**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**ДИПЛОМНА РАБОТА**

Тема: Мобилно приложение за водене на бележки

Дипломант: Научен ръководител:

*Илиян Емилов Германов ст. преп . Любомир Чорбаджиев*

СОФИЯ

2016

**Увод**

В днешното забързано ежедневие сме принудени да помним много и различни работи като например списъка с покупки, задачите, които трябва да свършим и така нататък. Доста често се случва да забравим за нещо важно, за което ще съжаляваме по-късно. Лесен начин за справяне с подобни проблеми е записването на бележки на хартиен носител, но при нарастването на обема на нашите записки се получава „каша“ и ефективността на този подход рязко намалява. Дори и да успеем да се справим с организацията на хартиените бележки все още можем да се сетим за нещо прекалено късно или даже да го пропуснем. Разчитането на близък приятел или роднина да ни подсети не е много надеждно решение. Остава ни единствено константно да преглеждаме нашето тефтерче, което би било доста досадно и трудоемко. Вторият по-умен и ефективен начин е да прибегнем до помощта на техниката или по-точно смартфоните.

В XXI век почти всеки разполага със смартфон (мобилен телефон с разширена функционалност, сравнима с джобните компютри). Използвайки мобилно устройство получаваме хиляди възможности: можем да съхраним неограничен брой бележки без да настъпва хаос в организацията, да редактираме записаното без да се налага да драскаме или преписваме, да получим напомняне в точното време, да прикрепяме снимков материал и много други. Появява се нов проблем не всички производители разполагат с фабрично приложение за водене на бележки. Вариантът да използваме чернови на съобщения за записване и аларми за известяване е доста непрактичен и нелеп при наличието на специализирани приложения.

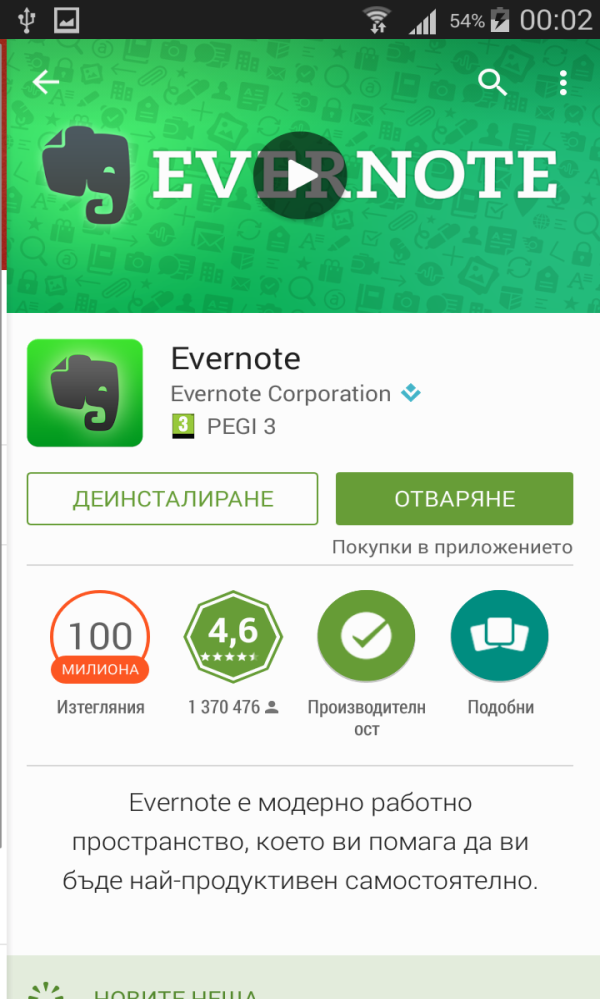
Целта на настоящата дипломна работа е да се създаде мобилно приложение за лесно, бързо и ефективно водене на бележки, позволяващо на потребителя да записва своите мисли, да получава напомняния по-късно, да прикрепя снимков материал и да организира своите бележки. Някои от най-важните функционалности на приложението са: създаване на текстова бележка, с възможност за добавяне на изображение (заснето с в момента с камерата или избрана от галерията); създаване на списъци; създаване на гласова бележка, която се транскрибира автоматично; раздел за важни бележки; раздел за частни бележки, които се криптират и се достъпват с парола; кошче; поставяне на подсещане. За целта ще бъде направено проучване, анализ и сравнение на съществуващи подобни продукти.

**I Глава**

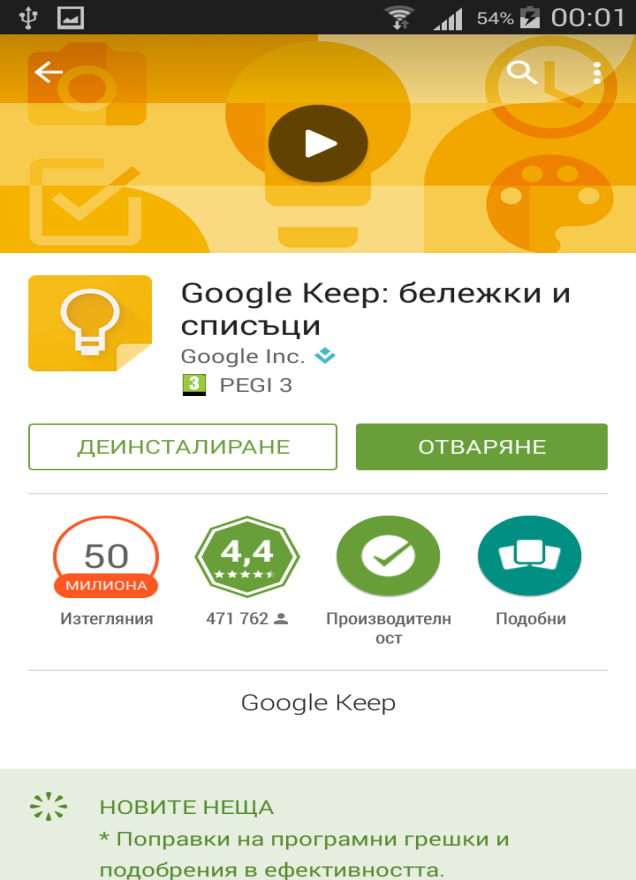
**Проучвателна част. Преглед на съществуващи продукти, развойни среди и развойни средства.**

* 1. **Преглед на съществуващи продукти**

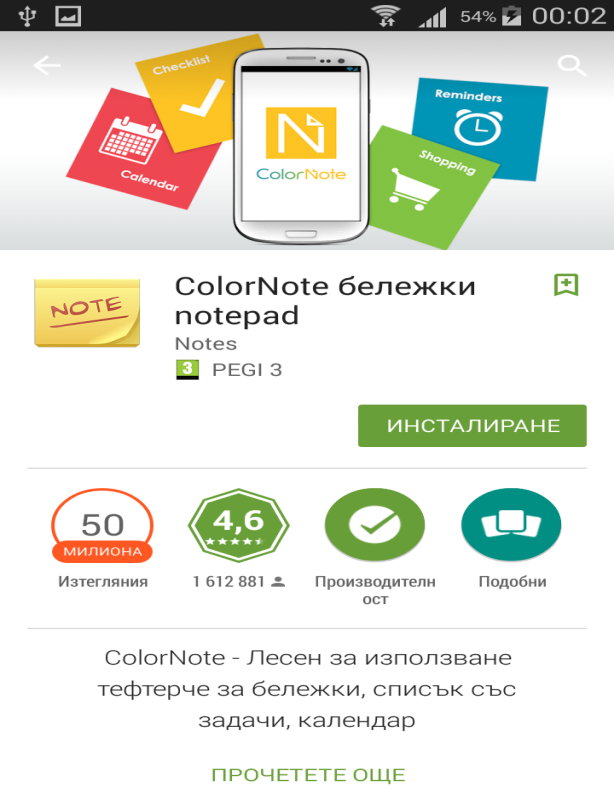
Едни от типовете приложения, които достигат най-голям брой сваляния в Google Play store са “note taking applications” или на български език приложения за водене на бележки. Примери за това са водещите приложения от тази категория като Evernote (100 000 000 – 500 000 000 инсталирания), Google Keep (50 000 000 – 100 000 000 инсталирания), ColorNote (50 000 000 – 100 000 000 инсталирания) и много други.



“Evernote в Google Play”



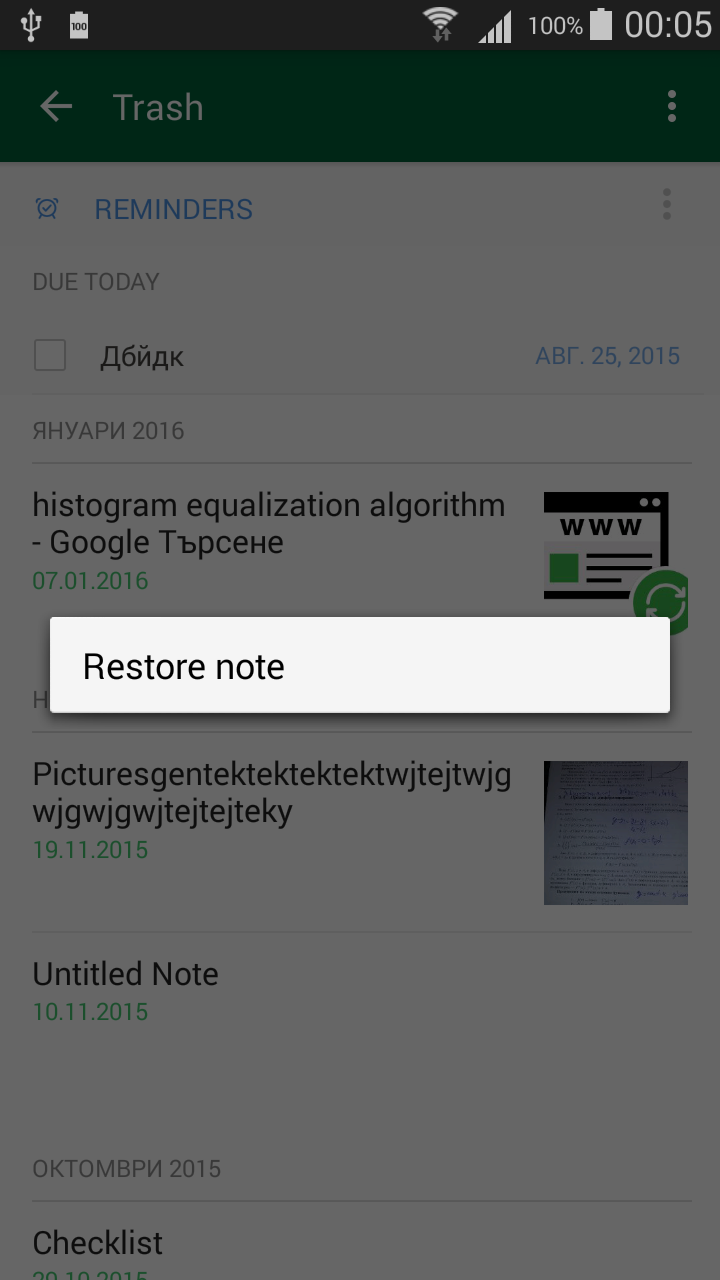
“Google Keep в Google Play”



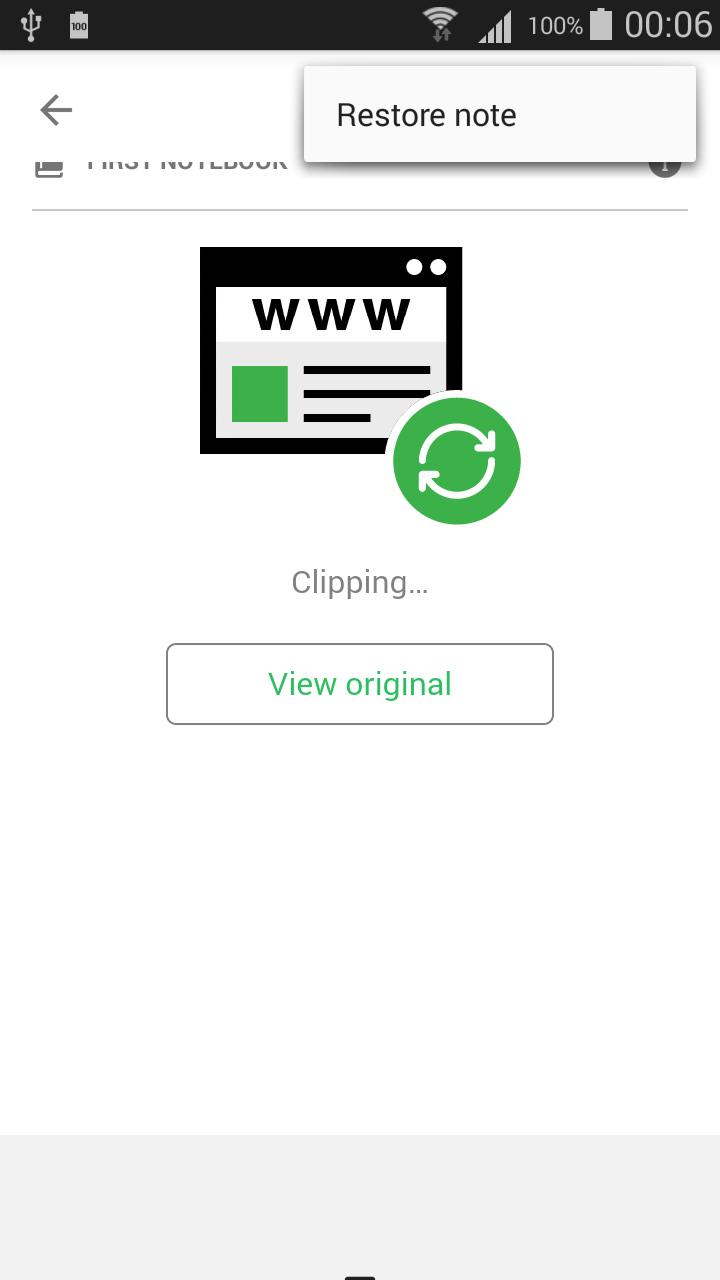
“ColorNote в Google Play”

Освен, че приложенията за водене на бележки достигат до много и различни по пол и възраст потребители, те съдържат на мобилните им устройства сравнително по-дълго от другите видове апликации. Например една игра от категорията “entertainment” (забавление) има голяма вероятност да бъде деинсталирана в момента, в който омръзне на потребителя, докато едно качествено приложение за водене на бележки би му вършило работа дълго време. В тази сфера изборът на добри на приложения е голям, но смятам, че може да бъде проектирано ново, което е да бъде по-лесно за използване, бързо и ефикасно. Идеята за настоящата ми дипломна работа се роди лятото, когато се нуждаех от приложение за удобно записване на просто бележки. Първо си инсталирах Evernote с идеята, че като е с най-много сваляния ще ми свърши най-добра работа. Още в самото начало не ми хареса това, че за да го ползваш си принуден да си направиш регистрация (ползваща се за синхронизирането между различни устройства, което е полезна функционалност), за да запиша просто един час с задача, който ще изтрия веднага след като премине. Друго, което ми направи негативно впечатление е сложността на приложението и хилядите функционалности, които няма да използвам като например Work Chat и т.н. Също така честите известия подканващи ме да си закупя “premium upgrade” на акаунта ме накараха да го деинсталирам. След това опитах с Google Keep. При него нямах проблем с входа към приложението, тъй като имам Google акаунт. Създаването на различни видове записки беше лесно, но тяхното управление и организиране беше трудно. Например, за да изпратиш бележка в кошчето имаш няколко варианта. При първият е необходимо да я отвориш и да намериш опцията за изтриване от менюто, което става сравнително бавно, тъй като отварянето зарежда нов екран. Вторият е да я изхвърлиш с “swipe” директно от главния екран, но вместо да отиде в коша се премества в архива, от който ми отне някъде около 10 минути да разбера как се трие.

Със същият проблем се сблъсках и в Evernote само където до момента, в който пиша тази документация така и не разбрах как се изтрива бележка завинаги и дали това въобще е възможно. Прилагам няколко екранни снимки от Evernote:



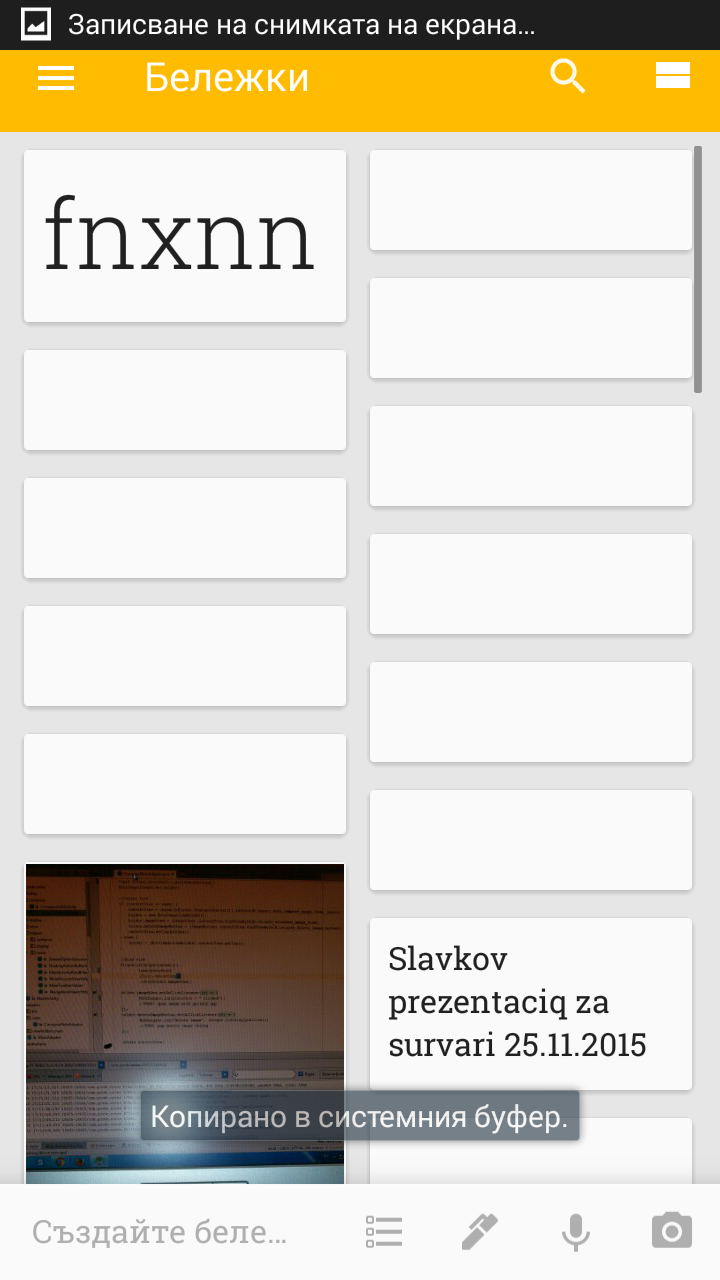
“Long click върху item в Trash”



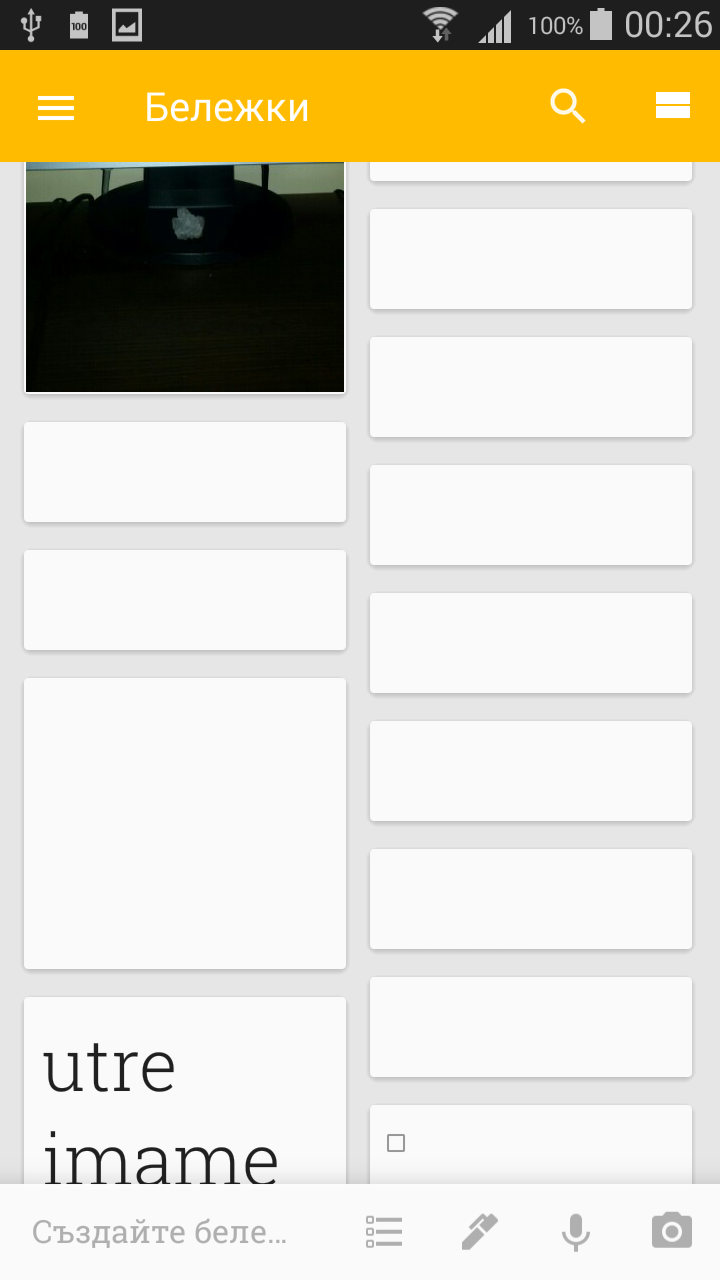
“Отворено меню от item в Trash”

Както ясно се вижда от екранните снимки опцията “Delete” просто липсва на всички места, за които се сетих.

Друг проблем, на който се натъкнах в Google Keep е, че при излизане от редактиране или създаване на нова бележка промените винаги се запазват. Заради тази функционалност немалко пъти запазих промени, които не исках да бъдат съхранени. Това лесно може да бъде забелязано на следните екранни снимки:



“Празни бележки в Google Keep”



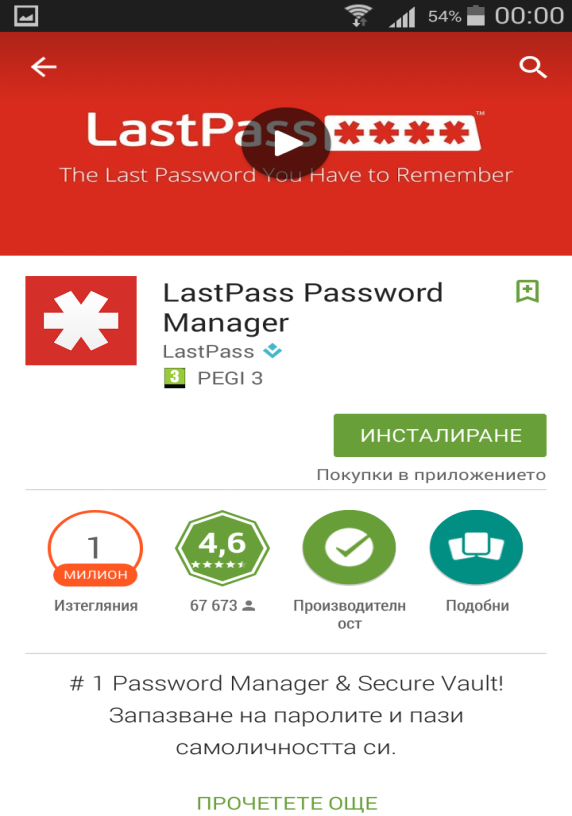
„Празни бележки в Google Keep (2)”

По време на проучването на Google Keep забелязах, че при натрупването на повече бележки и използването на приложението от по-слаб телефон (тествано върху Samsung Galaxy S3 Neo) се появява известно забавяне, което разваля потребителското усещане. Нещо, което ми направи добро впечатление в Google Keep, но отсъстваше в Evernote е това, че има възможност да се разместват бележките с “drag and drop”.

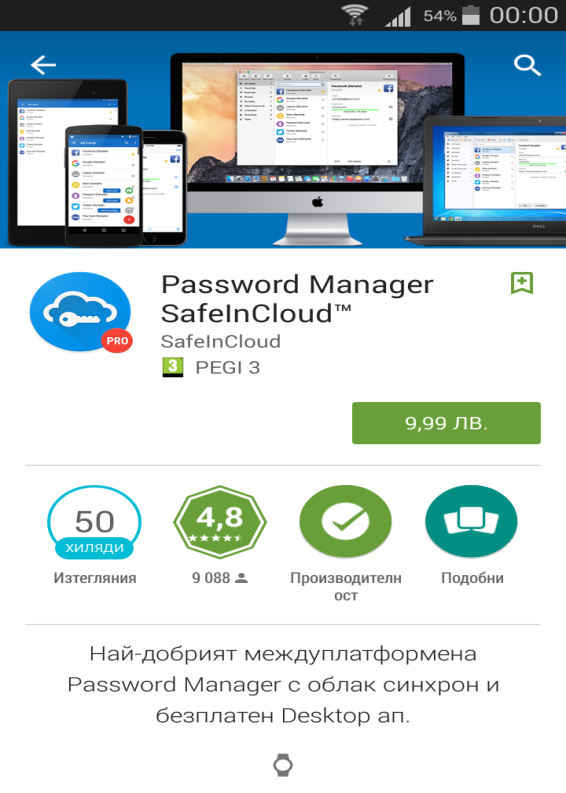
Друга важна за мен функционалност, която не открих в гореспоменатите водещи приложения е създаването на “private” (частни) бележки, които да не могат да се видят от всеки, който има достъп до мобилното устройство. Частните бележките биха могли да се използват за много цели и биха направили приложението много по-полезно за потребителя. Например с тяхна помощ може да съхранявате лична информация и пароли. Освен това приложенията проектирани специално за съхранение на пароли са доста сваляни в Google Play. Примери за това са Keeper® Password Manager (10 000 000 – 50 000 000), LastPass Password manager (1 000 000 – 5 000 000) и платеното Password Manager SafeInCloud™ с цена 9,99лв. в България (50 000 – 100 000).



“Keeper® Password Manager в Google Play”



“LastPass Password Manager в Google Play”



“Password Manager SafeInCloud™ в Google Play

**1.2 Развойни среди**

**1.2.1 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).

**1.2.2 “Android studio”**

„Android studio“ е официалното IDE предлагано от Google.

То има поддръжка на най-новият Android 6.0 (Marshmallow) като предлага и емулатор за него. Подобно на “Elcipse” има подходящ графичен редактор, като повечето от функционалностите му се припокриват с ADT плъгина.

**„Android studio“ предлага :**

* редактор
* примерни кодове на прости програми
* интеграция на Version Control System (Giт)
* 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

**1.2.3 SDK**

SDK (Software Development Kit) е добавка както за Eclipse така и за Android studio. Тя предоставя услугата Android manager, чрез която удобно могат да се свалят инструменти и библиотеки за разработването на различните версии на Android операционната система, както и различни конфигурации спомагащи за по-добра симулация на виртуалната машина.

**1.3 Развойни средства**

**1.3.1 Основни компоненти на Android приложение**

Основните компоненти на едно Android приложение са :

* „Activity“ - презентационен слой на приложението (екрани)
* „Content Provider“ - предоставя интерфейс към данните на приложението
* „Broadcast Receiver“ - компонент който получава и отговаря на системни съобщения и „Intents“
* „Services“ - услуги, които изпълняват задачи на зададен фон без да предоставят потребителски интерфейс.
* „Views“ (изгледи) - представляват различни части от потребителския интерфейс като бутони, текстови полета и други. Базовия клас на всички изгледи е „android.view.View“.
* „Intent“ - това са асинхронни съобщения, които изискват някаква функционалност от други компоненти като „Activities“.

**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 бързо, лесно и ефективно водене на бележки. Приложението трябва да може поддържа създаването и организирането на различни видове бележки – текстови, текстови с прикрепен снимков материал, списъци и автоматично транскрибирани гласови бележки. Да има раздели спомагащи за по-добра организация– „All notes” в него се съхраняват всички бележки, „Starred” – тук се намират всички важни за потребителя бележки, които той е отбелязал със звездичка, “Private” – раздел, който се достъпва само с парола и бележките в него са криптирани и “Bin” – кошче, в което попадат всички изтрити бележки, които не са частни (от “Private” раздела). Да има функционалност за напомняне в зададено време и да има търсене за различните раздели. Потребителския интерфейс да е опростен и разбираем за потребителя, и да позволява приятна работа с приложението.

**2.2. Избор на език за програмиране и софтуерните средства**

За създаването на приложението са използвани мобилната операционна система “Android”, езикът “Java” и средата “Android Studio”. Това е така поради три причини. Първо “Java” е официалният език под който се разработват Android приложения, а “Android Studio” е средата, която Google препоръчва. Второ, вече от година и половина разработвам 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. И трето както в момента, така и за в бъдеще искам да продължа да се занимавам с разработката на native Android приложени, тъй като тази област е една от най-бързо развиващите се такива в технологичния свят и предлага неограничен брой възможности както на потребителите, така и на програмистите.

**2.3. Структура на приложението**

**2.3.1. Зависимости (dependencies) на проекта**

В приложението са използвани 14 външни библиотеки за по-разбираемо описани нека ги разделим на два вида – “com.android.support” (официални зависимости, невлизащи 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:**

* 'com.android.support:appcompat-v7:23.1.1' – предлага класове съвместими и поддържащи по-стари версии на Android
* 'com.android.support:recyclerview-v7:23.1.1' – предлага оптимизиран контейнер подобен на класът ListView от Android SDK-а, но с тази разлика, че в него е имплементиран “View holder pattern”, което води до по-добро представяне при голям обем от данни
* 'com.android.support:cardview-v7:23.1.1' – съдържа класът CardView, който представлява повърхност с “Material Design”
* 'com.android.support:design:23.1.1' – помощна библиотека за дизайн, съдържащи основните практики в Android

* 'com.google.code.gson:gson:2.4' – не влиза изцяло в този тип, но пак едно от официалните библиотеки, тъй като неин създател е Google; предоставя възможност за превръщане на обекти в “JSON” формат

**“3rd party” зависимости:**

* 'com.github.clans:fab:1.6.2' – предоставя “Floating Action Button” и интерактивно меню към него, отговарящи на “Material Design”
* 'com.jakewharton:butterknife:7.0.1' – улеснява инициализирането на изгледи чрез идентификационен номер използван в xml файловете
* 'com.wdullaer:materialdatetimepicker:2.1.1' – предлага диалогови прозорци за избиране на дата или час, отговарящи на “Material Design”
* 'com.github.frankiesardo:linearlistview:1.0.1@aar' – съдържа контейнер, работещ вграден в друг “scrollable” контейнер
* 'com.squareup.picasso:picasso:2.5.2' – улеснява и оптимизира зареждането и представянето на изображения
* 'com.scottyab:aescrypt:0.0.1' – предоставя клас с възможност за криптиране и декриптиране на низови с “AES 256 encryption”
* ':libs:draglinearlayout-1.10' - локално добавена библиотека в проекта, предлага линейно оформление с възможност за влачене на елементите в него
* 'com.github.afollestad.material-dialogs:core:0.8.5.3@aar' - предоставя често използвани диалогови прозорци, отговарящи на “Material Design”
* 'com.github.afollestad.material-dialogs:commons:0.8.5.3@aar' – добавка към 'com.github.afollestad.material-dialogs:core:0.8.5.3@aar'

**2.3.2. Пакети на при приложението**

Приложението е разделено в 15 главни пакети, всеки от който отговаря за отделен проблем.

* Пакетът “com.gcode.notes”:

Главен пакет на приложението, съдържащ всички останали пакети.

* Пакетът “com.gcode.notes.activities”:

В този пакет се съдържат всички Activity класове на проекта и техните помощни класове.

* Пакетът “com.gcode.notes.adapters”:

Пакет, съдържащ всички адаптери използвани за управлението на контейнерите в приложението.

* Пакетът “com.gcode.notes.controllers”:

В този пакет са разположени всичките Controller класове, отговарящи за промяната и държанието на потребителския интерфейс в различните раздели на приложението.

* Пакетът “com.gcode.notes.data”:

В този пакет се съдържат всички логически единици, които представляват различните видове бележки. Това са базовите “ContentBase”, “ContentDetails”, “MyLocation”; от помощният пакет “list”, използвани за представянето на списъци – “ListDataItem” и “ListData”; “NoteData”, реализиращ бележка

* Пакетът “com.gcode.notes.database”:

Отговаря за създаването на база данните. Осигурява връзката между нея и приложението и контролира всички операции свързани с базата данни. В него се разполага класът “DatabaseController”, от който могат да бъдат достъпени публично всички методи за работа с базата. Съдържа и помощният пакет “extras” чрез който е реализиран класът “DatabaseController”.

* Пакетът “com.gcode.notes.extras”:

Съдържа всички помощни класове използвани в повече от един пакети. В него се намират пакетите: “utils” (от utilities – комунални услуги, предлага класове решаващи често срещани проблеми в приложението), “values” (буквално преведено стойности, в него са разположени всички класове, отговарящи за константите в проекта) и “builders” (съдържа класове, които служат за инстанциранието на други класове).

* Пакетът “com.gcode.notes.fragments”:

В този пакет се съдържат фраментите използвани в приложението.

* Пакетът “com.gcode.notes.helper”:

Временен пакет, отговарящ за разместването на бележки чрез влачене в главния контейнер. След реализиране на всички функционалности ще бъде преструктуриран.

* Пакетът “com.gcode.notes.motions”:

Съдържа всички класове реализиращи анимациите и преходите в приложението.

* Пакетът “com.gcode.notes.notes”:

В този пакет се намира singleton класът “MyApplication” , предоставящ достъп до контекста на приложението и инстанция на класът “DatabaseController” от всеки един пакет в проекта.

* Пакетът “com.gcode.notes.receivers”:

В него се съдържат всички „receivers” (приемници) използвани в дипломната работа.

* Пакетът “com.gcode.notes.serialization”:

Отговаря за сериализацията на обекти. Може да превърне дадена инстанция на обект в символен низ (string) в “JSON” формат.

* Пакетът “com.gcode.notes.services”:

Съдържа всички услуги (services) , използвани в приложението.

* Пакетът “com.gcode.notes.tasks”:

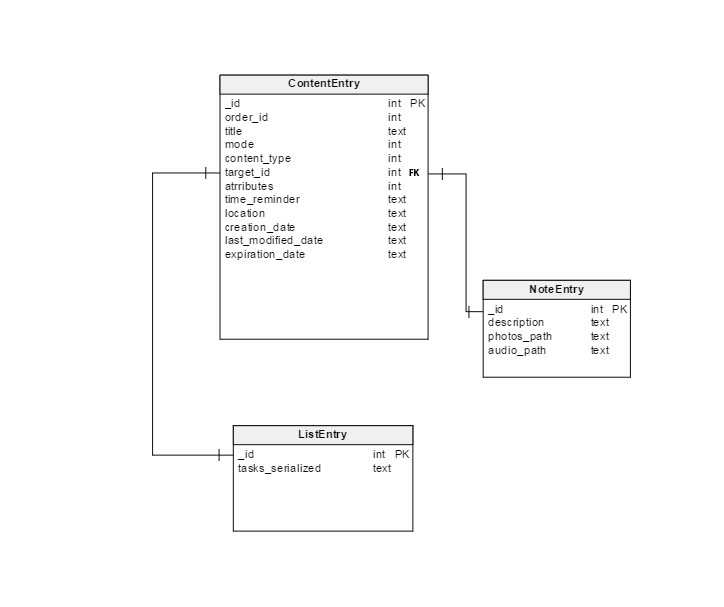
Съдържа всички задачи (tasks), използвани в приложението. Има 2 пакета – “async” (в него се намират всички класове, които са наследници на класът “AsyncTask”) и “other” (отговаря за всички, задачи които не са наследници на класът “AsyncTask”).

* Пакетът “com.gcode.notes.ui”:

В този пакет се разполагат класът “ActionExecutor” (отговарящ за всички действия свързани с потребителския интерфейс на приложението напр. изтриване на бележка и т.н.) и пакетът “helpers” (в него се намират всички помощни класове , реализиращи диалоговите прозорци и детайлите по потребителския интерфейс).

**2.4. Структура на базата данни**

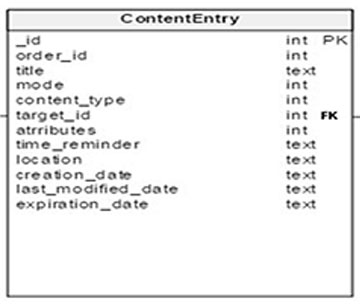
За база данни приложението използва “SQLite” със структура представена на „Фигура 2.4.1“:

****

Фигура 2.4.1

Базата данни се състои от следните 3 таблици – ContentEntry (съдържа общата информацията за бележка и списък), NoteEntry (съдържа атрибутите, които има една бележката), ListEntry (съдържа атрибутите, които има един списък).

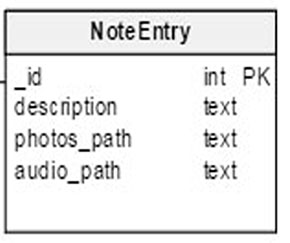
Таблицата „Content Entry” („Фигура 4.2.2):

****

**„**Фигура 4.2.2“

* \_id – първичен ключ на таблицата, уникален идентификационен номер на записа
* order\_id – идентификационен номер използван за определяне на мястото в подредбата на бележките
* title – заглавие на бележката
* mode – тип (режим) на бележката определящ към кой раздел принадлежи и как трябва да се третира от приложението ( normal, private, important, deleted\_normal, deleted\_important)
* content\_type – показва типа на записа, тоест дали е списък или бележка
* target\_id – вторичен ключ, служа за връзка с таблицата за атрибути на записа (NoteEntry / ListEntry)
* attributes – флаг показващ, дали даденият запис има атрибути и дали необходимо те да се извлекат
* time\_reminder – времето и датата, когато потребителя трябва да бъде подсетен, може да бъде NULL
* location – сериализирана инстанция на класа “MyLocation”, показваща местоположението, където е създадена бележката, може да бъде NULL
* creation\_date – датата на създаване на записа
* last\_modified\_date – датата на последна промяна на записа
* expiration\_date – датата на изтичане на записа (при попадане на запис в mode „deleted” той получава дата на изтичане след, която бива автоматично изтрит при следващото пускане на приложението), може да бъде NULL

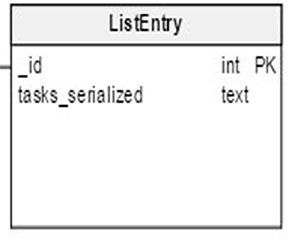
Таблицата “NoteEntry” („Фигура 4.2.3“):



„Фигура 4.2.3“

* \_id – първичен ключ на таблицата, уникален идентификационен номер на записа
* description – описание на бележката
* photos\_path – сериализиран лист от символни низове, представляващи път до прикрепеното изображение към бележката, може да бъде NULL
* audio\_path – път към прикрепеният звуков файл към бележката, може да бъде NULL

Таблицата “ListEntry” („Фигура 4.2.4“):



„Фигура 4.2.4“

* \_id – първичен ключ на таблицата, уникален идентификационен номер на записа
* tasks\_serialized – сериализирани лист от инстанции на класа “ListDataItem”, представляващи един предмет в списъка

За съхраняване на информация в приложението също са използвани “SharedPreferences” (физически представляващи един xml файл), които са подходящи за запазване на “key-value pairs”. В случая на текущата дипломна работа са в тях се запазва паролата на потребителя за частните бележки (криптирана с SHA1 алгоритъм), броят на грешните опити при въвеждане на паролата и флаг показващ дали потребителя е научил за навигационното меню.

**III глава**