**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**

**ДИПЛОМНА РАБОТА**

Тема: Android приложение за управление на принтери в търговски обекти

Дипломант : Научен ръководител:

Симеон Чакъров Ивайло Велков

СОФИЯ

2019

Увод

Във всеки търговски обект, независимо дали в него се извършват продажби на стоки или услуги, има ЕКАФП (Електронен Касов Апарат с Фискална Памет) или фискален принтер, свързан с компютър. И нека все пак да обобщим какво значи понятието „търговски обект“ и каква е разликата между ЕКАФП и фискален принтер.

Търговски обект е всяко помещение или съоръжение (например маси, сергии, бусове и др.) на открито или под навеси, от което се извършват продажби на стоки и услуги, в които заплащането не става по банков път, независимо от факта, че помещението или съоръжението може да служи същевременно и за други цели (офис, жилищно помещение), да е част от притежаван жилищен имот (гараж, мазе, стая) или да е производствен склад или превозно средство, от което се извършват продажби.

Разликата между фискалния принтер и ЕКАФП е това, че касовият апарат функционира самостоятелно – купуваш касов апарат и работиш с него – това е. Фискалният принтер не работи самостоятелно – свързан е с компютър и работи със специална програма – ресторантьорска или склад-магазинска. Работата с тази програма има и предимства, разбира се – поддържа складовите наличности, сигнализира за изчерпващи се стоки, улеснява ревизиите.

Примерна система предназначена за работа в бар/ресторант се състои от: Компютър, монитор, периферни устройства, фискален принтер, каса, баркод четец.

Какво става когато се развали някоя част от системата – компютърът изпуши, изгасне мониторът, баркод четецът не иска да работи? Или например е много трудно или невъзможно инсталирането на системата заради местоположението на различните компоненти? Вадиш касовия апарат? А как ще продължиш да следиш складовите наличности? На листче? Как ще покажеш подробна сметка на своите клиенти, които не отиват в магазинчето отсреща, само защото ти издаваш подробни бележки, за разлика от него, и сега ще ги разочароваш? Ако притежаваш заведение ще ги накараш да ровят в менюто за да си разделят сметката? Не! Реших да се захвана с решението на подобен проблем именно защото освен за самите клиенти, е неудобно и за самия търговец.

Затова започнах с конструирането на Smart Printer for Android - Android service, през когото човек може да управлява своя (не)фискален принтер.

# 

# ПЪРВА ГЛАВА

# Обзор за съществуващите решения

## **1.1 Подобни съществуващи решения**

След подробно и дълго проучване открих, че няма друг аналог на Smart Printer for Android, освен Desktop приложение, с което се запознах по време на лятната си производствена практика, организирана от моето училище. След запознаване на документацията, поричтане на коментари и много въпроси от моя страна към разрабочиците на това приложение, дойде идеята да се осъществи подобна апликация, но за Android, която освен функциите на Desktop приложението, да поддържа и функция за управление на повече от един принтер. Освен това, като се замислих, при разположение на уредбата би било много по-лесно и по-практично да се управлява през телефон, отколкото през компютър.

## **1.2. Развойни средства и среди. Проучване.**

Самото приложение заема ролята на сървър и приема XML като заявка, съдържащ инструкции за самия принтер, раздробява я и я изпраща по един от трите начина до принтера – чрез порт, LAN или Bluetooth. Първият екран съобщава дали сме свързани към локална мрежа, и ако е така ни показва IP-то й, а също така и порта, който се слуша от мобилното устройство. Във втория екран се избира държава и език, на когото искаме да пише в самото приложение, добавя се и активният принтер. Има бутон, който като се натисне излиза панел, където се изписва дали добавеният принтер е в изправност и готов за работа. Има още един бутон, който като се натисне, изписва характеристиките на принтера и може да се доконфигурират инструкциите изпращащи се към принтера. Съответно добавеният принтер може да се изтрие или да се редактира.

Има различни видове ресторантьорски и склад-магазински софтуерни продукти, предлагащи се от доста фирми, но според проучването ми, което се изрази в това да разпитам в квартални магазинчета, големи търговски вериги и различни заведения – нощни и за хранене, моят проект няма друг аналог – той може да се вгради във всяка една среда, нужен дали заради ненадеждната система(честото разваляне на различните устройства), или трудно постижимата инсталация заради разположението на различните компоненти на уредбата.

Повечето системи обикновено използват само един вид свързване – чрез порт, LAN или Bluetooth. Smart Printer for Android наистина е съвместим с всеки вид принтер – може да се свърже и по трите начина.

# ВТОРА ГЛАВА

# ПРОЕКТИРАНЕ НА СТРУКТУРАТА

## **2.1. Функционални изисквания**

Smart Printer for Android няма да е абсолютно автоматизиран продукт, който сам да принтира. То ще очаква да получи някаква заявка с това, което трябва да принтира самия принтер.

То е Android приложение, което само в себе си има много на брой драйвъри за различни принтери. В самите драйвъри са описани характеристиките на самия принтер(Baud rate – скоростта, с която печата принтера, code page-a му и др.), както и различни функции – общи и такива, използващи се в зависимост от това дали принтера е фискален или нефискален.

След като получи заявка, дали от настроен търговски софтуер, пригоден за Smart Printer for Android, или Postman, или лентата в браузъра, то започва да я раздрабя(има в себе си Parser), обработва и след това я

Приложението се дели на две основни части:

* Първият панел – loginPanel

Първият панел има етикет, изписващ името на приложението, а под него има два EditText-a. В единия EditText, ако си свързан към локална мрежа, показва IP-то, а другия – порта, през когото се очаква заявката, съдържаща XML-ът. Под двата EditText-a има още един етикет, който изписва съответно със зелено “Connected”, ако си свързан към локална мрежа, и съответно с червено „Not connected“, ако не си. Също така под гореизброените има бутон, водещ към втория панел.

* Вторият панел – SettingsScreen

Вторият панел има три Spinner-а, единият от които е да се избере в коя държава се използва принтера, а другият дава възможност да се избере езикът, на когото искате да изписва етикетите, обособяващи различните секции.

Чрез третия Spinner може да се избере активният принтер, тоест този, с когото работите и да се добави. Може да се добави повече от един принтер – тоест да работите с повече от един принтер и да печатате на повече от един принтер едновременно. При добавянето на принтерите, могат да се зададат скриптове:

* + - Headers – текст, който ще се изписва отгоре на бележката
    - Details – посредата
    - Totals – там, където пише общата сума на покупката
    - Footers – най-отдолу на касовата бележка

Има по един бутон за всеки вид скрипт, който добавя още един EditText, в когото може да пише човек, т.е. всеки вид скрипт може да има по повече от един ред – колкото ние поискаме.

След като се добавят един или повече принтери, чрез бутона, обозначен с плюс, можем да ги изтриваме и да ги редактираме. До самия Spinner, в когото се пълнят активните принтери, има още два бутона – този за редактиране, обозначен с моливче и такъв за изтриване на избрания принтер, обозначен с минус.

При цъкане на бутона за редактиране, излиза плавно панел от едната страна, в който можем да редактираме скриптовете, зададени при добавяне на устройството.

Идеята на тези скриптове е ако искаме да променим нещо, ако нямаме достъп до XML-a, който се изпраща.

Има и още един бутон, който като се цъкне, отново излиза от едната страна плавно панел, в когото биват изписани активните принтери и това дали те са в изправност и готови за печатане.

## **2.2. Съображения за избор на програмни средства и развойната среда**

Smart Printer for Android трябва да е лесно за използване от всеки потребител, търговец, на различна възраст. Затова то трябва да е направено така че човек директно да може да се ориентира, много семпло и просто откъм визуалната част. Поради тази причина избрах да използвам един много функционален език за създаване на Android приложения, а именно Basic4Android.

Едни от предимствата на езика са:

* + - Първото предимство на езика и средата за разработка е простият, но в същото време много мощен RAD (rapid application development) tool, който позволява на програмисти на всякакво ниво бързо да разработват реални приложения, не е трудно да се започне и да се научи – IDE-то и framework-а сами ще ти помогнат да завършиш работата си.
    - Цялостна среда за разработка и език за програмиране, фокусиран 100% върху развиването на Android приложения.
    - Компилира се до native bytecode и не са нужни Runtime библиотеки.
    - APK файловете са същите като APK файловете, направени с Java / Eclipse.
    - Можеш да преизползваш Java код, обгръщайки го като библиотека и да го свържеш от средата за разработка на B4A приложения.
    - Представянето/производтелността са много близки до тези на приложение, създадено с Java.
    - Обектно-ориентиран език за програмиране.
    - Бърз debugger, поддържащ бърз deployment (процес, който подготвя приложението да работи в съответната среда – включва инсталация и конфигурация, оптимизиране).
    - Debugger-ът също така е step-by-step.
    - Debugger-ът също така поддържа expressions watcher (минавайки през кода, може да се следи стойността в даден израз/променлива).
    - Изключително разтегателен език – не те ограничава по никакъв начин, освен големия набор от библиотеки, той поддържа и custom Java библиотеки.
    - Визуален редактор, поддържащ няколко екрана и резолюции.
    - Мощна функционалност за дизайнерски script-ове, чрез която лесно се създават сложни макети(layouts).
    - Поддържано е от всички Android устройства с андроид над 1.6.
    - Модерна среда за разработка с вградено автоматично довършване.
    - Голяма по обем документация и форум, в който при зададен въпрос, се отговаря адекватно и бързо.
    - Можеш да използваш B4J за да имплементираш back-end частта и можеш да използваш B4I за да създадеш. същото решение за iOS. 80% от кода от всяка от средите за разработка може да се използва в другите две.

Езикът е алтернатива на програмирането с Java и Android SDK. Платформата компилира до Java и след това отново го връща като Basic, използва смесен (Cross) компилатор.

Резултатът е изпълним код, предназначен за платформа различна от тази, на която е извършена компилацията. Използва се например при създаване на код за вградени системи (embedded systems), както и в Basic4Android, където няма възможност, или е по-трудно, да се създават програми.

B4A включва визуален дизайнер, който опростява процеса на изграждане на потребителски интерфейси, насочени към телефони и таблети с различни размери на екрана. Компилираните програми могат да бъдат тествани в емулатори на AVD Manager или на истински Android устройства, използващи Android Debug Bridge и B4A Bridge.

Самият език е подобен на Visual Basic и Visual Basic .NET, но е адаптиран към родната Android среда. Има разлики и прилики с Java, най-големия плюс е, че се пише много бързо и е много функционално, не е гъвкаво колкото Java, но по отношение на приложението, което е нужно за системата и ще се използва, е нужно просто бързодействие и лесно разгадаване на програмата.

За да може да работи приложението освен компилатора, който се предоставя с B4A платформата е нужно да се свърже към нея Java JDK, за да може да се установи компилация, Android SDK за да се инсталират платформите и инструментите за версиите на Andoid-а, който ще се използва. Също така най-новата версия на B4A е платена и е нужен лиценз.

По желание може да се използва B4A-Bridge, което компилира на устройството безконтактно чрез Wireless връзка, ако двете устройства са в една и съща мрежа. Може да се изтегли от Google Play безплатно.

B4A има пробна версия, която може да се използва 30дни и то не с всички функционалности, както ако се закупи. Може да се закупи стандартна версия, която включва поддръжка само 2 месеца, а може и Enterprise верия, включваща актуализиране на софтуера 2 години.

Другия малко по-гъвкав, но не практичен вариант, поради усложняване на работата за създаването на Android проложението е писането на Java на Android Studio. Но използването на Android Studio ще създаде повече работа по програмирането на това, не толкова сложно приложение.

Android Studio е официалната интегрирана среда за разработка (IDE) за операционната система Android на Google, изградена върху софтуера IntelliJ IDEA на JetBrains и специално разработена за разработка на Android. Той е достъпен за изтегляне за операционните системи Windows, MacOS и Linux. Това е заместител на Eclipse Android Development Tools (ADT) като основно IDE за разработка на естествени Android приложения. Android Studio бе обявено на 16 май 2013 г. на конференцията Google I/O. То беше в начален етап на предварителен достъп, като се започне от версия 0.1 през май 2013 г., след което бе въведен бета етап, като се започне от версия 0.8, която бе пусната през юни 2014 г. Първото стабилно изграждане бе пуснато през декември 2014 г., като се започне от версия 1.0. Сегашната стабилна версия е 3.2.1, пусната през октомври 2018 г.

## **2.3. Преглед на част от принтерите, свързващи се с приложението и работата с тях**

* + SPP-R300

Фигура 2.3.1



**Изисквания за Bluetooth принтерите:**

За да се свърже принтерът с Android мобилен телефон, използвайки Bluetooth, устройствата не трябва да са на повече от 10метра разстояние.

Това разстояние може да се скъси, ако например има някакви препядствия между двете устройства (да кажем стени). Ако друго устройство (като микровълнова печка или рутер за безжичен интернет), което използва същите честоти на предаване, е в близост до тях, предаването на информация може да бъде прекъснато. Преместете принтера и мобилното устройство от причинителя, имащ същите честоти на вълните на поне пет метра.

**Как да свържем принтера към устройство:**

1. Проверяваме дали всички устройства са включени, ако не са – ги включваме.
2. Пускаме Bluetooth програмата.
3. Намираме принтера, който трябва в случая да се казва SPP-R300.
4. Свързваме се с него и има вероятност да се изиска парола, която по подразбиране е зададена на „0000“.
5. Тестваме връзката, като принтираме .

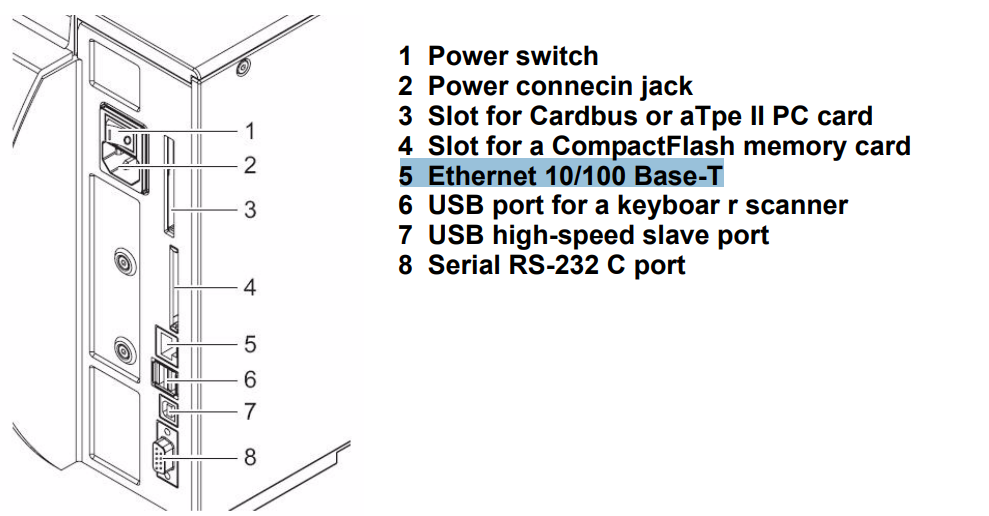
**Тези стъпки са валидни за устройства, свързаващи се с принтера, различни от:**

* Intermec PDA (Device: CN2B, OS: Pocket PC 4.20.0)
* SYMBOL PDA (Device: SYMBOL MC70, OS: Windows Mobile 5.0)
* HP PDA (Device: iPAQ hx2700, OS: Windows Mobile 5.1.1702)
* Mobile Compia PDA (Device: M3, OS: WinCE 5.0)
* InnoTeletek PDA (Device: MC3000, OS: WinCE 4.2)
* Widefly PDA (Device: WF35, OS: WinCE 5.0)
* Bluetooth Dongle (BlueSoleil IVT Stack)
* Bluetooth Dongle (Widcomm Stack)
  + PR300

Фигура 2.3.2



Фигура 2.3.3



При LAN принтерите идеята е да се свърже чрез Ethernet cable към мрежата и през нея да се изпращат от Android Service-a инструкции, подадени преди това от даден софтуер или проста заявка, като XML структура.

* Rongta RP326 COM

Фигура 2.3.4



Този вид принтер може да се свърже и чрез мрежа, и чрез сериен порт(Serial Port), но в случая ще го използваме като пример за свързване на принтера към мобилното устройство(Android Service-a) чрез сериен порт. Свързва се с телефона чрез кабел(нужен е преходник от Serial Port към USB и след това OTG (On – The - Go) USB - преходник от USB-то към мобилното устройство), като през всичко това преминават инструкциите, подадени в XML формат. OTG е съкращение от USB OTG (Universal Serial Bus On – The - Go) и представлява стандарт, който позволява на устройствата да ‘разговарят’ помежду си. Например, може да свързвате телефона си към четци на SD карти, камери, дори мишка или клавиатура и други различни перифрени устройства. USB (Универсалната серийна шина) е серийна шина за свързване на компютър с огромен набор от различни периферни устройства.

Фигура 2.3.5 – USB OTG



## **2.4. Принцип на софтуерения модел**