176         call.respond(HttpStatusCode.BadRequest, "
            Invalid conference ID")

```

```

177         return@getAuthorized
178     }
179
180     val result = conferenceService.
        getConferenceDetails(conferenceId, username)
181
182     result.onSuccess { conferenceDetails ->
183         call.respond(HttpStatusCode.OK,
            conferenceDetails)
184     }.onFailure { error ->
185         call.respond(
186             HttpStatusCode.NotFound,
187             error.message ?: "Conference not found"
188         )
189     }
190 }
191
192 putAuthorized(authService, "/conference/{id}") {
    username ->
193     val conferenceId = call.parameters["id"]?.
        toIntOrNull()
194     if (conferenceId == null) {
195         call.respond(HttpStatusCode.BadRequest, "
            Invalid conference ID")
196         return@putAuthorized
197     }
198
199     val updateRequest = call.receive<
        ConferenceUpdateRequest>()
200
201     val result = conferenceService.updateConference(
202         conferenceId = conferenceId,
203         username = username,
204         name = updateRequest.name,
205         cityId = updateRequest.cityId,
206         topicId = updateRequest.topicId,
207         startDate = updateRequest.startDate,
208         endDate = updateRequest.endDate,
209         description = updateRequest.description,
210         contacts = updateRequest.contacts,
211         imageUrl = updateRequest.imageUrl,

```

```

212         )
213
214         result.onSuccess {
215             call.respond(HttpStatusCode.OK, "Conference
                updated successfully")
216         }.onFailure { error ->
217             call.respond(
218                 HttpStatusCode.Conflict ,
219                 error.message ?: "Error updating
                    conference"
220             )
221         }
222     }
223
224     postAuthorized(authService , "/delete_conference/{id
        }") { username ->
225         val conferenceId = call.parameters["id"]?.
            toIntOrNull()
226         if (conferenceId == null) {
227             call.respond(HttpStatusCode.BadRequest , "
                Invalid conference ID")
228             return@postAuthorized
229         }
230
231         val result = conferenceService.deleteConference(
            conferenceId , username)
232
233         result.onSuccess {
234             call.respond(HttpStatusCode.OK, "Conference
                deleted successfully")
235         }.onFailure { error ->
236             call.respond(
237                 HttpStatusCode.Conflict ,
238                 error.message ?: "Error deleting
                    conference"
239             )
240         }
241     }
242 }
243 }
244

```

```

245 AuthorizedRouting.kt
246 package suai.vladcto.routing
247
248 import io.ktor.http.HttpStatusCode
249 import io.ktor.server.application.ApplicationCall
250 import io.ktor.server.response.respond
251 import io.ktor.server.routing.Route
252 import io.ktor.server.routing.RoutingContext
253 import io.ktor.server.routing.get
254 import io.ktor.server.routing.post
255 import io.ktor.server.routing.put
256 import suai.vladcto.services.auth.AuthService
257
258 private suspend fun authorizeAndHandle(
259     call: ApplicationCall, authService: AuthService, handler:
260         suspend (String) -> Unit
261 ) {
262     val username = authService.verifyToken(call.request.
263         headers["Authorization"])
264     if (username == null) {
265         call.respond(HttpStatusCode.Unauthorized, "Invalid
266             token")
267     } else {
268         handler(username)
269     }
270 }
271
272 fun Route.getAuthorized(
273     authService: AuthService,
274     path: String,
275     handler: suspend RoutingContext.(String) -> Unit
276 ) {
277     get(path) {
278         authorizeAndHandle(call, authService) { username ->
279             handler(username)
280         }
281     }
282 }
283
284 fun Route.postAuthorized(

```

```

283     authService: AuthService ,
284     path: String ,
285     handler: suspend RoutingContext.(String) -> Unit
286 ) {
287     post(path) {
288         authorizeAndHandle(call , authService) { username ->
289             handler(username)
290         }
291     }
292 }
293
294 fun Route.putAuthorized(
295     authService: AuthService ,
296     path: String ,
297     handler: suspend RoutingContext.(String) -> Unit
298 ) {
299     put(path) {
300         authorizeAndHandle(call , authService) { username ->
301             handler(username)
302         }
303     }
304 }
305
306 CityService.kt
307 package suai.vladcto.services.city
308
309 import suai.vladcto.db.DatabaseFactory
310 import java.sql.SQLException
311
312 class CityService {
313
314     fun getCityNameById(cityId: Int): String? {
315         val query = "SELECT name FROM Cities WHERE id = ?"
316
317         return try {
318             val connection = DatabaseFactory.getConnection()
319
320             connection.prepareStatement(query).use {
321                 preparedStatement ->
322                 preparedStatement.setInt(1, cityId)
323                 val resultSet = preparedStatement.

```



```

        executeQuery()
323
324        if (resultSet.next()) {
325            resultSet.getString("name")
326        } else {
327            null
328        }
329    }
330    } catch (e: SQLException) {
331        null
332    }
333 }
334
335 fun getAllCities(): List<City> {
336     try {
337         val query = "SELECT id, name FROM Cities"
338         val cities = mutableListOf<City>()
339         val connection = DatabaseFactory.getConnection()
340
341         connection.createStatement().use { statement ->
342             val resultSet = statement.executeQuery(query)
343
344             while (resultSet.next()) {
345                 val id = resultSet.getInt("id")
346                 val name = resultSet.getString("name")
347                 cities.add(City(id, name))
348             }
349         }
350         return cities
351     } catch (e: SQLException) {
352         return listOf()
353     }
354 }
355 }
356
357 City.kt
358 package suai.vladcto.services.city
359
360 import kotlinx.serialization.Serializable
361
362 @Serializable

```

```

363 data class City(val id: Int, val name: String)
364 ConferenceUpdateRequest.kt
365 package suai.vladcto.services.conference
366
367 import kotlinx.serialization.Serializable
368
369
370 @Serializable
371 data class ConferenceUpdateRequest(
372     val name: String?,
373     val cityId: Int?,
374     val topicId: Int?,
375     val startDate: String?,
376     val endDate: String?,
377     val description: String?,
378     val contacts: String?,
379     val imageUrl: String?,
380 )
381
382 ConferenceResponse.kt
383 package suai.vladcto.services.conference
384
385 import kotlinx.serialization.Serializable
386
387 @Serializable
388 data class ConferenceShortResponse (
389     val id: Int,
390     val name: String,
391     val city: String,
392     val topic: String,
393     val startDate: String,
394     val endDate: String,
395 )
396
397 @Serializable
398 data class ConferenceDetailResponse (
399     val id: Int,
400     val name: String,
401     val city: String,
402     val topic: String,
403     val startDate: String,

```

```

404     val endDate: String ,
405     val description: String ,
406     val contacts: String ,
407     val canRedact: Boolean ,
408 )
409 ConferenceService.kt
410 package suai.vladcto.services.conference
411
412 import suai.vladcto.db.DatabaseFactory
413 import suai.vladcto.services.city.CityService
414 import suai.vladcto.services.topic.TopicService
415 import java.sql.ResultSet
416 import java.sql.SQLException
417
418 class ConferenceService(
419     private val topicService: TopicService ,
420     private val cityService: CityService ,
421 ) {
422     fun createConference(
423         name: String ,
424         cityId: Int ,
425         topicId: Int ,
426         authorUsername: String ,
427         startDate: String ,
428         endDate: String ,
429         description: String = "" ,
430         contacts: String = "" ,
431         imageUrl: String = ""
432     ): Result<Int> {
433         return try {
434             val connection = DatabaseFactory.getConnection()
435             val sql = """
436                 INSERT INTO Conferences (
437                     name, city_id , topic_id , author_username ,
438                     start_date , end_date , description ,
439                     contacts , image_url
440
441                 ) VALUES (
442                     '$name' , $cityId , $topicId , '
443                     $authorUsername ' , '$startDate ' , '
444                     $endDate ' , '$description ' , '$contacts
445                     ' , '$imageUrl '

```

```

440         )
441         """
442
443         connection.createStatement().use { statement ->
444             statement.executeUpdate(sql, java.sql.
                Statement.RETURN_GENERATED_KEYS)
445
446             val generatedKeys = statement.generatedKeys
447             if (generatedKeys.next()) {
448                 Result.success(generatedKeys.getInt(1))
449             } else {
450                 Result.failure(SQLException("Failed to
                    retrieve conference ID"))
451             }
452         }
453     } catch (e: SQLException) {
454         Result.failure(e)
455     }
456 }
457
458
459 fun getConferences(filter: ConferenceRequest): List<
    ConferenceShortResponse> {
460     val queryBuilder = StringBuilder("SELECT * FROM
        Conferences WHERE 1=1")
461
462     if (filter.name.isNotEmpty()) {
463         queryBuilder.append(" AND name ILIKE '%${filter.
            name}%'")
464     }
465     if (filter.cityId != 0) {
466         queryBuilder.append(" AND city_id = ${filter.
            cityId}")
467     }
468     if (filter.topicId != 0) {
469         queryBuilder.append(" AND topic_id = ${filter.
            topicId}")
470     }
471     if (filter.startDate.isNotEmpty()) {
472         queryBuilder.append(" AND start_date >= '${filter
            .startDate}'")

```

```

473     }
474     if ( filter.endDate.isNotEmpty() ) {
475         queryBuilder.append(" AND end_date <= '${filter.
            endDate}'")
476     }
477
478     return try {
479         val connection = DatabaseFactory.getConnection()
480
481         connection.createStatement().use { statement ->
482             val resultSet = statement.executeQuery(
                queryBuilder.toString() )
483             val conferences = mutableListOf<
                ConferenceShortResponse>()
484
485             while ( resultSet.next() ) {
486                 conferences.add( mapRowToConference(
                    resultSet ) )
487             }
488
489             conferences
490         }
491     } catch ( e: SQLException ) {
492         emptyList()
493     }
494 }
495
496 fun getSearchOptions(): ConferenceSearchOptions {
497     val cities = cityService.getAllCities()
498     val topics = topicService.getAllTopics()
499
500     return ConferenceSearchOptions(
501         cities = cities ,
502         topics = topics ,
503     )
504 }
505
506 fun updateConference(
507     conferenceId: Int ,
508     username: String ,
509     name: String? = null ,

```

```

510         cityId: Int? = null ,
511         topicId: Int? = null ,
512         startDate: String? = null ,
513         endDate: String? = null ,
514         description: String? = null ,
515         contacts: String? = null ,
516         imageUrl: String? = null
517     ): Result<Boolean> {
518         return try {
519             if (!isUserAuthorOfConference(conferenceId ,
520                 username)) {
521                 return Result.failure( SecurityException("User
522                     $username is not the author of the
523                     conference.") )
524             }
525
526             val connection = DatabaseFactory.getConnection()
527             val updateQuery = StringBuilder("UPDATE
528                 Conferences SET ")
529
530             val updates = mutableListOf<String>()
531             name?.let { updates.add("name = '$it'") }
532             cityId?.let { updates.add("city_id = $it") }
533             topicId?.let { updates.add("topic_id = $it") }
534             startDate?.let { updates.add("start_date = '$it
535                 '") }
536             endDate?.let { updates.add("end_date = '$it'") }
537             description?.let { updates.add("description = '
538                 $it'") }
539             contacts?.let { updates.add("contacts = '$it'") }
540             imageUrl?.let { updates.add("image_url = '$it'") }
541
542             if (updates.isEmpty()) {
543                 return Result.failure(
544                     IllegalArgumentException("No fields to
545                     update") )
546             }
547
548             updateQuery.append(updates.joinToString(", "))
549             updateQuery.append(" WHERE id = $conferenceId")

```

```

542
543         // Debug: print the SQL query
544         println("Executing SQL: $updateQuery")
545
546         connection.createStatement().use { statement ->
547             val rowsAffected = statement.executeUpdate(
548                 updateQuery.toString())
549             if (rowsAffected > 0) {
550                 Result.success(true)
551             } else {
552                 Result.failure(SQLException("No
553                     conference found with id:
554                     $conferenceId"))
555             }
556         }
557     }
558
559     fun deleteConference(conferenceId: Int, username: String)
560       : Result<Boolean> {
561         return try {
562             if (!isUserAuthorOfConference(conferenceId,
563                 username)) {
564                 return Result.failure(SQLException("User is
565                     not the author of the conference"))
566             }
567
568             val connection = DatabaseFactory.getConnection()
569             val sql = "DELETE FROM Conferences WHERE id = ?"
570
571             connection.prepareStatement(sql).use {
572                 preparedStatement ->
573                 preparedStatement.setInt(1, conferenceId)
574                 val rowsAffected = preparedStatement.
575                     executeUpdate()
576
577                 if (rowsAffected > 0) {
578                     Result.success(true)
579                 } else {

```

```

575         Result.failure(SQLException("Could not
           delete conference with id:
           $conferenceId"))
576     }
577 }
578 } catch (e: SQLException) {
579     Result.failure(e)
580 }
581 }
582
583
584 fun getConferenceDetails(
585     conferenceId: Int, username: String
586 ): Result<ConferenceDetailResponse> {
587     return try {
588         val connection = DatabaseFactory.getConnection()
589
590         val sql = "SELECT * FROM Conferences WHERE id =
           ?"
591
592         connection.prepareStatement(sql).use {
593             preparedStatement ->
594                 preparedStatement.setInt(1, conferenceId)
595
596             val resultSet = preparedStatement.
597                 executeQuery()
598             if (resultSet.next()) {
599                 val conferenceDetail =
600                     ConferenceDetailResponse(
601                         id = resultSet.getInt("id"),
602                         name = resultSet.getString("name"),
603                         city = cityService.getCityNameById(
604                             resultSet.getInt("city_id")) ?:
605                             "",
606                         topic = topicService.getTopicNameById(
607                             resultSet.getInt("topic_id")) ?:
608                             "",
609                         startDate = resultSet.getString("
610                             start_date"),
611                         endDate = resultSet.getString("
612                             end_date"),

```



```

604         description = resultSet.getString("
        description"),
605         contacts = resultSet.getString("
        contacts"),
606         canRedact = isUserAuthorOfConference(
        conferenceId, username)
607     )
608     Result.success(conferenceDetail)
609 } else {
610     Result.failure(SQLException("No
        conference found with id:
        $conferenceId"))
611 }
612 }
613 } catch (e: SQLException) {
614     Result.failure(e)
615 }
616 }
617
618 private fun isUserAuthorOfConference(conferenceId: Int,
        username: String): Boolean {
619     return try {
620         val connection = DatabaseFactory.getConnection()
621         val sql = "SELECT author_username FROM
        Conferences WHERE id = ?"
622
623         connection.prepareStatement(sql).use {
        preparedStatement ->
624             preparedStatement.setInt(1, conferenceId)
625
626             val resultSet = preparedStatement.
        executeQuery()
627             if (resultSet.next()) {
628                 val authorUsername = resultSet.getString
        ("author_username")
629                 authorUsername == username
630             } else {
631                 false
632             }
633         }
634     } catch (e: SQLException) {

```

```

635         false
636     }
637 }
638
639 private fun mapRowToConference(rs: ResultSet):
        ConferenceShortResponse {
640     return ConferenceShortResponse(
641         id = rs.getInt("id"),
642         name = rs.getString("name"),
643         city = cityService.getCityNameById(rs.getInt("
            city_id")) ?: "",
644         topic = topicService.getTopicNameById(rs.getInt("
            topic_id")) ?: "",
645         startDate = rs.getString("start_date"),
646         endDate = rs.getString("end_date"),
647     )
648 }
649 }
650
651 ConferenceRequest.kt
652 package suai.vladcto.services.conference
653
654 import kotlinx.serialization.Serializable
655
656 @Serializable
657 data class ConferenceRequest(
658     val name: String,
659     val cityId: Int,
660     val topicId: Int,
661     val startDate: String,
662     val endDate: String,
663     val description: String,
664     val contacts: String,
665     val imageUrl: String,
666 )
667
668 ConferencSearchOptions.kt
669 package suai.vladcto.services.conference
670
671 import kotlinx.serialization.Serializable
672 import suai.vladcto.services.city.City

```

```

673 import suai.vladcto.services.topic.Topic
674
675 @Serializable
676 data class ConferenceSearchOptions(
677     val cities: List<City>,
678     val topics: List<Topic>
679 )
680 AuthService.kt
681 package suai.vladcto.services.auth
682
683 import suai.vladcto.db.DatabaseFactory
684 import java.security.MessageDigest
685 import java.sql.Connection
686 import java.util.UUID
687
688 class AuthService {
689
690     fun registerUser(username: String, password: String):
        Result<String> {
691         val connection = DatabaseFactory.getConnection()
692
693         val existingToken = getTokenIfUserExist(connection,
            username)
694         if (existingToken != null) {
695             return Result.success(existingToken)
696         }
697
698         val passwordHash = hashPassword(password)
699         val token = generateToken()
700         insertUser(connection, username, passwordHash, token)
701
702         return Result.success(token)
703     }
704
705     fun verifyToken(token: String?): String? {
706         if (token == null) return null
707
708         val connection = DatabaseFactory.getConnection()
709         val query = "SELECT username FROM Users WHERE token =
            ?"
710         connection.prepareStatement(query).use {

```

```

        preparedStatement ->
711         preparedStatement.setString(1, token)
712         val resultSet = preparedStatement.executeQuery()
713         return if (resultSet.next()) {
714             resultSet.getString("username")
715         } else {
716             null
717         }
718     }
719 }
720
721 private fun getTokenIfUserExist(connection: Connection,
722     username: String): String? {
723     val query = "SELECT token FROM Users WHERE username = ?"
724     connection.prepareStatement(query).use {
725         preparedStatement ->
726         preparedStatement.setString(1, username)
727         val resultSet = preparedStatement.executeQuery()
728         return if (resultSet.next()) {
729             resultSet.getString("token")
730         } else {
731             null
732         }
733     }
734 }
735
736 private fun insertUser(
737     connection: Connection, username: String,
738     passwordHash: String, token: String
739 ) {
740     val query = "INSERT INTO Users (username,
741         password_hash, token) VALUES (?, ?, ?)"
742     connection.prepareStatement(query).use {
743         preparedStatement ->
744         preparedStatement.setString(1, username)
745         preparedStatement.setString(2, passwordHash)
746         preparedStatement.setString(3, token)
747         preparedStatement.executeUpdate()
748     }
749 }

```

```

745
746     private fun generateToken(): String = UUID.randomUUID().
           toString()
747
748     private fun hashPassword(password: String): String =
749         MessageDigest.getInstance("SHA-256")
750             .digest(password.toByteArray())
751             .joinToString("") { "%02x".format(it) }
752 }
753
754 Topic.kt
755 package suai.vladcto.services.topic
756
757 import kotlinx.serialization.Serializable
758
759 @Serializable
760 data class Topic(val id: Int, val name: String)
761 TopicService.kt
762 package suai.vladcto.services.topic
763
764 import suai.vladcto.db.DatabaseFactory
765 import java.sql.SQLException
766
767 class TopicService {
768
769     fun getTopicNameById(topicId: Int): String? {
770         val query = "SELECT name FROM Topics WHERE id = ?"
771
772         return try {
773             val connection = DatabaseFactory.getConnection()
774
775             connection.prepareStatement(query).use {
776                 preparedStatement ->
777                     preparedStatement.setInt(1, topicId)
778                     val resultSet = preparedStatement.
779                         executeQuery()
780
781                     if (resultSet.next()) {
782                         resultSet.getString("name")
783                     } else {
784                         null
785                     }
786                 }
787             }
788         } catch (e: SQLException) {
789             null
790         }
791     }
792 }

```

```

783         }
784     }
785     } catch (e: SQLException) {
786         null
787     }
788 }
789
790 fun getAllTopics(): List<Topic> {
791     try {
792         val query = "SELECT id, name FROM Topics"
793         val topics = mutableListOf<Topic>()
794         val connection = DatabaseFactory.getConnection()
795         connection.createStatement().use { statement ->
796             val resultSet = statement.executeQuery(query)
797             while (resultSet.next()) {
798                 val id = resultSet.getInt("id")
799                 val name = resultSet.getString("name")
800                 topics.add(Topic(id, name))
801             }
802         }
803         return topics
804     } catch (e: SQLException) {
805         return listOf()
806     }
807 }
808 }
809
810
811 app_providers.dart
812 import 'package:conference_app/features/app/
      navigation_manager.dart';
813 import 'package:conference_app/features/di/auth_providers.
      dart';
814 import 'package:dio/dio.dart';
815 import 'package:flutter/cupertino.dart';
816 import 'package:flutter_riverpod/flutter_riverpod.dart';
817
818 sealed class AppProviders {
819     static final dioBaseOptions = Provider(
820         (_) => BaseOptions(baseUrl: 'http://10.0.2.2:8080'),
821     );

```

```

822
823     static final dio = Provider(
824         (ref) => Dio(
825             ref.watch(dioBaseOptions),
826         )..interceptors.add(
827             ref.watch(
828                 AuthProviders.interceptor,
829             ),
830         ),
831     );
832
833     static final navigatorKey = Provider(
834         (_) => GlobalKey<NavigatorState>(),
835     );
836
837     static final navigationManager = Provider(
838         (ref) => NavigationManager(ref.watch(navigatorKey)),
839     );
840 }
841
842 auth_providers.dart
843 import 'package:conference_app/features/auth/data/auth_api.
844     dart';
845 import 'package:conference_app/features/auth/logic/%20network
846     /auth_interceptor.dart';
847 import 'package:conference_app/features/auth/logic/manager/
848     auth_manager.dart';
849 import 'package:conference_app/features/auth/logic/state/
850     token_holder.dart';
851 import 'package:conference_app/features/di/app_providers.dart
852     ';
853 import 'package:dio/dio.dart';
854 import 'package:flutter_riverpod/flutter_riverpod.dart';
855
856 sealed class AuthProviders {
857     static final _tokenProvider = Provider(
858         (_) => TokenStateHolder(),
859     );
860
861     static final _dio = Provider(
862         (ref) => Dio(

```

```

858         ref.watch(AppProviders.dioBaseOptions),
859     ),
860 );
861
862     static final _api = Provider(
863         (ref) => AuthApi(
864             ref.watch(_dio),
865         ),
866     );
867
868     static final manager = Provider(
869         (ref) => AuthManager(
870             ref.watch(_api),
871             ref.watch(_tokenProvider),
872             ref.watch(AppProviders.navigationManager),
873         ),
874     );
875
876     static final interceptor = Provider(
877         (ref) => AuthInterceptor(
878             ref.watch(_tokenProvider),
879         ),
880     );
881 }
882
883 conference_providers.dart
884 import 'package:conference_app/features/conferences/data/
885     conference_api.dart';
886 import 'package:conference_app/features/conferences/logic/
887     manager/conference_detailed_manager.dart';
888 import 'package:conference_app/features/di/app_providers.dart';
889 import 'package:flutter_riverpod/flutter_riverpod.dart';
890 import 'package:conference_app/features/conferences/data/models/conference_search_options.dart';
891 import 'package:conference_app/features/conferences/data/models/conference_short_response.dart';
892 import 'package:conference_app/features/conferences/logic/manager/conference_list_manager.dart';
893 import 'package:conference_app/features/conferences/logic/state/conference_list_notifier.dart';

```



```

    dart';
893 import '../conferences/logic/state/
    conference_search_options_notifier.dart';
894
895 sealed class ConferenceProviders {
896     static final _api = Provider(
897         (ref) => ConferenceApi(
898             ref.watch(AppProviders.dio),
899         ),
900     );
901
902     static final searchOptionsProvider = StateNotifierProvider<
903         ConferenceSearchOptionsNotifier,
904         ConferenceSearchOptions?>(
905         (ref) => ConferenceSearchOptionsNotifier(),
906     );
907
908     static final listProvider = StateNotifierProvider<
909         ConferenceListNotifier,
910         List<ConferenceShortResponse>?>(
911         (ref) => ConferenceListNotifier(),
912     );
913
914     static final conferenceListManager = Provider(
915         (ref) => ConferenceListManager(
916             ref.watch(_api),
917             ref.watch(AppProviders.navigationManager),
918             ref.watch(searchOptionsProvider.notifier),
919             ref.watch(listProvider.notifier),
920         ),
921     );
922
923     static final conferenceDetailedManager = Provider(
924         (ref) => ConferenceDetailedManager(
925             ref.watch(_api),
926             ref.watch(AppProviders.navigationManager),
927             ref.watch(listProvider.notifier),
928         ),
929     );
930 }

```

```

930 navigation_manager.dart
931 import 'package:flutter/material.dart';
932
933 import '../conferences/ui/conference_detail_screen.dart';
934 import '../conferences/ui/conference_list.dart';
935 import '../conferences/ui/create_conference_screen.dart';
936
937 class NavigationManager {
938     final GlobalKey<NavigatorState> _navigatorKey;
939
940     NavigationManager(this._navigatorKey);
941
942     void pop() => _navigatorKey.currentState?.pop();
943
944     void openConferenceList() {
945         _navigatorKey.currentState?.pushReplacement(
946             MaterialPageRoute(
947                 builder: (context) => const ConferenceListScreen(),
948             ),
949         );
950     }
951
952     void openDetailedConference(int conferenceId) {
953         _navigatorKey.currentState?.push(
954             MaterialPageRoute(
955                 builder: (context) =>
956                     ConferenceDetailScreen(conferenceId: conferenceId),
957             ),
958         );
959     }
960
961     void openCreateConferenceScreen() {
962         _navigatorKey.currentState?.push(
963             MaterialPageRoute(builder: (context) => const
964                 CreateConferenceScreen()),
965         );
966     }
967
968     void showErrorDialog() {
969         showDialog(

```

```

969         context: _navigatorKey.currentContext!,
970         builder: (context) {
971             return AlertDialog(
972                 title: const Text('Error'),
973                 content: const Text('An error occurred. Please try
                    again later.'),
974                 actions: [
975                     TextButton(
976                         onPressed: () => Navigator.of(context).pop(),
977                         child: const Text('OK'),
978                     ),
979                 ],
980             );
981         },
982     );
983 }
984 }
985
986 auth_screen.dart
987 import 'package:conference_app/features/di/auth_providers.
    dart';
988 import 'package:flutter/material.dart';
989 import 'package:flutter_riverpod/flutter_riverpod.dart';
990
991 class LoginForm extends ConsumerStatefulWidget {
992     const LoginForm({super.key});
993
994     @override
995     ConsumerState<LoginForm> createState() => _LoginFormState();
996 }
997
998 class _LoginFormState extends ConsumerState<LoginForm> {
999     final TextEditingController _usernameController =
        TextEditingController();
1000     final TextEditingController _passwordController =
        TextEditingController();
1001
1002     @override
1003     Widget build(BuildContext context) {
1004         final authManager = ref.watch(AuthProviders.manager);

```

```

1005
1006     return Scaffold(
1007         appBar: AppBar(
1008             title: const Text('Login'),
1009         ),
1010         body: Center(
1011             child: Padding(
1012                 padding: const EdgeInsets.all(16.0),
1013                 child: Column(
1014                     mainAxisAlignment: MainAxisAlignment.min,
1015                     children: [
1016                         TextField(
1017                             controller: _usernameController,
1018                             decoration: const InputDecoration(
1019                                 labelText: 'Username',
1020                                 border: OutlineInputBorder(),
1021                             ),
1022                         ),
1023                         const SizedBox(height: 16),
1024                         TextField(
1025                             controller: _passwordController,
1026                             decoration: const InputDecoration(
1027                                 labelText: 'Password',
1028                                 border: OutlineInputBorder(),
1029                             ),
1030                             obscureText: true,
1031                         ),
1032                         const SizedBox(height: 24),
1033                         ElevatedButton(
1034                             onPressed: () => authManager.login(
1035                                 _usernameController.text.trim(),
1036                                 _passwordController.text.trim(),
1037                             ),
1038                             style: ElevatedButton.styleFrom(
1039                                 padding:
1040                                     const EdgeInsets.symmetric(horizontal:
1041                                         32, vertical: 12),
1042                                 child: const Text('Login'),
1043                             ),
1044                     ],

```

```

1045         ),
1046     ),
1047 ),
1048 );
1049 }
1050
1051 @override
1052 void dispose() {
1053     _usernameController.dispose();
1054     _passwordController.dispose();
1055     super.dispose();
1056 }
1057 }
1058
1059 logic
1060 import 'package:conference_app/features/auth/logic/state/
    token_holder.dart';
1061 import 'package:dio/dio.dart';
1062
1063 class AuthInterceptor extends Interceptor {
1064     final TokenGetter _tokenGetter;
1065
1066     AuthInterceptor(this._tokenGetter);
1067
1068     @override
1069     void onRequest(RequestOptions options,
        RequestInterceptorHandler handler) {
1070         final token = _tokenGetter.token;
1071         handler.next(
1072             options..headers['Authorization'] = token,
1073         );
1074     }
1075 }
1076
1077 token_holder.dart
1078 abstract interface class TokenGetter {
1079     String? get token;
1080 }
1081
1082 class TokenStateHolder implements TokenGetter {
1083     @override

```

```

1084     String? token;
1085 }
1086
1087 auth_manager.dart
1088 import 'package:conference_app/features/app/
      navigation_manager.dart';
1089 import 'package:conference_app/features/auth/data/auth_api.
      dart';
1090 import 'package:conference_app/features/auth/logic/state/
      token_holder.dart';
1091
1092 class AuthManager {
1093     final AuthApi _api;
1094     final TokenStateHolder _tokenStateHolder;
1095     final NavigationManager _navigationManager;
1096
1097     AuthManager(
1098         this._api,
1099         this._tokenStateHolder,
1100         this._navigationManager,
1101     );
1102
1103     Future<void> login(String username, String password) async
1104     {
1105         try {
1106             final token = await _api.authorize(username, password);
1107             _tokenStateHolder.token = token;
1108             _navigationManager.openConferenceList();
1109         } on Object {
1110             _navigationManager.showAlertDialog();
1111         }
1112     }
1113
1114 auth_api.dart
1115 import 'package:dio/dio.dart';
1116
1117 class AuthApi {
1118     final Dio _dio;
1119
1120     AuthApi(this._dio);

```

```

1121
1122     Future<String> authorize(String username, String password)
1123         async {
1124         final response = await _dio.post('/authorize',
1125             queryParameters: {
1126                 'name': username,
1127                 'password': password,
1128             });
1129         return response.data['token'];
1130     }
1131 }
1132
1133 conference_list.dart
1134 import 'package:conference_app/features/conferences/data/
1135     models/topic.dart';
1136 import 'package:conference_app/features/conferences/ui/
1137     widgets/conference_card.dart';
1138 import 'package:conference_app/features/di/
1139     conference_providers.dart';
1140 import 'package:flutter/material.dart';
1141 import 'package:flutter_riverpod/flutter_riverpod.dart';
1142
1143 import '../data/models/city.dart';
1144 import '../logic/manager/conference_list_manager.dart';
1145
1146 class ConferenceListScreen extends ConsumerStatefulWidget {
1147     const ConferenceListScreen({super.key});
1148
1149     @override
1150     ConsumerState<ConferenceListScreen> createState() =>
1151         _ConferenceListScreenState();
1152 }
1153
1154 class _ConferenceListScreenState extends ConsumerState<
1155     ConferenceListScreen> {
1156     late ConferenceListManager _conferenceListManager;
1157     final _nameController = TextEditingController();
1158     String? _nameFilter;
1159     int? _cityFilter;
1160     int? _topicFilter;

```

```

1156
1157 @override
1158 void initState() {
1159     super.initState();
1160     _conferenceListManager =
1161         ref.read(ConferenceProviders.conferenceListManager);
1162     WidgetsBinding.instance.addPostFrameCallback((_) {
1163         _conferenceListManager.fetch();
1164     });
1165 }
1166
1167 @override
1168 Widget build(BuildContext context) {
1169     final searchOptions = ref.watch(ConferenceProviders.
1170         searchOptionsProvider);
1171     final conferenceList = ref.watch(ConferenceProviders.
1172         listProvider);
1173
1174     return Scaffold(
1175         appBar: AppBar(title: const Text('Conferences')),
1176         floatingActionButton: FloatingActionButton(
1177             onPressed: () {
1178                 _conferenceListManager.openCreateConferenceScreen()
1179                 ;
1180             },
1181             child: const Icon(Icons.add),
1182         ),
1183         body: Column(
1184             children: [
1185                 Padding(
1186                     padding: const EdgeInsets.all(16.0),
1187                     child: Column(
1188                         children: [
1189                             _buildTextInput(
1190                                 'Name',
1191                                 (value) => setState(() => _nameFilter =
1192                                     value),
1193                             ),
1194                             if (searchOptions != null)
1195                                 Row(
1196                                     children: [

```



```

1193         Expanded(
1194             child: _buildDropDownInput(
1195                 'City ',
1196                 searchOptions.cities.cast<City?>().
1197                     firstWhere(
1198                         (city) => city?.id ==
1199                             _cityFilter,
1200                         orElse: () => null,
1201                     ),
1202                 searchOptions.cities,
1203                 (value) => value.name,
1204                 (value) => setState(() =>
1205                     _cityFilter = value?.id),
1206             ),
1207         ),
1208         const SizedBox(width: 16),
1209         Expanded(
1210             child: _buildDropDownInput(
1211                 'Topic ',
1212                 searchOptions.topics.cast<Topic?>()
1213                     .firstWhere(
1214                         (topic) => topic?.id ==
1215                             _topicFilter,
1216                         orElse: () => null,
1217                     ),
1218                 searchOptions.topics,
1219                 (value) => value.name,
1220                 (value) => setState(() =>
1221                     _topicFilter = value?.id),
1222             ),
1223         ),
1224     ],
1225 ),
1226 Row(
1227     children: [
1228         ElevatedButton(
1229             onPressed: () {
1230                 _conferenceListManager.updateSearch(
1231                     name: _nameFilter,
1232                     cityId: _cityFilter,
1233                     topicId: _topicFilter,

```

```

1228         );
1229     },
1230     child: const Text('Apply Filters '),
1231 ),
1232 const SizedBox(width: 16),
1233 ElevatedButton(
1234     onPressed: () {
1235         setState(() {
1236             _nameFilter = null;
1237             _cityFilter = null;
1238             _topicFilter = null;
1239             _nameController.clear();
1240         });
1241         _conferenceListManager.updateSearch(
1242             name: null,
1243             cityId: null,
1244             topicId: null,
1245         );
1246     },
1247     child: const Text('Reset Filters '),
1248 ),
1249 ],
1250 ),
1251 ],
1252 ),
1253 ),
1254 Expanded(
1255     child: conferenceList == null
1256         ? const Center(child:
1257             CircularProgressIndicator())
1258         : ListView.builder(
1259             itemCount: conferenceList.length,
1260             itemBuilder: (context, index) {
1261                 return GestureDetector(
1262                     onTap: () => _conferenceListManager
1263                         .openDetailedScreen(
1264                             conferenceList[index].id),
1265                     child:
1266                         ConferenceCard(conference:
1267                             conferenceList[index]),
1268                 );

```

```

1266         },
1267     ),
1268 ),
1269 ],
1270 ),
1271 );
1272 }
1273
1274 @override
1275 void dispose() {
1276     super.dispose();
1277     _nameController.dispose();
1278 }
1279
1280 Widget _buildTextInput(String label, ValueChanged<String>
    onChanged) {
1281     return Padding(
1282         padding: const EdgeInsets.symmetric(vertical: 8.0),
1283         child: TextField(
1284             controller: _nameController,
1285             decoration: InputDecoration(
1286                 labelText: label,
1287                 border: const OutlineInputBorder(),
1288             ),
1289             onChanged: onChanged,
1290         ),
1291     );
1292 }
1293
1294 Widget _buildDropdownInput<T, R>(
1295     String label,
1296     T? value,
1297     List<T> items,
1298     String Function(T) mapper,
1299     ValueChanged<T?> onChanged,
1300 ) {
1301     return Padding(
1302         padding: const EdgeInsets.symmetric(vertical: 8.0),
1303         child: DropdownButtonFormField<T>(
1304             value: value,
1305             decoration: InputDecoration(

```

```

1306         labelText: label ,
1307         border: const OutlineInputBorder() ,
1308     ),
1309     items: items
1310         .map(
1311             (item) => DropdownMenuItem<T>(
1312                 value: item ,
1313                 child: Text( mapper( item ) ) ,
1314             ) ,
1315         )
1316         .toList() ,
1317     onChanged: onChanged ,
1318 ),
1319 );
1320 }
1321 }
1322
1323 conference_detail_screen.dart
1324 import 'package:conference_app/features/conferences/data/
    models/conference_detail_response.dart';
1325 import 'package:conference_app/features/di/
    conference_providers.dart';
1326 import 'package:flutter/material.dart';
1327 import 'package:flutter_riverpod/flutter_riverpod.dart';
1328
1329 import '../data/models/conference_updating_request.dart';
1330 import '../logic/manager/conference_detailed_manager.dart';
1331
1332 class ConferenceDetailScreen extends ConsumerStatefulWidget {
1333     final int conferenceId;
1334
1335     const ConferenceDetailScreen({super.key, required this.
        conferenceId});
1336
1337     @override
1338     ConsumerState<ConferenceDetailScreen> createState() =>
1339         _ConferenceDetailScreenState();
1340 }
1341
1342 class _ConferenceDetailScreenState
1343     extends ConsumerState<ConferenceDetailScreen> {

```

```

1344 late ConferenceDetailedManager _conferenceDetailedManager;
1345 late Future<ConferenceDetailResponse>
    _conferenceDetailFuture;
1346 final TextEditingController _nameController =
    TextEditingController();
1347
1348 @override
1349 void initState() {
1350     super.initState();
1351     _conferenceDetailedManager =
1352         ref.read(ConferenceProviders.
            conferenceDetailedManager);
1353     _conferenceDetailFuture =
1354         _conferenceDetailedManager.getDetailedConference(
            widget.conferenceId);
1355 }
1356
1357 @override
1358 Widget build(BuildContext context) {
1359     return Scaffold(
1360         appBar: AppBar(title: const Text('Conference Details')),
1361         body: FutureBuilder<ConferenceDetailResponse>(
1362             future: _conferenceDetailFuture,
1363             builder: (context, snapshot) {
1364                 if (snapshot.connectionState == ConnectionState.
                    waiting) {
1365                     return const Center(child:
                        CircularProgressIndicator());
1366                 }
1367                 if (snapshot.hasError || !snapshot.hasData) {
1368                     return const Center(
1369                         child: Text('Error loading conference details')
1370                     );
1371                 }
1372
1373                 final conference = snapshot.data!;
1374                 _nameController.text = conference.name;
1375
1376                 return Padding(

```

```

1377 padding: const EdgeInsets.all(16.0),
1378 child: Column(
1379   crossAxisAlignment: CrossAxisAlignment.start,
1380   children: [
1381     if (conference.canRedact)
1382       TextField(
1383         enabled: conference.canRedact,
1384         controller: _nameController,
1385         decoration: const InputDecoration(
1386           labelText: 'Name', border:
1387             OutlineInputBorder()),
1388       else
1389         Text('Name: ${conference.name}',
1390           style: Theme.of(context).textTheme.
1391             bodyLarge),
1392       const SizedBox(height: 8),
1393       Text('City: ${conference.city}'),
1394       Text('Topic: ${conference.topic}'),
1395       const SizedBox(height: 8),
1396       Text('Start Date: ${conference.startDate}'),
1397       Text('End Date: ${conference.endDate}'),
1398       const SizedBox(height: 16),
1399       Text('Description: ${conference.description}'),
1400       Text('Contacts: ${conference.contacts}'),
1401       if (conference.canRedact) ...[
1402         const SizedBox(height: 16),
1403         Row(
1404           mainAxisAlignment: MainAxisAlignment.
1405             spaceBetween,
1406           children: [
1407             ElevatedButton(
1408               onPressed: () async {
1409                 await _conferenceDetailedManager.
1410                   updateConference(
1411                     conference.id,
1412                     ConferenceUpdatingRequest(
1413                       name: _nameController.text,
1414                     ),
1415                   );

```

```

1413         },
1414         child: const Text('Update'),
1415       ),
1416       ElevatedButton(
1417         onPressed: () async {
1418           await _conferenceDetailedManager
1419             .deleteConference(conference.id
1420               );
1421           // Optionally, navigate back or
1422             show a confirmation
1423         },
1424         style: ElevatedButton.styleFrom(
1425           foregroundColor: Colors.red,
1426         ),
1427         child: const Text('Delete'),
1428       ),
1429     ],
1430   ),
1431 ),
1432 );
1433 },
1434 ),
1435 );
1436 }
1437
1438 @override
1439 void dispose() {
1440   _nameController.dispose();
1441   super.dispose();
1442 }
1443 }
1444
1445 create_conference_screen.dart
1446 import 'package:conference_app/features/conferences/data/
1447   models/conference_request.dart';
1448 import 'package:conference_app/features/di/
1449   conference_providers.dart';
1450 import 'package:flutter/material.dart';
1451 import 'package:flutter_riverpod/flutter_riverpod.dart';

```

```

1450
1451 class CreateConferenceScreen extends ConsumerStatefulWidget {
1452     const CreateConferenceScreen({super.key});
1453
1454     @override
1455     ConsumerState<CreateConferenceScreen> createState() =>
1456         _CreateConferenceScreenState();
1457 }
1458
1459 class _CreateConferenceScreenState
1460     extends ConsumerState<CreateConferenceScreen> {
1461     final TextEditingController _nameController =
1462         TextEditingController();
1463     final TextEditingController _descriptionController =
1464         TextEditingController();
1465     final TextEditingController _contactsController =
1466         TextEditingController();
1467     final TextEditingController _imageUrlController =
1468         TextEditingController();
1469
1470     String? _selectedCity;
1471     String? _selectedTopic;
1472     DateTime? _selectedStartDate;
1473     DateTime? _selectedEndDate;
1474
1475     @override
1476     Widget build(BuildContext context) {
1477         final searchOptions = ref.watch(ConferenceProviders.
1478             searchOptionsProvider);
1479         final conferenceListManager =
1480             ref.watch(ConferenceProviders.conferenceListManager);
1481
1482         return Scaffold(
1483             appBar: AppBar(title: const Text('Create Conference')),
1484             body: Padding(
1485                 padding: const EdgeInsets.all(16.0),
1486                 child: ListView(
1487                     children: [
1488                         _buildTextInput('Name', _nameController),
1489                         _buildTextInput('Description',
1490                             _descriptionController),

```



```

1485         _buildTextInput('Contacts', _contactsController),
1486         _buildDatePicker(context, 'Start Date',
1487             _selectedStartDate, (date) {
1488                 setState(() => _selectedStartDate = date);
1489             }),
1490         _buildDatePicker(context, 'End Date',
1491             _selectedEndDate, (date) {
1492                 setState(() => _selectedEndDate = date);
1493             }),
1494         // _buildTextInput('Image URL',
1495             _imageUrlController),
1496         if (searchOptions != null) ...[
1497             _buildDropdownInput(
1498                 'City', searchOptions.cities.map((e) => e.
1499                     name).toList(),
1500                 (value) {
1501                     setState(() => _selectedCity = value);
1502                 }),
1503             _buildDropdownInput(
1504                 'Topic', searchOptions.topics.map((e) => e.
1505                     name).toList(),
1506                 (value) {
1507                     setState(() => _selectedTopic = value);
1508                 }),
1509         ],
1510         const SizedBox(height: 16),
1511         ElevatedButton(
1512             onPressed: () async {
1513                 if (_selectedCity == null ||
1514                     _selectedTopic == null ||
1515                     _selectedStartDate == null ||
1516                     _selectedEndDate == null) {
1517                     return;
1518                 }
1519                 final request = ConferenceRequest(
1520                     name: _nameController.text,
1521                     cityId: searchOptions?.cities
1522                         .firstWhere((e) => e.name ==
1523                             _selectedCity)
1524                         .id ??

```

```

1520         -1,
1521         topicId: searchOptions?.topics
1522             .firstWhere((e) => e.name ==
1523                 _selectedTopic)
1524             .id ??
1525         -1,
1526         startDate: _selectedStartDate!.
1527             toIso8601String(),
1528         endDate: _selectedEndDate!.toIso8601String
1529             (),
1530         description: _descriptionController.text,
1531         contacts: _contactsController.text,
1532         imageUrl: _imageUrlController.text,
1533     );
1534     await conferenceListManager.createConference(
1535         request);
1536     },
1537     child: const Text('Create Conference'),
1538 ],
1539 ),
1540 );
1541 }
1542 Widget _buildTextInput(String label, TextEditingController
1543     controller) {
1544     return Padding(
1545         padding: const EdgeInsets.symmetric(vertical: 8.0),
1546         child: TextField(
1547             decoration: InputDecoration(
1548                 labelText: label,
1549                 border: const OutlineInputBorder(),
1550             ),
1551             controller: controller,
1552         ),
1553     );
1554 }
1555 Widget _buildDropDownInput(

```

```

1556         String label , List<String> items , ValueChanged<String?>
           onChanged) {
1557     return Padding(
1558         padding: const EdgeInsets.symmetric(vertical: 8.0) ,
1559         child: DropdownButtonFormField<String>(
1560             decoration: InputDecoration(
1561                 labelText: label ,
1562                 border: const OutlineInputBorder() ,
1563             ) ,
1564             items: items
               .map((item) =>
1565                 DropdownMenuItem<String>(value: item , child:
                   Text(item)))
1566             .toList() ,
1567             onChanged: onChanged ,
1568         ) ,
1569     );
1570 }
1571
1572
1573 Widget _buildDatePicker(BuildContext context , String label ,
1574     DateTime? selectedDate , ValueChanged<DateTime?>
       onDateSelected) {
1575     return Padding(
1576         padding: const EdgeInsets.symmetric(vertical: 8.0) ,
1577         child: InkWell(
1578             onTap: () async {
1579                 DateTime? pickedDate = await showDatePicker(
1580                     context: context ,
1581                     initialDate: selectedDate ?? DateTime.now() ,
1582                     firstDate: DateTime(2000) ,
1583                     lastDate: DateTime(2100) ,
1584                 );
1585                 onDateSelected(pickedDate);
1586             } ,
1587             child: InputDecorator(
1588                 decoration: InputDecoration(
1589                     labelText: label ,
1590                     border: const OutlineInputBorder() ,
1591                 ) ,
1592                 child: Text(selectedDate != null
1593                     ? selectedDate.toLocal().toString().split(' ')

```

```

                                [0]
1594         : 'Select Date' ),
1595     ),
1596 ),
1597 );
1598 }
1599
1600 @override
1601 void dispose() {
1602     _nameController.dispose();
1603     _descriptionController.dispose();
1604     _contactsController.dispose();
1605     _imageUrlController.dispose();
1606     super.dispose();
1607 }
1608 }
1609
1610 conference_card.dart
1611 import 'package:flutter/material.dart';
1612
1613 import '../.. / data / models / conference_short_response.dart';
1614
1615 class ConferenceCard extends StatelessWidget {
1616     final ConferenceShortResponse conference;
1617
1618     const ConferenceCard({super.key, required this.conference})
1619         ;
1620
1621 @override
1622 Widget build(BuildContext context) => Card(
1623     elevation: 6,
1624     shape: RoundedRectangleBorder(
1625         borderRadius: BorderRadius.circular(12),
1626     ),
1627     margin: const EdgeInsets.symmetric(vertical: 8,
1628         horizontal: 16),
1629     child: Padding(
1630         padding: const EdgeInsets.all(16),
1631         child: Column(
1632             crossAxisAlignment: CrossAxisAlignment.start,
1633             children: [

```

```

1632         Text(
1633             conference.name,
1634             style: Theme.of(context).textTheme.
                titleMedium?.copyWith(
1635                 fontWeight: FontWeight.bold,
1636                 color: Colors.black45,
1637             ),
1638         ),
1639         const SizedBox(height: 12),
1640         Row(
1641             children: [
1642                 const Icon(Icons.location_city, size: 16,
                    color: Colors.black45),
1643                 const SizedBox(width: 4),
1644                 Text(
1645                     conference.city,
1646                     style: Theme.of(context).textTheme.
                        bodyMedium,
1647                 ),
1648             ],
1649         ),
1650         Row(
1651             children: [
1652                 const Icon(Icons.topic, size: 16, color:
                    Colors.black45),
1653                 const SizedBox(width: 4),
1654                 Text(
1655                     conference.topic,
1656                     style: Theme.of(context).textTheme.
                        bodyMedium,
1657                 ),
1658             ],
1659         ),
1660         const SizedBox(height: 12),
1661         Divider(color: Colors.grey[300]),
1662         const SizedBox(height: 8),
1663         Row(
1664             children: [
1665                 const Icon(Icons.date_range, size: 16,
                    color: Colors.black45),
1666                 const SizedBox(width: 4),

```

```

1667         Text(
1668             'Start: ${conference.startDate}',
1669             style: Theme.of(context).textTheme.
                bodyMedium,
1670         ),
1671     ],
1672 ),
1673 Row(
1674     children: [
1675         const Icon(Icons.date_range, size: 16,
            color: Colors.black45),
1676         const SizedBox(width: 4),
1677         Text(
1678             'End: ${conference.endDate}',
1679             style: Theme.of(context).textTheme.
                bodyMedium,
1680         ),
1681     ],
1682 ),
1683 ],
1684 ),
1685 ),
1686 );
1687 }
1688
1689 conference_search_options_notifier.dart
1690 import 'package:flutter_riverpod/flutter_riverpod.dart';
1691
1692 import '../.. / data / models / conference_search_options.dart';
1693
1694 class ConferenceSearchOptionsNotifier
1695     extends StateNotifier<ConferenceSearchOptions?> {
1696     ConferenceSearchOptionsNotifier() : super(null);
1697
1698     void updateSearchOptions(ConferenceSearchOptions? options)
1699     {
1700         state = options;
1701     }
1702 }
1703 conference_list_notifier.dart
1704 import 'package:flutter_riverpod/flutter_riverpod.dart';

```

```

1704
1705 import '../.. / data / models / conference_short_response . dart ';
1706
1707 class ConferenceListNotifier
1708     extends StateNotifier <List<ConferenceShortResponse>?> {
1709     ConferenceListNotifier() : super(null);
1710
1711     void updateConferenceList(List<ConferenceShortResponse>?
1712         conferenceList) {
1713         state = conferenceList;
1714     }
1715
1716     void removeConference(int conferenceId) {
1717         state = state?.where((item) => item.id != conferenceId).
1718             toList();
1719     }
1720 }
1721
1722 conference_list_manager . dart
1723 import 'package:conference_app / features / app /
1724     navigation_manager . dart ';
1725 import 'package:conference_app / features / conferences / data /
1726     conference_api . dart ';
1727 import 'package:conference_app / features / conferences / data /
1728     models / conference_request . dart ';
1729
1730 import '../ state / conference_list_notifier . dart ';
1731 import '../ state / conference_search_options_notifier . dart ';
1732
1733 class ConferenceListManager {
1734     final ConferenceApi _api;
1735     final NavigationManager _navigationManager;
1736     final ConferenceSearchOptionsNotifier
1737         _conferenceSearchOptionsNotifier;
1738     final ConferenceListNotifier _conferenceListNotifier;
1739
1740     ConferenceListManager(
1741         this._api ,
1742         this._navigationManager ,
1743         this._conferenceSearchOptionsNotifier ,
1744         this._conferenceListNotifier ,

```

```

1739 );
1740
1741 Future<void> fetch() async {
1742     try {
1743         final options = await _api.getSearchOptions();
1744         _conferenceSearchOptionsNotifier.updateSearchOptions(
1745             options);
1746         final list = await _api.getConferencesList();
1747         _conferenceListNotifier.updateConferenceList(list);
1748     } on Object {
1749         _navigationManager.showErrorDialog();
1750         rethrow;
1751     }
1752 }
1753
1754 Future<void> updateSearch({int? cityId, int? topicId,
1755     String? name}) async {
1756     final list = await _api.getConferencesList(
1757         cityId: cityId,
1758         topicId: topicId,
1759         name: name,
1760     );
1761     _conferenceListNotifier.updateConferenceList(list);
1762 }
1763
1764 void openDetailedScreen(int conferenceId) =>
1765     _navigationManager.openDetailedConference(conferenceId)
1766     ;
1767
1768 void openCreateConferenceScreen() =>
1769     _navigationManager.openCreateConferenceScreen();
1770
1771 Future<void> createConference(ConferenceRequest request)
1772     async {
1773     try {
1774         await _api.createConference(request);
1775         _navigationManager.pop();
1776     } on Object {
1777         _navigationManager.showErrorDialog();
1778     }
1779 }

```



```

1776 }
1777
1778 conference_detailed_manager.dart
1779 import '../../../app/navigation_manager.dart';
1780 import '../../../data/conference_api.dart';
1781 import '../../../data/models/conference_detail_response.dart';
1782 import '../../../data/models/conference_updating_request.dart';
1783 import '../../../state/conference_list_notifier.dart';
1784
1785 class ConferenceDetailedManager {
1786   final ConferenceApi _api;
1787   final NavigationManager _navigationManager;
1788   final ConferenceListNotifier _conferenceListNotifier;
1789
1790   ConferenceDetailedManager(
1791     this._api,
1792     this._navigationManager,
1793     this._conferenceListNotifier,
1794   );
1795
1796   Future<ConferenceDetailResponse> getDetailedConference(int
1797     conferenceId) =>
1798     _api.getConferenceDetails(conferenceId);
1799
1800   Future<void> deleteConference(int conferenceId) async {
1801     try {
1802       await _api.deleteConference(conferenceId);
1803       _conferenceListNotifier.removeConference(conferenceId);
1804     } on Object {
1805       _navigationManager.showAlertDialog();
1806     }
1807   }
1808
1809   Future<void> updateConference(
1810     int conferenceId,
1811     ConferenceUpdatingRequest request,
1812   ) async {
1813     try {
1814       await _api.updateConference(conferenceId, request);
1815     } on Object {
1816       _navigationManager.showAlertDialog();
1817     }
1818   }
1819 }

```

```

1816     }
1817   }
1818 }
1819
1820 conference_updating_request.dart
1821 import 'package:freezed_annotation/freezed_annotation.dart';
1822
1823 part 'conference_updating_request.freezed.dart';
1824 part 'conference_updating_request.g.dart';
1825
1826 @freezed
1827 class ConferenceUpdatingRequest with
1828   _$ConferenceUpdatingRequest {
1829   const factory ConferenceUpdatingRequest({
1830     String? name,
1831     int? cityId,
1832     int? topicId,
1833     String? startDate,
1834     String? endDate,
1835     String? description,
1836     String? contacts,
1837     String? imageUrl,
1838   }) = _ConferenceUpdatingRequest;
1839
1840   factory ConferenceUpdatingRequest.fromJson(Map<String,
1841     Object?> json) =>
1842     _$ConferenceUpdatingRequestFromJson(json);
1843 }
1844
1845 conference_search_options.g.dart
1846 // GENERATED CODE – DO NOT MODIFY BY HAND
1847
1848 part of 'conference_search_options.dart';
1849
1850 //
1851     *****
1852
1853 // JsonSerializerGenerator
1854 //
1855     *****

```

```

1851
1852 _$ConferenceSearchOptionsImpl _\begin{equation}\begin{
gathered}ConferenceSearchOptionsImplFromJson(
1853     Map<String , dynamic> json) =>
1854     _$ConferenceSearchOptionsImpl(
1855         cities: (json['cities '] as List<dynamic>)
1856             .map((e) => City.fromJson(e as Map<String , dynamic
>))
1857             .toList() ,
1858         topics: (json['topics '] as List<dynamic>)
1859             .map((e) => Topic.fromJson(e as Map<String , dynamic
>))
1860             .toList() ,
1861     );
1862
1863 Map<String , dynamic> _\end{gathered}\end{equation}
ConferenceSearchOptionsImplToJson(
1864     _$ConferenceSearchOptionsImpl instance) =>
1865     <String , dynamic>{
1866         'cities ': instance.cities ,
1867         'topics ': instance.topics ,
1868     };
1869
1870 conference_detail_response.dart
1871 import 'package:freezed_annotation/freezed_annotation.dart';
1872
1873 part 'conference_detail_response.freezed.dart';
1874 part 'conference_detail_response.g.dart';
1875
1876 @freezed
1877 class ConferenceDetailResponse with
    _$ConferenceDetailResponse {
1878     const factory ConferenceDetailResponse({
1879         required int id ,
1880         required String name ,
1881         required String city ,
1882         required String topic ,
1883         required String startDate ,
1884         required String endDate ,
1885         required String description ,
1886         required String contacts ,

```

```

1887     required bool canRedact ,
1888   }) = _ConferenceDetailResponse ;
1889
1890   factory ConferenceDetailResponse.fromJson(Map<String ,
1891     dynamic> json) =>
1892     _$ConferenceDetailResponseFromJson(json) ;
1893 }
1894
1895 conference_updating_request.freezed.dart
1896 // coverage:ignore-file
1897 // GENERATED CODE - DO NOT MODIFY BY HAND
1898 // ignore_for_file: type=lint
1899 // ignore_for_file: unused_element, deprecated_member_use ,
1900 // deprecated_member_use_from_same_package ,
1901 // use_function_type_syntax_for_parameters , unnecessary_const
1902 // , avoid_init_to_null ,
1903 // invalid_override_different_default_values_named ,
1904 // prefer_expression_function_bodies , annotate_overrides ,
1905 // invalid_annotation_target , unnecessary_question_mark
1906
1907 part of 'conference_updating_request.dart' ;
1908
1909 //
1910
1911 *****
1912
1913 // FreezedGenerator
1914 //
1915 *****
1916
1917 T _$identity<T>(T value) => value ;
1918
1919 final _privateConstructorUsedError = UnsupportedError(
1920   'It seems like you constructed your class using 'MyClass.
1921     _()' '. This constructor is only meant to be used by
1922     freezed and you are not supposed to need it nor use it
1923     .\nPlease check the documentation here for more
1924     information: https://github.com/rrousselGit/freezed#
1925     adding-getters-and-methods-to-our-models ');
1926
1927 ConferenceUpdatingRequest _$ConferenceUpdatingRequestFromJson

```

```

1912     Map<String , dynamic> json) {
1913     return _ConferenceUpdatingRequest.fromJson(json);
1914 }
1915
1916 /// @nodoc
1917 mixin _$ConferenceUpdatingRequest {
1918     String? get name => throw _privateConstructorUsedError;
1919     int? get cityId => throw _privateConstructorUsedError;
1920     int? get topicId => throw _privateConstructorUsedError;
1921     String? get startDate => throw _privateConstructorUsedError
1922     ;
1923     String? get endDate => throw _privateConstructorUsedError;
1924     String? get description => throw
1925         _privateConstructorUsedError;
1926     String? get contacts => throw _privateConstructorUsedError;
1927     String? get imageUrl => throw _privateConstructorUsedError;
1928     Map<String , dynamic> toJson() => throw
1929         _privateConstructorUsedError;
1930     @JsonKey(ignore: true)
1931     _$ConferenceUpdatingRequestCopyWith<
1932         ConferenceUpdatingRequest> get copyWith =>
1933         throw _privateConstructorUsedError;
1934 }
1935
1936 /// @nodoc
1937 abstract class _$ConferenceUpdatingRequestCopyWith<$Res> {
1938     factory _$ConferenceUpdatingRequestCopyWith(
1939         ConferenceUpdatingRequest value ,
1940         $Res Function(ConferenceUpdatingRequest) then) =
1941         _$ConferenceUpdatingRequestCopyWithImpl<$Res ,
1942             ConferenceUpdatingRequest>;
1943
1944     @useResult
1945     $Res call(
1946         {String? name ,
1947         int? cityId ,
1948         int? topicId ,
1949         String? startDate ,
1950         String? endDate ,
1951         String? description ,

```

```

1946         String? contacts ,
1947         String? imageUrl});
1948     }
1949
1950     /// @nodoc
1951     class _$ConferenceUpdatingRequestCopyWithImpl<$Res ,
1952         $Val extends ConferenceUpdatingRequest>
1953         implements $ConferenceUpdatingRequestCopyWith<$Res> {
1954         _$ConferenceUpdatingRequestCopyWithImpl(this._value , this.
            _then);
1955
1956         // ignore: unused_field
1957         final $Val _value;
1958         // ignore: unused_field
1959         final $Res Function($Val) _then;
1960
1961         @pragma('vm:prefer-inline ')
1962         @override
1963         $Res call({
1964             Object? name = freezed ,
1965             Object? cityId = freezed ,
1966             Object? topicId = freezed ,
1967             Object? startDate = freezed ,
1968             Object? endDate = freezed ,
1969             Object? description = freezed ,
1970             Object? contacts = freezed ,
1971             Object? imageUrl = freezed ,
1972         }) {
1973             return _then(_value.copyWith(
1974                 name: freezed == name
1975                     ? _value.name
1976                     : name // ignore: cast_nullable_to_non_nullable
1977                         as String?,
1978                 cityId: freezed == cityId
1979                     ? _value.cityId
1980                     : cityId // ignore: cast_nullable_to_non_nullable
1981                         as int?,
1982                 topicId: freezed == topicId
1983                     ? _value.topicId
1984                     : topicId // ignore: cast_nullable_to_non_nullable
1985                         as int?,

```

```

1986         startDate: freezed == startDate
1987             ? _value.startDate
1988             : startDate // ignore:
1989                 cast_nullable_to_non_nullable
1990                 as String?,
1991     endDate: freezed == endDate
1992         ? _value.endDate
1993         : endDate // ignore: cast_nullable_to_non_nullable
1994         as String?,
1995     description: freezed == description
1996         ? _value.description
1997         : description // ignore:
1998             cast_nullable_to_non_nullable
1999             as String?,
2000     contacts: freezed == contacts
2001         ? _value.contacts
2002         : contacts // ignore: cast_nullable_to_non_nullable
2003         as String?,
2004     imageUrl: freezed == imageUrl
2005         ? _value.imageUrl
2006         : imageUrl // ignore: cast_nullable_to_non_nullable
2007         as String?,
2008     ) as $Val);
2009 }
2010 }
2011 /// @nodoc
2012 abstract class _\begin{equation}\begin{gathered}
2013     ConferenceUpdatingRequestImplCopyWith<$Res>
2014     implements $ConferenceUpdatingRequestCopyWith<$Res> {
2015     factory _\end{gathered}\end{equation}
2016         ConferenceUpdatingRequestImplCopyWith(
2017         _$ConferenceUpdatingRequestImpl value,
2018         $Res Function(_$ConferenceUpdatingRequestImpl) then
2019         ) =
2020         __\begin{equation}\begin{gathered}
2021             ConferenceUpdatingRequestImplCopyWithImpl<$Res>;
2022         @override
2023         @useResult
2024         $Res call(
2025             {String? name,

```

```

2021         int? cityId ,
2022         int? topicId ,
2023         String? startDate ,
2024         String? endDate ,
2025         String? description ,
2026         String? contacts ,
2027         String? imageUrl});
2028     }
2029
2030     /// @nodoc
2031     class __\end{gathered}\end{equation}
        ConferenceUpdatingRequestImplCopyWithImpl<$Res>
2032         extends _$ConferenceUpdatingRequestCopyWithImpl<$Res ,
2033             _$ConferenceUpdatingRequestImpl>
2034         implements _\begin{equation}\begin{gathered}
            ConferenceUpdatingRequestImplCopyWith<$Res> {
2035     __\end{gathered}\end{equation}
        ConferenceUpdatingRequestImplCopyWithImpl(
2036         _$ConferenceUpdatingRequestImpl _value ,
2037         $Res Function(_$ConferenceUpdatingRequestImpl) _then)
2038         : super(_value , _then);
2039
2040     @pragma('vm:prefer-inline ')
2041     @override
2042     $Res call({
2043         Object? name = freezed ,
2044         Object? cityId = freezed ,
2045         Object? topicId = freezed ,
2046         Object? startDate = freezed ,
2047         Object? endDate = freezed ,
2048         Object? description = freezed ,
2049         Object? contacts = freezed ,
2050         Object? imageUrl = freezed ,
2051     }) {
2052         return _then(_$ConferenceUpdatingRequestImpl(
2053             name: freezed == name
2054                 ? _value.name
2055                 : name // ignore: cast_nullable_to_non_nullable
2056                 as String?,
2057             cityId: freezed == cityId
2058                 ? _value.cityId

```



```

2059         : cityId // ignore: cast_nullable_to_non_nullable
2060           as int?,
2061     topicId: freezed == topicId
2062       ? _value.topicId
2063       : topicId // ignore: cast_nullable_to_non_nullable
2064         as int?,
2065     startDate: freezed == startDate
2066       ? _value.startDate
2067       : startDate // ignore:
2068         cast_nullable_to_non_nullable
2069         as String?,
2070     endDate: freezed == endDate
2071       ? _value.endDate
2072       : endDate // ignore: cast_nullable_to_non_nullable
2073         as String?,
2074     description: freezed == description
2075       ? _value.description
2076       : description // ignore:
2077         cast_nullable_to_non_nullable
2078         as String?,
2079     contacts: freezed == contacts
2080       ? _value.contacts
2081       : contacts // ignore: cast_nullable_to_non_nullable
2082         as String?,
2083     imageUrl: freezed == imageUrl
2084       ? _value.imageUrl
2085       : imageUrl // ignore: cast_nullable_to_non_nullable
2086         as String?,
2087   ));
2088 }
2089
2090 /// @nodoc
2091 @JsonSerializable()
2092 class _$ConferenceUpdatingRequestImpl implements
2093   _ConferenceUpdatingRequest {
2094   const _$ConferenceUpdatingRequestImpl(
2095     {this.name,
2096     this.cityId,
2097     this.topicId,
2098     this.startDate,

```

```

2097         this.endDate,
2098         this.description,
2099         this.contacts,
2100         this.imageUrl));
2101
2102     factory _$ConferenceUpdatingRequestImpl.fromJson(Map<String
        , dynamic> json) =>
2103         _\begin{equation}\begin{gathered}
            ConferenceUpdatingRequestImplFromJson(json);
2104
2105     @override
2106     final String? name;
2107     @override
2108     final int? cityId;
2109     @override
2110     final int? topicId;
2111     @override
2112     final String? startDate;
2113     @override
2114     final String? endDate;
2115     @override
2116     final String? description;
2117     @override
2118     final String? contacts;
2119     @override
2120     final String? imageUrl;
2121
2122     @override
2123     String toString() {
2124         return 'ConferenceUpdatingRequest(name: $name, cityId:
            $cityId, topicId: $topicId, startDate: $startDate,
            endDate: $endDate, description: $description, contacts
            : $contacts, imageUrl: $imageUrl)';
2125     }
2126
2127     @override
2128     bool operator ==(Object other) {
2129         return identical(this, other) ||
2130             (other.runtimeType == runtimeType &&
2131                 other is _$ConferenceUpdatingRequestImpl &&
2132                 (identical(other.name, name) || other.name ==

```

```

        name) &&
2133         (identical(other.cityId , cityId) || other.cityId
           == cityId) &&
2134         (identical(other.topicId , topicId) || other.
           topicId == topicId) &&
2135         (identical(other.startDate , startDate) ||
2136           other.startDate == startDate) &&
2137         (identical(other.endDate , endDate) || other.
           endDate == endDate) &&
2138         (identical(other.description , description) ||
2139           other.description == description) &&
2140         (identical(other.contacts , contacts) ||
2141           other.contacts == contacts) &&
2142         (identical(other.imageUrl , imageUrl) ||
2143           other.imageUrl == imageUrl));
2144     }
2145
2146     @JsonKey(ignore: true)
2147     @override
2148     int get hashCode => Object.hash(runtimeType , name , cityId ,
        topicId , startDate ,
2149         endDate , description , contacts , imageUrl);
2150
2151     @JsonKey(ignore: true)
2152     @override
2153     @pragma('vm:prefer-inline ')
2154     _\end{gathered}\end{equation}
        ConferenceUpdatingRequestImplCopyWith<
        _$ConferenceUpdatingRequestImpl>
2155         get copyWith => __\begin{equation}\begin{gathered}
            ConferenceUpdatingRequestImplCopyWithImpl<
2156                 _$ConferenceUpdatingRequestImpl>(this , _$identity);
2157
2158     @override
2159     Map<String , dynamic> toJson() {
2160         return _\end{gathered}\end{equation}
            ConferenceUpdatingRequestImplToJson(
2161                 this ,
2162             );
2163     }
2164 }

```

```

2165
2166 abstract class _ConferenceUpdatingRequest implements
      ConferenceUpdatingRequest {
2167     const factory _ConferenceUpdatingRequest(
2168         {final String? name,
2169         final int? cityId,
2170         final int? topicId,
2171         final String? startDate,
2172         final String? endDate,
2173         final String? description,
2174         final String? contacts,
2175         final String? imageUrl}) =
          _$ConferenceUpdatingRequestImpl;
2176
2177     factory _ConferenceUpdatingRequest.fromJson(Map<String,
          dynamic> json) =
2178         _$ConferenceUpdatingRequestImpl.fromJson;
2179
2180     @override
2181     String? get name;
2182     @override
2183     int? get cityId;
2184     @override
2185     int? get topicId;
2186     @override
2187     String? get startDate;
2188     @override
2189     String? get endDate;
2190     @override
2191     String? get description;
2192     @override
2193     String? get contacts;
2194     @override
2195     String? get imageUrl;
2196     @override
2197     @JsonKey(ignore: true)
2198     _\begin{equation}\begin{gathered}
          ConferenceUpdatingRequestImplCopyWith<
          _$ConferenceUpdatingRequestImpl>
2199         get copyWith => throw _privateConstructorUsedError;
2200 }

```

```

2201
2202 conference_short_response.dart
2203 import 'package:freezed_annotation/freezed_annotation.dart';
2204
2205 part 'conference_short_response.freezed.dart';
2206 part 'conference_short_response.g.dart';
2207
2208 @freezed
2209 class ConferenceShortResponse with _$ConferenceShortResponse
2210 {
2211   const factory ConferenceShortResponse({
2212     required int id,
2213     required String name,
2214     required String city,
2215     required String topic,
2216     required String startDate,
2217     required String endDate,
2218   }) = _ConferenceShortResponse;
2219
2220   factory ConferenceShortResponse.fromJson(Map<String,
2221     dynamic> json) =>
2222     _ConferenceShortResponseFromJson(json);
2223 }
2224
2225 conference_short_response.freezed.dart
2226 // coverage:ignore-file
2227 // GENERATED CODE - DO NOT MODIFY BY HAND
2228 // ignore_for_file: type=lint
2229 // ignore_for_file: unused_element, deprecated_member_use,
2230   deprecated_member_use_from_same_package,
2231   use_function_type_syntax_for_parameters, unnecessary_const
2232   , avoid_init_to_null,
2233   invalid_override_different_default_values_named,
2234   prefer_expression_function_bodies, annotate_overrides,
2235   invalid_annotation_target, unnecessary_question_mark
2236
2237 part of 'conference_short_response.dart';
2238
2239 //
2240
2241 *****

```

```

2232 // FreezedGenerator
2233 //
2234 *****
2235 T _$identity<T>(T value) => value;
2236
2237 final _privateConstructorUsedError = UnsupportedError(
2238     'It seems like you constructed your class using `MyClass.
2239         _()`. This constructor is only meant to be used by
2240         freezed and you are not supposed to need it nor use it
2241         .\nPlease check the documentation here for more
2242         information: https://github.com/rrousselGit/freezed#
2243         adding-getters-and-methods-to-our-models`);
2244
2245 ConferenceShortResponse _$ConferenceShortResponseFromJson(
2246     Map<String, dynamic> json) {
2247     return _ConferenceShortResponse.fromJson(json);
2248 }
2249
2250 /// @nodoc
2251 mixin _$ConferenceShortResponse {
2252     int get id => throw _privateConstructorUsedError;
2253     String get name => throw _privateConstructorUsedError;
2254     String get city => throw _privateConstructorUsedError;
2255     String get topic => throw _privateConstructorUsedError;
2256     String get startDate => throw _privateConstructorUsedError;
2257     String get endDate => throw _privateConstructorUsedError;
2258
2259     Map<String, dynamic> toJson() => throw
2260         _privateConstructorUsedError;
2261     @JsonKey(ignore: true)
2262     $ConferenceShortResponseCopyWith<ConferenceShortResponse>
2263         get copyWith =>
2264         throw _privateConstructorUsedError;
2265 }
2266
2267 /// @nodoc
2268 abstract class $ConferenceShortResponseCopyWith<$Res> {
2269     factory $ConferenceShortResponseCopyWith(
2270         ConferenceShortResponse value,

```

```

2263         $Res Function(ConferenceShortResponse) then) =
2264         _$ConferenceShortResponseCopyWithImpl<$Res ,
                ConferenceShortResponse >;
2265     @useResult
2266     $Res call(
2267         {int id ,
2268         String name ,
2269         String city ,
2270         String topic ,
2271         String startDate ,
2272         String endDate});
2273 }
2274
2275 /// @nodoc
2276 class _$ConferenceShortResponseCopyWithImpl<$Res ,
2277         $Val extends ConferenceShortResponse>
2278     implements $ConferenceShortResponseCopyWith<$Res> {
2279     _$ConferenceShortResponseCopyWithImpl(this._value , this .
        _then);
2280
2281     // ignore: unused_field
2282     final $Val _value;
2283     // ignore: unused_field
2284     final $Res Function($Val) _then;
2285
2286     @pragma('vm:prefer-inline ')
2287     @override
2288     $Res call({
2289         Object? id = null ,
2290         Object? name = null ,
2291         Object? city = null ,
2292         Object? topic = null ,
2293         Object? startDate = null ,
2294         Object? endDate = null ,
2295     }) {
2296         return _then(_value.copyWith(
2297             id: null == id
2298                 ? _value.id
2299                 : id // ignore: cast_nullable_to_non_nullable
2300                 as int ,
2301             name: null == name

```

```

2302         ? _value.name
2303         : name // ignore: cast_nullable_to_non_nullable
2304         as String ,
2305     city: null == city
2306         ? _value.city
2307         : city // ignore: cast_nullable_to_non_nullable
2308         as String ,
2309     topic: null == topic
2310         ? _value.topic
2311         : topic // ignore: cast_nullable_to_non_nullable
2312         as String ,
2313     startDate: null == startDate
2314         ? _value.startDate
2315         : startDate // ignore:
2316             cast_nullable_to_non_nullable
2317             as String ,
2318     endDate: null == endDate
2319         ? _value.endDate
2320         : endDate // ignore: cast_nullable_to_non_nullable
2321         as String ,
2322     ) as $Val);
2323 }
2324 }
2325 /// @nodoc
2326 abstract class _\end{gathered}\end{equation}
2327     ConferenceShortResponseImplCopyWith<$Res>
2328     implements $ConferenceShortResponseCopyWith<$Res> {
2329     factory _\begin{equation}\begin{gathered}
2330         ConferenceShortResponseImplCopyWith(
2331             _$ConferenceShortResponseImpl value ,
2332             $Res Function(_$ConferenceShortResponseImpl) then)
2333         =
2334         __\end{gathered}\end{equation}
2335         ConferenceShortResponseImplCopyWithImpl<$Res>;
2336 @override
2337 @useResult
2338 $Res call(
2339     {int id ,
2340     String name ,
2341     String city ,

```



```

2338         String topic ,
2339         String startDate ,
2340         String endDate));
2341     }
2342
2343     /// @nodoc
2344     class __\begin{equation}\begin{gathered}
2345         ConferenceShortResponseImplCopyWithImpl<$Res>
2346         extends _$ConferenceShortResponseCopyWithImpl<$Res ,
2347             _$ConferenceShortResponseImpl>
2348         implements __\end{gathered}\end{equation}
2349             ConferenceShortResponseImplCopyWith<$Res> {
2350         __\begin{equation}\begin{gathered}
2351             ConferenceShortResponseImplCopyWithImpl(
2352                 _$ConferenceShortResponseImpl _value ,
2353                 $Res Function(_$ConferenceShortResponseImpl) _then)
2354             : super(_value , _then);
2355
2356         @pragma('vm:prefer-inline ')
2357         @override
2358         $Res call({
2359             Object? id = null ,
2360             Object? name = null ,
2361             Object? city = null ,
2362             Object? topic = null ,
2363             Object? startDate = null ,
2364             Object? endDate = null ,
2365         }) {
2366             return _then(_$ConferenceShortResponseImpl(
2367                 id: null == id
2368                     ? _value.id
2369                     : id // ignore: cast_nullable_to_non_nullable
2370                     as int ,
2371                 name: null == name
2372                     ? _value.name
2373                     : name // ignore: cast_nullable_to_non_nullable
2374                     as String ,
2375                 city: null == city
2376                     ? _value.city
2377                     : city // ignore: cast_nullable_to_non_nullable
2378                     as String ,

```

```

2376         topic: null == topic
2377             ? _value.topic
2378             : topic // ignore: cast_nullable_to_non_nullable
2379                 as String ,
2380         startDate: null == startDate
2381             ? _value.startDate
2382             : startDate // ignore:
2383                 cast_nullable_to_non_nullable
2384                 as String ,
2384         endDate: null == endDate
2385             ? _value.endDate
2386             : endDate // ignore: cast_nullable_to_non_nullable
2387                 as String ,
2388     ));
2389 }
2390 }
2391
2392 /// @nodoc
2393 @JsonSerializable()
2394 class _$ConferenceShortResponseImpl implements
2395     _ConferenceShortResponse {
2395     const _$ConferenceShortResponseImpl(
2396         {required this.id ,
2397         required this.name ,
2398         required this.city ,
2399         required this.topic ,
2400         required this.startDate ,
2401         required this.endDate});
2402
2403     factory _$ConferenceShortResponseImpl.fromJson(Map<String ,
2404         dynamic> json) =>
2405         _\end{gathered}\end{equation}
2406             ConferenceShortResponseImplFromJson(json);
2407
2408     @override
2409     final int id;
2410     @override
2411     final String name;
2412     @override
2413     final String city;
2414     @override

```

```

2413     final String topic;
2414     @override
2415     final String startDate;
2416     @override
2417     final String endDate;
2418
2419     @override
2420     String toString() {
2421         return 'ConferenceShortResponse(id: $id, name: $name,
            city: $city, topic: $topic, startDate: $startDate,
            endDate: $endDate)';
2422     }
2423
2424     @override
2425     bool operator ==(Object other) {
2426         return identical(this, other) ||
2427             (other.runtimeType == runtimeType &&
2428                 other is _ConferenceShortResponseImpl &&
2429                 (identical(other.id, id) || other.id == id) &&
2430                 (identical(other.name, name) || other.name ==
                    name) &&
2431                 (identical(other.city, city) || other.city ==
                    city) &&
2432                 (identical(other.topic, topic) || other.topic ==
                    topic) &&
2433                 (identical(other.startDate, startDate) ||
                    other.startDate == startDate) &&
2434                 (identical(other.endDate, endDate) || other.
                    endDate == endDate));
2435     }
2436
2437
2438     @JsonKey(ignore: true)
2439     @override
2440     int get hashCode =>
2441         Object.hash(runtimeType, id, name, city, topic,
            startDate, endDate);
2442
2443     @JsonKey(ignore: true)
2444     @override
2445     @pragma('vm:prefer-inline')
2446     _\begin{equation}\begin{gathered}

```

```

ConferenceShortResponseImplCopyWith<
  _$ConferenceShortResponseImpl>
2447   get copyWith => __\end{gathered}\end{equation}
        ConferenceShortResponseImplCopyWithImpl<
2448         _$ConferenceShortResponseImpl>(this , _$identity);
2449
2450   @override
2451   Map<String , dynamic> toJson() {
2452     return _\begin{equation}\begin{gathered}
        ConferenceShortResponseImplToJson(
2453         this ,
2454       );
2455   }
2456 }
2457
2458 abstract class _ConferenceShortResponse implements
        ConferenceShortResponse {
2459   const factory _ConferenceShortResponse(
2460     {required final int id ,
2461     required final String name ,
2462     required final String city ,
2463     required final String topic ,
2464     required final String startDate ,
2465     required final String endDate}) =
        _$ConferenceShortResponseImpl;
2466
2467   factory _ConferenceShortResponse.fromJson(Map<String ,
        dynamic> json) =
2468     _$ConferenceShortResponseImpl.fromJson;
2469
2470   @override
2471   int get id;
2472   @override
2473   String get name;
2474   @override
2475   String get city;
2476   @override
2477   String get topic;
2478   @override
2479   String get startDate;
2480   @override

```

```

2481   String get endDate;
2482   @override
2483   @JsonKey(ignore: true)
2484   _\end{gathered}\end{equation}
        ConferenceShortResponseImplCopyWith<
        _$ConferenceShortResponseImpl>
2485       get copyWith => throw _privateConstructorUsedError;
2486 }
2487
2488 topic.g.dart
2489 // GENERATED CODE – DO NOT MODIFY BY HAND
2490
2491 part of 'topic.dart';
2492
2493 //
        *****
2494 // JsonSerializableGenerator
2495 //
        *****
2496
2497 _$TopicImpl _\begin{equation}\begin{gathered}
    TopicImplFromJson(Map<String, dynamic> json) =>
    _$TopicImpl(
2498       id: (json['id'] as num).toInt(),
2499       name: json['name'] as String,
2500     );
2501
2502 Map<String, dynamic> _\end{gathered}\end{equation}
    TopicImplToJson(_$TopicImpl instance) =>
2503     <String, dynamic>{
2504       'id': instance.id,
2505       'name': instance.name,
2506     };
2507
2508 topic.freezed.dart
2509 // coverage:ignore-file
2510 // GENERATED CODE – DO NOT MODIFY BY HAND
2511 // ignore_for_file: type=lint
2512 // ignore_for_file: unused_element, deprecated_member_use,

```

```

    deprecated_member_use_from_same_package ,
    use_function_type_syntax_for_parameters , unnecessary_const
    , avoid_init_to_null ,
    invalid_override_different_default_values_named ,
    prefer_expression_function_bodies , annotate_overrides ,
    invalid_annotation_target , unnecessary_question_mark
2513
2514 part of 'topic.dart';
2515
2516 //
    *****

2517 // FreezedGenerator
2518 //
    *****

2519
2520 T _$identity<T>(T value) => value;
2521
2522 final _privateConstructorUsedError = UnsupportedError(
2523     'It seems like you constructed your class using 'MyClass.
        _()' '. This constructor is only meant to be used by
        frozen and you are not supposed to need it nor use it
        .\nPlease check the documentation here for more
        information: https://github.com/rrousselGit/frozen#
        adding-getters-and-methods-to-our-models ');
2524
2525 Topic _$TopicFromJson(Map<String , dynamic> json) {
2526     return _Topic.fromJson(json);
2527 }
2528
2529 /// @nodoc
2530 mixin _$Topic {
2531     int get id => throw _privateConstructorUsedError;
2532     String get name => throw _privateConstructorUsedError;
2533
2534     Map<String , dynamic> toJson() => throw
        _privateConstructorUsedError;
2535     @JsonKey(ignore: true)
2536     $TopicCopyWith<Topic> get copyWith => throw
        _privateConstructorUsedError;

```

```

2537 }
2538
2539 /// @nodoc
2540 abstract class $TopicCopyWith<$Res> {
2541     factory $TopicCopyWith(Topic value , $Res Function(Topic)
2542         then) =
2543         _$TopicCopyWithImpl<$Res , Topic>;
2544     @useResult
2545     $Res call({int id , String name});
2546 }
2547
2548 /// @nodoc
2549 class _$TopicCopyWithImpl<$Res , $Val extends Topic>
2550     implements $TopicCopyWith<$Res> {
2551     _$TopicCopyWithImpl(this._value , this._then);
2552
2553     // ignore: unused_field
2554     final $Val _value;
2555     // ignore: unused_field
2556     final $Res Function($Val) _then;
2557
2558     @pragma('vm:prefer-inline ')
2559     @override
2560     $Res call({
2561         Object? id = null ,
2562         Object? name = null ,
2563     }) {
2564         return _then(_value.copyWith(
2565             id: null == id
2566                 ? _value.id
2567                 : id // ignore: cast_nullable_to_non_nullable
2568                     as int ,
2569             name: null == name
2570                 ? _value.name
2571                 : name // ignore: cast_nullable_to_non_nullable
2572                     as String ,
2573         ) as $Val);
2574     }
2575 }
2576 /// @nodoc

```

```

2577 abstract class _\begin{equation}\begin{gathered}
      TopicImplCopyWith<$Res> implements $TopicCopyWith<$Res> {
2578   factory _\end{gathered}\end{equation}TopicImplCopyWith(
2579     _$TopicImpl value, $Res Function(_$TopicImpl) then)
      =
2580     __\begin{equation}\begin{gathered}TopicImplCopyWithImpl
      <$Res>;
2581   @override
2582   @useResult
2583   $Res call({int id, String name});
2584 }
2585
2586 /// @nodoc
2587 class __\end{gathered}\end{equation}TopicImplCopyWithImpl<
      $Res>
2588   extends _$TopicCopyWithImpl<$Res, _$TopicImpl>
2589   implements _\begin{equation}\begin{gathered}
      TopicImplCopyWith<$Res> {
2590     __\end{gathered}\end{equation}TopicImplCopyWithImpl(
2591       _$TopicImpl _value, $Res Function(_$TopicImpl) _then)
2592       : super(_value, _then);
2593
2594   @pragma('vm:prefer-inline')
2595   @override
2596   $Res call({
2597     Object? id = null,
2598     Object? name = null,
2599   }) {
2600     return _then(_$TopicImpl(
2601       id: null == id
2602         ? _value.id
2603         : id // ignore: cast_nullable_to_non_nullable
2604         as int,
2605       name: null == name
2606         ? _value.name
2607         : name // ignore: cast_nullable_to_non_nullable
2608         as String,
2609     ));
2610   }
2611 }
2612

```



```

2613 /// @nodoc
2614 @JsonSerializable()
2615 class _$TopicImpl implements _Topic {
2616     const _$TopicImpl({required this.id, required this.name});
2617
2618     factory _$TopicImpl.fromJson(Map<String, dynamic> json) =>
2619         _\begin{equation}\begin{gathered}TopicImplFromJson(json
2620             );
2621
2622     @override
2623     final int id;
2624     @override
2625     final String name;
2626
2627     @override
2628     String toString() {
2629         return 'Topic(id: $id, name: $name)';
2630     }
2631
2632     @override
2633     bool operator ==(Object other) {
2634         return identical(this, other) ||
2635             (other.runtimeType == runtimeType &&
2636                 other is _$TopicImpl &&
2637                 (identical(other.id, id) || other.id == id) &&
2638                 (identical(other.name, name) || other.name ==
2639                     name));
2640     }
2641
2642     @JsonKey(ignore: true)
2643     @override
2644     int get hashCode => Object.hash(runtimeType, id, name);
2645
2646     @JsonKey(ignore: true)
2647     @override
2648     @pragma('vm:prefer-inline')
2649     _\end{gathered}\end{equation}TopicImplCopyWith<_$TopicImpl>
2650         get copyWith =>
2651         __\begin{equation}\begin{gathered}TopicImplCopyWithImpl
2652             <_$TopicImpl>(this, _$identity);

```

```

2650     @override
2651     Map<String , dynamic> toJson() {
2652         return _\end{gathered}\end{equation}TopicImplToJson(
2653             this ,
2654         );
2655     }
2656 }
2657
2658 abstract class _Topic implements Topic {
2659     const factory _Topic({required final int id , required final
2660         String name}) =
2661         _$TopicImpl;
2662
2663     factory _Topic.fromJson(Map<String , dynamic> json) =
2664         _$TopicImpl.fromJson;
2665
2666     @override
2667     int get id;
2668     @override
2669     String get name;
2670     @override
2671     @JsonKey(ignore: true)
2672     _\begin{equation}\begin{gathered}TopicImplCopyWith<
2673         _$TopicImpl> get copyWith =>
2674         throw _privateConstructorUsedError;
2675 }
2676
2677 conference_request.dart
2678
2679 import 'package:freezed_annotation/freezed_annotation.dart';
2680
2681 part 'conference_request.freezed.dart';
2682 part 'conference_request.g.dart';
2683
2684 @freezed
2685 class ConferenceRequest with _$ConferenceRequest {
2686     const factory ConferenceRequest({
2687         required String name,
2688         required int cityId ,
2689         required int topicId ,
2690         required String startDate ,

```

```

2688     required String endDate ,
2689     required String description ,
2690     required String contacts ,
2691     required String imageUrl ,
2692   }) = _ConferenceRequest;
2693
2694   factory ConferenceRequest.fromJson(Map<String , dynamic>
        json) => _$ConferenceRequestFromJson(json);
2695 }
2696
2697 city.dart
2698
2699 import 'package:freezed_annotation/freezed_annotation.dart';
2700
2701 part 'city.freezed.dart';
2702 part 'city.g.dart';
2703
2704 @freezed
2705 class City with _$City {
2706   const factory City({
2707     required int id ,
2708     required String name ,
2709   }) = _City;
2710
2711   factory City.fromJson(Map<String , dynamic> json) =>
        _$CityFromJson(json);
2712 }
2713
2714 conference_search_options.freezed.dart
2715 // coverage:ignore-file
2716 // GENERATED CODE – DO NOT MODIFY BY HAND
2717 // ignore_for_file: type=lint
2718 // ignore_for_file: unused_element, deprecated_member_use ,
        deprecated_member_use_from_same_package ,
        use_function_type_syntax_for_parameters , unnecessary_const
        , avoid_init_to_null ,
        invalid_override_different_default_values_named ,
        prefer_expression_function_bodies , annotate_overrides ,
        invalid_annotation_target , unnecessary_question_mark
2719
2720 part of 'conference_search_options.dart';

```

```

2721
2722 //
                *****

2723 // FreezedGenerator
2724 //
                *****

2725
2726 T _$identity<T>(T value) => value;
2727
2728 final _privateConstructorUsedError = UnsupportedError(
2729     'It seems like you constructed your class using `MyClass.
        _()`. This constructor is only meant to be used by
        freezed and you are not supposed to need it nor use it
        .\nPlease check the documentation here for more
        information: https://github.com/rrousselGit/freezed#
        adding-getters-and-methods-to-our-models`);

2730
2731 ConferenceSearchOptions _$ConferenceSearchOptionsFromJson(
2732     Map<String, dynamic> json) {
2733     return _ConferenceSearchOptions.fromJson(json);
2734 }
2735
2736 /// @nodoc
2737 mixin _ConferenceSearchOptions {
2738     List<City> get cities => throw _privateConstructorUsedError
        ;
2739     List<Topic> get topics => throw
        _privateConstructorUsedError;

2740
2741     Map<String, dynamic> toJson() => throw
        _privateConstructorUsedError;
2742     @JsonKey(ignore: true)
2743     $ConferenceSearchOptionsCopyWith<ConferenceSearchOptions>
        get copyWith =>
2744         throw _privateConstructorUsedError;
2745 }
2746
2747 /// @nodoc
2748 abstract class $ConferenceSearchOptionsCopyWith<$Res> {

```

```

2749     factory $ConferenceSearchOptionsCopyWith(
2750         ConferenceSearchOptions value ,
2751         $Res Function(ConferenceSearchOptions) then) =
2752         _$ConferenceSearchOptionsCopyWithImpl<$Res ,
2753         ConferenceSearchOptions>;
2754     @useResult
2755     $Res call({List<City> cities , List<Topic> topics});
2756 }
2757 /// @nodoc
2758 class _$ConferenceSearchOptionsCopyWithImpl<$Res ,
2759     $Val extends ConferenceSearchOptions>
2760     implements $ConferenceSearchOptionsCopyWith<$Res> {
2761     _$ConferenceSearchOptionsCopyWithImpl(this._value , this.
2762         _then);
2763
2764     // ignore: unused_field
2765     final $Val _value;
2766     // ignore: unused_field
2767     final $Res Function($Val) _then;
2768
2769     @pragma( 'vm:prefer-inline ' )
2770     @override
2771     $Res call({
2772         Object? cities = null ,
2773         Object? topics = null ,
2774     }) {
2775         return _then(_value.copyWith(
2776             cities: null == cities
2777                 ? _value.cities
2778                 : cities // ignore: cast_nullable_to_non_nullable
2779                 as List<City> ,
2780             topics: null == topics
2781                 ? _value.topics
2782                 : topics // ignore: cast_nullable_to_non_nullable
2783                 as List<Topic> ,
2784         ) as $Val);
2785     }
2786 }
2787 /// @nodoc

```

```

2787 abstract class _\end{gathered}\end{equation}
        ConferenceSearchOptionsImplCopyWith<$Res>
2788 implements $ConferenceSearchOptionsCopyWith<$Res> {
2789 factory _\begin{equation}\begin{gathered}
        ConferenceSearchOptionsImplCopyWith(
2790         _$ConferenceSearchOptionsImpl value ,
2791         $Res Function(_$ConferenceSearchOptionsImpl) then)
        =
2792         __\end{gathered}\end{equation}
        ConferenceSearchOptionsImplCopyWithImpl<$Res>;
2793 @override
2794 @useResult
2795 $Res call({List<City> cities , List<Topic> topics});
2796 }
2797
2798 /// @nodoc
2799 class __\begin{equation}\begin{gathered}
        ConferenceSearchOptionsImplCopyWithImpl<$Res>
2800 extends _$ConferenceSearchOptionsCopyWithImpl<$Res ,
2801         _$ConferenceSearchOptionsImpl>
2802 implements _\end{gathered}\end{equation}
        ConferenceSearchOptionsImplCopyWith<$Res> {
2803 __\begin{equation}\begin{gathered}
        ConferenceSearchOptionsImplCopyWithImpl(
2804         _$ConferenceSearchOptionsImpl _value ,
2805         $Res Function(_$ConferenceSearchOptionsImpl) _then)
2806         : super(_value , _then);
2807
2808 @pragma('vm:prefer-inline')
2809 @override
2810 $Res call({
2811         Object? cities = null ,
2812         Object? topics = null ,
2813 }) {
2814         return _then(_$ConferenceSearchOptionsImpl(
2815                 cities: null == cities
2816                 ? _value._cities
2817                 : cities // ignore: cast_nullable_to_non_nullable
2818                 as List<City> ,
2819                 topics: null == topics
2820                 ? _value._topics

```

```

2821         : topics // ignore: cast_nullable_to_non_nullable
2822           as List<Topic>,
2823     ));
2824   }
2825 }
2826
2827 /// @nodoc
2828 @JsonSerializable()
2829 class _$ConferenceSearchOptionsImpl implements
    _ConferenceSearchOptions {
2830   const _$ConferenceSearchOptionsImpl(
2831     {required final List<City> cities , required final List<
        Topic> topics })
2832     : _cities = cities ,
2833       _topics = topics;
2834
2835   factory _$ConferenceSearchOptionsImpl.fromJson(Map<String ,
        dynamic> json) =>
2836     _\end{gathered}\end{equation}
        ConferenceSearchOptionsImplFromJson(json);
2837
2838   final List<City> _cities;
2839   @override
2840   List<City> get cities {
2841     if (_cities is EqualUnmodifiableListView) return _cities;
2842     // ignore: implicit_dynamic_type
2843     return EqualUnmodifiableListView(_cities);
2844   }
2845
2846   final List<Topic> _topics;
2847   @override
2848   List<Topic> get topics {
2849     if (_topics is EqualUnmodifiableListView) return _topics;
2850     // ignore: implicit_dynamic_type
2851     return EqualUnmodifiableListView(_topics);
2852   }
2853
2854   @override
2855   String toString() {
2856     return 'ConferenceSearchOptions(cities: $cities , topics:
        $topics)';

```

```

2857 }
2858
2859 @override
2860 bool operator ==(Object other) {
2861     return identical(this, other) ||
2862         (other.runtimeType == runtimeType &&
2863             other is _$ConferenceSearchOptionsImpl &&
2864             const DeepCollectionEquality().equals(other._cities, _cities) &&
2865             const DeepCollectionEquality().equals(other._topics, _topics));
2866 }
2867
2868 @JsonKey(ignore: true)
2869 @override
2870 int get hashCode => Object.hash(
2871     runtimeType,
2872     const DeepCollectionEquality().hash(_cities),
2873     const DeepCollectionEquality().hash(_topics));
2874
2875 @JsonKey(ignore: true)
2876 @override
2877 @pragma('vm:prefer-inline')
2878 _\begin{equation}\begin{gathered}
2879     ConferenceSearchOptionsImplCopyWith<
2880         _$ConferenceSearchOptionsImpl>
2881         get copyWith => __\end{gathered}\end{equation}
2882         ConferenceSearchOptionsImplCopyWithImpl<
2883             _$ConferenceSearchOptionsImpl>(this, _$identity);
2884
2885 @override
2886 Map<String, dynamic> toJson() {
2887     return _\begin{equation}\begin{gathered}
2888         ConferenceSearchOptionsImplToJson(
2889             this,
2890         );
2891     }
2892 }
2893
2894 abstract class _ConferenceSearchOptions implements
2895     ConferenceSearchOptions {

```



```

2891     const factory _ConferenceSearchOptions(
2892         {required final List<City> cities ,
2893         required final List<Topic> topics}) =
2894         _$ConferenceSearchOptionsImpl;
2895
2896     factory _ConferenceSearchOptions.fromJson(Map<String ,
2897         dynamic> json) =
2898         _$ConferenceSearchOptionsImpl.fromJson;
2899
2900     @override
2901     List<City> get cities;
2902     @override
2903     List<Topic> get topics;
2904     @override
2905     @JsonKey(ignore: true)
2906     _\end{gathered}\end{equation}
2907         ConferenceSearchOptionsImplCopyWith<
2908         _$ConferenceSearchOptionsImpl>
2909         get copyWith => throw _privateConstructorUsedError;
2910 }
2911
2912 topic.dart
2913 import 'package:freezed_annotation/freezed_annotation.dart';
2914
2915 part 'topic.freezed.dart';
2916 part 'topic.g.dart';
2917
2918 @freezed
2919 class Topic with _$Topic {
2920     const factory Topic({
2921         required int id,
2922         required String name,
2923     }) = _Topic;
2924
2925     factory Topic.fromJson(Map<String , dynamic> json) =>
2926         _$TopicFromJson(json);
2927 }
2928
2929 conference_updating_request.g.dart
2930 // GENERATED CODE – DO NOT MODIFY BY HAND
2931

```

```

2927 part of 'conference_updating_request.dart';
2928
2929 //
    *****

2930 // JsonSerializerGenerator
2931 //
    *****

2932
2933 _$ConferenceUpdatingRequestImpl _\begin{equation}\begin{
gathered}ConferenceUpdatingRequestImplFromJson(
2934     Map<String , dynamic> json) =>
2935     _$ConferenceUpdatingRequestImpl(
2936         name: json['name'] as String?,
2937         cityId: (json['cityId'] as num?)?.toInt(),
2938         topicId: (json['topicId'] as num?)?.toInt(),
2939         startDate: json['startDate'] as String?,
2940         endDate: json['endDate'] as String?,
2941         description: json['description'] as String?,
2942         contacts: json['contacts'] as String?,
2943         imageUrl: json['imageUrl'] as String?,
2944     );
2945
2946 Map<String , dynamic> _\end{gathered}\end{equation}
ConferenceUpdatingRequestImplToJson(
2947     _$ConferenceUpdatingRequestImpl instance) =>
2948     <String , dynamic>{
2949         'name': instance.name,
2950         'cityId': instance.cityId,
2951         'topicId': instance.topicId,
2952         'startDate': instance.startDate,
2953         'endDate': instance.endDate,
2954         'description': instance.description,
2955         'contacts': instance.contacts,
2956         'imageUrl': instance.imageUrl,
2957     };
2958
2959 conference_short_response.g.dart
2960 // GENERATED CODE – DO NOT MODIFY BY HAND
2961

```

```

2962 part of 'conference_short_response.dart';
2963
2964 //
    *****

2965 // JsonSerializerGenerator
2966 //
    *****

2967
2968 _$ConferenceShortResponseImpl _\begin{equation}\begin{
gathered}ConferenceShortResponseImplFromJson(
2969     Map<String , dynamic> json) =>
2970     _$ConferenceShortResponseImpl(
2971         id: (json['id'] as num).toInt(),
2972         name: json['name'] as String ,
2973         city: json['city'] as String ,
2974         topic: json['topic'] as String ,
2975         startDate: json['startDate'] as String ,
2976         endDate: json['endDate'] as String ,
2977     );
2978
2979 Map<String , dynamic> _\end{gathered}\end{equation}
ConferenceShortResponseImplToJson(
2980     _$ConferenceShortResponseImpl instance) =>
2981     <String , dynamic>{
2982         'id': instance.id ,
2983         'name': instance.name ,
2984         'city': instance.city ,
2985         'topic': instance.topic ,
2986         'startDate': instance.startDate ,
2987         'endDate': instance.endDate ,
2988     };
2989
2990 city.g.dart
2991 // GENERATED CODE – DO NOT MODIFY BY HAND
2992
2993 part of 'city.dart';
2994
2995 //
    *****

```

```

2996 // JsonSerializerGenerator
2997 //
    *****

2998
2999 _$CityImpl _\begin{equation}\begin{gathered}CityImplFromJson(
    Map<String , dynamic> json) => _$CityImpl(
3000     id: (json['id'] as num).toInt() ,
3001     name: json['name'] as String ,
3002 );
3003
3004 Map<String , dynamic> _\end{gathered}\end{equation}
    CityImplToJson(_$CityImpl instance) =>
3005     <String , dynamic>{
3006         'id': instance.id ,
3007         'name': instance.name ,
3008     };
3009
3010 conference_detail_response.g.dart
3011 // GENERATED CODE – DO NOT MODIFY BY HAND
3012
3013 part of 'conference_detail_response.dart';
3014
3015 //
    *****

3016 // JsonSerializerGenerator
3017 //
    *****

3018
3019 _$ConferenceDetailResponseImpl _\begin{equation}\begin{gathered}ConferenceDetailResponseImplFromJson(
    Map<String , dynamic> json) =>
3020     _$ConferenceDetailResponseImpl(
3021         id: (json['id'] as num).toInt() ,
3022         name: json['name'] as String ,
3023         city: json['city'] as String ,
3024         topic: json['topic'] as String ,
3025         startDate: json['startDate'] as String ,

```

```

3027         endDate: json['endDate'] as String ,
3028         description: json['description'] as String ,
3029         contacts: json['contacts'] as String ,
3030         canRedact: json['canRedact'] as bool ,
3031     );
3032
3033 Map<String , dynamic> _\end{gathered}\end{equation}
    ConferenceDetailResponseImplToJson(
3034         _$ConferenceDetailResponseImpl instance) =>
3035     <String , dynamic>{
3036         'id': instance.id ,
3037         'name': instance.name ,
3038         'city': instance.city ,
3039         'topic': instance.topic ,
3040         'startDate': instance.startDate ,
3041         'endDate': instance.endDate ,
3042         'description': instance.description ,
3043         'contacts': instance.contacts ,
3044         'canRedact': instance.canRedact ,
3045     };
3046
3047 conference_request.freezed.dart
3048 // coverage:ignore-file
3049 // GENERATED CODE – DO NOT MODIFY BY HAND
3050 // ignore_for_file: type=lint
3051 // ignore_for_file: unused_element , deprecated_member_use ,
    deprecated_member_use_from_same_package ,
    use_function_type_syntax_for_parameters , unnecessary_const
    , avoid_init_to_null ,
    invalid_override_different_default_values_named ,
    prefer_expression_function_bodies , annotate_overrides ,
    invalid_annotation_target , unnecessary_question_mark
3052
3053 part of 'conference_request.dart';
3054
3055 //
    *****
3056 // FreezedGenerator
3057 //
    *****

```

```

3058
3059 T _$identity<T>(T value) => value;
3060
3061 final _privateConstructorUsedError = UnsupportedError(
3062     'It seems like you constructed your class using `MyClass.
        _()`. This constructor is only meant to be used by
        freezeD and you are not supposed to need it nor use it
        .\nPlease check the documentation here for more
        information: https://github.com/rrousselGit/freezeD#
        adding-getters-and-methods-to-our-models`);
3063
3064 ConferenceRequest _$ConferenceRequestFromJson(Map<String ,
        dynamic> json) {
3065     return _ConferenceRequest.fromJson(json);
3066 }
3067
3068 /// @nodoc
3069 mixin _$ConferenceRequest {
3070     String get name => throw _privateConstructorUsedError;
3071     int get cityId => throw _privateConstructorUsedError;
3072     int get topicId => throw _privateConstructorUsedError;
3073     String get startDate => throw _privateConstructorUsedError;
3074     String get endDate => throw _privateConstructorUsedError;
3075     String get description => throw
        _privateConstructorUsedError;
3076     String get contacts => throw _privateConstructorUsedError;
3077     String get imageUrl => throw _privateConstructorUsedError;
3078
3079     Map<String , dynamic> toJson() => throw
        _privateConstructorUsedError;
3080     @JsonKey(ignore: true)
3081     $ConferenceRequestCopyWith<ConferenceRequest> get copyWith
        =>
3082         throw _privateConstructorUsedError;
3083 }
3084
3085 /// @nodoc
3086 abstract class $ConferenceRequestCopyWith<$Res> {
3087     factory $ConferenceRequestCopyWith(
3088         ConferenceRequest value , $Res Function(

```

```

ConferenceRequest) then) =
3089     _$ConferenceRequestCopyWithImpl<$Res, ConferenceRequest
        >;
3090     @useResult
3091     $Res call(
3092         {String name,
3093          int cityId,
3094          int topicId,
3095          String startDate,
3096          String endDate,
3097          String description,
3098          String contacts,
3099          String imageUrl});
3100 }
3101
3102 /// @nodoc
3103 class _$ConferenceRequestCopyWithImpl<$Res, $Val extends
    ConferenceRequest>
3104     implements $ConferenceRequestCopyWith<$Res> {
3105     _$ConferenceRequestCopyWithImpl(this._value, this._then);
3106
3107     // ignore: unused_field
3108     final $Val _value;
3109     // ignore: unused_field
3110     final $Res Function($Val) _then;
3111
3112     @pragma('vm:prefer-inline')
3113     @override
3114     $Res call({
3115         Object? name = null,
3116         Object? cityId = null,
3117         Object? topicId = null,
3118         Object? startDate = null,
3119         Object? endDate = null,
3120         Object? description = null,
3121         Object? contacts = null,
3122         Object? imageUrl = null,
3123     }) {
3124         return _then(_value.copyWith(
3125             name: null == name
3126                 ? _value.name

```

```

3127         : name // ignore: cast_nullable_to_non_nullable
3128             as String ,
3129         cityId: null == cityId
3130             ? _value.cityId
3131             : cityId // ignore: cast_nullable_to_non_nullable
3132                 as int ,
3133         topicId: null == topicId
3134             ? _value.topicId
3135             : topicId // ignore: cast_nullable_to_non_nullable
3136                 as int ,
3137         startDate: null == startDate
3138             ? _value.startDate
3139             : startDate // ignore:
3140                 cast_nullable_to_non_nullable
3141                 as String ,
3142         endDate: null == endDate
3143             ? _value.endDate
3144             : endDate // ignore: cast_nullable_to_non_nullable
3145                 as String ,
3146         description: null == description
3147             ? _value.description
3148             : description // ignore:
3149                 cast_nullable_to_non_nullable
3150                 as String ,
3151         contacts: null == contacts
3152             ? _value.contacts
3153             : contacts // ignore: cast_nullable_to_non_nullable
3154                 as String ,
3155         imageUrl: null == imageUrl
3156             ? _value.imageUrl
3157             : imageUrl // ignore: cast_nullable_to_non_nullable
3158                 as String ,
3159     ) as $Val);
3160 }
3161 }
3162 /// @nodoc
3163 abstract class _\begin{equation}\begin{gathered}
3164     ConferenceRequestImplCopyWith<$Res>
3165     implements $ConferenceRequestCopyWith<$Res> {
3166     factory _\end{gathered}\end{equation}

```



```

        ConferenceRequestImplCopyWith(_$ConferenceRequestImpl
value ,
3165         $Res Function(_$ConferenceRequestImpl) then) =
3166         __\begin{equation}\begin{gathered}
            ConferenceRequestImplCopyWithImpl<$Res>;
3167     @override
3168     @useResult
3169     $Res call(
3170         {String name,
3171         int cityId ,
3172         int topicId ,
3173         String startDate ,
3174         String endDate ,
3175         String description ,
3176         String contacts ,
3177         String imageUrl});
3178 }
3179
3180 /// @nodoc
3181 class __\end{gathered}\end{equation}
    ConferenceRequestImplCopyWithImpl<$Res>
3182     extends _$ConferenceRequestCopyWithImpl<$Res ,
        _$ConferenceRequestImpl>
3183     implements _\begin{equation}\begin{gathered}
        ConferenceRequestImplCopyWith<$Res> {
3184     __\end{gathered}\end{equation}
        ConferenceRequestImplCopyWithImpl(
            _$ConferenceRequestImpl _value ,
3185            $Res Function(_$ConferenceRequestImpl) _then)
3186            : super(_value , _then);
3187
3188     @pragma('vm:prefer-inline ')
3189     @override
3190     $Res call({
3191         Object? name = null ,
3192         Object? cityId = null ,
3193         Object? topicId = null ,
3194         Object? startDate = null ,
3195         Object? endDate = null ,
3196         Object? description = null ,
3197         Object? contacts = null ,

```

```

3198     Object? imageUrl = null ,
3199 )) {
3200     return _then(_$ConferenceRequestImpl(
3201         name: null == name
3202             ? _value.name
3203             : name // ignore: cast_nullable_to_non_nullable
3204                 as String ,
3205         cityId: null == cityId
3206             ? _value.cityId
3207             : cityId // ignore: cast_nullable_to_non_nullable
3208                 as int ,
3209         topicId: null == topicId
3210             ? _value.topicId
3211             : topicId // ignore: cast_nullable_to_non_nullable
3212                 as int ,
3213         startDate: null == startDate
3214             ? _value.startDate
3215             : startDate // ignore:
3216                 cast_nullable_to_non_nullable
3217                 as String ,
3218         endDate: null == endDate
3219             ? _value.endDate
3220             : endDate // ignore: cast_nullable_to_non_nullable
3221                 as String ,
3222         description: null == description
3223             ? _value.description
3224             : description // ignore:
3225                 cast_nullable_to_non_nullable
3226                 as String ,
3227         contacts: null == contacts
3228             ? _value.contacts
3229             : contacts // ignore: cast_nullable_to_non_nullable
3230                 as String ,
3231         imageUrl: null == imageUrl
3232             ? _value.imageUrl
3233             : imageUrl // ignore: cast_nullable_to_non_nullable
3234                 as String ,
3235     ));
3236 }

```

```

3237 /// @nodoc
3238 @JsonSerializable()
3239 class _$ConferenceRequestImpl implements _ConferenceRequest {
3240     const _$ConferenceRequestImpl(
3241         {required this.name,
3242          required this.cityId,
3243          required this.topicId,
3244          required this.startDate,
3245          required this.endDate,
3246          required this.description,
3247          required this.contacts,
3248          required this.imageUrl});
3249
3250     factory _$ConferenceRequestImpl.fromJson(Map<String,
3251         dynamic> json) =>
3252         _\begin{equation}\begin{gathered}
3253             ConferenceRequestImplFromJson(json);
3254
3255     @override
3256     final String name;
3257     @override
3258     final int cityId;
3259     @override
3260     final int topicId;
3261     @override
3262     final String startDate;
3263     @override
3264     final String endDate;
3265     @override
3266     final String description;
3267     @override
3268     final String contacts;
3269     @override
3270     final String imageUrl;
3271
3272     @override
3273     String toString() {
3274         return 'ConferenceRequest(name: $name, cityId: $cityId,
3275             topicId: $topicId, startDate: $startDate, endDate:
3276             $endDate, description: $description, contacts:
3277             $contacts, imageUrl: $imageUrl)';

```

```

3273 }
3274
3275 @override
3276 bool operator ==(Object other) {
3277     return identical(this, other) ||
3278         (other.runtimeType == runtimeType &&
3279             other is _$ConferenceRequestImpl &&
3280             (identical(other.name, name) || other.name ==
3281                 name) &&
3282             (identical(other.cityId, cityId) || other.cityId
3283                 == cityId) &&
3284             (identical(other.topicId, topicId) || other.
3285                 topicId == topicId) &&
3286             (identical(other.startDate, startDate) ||
3287                 other.startDate == startDate) &&
3288             (identical(other.endDate, endDate) || other.
3289                 endDate == endDate) &&
3290             (identical(other.description, description) ||
3291                 other.description == description) &&
3292             (identical(other.contacts, contacts) ||
3293                 other.contacts == contacts) &&
3294             (identical(other.imageUrl, imageUrl) ||
3295                 other.imageUrl == imageUrl));
3296 }
3297
3298 @JsonKey(ignore: true)
3299 @override
3300 int get hashCode => Object.hash(runtimeType, name, cityId,
3301     topicId, startDate,
3302     endDate, description, contacts, imageUrl);
3303
3304 @JsonKey(ignore: true)
3305 @override
3306 @pragma('vm:prefer-inline')
3307 _\end{gathered}\end{equation}ConferenceRequestImplCopyWith<
3308     _$ConferenceRequestImpl> get copyWith =>
3309     __\begin{equation}\begin{gathered}
3310         ConferenceRequestImplCopyWithImpl<
3311             _$ConferenceRequestImpl>(<
3312             this, _$identity);
3313 
```

```

3306     @override
3307     Map<String , dynamic> toJson() {
3308         return _\end{gathered}\end{equation}
3309             ConferenceRequestImplToJson(
3310                 this ,
3311             );
3312     }
3313 }
3314 abstract class _ConferenceRequest implements
3315     ConferenceRequest {
3316     const factory _ConferenceRequest(
3317         {required final String name,
3318         required final int cityId ,
3319         required final int topicId ,
3320         required final String startDate ,
3321         required final String endDate ,
3322         required final String description ,
3323         required final String contacts ,
3324         required final String imageUrl}) =
3325         _$ConferenceRequestImpl;
3326
3327     factory _ConferenceRequest.fromJson(Map<String , dynamic>
3328         json) =
3329         _$ConferenceRequestImpl.fromJson;
3330
3331     @override
3332     String get name;
3333     @override
3334     int get cityId;
3335     @override
3336     int get topicId;
3337     @override
3338     String get startDate;
3339     @override
3340     String get endDate;
3341     @override
3342     String get description;
3343     @override
3344     String get contacts;
3345     @override

```

```

3343   String get imageUrl;
3344   @override
3345   @JsonKey(ignore: true)
3346   _\begin{equation}\begin{gathered}
        ConferenceRequestImplCopyWith<_$_ConferenceRequestImpl>
        get copyWith =>
3347       throw _privateConstructorUsedError;
3348   }
3349
3350   conference_search_options.dart
3351   import 'package:freezed_annotation/freezed_annotation.dart';
3352
3353   import 'city.dart';
3354   import 'topic.dart';
3355
3356   part 'conference_search_options.freezed.dart';
3357   part 'conference_search_options.g.dart';
3358
3359   @freezed
3360   class ConferenceSearchOptions with _$_ConferenceSearchOptions
        {
3361     const factory ConferenceSearchOptions({
3362       required List<City> cities ,
3363       required List<Topic> topics ,
3364     }) = _ConferenceSearchOptions;
3365
3366     factory ConferenceSearchOptions.fromJson(Map<String ,
        dynamic> json) =>
3367         _$_ConferenceSearchOptionsFromJson(json);
3368   }
3369
3370   city.freezed.dart
3371   // coverage:ignore-file
3372   // GENERATED CODE – DO NOT MODIFY BY HAND
3373   // ignore_for_file: type=lint
3374   // ignore_for_file: unused_element, deprecated_member_use ,
        deprecated_member_use_from_same_package ,
        use_function_type_syntax_for_parameters , unnecessary_const
        , avoid_init_to_null ,
        invalid_override_different_default_values_named ,
        prefer_expression_function_bodies , annotate_overrides ,

```

```

invalid_annotation_target , unnecessary_question_mark
3375
3376 part of 'city.dart';
3377
3378 //
    *****

3379 // FreezedGenerator
3380 //
    *****

3381
3382 T _$identity<T>(T value) => value;
3383
3384 final _privateConstructorUsedError = UnsupportedError(
3385     'It seems like you constructed your class using 'MyClass.
        _()' '. This constructor is only meant to be used by
        freedzed and you are not supposed to need it nor use it
        .\nPlease check the documentation here for more
        information: https://github.com/rrousselGit/freedzed#
        adding-getters-and-methods-to-our-models ');
3386
3387 City _$CityFromJson(Map<String , dynamic> json) {
3388     return _City.fromJson(json);
3389 }
3390
3391 /// @nodoc
3392 mixin _$City {
3393     int get id => throw _privateConstructorUsedError;
3394     String get name => throw _privateConstructorUsedError;
3395
3396     Map<String , dynamic> toJson() => throw
        _privateConstructorUsedError;
3397     @JsonKey(ignore: true)
3398     $CityCopyWith<City> get copyWith => throw
        _privateConstructorUsedError;
3399 }
3400
3401 /// @nodoc
3402 abstract class $CityCopyWith<$Res> {
3403     factory $CityCopyWith(City value , $Res Function(City) then)

```

```

=
3404     _$CityCopyWithImpl<$Res, City>;
3405     @useResult
3406     $Res call({int id, String name});
3407 }
3408
3409 /// @nodoc
3410 class _$CityCopyWithImpl<$Res, $Val extends City>
3411     implements $CityCopyWith<$Res> {
3412     _$CityCopyWithImpl(this._value, this._then);
3413
3414     // ignore: unused_field
3415     final $Val _value;
3416     // ignore: unused_field
3417     final $Res Function($Val) _then;
3418
3419     @pragma('vm:prefer-inline')
3420     @override
3421     $Res call({
3422         Object? id = null,
3423         Object? name = null,
3424     }) {
3425         return _then(_value.copyWith(
3426             id: null == id
3427                 ? _value.id
3428                 : id // ignore: cast_nullable_to_non_nullable
3429                 as int,
3430             name: null == name
3431                 ? _value.name
3432                 : name // ignore: cast_nullable_to_non_nullable
3433                 as String,
3434         ) as $Val);
3435     }
3436 }
3437
3438 /// @nodoc
3439 abstract class _\end{gathered}\end{equation}CityImplCopyWith<
    $Res> implements $CityCopyWith<$Res> {
3440     factory _\begin{equation}\begin{gathered}CityImplCopyWith(
3441         _$CityImpl value, $Res Function(_$CityImpl) then) =
3442         __\end{gathered}\end{equation}CityImplCopyWithImpl<$Res

```



```

>;
3443 @override
3444 @useResult
3445 $Res call({int id, String name});
3446 }
3447
3448 /// @nodoc
3449 class __\begin{equation}\begin{gathered}CityImplCopyWithImpl<
    $Res>
3450     extends _$CityCopyWithImpl<$Res, _$CityImpl>
3451     implements __\end{gathered}\end{equation}CityImplCopyWith<
        $Res> {
3452     __\begin{equation}\begin{gathered}CityImplCopyWithImpl(
        _$CityImpl _value, $Res Function(_$CityImpl) _then)
3453         : super(_value, _then);
3454
3455     @pragma('vm:prefer-inline')
3456     @override
3457     $Res call({
3458         Object? id = null,
3459         Object? name = null,
3460     }) {
3461         return _then(_$CityImpl(
3462             id: null == id
3463                 ? _value.id
3464                 : id // ignore: cast_nullable_to_non_nullable
3465                     as int,
3466             name: null == name
3467                 ? _value.name
3468                 : name // ignore: cast_nullable_to_non_nullable
3469                     as String,
3470         ));
3471     }
3472 }
3473
3474 /// @nodoc
3475 @JsonSerializable()
3476 class _$CityImpl implements _City {
3477     const _$CityImpl({required this.id, required this.name});
3478
3479     factory _$CityImpl.fromJson(Map<String, dynamic> json) =>

```

```

3480         _\end{gathered}\end{equation} CityImplFromJson(json);
3481
3482     @override
3483     final int id;
3484     @override
3485     final String name;
3486
3487     @override
3488     String toString() {
3489         return 'City(id: $id, name: $name)';
3490     }
3491
3492     @override
3493     bool operator ==(Object other) {
3494         return identical(this, other) ||
3495             (other.runtimeType == runtimeType &&
3496                 other is _$CityImpl &&
3497                 (identical(other.id, id) || other.id == id) &&
3498                 (identical(other.name, name) || other.name ==
3499                     name));
3500     }
3501
3502     @JsonKey(ignore: true)
3503     @override
3504     int get hashCode => Object.hash(runtimeType, id, name);
3505
3506     @JsonKey(ignore: true)
3507     @override
3508     @pragma('vm:prefer-inline')
3509     _\begin{equation}\begin{gathered} CityImplCopyWith<
3510         _$CityImpl> get copyWith =>
3511         __\end{gathered}\end{equation} CityImplCopyWithImpl<
3512         _$CityImpl>(this, _$identity);
3513
3514     @override
3515     Map<String, dynamic> toJson() {
3516         return _\begin{equation}\begin{gathered} CityImplToJson(
3517             this,
3518         );
3519     }
3520 }

```

```

3518
3519 abstract class _City implements City {
3520     const factory _City({required final int id, required final
        String name}) =
3521         _$CityImpl;
3522
3523     factory _City.fromJson(Map<String, dynamic> json) =
        _$CityImpl.fromJson;
3524
3525     @override
3526     int get id;
3527     @override
3528     String get name;
3529     @override
3530     @JsonKey(ignore: true)
3531     _\end{gathered}\end{equation} CityImplCopyWith<_$CityImpl>
        get copyWith =>
3532         throw _privateConstructorUsedError;
3533 }
3534
3535 conference_request.g.dart
3536 // GENERATED CODE – DO NOT MODIFY BY HAND
3537
3538 part of 'conference_request.dart';
3539
3540 //
        *****
3541 // JsonSerializerGenerator
3542 //
        *****
3543
3544 _$ConferenceRequestImpl _\begin{equation}\begin{gathered}
    ConferenceRequestImplFromJson(
3545         Map<String, dynamic> json) =>
3546         _$ConferenceRequestImpl(
3547             name: json['name'] as String,
3548             cityId: (json['cityId'] as num).toInt(),
3549             topicId: (json['topicId'] as num).toInt(),
3550             startDate: json['startDate'] as String,

```

```

3551         endDate: json['endDate'] as String ,
3552         description: json['description'] as String ,
3553         contacts: json['contacts'] as String ,
3554         imageUrl: json['imageUrl'] as String ,
3555     );
3556
3557 Map<String , dynamic> _\end{gathered}\end{equation}
    ConferenceRequestImplToJson(
3558     _$ConferenceRequestImpl instance) =>
3559     <String , dynamic>{
3560         'name': instance.name ,
3561         'cityId': instance.cityId ,
3562         'topicId': instance.topicId ,
3563         'startDate': instance.startDate ,
3564         'endDate': instance.endDate ,
3565         'description': instance.description ,
3566         'contacts': instance.contacts ,
3567         'imageUrl': instance.imageUrl ,
3568     };
3569
3570 conference_detail_response.freezed.dart
3571 // coverage:ignore-file
3572 // GENERATED CODE – DO NOT MODIFY BY HAND
3573 // ignore_for_file: type=lint
3574 // ignore_for_file: unused_element , deprecated_member_use ,
    deprecated_member_use_from_same_package ,
    use_function_type_syntax_for_parameters , unnecessary_const
    , avoid_init_to_null ,
    invalid_override_different_default_values_named ,
    prefer_expression_function_bodies , annotate_overrides ,
    invalid_annotation_target , unnecessary_question_mark
3575
3576 part of 'conference_detail_response.dart';
3577
3578 //
    *****
3579 // FreezedGenerator
3580 //
    *****

```

```

3581
3582 T _$identity<T>(T value) => value;
3583
3584 final _privateConstructorUsedError = UnsupportedError(
3585     'It seems like you constructed your class using `MyClass.
        _()`. This constructor is only meant to be used by
        freezed and you are not supposed to need it nor use it
        .\nPlease check the documentation here for more
        information: https://github.com/rrousselGit/freezed#
        adding-getters-and-methods-to-our-models`);
3586
3587 ConferenceDetailResponse _$ConferenceDetailResponseFromJson(
3588     Map<String, dynamic> json) {
3589     return _ConferenceDetailResponse.fromJson(json);
3590 }
3591
3592 /// @nodoc
3593 mixin _$ConferenceDetailResponse {
3594     int get id => throw _privateConstructorUsedError;
3595     String get name => throw _privateConstructorUsedError;
3596     String get city => throw _privateConstructorUsedError;
3597     String get topic => throw _privateConstructorUsedError;
3598     String get startDate => throw _privateConstructorUsedError;
3599     String get endDate => throw _privateConstructorUsedError;
3600     String get description => throw
        _privateConstructorUsedError;
3601     String get contacts => throw _privateConstructorUsedError;
3602     bool get canRedact => throw _privateConstructorUsedError;
3603
3604     Map<String, dynamic> toJson() => throw
        _privateConstructorUsedError;
3605     @JsonKey(ignore: true)
3606     $ConferenceDetailResponseCopyWith<ConferenceDetailResponse>
        get copyWith =>
3607         throw _privateConstructorUsedError;
3608 }
3609
3610 /// @nodoc
3611 abstract class $ConferenceDetailResponseCopyWith<$Res> {
3612     factory $ConferenceDetailResponseCopyWith(
        ConferenceDetailResponse value,

```

```

3613         $Res Function(ConferenceDetailResponse) then) =
3614         _$ConferenceDetailResponseCopyWithImpl<$Res,
           ConferenceDetailResponse>;
3615     @useResult
3616     $Res call(
3617         {int id,
3618         String name,
3619         String city,
3620         String topic,
3621         String startDate,
3622         String endDate,
3623         String description,
3624         String contacts,
3625         bool canRedact});
3626 }
3627
3628 /// @nodoc
3629 class _$ConferenceDetailResponseCopyWithImpl<$Res,
3630         $Val extends ConferenceDetailResponse>
3631     implements $ConferenceDetailResponseCopyWith<$Res> {
3632     _$ConferenceDetailResponseCopyWithImpl(this._value, this.
        _then);
3633
3634     // ignore: unused_field
3635     final $Val _value;
3636     // ignore: unused_field
3637     final $Res Function($Val) _then;
3638
3639     @pragma('vm:prefer-inline')
3640     @override
3641     $Res call({
3642         Object? id = null,
3643         Object? name = null,
3644         Object? city = null,
3645         Object? topic = null,
3646         Object? startDate = null,
3647         Object? endDate = null,
3648         Object? description = null,
3649         Object? contacts = null,
3650         Object? canRedact = null,
3651     }) {

```

```

3652     return _then(_value.copyWith(
3653         id: null == id
3654             ? _value.id
3655             : id // ignore: cast_nullable_to_non_nullable
3656             as int ,
3657         name: null == name
3658             ? _value.name
3659             : name // ignore: cast_nullable_to_non_nullable
3660             as String ,
3661         city: null == city
3662             ? _value.city
3663             : city // ignore: cast_nullable_to_non_nullable
3664             as String ,
3665         topic: null == topic
3666             ? _value.topic
3667             : topic // ignore: cast_nullable_to_non_nullable
3668             as String ,
3669         startDate: null == startDate
3670             ? _value.startDate
3671             : startDate // ignore:
3672                 cast_nullable_to_non_nullable
3673                 as String ,
3674         endDate: null == endDate
3675             ? _value.endDate
3676             : endDate // ignore: cast_nullable_to_non_nullable
3677             as String ,
3678         description: null == description
3679             ? _value.description
3680             : description // ignore:
3681                 cast_nullable_to_non_nullable
3682                 as String ,
3683         contacts: null == contacts
3684             ? _value.contacts
3685             : contacts // ignore: cast_nullable_to_non_nullable
3686             as String ,
3687         canRedact: null == canRedact
3688             ? _value.canRedact
3689             : canRedact // ignore:
3690                 cast_nullable_to_non_nullable
3691                 as bool ,
3692     ) as $Val);

```

```

3690     }
3691 }
3692
3693 /// @nodoc
3694 abstract class _\begin{equation}\begin{gathered}
    ConferenceDetailResponseImplCopyWith<$Res>
3695     implements $ConferenceDetailResponseCopyWith<$Res> {
3696     factory _\end{gathered}\end{equation}
        ConferenceDetailResponseImplCopyWith(
3697         _$ConferenceDetailResponseImpl value ,
3698         $Res Function(_$ConferenceDetailResponseImpl) then)
        =
3699         __\begin{equation}\begin{gathered}
            ConferenceDetailResponseImplCopyWithImpl<$Res>;
3700     @override
3701     @useResult
3702     $Res call(
3703         {int id ,
3704         String name ,
3705         String city ,
3706         String topic ,
3707         String startDate ,
3708         String endDate ,
3709         String description ,
3710         String contacts ,
3711         bool canRedact});
3712 }
3713
3714 /// @nodoc
3715 class __\end{gathered}\end{equation}
    ConferenceDetailResponseImplCopyWithImpl<$Res>
3716     extends _$ConferenceDetailResponseCopyWithImpl<$Res ,
3717         _$ConferenceDetailResponseImpl>
3718     implements _\begin{equation}\begin{gathered}
        ConferenceDetailResponseImplCopyWith<$Res> {
3719     __\end{gathered}\end{equation}
        ConferenceDetailResponseImplCopyWithImpl(
3720         _$ConferenceDetailResponseImpl _value ,
3721         $Res Function(_$ConferenceDetailResponseImpl) _then)
3722         : super(_value , _then);
3723

```



```

3724 @pragma( 'vm:prefer-inline ' )
3725 @override
3726 $Res call({
3727     Object? id = null ,
3728     Object? name = null ,
3729     Object? city = null ,
3730     Object? topic = null ,
3731     Object? startDate = null ,
3732     Object? endDate = null ,
3733     Object? description = null ,
3734     Object? contacts = null ,
3735     Object? canRedact = null ,
3736 }) {
3737     return _then(_$ConferenceDetailResponseImpl(
3738         id: null == id
3739             ? _value.id
3740             : id // ignore: cast_nullable_to_non_nullable
3741                 as int ,
3742         name: null == name
3743             ? _value.name
3744             : name // ignore: cast_nullable_to_non_nullable
3745                 as String ,
3746         city: null == city
3747             ? _value.city
3748             : city // ignore: cast_nullable_to_non_nullable
3749                 as String ,
3750         topic: null == topic
3751             ? _value.topic
3752             : topic // ignore: cast_nullable_to_non_nullable
3753                 as String ,
3754         startDate: null == startDate
3755             ? _value.startDate
3756             : startDate // ignore:
3757                 cast_nullable_to_non_nullable
3758                 as String ,
3759         endDate: null == endDate
3760             ? _value.endDate
3761             : endDate // ignore: cast_nullable_to_non_nullable
3762                 as String ,
3763         description: null == description
3764             ? _value.description

```

```

3764         : description // ignore:
3765             cast_nullable_to_non_nullable
3766             as String ,
3767     contacts: null == contacts
3768         ? _value.contacts
3769         : contacts // ignore: cast_nullable_to_non_nullable
3770             as String ,
3771     canRedact: null == canRedact
3772         ? _value.canRedact
3773         : canRedact // ignore:
3774             cast_nullable_to_non_nullable
3775             as bool ,
3776     ));
3777 }
3778 }
3779 /// @nodoc
3780 @JsonSerializable()
3781 class _$ConferenceDetailResponseImpl implements
3782     _ConferenceDetailResponse {
3783     const _$ConferenceDetailResponseImpl(
3784         {required this.id ,
3785         required this.name ,
3786         required this.city ,
3787         required this.topic ,
3788         required this.startDate ,
3789         required this.endDate ,
3790         required this.description ,
3791         required this.contacts ,
3792         required this.canRedact});
3793
3794     factory _$ConferenceDetailResponseImpl.fromJson (Map<String ,
3795         dynamic> json) =>
3796         _\begin{equation}\begin{gathered}
3797             ConferenceDetailResponseImplFromJson(json);
3798
3799     @override
3800     final int id;
3801     @override
3802     final String name;
3803     @override

```

```

3800     final String city;
3801     @override
3802     final String topic;
3803     @override
3804     final String startDate;
3805     @override
3806     final String endDate;
3807     @override
3808     final String description;
3809     @override
3810     final String contacts;
3811     @override
3812     final bool canRedact;
3813
3814     @override
3815     String toString() {
3816         return 'ConferenceDetailResponse(id: $id, name: $name,
            city: $city, topic: $topic, startDate: $startDate,
            endDate: $endDate, description: $description, contacts
            : $contacts, canRedact: $canRedact)';
3817     }
3818
3819     @override
3820     bool operator ==(Object other) {
3821         return identical(this, other) ||
3822             (other.runtimeType == runtimeType &&
3823                 other is _$ConferenceDetailResponseImpl &&
3824                 (identical(other.id, id) || other.id == id) &&
3825                 (identical(other.name, name) || other.name ==
                    name) &&
3826                 (identical(other.city, city) || other.city ==
                    city) &&
3827                 (identical(other.topic, topic) || other.topic ==
                    topic) &&
3828                 (identical(other.startDate, startDate) ||
                    other.startDate == startDate) &&
3829                 (identical(other.endDate, endDate) || other.
                    endDate == endDate) &&
3830                 (identical(other.description, description) ||
                    other.description == description) &&
3831                 (identical(other.contacts, contacts) ||

```

```

3834         other.contacts == contacts) &&
3835         (identical(other.canRedact, canRedact) ||
3836         other.canRedact == canRedact));
3837     }
3838
3839     @JsonKey(ignore: true)
3840     @override
3841     int get hashCode => Object.hash(runtimeType, id, name, city
3842         , topic, startDate,
3843         endDate, description, contacts, canRedact);
3844
3845     @JsonKey(ignore: true)
3846     @override
3847     @pragma('vm:prefer-inline')
3848     _\end{gathered}\end{equation}
3849     ConferenceDetailResponseImplCopyWith<
3850     _$ConferenceDetailResponseImpl>
3851     get copyWith => __\begin{equation}\begin{gathered}
3852     ConferenceDetailResponseImplCopyWithImpl<
3853     _$ConferenceDetailResponseImpl>(this, _$identity);
3854
3855     @override
3856     Map<String, dynamic> toJson() {
3857         return _\end{gathered}\end{equation}
3858         ConferenceDetailResponseImplToJson(
3859         this,
3860     );
3861     }
3862 }
3863
3864 abstract class _ConferenceDetailResponse implements
3865     ConferenceDetailResponse {
3866     const factory _ConferenceDetailResponse(
3867         {required final int id,
3868         required final String name,
3869         required final String city,
3870         required final String topic,
3871         required final String startDate,
3872         required final String endDate,
3873         required final String description,
3874         required final String contacts,

```

```

3869         required final bool canRedact})) =
3870             _$ConferenceDetailResponseImpl;
3871     factory _$ConferenceDetailResponse.fromJson(Map<String,
3872         dynamic> json) =
3873         _$ConferenceDetailResponseImpl.fromJson;
3874     @override
3875     int get id;
3876     @override
3877     String get name;
3878     @override
3879     String get city;
3880     @override
3881     String get topic;
3882     @override
3883     String get startDate;
3884     @override
3885     String get endDate;
3886     @override
3887     String get description;
3888     @override
3889     String get contacts;
3890     @override
3891     bool get canRedact;
3892     @override
3893     @JsonKey(ignore: true)
3894     _$ConferenceDetailResponseImplCopyWith<
3895         _$ConferenceDetailResponseImpl>
3896         get copyWith => throw _privateConstructorUsedError;
3897 }
3898 conference_api.dart
3899 import 'package:dio/dio.dart';
3900
3901 import 'models/conference_detail_response.dart';
3902 import 'models/conference_request.dart';
3903 import 'models/conference_search_options.dart';
3904 import 'models/conference_short_response.dart';
3905 import 'models/conference_updating_request.dart';
3906

```

```

3907 class ConferenceApi {
3908     final Dio _dio;
3909
3910     ConferenceApi(this._dio);
3911
3912     Future<ConferenceSearchOptions> getSearchOptions() async {
3913         final response = await _dio.get('/
            conferences_search_options');
3914         return ConferenceSearchOptions.fromJson(response.data);
3915     }
3916
3917     Future<List<ConferenceShortResponse>> getConferencesList({
3918         String? name,
3919         int? cityId,
3920         int? topicId,
3921         String? startDate,
3922         String? endDate,
3923         String? description,
3924         String? contacts,
3925         String? imageUrl,
3926     }) async {
3927         final response = await _dio.get('/conferences_list',
            queryParameters: {
3928             'name': name,
3929             'cityId': cityId,
3930             'topicId': topicId,
3931             'startDate': startDate,
3932             'endDate': endDate,
3933             'description': description,
3934             'contacts': contacts,
3935             'imageUrl': imageUrl,
3936         });
3937
3938         return (response.data as List<dynamic>)
3939             .map((item) => ConferenceShortResponse.fromJson(item)
3940                 )
3941             .toList();
3942     }
3943
3944     Future<void> createConference(ConferenceRequest request) =>
        _dio.post(

```

```

3944         '/create_conference ',
3945         data: request.toJson(),
3946     );
3947
3948     Future<ConferenceDetailResponse> getConferenceDetails(
3949         int conferenceId) async {
3950         final response = await _dio.get(
3951             '/conference/$conferenceId ',
3952         );
3953         return ConferenceDetailResponse.fromJson(response.data);
3954     }
3955
3956     Future<void> updateConference(int conferenceId,
3957         ConferenceUpdatingRequest request) =>
3958         _dio.put(
3959             '/conference/$conferenceId ',
3960             data: request.toJson(),
3961         );
3962
3963     Future<void> deleteConference(int conferenceId) => _dio.
3964         post(
3965             '/delete_conference/$conferenceId ',
3966         );
3967
3968     main.dart
3969     import 'package:conference_app/features/di/app_providers.dart';
3970
3971     import 'package:flutter/material.dart';
3972     import 'package:flutter_riverpod/flutter_riverpod.dart';
3973
3974     import 'features/auth/ui/auth_screen.dart';
3975
3976     void main() {
3977         runApp(const MyApp());
3978     }
3979
3980     class MyApp extends StatelessWidget {
3981         const MyApp({super.key});

```

```

3982 Widget build(BuildContext context) {
3983   return ProviderScope(
3984     child: Consumer(
3985       builder: (_, ref, __) => MaterialApp(
3986         title: 'Flutter Demo',
3987         navigatorKey: ref.watch(AppProviders.navigatorKey),
3988         theme: ThemeData(
3989           colorScheme: ColorScheme.fromSeed(seedColor:
3990             Colors.deepPurple),
3991           useMaterial3: true,
3992         ),
3993         home: const LoginForm(),
3994       ),
3995     );
3996   }
3997 }

```