

**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ
КЪМ ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

Д И П Л О М Н А Р А Б О Т А

Тема: Разработка на B-Chat

Дипломант:

Константин Костов

Научен ръководител:

маг. инж. Кирил Митов

С О Ф И Я

2 0 1 6

Съдържание

3.1 Проектиране на софтуера	32
3.1 Кратък преглед на модулите	32
3.1.2 Основни екрани	32

Увод

Чатът е система, която позволява на реално време, използвайки лесно достъпни интерфейси. Това е вид на интернет онлайн чат, който се отличава със своята простота и достъпност за потребителите, които не желаят да се учат да използват специализиран софтуер за чат.

Непрестанният достъп до интернет, който мобилните ни устройства ни предлагат промени изцяло представата ни за съвременна комуникация. Мобилните ни оператори ни предлагат все по-атрактивни планове с огромен брой включени минути и съобщения, за да привлекат вниманието ни. Въпреки това много от нас продължават искат да се свързват с приятелите си на максимално ниска цена.

Водещите мобилни платформи поддържат много атрактивни предложения, които ни предоставят възможността да чатим и говорим с приятелите си безплатно през Wi-Fi. По този начин не само имаме постоянен достъп до тях независимо къде се намираме, но също така това не се отразява на месечната ни сметка.

Целта на настоящата дипломна работа е да се разработи приложение, което да помага при свързването на различни хора, интегриран в Android, с който хората могат да си пишат, да четат постове, да срещнат нови хора, да участват в групи. Всеки потребител може да споделя видеота, картинки и някакви новини.

I глава

Проучвателна част. Преглед на съществуващи продукти, развойни среди и развойни средства.

Първоначално идеята на приложението беше да се изработи business chat, в който директора да може да пише на своите работодатели и да слага постове. Дипломната работа има за цел всеки да качва някакво съдържание, то да бъде коментирано под формата на дискусия. Всеки потребител може да избира тема, като темата може да бъде различна от бизнес, както беше в началото. Темата може да бъде работа, училище и други.

SMS-ите наистина са полезна услуга: понякога е много по-лесно нещо да се каже с текст, отколкото на глас. Освен това, в определени случаи просто не е удобно да се обадим ако, например, човекът от другата страна е в офиса или в час. Само, обаче, тези кратки съобщения да нямаха навика да увеличават толкова месечната ни сметка!

Нека се върнем малко назад във времето преди 10 години и се замислим къде уеб програмирането например е било. Програμισите са използвали езици като PHP/Ruby на беккенда, които свързват релационната база данни (като MySQL) и фронт енда бил изпълстен с javascript файлове. Много от новите технологии използват cloud услуга, и не е необичайно разработчиците да се възползват от тези услуги, като AWS, Amazon S3, Heroku и други.

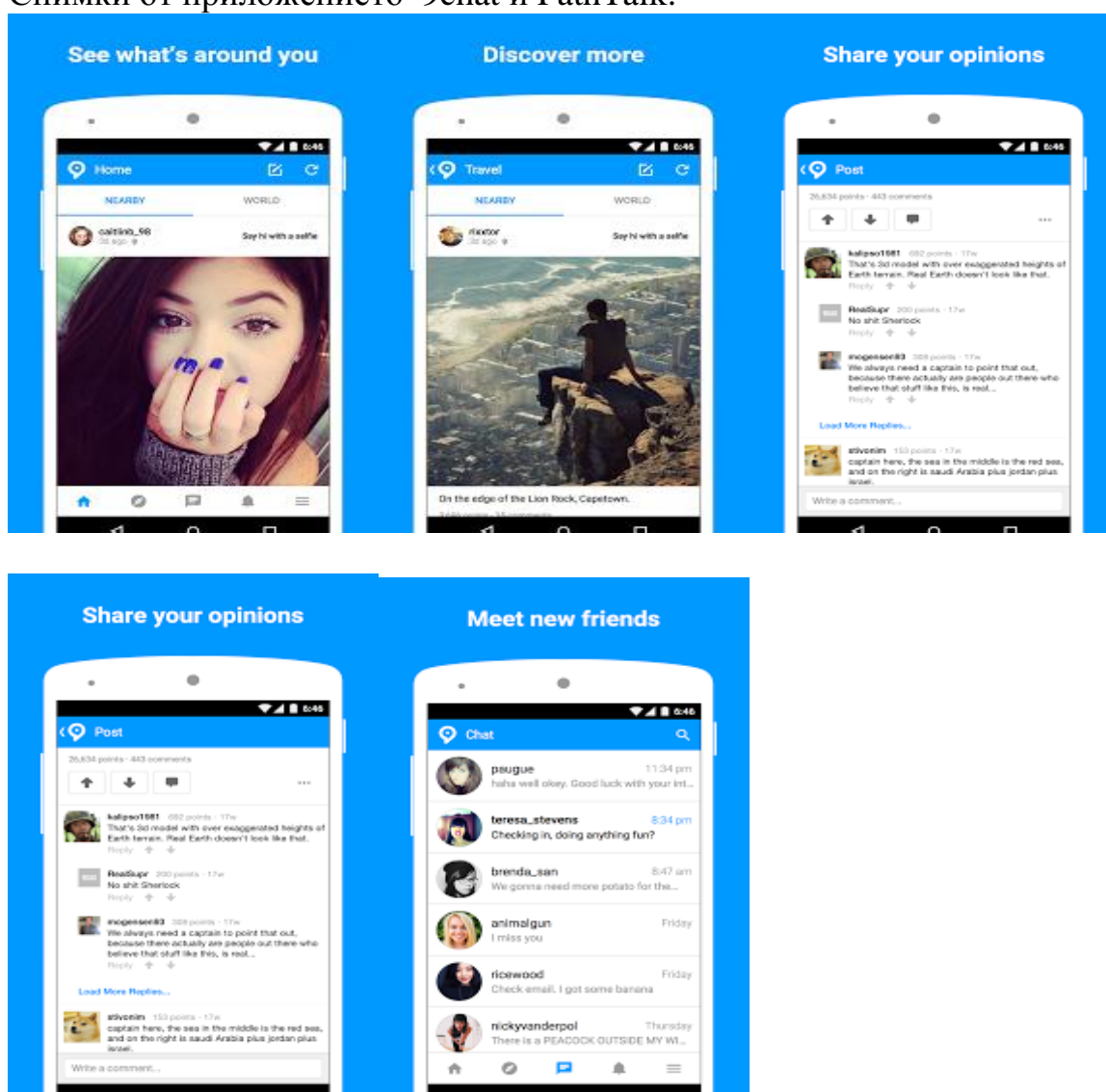
В рамките проучването бяха разгледани две такива приложения: 9chat и Path Talk.

1.1 Подобни приложения

9Chat андроид приложение

9Chat е андроид приложение, което помага на хора да споделят и четат различни истории, да си чатят с други потребители. Хората могат да влязат и да напишат някой пост и другите потребители да влязат и да го коментират. Разбира се има и чат, изпращане на картинка. В разгледаното приложение няма статистики, съобщение, което ще се самоунищожи и групов чат.

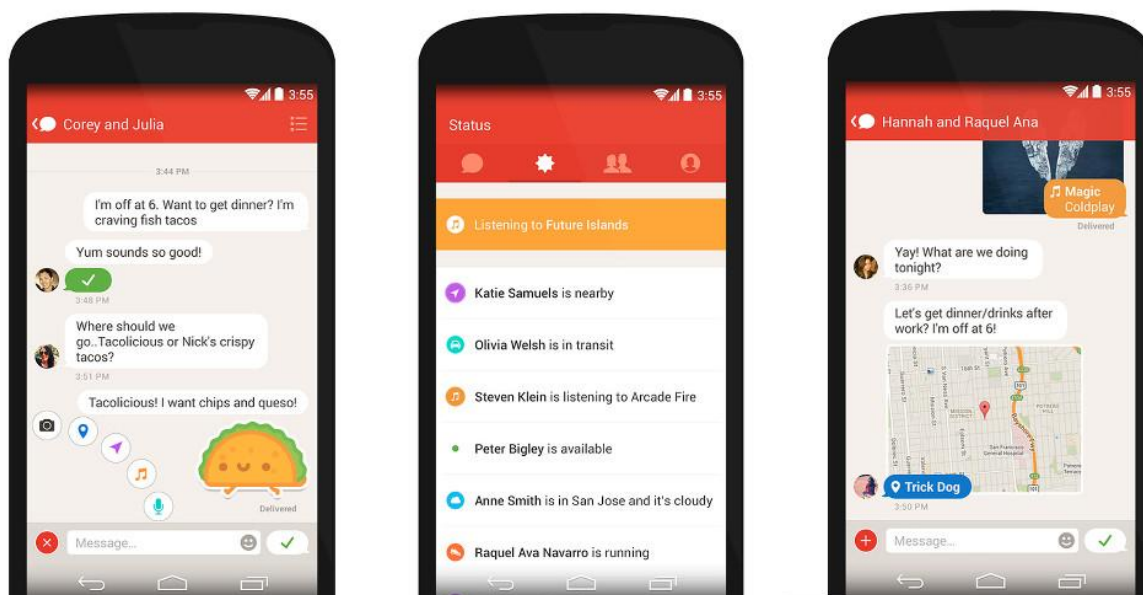
Снимки от приложението 9chat и PathTalk:



9Chat като цяло събира хиляди потребители с общи интереси и в

1.2 Path Talk

Path Talk е сравнително непознато, но доста функционално приложение. Освен че предлага пращане на безплатни съобщения, услугата уведомява постоянно приятелите ви за статуса ви – например в случай, че разполагате с малко батерия. Освен текстови, Path Talk позволява и изпращането на кратки гласови съобщения. Най-голямото преимущество на Path Talk, обаче, е, че пази данните ви сигурни: всяко пратено съобщение се съхранява в сървърите на приложението само в продължение на 24 часа и след това автоматично се изтрива.



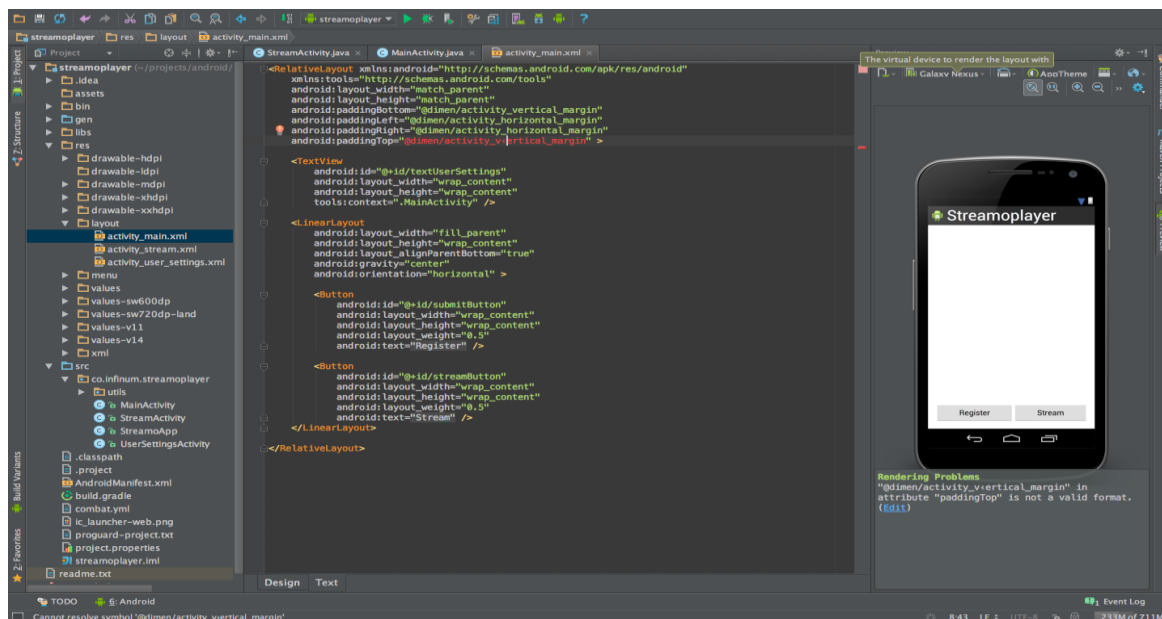
1.3 Езици и среди за програмиране ,които могат да бъдат използвани за построяване на приложението

1.3.1 Java-Android Studio

Това е Android фокусирани IDE, проектиран специално за развитието на Android. Тя стартира на 16 май 2013 г., по време на Google I / O 2013 ежегодно събитие. Android студио съдържа всички Android инструменти SDK за проектиране, изпитване, отстраняване на грешки и профила на приложението си.

„Android studio“ предлага :

- редактор
- примерни кодове на прости програми
- интеграция на Version Control System (Git)
- създаване на виртуални устройства с всякакви размери и форми (телевизори, телефони, таблети и др.)
- възможност за създаване на различни .apk файлове с различни добавки и версии на базата на един и същ проект



1.3.2 Java-Eclipse

ADT

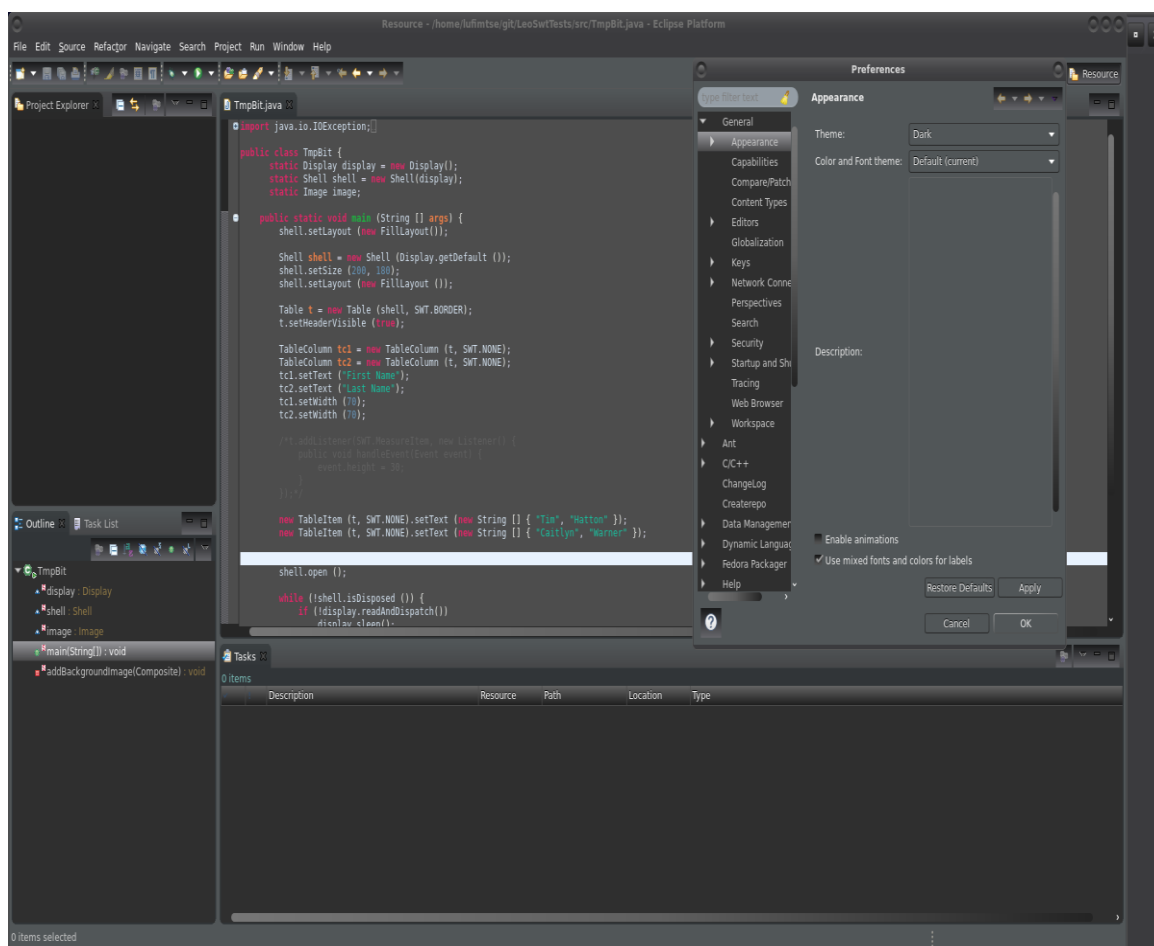
ADT (Android Development Tools) е развойна среда под формата на плъгин за Eclipse. Тя се предлага безплатно и съдържа набор от полезни за разработването инструменти, към които и визуализация, и улеснена работа с UI (User Interface), с цел по-добро разработване и по-добре оформен дизайн.

Някои от предлаганите инструменти са :

- автоматично дописване на код
- анализатор за XML
- анализатор на JAVA
- поддръжка на всички основни Java библиотеки
- debugger

- Dalvik Virtual Machine – виртуална машина за Android
- компилатор
- добавка за Version Control Systems (Git).

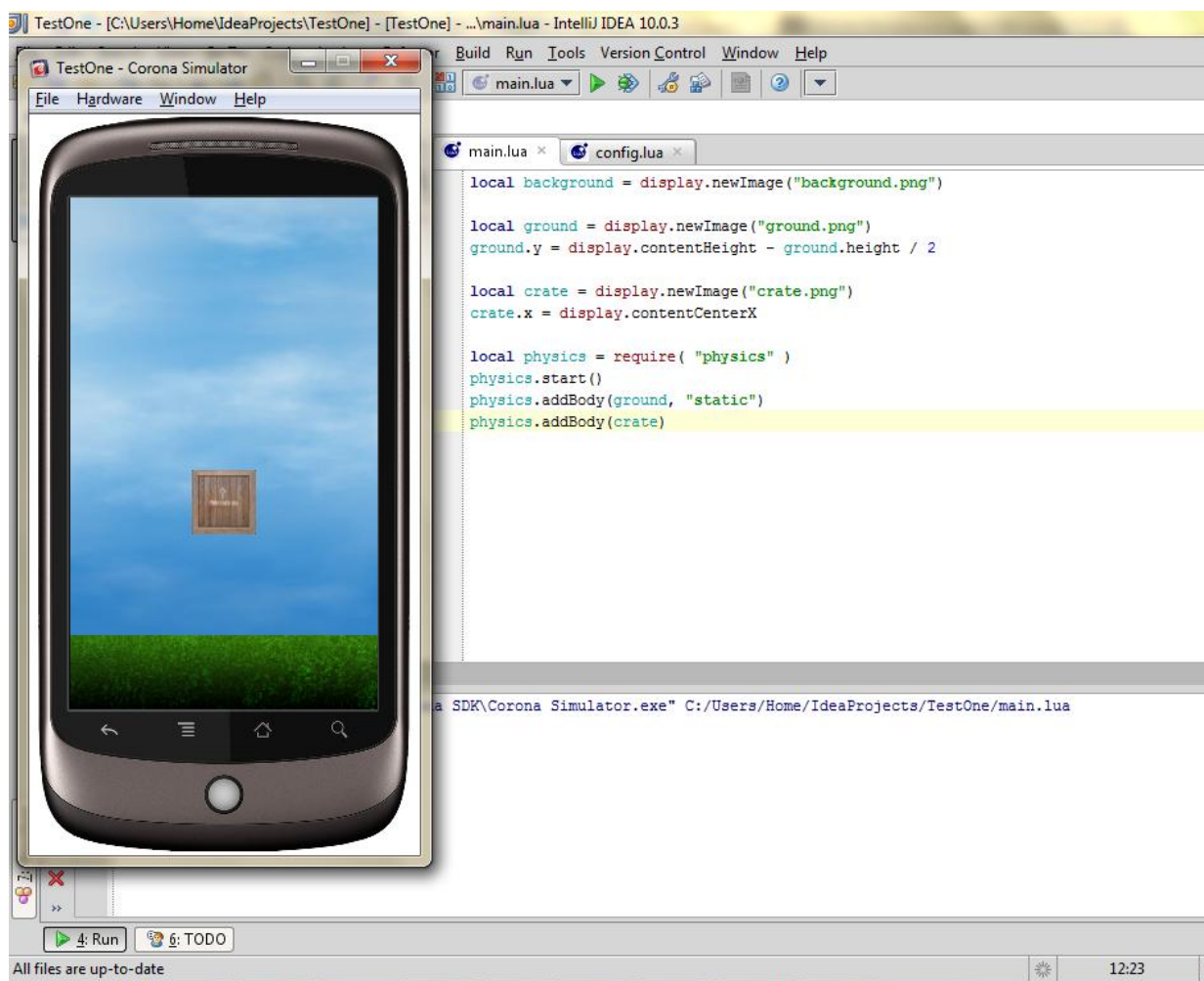
разработка на други програмни езици също така могат да напишат свой плъгин модули.



Ето как изглежда eclipse.

Corona

При създаване на приложения за Андроид можем да използваме Corona SDK. Аз до този момент нямам опит с Corona, но се постарах да проуча какво представлява.



Corona е високо ниво SDK, което използва LUA programming language. Ето как изглежда LUA кода ,който прави също като Java кода.

```
for i=1,5 do
  for j=1,i do
    io.write("*")
  end
  io.write("\n")
end
1, 110/255, 110/255 )
```

Почти всичко в Corona е представено чрез OpenGL.

Corona притежава сложен емулатор, който ви позволява да стартирате програмата веднага, без да се налага да компилирате вашия код. Когато искате да създадете вашия android.apk файл трябва да стартирате Corona е разработена предимно за игри и включва библиотеки за audio, game networking и 2D physics engine. Почти всичко в Corona е представено чрез OpenGL. Corona е безплатна за теглене и използване

Phonegap



Ако знаете Html,CSS, и най-важното Javascript можете да създадете Android приложение.Phonegap е базиран на Apache's Cordova проект.Basically it creates a webview which you can then populate and manipulate using Javascript. The web app can interact with the various device features, just like a native app,

by referencing the cordova.js file to get the API bindings. Native functions, които PhoneGap поддържа включват акселерометър, камера, настоящо местоположение.

1.3 Избор на бекенд

Изборът на бекенд не е лесно.

BaaS- backend as a service;

BaaS ви позволява да си осигурите база данни, която дава разрешение на потребителите да добавят обекти в cloud. В миналото разработчиците са създавали бекенд като са използвали технологии като Ruby и PHP.

Това често се оказва трудна задача, защото отнемало време, ресурси и изисквало специални умения за да се създаде нещо което работи гладко (ефективно).

Изборът на бекенд зависи от лични предпочитания и какво се целите да постигнете с приложението ви.

Parse



Parse имат идеален план ако сте нови в BaaS и трябва да си завършите бързо приложението ви .

Документацията е добре представена и лесно може да се намери онова, което ни трябва. Платформата е лесно използвана за начинаещи. Съществуват много полезни приложения, от които можем да се учим, ако сте нови. Parse е доста добър защото съществуват доста 3d party libraries to utilise. Звучи доста добре, но ако потръгне приложението и съберем доста потребители, ще трябва да плащаме за услугата, което е и единствения минус.

Фиг. 1.3.2 Kumulos



Kumulos поддържа IOS, OSX, PHP и Android и е напълно безплатна за употреба, като същевременно си приложение проект се разработва. Когато стартирате приложението си цената е \$ 10 на приложението, на месец с един куп допълнителни функции за използване според вашите изисквания.

Ако сте нов в БАС играта Kumulos ще ви помогне да започнете бързо. Интуитивният интерфейс за създаване на уеб услуги позволява на приложения да комуникират с услугата Kumulos за изпълнение на основни операции CRUD на вашата база данни. Това може да бъде много полезно,

когато се движат на данни от съществуващите си сървъри към платформата Kumulos. Този вид достъп е наистина важно, ако сте изправени поддръжка на системата и мащабиране умора, тъй като това ще ви позволи да прехвърлите данните си бързо и лесно.

1.3.3 Kinvey



Kinvey е голяма платформа, в която нещата се движат бързо. Kinvey ви дава възможност да имат 200 активни потребители на месец, напълно безплатно. Ако се превиши тази граница, който може да бъде ранен признак на успех, а след това ще стане платен.

Kinvey е особено фокусира върху корпоративните мобилни приложения и това е отразено в цената. Екипите за поддръжка Kinvey са бързи, отзивчиви и ако се нуждаете от помощ са на разположение. Те също така са страстни за да ти помогне да успее в търговската - ако сте агенция (или на свободна практика), има и удобен инструмент, който ви дава възможност за ценовата до проекти за развитие на приложението въз основа на вашите игрални изисквания и оценки за ползване.

1.4 Развойни средства

1.4.1 Основни компоненти на Android приложение

Activity – Представя тип слой от приложението например екран, който потребителя ще вижда. Едно приложение може да има няколко слоя активности и може да превключва между тях по време на работа на приложението. Потребителския интерфейс на тези слоеве е изграден от widget класове които са свързани от "android.view.View" Слоеве view се управляват от ViewGroups.

Services – Услугите ви осигуряват фоновы задачи без наличието на интерфейс. Те могат да известяват потребителя посредством notification framework в Андроид.

Content Provider – Доставя данни от приложението . Благодарение на този компонент можете да споделяте данни с други приложения. Андроид съдържа SQLite база данни, която може да служи като доставчик на тези данни.

Intents – Това са асинхронни съобщения, които позволяват приложението да иска функционалност от други услуги или дейности. Приложението може директно да извика услуга или дейност (explicit intent) или да извести “системата за регистрация на услуги и приложения на Андроид” за тези си намерения (implicit intents). Например приложението може да извика чрез Intent приложението за контакти. Приложението се регистрира посредством Intent филтър (IntentFilter). Компонента Intent е много мощен инструмент благодарение на който можете да създавате свободно свързани приложения.

Broadcast Receiver - Получава системни съобщения, както и асинхронни съобщения implicit intents (за справка виж малко по на-горе). Може да се използва за реакция при променящи се условия в системата. Приложението може да се регистрира като приемник (Broadcast Receiver) за определени събития, и може да се стартира при наличието на такова (събитие).

II глава

Изисквания към програмния продукт. Избор на езика за програмиране и софтуерните средства. Описание на структурата на програмата и базата данни

2.1. Изисквания към програмния продукт

Създаване на мобилно приложение за „Android“ платформата, което да служи за четене на публикации и лесно свързване на потребителите. Приложението трябва да може да поддържа форма за регистрация и изпращане на съобщения. Да има раздели спомагащи за по-добра организация на публикациите – „All posts“ в него се съхраняват всички публикации, „My posts“ – тук се намират всички постове, които са били създадени от потребителя. Да има опция да се изтриват съобщения след като са били изпратени. Всеки потребител да притежава статистики на базата на изпратените съобщения.

2.2 Избор на развойна среда за B-Chat

При създаването на приложението са използвани мобилната операционна система “Android”, езикът “Java” и средата “Android Studio”.

Използвал съм ги поради три причини:

Първата причина е, че “Java” е официалният език под който се разработват Android приложения, а “Android Studio” е средата, която Google препоръчва. Второ, вече от година се занимавам с Android приложения използвайки “Java” и “Android Studio”.

Натрупаният опит и познания би ми помогнал да се съсредоточа изцяло върху проектирането на приложението и така да създам един завършен продукт готов да се конкурира с водещите от този тип.

2.3 Графична структура на приложението

Графика на приложението генерирана чрез CodeIris;



2.4. Структура на приложението

2.4.1. Зависимости (dependencies) на проекта

В приложението са използвани външни библиотеки за по-разбираемо описани нека ги разделим на два вида – “com.android.support” (официални

зависимости, невлизаци в SDK-а) и “3rd party” (както от името става ясно библиотеки принадлежащи на трети лица, пуснати с лиценз за свободни използване).

“com.android” зависимости:

```
compile 'com.android.support:support-v4:23.1.1'
```

– основната библиотека

```
compile files('libs/Parse-1.4.0.jar')
```

– библиотека, която използвам като сървър

```
compile 'com.android.support:appcompat-v7:23.1.1'
```

– предлага класове съвместими и поддържащи по-стари версии на Android

```
compile 'com.android.support:design:23.1.1'
```

– помощна библиотека за дизайн, съдържаща основните практики за потребителски интерфейс в Android

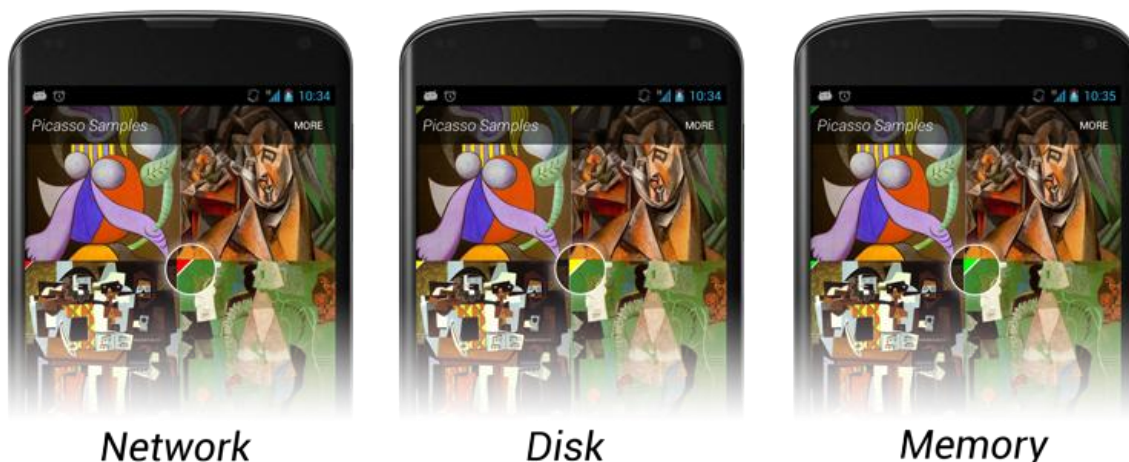
```
compile 'com.jakewharton:butterknife:7.0.1'
```

– улеснява инициализирането на изгледи чрез идентификационен номер използван в xml файловете като предлага анотации, с които това да става с доста по-малко код. Единственото изискване е да извика статичния метод на класа „ButterKnife“ – „ButterKnife#bind(activity)“;

“3rd party” зависимости:

```
'com.squareup.picasso:picasso:2.5.2'
```

За искарване на изображенията използвам 3d party библиотека `picasso`.



Защо да използвам?

Това опростява процеса на зареждане на изображения от външни адреси и дисплей на вашето приложение. Например, изтегляне на изображение от сървър, е един от най-често задача във всяко приложение. И тя се нуждае от доста по-голямо количество код, за да се постигне това чрез Android мрежа API. Чрез използването на Пикасо, можете да постигнете това с няколко реда код.

Тя не винаги е за сваляне на изображения от отдалечено място. Вие също трябва да се мисли за прилагане на изображението кеширане логика, за да се осигури безпроблемна работа на потребителя. Пикасо осигурява автоматично кеширане на изображението трансформация на снимката е скъпо афера.

Ако вашето приложение нужда се справят с такова изпълнение трансформация на изображения, трябва да сте бдителни за `OutOfMemoryException`. Пикасо се занимава с него, така че ти не трябва да го направя сам.

2.5 Избор на език за приложението

За език за програмиране беше избран Java. Java е обектно-ориентиран език за програмиране. Поради сложността на съвременните софтуерни приложения, обектно-ориентираният стил на програмиране и обектно-ориентираните езици за програмиране, в това число и Java, са предпочитани. Те осигуряват разделяне на приложението на самостоятелни модули и по този начин се постига по-голяма гъвкавост, която е изключително важна при бъдещето им развитие. Освен това се осигурява капсулация на данните и вътрешното устройство във всеки модул (наричан още клас), като достъпа до тези данни се определя от специално дефинирани интерфейси. По този начин се осигурява независимо развитие на всеки компонент от програмната система, с минимална зависимост от останалите. Плюсовете на

Сред главните предимства са:

лесната преносимост между различните платформи (софтуерни или хардуерни) - веднъж написана и компилирана, една Java-програма може да бъде стартирана на компютри независимо от архитектурата или от операционната им система. За целта е необходимо само да се инсталира виртуална машина за съответната платформа; допълнителните действия, извършвани от виртуалната машина като освобождаване на паметта от обекти(класове) които не се използват(Garbage collector), проверка за размерността на масивите; възможността за контрол на правата на потребителя на ниво виртуална машина първоначално заделяна на heap, част от паметта резервирана за джава висока степен на сигурност поради факта, че програмистите не работят директно с паметта и др.

2.6 Избор на Бекенд- Parse

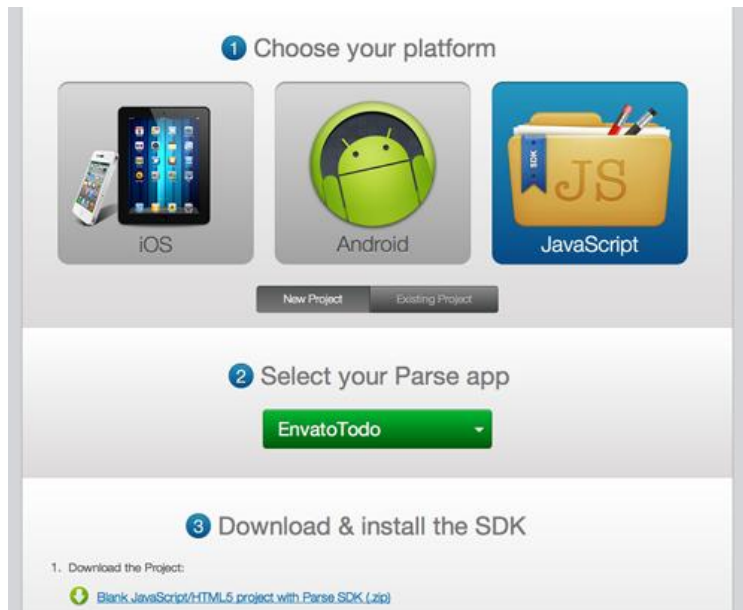
Parse е система за управление на данни, cloud- базирана, която ви позволява бързо да създадете уеб или мобилно приложение. Можете да добавите файла Parse Java ,с трябва да си вземете ключ за API, и да започнете да създавате "обекти" в cloud-сървъра само с няколко реда код. Parse ви освобождава от създаването на сървърния код.

В тази дипломна работа използвам Parse Java SDK. Вие не сте ограничени да използват само Java.Имате избор от библиотеки в много различни езици, включително PHP, Node JS, Javascript, C # и повече.

2.6.1 Работа с Parse

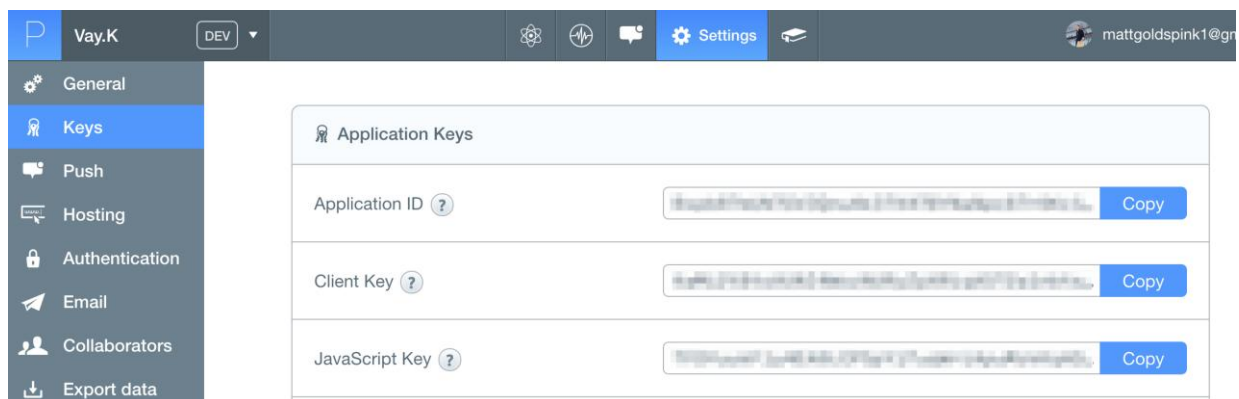
Първото нещо, което трябва да направим е да си направим акаунт в parse.com. След това да създадем нов проект и да му измислим име.





Трябва да изтеглим съответното sdk за съответния език.

Parse ни предоставя Application Id и Client key, които трябва да използваме в приложението ни.



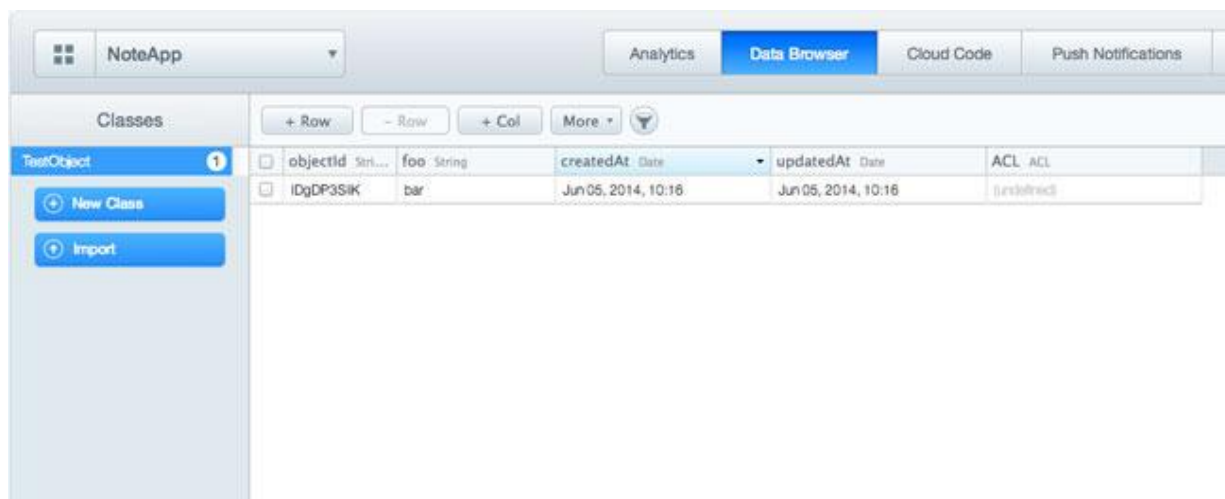
```
public void onCreate() {
    super.onCreate();
    Parse.initialize(this, APPLICATION_ID, CLIENT_KEY);
}
```



```
//Parse.initialize(this,93589519915951,41299419414)- Това са кодовете които  
сме получили  
ParseObject testObject = new ParseObject("TestObject");  
    testObject.put("foo", "bar");  
    testObject.saveInBackground();  
}
```

Чрез този код си инициализираме приложението като сме взели съответното ApplicationId и Client Key ,който сме получили от Parse.
След това създаваме обект ,който се казва TestObject.

Ето как изглежда базата ни след създаване на обекта TestObject:



```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission  
android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Трябва да добавим и съответните permission.

2.6.2 Характеристики

1. Strong Object to Object Mapping –

Parse предоставя мобилен бекенд решение,където не се нуждаете . Вие просто създавате обекта, който държи цялата информация. Същият обект ще бъде запаметен в Parse сървър , без да създавате таблици.

2. No SQLite –

Parse предоставя локална база , където ти можеш да запазваш информацията локално без да да създаващ своя база данни. Вие можете да го използвате по същия начин както използвате sqlite ,да запазвате файлове и друг тип дата.

3. User Management –

Ако приложението ви също се нуждае от потребител, Parse предоставя ParseUser клас, който може да се използва за съхраняване на данните, свързани с потребителя. Също така този клас има някои вградени методи като signUpInBackground, loginInBackground ,които можете да използвате, за да се регистрирате и да влезете в системата функционалност в приложението си, без да пишете много код.

4. Storing Audio, Video and Image files –

Можеш лесно да запаметиш своето audio, video и изображения на parse сървър използвайки ParseFile class.

5. Pricing –

Най-добрата характеристика на тази платформа е, че то е базплатно, докато приложението ви не стане хит. Parse започва да се зарежда своите потребители, когато приложението им хитове на Parse сървър повече от 30

пъти за една секунда. Това означава, ученици и програмисти могат на свободна практика да използват тази платформа, без да пишете сървърен код и такси за използване.

2.7 Изглед на обекта Chat,където са запаметени съобщенията.

objectId	ImageName	message	ACL	updatedAt	receiver	sender	ImageFile	special	createdAt	Actions
a2mg6gMez	(undefined)	Image	Public Read, 1j7L3M...	11 Feb 2016 at 17:5...	alex	kosio	androidlogin.png	(undefined)	11 Feb 2016 at 17:5...	
38p9N3P9pd	(undefined)	Click here to see 1...	Public Read, 1j7L3M...	11 Feb 2016 at 17:5...	alex	kosio	(undefined)	(undefined)	11 Feb 2016 at 17:5...	
72c1W5JST	(undefined)	Click here to see 1...	Public Read, 1j7L3M...	11 Feb 2016 at 17:5...	jedaitemple	kosio	(undefined)	(undefined)	11 Feb 2016 at 17:5...	
17f81dV7xV	(undefined)	Image	Public Read, 1j7L3M...	11 Feb 2016 at 17:4...	kosi	kosio	androidlogin.png	(undefined)	11 Feb 2016 at 17:4...	
c8Lpv6u4QH	(undefined)	Click here to see 1...	Public Read, 1j7L3M...	8 Feb 2016 at 09:41...	alex	kosio	(undefined)	(undefined)	8 Feb 2016 at 09:41...	
hP6o5C9qh	(undefined)	Click here to see 1...	Public Read, 1j7L3M...	8 Feb 2016 at 09:17...	group chat	kosio	(undefined)	(undefined)	8 Feb 2016 at 09:17...	
n55YcyM6	(undefined)	Click here to see 1...	Public Read, 1j7L3M...	8 Feb 2016 at 09:17...	group chat	kosio	(undefined)	(undefined)	8 Feb 2016 at 09:17...	
VsqQHP2D0G	(undefined)	fc	Public Read, 1j7L3M...	8 Feb 2016 at 09:16...	group chat	kosio	(undefined)	(undefined)	8 Feb 2016 at 09:16...	
EN7apTCT8	(undefined)	rr	Public Read, 1j7L3M...	5 Feb 2016 at 14:22...	group chat	kosio	(undefined)	(undefined)	5 Feb 2016 at 14:22...	
u1XHSV0Tmc	(undefined)	Special MESSAGE WIL...	Public Read, 1j7L3M...	5 Feb 2016 at 09:38...	jedaitemple	kosio	(undefined)	special	5 Feb 2016 at 09:38...	
stVQ16oCfy	(undefined)	dd	Public Read, 1j7L3M...	5 Feb 2016 at 07:51...	alex	kosio	(undefined)	(undefined)	5 Feb 2016 at 07:51...	
xNues8vfo	(undefined)	66	Public Read + Write	2 Feb 2016 at 17:51...	alex	kosio	(undefined)	(undefined)	2 Feb 2016 at 17:51...	
hRd09Ht2	(undefined)	https://www.google...	Public Read + Write	2 Feb 2016 at 17:19...	group chat	kosio	(undefined)	(undefined)	2 Feb 2016 at 17:19...	
T8tU8Kc4y	(undefined)	fff	Public Read + Write	2 Feb 2016 at 16:55...	alex	kosio	(undefined)	(undefined)	2 Feb 2016 at 16:55...	
hkorC1o0d6	(undefined)	66	Public Read + Write	2 Feb 2016 at 16:37...	group chat	kosio	(undefined)	(undefined)	2 Feb 2016 at 16:37...	
A87p2N02p	(undefined)	ffo	Public Read + Write	2 Feb 2016 at 13:38...	alex	kosio	(undefined)	(undefined)	2 Feb 2016 at 13:38...	
pKL17YH6J	(undefined)	dooffghogaaaaa11...	Public Read + Write	2 Feb 2016 at 13:26...	kosi	kosio	(undefined)	(undefined)	2 Feb 2016 at 13:26...	
7cMa940V4	(undefined)	d	Public Read + Write	2 Feb 2016 at 13:26...	kosi	kosio	(undefined)	(undefined)	2 Feb 2016 at 13:26...	
6uHvU60	(undefined)	666	Public Read + Write	2 Feb 2016 at 13:11...	group chat	kosio	(undefined)	(undefined)	2 Feb 2016 at 13:11...	
yonYvudnL	(undefined)	df	Public Read + Write	2 Feb 2016 at 13:07...	group chat	kosio	(undefined)	(undefined)	2 Feb 2016 at 13:07...	
vpp1C7D7ex	(undefined)	ff	Public Read + Write	2 Feb 2016 at 12:59...	kosio	jedaitemple	(undefined)	(undefined)	2 Feb 2016 at 12:59...	
AU40XKJ2n8	(undefined)	d	Public Read + Write	2 Feb 2016 at 12:57...	alex	kosio	(undefined)	(undefined)	2 Feb 2016 at 12:57...	
L14KFTWgJ	(undefined)	ff	Public Read + Write	2 Feb 2016 at 12:50...	alex	kosio	(undefined)	(undefined)	2 Feb 2016 at 12:50...	
qC1q9pzhT	(undefined)	Special MESSAGE WIL...	Public Read + Write	2 Feb 2016 at 12:56...	alex	kosio	(undefined)	special	2 Feb 2016 at 12:56...	
U2E1u49sY	(undefined)	izglezda doore spor...	Public Read + Write	2 Feb 2016 at 09:32...	alex	kosio	(undefined)	(undefined)	2 Feb 2016 at 09:32...	
Tvx4vAN6Q	(undefined)	sasho	Public Read + Write	2 Feb 2016 at 09:31...	alex	kosio	(undefined)	(undefined)	2 Feb 2016 at 09:31...	
Nwi0tEFQNT	(undefined)	s	Public Read + Write	2 Feb 2016 at 09:30...	alex	kosio	(undefined)	(undefined)	2 Feb 2016 at 09:30...	

2.8 Структурата на базата данни

В Parse една таблица се явява обект.

class User

Този клас държи регистрацията на всички потребители.

Parse предоставя добър обект за регистриране- User.

Това е единствения готов клас,който те предоставят,като са преценили

какво ще е необходимо за хората

objectId
EmailVerified
ACL
UpdatedAt
authData
online
username
createdAt
password
email

ObjectId (String)- някакво уникално Id ,което разпознава останалите потребители.

Email (String) - email на потребителя

Password (String)- парола ,която е така криптирана ,че никой не знае каква е

Username (String)- името на потребителя

EmailVerified- не го използвам

ACL- permission

Online-статуса на потребителя онлайн или офлайн

createdAt (Date)-- кога е създаден профила

updatedAt (Date)- - последен update на профила

class Chat

Този клас представя клас ,където са представени всички. съобщения.

objectId
ACL
createdAt
message
receiver
sender
updatedAt

ObjectId (String) - някакво уникално Id ,което разпознава останалите съобщения.

Sender (String)- името на потребителя ,който изпраща

Receiver (String) - потребителя ,който приема

createdAt (Date)- кога е създадено съобщението

updatedAt (Date)-- последен update на съобщението

Message (String) - съобщението

ACL- permission

Class Post

Този клас представя таблицата ,където са представени всички блокове.

objectId

ImageName
ACL
Author
UpdatedAt
ImageFile
title
content
createdAt

ObjectId (String) - някакво уникално Id ,което разпознава останалите блогове.

ImageName(String)- името на изображението ,който изпраща

Author (Pointer<_User>)-Пойнтер към потребителя

ImageFile(File)- самото изображение

Receiver (String) - потребителя ,който приема

createdAt (Date)- кога е създадено съобщението

updatedAt (Date)-- последен update на съобщението

ACL- permission

Content-

Title(String)- заглавието на блога

Content(String) -съдържание на поста(новина ,блог)

Class User1

ObjectId (String) - някакво уникално Id ,което разпознава останалите профилни снимки.

Sender (String)- името на потребителя на профилната снимка

createdAt (Date)- кога е качена снимката

updatedAt (Date)-- последен update на снимката

ImageFile(File)- самото изображение

ImageName(String)- името на изображението ,който изпраща

ACL- permission

objectId
ACL
ImageFile
ImageName
CreatedAt
sender
updatedAt

III глава

Описание на начина на реализация на приложението

3.1 Проектиране на софтуера

Както казах първо трябва да си вземем application id и client key, което в глава 2 съм обяснил как се взима.

3.1 Кратък преглед на модулите

Дипломната работа е андроид приложение , което е написано на android studio. Тук ще бъде разяснено как работи цялото приложение, а също така и как е проектирано.

3.1.2 Основни екрани

- Login- чрез този клас потребителите се логват ;
- Register –във този клас става регистрирането
- MainActivity- изкарва всички постове
- Beginning– инициализация на приложението
- Chat – чата с другите потребители.Възможност за приемане и изпращане на съобщения.
- SeeBlog- вижда се дадената публикация
- CreateBlog–създаване на пост;
- group – групов чат

- MainActivity – изкарва на екрана всички постове.
- Statistics – статистики изпратени получени съобщения.

3.1.2 NavigationDrawer

NavigationDrawer е панел, който показва основните опции за навигация на приложението на левия край на екрана. NavigationDrawer е скрит по-голямата част от времето, но се разкрива, когато потребителят докосне левия край на екрана или пък най-горния бутон от ляво.

В текущата дипломна работа се използва готовия NavigationDrawer предоставен от Android-Studio. Създава се като NavigationDrawer-Activity.

3.2.0 Начало на приложението

Първото нещо, което трябва да направим е да инициализираме нашата връзка с Parse. В точка 2.4.1 съм обяснил как се взима Parse ApplicationId и ClientKey.

Ето как изглежда класа Beginning:

```
Parse.initialize(this, "nE2rM3YtEq45TU6kt", "vnafbqwjkrqjk");  
  
    //кодовете, които трябва да вземем за да инициализираме апа  
    ParseUser.enableAutomaticUser();  
    ParseACL defaultACL = new ParseACL();
```

```
defaultACL.setPublicReadAccess(true);
Parse.setLogLevel(Parse.LOG_LEVEL_DEBUG);
```

3.2.1 Логване в приложението

Класът Login е activity class, който представя Логин екрана. Ако нямате акаунт може да се регистрирате като натиснете бутона Sign Up и той ще ви отведе до Register activity.

Ако натиснете Login то ще вземе съответно името и паролата , която сте подали и ако съществува потребител с такава парола ще ви отведе в класа Userlist. Логването се изпълнява чрез функцията loginInBackground , която логва потребителя и създава нова сесия така, че да можеш да достъпиш потребителя.

```
ParseUser.logInBackground(username, password, new LogInCallback() {
    public void done(ParseUser user,
        ParseException e) {
        if (user != null) {
            The user is logged in.
        } else {
            // Signup failed. Look at the
            ParseException to see what happened.
        }
    }
});
```

3.2.2 Регистриране

Класът Register е Activity клас който показва регистрационния екран , който позволява потребителите да се регистрират.

След натискане на бутона

Register потребителя,паролата и имейла се запаметяват. За регистриране на нов потребител използвам функцията

signUpInBackground(SignUpCallback callback).Тя създава нов ParseUser на сървъра и създава нова сесия така,че да можем да достъпим съответния потребител . Потребителското име и паролата трябва да бъдат въведени преди извикване на метода.

callback-извиква се след като регистрирането приключи.

Пример от кода ми:

```
ParseUser user = new ParseUser();
// Set core properties
user.setUsername(username);
user.setPassword(password);
// Invoke signUpInBackground
user.signUpInBackground(new SignUpCallback() {
    public void done(ParseException e) {
        if (e == null) {
            // Hooray! Let them use the app now.
        } else {
            // Sign up didn't succeed. Look at the ParseException
            // to figure out what went wrong
        }
    }
});
```

3.2.3 Чат с потребителите

Класът User е Activity class ,който показва всички регистрирани потребители. Също показва online/offline статуса на потребителя. В класа има бутони ,които те отвеждат към други активитита.Изкарването на потребителите го правя с класа ParseQuery<ParseUser>query;

```
ParseQuery<ParseUser> query = ParseUser.getQuery();
query.whereNotEqualTo("username", user);//not equal because we don't want
our name
query.findInBackground(new FindCallback<ParseUser>() {
public void done(List<ParseUser> objects, ParseException e) {
if (e == null) {
// The query was successful. }
else {
// Something went wrong.
}
}
});
```

Класът UserAdapter е adapter class за User Listview.Той показва името и картинката на всеки потребител. Адаптера конвертира ArrayList от обекти във View и могат да бъдат изписани на екрана чрез listview.Имаме Tabhost Чрез , които може да се превключва през отделните табове.

3.2.4 Изпращане на съобщение

Класът Chat е Activity class ,който показва личния чат между потребителите. Показва разговорите между потребителите и позволява на потребителите да изпращат и да получават съобщения.

Функцията,която изпълнява изпращането на съобщения е sendMessage(); Тя се извиква при натискане на бутона send.Ако текстът не е празен , създава обект и изпраща текста до Parse сървъра.

```
ParseObject po = new ParseObject("Chat");
Po.put("sender",user);//user which sends a message
Po.put("receiver",user1);//user which will see the message
Po.put("message",string);
Po.saveEventually((e) -> {
    If( e ==n null){
        //save-succesfull
    }else{
        //problem
    }
}
```

saveEventually() vs save()

Повечето save-функции се изпълняват веднага , и информират вашето приложение, когато запазването е приключено. Ако не ви е необходимо да знаете кога запаметяването е приключило можете да използвате saveEventually. Предимството е ,че ако нямате интернет saveEventually ще го запази в устройството и ще го изпрати когато връзката бъде обновена.

Чрез функцията loadconversationlist изкарваме на екрана всички съобщения.

ParseQuery е клас,който дефинира query ,което се използва за извличане на ParseObjects.Това е най-често срещаният метод за намиране на обекти ,които съвпадат с query през ParseQuery.findInBackground() метода,използвайки FindCallback.

```
ParseQuery<ParseObject> query = ParseQuery.getQuery("Chat");
query.findInBackground(new FindCallback<ParseObject>() {
    public void done(List<ParseObject> objects, ParseException e) {
        if (e == null) {
            //error
        } else {
            //fail
        }
    }
}
```

Класът `ChatAdapter` е адаптер клас за `Chat Listview`. То показва изпратените и приетите съобщения. (изскарването им на екрана)

Ако натиснем бутона `Send Photo`, ще имаме възможност да изберем снимка, която ще бъде искарана на екрана за да потвърдим, че искаме да я изпратим.

3.2.5 Създаване на блогове

Класът `CreateBlog Activity class`, който позволява на потребителя да създаде пост. Чрез функцията `savenote()`. Тя се извиква при натискане на бутона `save`.

```
ParseObject post = new ParseObject("Post");
```

```
post.put("title",postTitle);//string
post.put("content",postContent);//string
post.saveInBackground(new SaveCall() {
    public void done( ParseException e) {
        if (e == null) {
            // The save was  was successful. }
        else {
            // Something went wrong.
        }
    }
}
```

В разгледания код първо създаваме обекта Post. След това избираме имена на заглавие и съдържание на Поста. След това запамятаваме обекта и можем да викнем различни функции след това в зависимост от това дали запамятаването е успешно или не.

Освен запамятаване на темата и съдържанието аз запамятавам и изображение във поста.

```
ByteArrayOutputStream stream = new ByteArrayOutputStream();
// Compress image to lower quality scale 1 - 100
bitmap.compress(Bitmap.CompressFormat.PNG, 100, stream);
byte[] image = stream.toByteArray();

// Create the ParseFile
ParseFile file = new ParseFile("androidbegin.png", image);
// Upload the image into Parse Cloud
file.saveInBackground();
```

```
// Create a New Class called "Post" in Parse
final ParseObject post = new ParseObject("Post");
// Create a column named "ImageName" and set the string
post.put("ImageName", "AndroidBegin Logo");
// Create a column named "ImageFile" and insert the image
post.put("ImageFile", file);

// Create the class and the columns
post.saveInBackground();
```

3.2.5 Отбелязване на статистики

Класът Statistics е Activity class, който показва статистиките на броя на изпратени и получени съобщения. ParseQuery може да бъде използвано като брои всички обекти без да ги достъпваме всички тези обекти. Като пример можем да вземем този код. Този код брои всички обекти, а тези обекти са съобщенията.

```
ParseQuery<ParseObject> query = ParseQuery.getQuery("Chat");
query.countInBackground(new CountCallback() {
    public void done(int count, ParseException e) {
        if (e == null) {
            objectsWereCounted(count);
        } else {
            objectCountFailed();
        }
    }
})
```


В дипломната използвам тази идея за да преброя всички обекти като добавя името на потребителя, на който трябва да преброим съобщенията.

3.2.5 Изкарване на всички блокове

Класът MainActivity е Activity клас, който представя всички блокове записани. При натискане на дадения блог, потребителя отива до SeeBlogActivity и може да прочете цялата новина. Функцията displayblogs взима всички обекти. В нея функцията FindInBackground() връща всички обекти от блога.

```
ParseQuery<ParseObject> query = ParseQuery.getQuery("Post");
query.whereEqualTo("author", ParseUser.getCurrentUser());
setProgressBarIndeterminateVisibility(true);
query.findInBackground(new FindCallback<ParseObject>() {
    public void done(List<ParseObject> postList, ParseException e) {
        setProgressBarIndeterminateVisibility(false);
        if (e == null) {
            // If there are results, update the list of posts // and notify the adapter posts.clear();
            for (ParseObject post : postList) {
                Note note = new Note(post.getObjectId(), post.getString("title"),
                post.getString("content"));
                posts.add(note); }
            ((ArrayAdapter<Note>) getListAdapter()).notifyDataSetChanged();
        }
        else { Log.d(getClass().getSimpleName(), "Error: " + e.getMessage());
        }
    }
});
```

3.2.6 Преглед на определения блог

Класът SeeBlog е activity клас ,който предоставя на потребителя възможността да прочете цялата публикация.

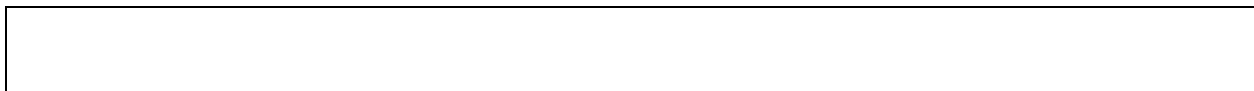
Чрез Picasso успявам да изкарам изображението, като преди това съм взел Uri на обекта.

```
ParseQuery<ParseObject> query = ParseQuery.getQuery("Post");
queryInBackground(blog.getId(), new GetCallback<ParseObject>() {
    public void done(ParseObject object, ParseException e) {
        if (e == null) {

            ImageView imageView = (ImageView) findViewById(R.id.imageView);
            ParseFile file = object.getParseFile("ImageFile");
            Uri fileUri1 = Uri.parse(file.getUrl());

            Picasso.with(getApplicationContext())
                .load(fileUri1)
                .resize(300, 300)
                .into(imageView);
        } else {

        }
    }
});
```



3.2.6 Класът Images

Изкарва всички картинки във публикацията. Чрез Picasso изкарвам последното изпратено изображение.

Четвърта Глава – Ръководство за потребителя

4.1 Изискване към хардуера

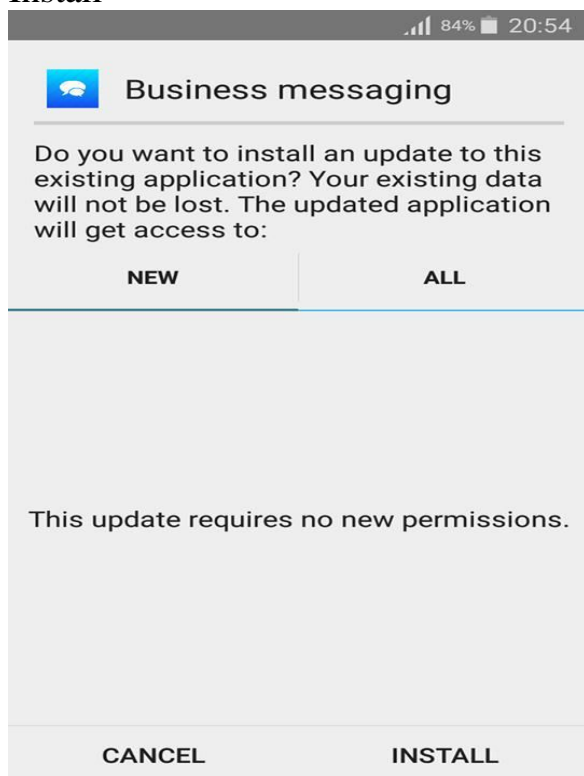
Минималните изисквания към мобилния телефон :

Операционна система-Андроид;

4.2 Инсталиране

1 – Изтегляне на приложението може би след време от Google Play

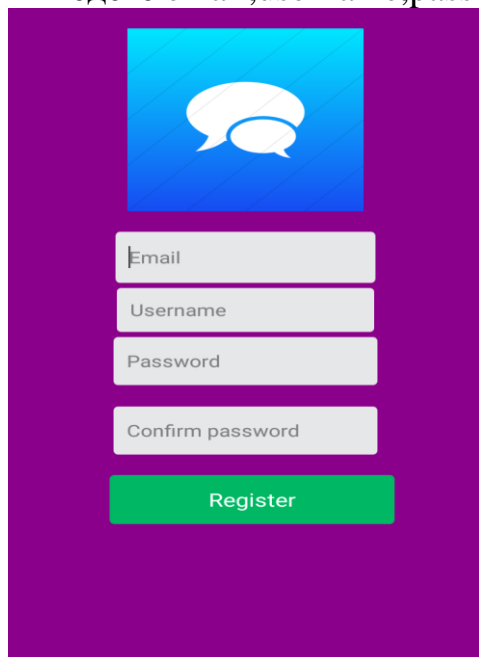
2 – стартирайте си apk файла и цъкнете бутона
Install



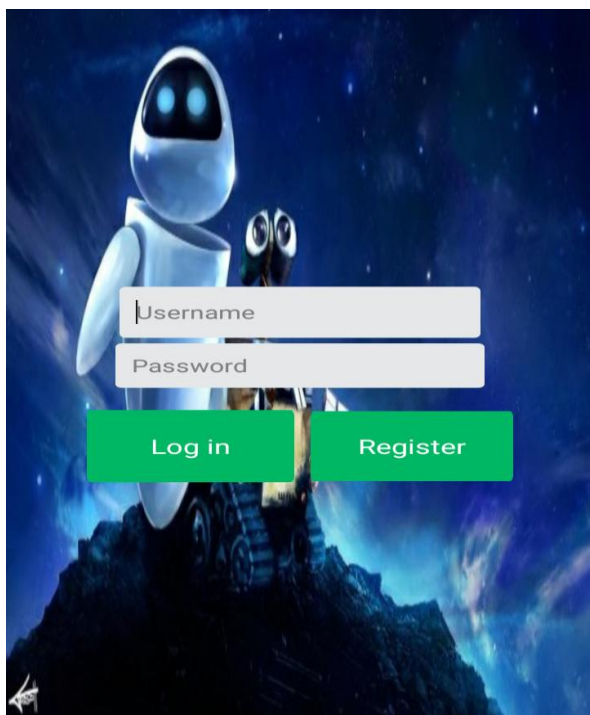
4.3 Ръководство на потребителя

B-Chat работи подобно на всяко чат и приложение и могат да се четат постове от тема, която им е интересна.

Първо всеки потребител трябва да си създаде акаунт като се регистрира. Въведете email, username, password, confirm password.

The image shows a registration form for the B-Chat application. It features a purple background. At the top, there is a blue square icon with two white speech bubbles. Below the icon, there are four white input fields with rounded corners, each containing a placeholder text: 'Email', 'Username', 'Password', and 'Confirm password'. At the bottom of the form, there is a green rectangular button with the word 'Register' in white text.

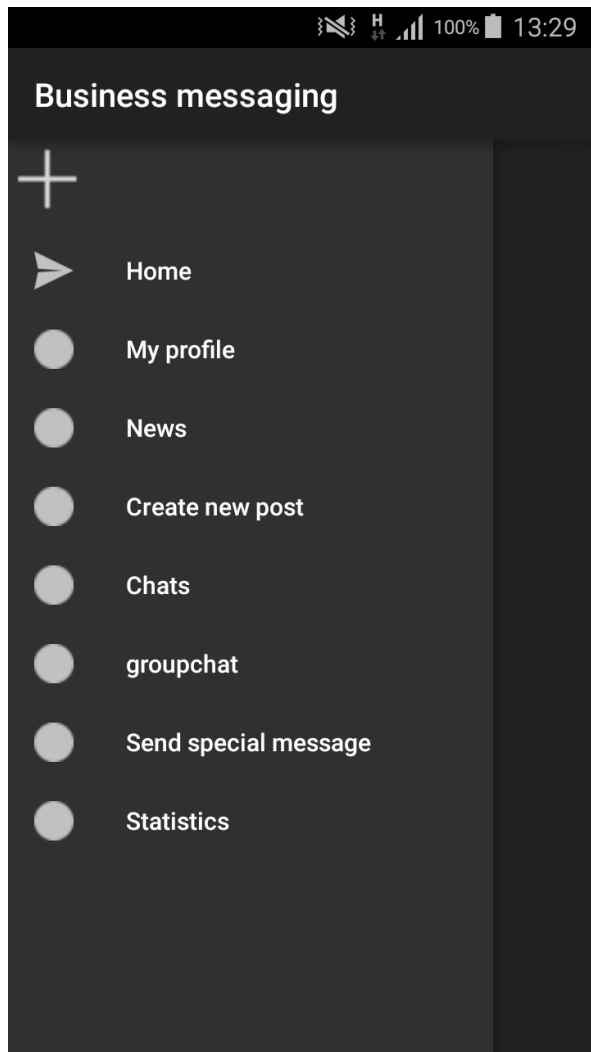
Ако притежава акаунт ще трябва да се логне.



На мястото на Username напише своето потребителско име.

На мястото на Password напише своята парола.

След това ще влезнем във основното активити. Така изглежда основната страница във дипломната работа.



Мой профил

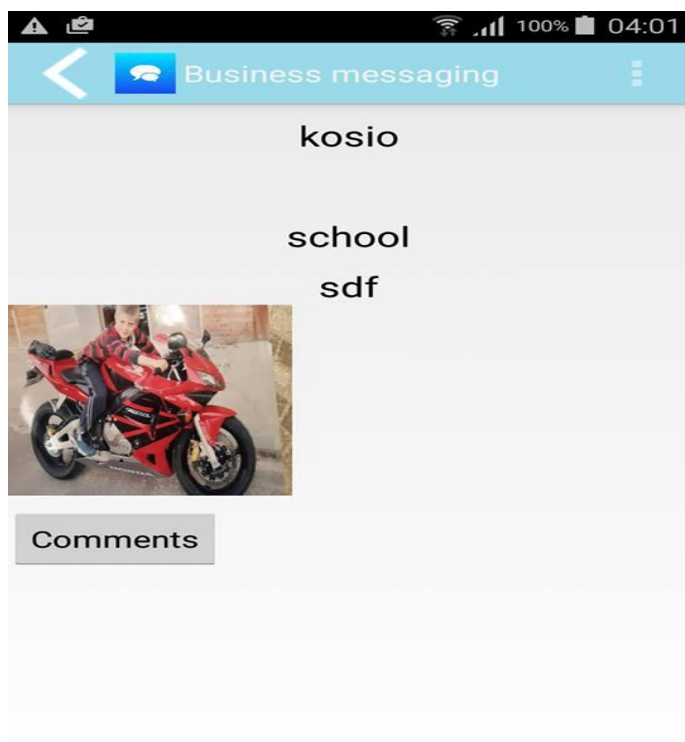
При натискане на “my profile” можем да видим нашия профил.

Публикациите

Публикациите можем да ги видим като натиснем News.

Прочитане на блога(публикацията)

Когато натиснем някоя публикация ще можем да прочетем и видим дадената публикация.



Писане на коментари и изписване на коментарите:

Ако натиснем бутона Comments ще ни излезе дискусията мижду потребителите в зададения пост.

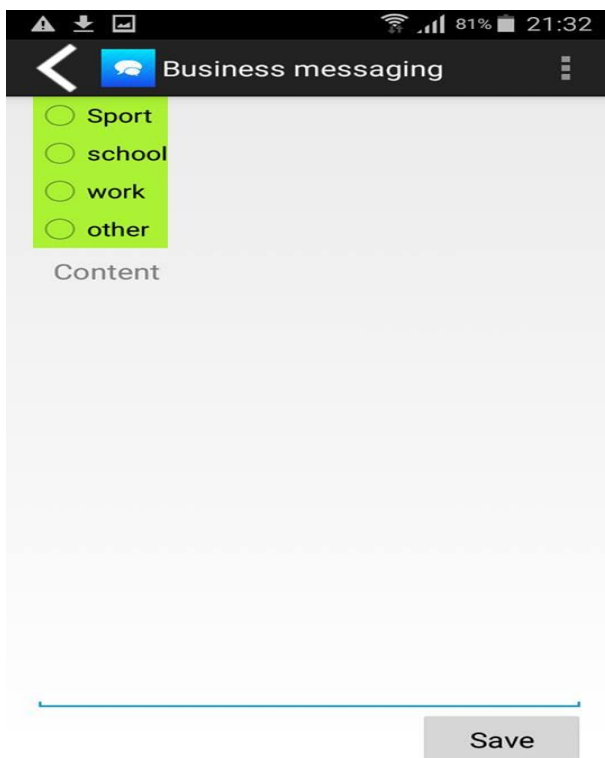
Добавянето на коментар става по начина по който се изпращат съобщения.

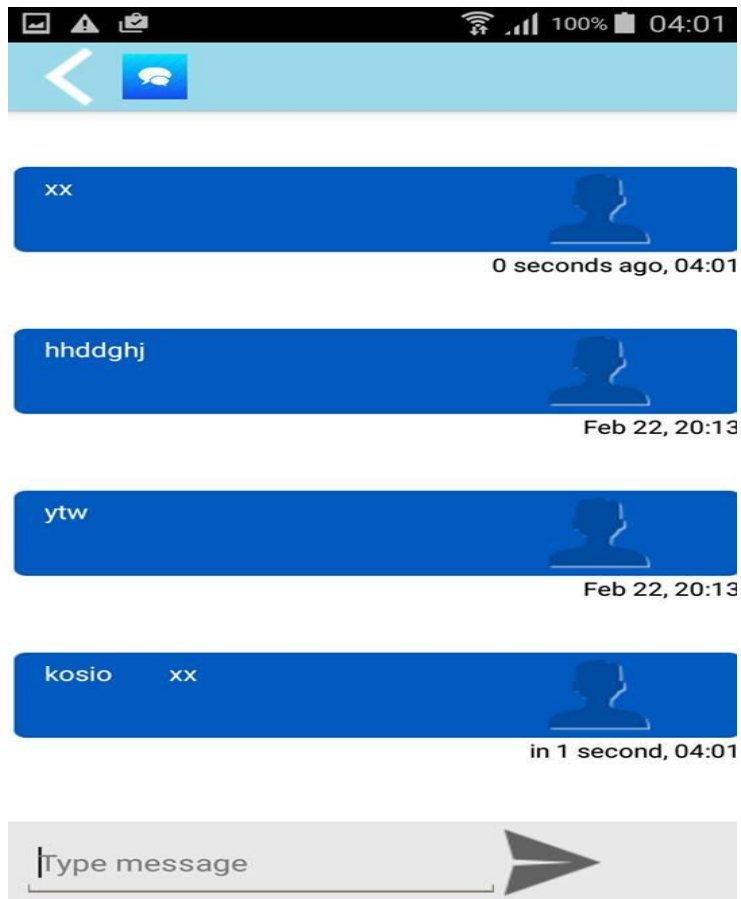
Добавяне на блог

Ако натиснем бутона Create new post най отгоре в менюто ще можем да създадем нова публикация.

При създаване на публикация трябва да натиснем радио бутончето в зависимост каква тема искаме да изберем.

И при натискане на бутона Save създаваме публикация.

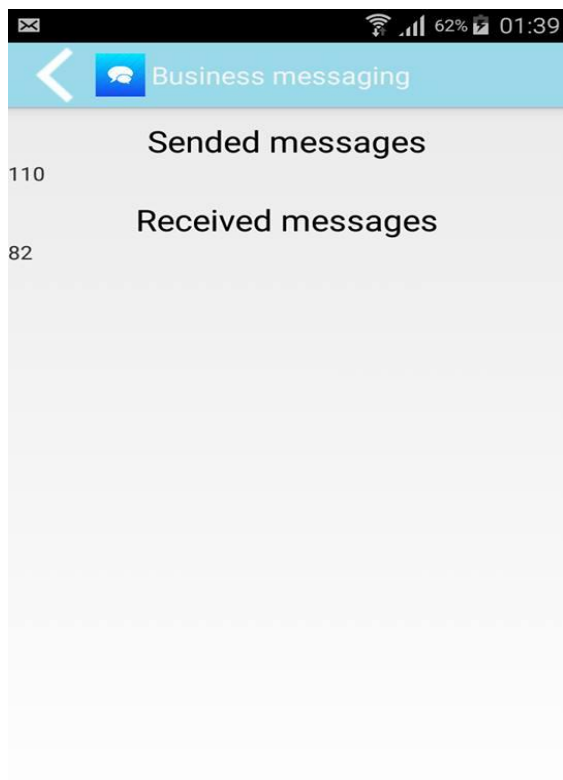




Статистики

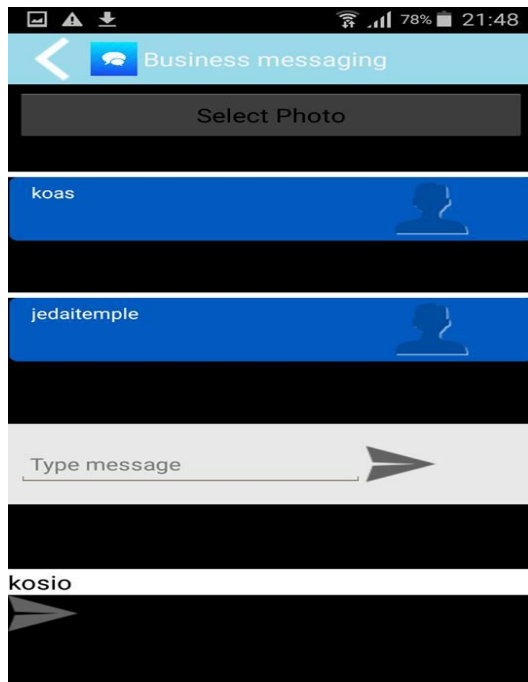
Ако натиснем от менюто Statistics ще отидем и ще видим броя на изпратените и получените ви съобщения.

Ако натиснем бутона Choose Image ще изберем



Специален чат

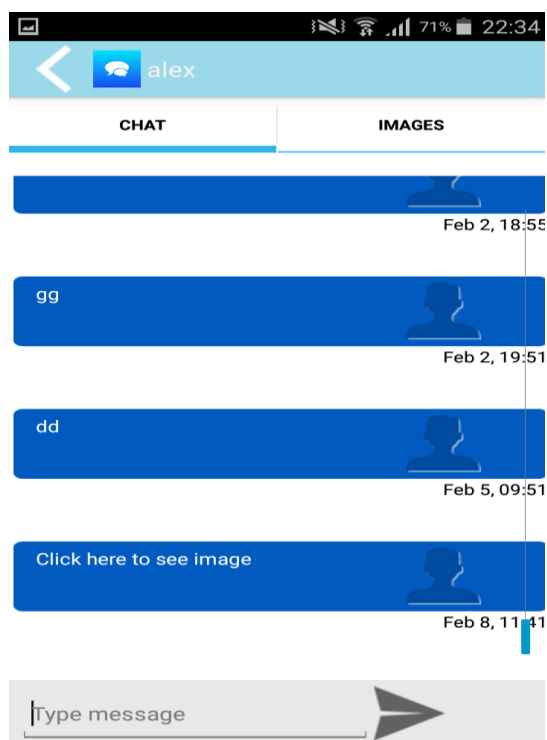
Ако отидем в specialchat ще можем да изпратим изображение, което ще бъде изтрито след като получателя го види. Ок това е процедура с малко повече внимание.Трябва първо да се въведе потребителя на който искаме да изпратим съобщението.след това насискаме долната стрелкичка.След това пишем съобщението което ще бъде изтрито.



Чат

Ако натиснем бутона chats отиваме и виждаме всички потребители в чата и ни и ако натиснем върху потребителя с,който искаме да си пишем.

След като сме натиснали върху потребителя ни отиваме във чата. Ето как изглежда чата.Като например сме натиснали на потребител alex.



В син цвят са изписани вашите съобщения, а в бял съобщението на потребителя с когото си пишете. За да изпратите съобщение трябва да натиснете бутона Send. Ако натиснем Images ще отидем и ще можем да добавим изображение и можем да видим всички изображения с този потребител.

Ако искаме да изпратим изображение отиваме до Images и натискаме Send photo и избираме снимка от галерията. След това натискаме ImagesinConv и ще видим всички изображения в разговора.

Групов чат

Ако натиснем бутона group chat ще отидем до груповия чат. Груповия чат работи като обикновения чат и също могат а се изпращат изображения в чата.

Писане на коментари и изписване на коментарите:

Класът Comments ни показва дискусията мижду потребителите в зададения пост.

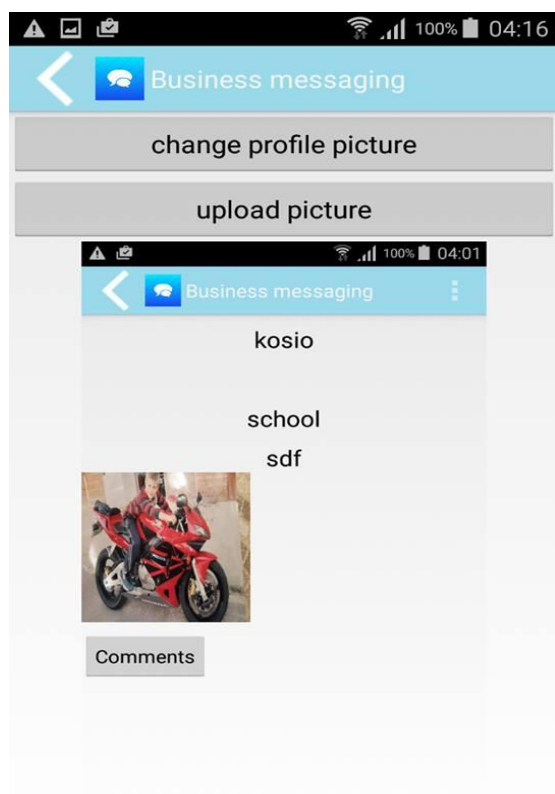
Добавянето на коментар става по начина по който се изпращат съобщения.

Промяна на профиланата снимка

Става като от менюто изберем бутона промяна на профилната снимка. След това натискаме бутона Select Image и след това ако сме харесали снимката можем да си я сложим като профилна.

Изход

При натискане на бутона Log out ще излезем от приложението.



Заклучение

Всички първоначално поставени цели за разработване са напълно изпълнени. Успях да направя едно добро, по моите стандарти доста добро приложение.

По време на разработката успях да науча доста нови технологии. Успях да придобия нов опит в разработката на андроид приложения.

Бъдещите версии на приложението предвиждат подобрен потребителски интерфейс, включващ много и различни анимации и преходи, които да направят използването му по-приятно и интерактивно. Програмата ще бъде преведена на няколко езика включително и български, което ще позволи на повече потребители да работят с него.

ИЗПОЛЗВАНА ЛИТЕРАТУРА

1. <https://www.linkedin.com/pulse/20140901105910-82582062-five-benefits-of-using-parse-com-as-mobile-app-backend> -Parse
2. <https://play.google.com/store/apps/details?id=com.path.paperboy> - PathTalk
3. <https://play.google.com/store/apps/details?id=com.ninechat.android.chat> - 9Chat
4. <http://developer.android.com/sdk/index.html> - Android Studio
5. Eclipse
6. <http://code.tutsplus.com/tutorials/getting-started-with-lua-and-the-corona-sdk-mobile-21457> -Corona
7. <http://www.kinvey.com/> -Kinvey
8. <https://parse.com/> -Parse
9. <https://www.kumulos.com/> - kumulos
10. <http://square.github.io/picasso/> -picasso
11. <https://parse.com/docs/android/api/com/parse/ParseQuery.html> -Parse retrieving objects
12. <https://parse.com/docs/android/guide> -Parse documentation