



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. Ломоносова

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ
ПРОГРАММА «РАЗРАБОТЧИК ПРОФЕССИОНАЛЬНО-ОРИЕНТИРОВАННЫХ
КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА НА ТЕМУ

Геоотметки на базе точек доступа WiFi

Работу выполнил:

Шабалин Дмитрий Григорьевич

Научный руководитель:

с.н.с., к.ф.-м.н.
Намиот Дмитрий Евгеньевич

МОСКВА 2016

Оглавление

Аннотация	2
Глава 1. Приложения с зависимостью от местоположения	3
Глава 2. Разработка архитектуры системы геоотметки.....	7
2.1 Общее описание	7
2.2 Детали функционирования	8
2.3 Статический аспект	9
2.4 Динамический аспект.	11
2.5 Примеры конкретных схем.	12
2.7 Способы организации хранилища данных о местоположении пользователей.....	13
Глава 3. Демонстрационная реализация системы геоотметки.	20
Список литературы.....	22

Аннотация

В работе изложен подход к созданию сервиса геоотметки на базе точек доступа WiFi. Подход основан на использовании сетевой близости для идентификации местоположения мобильных пользователей. Местоположение пользователя «привязывается» непосредственно к беспроводным точкам доступа, без вычисления координат. Описаны компоненты системы, предоставляющей сервис геоотметки, предложены схемы с различными функциональными характеристиками. Предложен способ организации эффективного и масштабируемого хранилища данных о местоположении мобильных пользователей в реляционной СУБД. Создана демонстрационная система, предоставляющая сервис геоотметки, доступная в виде open-source проекта.

Глава 1. Приложения с зависимостью от местоположения

Все более широкое использование мобильных вычислительных устройств и рост их производительности создает запрос на приложения, зависящие от местоположения. Зависимость от местоположения означает, что эти приложения в своей работе учитывают текущее местоположение мобильного пользователя. Разработка таких приложений относится к области так называемых контекстно-зависимых вычислений.

Существуют различные определения контекста. Первоначальное определение [1] описывает контекст как информацию о местоположении, людях, находящихся рядом, и их изменениях. Соответственно, контекстно-зависимые вычисления - это вычисления, использующие контекст. В общем случае под контекстом понимается любая информация, характеризующая ситуацию, в которой оказалась сущность [2]. Местоположение входит в понятие контекста. В случае мобильного устройства, данные, образующие контекст, обычно поступают от датчиков, которыми оборудовано устройство.

Концепция контекстно-зависимых вычислений позволяет рассматривать понятие местоположения в обобщенном смысле, как любую информацию, позволяющую связать конкретного пользователя с некоторой точкой в пространстве или локацией. Например, в качестве данных о местоположении может служить адрес дома, название станции метро или торгового центра. Нахождение устройства пользователя в зоне действия определенной беспроводной сети также позволяет с некоторой точностью локализовать его положение в пространстве.

Приложения с зависимостью от местоположения базируются на различных методах определения местоположения. Традиционным способом определения местоположения является использование глобальной системы позиционирования (GPS, ГЛОНАСС). Однако сигнал системы глобальной навигации может быть недоступен в помещениях или густо застроенной местности. Для определения местоположения в этих условиях используются другие методы [3, 4]. Некоторые из них требуют использования специального оборудования. Так, система локации, основанная на метках RFID, предполагает наличие у пользователей радиометок (активных или пассивных) и установленных в помещении сканеров, которые определяют положение меток. Система локации UWB (Ultra-Wideband) состоит из установленных в помещении сканеров и приемопередатчиков у пользователей. Сканеры определяют положение приемопередатчиков, и, следовательно, их носителей, анализируя время прохождения низкочастотного сигнала.

Системы позиционирования, основанные на анализе сигналов беспроводных точек

доступа, WiFi или Bluetooth, имеют ряд преимуществ. Такие системы часто представляют чисто программные решения, не требующие использования специального оборудования и соответствующего проприетарного ПО. Источником сигнала для таких систем могут быть точки доступа, имеющиеся в большом количестве в современных городах. Возможно и установление дополнительных беспроводных точек. Приемником сигнала может служить любое устройство, поддерживающее взаимодействие по беспроводной сети. Использование стандартных частот и протоколов множественного доступа в сетях WiFi или Bluetooth существенно снижают опасность радио-интерференции с другими системами.

Протокол WiFi описывается серией стандартов IEEE 802.11 [5]. Стандарт определяет набор служб (service set) как множество устройств, относящихся к некоторой локальной беспроводной сети (WLAN). Идентификатор набора служб (service set identifier, SSID) состоит из 32 октетов и обычно представляет понятное для человека имя беспроводной сети. Существуют два режима функционирования сети WiFi - инфраструктурный (infrastructure mode) и Adhoc. В инфраструктурном режиме беспроводные устройства общаются между собой через точки доступа (access point). В режиме Adhoc связь устанавливается непосредственно между устройствами, оборудованными WiFi адаптерами. Беспроводная сеть состоит из блоков, именуемых как базовый набор служб (basic service set). В инфраструктурном режиме базовый набор служб представляет собой одну точку доступа с соответствующей ей станцией. Базовый набор служб имеет идентификатор, обозначаемый как BSSID (basic service set identifier). В инфраструктурном режиме в качестве BSSID выступает MAC-адрес беспроводной точки доступа. Таким образом, идентификатором отдельной точки доступа сети WiFi, работающей в инфраструктурном режиме, является BSSID, представленный MAC-адресом этой точки.

Методы определения положения с использованием точек доступа WiFi могут быть разделены на четыре основных класса. Метод интенсивности сигнала (RSSI) основан на измерении интенсивности сигнала WiFi от нескольких точек доступа. Для вычисления позиции используется трилатерация. Метод достаточно прост в реализации, однако имеет низкую точность и подвержен ошибкам из-за наличия препятствий с различной радиопрозрачностью. Метод отпечатков (FingerPrints) также основывается на измерении интенсивности входящего сигнала. Векторы значений интенсивности сигнала от различных точек доступа сохраняются в базе данных вместе с сопутствующими координатами. Эта информация может быть детерминированной или вероятностной. Для определения координат, текущий вектор значений интенсивности сравнивается с векторами интенсивностей, сохраненными в базе, с целью нахождения записи, наилучшим образом соответствующей текущему положению. Имеются также методики, основанные на

измерении времени прохода (ToF) и углов получения (AoA) сигналов [5].

Общей особенностью указанных методов являются необходимость предварительного составления базы данных (карты) с координатами беспроводных точек доступа. Составление такой базы и поддержание ее в актуальном состоянии является трудной задачей. Имеются различные коммерческие и открытые базы данных беспроводных точек доступа с различной степенью покрытия, такие как Mozilla Location Service, Navizon, OpenWLANMap и др..

Альтернативный подход [3, 6-7] не нуждается в базе данных координат беспроводных точек доступа. Идея заключается в том, что для идентификации положения пользователя не обязательно вычислять его координаты. Информация может быть «привязана» непосредственно к беспроводным сетям или к отдельным точкам доступа. Позиция пользователя, таким образом, будет идентифицироваться параметрами беспроводных сетей, видимых устройством пользователя. Для наиболее точной идентификации позиции пользователя и принятия решения, какая информация является для него в данный момент актуальной, могут использоваться различные параметры беспроводной сети: SSID, BSSID, сила сигнала (RSSI) и др.

Этот подход имеет реализацию в виде экспертной системы SpotEx. Эта система позволяет привязывать данные (например, справочную или рекламную информацию) к публично видимым беспроводным сетям или к отдельным точкам доступа. Привязка проходит путем составления базы данных правил. Правила создаются и редактируются заинтересованными лицами и организациями через веб-интерфейс. Устройство пользователя сканирует беспроводные точки доступа и отправляет их параметры на сервер. Сервер сопоставляет полученный набор с хранящимися у него правилами и определяет, какие правила соответствуют полученному набору. Предоставление информации пользователю происходит в соответствии с этими правилами.

Особенностью системы SpotEx является то, что изменение базы данных правил происходит значительно реже, чем обращение к ней за получением информации. Потребителями информации являются мобильные пользователи, для которых в общем случае не требуется авторизация, т.к. информация является публично доступной. Авторизация требуется лишь при добавлении и редактировании правил.

В данной работе предлагается использовать аналогичный подход для создания сервиса геоотметки, т.е. сервиса, позволяющего пользователям обмениваться информацией о собственном местоположении. С помощью этого сервиса пользователь, по прибытию в определенную локацию, может «отметиться» в ней. Эта отметка «привязывается» к находящимся в зоне видимости точкам доступа WiFi и становится доступной для других

пользователей, находящихся в этой же локации. В отличие от экспертной системы SpotEx, сервис геоотметки предполагает, что конечные пользователи являются как источниками, так и потребителям информации, привязанной к беспроводным точкам доступа. Подобный сервис может иметь реализацию как в виде самостоятельного приложения, так и являться компонентом более общей системы с зависимостью от местоположения.

Целью данной работы является анализ, разработка архитектуры и демонстрационная реализация системы, предоставляющей сервис геоотметки, с использованием для определения местоположения пользователей беспроводных точек доступа WiFi.

Глава 2. Разработка архитектуры системы геотметки

2.1 Общее описание

Под сервисом геотметки будем понимать сервис обмена между пользователями информацией об их местоположении. Систему, реализующую сервис геотметки, будем называть системой геотметки.

Введем понятие отметки местоположения пользователя, или геотметки. Под отметкой местоположения будем понимать информационную единицу, связывающую пользователя с определенным положением в пространстве. Она может иметь атрибуты, такие как время отметки, срок действия, и др.. В данной работе местоположение в пространстве представлено набором идентификаторов BSSID беспроводных точек доступа, в зоне действия которых находится устройство пользователя. Таким образом, отметку местоположения в узком смысле можно понимать как структуру данных, состоящую из идентификатора пользователя, набора BSSID и набора атрибутов. Функциями сервиса геотметки являются создание отметок местоположения, их хранение и обеспечение выдачи по результатам поиска среди сохраненных отметок.

Предлагаемая в данной работе система геотметки состоит из сервера, агрегирующего данные о местоположении пользователей, и клиентской программы, устанавливаемой на мобильные устройства. Клиентская программа обеспечивает сбор и отправку на сервер списка идентификаторов BSSID обнаруженных беспроводных точек доступа, а также служебных команд. Получив список BSSID сервер может:

а) Использовать полученный список для фиксации местоположения владельца устройства, создав отметку местоположения. Информация о местоположении владельца устройства, сохраненная в виде отметки местоположения, может быть доступна для других пользователей.

б) Вернуть клиентской программе список пользователей, расположенных в той же локации, что и владелец устройства.

Клиентская программа, получив список пользователей, отображает его на экране.

Предполагается, что пользователь представлен в системе в виде идентификатора учетной записи в одной из социальных сетей. Для получения идентификатора при первом старте клиентской программы пользователю предлагается авторизоваться в социальной сети. Для получения фамилий, имен, фотографий и других необходимых для отображения данных

о пользователях, клиентская программа формирует запрос к соответствующим сервисам социальной сети. Общая архитектура системы изображена на рис. 1.

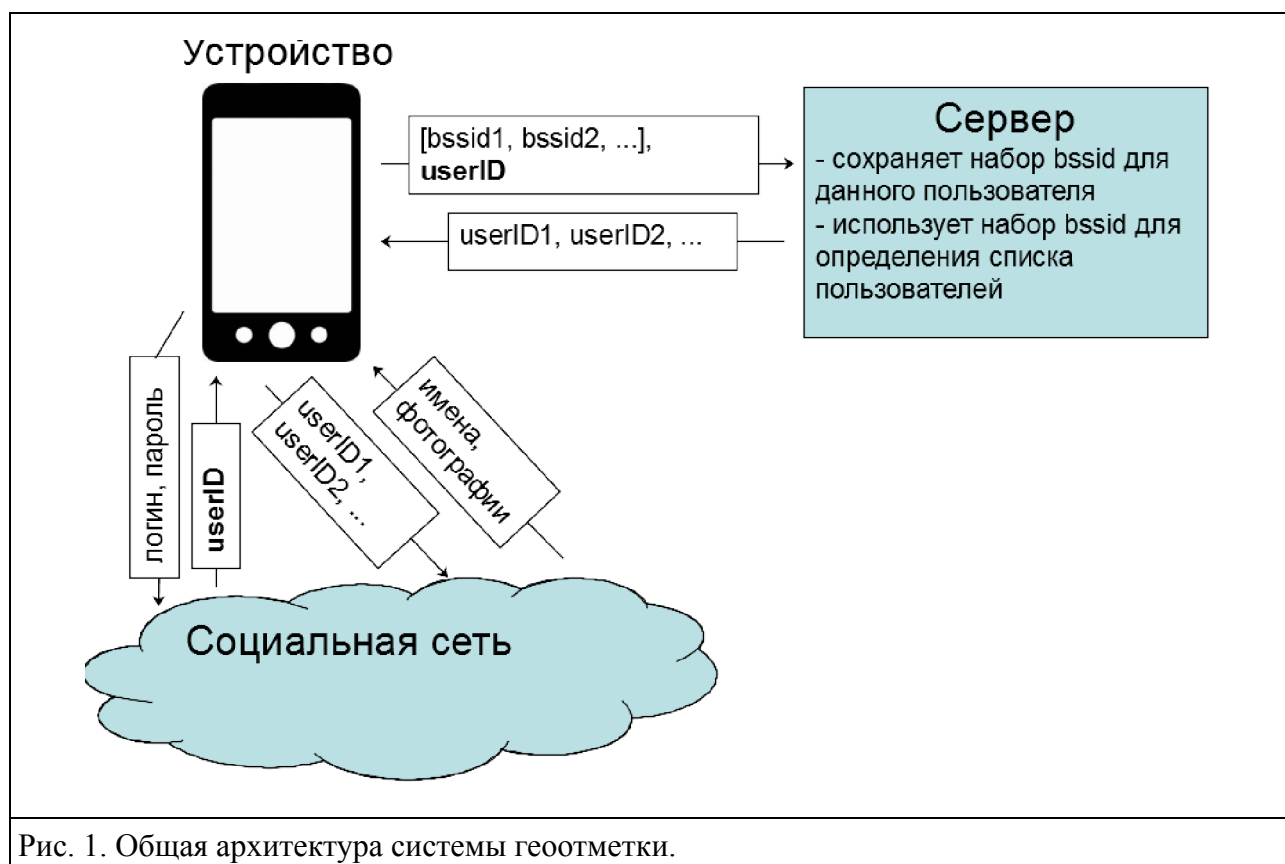


Рис. 1. Общая архитектура системы геоотметки.

2.2 Детали функционирования

Для определения, находятся ли пользователи в одной локации, используется следующий критерий: наличие хотя бы одной точки доступа, видимой устройствами обоих пользователей. В терминах отметок о местоположении наличие общей точки доступа означает, что у каждого пользователя имеется действующая отметка и множества BSSID у этих отметок имеют не пустое пересечение. Рассмотрение более сложных критериев близости, учитывающих, например, количество совместно видимых точек или интенсивность сигнала (RSSI), выходит за рамки данной работы. Отметим лишь, что задачу поиска пользователей с использованием сложных критериев близости можно свести к задаче поиска по простому критерию с последующим наложением некоторых фильтров на результат выборки.

Отметка местоположения пользователя может характеризоваться таким параметром, как видимость. Видимая отметка является публичной и содержащаяся в ней информация о местоположении пользователя может быть отображена устройствами других пользователей.

Если отметка невидима, то информация о местоположении пользователя доступна только серверу и самому пользователю. Невидимые отметки могут использоваться для сбора статистики посещений, или в других целях. В простейшем случае все отметки являются видимыми.

Отметим следующую особенность, которую следует учитывать при выборе архитектуры системы геоотметки. Как было сказано, список BSSID, полученный сервером, может использоваться в двух целях: для фиксации местоположения владельца устройства (создания отметки), и поиска пользователей, находящихся в той же локации, что и владелец устройства. С точки зрения экономии трафика целесообразно организовать взаимодействие клиентской программы и сервера таким образом, чтобы полученный сервером список BSSID использовался одновременно в обоих целях. С другой стороны такой подход может быть недостаточно гибким для придания системе нужных эксплуатационных характеристик.

2.3 Статический аспект

Для описания различных способов организации работы системы геоотметки и выбора конкретной схемы с необходимой функциональностью, рассмотрим статический и динамический аспекты функционирования.

Статический аспект функционирования системы определяет, какого рода структуры данных хранятся в системе и могут быть доступны для обработки и отображения. Для описания структуры хранимых данных удобно использовать объектную модель. Заметим, что в данном разделе объектная модель используется для описания логической структуры данных, а не конкретных классов языка программирования. Базовыми элементами объектной модели являются:

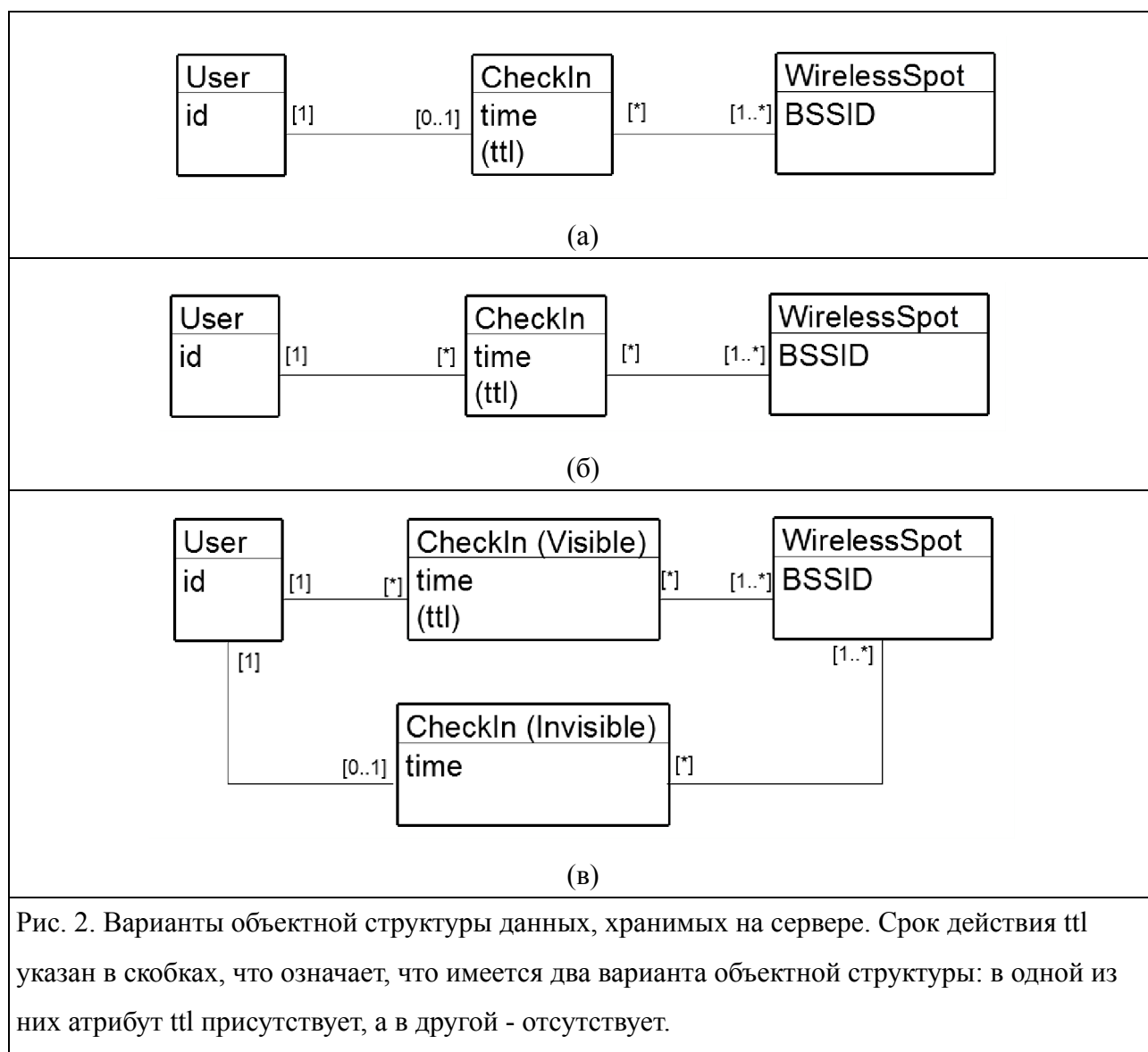
- пользователь (User)
- отметка местоположения пользователя (Checkin).
- беспроводная точка доступа (WirelessSpot).

Объектная модель данных может характеризоваться:

1. Возможным количеством отметок местоположения на одного пользователя. В простейшем случае пользователю соответствует одна отметка, соответствующая его последнему местоположению. В более сложных случаях может храниться более одной отметки. Это позволяет получать данные не только о текущем местоположении, но и, например, историю перемещений пользователя.

2. Состояниями или типами отметок. Как было указано выше, предлагается рассматривать видимые и невидимые отметки.
3. Набором атрибутов, характеризующих базовые элементы объектной модели. Например, атрибутами отметки местоположения могут быть время создания, срок действия, и др..
4. Дополнительной информацией, привязанной к базовым элементам модели: к пользователю, к отметке, к точкам доступа. Например, к точкам доступа может быть привязана справочная или рекламная информация. Отметим, что детальное обсуждение дополнительных сервисов, которые может предоставлять система геоотметки, выходит за рамки текущей работы.

Примеры диаграмм для объектной структуры данных приведены на рис. 2.



2.4 Динамический аспект.

Динамический аспект функционирования системы определяется главным образом правилами создания, удаления и изменения атрибутов отметок местоположения, т.е. жизненным циклом отметки. Жизненный цикл отметки характеризуется двумя аспектами:

1. Правилами фиксации положения пользователя, т.е. создания отметки.
2. Правилами отслеживания положения пользователя после создания отметки. Эти правила нужны для определения момента, после которого будет считаться, что пользователь покинул отмеченное положение.

В части фиксации положения пользователя (создания отметки) можно предложить следующие варианты организации жизненного цикла отметки местоположения (рис. 3):

1. Отметка всегда является видимой и создается при каждом получении списка BSSID от устройства пользователя. См. рис. 3 а.
2. Отметка всегда является видимой и создается при получении списка BSSID совместно с явным указанием создать отметку. См. рис. 3 б.
3. Отметка может быть невидимой. При получении списка BSSID создается невидимая отметка. Впоследствии она может стать видимой при получении соответствующей команды от пользователя. См. рис. 3 в.
4. Отметка может быть невидимой. При получении списка BSSID создается невидимая отметка, а при получении списка BSSID одновременно с соответствующей командой - видимая. См. рис. 3 г.

В части отслеживания местоположения пользователя можно предложить следующие подходы:

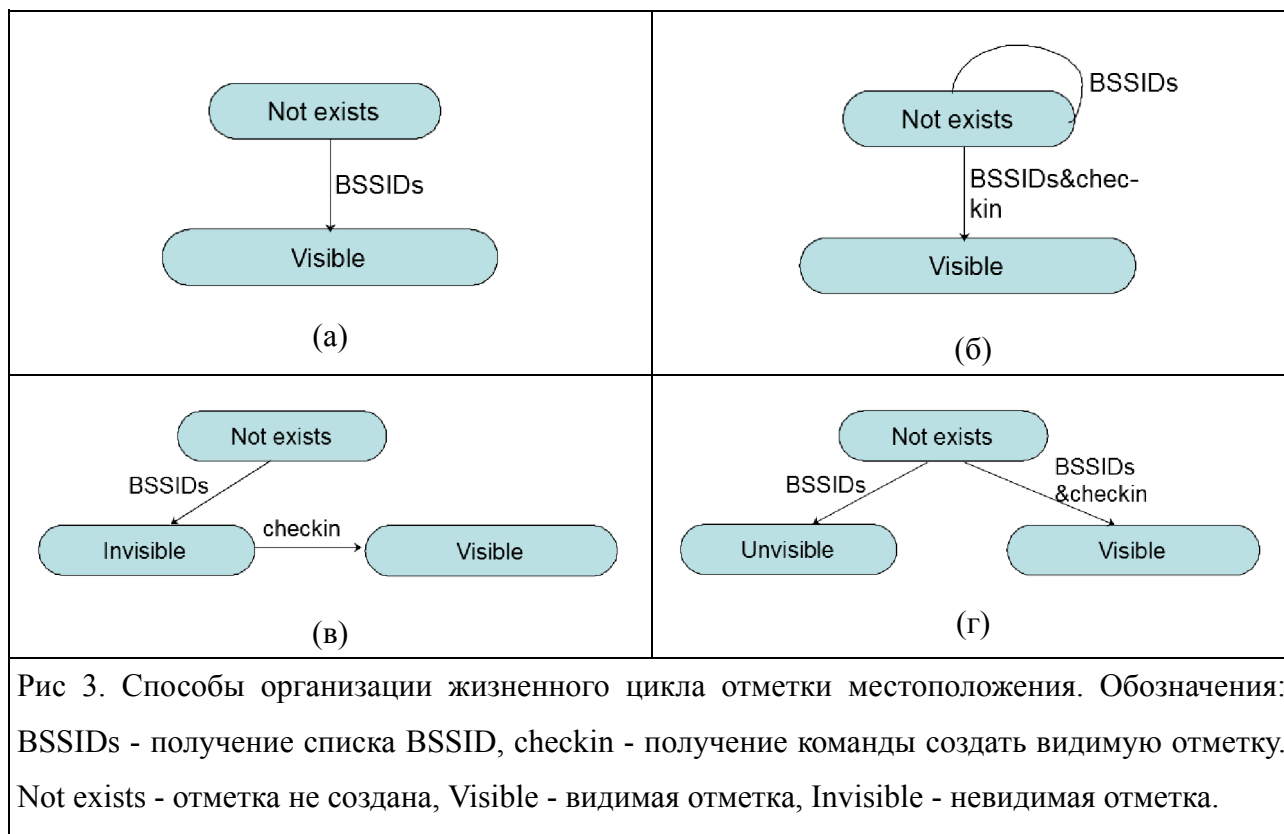
1. Без отслеживания. Отметка имеет единственный атрибут - время создания, и удаляется только тогда, когда становится заведомо не актуальной. При попадании в определенную локацию пользователь сможет увидеть список других пользователей, отмечавшихся в этой локации, сортированный по времени отметки.
2. Отслеживание по времени. Отметка имеет два атрибута - время создания и срок действия. По истечении срок действия пользователь считается покинувшим локацию. Срок действия может быть:
 - фиксированным.

- задаваемым пользователем.

При получении нового списка BSSID или команд пользователя возможно автоматическое изменение срока действия отметки в сторону уменьшения или увеличения.

3. Непрерывное отслеживание. Передача списка BSSID происходит через определенные интервалы. Отслеживание может происходить:

- все время при включенном приложении (режим on/off).
- начинаться или заканчиваться при наступлении определенных событий. Например, отслеживание может начинаться после создания отметки и заканчиваться через некоторый промежуток времени. Отметим, что режим с непрерывным отслеживанием требует большего расхода батареи и потребляет больше трафика.



2.5 Примеры конкретных схем.

Схема А. Данная схема является одной из простейших. Для каждого пользователя на сервере храниться единственная отметка, соответствующая его последнему местоположению (рис. 2 а). Она имеет единственный атрибут - время создания. При каждом получении списка BSSID от устройства пользователя происходит обновление данных о местоположении

пользователя (рис. 3 а). Отслеживание положения пользователя не ведется.

При прибытии в некоторую локацию L пользователь запускает клиентскую программу с целью узнать, какие еще пользователи, кроме него, находятся в локации L. Устройство пользователя сканирует список беспроводных точек доступа и отправляет его на сервер. Предыдущая отметка удаляется, и вместо нее создается новая, соответствующая локации L. В ответ пользователь получает список других пользователей, отмеченных в локации L. Для каждого пользователя отображается время его отметки в локации.

Схема В. Следующая схема была выбрана для демонстрационной реализации. Для каждого пользователя на сервере храниться единственная отметка, соответствующая его последнему местоположению (рис. 2 а). Она имеет в качестве атрибутов время создания и срок действия. Отметка создается при получении данных позиционирования с указанием создать отметку (рис 3. б). Срок действия выбирается пользователем при создании отметки.

При прибытии в некоторую локацию L пользователь может запустить клиентскую программу и узнать, какие еще пользователи, кроме него, находятся в локации L. Кроме того, пользователь при желании может сам отметить себя в локации L. При создании отметки пользователь выбирает интервал времени, в течении которого он планирует находится в данной локации (15 мин, 30 мин, 1 ч, и т.д.). Список видимых пользователем BSSID и выбранный интервал времени пересылается на сервер с целью создания отметки местоположения. После создания отметки информация том, что пользователь находится в локации L, будет доступна другим пользователям, находящимся в этой локации, в течении указанного интервала времени.

2.7 Способы организации хранилища данных о местоположении пользователей

Проанализировав различные аспекты функционирования системы, предоставляющей сервис геоотметки, можно сделать вывод, что организация хранилища отметок местоположения пользователя требует эффективной реализации следующих базовых операций:

1. Поиск пользователей по списку BSSID. Устройство пользователя посылает список идентификаторов точек доступа, которые оно видит в настоящий момент, и в ответ получает список пользователей, находящихся в радиусе действия этих точек.
2. Создание отметки о местоположении пользователя. Сервер получает список BSSID и сохраняет его в виде отметки;

3. Удаление отметки о местоположении пользователя.

Операция удаления единичной отметки в некоторых случаях может быть не обязательной. Неактуальные отметки могут накапливаться в системе и удаляться групповым образом во время запуска обслуживающей процедуры. В некоторых случаях две или три операции могут быть объединены в одну с целью оптимизации. Например, в системе, использующей схему А, все три операции всегда выполняются последовательно: удаление предыдущей отметки, создание новой и поиск пользователей по списку BSSID. Кроме перечисленных также могут требоваться сервисные операции, связанные с удалением из хранилища неактуальной информации: неактивных пользователей, их отметок, неиспользуемых BSSID.

Структуры данных, описанные в разделе 2.3, можно хранить в реляционной СУБД естественным образом, сохраняя каждую пару {BSSID, отметка} в отдельной строке таблицы. Однако, в таком случае, операция поиска пользователей по списку BSSID потребует выборки всех этих строк. Если обозначить за u - среднее количество пользователей в радиусе видимости одной беспроводной точки доступа, w - среднее количество точек доступа, видимых устройством пользователя, то количество обращений к диску, необходимых для выполнения операции поиска пользователей, можно оценить как $u \cdot w$.

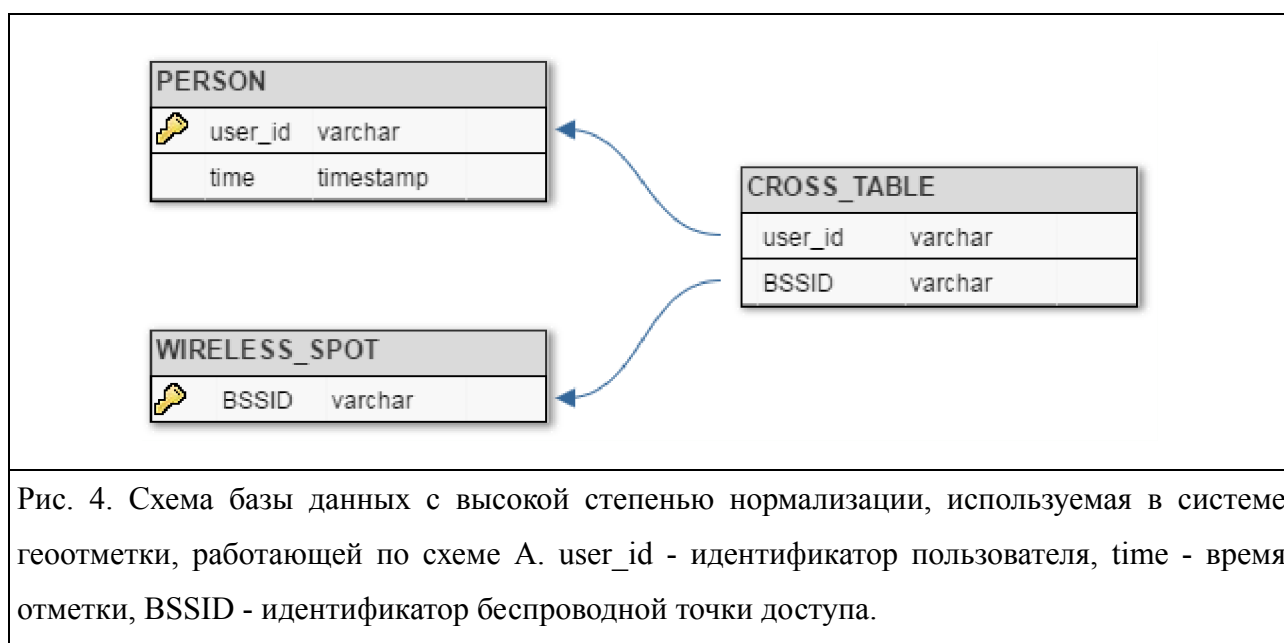


Рис. 4. Схема базы данных с высокой степенью нормализации, используемая в системе геоотметки, работающей по схеме А. user_id - идентификатор пользователя, time - время отметки, BSSID - идентификатор беспроводной точки доступа.

Поясним сказанное на примере системы, работающей по схеме А (см. раздел 2.5). Для этой системы можно использовать базу данных со схемой, изображенной на рис. 4. Операция поиска пользователей по списку BSSID может быть выполнена с помощью

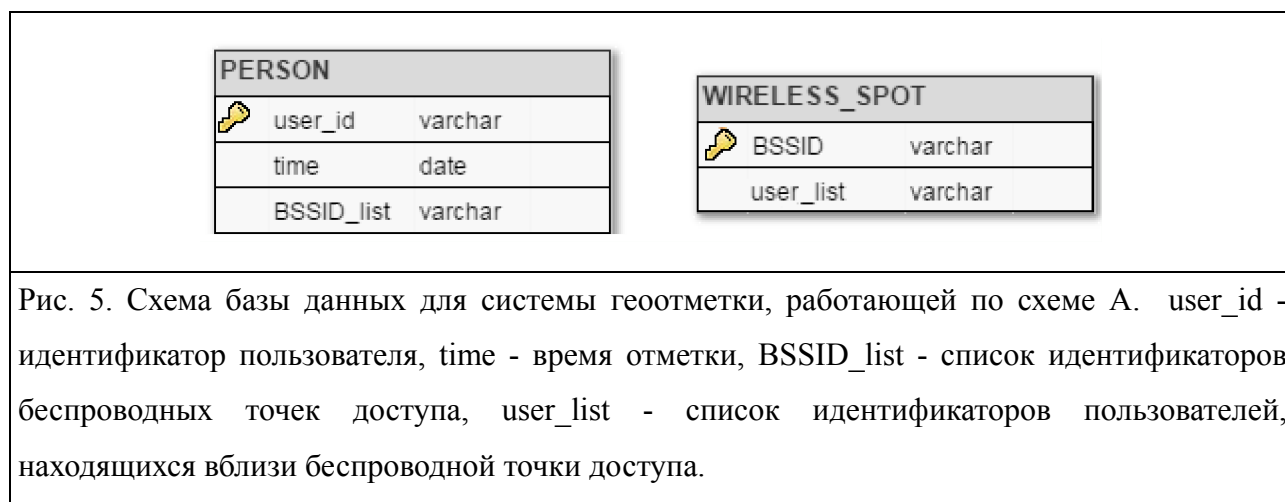
запроса:

```
SELECT DISTINCT USER_ID FROM CROSS_TABLE WHERE BSSID IN {список BSSID}
```

Выполнение этого запроса потребует выборки из таблицы CROSS_TABLE всех строк, BSSID которых находится в заданном списке. В общем случае строки таблицы CROSS_TABLE физически расположены на диске не упорядоченно, поэтому выборка каждой строки потребует перемещения головки диска.

В настоящей работе мы предлагаем способ организации хранилища данных, позволяющий уменьшить количество дисковых операций, необходимых для выполнения запросов. При этом объем данных, хранимых на диске, увеличивается, а сами данные хранятся в реляционной СУБД в ненормализованной форме. Рассмотрим суть этого подхода на примере организации хранилища для системы со схемой А.

Данные хранятся в реляционной базе данных со схемой, изображенной на рис 5. В базе данных поддерживается структура из двух таблиц. Одна из них содержит списки беспроводных точек доступа для каждого пользователя, другая - списки пользователей для каждой беспроводной точки доступа. Списки хранятся в виде строки с использованием разделителя (например, запятой) и обновляются при получении новых данных о местоположении пользователей.



Для осуществления операции создания отметки требуется: сохранить идентификатор пользователя user_id и список BSSID в таблицу PERSON; для каждого BSSID из присланного списка исправить поле user_list таблицы WIRELESS_SPOT, добавив к нему идентификатор пользователя user_id; Если строка с требуемым BSSID, отсутствует в таблице WIRELESS_SPOT, она должна быть создана. Операция удаления отметки требует обратных действий - удаления записи из таблицы PERSON и исправления строк таблицы

WIRELESS_SPOT. Для получения списка пользователей по списку BSSID следует выбрать соответствующие значения user_list из таблицы WIRELESS_SPOT. Каждое значение user_list нужно разбить на множество отдельных идентификаторов, и объединить полученные множества в одно.

Ниже приводится псевдокод, иллюстрирующий порядок выполнения базовых операций.

1. Сохранение отметки:

```
void saveCheckin(String [] bssids, String userID){

    for (String bssid: bssids) {
        // Получение списка пользователей для данной BSSID
        String user_list = {SELECT USER_LIST FROM WIRELESS_SPOT WHERE BSSID =:bssid};

        if (user_list == null){
            // создание новой строки в таблице WIRELESS_SPOT
            INSERT INTO WIRELESS_SPOT VALUES (:userId, :bssid );
        }
        else{
            // обновление имеющейся строки в таблице WIRELESS_SPOT
            user_list = user_list + ',' + userID; // конкатенация строк

            UPDATE WIRELESS_SPOT SET USER_LIST =:user_list WHERE BSSID
            = :bssid;
        }
    }
    // полученный список BSSID преобразуется в строку с разделителем запятая.
    String bssidList = makeString(bssids);
    // Результат сохраняется
    INSERT INTO PERSON VALUES (:userId, current_time, :BSSID_List);
}
```

2. Удаление отметки:

```
void deleteCheckin(String userID){
    // Получение списка BSSID для имеющейся отметки
    String bssidList = {SELECT BSSID_LIST FROM PERSON WHERE USER_ID = :u.userId};

    // Разделение строки на отдельные идентификаторы BSSID
    String [] bssidsArray = split(bssidList);

    for (String bssid : bssidsArray){
        // Получение списка пользователей для данной BSSID в виде строки
        String user_list = {SELECT USER_LIST FROM WIRELESS_SPOT WHERE BSSID =:bssid};

        // Из списка user_list вычеркиваются userID, с учетом наличия разделителя;
        user_list = removeOccurence(user_list, userID);
    }
}
```

```

        // сохранение измененного списка:
        UPDATE WIRELESS_SPOT SET USER_LIST =:user_list WHERE USER_ID = :userId;
    }

    // Удаление записи из таблицы PERSON
    DELETE FROM PERSON WHERE USER_ID=:u.userId;
}

```

3. Поиск пользователей по списку BSSID.

```

class User {
    String userID;
    Date time;
}

User [] getUsersNear(String [] bssidList){
    // Выборка строк из таблицы PERSON
    String [] user_lists = {SELECT USER_LIST FROM WIRELESS_SPOT WHERE BSSID IN (:bssidList)};

    // разделение всех строк из user_lists на отдельные идентификаторы, выборка
    неповторяющихся идентификаторов и создание на их основе массива из User.
    User [] united_user_list = makeUnitedList(user_lists);


    // нахождение и установка атрибута времени для каждого пользователя.
    for(User [] u : united_user_list){
        u.time = {SELECT TIME FROM PERSON WHERE USER_ID = :u.userId};
    }
    return united_user_list;
}

```

При выполнении операции поиска пользователей по списку BSSID, после создания объединенного множества идентификаторов пользователей происходит обращение таблице PERSON за временем отметки, что требует дополнительных дисковых операций. Решением может быть хранение времени создания отметки в таблице WIRELESS_SPOT вместе с идентификаторами пользователей.

Отметим, что во многих случаях, и для схемы А в частности, операция создания отметки и операция поиска пользователей часто выполняются последовательно в рамках одного запроса. Как нетрудно заметить, эти операции требуют обращения к одним и тем же строчкам таблицы WIRELESS_SPOT. Поэтому целесообразно объединить эти операции, избежав лишних обращений к базе данных.

В системе геоотметки, работающей по схеме В (используется в демонстрационной реализации), используется схема изображенная на рис. 6.

PERSON		
	user_id	varchar
	BSSID_list	varchar




WIRELESS_SPOT		
	BSSID	varchar
	user_list	varchar

Рис. 6. Схема базы данных, используемая в системе, работающей по схеме В. user_id - идентификатор пользователя, BSSID_list - список беспроводных точек доступа, user_list - строка, содержащая список идентификаторов пользователей, находящихся вблизи беспроводной точки доступа, и сроки действия их отметок.

Для наибольшей производительности в поле user_list вместе с каждым идентификатором пользователя хранится закодированное значение времени, после которого отметка становится не актуальной. Базовые операции происходят аналогичным образом. При выполнении операции поиска пользователей по списку BSSID учитывается срок действия отметки: пользователи, срок действия отметок которых истек, не попадают в выборку. Имеется также сервисная операция, удаляющая отметки пользователей, срок действия которых завершился. Она представляет собой проход по всем строкам таблицы WIRELESS_SPOT и вычеркивание из списков user_list идентификаторов пользователей, срок действия отметки которых истек. Операция нужна, чтобы избежать разрастания списков user_list в таблице WIRELESS_SPOT.

Для систем геоотметки, использующих модель данных с более чем одной отметкой на одного пользователя (рис. 2 б), можно использовать схему, изображенную на рис. 7. Роль первичного ключа в таблице CHECKIN играют идентификатор пользователя и время отметки. Базовые операции происходят аналогично.

Предлагаемый способ организации хранилища данных легко обобщается на случай, если в архитектуре используются видимые и невидимые отметки. Одним из способов является хранение данных по видимым и невидимым отметкам в различных таблицах, как показано на рис. 8.

CHECKIN		
	user_id	varchar
	time	date
	BSSID_list	varchar
	ttd	integer





WIRELESS_SPOT		
	BSSID	varchar
	checkin_list	varchar

Рис. 7. Схема базы данных, позволяющая хранить более одной отметки местоположения на каждого пользователя. user_id - идентификатор пользователя, time - время отметки, ttl - срок действия отметки, BSSID_list - список беспроводных точек доступа, checkin_list - строка, содержащая список отметок, совершенных в радиусе действия беспроводной точки доступа; строка checkin_list содержит идентификаторы пользователей, времена и сроки действия их отметок.

VISIBLE_CHECKIN		
	user_id	varchar
	time	date
	BSSID_list	varchar
	(ttl)	integer

WIRELESS_SPOT		
	BSSID	varchar
	visible_checkin_list	varchar
	invisible_checkin_list	varchar


INVISIBLE_CHECKIN		
	user_id	varchar
	time	date
	BSSID_list	varchar

Рис. 8 . Возможная схема базы данных, обеспечивающая хранение отметок различных типов - видимых и невидимых. user_id - идентификатор пользователя, time - время отметки, ttl - срок действия отметки, BSSID_list - список видимых пользователем беспроводных точек доступа, visible_checkin_list - список видимых отметок, invisible_checkin_list - список невидимых отметок.

Глава 3. Демонстрационная реализация системы геоотметки

Для демонстрационной реализации системы геоотметки, была выбрана схема В, описание которой дается в разделе 2.5. Клиентское приложение реализовано для платформы Android. Приложение работает с аккаунтами из социальной сети Facebook. Для входа в Facebook, а также получения от Facebook данных, необходимых для отображения списка пользователей, использовалась предлагаемая Facebook библиотека Facebook SDK for Android [8]. Снимки экрана клиентского приложения приведены на рис. 9. Скомпилированное приложение и его исходный код лежат в репозитории: https://github.com/shabalindm/SSID_Checkin_Demo, ссылка для скачивания скомпилированного приложения: https://github.com/shabalindm/SSID_Checkin_Demo/blob/master/release/SSID_Checkin_Demo.apk.

Серверная часть написана на java и выполняется под управлением сервера приложений. Взаимодействие с базой данных происходит с использованием технологии JDBC через интерфейс javax.sql.DataSource. Реализация этого интерфейса обеспечивает поддержание пула соединений с базой данных и пула подготовленных запросов (PreparedStatement), обеспечивая эффективную работу в многопоточном режиме. Схема базы данных представлена на рис. 6. Сервер запущен на бесплатном хостинге OpenShift [9], с использованием СУБД PostgreSQL. Исходный код серверной части лежит в репозитории: https://github.com/shabalindm/SSID_Checkin_Demo_S.

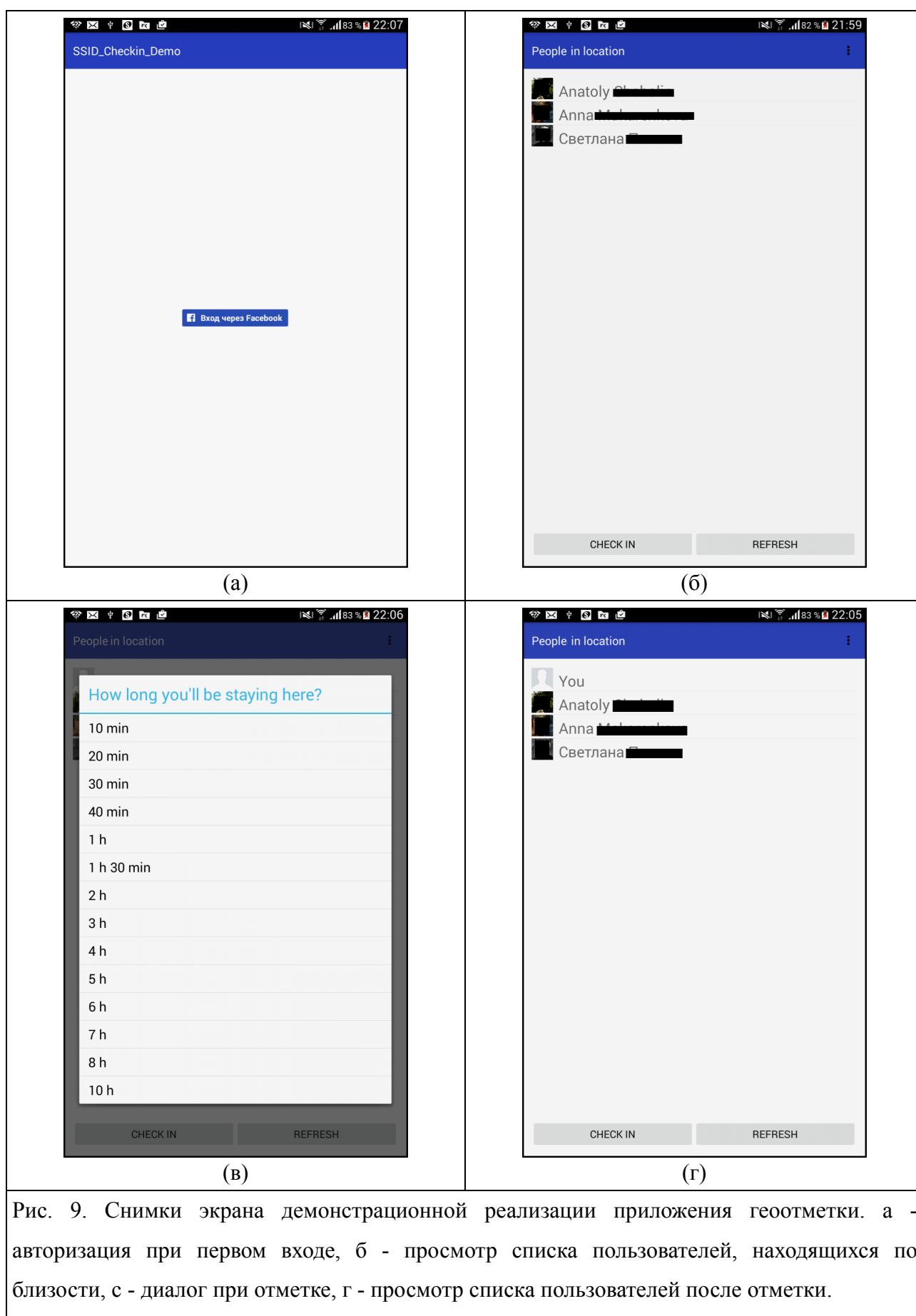


Рис. 9. Снимки экрана демонстрационной реализации приложения геоотметки. а - авторизация при первом входе, б - просмотр списка пользователей, находящихся по близости, с - диалог при отметке, г - просмотр списка пользователей после отметки.

Список литературы

1. Schilit B., Adams N., Want R. Context-aware computing applications. // IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94). - Santa Cruz, CA, US: 1994. - P. 89–101.
2. Anind K. Dey. Understanding and Using Context // Personal Ubiquitous Computing. - 2001. - №5 (1). - P. 4-7.
3. Namiot D., Schneps-Schneppe M. About location-aware mobile messages // Next Generation Mobile Applications, Services and Technologies (NGMAST), 2011 5th International Conference. - Cardiff: 2011. -P. 48–53.
4. Namiot D. On Indoor Positioning // International Journal of Open Information Technologies. - 2015. - №3.3. - P. 23-26.
5. IEEE 802.11 // Википедия. [205—2016]. Дата обновления: 03.04.2016. URL: <http://ru.wikipedia.org/?oldid=77553086> (дата обращения: 23.05.2016).
5. Wi-Fi positioning system // Wikipedia, The Free Encyclopedia. [2008 - 2016]. Дата обновления: 5.02.2015, URL: https://en.wikipedia.org/w/index.php?title=Wi-Fi_positioning_system&oldid=718896484 (дата обращения: 23.05.2016).
6. Namiot D., Sneys-Sneppe M. Context-aware data discovery // Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on. — IEEE Conference, 2012. — P. 134–141.
7. Namiot D., Schneps-Schneppe M. Proximity as a service // Future Internet Communications (BCFIC), 2012 2nd Baltic Congress on. - Vilnius: 2012. - P. 199-205.
8. Facebook SDK для Android URL: <https://developers.facebook.com/docs/android> (дата обращения: 23.05.2016).
9. OpenShift: PaaS by Red Hat, Built on Docker and Kubernetes URL: <https://www.openshift.com/> (дата обращения: 23.05.2016).