



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ
ФАКУЛТЕТ КОМПЮТЪРНИ СИСТЕМИ И
УПРАВЛЕНИЕ

Курсов проект
по Програмиране за мобилни устройства
„InstantMessage“

Изготвил:

Теодор Миленов
ФКСУ, 4-ти курс
Фак. номер: 121212186, 54гр.
Гр. София , 04.13.2016

Проверил:

гл.ас. Антония Ташева

Съдържание:

Обща информация	3
Условие на заданието	3
Цел на приложението	3
Обяснение на проблема	3
Съществуващи решения	3
Facebook Messenger	3
Позитивни	3
Негативни	4
Viber	4
Позитивни	4
Негативни	4
Цели и задачи	4
Проектиране	5
Потребители	5
Съхранение на информацията	5
Комуникация	5
Архитектура	6
Клиентско приложение	6
Сървърно приложение	7
База данни	7
Реализация	8
Android приложение. Основни компоненти.....	8
MainActivity	8
ChatActivity	8
Java Spring приложение. Google App Engine.	8
Бъдещи подобрения на приложението.	9
Източници	11

Обща информация

Условие на заданието

Да се реализира мобилно приложение „чат“.

Цел на приложението

Да се предостави възможност на потребителите на приложението за комуникация, независима от местоположението или вида устройство. Времето за предаване на съобщения да се сведе до минимум. Възможност за използване на Google акаунт за регистрация. Потребителски интерфейс, който да привлича вниманието на потребителя.

Обяснение на проблема

За изпълнение на изискванията на пазара относно „чат“ приложенията, е необходим внимателен избор на целевата група потребители, както и перфектен набор от качества на програмното решение като красив и лесен за употреба потребителски интерфейс, надеждност и бързодействие. Понеже функционалността за предаване на съобщения от разстояние съществува от векове, е нужно нещо много повече от реализацията ѝ. Именно това може да бъде перфектен потребителски интерфейс с много възможности за разнообразяване на комуникацията и безкомпромисна достъпност.

Съществуващи решения

Съществуват хиляди имплементации на функционалност тип „чат“. Наистина това е едно от най-големите желания на потребителите по света. Някои от тези приложения изпъкват пред други и в последно време са най-използвани. Нека разгледаме две от тях с техните добри и лоши страни.

Facebook Messenger

<https://play.google.com/store/apps/details?id=com.facebook.orca&hl=bg>

Позитивни

- Директна интеграция с профил във Facebook.
- Разнообразие от начини за изразяване (емотикони, стикери).
- Възможност за изпращане на снимки, както от галерия, така и директно чрез камерата на устройството.
- Възможност за изпращане на звукозапис и видеозапис.
- Възможност за заглушаване на разговори.
- Възможност за провеждане на разговори от повече от двама души.

Негативни

- Приложението е „тежко“. Това ще рече, че заема прекалено много ресурси от устройството.
- Изисква Facebook регистрация.
- Прекалено много функционалност, която усложнява потребителския интерфейс.

Viber

<http://www.viber.com>

Позитивни

- Приложението е изключително леко и лесно за употреба.
- Разнообразие от начини за изразяване (емотикони, стикери).
- Възможност за изпращане на снимки, както от галерия, така и директно чрез камерата на устройството.
- Възможност за изпращане на звукозапис и видеозапис.
- Поддръжка на телефонни разговори и телефонни видеоразговори.
- Иновативен начин за известяване при съобщение.
- Добавяне на контакти от адресния списък на мобилното устройство.
- Известие при присъединил се към Viber контакт от адресния списък на устройството.
- Реализация за различни устройства – мобилни и десктоп.

Негативни

- Обвързване с телефонния номер на устройството.

Цели и задачи

- Създаване на функционалност за предаване на съобщения между мобилни устройства.
- Лесен за употреба и красив потребителски интерфейс.
- Използване на съществуващ Google акаунт за регистрация в приложението.
- Бързодействие.
- Минимално използване на ресурсите на устройството.
- Приложението да е съвместимо с възможно най-много версии на операционната система.
- Известяване при получаване на съобщение.

Проектиране

Потребители

Приложението е насочено към голяма група потребители поради неговия характер. Целта е да привлече внимание с дизайн и бързодействие, като същевременно не обвързва клиентите с редица регистрации. Това ще бъде новост за потребителите, защото са свикнали преди използване на дадено приложение за чат, да преминат през процедура за автентикация и предоставяне на лични данни. Решението за премахване на тази стъпка ще олекоти потребителското преживяване, като ще допринесе и за повече печалба от приложението.

Съхранение на информацията

За да може да се поддържа голям брой потребители, е нужна база данни. Това ще допринесе за сигурността на информацията, както и правилното ѝ подреждане логически в системата за „чат“. Също така базата трябва да бъде изключително надеждна и достъпна.

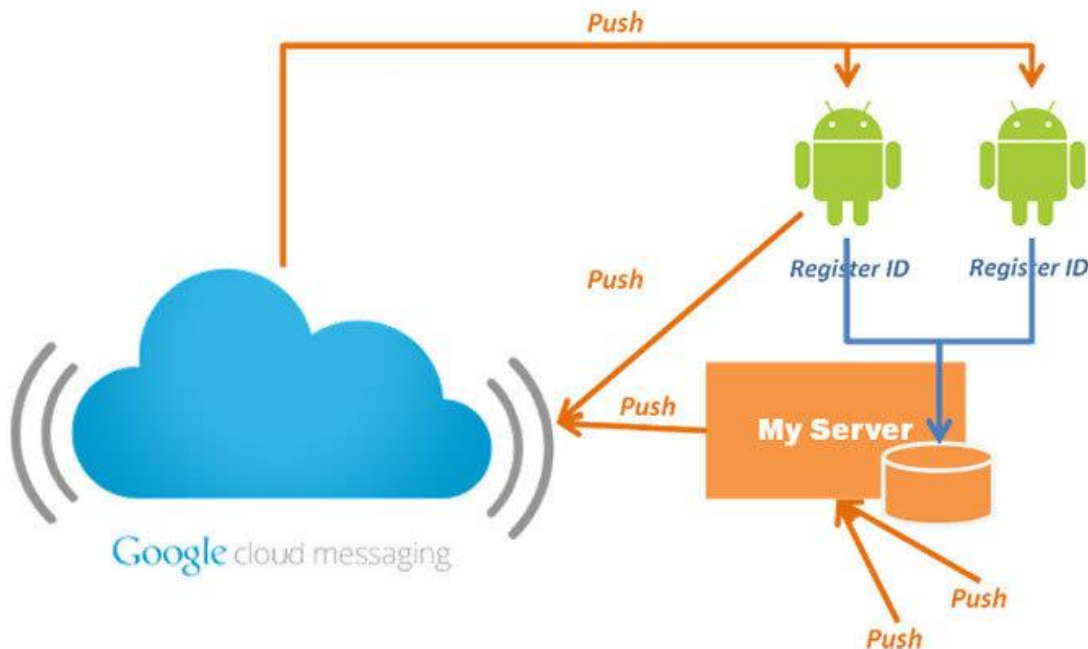
Решението за база данни е тя да бъде облачна услуга. Приложението ще трябва да се доверява на някой, който предоставя тази услуга, затова и е избран Google Cloud Datastore. С тази облачна услуга се предоставя надеждно решение на проблема със съхранението на информацията.

За да бъде използвана Google Cloud Datastore услугата, е нужен developer акаунт в Google. При цена от 25\$, се получава достъп до множество подобни услуги, които ще бъдат много полезни в реализацията на приложението. Голямото им предимство е това, че с много малко усилие се постига голям резултат относно достъпността на приложението и надеждното съхранение на информацията. Без съмнение това ще бъде изборът за решение.

Комуникация

За изпълнение на целите на приложението, може би най-важната стъпка е да се избере правилен начин за комуникация и предаване на съобщения. Изискванията са бързодействие и надеждност. Дори при малко забавяне на съобщение, потребителят ще бъде много разочарован и е възможно да спре да използва приложението, както и да не го препоръчва на други потребители. Също така информацията, предавана чрез съобщенията, трябва да има гаранция за конфиденциалност. Това ще рече, че потребителят трябва да бъде убеден по някакъв начин, че съобщенията му ще пристигнат до крайната си дестинация и че никой няма да има достъп до тях, освен получателя.

При положение, че вече имаме акаунт за разработчик в Google, най-доброто решение е да се използва Google Cloud Messaging (GCM). Това е облачна услуга, предоставена от Google, чрез която много лесно може да се имплементира функционалност за предаване на съобщения между мобилни устройства.



Архитектурата на GCM приложение е проста:

- ✓ Google сървър поема отговорността за предаване и разпределение на съобщения.
- ✓ Приложен сървър обработва всички съобщения, като получава или изпраща такива от и към Google услугата.
- ✓ Клиентското приложение комуникира единствено с GCM.

Какво предлага GCM?

- Сигурност
- Надеждност
- Бързодействие
- Лесна интеграция

Архитектура

Клиентско приложение

Клиентското приложение ще представлява Android приложение. Главните цели при имплементацията му са красив и удобен, лесен за употреба потребителски дизайн. Същевременно не трябва да е тежък за операционната система, защото целевата група устройства е обширна. Версиите Android, които ще поддържа приложението са всички от Android 2.3.2 Gingerbread до Android 6.0 M.

За изпълнението на функционалността за комуникация ще се използва GCM. Приложението ще изпраща съобщения към GCM сървър, а от там те ще се препращат към сървърното приложение. Поради ограничения, наложени от GCM спецификацията, съобщенията не бива да надвишават размер от 4kB. Това означава, че съобщения със снимки или записи няма да бъдат възможни.

Сървърно приложение

Сървърното приложение ще се грижи за запазването на цялата информация, а именно потребителски имена и техните имейли. След като GCM получи дадено съобщение от клиентско приложение, то бива вкарано в опашка заедно с всички други съобщения, които чакат своя ред. От GCM сървъра съобщението се изпраща към сървърното приложение, където първо за запазва в базата данни с полета за изпращач и получател. Сървърното приложение на свой ред препраща съобщението към GCM, от където то достига и своят получател. Целият този процес се случва асинхронно. Това ще рече, че няма да блокира използването на Android приложението. При получаване на ново съобщение, приложението ще нотифицира потребителя за него по приетият вече стандарт в Android приложенията.

База данни

Базата данни ще се състои от една таблица, в която се съхраняват всички регистрации в приложението. При всяко препращане на съобщение, тази таблица ще се използва за извличане на регистрационния номер на всяко приложение. Тъй като всяко клиентско приложение ще има еднозначна връзка с имейл на потребител, ще ни е нужно да пазим и имейли.

Contacts
id : integer
email: varchar
regId: varchar

Като бъдещо разширение на базата и предлаганата функционалност от приложението, може да се имплементира и друга таблица, която да съдържа информация за аватари(картинки) за всеки потребител.

Реализация

Android приложение. Основни компоненти.

MainActivity

Първото, което потребителят ще вижда при стартиране на приложението е MainActivity. То ще представлява отправна точка за всички операции с приложението. Състои се от ActionBar, разположен най-отгоре в изгледа. В него са разположени както следва:

- Заглавие на приложението
- Подзаглавие със съдържание имейл на Google акаунта на потребителя.
- Бутон за добавяне на нов контакт.
- Бутон опции.

Централно разположен в главния изглед се намира списъкът с контакти.

ChatActivity

При натискане върху някой от контактите в главният изглед, се отваря нов прозорец – ChatActivity. Той имплементира функционалността за комуникация чрез съобщения. Тук, както и в MainActivity, се вижда ActionBar със следното съдържание:

- Заглавие с името на контакта, с който се осъществява комуникацията.
- Подзаглавие със статусът на контакта – online, offline
- Бутон Edit, с който може да се редактира името на контакта, но не и имейла му.

Централно разположен е прозорец, в който се вижда историята на чата с всяко съобщение. Най-отдолу е текстово поле, служещо за въвеждане на ново съобщение.

Java Spring приложение. Google App Engine.

Технологията за имплементация на сървърното приложение е Spring. Представлява монолитен слой от бизнес логика, предоставяща услуги за регистриране, изтриване на регистрация и изпращане на съобщение към GCM. Регистрирани са следните услуги:

- /register – Регистрира дадено клиентско приложение.

Примерна HTTP заявка:

POST /register HTTP/1.1

Host: localhost:8888

Content-Type: application/x-www-form-urlencoded

Cache-Control: no-cache

Postman-Token: b567d7ca-8dfb-afe4-d5b3-4fafc33da735

senderEmail=fddasd%40abv.bg®Id=12345

- /unregister – Изтрива дадено клиентско приложение по подаден имейл.

Примерна HTTP заявка:

```
POST /unregister HTTP/1.1
Host: localhost:8888
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache
Postman-Token: b567d7ca-8dfb-afe4-d5b3-4fafc33da735
senderEmail=fddasd%40abv.bg
```

- /send – Изпраща съобщение от изпращач към получател.

Примерна HTTP заявка:

```
POST /send HTTP/1.1
Host: localhost:8888
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache
Postman-Token: 5f5b84a6-547b-5de8-0fd9-6dfaef064d61
message=12312312312&senderEmail=fddasd%40abv.bg&receiverEmail=asdasd%40abv.bg
```

Сървърното приложение е разположено на Google App Engine. Тази услуга предоставя много удобен начин за наблюдение и статистика на всички приложения, които са в работен режим. При увеличаване на броя заявки обаче се налага заплащане за услугата.

Бъдещи подобрения на приложението.

Основната идея на приложението за опростен дизайн и бързодействие, може да се развие още, до достигане до безупречно потребителско преживяване примесено с безотказна функционалност. Най-добрата му черта, а именно елементарната концепция, е това, което ще привлече вниманието на потребителя. На фона на сегашните известни чат приложения, InstantMessage предоставя просто размяна на съобщения. Без допълнителни обвързвания с телефонен номер или даване на достъп до лични данни. Все пак за в бъдеще могат да се имплементират подобрения:

- Разширяване на базата данни, с информация за аватар на потребител.
- Функционалност за самостоятелно задаване на личен имейл от потребителя.
- Подобряване на cloud услугата, като се мигрира от Google към лични сървърни машини и имплементации за препращане на съобщения.

- Опция за намиране на най-близките регистрирани потребители до устройството.
Използване на Google Maps API.

Източници

<https://developers.google.com/cloud-messaging/>

<https://bg.wikipedia.org/wiki/%D0%A7%D0%B0%D1%82>

<https://cloud.google.com/appengine/>

<https://spring.io/>

<https://cloud.google.com/datastore/>