**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**Д И П Л О М Н А** **Р А Б О Т А**

Тема: Мобилно приложение за пресмятане на дял от сметка въз основа на снимка на касова бележка

Дипломант: Научен ръководител:

*Гергана Пейкова* *Веселин Русинов*

С О Ф И Я

2 0 1 8

**Увод**

След тежка работна седмица, всички мечтаем да разпуснем. Някои се прибират у дома и веднага си лягат в леглото, други прекарват вечерта със своя любим, а трети нямат търпение да бие последната минута на работния им ден, за да се видят с приятелите си и да им разкажат за всичко случило се след предната им среща. Естествено всички обичат да си похапват, затова каква по-добра идея от тази да се съберат в ресторант. Говорят си, смеят се, ядат и какво ли не още. Времето минава неусетно – час, два – и сякаш всичко е идеално, докато сервитьорката не им даде сметката. Тогава започва един дълъг и неприятен процес – всеки трябва да си намери поръчаната храна и да си пресметне своя дял от сметката, а оставянето на бакшиш означава още пресмятане, което в улеснения ни живот се опитваме да избегнем.

Благодарение на иновативните технологии вече сме в състояние да си опростим живота значително. Настоящата диплома има за цел разработването на технология, с помощта на която да се пресметне дял от сметката въз основа на снимка на касовата бележка. Също така Android приложението ще има функционалностите да се раздели цялата сметка на равни части, потребителя да избере кои продукти да заплати и добавянето на бакшиш

Първа глава

Преглед на съществуващи решения и развойни

среди

# 1.1 Преглед на съществуващи приложения за разделяне на сметката

* интернет пространството има мобилни приложения за пресмятане на дял от касова бележка. В тази глава на дипломната работа са представени подобни разработки.

## 1.1.1 Tab- the simple bill splitter

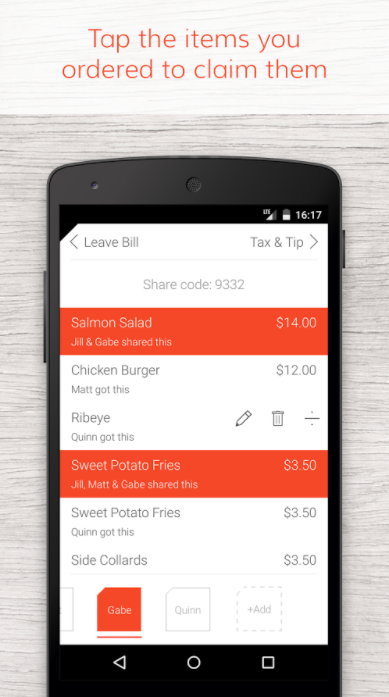
**Tab** ([Фигура1-1)](#page4)емобилно приложение за пресмятанена дял от сметка.Излиза на 22.08.2013г. от bring10, LLC. Фирмата е основана само от 3 представителя, като това е единственото им разработено приложение пуснато на пазара и има над 20 хиляди тегления. Услугата е достъпна както за IOS потребители, така иза Android.

*Фигура 1-1. Лого на Tab*

**Tab** има няколко предимства,някои,от които са следните:

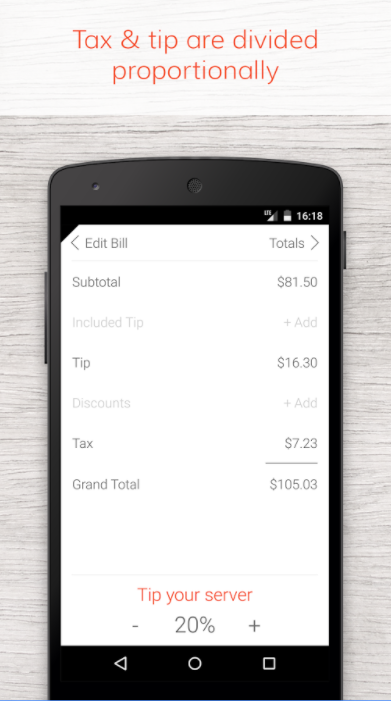
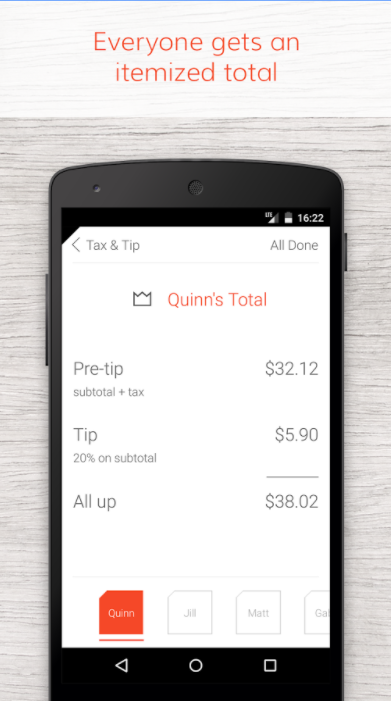
* Лесно достъпно и удобно за ползване от потребителите.
* Съдържа функционалност за пресмятане на дял от сметката и разделяне по равно.
* Една сметка може да се сподели между няколко потребителя.
* Съдържа функционалност за пресмятане на оставения бакшиш, като процент от цялата сметка.

*Фигура 1-1.1 Фигура 1-1.2*



Недостатъци на **Tab***:*

* Работи само с валутата долар($).
* Текста от снимката го засича правилно около 90%.
* След избирането на твойте сметки няма функционалност за пресмятане на същата бележка и трябва да снимаш повторно.

## 1.1.2 Billy: Scan & Split the bill

**Billy** ([Фигура1-2)](#page5)емобилно приложение за пресмятане на дял от сметка.Излиза на 09.05.2013г. от Workaholic. Фирмата е основана през Юли 2010г, като са разработили 5 приложения. Billy е най-новото им приложение и с най-нисък рейтинг. Достъпна е както за IOS потребители, така и за Android и е достигнала над 500 тегления.



*Фигура 1-2. Лого на Billy*

**Billy** има следните предимства:

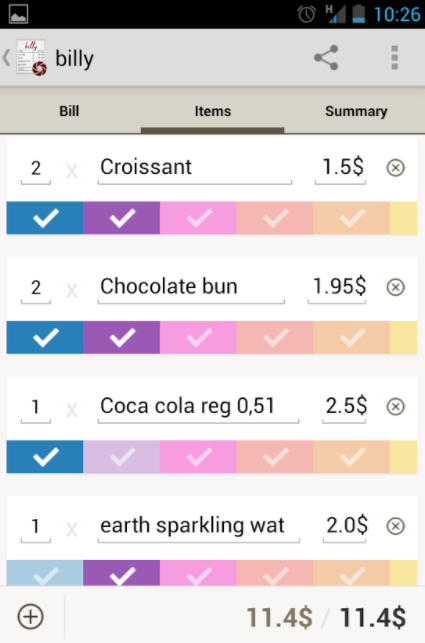
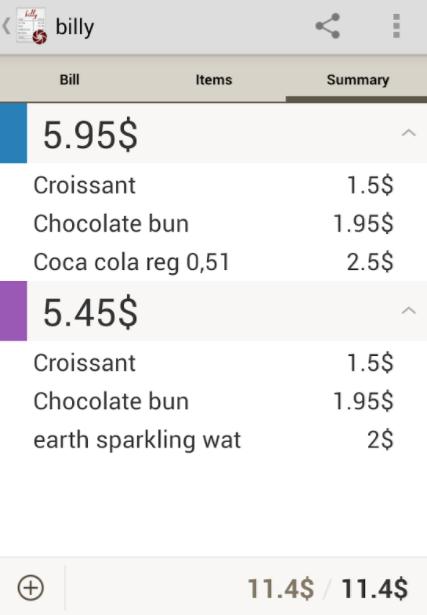
* Лесно достъпно и удобно за ползване от потребителите.
* Има функционалност за пресмятане на сметката на няколко потребителя едновременно.
* Има функционалност за споделяне.

 *Фигура 1-2.1*

Недостатъци на **Billy**:

* Сканирането отнема много време и е с много грешки.
* Не може да сканира бележки на кирилица.
* Дизайнът има нужда от подобрения.

*Фигура 1-2.2 Фигура 1-2.3*

****

## 1.1.3 Clever Bill Splitter

**Clever Bill Splitter** ([Фигура1-3)](#page6)емобилно приложение за пресмятане надял от сметка. Излиза на 15.12.2013г. от Clever Turtles. Android developers, които са разработили само едно приложение. Услугата е достигнала над 10 000 тегления и е достъпна само за Android.

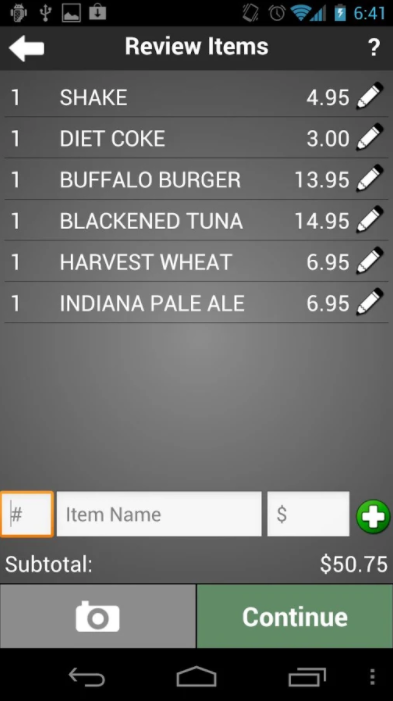


*Фигура 1-3. Лого на Clever Bill*

*Splitter*

**Clever Bill Splitter** има няколко предимства,някои,от които са следните:

* Идейно направено, подобно на игра.
* Има функционалност за разпределяне на поръчките между няколко потребителя.
* Има функционалност за редактиране на данните.

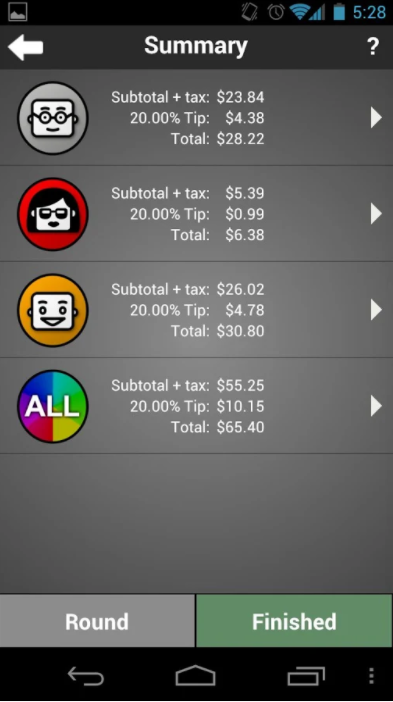
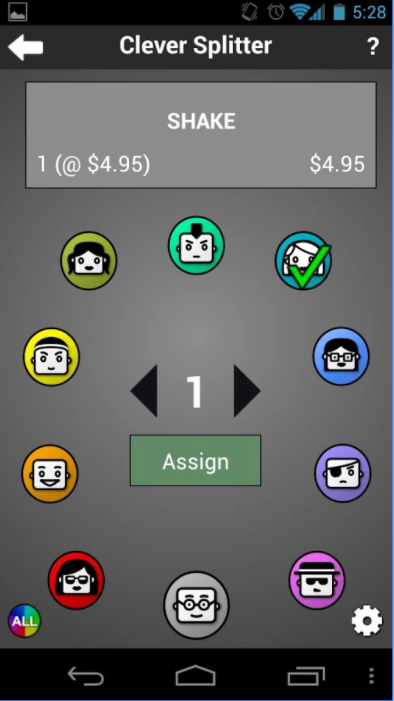
 *Фигура 1-3.1* *Фигура 1-3.2*

Недостатъци на **Clever Bill Splitter***:*

Сметката може да се разпредели до 10 човека. Приложението може да направи 7 Дизайна има нужда от подобрение.

Не работи за различни видове формати касови бележки.

*Фигура 1-3.3 Фигура 1-3.4*



# 1.2 Основни принципи, технологии и развойни среди за реализиране на мобилни приложения

Мобилното приложение представлява компютърна прорама, създадена за да работи, на мобилни устройства. Приложенията, които не са предварително инсталирани на устройството се предлагат в ,, Магазин за приложения“, като всяка мобилна платформа си има отделен магазин:

Android – Google Play iOS – App Store

Windows – Windows Store

Приложенията, които разширяват функционалността на устройствата, са написани, използвайки Android software development kit (SDK), на езика Java, който има пълен достъп до Android APIs. Java може да се комбинира със С/С++ при разработката на приложения, като от версия 1.4 насам езикът Go също се поддържа, но не е препоръчителен. SDK притежава разнообразие от инструменти за разработка, включвайки дебъгер, софтуерни библиотеки, QEMU емулатор, документация, примерни кодове и ръководства. В началото официалното IDE на Google беше Eclipse, който използва приставката Android Development Tools (ADT). През декември 2014 г., Google пуснаха Android Studio, базирано на IntelliJ IDEA, като свое основно IDE за Android. Това е платформата използвана за мобилното приложение описано в тази дипломна работа.

Android притежава изобилие от приложения, които потребителят може да изтегли от магазина, който му позволява да ги инсталира, ъпгрейдва и премахва от устройството си. Google Play е главното приложение, инсталирано на устройствата, използващи Android, което се съобразява с изискванията на Google и което има лиценз за Google Mobile Services Software. Google Play позволява на потребителите да използват приложения, публикувани от Google и някои софтуерни разработчици.

Броят на новосъздадените приложения продължава да нараства със страхотна скорост. Всяка седмица в Google Play излиза нещо иновативно, което впоследствие бива използвано от милиони хора. Така магазинът за приложения на Google увеличава популярността си и убеждава хората, че Android е най-използваната система за мобилни устройства в света за момента.

## 1.2.1 Android Studio

**Android Studio** е официалната интегрирана среда за разработка(IDE)наоперационната система за Android на Google, изградена върху софтуера JetBrains IntelliJ IDEA и специално разработена за разработка на Android. Той е достъпен за изтегляне на операционни системи, базирани на Windows, MacOS и Linux. *Android Studio* е заместител на Eclipse Android Development Tools (ADT) като основен IDE за разработка на Android приложение.

Цялата функционалност на Intellij остава и в *Android Studio*, включително и добавки, помагащи в разработката на Android приложение. Добавките биват:

Instant Run - позволява инсталиране без рестартиране на цялото приложение

Layout Editor - визуализация и drag&drop редактор на потребителския интерфейс, спомагащо за по-бързо и ефективно оформление на дизайна

Google Cloud Platform - платформа, която позволява интеграцията на

Google Cloud Messaging

Virtual Android Device – симулатор, който може да се използва за тестване и debugging на приложението

Спрямо платформата за разработване Eclipse, Android идва с нужните функционалности вградени, позволявощо по-ефективна работа

## 1.2.2 OCR

**Optical Character Recognition (OCR)** е технология за конвертиране напечатен текст в текстов файл. Тя е широко използвана за въвеждане на печатни данни от хартия или файл, включително от лични документи, фактури, банкови извлечения, компютърни разпечатки, визитки, поща.

Това е често срещан метод на запис на печатни текстове, за да може текста да бъде редактиран с текстов редактор, да се търси в него, да се съхранява по-компактно, да се показва онлайн, както и да се използва в компютърни програми, като автоматизиран превод, конвертиране на текст към говор.

Процеса за разпознаване е следния:

1. Програмата анализира структурата на документа
2. Разделяне на страницата на блокове : текст, таблици, изображения и др.
3. Редовете се разделят на отделни думи. Думите се разделят на отделни символи.
4. След като текстът е разделен на единични символи, всеки един от тях бива сравнен с набор от шаблони.
5. След осъществяването на сравнението, програмата излиза с няколко хипотези за това какъв е символът.
6. Символите биват съединени обратно в думи и след това в редове.
7. Програмата предоставя на клиента своя финален резултат

Оптичното разпознаване на символи (OCR) бива прилагано в множество програми. Основна причина за разработката на подобни приложения са слабостите на хартиения носител като средство за съхраняване на информация.

Колкото повече се увеличават документите на хартиен носител, толкова повече продуктивността спада. Става все по-трудно информацията да бъде намирана, а нейното използване става бавно и неефикасно. Хартията е скъпа и заема голямо пространство. Информацията на хартиен носител не може да бъде редактирана на момента, както би било възможно при работата с текстов файл на вашия компютър. Всички тези фактори довеждат до внедряването на Оптичното разпознаване на символи (OCR), което позволява извличането на информация от остарелите, скъпи и трудно-обработваеми хартиени носители

* поставянето и в удобни за работа текстови формати, които могат да бъдат лесно намирани, използвани и редактирани.

Втора глава

Изисквания

# 2.1 Основна функционалност

Функционалните изисквания към Android приложението са следните:

* Заснемане на касова бележка
* Вид подялба на сметка
* Равна подялба
* Самостоятелна подялба
* Функционалност за избиране на брой потребители
* Функционалност за добавяне на бакшиш въз основа на процент от цялата сметка

# 2.2 Избор на програмен език и библиотеки

## 2.2.1 Java

В днешно време Java е един от най-популярните езици за програмиране. На него пишат милиони разработчици по цял свят. Най-големите световни софтуерни корпорации като IBM, Oracle, Google и SAP базират своите решения на Java платформата и използват Java като основен език за разработка на своите продукти. Java е обектно ориентиран програмен език, който е силно разпространен - има около 9 милиона програмисти, използващи го. Програмният език се използва от браншови фирми като Google, SAP, Oracle и IBM. Java е мултиплатформена, което води до лесно интегриране в различни платформи - единственото нужно за да се компилира на друга машина, тя да има инсталирана JVM.(WORA - Write Once, Run Anywere).

Java не се компилира до машинен код, а до byte code, за което е нужна JVM(Java Virtual Machine). Друго от предимствата е, че виртуалната машина се грижи за освобождаване на паметта от тъй наречения Garbage collector.

Java също така е универсален език , което дава възможността за изграждане на десктоп, андроид и уеб базирани приложения. Общността на Java разработчиците, в интернет услуги като StackOverflow и GitHub, e сред първите по брой на членове, което води до лесно намиране на решения при възникване на проблем.

## 2.2.2 Tesseract

**Tesseract** в геометрията е четиримерен хиперкуб,но в света натехнологиите е оптичен софтуер за разпознаване на символи за различни операционни системи. Това е свободен софтуер, пуснат под лиценза Apache, версия 2.0, а разработката е спонсорирана от Google от 2006 г. насам. През 2006 г. Tesseract се счита за най-добрия отворен код за OCR. Достъпен е както за Linux и Windows, така и за Mac OS X.

В първите версии, технологията е могла да приема само снимки с TIFF формат, в една колона и разчитал само Английски език. В днешно време благодарение на библиотеката Leptonica, Tesseract поддържа няколко нови формати на изображения, също така поддържа изходно-текстово форматиране, позиционна информация и анализ на оформлението на страницата.

Със новите версии на Tesseract излизат и нови езици. Сега разпознава над 100 езика, като може да бъде обучен да работи и с други езици.

**Tesseract** се използва за обработка на текст отляво надясно.Подходящ еза използване като основа и може да се използва за по-сложни задачи на OCR, включително анализ на оформлението чрез използване на интерфейс като OCRopus.

Изходът на Tesseract ще има много лошо качество, ако входните изображения не са преработени така, че да отговарят на него: Изображенията трябва да бъдат мащабирани така, че текстът x-height да е поне 20 пиксела, при всяко завъртане текстът няма да бъде разпознат, нискочестотните промени в яркостта трябва да бъдат филтрирани с висока честота или голяма част от страницата ще бъде унищожена, а тъмните граници трябва да бъдат премахнати ръчно или ще бъдат погрешно интерпретирани като символи.

# 2.3 Структура на приложението

Първо потребителят трябва да направи снимка на касовата бележка. След това има възможността да избере по какъв начин да се разпредели сметката – поравно между няколко човека или за всеки поотделно. И в двата случая въвеждаш брой хора, между които да се разпредели смеката.

Ако избереш разпределение поравно, получаваш като резултат пълната сума, цената, която всеки трябва да заплати и функционалност за задаване на бакшиш въз основа на процент от пълната сума.

Ако избереш разпределение поотделно, въвеждаш поръчките на всеки участник, след което излиза като информация сметката, която всеки трябва да заплати, с опция за допълнителна информация за всеки участник, в която е дадено наименованието на поръчките, броя и цените им. Също така е изписана пълната сума и има функционалност за задаване на бакшиш въз основа на процент от пълната сума.

**Трета глава**

Реализация на проекта

За реализацията на мобилно приложение е използвана платформата Android Studio. Благодарение на вградения редактор, гъвкавата система за изграждане и дебъгване, процеса за реализация премина много по-гладко.

# 3.1. Activities

Какво е Activity? Това е презентационен слой което осигурява презентационния слой на потребителя и е от съществено значение за начина, по който даден потребител ще се придвижва в приложението - тоест създава прозорец, отговарящ за всички бутони, текстови полета, полета за снимки и други.

За реализацията на Android приложението в тази дипломна работа са създадени шест Acitivity-та. За разработката на първият Layoult – capture.xml, е използван следния .xml код:

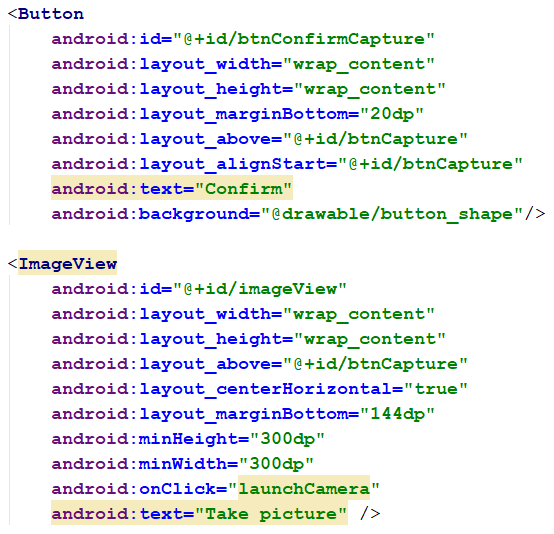
Използваната XML версия е ,,1.0“.Стандарта използван за символно кодиране е UTF-8.

Именното пространство (Namespace) представлява контейнер, съдържащ набор от идентификатори, и осигурява пояснение на омонимните идентификатори, които се намират в различни именни пространства. Именните пространства обикновено обединяват имената, базирайки се на функционалността им. В този XML файл има два Namespace-а:

* android, Namespace URI - <http://schemas.android.com/apk/res/android>
* app, Namespace URI - <http://schemas.android.com/apk/res-auto>

Ориентацията на приложението е вертикална, следователно тя не се променя при завъртане на мобилното устройство. Приложението захваща целия екран на устройството. Фонът на прилоцежението представлява друг XML файл, който се намира в папката за ресурси – drawable.

В capture.xml има два бутона – btnCapture и btnConfirmCapture. За дизайна е използван друг XML файл. В capture.xml, единствената разлика между двата бутона е текста и идентификационния номер (ID).



Capture.xml има ImageView, чиято функционалност е да показва ресурси на изображения, например Bitmap или Drawable resources. Този ImageView приема Bitmap-а на заснетата снимка и се изписва на екрана.

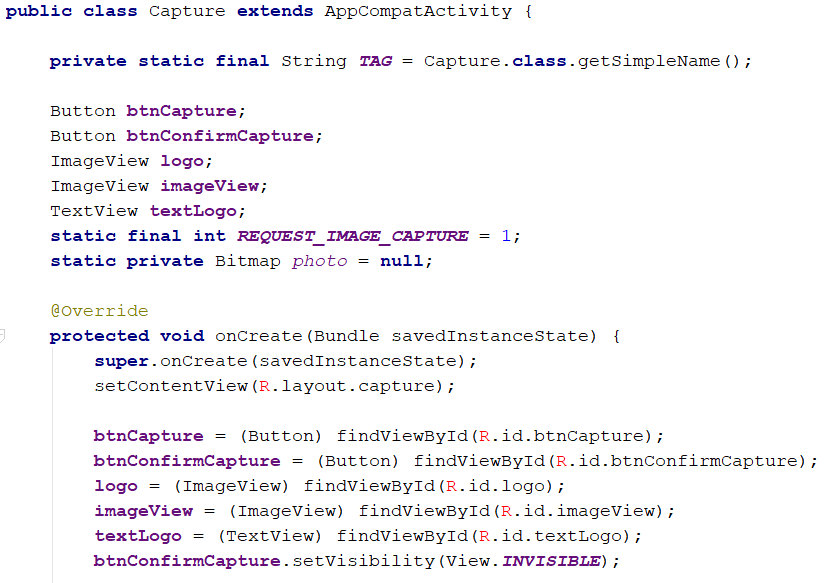
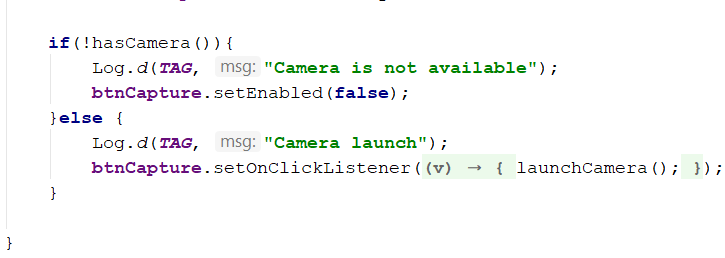


За логото на приложението е използван ImageView и TextView.

# 3.2 Заснемане на касовата бележка

Capture.java разширява класа AppCompatActivity.java, който съдържа базовите функционалности за действия в Android приложението. В началото са инициализирани бутоните, снимките и текста.

Във функцията OnCreate, която е от класа AppCompatActivity.java, се отваря първия Layoult – capture.xml. След това горе инициализираните променливи се свързват със съответните интерфейс елементи.

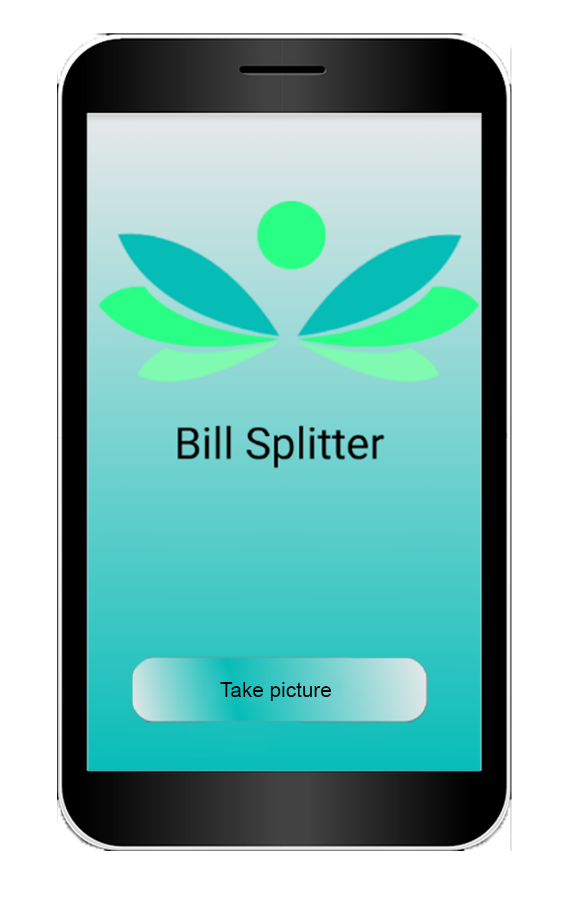


Следващата част от кода проверява дали камерата на мобилното устройство е в изправност и дали притежава необходимите характеристики за правилното изпълнение на функционалността, която приложението изисква.

**Четвърта глава**

Ръководство на потребителя

# 4.1 Използване на приложението

Приложението ни приветства с екран с логото на приложението и бутон за заснемане на касовата бележка. Приложението е изчистено и лесно за ползване от всеки потребител.

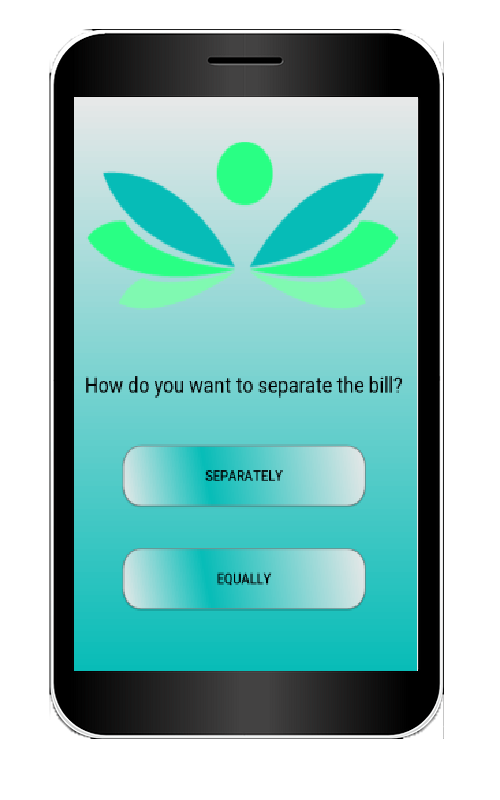
След натискане на бутона **Take Picture**, приложението отваря камерата на телефона, като при първото отваряне, след инсталацията, приложението пита потребителя за достъп до камерата на мобилното устройство. При отказване на достъпа, приложението те връща на първото Activity и това се повтаря докато потребителя не позволи достъпа до функционалността.



След заснемането на касовата бележка, се отваря нов екран подобен на първия, като логото е заменето със снимката и е добавен още един бутон. Ако потребителят не е доволен от снимката, поради размазване или някаква друга причина, при натискане на бутона ,,Take picture” камерата се отваря отново и се заснема нова снимка. А ако потребителят е доволен от снимката и иска да продължи по-нататък, натиска бутона **Confirm**.

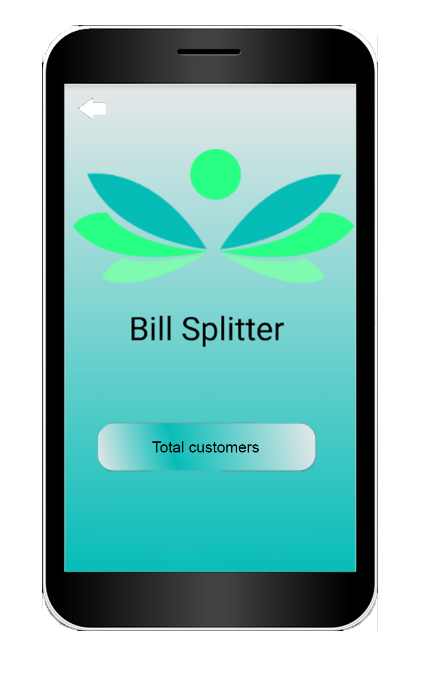
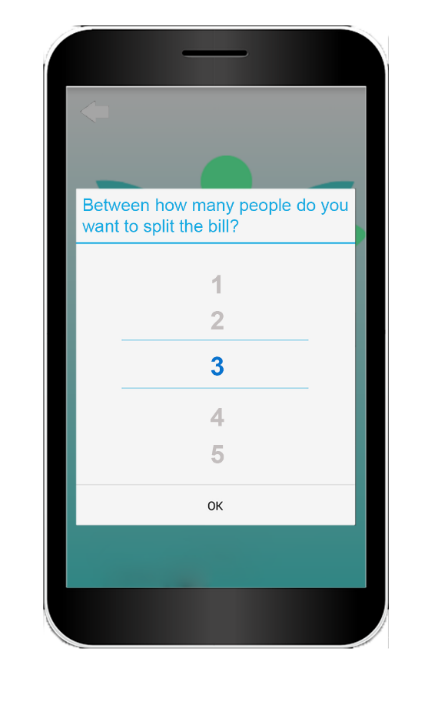
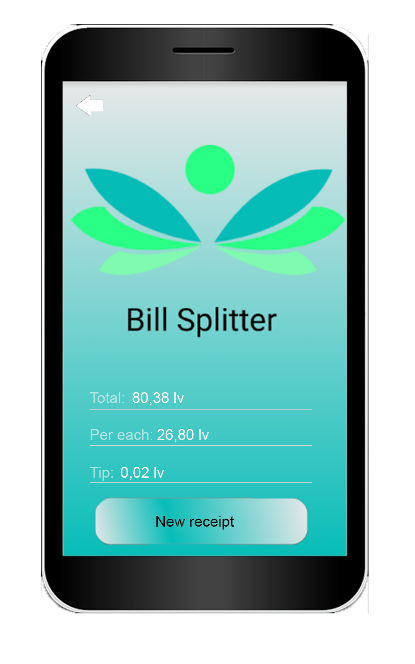
Приложението те прехвърля към нов прозорец, с логото на приложението

* 2 бутона **Equally** и **Separately**. Потребителя трябва да избере по какъв начин да разпредели сметката. В горния ляв ъгъл има бутон във формата на стрелка, чиято функционаност е да върне потрбителя на предната страница с заснемането на бележката.

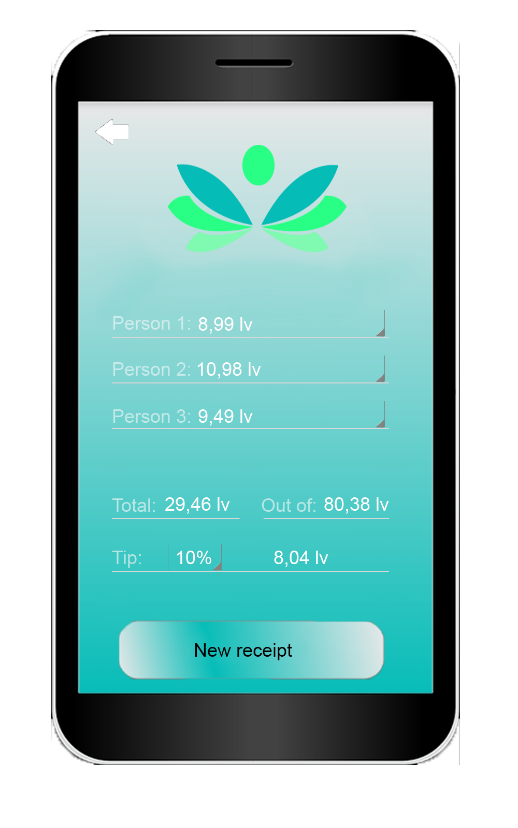


При натискане на бутона **Equally**, се отваря нов прозорец с логото и 2

бутона. Първият бутон е за потребителя да въведе броя хора, между които да разпредели сметката, а вторият за продължаване. В горния ляв ъгъл има още един бутон във формата на стрелка. При натискане, приложението се връща на предната страница с избора за разделяне на сметката.

След натискането на бутона **Confirm**, излиза последното Activity с крайния резултат – информация за общата сума, цената,която всеки трябва да заплати и добавяне на бакшиш. В горния ляв ъгъл има бутон за връщане на предната страница за въвеждане на броя участници, а най-отдолу бутон за започване отначало и въвеждане на нова касова бележка.

Ако при Activity-то със избора за вид разпределение на сметката, при натискане на бутона **Separately** се отваря нов прозорец с избор за брой участници в разпределението. След което се отваря нов прозорец, където рецептите се появяват една след друга. Потребителя може да връща назад рецептите и при натискане на даден участник, тази поръчка се причислява към неговите разходи. Когато потребителя реши че е готов натиска бутона за потвърждение и се отваря последния прозорец разпределението между участниците, общата сума, кочто трябва да заплатят само участниците, цялостната сума на абсолютно всички поръчки и функционалност за пресмятане на бакшиш въз основа на процент от касовата бележка. Също така има функционалност при натискане на стрелката отстрани на даден участник, да излезе подробна информация за поръчката му – поръчката, цената и броя.



# 4.2 Бъдещо развитие

В бъдещото развитие на приложението се планират следните неща:

* Функционалност за запазване на касовите белжки.
* Функционалност за споделяне между няколко потребителя снимки и разпределение.
* Функционалност за работа с касови бележки от различни страни, на различни езици и с различни валути.
* Функционалност за конвертиране на различни валути.
* Меню с различните функционалности.
* Подобрение на дизайна.

# 4.3 Използвана литература

* Android Developers - <https://developer.android.com/index.html>
* Github Tesseract - <https://github.com/imperialsoup/SimpleTesseractExample>
* Android design - <https://drive.google.com/file/d/0B5XIkMkayHgRMVljUVIyZzNmQUU/view>
* XML code - <https://www.w3schools.com/xml/default.asp>

**Заключение**

**Използвана литература**

**Фигури**

[Фигура 1-1. Лого на Tab](file:///C:/Users/Cepe6/Desktop/School/Docs/ÐÐ¾ÐºÑÐ¼ÐµÐ½ÑÐ°ÑÐ¸Ñ/Geri.docx%23_Toc507661630) [2](file:///C:/Users/Cepe6/Desktop/School/Docs/ÐÐ¾ÐºÑÐ¼ÐµÐ½ÑÐ°ÑÐ¸Ñ/Geri.docx%23_Toc507661630)

[Фигура 1-2. Лого на Billy](file:///C:/Users/Cepe6/Desktop/School/Docs/ÐÐ¾ÐºÑÐ¼ÐµÐ½ÑÐ°ÑÐ¸Ñ/Geri.docx%23_Toc507661631) [3](file:///C:/Users/Cepe6/Desktop/School/Docs/ÐÐ¾ÐºÑÐ¼ÐµÐ½ÑÐ°ÑÐ¸Ñ/Geri.docx%23_Toc507661631)

[Фигура 1-3. Лого на Clever Bill Splitter](file:///C:/Users/Cepe6/Desktop/School/Docs/ÐÐ¾ÐºÑÐ¼ÐµÐ½ÑÐ°ÑÐ¸Ñ/Geri.docx%23_Toc507661632) [4](file:///C:/Users/Cepe6/Desktop/School/Docs/ÐÐ¾ÐºÑÐ¼ÐµÐ½ÑÐ°ÑÐ¸Ñ/Geri.docx%23_Toc507661632)

**Съдържание**

|  |  |  |  |
| --- | --- | --- | --- |
| [Увод ......................................................................................................................](#page3) | |  | [1](#page3) |
| [Глава 1. Преглед на съществуващи решения и развойни среди ...................](#page4) | | | [2](#page4) |
| [1.1](#page4) | [Преглед на съществуващи приложения за разделяне на сметката ....](#page4) | | [2](#page4) |
| [1.1.1](#page4) | | [Tab- the simple bill splitter ...............................................................](#page4) | [2](#page4) |
| [1.1.2](#page5) | | [Billy: Scan & Split the bill................................................................](#page5) | [3](#page5) |
| [1.1.3](#page6) | | [Clever Bill Splitter............................................................................](#page6) | [4](#page6) |
| [1.2](#page7) | [Основни принципи, технологии и развойни среди за реализиране на](#page7) | | |
| [мобилни приложения .......................................................................................](#page7) | | | [5](#page7) |
| [1.2.1](#page8) | | [Android Studio .................................................................................](#page8) | [6](#page8) |
| [1.2.2](#page9) | | [OCR .................................................................................................](#page9) | [7](#page9) |
| [Глава 2.](#page11) |  | [Изисквания .........................................................................................](#page11) | [9](#page11) |
| [2.1](#page11) | [Основна функционалност ....................................................................](#page11) | | [9](#page11) |
| [2.2](#page11) | [Избор на програмен език и библиотеки ..............................................](#page11) | | [9](#page11) |
| [2.2.1](#page11) | | [Java ..................................................................................................](#page11) | [9](#page11) |
| [2.2.2](#page12) | | [Tesseract .........................................................................................](#page12) | [10](#page12) |
| [2.3](#page13) | [Структура на приложението ...............................................................](#page13) | | [11](#page13) |
| [Глава 3. Реализация на проекта ......................................................................](#page14) | | | [12](#page14) |
| [3.1](#page14) | [Реализация на Android приложението................................................](#page14) | | [12](#page14) |
| [3.1.1](#page14) | | [Activities .........................................................................................](#page14) | [12](#page14) |
| [Глава 4. Ръководство на потребителя ............................................................](#page15) | | | [13](#page15) |
| [4.1](#page15) | [Използване на приложението .............................................................](#page15) | | [13](#page15) |
|  |  |  | 18 |

[Заключение](#page17) [15](#page17)

[Използвана литература](#page18) [16](#page18)

[Фигури](#page19) [17](#page19)