**ТЕХНОЛОГИЧНО УЧИЛИЩЕ “ЕЛЕКТРОННИ СИСТЕМИ”**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**ДИПЛОМНА РАБОТА**

Тема: Мобилно приложение социална мрежа за готвачи и любители на готвенето – „EverydayChef“

Дипломант: Научен ръководител:

*Мартин Пенев Александър Илиев*

СОФИЯ

2020

Дата на заданието: 15.11.2019 г. Утвърждавам:..............................

Дата на предаване: 15.02.2020 г. /проф. д-р инж. Т. Василева/

**ЗАДАНИЕ**

**за дипломна работа**

на ученика Мартин Василев Пенев 12A клас

Тема: Мобилно приложение социална мрежа за готвачи и любители на готвенето - “EverydayChef”

1. Изисквания:
   1. Регистрация и логин чрез нормален имейл и чрез фейсбук
   2. Създаване и менажиране на семейства или взимане на съществуващи такива от фейсбук
   3. Списък за пазаруване за всяко семейство
   4. Налични продукти за всяко семейство
   5. Създаване и менажиране на рецепти
   6. Оценяване на рецепти
   7. Любими рецепти на всеки един потребител
   8. Търсачка на рецепти по налични продукти
   9. Нотификации при различни събития
2. Съдържание 3.1 Обзор

3.2 Същинска част

3.3 Приложение

Дипломант :...........................................

Ръководител:..........................................

/Александър Илиев/

Директор:................................................

/ доц. д-р инж. Ст. Стефанова /

**УВОД**

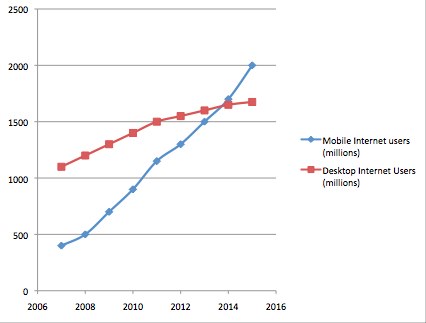
В ежедневието си човек много пъти на ден си задава въпроса: „Какво ще закусвам/обядвам/вечерям днес?“. Понякога липсват продукти за желаното от него ястие, друг път липсват идеи за това какво да си сготви или знания за това как да го направи, а често липсват и трите. Приложението „EverydayChef“ има за цел да отсрани подобни проблеми и да улесни доколкото е възможно живота в кухнята. Свързвайки готвачи, любители на готвенето и хора без много опит в сферата, софтуерът се стреми да изгради социална мрежа, специално създадена за изкуството на приготвяне на храна. В нея всеки участник може да черпи от опита на другите и да споделя своя собствен, да получи обратна връзка за работата си и да оценява от своя страна тази на другите. Това основнo се изразява в създаване на рецепти и коментирането и оценяването на такива. / Следва дописване /

**ПЪРВА ГЛАВА**

***Проучвателна част***

***Технологии и подобни приложения***

* 1. **Защо мобилно приложение**
     1. **Мобилни устройства**

Днешно време технологиите са навсякъде около нас и съответно приложение като „EverydayChef“ може да бъде разработено за всякакви устройства. Това повдига въпроса защо е избрана точно мобилната среда.

(ще го оправя)

*Фиг. 1.1 Използваемост на мобилни и настолни устройства за достъп до интернет*

Както се вижда на графиката, бройката на потребителите на мобилни устройства през 2016 е вече по-голяма от тази на хората, използващи настолни такива. Само една година след това, през 2017 вече 67% от глобалния трафик идва от джобната технология, което вече ясно изгражда тенденцията и за в бъдеще – използваемостта на мобилните устройства за връзка с интернет расте, за разлика от тази на десктоп компютрите. Много по-удобно е за един човек да използва телефона си, за да свърши каквато работа му е нужна, където и когато му е нужна. Това е главната причина да избера тази среда като основна за моето приложение.

* + 1. **Мобилно приложение или мобилен уеб[1]**

Връзката през мобилното устройство до интернет се осъществява чрез мобилно приложение или чрез мобилен уеб. Ако компании с по-стабилни финанси могат да си позволят разработката и на двете за техния продукт, то в повечето случаи, особено в началото на реализацията на проект, се налага да се избира една от двете опции. Изборът обикновено се основава на разликите в цената на изпълнение, целта на продукта и клиентите, за които е предназначен. Въпреки това изследвания показват, че хората предпочитат мобилните приложения пред мобилните уеб сайтове. Ето защо:

* По-добра персонализация

Когато се очаква продуктът да се използва ежедневно, мобилното приложение е най-добрия вариант поради възможнстта за настройването му по вкус на потребителя, менажирането на различни акаунти и поддръжката на важна информация под ръка.

* Push notifications (Директни съобщения)

Вградена функция на мобилните приложения е изпращането на директни съобщения към потребителя при различни събития. Тази способност за комуникация с клиента често накланя везните при много бизнеси към създаването на мобилно приложение за начало.

* Лесен достъп до устройството и неговите функционалности

През мобилното приложение много лесно се използват камерата, контактите, историята на обажданията и подобни функции на телефона. По този начин може да се осигури много по-приятна и улеснена работа с продукта за потребителя, както и се намалява времето, за което той би свършил нещо, изискващо тези функции, чрез мобилния уеб.

* Няма нужда от интернет връзка

Много полезно предимство на мобилните приложения е това, че могат да се ползват и при липса на обхват. Това е, защото понеже приложението е инсталирано на самото устройстройство то запазва локално последното заредено съдържание и винаги показва него. При наличие на интернет връзка това съдържание се обновява.

* По-лесни и удобни за използване

Удобоството винаги се търси от потребителите без значение от целта на продукта. В тази насока мобилните приложения отново превъзхождат своя опонент с опростеността и интуитивността си. Освен това, за разлика от уеб сайтовете, приложенията използват жестове, което още спомага за тяхното приятно използване.

* По-висока скорост

Едно добре направено приложение може да процедира много по-бързо от една уеб страница. Една от причините е именно локално запазената (кешираната) информация в приложението.

* 1. **Използвани технологии**
     1. **REST**

„EverydayChef“ е имплементирано на базата на REST. REST[1] (Representational state transfer) представлява софтуерна архитектура, предоставяща дефинирани стандарти за връзката между компютърните устройства в мрежата, за по-лесна комуникация помежду им. Ще бъдат описани само най-отличителните черти на архитектурния стил. Характерно за създадените по този начин системи (наричани RESTful системи) е, че при тях са разделени понятията клиент и сървър, както и на сървъра не му трябва да знае какво е състоянието на клиента във всеки един момент от време и обратното.

* + - 1. **Разделение на клиент от съръвър**

При REST архитектурата клиента и сървъра се изграждат напълно отделно без едното да се интересува от другото. Това значи, че могат да бъдат променяни по всяко време и това да не засяга другата страна.

По този начин интерфейсът по-лесно може да бъде интегриран на друга платформа, и има много възможности за подобрение.

Чрез архитектурния стил различни клиенти могат да се закачат за същите пътища на интерфейса, да извършват еднакви операции, както и да получават еднкави данни.

* + - 1. **Не се знае текущото състояние на отсрещната страна**

При всички приложения, следващи REST, сървърът не му трябва да знае текущото състояние на клиента и обратното. По този начин и двете страни разбират всяко едно съобщение помежду си без да се налага да виждат предишните. Това се осъществява чрез използването на ресурси вместо на команди. Това са обекти, документи или просто информация, която трябва да бъде запазена или обработена по някакъв начин и/или изпратена към друго устройство.

Това спомага за ефективността и скоростта на RESTful приложенията, както и за тяхното лесно разрастване, тъй като отделните компоненти могат да бъдат променяни и преизползвани без това да се отразява на системата като цяло.

* + 1. **Android studio – frontend**

Частта от приложението, която потребителя ще вижда и с която ще работи(frontend), е писана чрез Android Studio [3]. Това е официалното IDE(integrated development environment - среда за разработка), специализирано за операционната система андроид. Излиза за първи път на 16.05.2013г., а първата стабилна версия е обявена декември 2014 година.

Поддържа се както новоизлезлият и обявен за предпочитан от Google език Kotlin, така и старите основни езици като Java и C++. Предоставени са множество функционалности, които улесняват разработката на приложения и увеличават удобството на използване като:

* Gradle build
* Специфични за андроид рефактори и quick fix опции
* Lint средства за засичане на различни проблеми
* ProGuard интеграция
* Темплейт-базирана обработка на дизайна и компонентите на приложенията
* Среда за обработка на дизайна
* Виртуален емулатор
  + 1. **Spring boot – backend**

Другата част от приложението – сървъра(backend), е написана чрез Spring Boot. Това e open-source micro framework, поддържан от компания на име Pivotal. Разработен е върху framework на име Spring. Чрез него се осигурява автоматична конфиргурация на Spring проект, с което се спестява много време и усилия на програмистите.

Spring излиза през 2002 година и много бързо се утвърждава като доминиращ начин, по които се пише Java код. Разработен е open-source (целият код е достъпен) и има за цел да направи писането на сървърната част на Java по-лесен и удобен. През 2012 година започват дискусии по начин за подобрение на framework-a. 06.07.2013 Фил Уеб пише, че вместо да се променят вече съществуващия Spring, са започнали да работят по нов проект, наречен Spring Boot, който да оптимизира стария. По тази причина може да се каже, че Spring Boot излиза за да се направи разработката на уеб приложения с Spring по-достъпна и разбираема.

* 1. **Подобни приложения** 
     1. **SuperCook -** <https://www.supercook.com/#/recipes>

**ИЗТОЧНИЦИ**

1. Приложение или уеб - <https://www.whizsolutions.co.uk/7-reasons-mobile-apps-better-website/>
2. REST - <https://www.codecademy.com/articles/what-is-rest>
3. “Android Studio” - <http://developer.android.com/tools/studio/index.html>