

*Баркаръ Д.С., студент,
Институт Информационных технологий,
МИРЭА — Российский технологический университет.
Петренко А.А., к.т.н., доцент
МИРЭА — Российский технологический университет,
Россия, г. Москва.*

**РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО ПОИСКУ
ПОПУТЧИКОВ "CARSEAR"**

***Аннотация:** Статья посвящена разработке мобильного приложения по поиску попутчиков "CarSear" на платформе Android. В работе исследованы характеристики предметной области, поставлены задачи для разработки системы и проведён обзор существующих аналогов. Оценены экономические параметры разработки и внедрения системы, обоснован выбор средств разработки. Описаны реализации клиентской части и базы данных системы, меры по обеспечению информационной безопасности, а также технология работы с системой. Представлены графические и табличные материалы для наглядности и обоснования выводов.*

***Ключевые слова:** карточка поездки, Android, мобильное приложение, база данных, Root, архитектура Clean Architecture, управление пользователями, программное обеспечение, карпулинг.*

*Barkar D.S., student,
Institute of Information Technologies
MIREA - Russian Technological University*

*Petrenko A.A., Associate Professor of the Department of MOSIT,
candidate of technical sciences,
MIREA - Russian Technological University,
Russia, Moscow.*

***DEVELOPMENT OF A MOBILE APPLICATION FOR SEARCHING FOR
TRAVEL FRIENDS "CARSEAR"***

Annotation: *The article is devoted to the development of a mobile application for finding travel companions "CarSear" on the Android platform. The work examined the characteristics of the subject area, set tasks for developing the system, and reviewed existing analogues. The economic parameters of the development and implementation of the system were assessed, and the choice of development tools was justified. The implementation of the client part and the system database, measures to ensure information security, as well as the technology for working with the system are described. Graphic and tabular materials are presented for clarity and substantiation of conclusions.*

Keywords: *trip card, Android, mobile application, database, Room, Clean Architecture, user management, software, carpool.*

Поиск попутчиков становится все более популярным способом путешествовать экономично и интересно. Однако, несмотря на множество существующих приложений и сайтов, многие пользователи сталкиваются с проблемами в организации совместных поездок. Целью данного проекта является создание приложения, которое поможет пользователям находить попутчиков для любых видов путешествий.

Целью данной работы является создание приложения, которое облегчит поиск попутчиков. Приложение будет предлагать пользователям различные варианты попутчиков, учитывая их предпочтения и маршрут. Пользователи смогут общаться друг с другом внутри приложения, обсуждать детали поездки и делиться опытом.

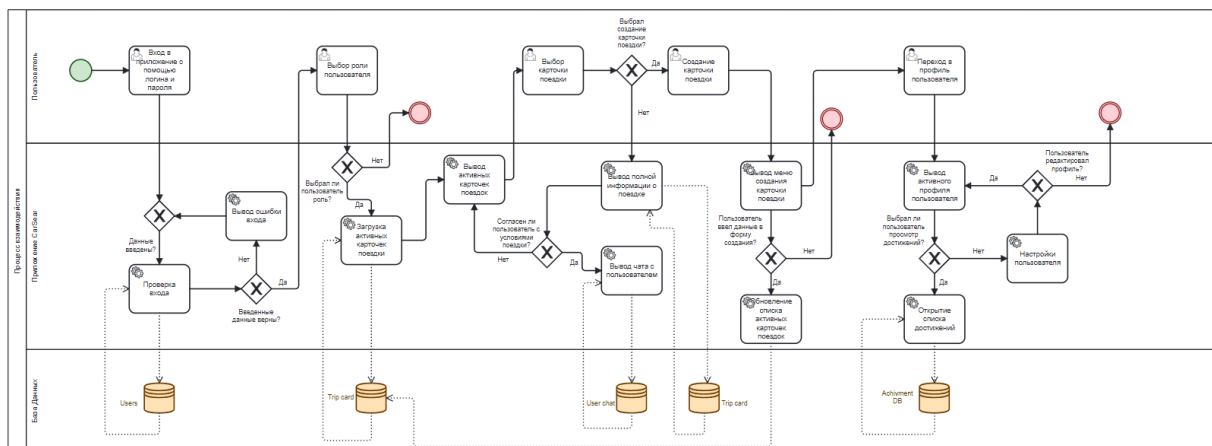
Сравнение аналогов важно для эффективного проектирования собственной системы, поскольку позволяет оценить различные альтернативы и выбрать наиболее подходящее решение в соответствии с уникальными потребностями и целями. В таблице 1 представлен сравнительный анализ некоторых аналогов, чтобы продемонстрировать важность этого процесса при принятии решения о выборе наилучшей системы для конкретных задач и контекста.

Параметр/система	BlaBlaCar	Waze Carpool	UberPOOL	Gett	Via
Доступность в регионах	+	+	+	-	-
Поддержка мобильных устройств	+	+	+	-	+
Безопасность данных	+	+	+	+	+
Удобство интерфейса	-	+	-	-	-
Функции обратной связи	+	-	+	+	-
Быстродействие приложения	+	+	+	-	+

Таблица 1.1 – Приложения-аналоги

Как видно из таблицы, каждое приложение имеет свои достоинства и недостатки.

Рассмотрим основной бизнес-процесс взаимодействия пользователя с системой. На рисунке 1 изображена BPMN-схема с процессом авторизации и проверки входа в приложение. После попытки пользователя войти в приложение система проверяет введенные данные с базой данных пользователей. В случае успешной проверки, система подключается и запускает интерфейс выбора роли пользователя. В случае отсутствия подключения выводит ошибку на экран и предлагает пользователю войти заново. Если пользователь решает повторить авторизацию, то весь процесс проверки проходит заново. Далее на ней происходят процессы взаимодействия с приложением: выбор карточки поездки, создание карточки поездки, переход в профиль пользователя.



Мобильное приложение "CarSear" разрабатывается с целью предоставить пользователям современный и удобный инструмент для комфортного карпулинга на основе быстрого взаимодействия с пользователями.

Особенности данной системы:

Приложение разрабатывается с фокусом на удобстве использования. Пользователи смогут легко и быстро открывать, выбирать карточки поездки по направлению их движения. Интуитивно понятный интерфейс делает систему доступной для широкой аудитории.

Все данные о пользователях, карточках поездок, достижениях и пользовательских чатах будут храниться в единой базе данных. Это обеспечит прозрачность и удобство работы для пользователей.

Приложение создается как ядро, которое может легко расширяться и дополняться новым функционалом в будущем. Это дает возможность системе расти вместе с изменяющимися потребностями пользователей и рынка.

С учетом растущей потребности в удобных и эффективных решениях для карпулинга, разработка мобильного приложения "CarSear" представляется весьма оправданной. Она обеспечивает удовлетворение потребностей пользователей, обеспечивая прозрачность, доступность и гибкость в использовании.

Следующим шагом можно выделить определение функциональных и пользовательских требований.

Функциональные требования — это спецификации и описания функций и поведения системы или программного продукта. Они определяют, как система должна взаимодействовать с пользователем:

- Пользователи должны иметь возможность зарегистрироваться в приложении;

- Пользователи должны иметь возможность создавать профили, содержащие информацию о них (например, имя, фотографию, местоположение, предпочтения по путешествиям и т.д.);
- Приложение должно обеспечивать возможность поиска и подбора попутчиков на основе различных критериев, таких как местоположение, направление, время выезда и т.д.;
- Приложение должно обеспечивать возможность безопасных онлайн- платежей для оплаты услуг по поиску попутчиков;
- Система должна давать пользователям возможность авторизации.

Пользовательские требования — это конкретные функциональные и нефункциональные характеристики, которые определяются потребностями и ожиданиями пользователей от системы или приложения. Они описывают, что система должна делать и какие условия должны быть удовлетворены, чтобы удовлетворить потребности пользователей.

User Story — это формат описания требований, который сосредотачивается на потребностях конечного пользователя. Он используется для описания функциональности или особенностей, которые пользователь хотел бы видеть в системе. User Story формулируется в виде короткого описания, которое включает в себя роль пользователя, его потребность и цель, которую он хочет достичь.

User Story:

- Как пользователь, я хочу иметь доступ к актуальному списку карточек поездок;
- Как пользователь, я хочу, чтобы приложение было простым и интуитивно понятным, без перегрузки информацией, чтобы легко найти нужную мне информацию;
- Как пользователь, я хочу иметь возможность просматривать информацию о каждом водителе;

- Как группа пользователей, мы хотим получать обратную связь от модераторов;
- Как пользователь, я хочу иметь возможность просмотра рейтинга и отзывов о конкретном пользователе;
- Как пользователь, я хочу иметь возможность авторизоваться в приложении, чтобы сохранить личные данные.

Для обеспечения информационной безопасности необходимо разграничить права доступа по ролям. Для этого составим матрицу доступа.

В системе предполагается наличие трёх ролей:

1. Пользователь.
2. Модератор.
3. Администратор.

Пользователь — это обычный пользователь, который имеет функционал выбора уровня сложности задачи, просмотра задач, решение задач, просмотр правильного решения.

Модератор — это пользователь на общественных сетевых ресурсах, имеющий более широкие права, чем обычные пользователи, но менее широкие, чем администратор.

Администратор — это лицо, которое управляет работой веб-сайта, форума или другого сетевого ресурса. Администратор имеет полный контроль над всеми функциями и настройками ресурса.

Матрица доступа отображена в Таблице 2.3. В качестве сокращений будут использованы следующие буквы:

- R — чтение;
- C — создание;
- D — удаление;
- U — обновление.

В каждом поле будет указываться набор букв, который будет обозначать доступность функционала для каждой роли.

Таблица 2.3 — Матрица доступа

№	Прецеденты	Пользователь	Модератор	Администратор
1.	Аутентификация	C, R, D	C, R, D	C, R, U, D
2.	Профиль	C, R, U, D	R, U, D	R, U, D
3.	Карточка поездки	C, R, U, D	R, D	R, D
4.	Чат	C, R, U, D	R, D	R, D
5.	Рейтинг	C, R, U	R, U, D	R, U, D
6.	Доступ к данным пользователей		R, U, D	C, R, U, D

Матрица доступа — это модель безопасности состояния защиты в системе. Она используется для определения прав каждого пользователя и каждой роли в системе.

Рассмотрим модель базы данных на рисунке 2.

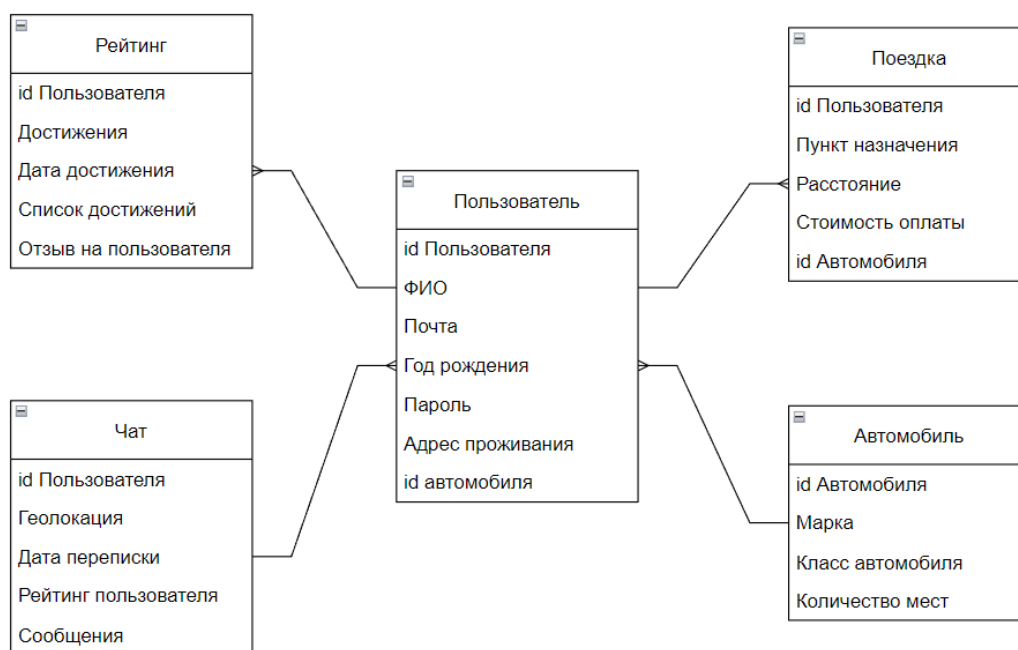


Рисунок 2 — Логическая схема базы данных

В данной модели отображено 5 блоков таблиц, взаимодействующих друг с другом. Данная база данных отображает взаимодействия между таблицами и помогают понять реализацию её работы внутри нашей системы.

Словарь данных представляет собой определенным образом, организованный список всех элементов данной системы с их точными определениями.

Словарь данных отображен в Таблицах 3-7.

Таблица 3 — Таблица пользователь

№	Наименование	Предназначение	Тип
1.	Id Пользователя	Уникальный идентификатор пользователя в БД	int
2.	ФИО	Отображающееся в системе имя пользователя	String
3.	Почта	Контактная информация и логин пользователя, указанный при регистрации	String
4.	Год рождения	Дата рождения пользователя	String
5.	Пароль	Набор символов, который пользователь использует для авторизации	String
6.	Адрес проживания	Страна проживания пользователя	String
7.	Id Автомобиля	Уникальный идентификатор автомобиля в БД	int

В таблице пользователь будут храниться записи о каждом пользователе. Благодаря ей у нас будет доступ к ним и возможность просмотра и редактирования.

Таблица 4 — Таблица поездки

№	Наименование	Предназначение	Тип
1.	id Пользователя	Уникальный идентификатор пользователя в БД	int
2.	Пункт назначения	Информация о начальном и конечных пунктах поездки	String
3.	Расстояние	Общее расстояние всего маршрута	String
4.	Стоимость оплаты	Стоимость оплаты попутчиком поездки	String
5.	id Автомобиля	Уникальный идентификатор автомобиля в БД	int

В таблице поездки будут храниться все поездки вместе с информацией о пользователе, адреса его проживания.

Таблица 5 — Таблица автомобиль

№	Наименование	Предназначение	Тип
1.	id Автомобиля	Уникальный идентификатор автомобиля в БД	int
2.	Марка и название	Общая информация об автомобиле	String
3.	Класс автомобиля	Информация о классе автомобиля и классе удобства	String
4.	Количество мест	Информация о кол-ве мест в автомобиле	String

В таблице автомобиль будут храниться все информация об автомобиле, предназначенном для поездки.

Таблица 6 — Таблица чат

№	Наименование	Предназначение	Тип
1.	id Пользователя	Уникальный идентификатор пользователя в БД	int
2.	Геолокация	Отображающееся в системе местонахождение пользователя	String
3.	Дата переписки	Набор символов, который показывает пользователю дату прошедшего диалога	String
4.	Рейтинг пользователя	Рейтинг пользователя в системе	String
5.	Сообщения	Система связи между пользователями	String

В таблице чат будет храниться информация о пользовательских чатах, при помощи нее можно модерировать пользовательские чаты.

Таблица 7 — Таблица Рейтинг и достижения

№	Наименование	Предназначение	Тип
1.	id Пользователя	Уникальный идентификатор пользователя в БД	int
2.	Достижения	Отображающееся в системе активное достижение пользователя	String
3.	Дата достижения	Набор символов, который показывает дату получения достижения	String
4.	Список достижений	Общий список всех достижений и информации по их получению	String

5.	Отзыв на пользователя	Система отзывов пользователя на другого пользователя	String
----	--------------------------	---------------------------------------------------------	--------

В таблице рейтинга и достижения будет храниться информация о рейтинге определенного пользователя, список выполненных им достижений.

В процессе разработки мобильного приложения "CarSear" для решения шахматных задач на платформе Android были проведены исследования предметной области, поставлены задачи для создания системы и выполнен обзор существующих аналогов. В результате было разработано программное обеспечение, которое включает в себя клиентскую часть и базу данных, реализованную с использованием библиотеки Room и архитектуры Clean Architector. Также были приняты меры для обеспечения информационной безопасности пользователей приложения.

Оценка основных параметров разработки и внедрения системы показала, что выбранные средства разработки и подходы являются эффективными и оправданными. Представленные графические и табличные материалы подтверждают обоснованность принятых решений и достигнутых результатов. Разработанное мобильное приложение "CarSear" демонстрирует высокую функциональность и удобство использования для поиска попутчиков, что делает его конкурентоспособным на рынке аналогичных приложений.

Использованные источники:

1. Антамошкин О.А. Программная инженерия. Теория и практика. Учебник. М: НИЦ Инфра-М, 2012 — 368 с.
2. Зубкова Т.М. Проектирование программных систем по обработке и анализу информации: метод. указания к курсовому проектированию по дисциплине "Технология разработки

программного обеспечения". — ГОУ ОГУ, 2011 — 53 с.

3. Алексанкин Я.Я., Бржозовский А.Э., Жданов В.А. и др. Автоматизированное проектирование систем автоматического управления. — М.: Машиностроение, 1990.
4. Заботина Н. Н. Проектирование информационных систем: учебное пособие. — М.: ИНФРА-М, 2020 — 331 с.
5. Грекул В.И. Проектное управление в сфере информационных технологий. — М.: БИНОМ. Лаборатория знаний, 2013 — 336 с.
6. Варфоломеева А.О., Коряковский А.В., Романов В.П. Информационные системы предприятия: учебное пособие. — М.: НИЦ ИНФРА-М, 2016 — 283 с.