

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Software Science

Aleksander Fadejev 155047IAPB

CREATING MOBILE APPLICATION FOR THE CONVENIENT USE OF TALTECH DINING

Bachelor's thesis

Supervisor: Roger Kerse
MSc

Tallinn 2018

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Aleksander Fadejev 155047IAPB

**MOBIILRAKENDUSE LOOMINE TALTECH
SÖÖGIKOHTADE MUGAVAKS
KASUTAMISEKS**

Bakalaureusetöö

Juhendaja: Roger Kerse
Magister

Tallinn 2018

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Aleksander Fadejev

07.01.2019

Abstract

The purpose of this bachelor's thesis is to create a functional and user-friendly mobile application, that solves everyday problems of choosing a meal and makes it quick and comprehensive. The mobile application user can choose the appropriate TalTech eatery, using different filters or based on the prices of meal, type of meal or dietary preferences. To achieve this purpose, author analysed different types of mobile applications and analysed existing mobile application with similar functionality.

The main problems in this work are the parsing data from Facebook and Web, processing data with saving it to the database and allowing them to be used within the mobile application.

The result of this work is a mobile application that can be downloaded from the Google Play Store and everyone can use it.

This thesis is written in English and is 40 pages long, including 6 chapters, 27 figures and 2 tables.

Annotatsioon

Bakalaurusetöö eesmärgiks on luua mugavat ja funktsionaalset mobiilirakendust Apache Cordova platvormi abil, mis lahendab igapäevaselt söögikoha valiku probleeme ning teeb selle kiireks ja ülevaatlikuks. Mobiilirakenduse kasutaja saab valida endale sobiva söögikoha, kasutades erinevaid filtreid, lähtudes roogade hindadest, tüübist ja dieetlistest eelistustest. Selle eesmärgi saavutamiseks, töö autor analüüsis mobiilirakenduste tüüpe ja olemasolevaid mobiilirakendusi sarnase funktsionaalsusega.

Töö põhilised probleemid on andmete välja võtmine Facebook'ist ja veebist, andmete töötlemine, salvestamine andmebaasi ning võimaldamine need kasutada mobiilirakenduse sees.

Töö tulemuseks on mobiilirakendus, mille saab alla laadida Google Play Store'st ning mida huvilised saavad kasutada.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 40 leheküljel, 6 peatükki, 27 joonist, 2 tabelit.

List of abbreviations and terms

TalTech	Tallinn University of Technology
HTML	HyperText Markup Language Describes the structure of Web pages using markup
CSS	Cascading Style Sheets Describes how HTML elements are to be displayed on screen
API	Application programming interface
PHP	Hypertext Preprocessor Server-side scripting language
OS	Operating System
cURL	Command line tool for getting or sending files using URL syntax
MySQL	Database system used on the web
UI	User Interface
FTP	File Transfer Protocol
NPM	Package manager for JavaScript
SDK	Software Development Kit
PDF	Portable Document Format

Table of contents

1	Introduction	10
1.1.	Problem and background	10
1.2.	Purpose.....	10
1.3	Methodology.....	11
1.4	Thesis overview	11
2	Technology.....	12
2.1	HTML, CSS, JavaScript	12
2.2	PHP	13
2.3	Facebook Graph API	13
3	Analysis types of mobile applications	16
3.1	Native applications	16
3.2	Web applications.....	17
3.3	Hybrid applications.....	18
3.3.1	Apache Cordova	19
3.3.2	Adobe PhoneGap.....	20
3.3.3	Ionic	20
3.4	Difference between types of mobile applications.....	21
4	Analysis of existing mobile applications.....	23
4.1	North Dakota Dining	23
4.2	ClarkDining	24
5	Mobile Application Implementation	26
5.1	Parsing and processing data from Facebook.....	26
5.2	Parsing and processing data from PDF.....	27
5.3	Cordova setup	29
5.4	Mobile application	31
5.4.1	Search	32
5.4.2	Filters.....	33
5.4.3	Google maps.....	35
5.4.4	Feedback from user	36
6	Summary.....	38
	References	39
	Appendix 1 – Mobile application in Google Play Store.....	40

List of figures

Figure 1 Facebook Graph API response.....	14
Figure 2 Facebook Graph API response on ID	14
Figure 3 Facebook post information.....	15
Figure 4 Cordova application architecture [10].....	19
Figure 5 Main menu of North Dakota Dining application	24
Figure 6 ClarkDining Main menu (a), (b) menu design	25
Figure 7 cURL in PHP	27
Figure 8 composer.json file	28
Figure 9 Command for composer to download the bundle	28
Figure 10 Daily cafeteria menu	28
Figure 11 Parse data from PDF document	29
Figure 12 Array of PDF documents	29
Figure 13 Adding platforms to project	29
Figure 14 Commands to build and run Cordova project	30
Figure 15 Cordova successful launch.....	30
Figure 16 Package.json file	31
Figure 17 Main data about cafeteria	31
Figure 18 Feeds information	31
Figure 19 Feeds response	32
Figure 20 Mobile application main menu	32
Figure 21 HTML for search.....	32
Figure 22 Implementation of search function.....	33
Figure 23 Applying search in application.....	33
Figure 24 Applying filter to menu (a), (b) filtered menu	34
Figure 25 Key-value pairs	35
Figure 26 Google maps integration into application	36
Figure 27 Feedback section before submitting (a), (b) after submitting	37

List of tables

Table 1 Hybrid applications properties	21
Table 2 Capabilities of mobile applications	22

1 Introduction

1.1. Problem and background

The author considers, that finding a high quality and low-priced caterer has always been topical for students and university workers as nutrition is an important part of our lives and energy source.

Cafeterias offer a wide variety of dining options and offers of these days can be viewed either on Facebook [1] or on websites [2].

After analyzing the mobile applications market, the author did not find any alternative applications in Estonia. In addition, several similar applications present on the market, however due to inaccessibility in the Estonia region, it was not possible to download any. There are relatively few mobile applications, created for university cafeterias. The author proposes to create a mobile application, which will help students and university employees in short time determine with appropriate dining place.

1.2. Purpose

The main purpose of this work is to create and put into use a mobile application with the following functionality:

- Collect and show data from different sources about cafeteria menu
- Show user a map, where cafeteria is located
- Give user a possibility to filter food by food type and price
- Give user a possibility to search the dish of interest
- Give user a possibility to leave feedback
- Give user a possibility to switch between days, to see different daily menu
- Notify user, when the new menu is added

1.3 Methodology

Before starting implementation, it is necessary to find all TalTech cafeteria service providers and their menu publishing methods. After that, explore mobile application types, technologies, programming languages and choose most appropriate.

To accomplish purpose of this work, author have to implement a mobile application, using HTML, CSS and JavaScript through Cordova Apache development platform. The server-side part of the mobile application is written in PHP programming language and communicating with MySQL database. For reading posts data from Facebook, author will use Facebook Graph API. The author will analyze types of mobile applications, as well as mobile applications with similar functionality.

1.4 Thesis overview

In the first part of thesis, author in details analyses technology stack of this work, also libraries and API's.

In the second part of thesis, author analyses types of mobile applications and compared them in detail.

In the third part, author analyses the existing mobile applications with similar functionality.

In the last part, author described steps of creating mobile application.

2 Technology

2.1 HTML, CSS, JavaScript

HyperText Markup Language (HTML) is the most basic building block of the Web. It describes and defines the content of a webpage along with the basic layout of the webpage [3]. HTML adds markup to plain text. Hypertext contains links that link web pages to each other, making the World Wide Web what it is today. HTML supports both images and other media content, such as video and audio. HTML is a language that describes the structure and semantics of the contents of a web document. The content of a web page is marked with tags, such tags are ``, `<p>`, `<div>` and many others. These elements form the building blocks for any website.

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML. CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. [4] For example, using CSS it is possible to change the font size, font color, line spacing, element width and height, add animation.

JavaScript (JS) is a full-fledged dynamic programming language that is applied to an HTML document and can provide dynamic interactivity on websites. JavaScript itself is quite compact, but very flexible. [5] Developers have written many tools on top of the main JavaScript language, which unlock a huge number of additional functions with little effort. These include:

- API built into browsers that provide various functionality such as dynamic HTML creation and CSS style setting.
- Third-party APIs allow developers to incorporate functionality into their sites from other developers, such as Twitter or Facebook.
- Also, some third-party frameworks and libraries can be implemented into HTML, which will allow to speed up the creation of sites and applications.

2.2 PHP

For server side application development was used PHP - a server-side scripting language designed for Web development. PHP code may be embedded into HTML code.

PHP supports libcurl, a library created by Daniel Stenberg, which allows connecting and communicating to several types of servers with many different types of protocols. Libcurl currently supports the http, https, ftp, gopher, telnet, dict, file and ldap protocols. Libcurl also supports HTTPS certificates, HTTP POST, HTTP PUT, FTP uploading, HTTP form-based upload, proxies, cookies and user + password authentication.

PHP/cURL is the module for PHP that makes it possible for PHP programs to use libcurl. [6]

2.3 Facebook Graph API

The Graph API is the primary way to get data into and out of the Facebook platform. It is low-level HTTP-based API, that applications can use to programmatically query data, post new stories or posts, manage ads, upload photos and perform a wide variety of other tasks. The Graph API consists of the following elements:

- Nodes – in fact, these are separate “objects”, for example (User, Comment, Page, Post).
- Edges – the connections between a collection of objects and a single object, for example (Comment to Photo).
- Fields – data about an object, such as a User’s birthday.

Using nodes, it is possible to get data about a specific object, edges allow collecting object collections connected with a single object and fields help to learn data about one object or each of them within a collection. This approach provides the whole design with no less qualitative detectability than the REST API, but it still shows the client to optimize the extraction of data.

Graph API is based on the HTTP protocol and works with any programming language that has an HTTP library, for example (cURL). In other words, the Graph API can be used directly in the browser. For example, author requests this URL in browser:

<https://graph.facebook.com/facebook/picture?redirect=false>, which is equal to performing this cURL request: `curl -i -X GET \ [7]`

`https://graph.facebook.com/facebook/picture?redirect=false&access_token={valid-access-token-goes-here}`

As a result of this request, author can get Facebook logo URL and logo dimensions.

```
{
  "data": {
    "height": 50,
    "is_silhouette": false,
    "url": "https://scontent.xx.fbcdn.net/v/t1.0-1/p50x50/12006203_10154088276211729_2432197377106462187_n.png?_nc_cat=0&oh=d9e0a1624d9d6df30864518903bc6c75&oe=5B628023",
    "width": 50
  }
}
```

Figure 1 Facebook Graph API response

For most Graph API queries, an access token is required and the easiest way to get an access token is to integrate the login through Facebook. With the help of access tokens, the application can interact with the Graph API. Access token perform two functions:

- Allow access to Facebook user data without a password
- Allow to identify application, the User who is using application, and the type of data the User has permitted to access.

Nodes are individual objects, each with a unique ID. Read operations always begin with node. There are many page objects and each of them has its unique ID. In this work author needs to get node of “Rahva Toit” cafeteria, which is 357117511056357, so URL will be https://graph.facebook.com/357117511056357?access_token, the response from Facebook is

```
{
  "name": "Rahva Toit",
  "id": "357117511056357"
}
```

Figure 2 Facebook Graph API response on ID

In the thesis author needs to get all posts from this page, for that operation usually use edges. With edges, it is possible to get collections of related nodes. The /page nodes have a /posts context that returns all publication posts on the page: https://graph.facebook.com/357117511056357/posts?access_token. All posts data for that page is returned in JSON format.

```
{
  "created_time": "2018-04-27T07:58:57+0000",
  "message": "Raamatukogu kohviku men\u00fc\u00fc 27.04:\n\nSealiha-  
kartliklimbi supp / Pork potato dumpling soup 1.70\nSeljanka /  
Solyanka soup 1.90\nLihavaba: Metsaseene-kruubi panniroog / Mushroom-  
groats dish 3.60\nMakaronid hakklihaga / Macaroni with minced meat  
3.50\nHakk-kotlet p\u00e4ikesekuivatatud tomatiga / Minced meat cutlet  
with sundried tomatoes 3.70\nTursafilee kartulihelbe koorikus / Fish  
fillet in potato flake coating 3.85\nKanafilee juustu paneeringus /  
Chicken fillet in cheese coating 4.50\nLambapada / Lamb hotpot 5.-  
\n\nHead isu!\nNB! Men\u00fc\u00fc v\u00f5ib esineda muudatusi...",
  "id": "357117511056357_1331685880266177"
}
```

Figure 3 Facebook post information

3 Analysis types of mobile applications

Currently, mobile applications are in great demand, because people need them for each of their small goals. Mobile applications are classified based on the technologies used to create them and the platforms on which they are able to work.

There are three main types of mobile applications.

3.1 Native applications

The native applications are native to each platform, whether it is iOS or Android and is written specifically for it in a programming language. To write a native application for iOS, Swift or Objective-C will be used, for native Android applications, Java or Kotlin are suitable.

Since this kind of applications are designed for one specific platform, it can take the full advantage of the device's features — it can use the camera, GPS, accelerometer, compass, the list of contacts and so on. Native applications can fully or partially work with a missing Internet connection, so users are less dependent on the quality of communication and can use the application when and where it suits them.

The main distinctive features of abilities:

- Highly responsive
- Fastest speed for users
- Efficient use of phone resources: battery, memory
- Perform better on the devices
- Very large documentation database: on the Internet and on book shelves of stores can be found a lot of information about the development of applications for such mobile platforms, like Android or iOS, as well as study materials, which are supplied by mobile operating systems developers and are actively supported by developer's community.

Native applications are usually developed using integrated development environments that provide tools for creating and debugging mobile applications.

Native applications have disadvantages:

- High price and long development time
- Complicated and expensive maintenance
- Application multiple versions support

3.2 Web applications

Mobile web applications or HTML5 applications are the web applications to render pages on web browsers running in mobile devices and using standard web-technology: HTML, CSS, JavaScript. This approach involves the development of cross-platform applications which will work on different in terms of software platform devices, such as Android, iOS or Windows. This kind of applications also work on PC modern web browsers. Mobile HTML5 application is usually a web page or set of linked web pages with a specially designed display on mobile devices with markup. Modern HTML5 applications are platform independent and may be satisfied by any modern mobile web browser. Entry threshold for this application development is usually lower, but the layout specifics of web pages leads certain difficulties: software platforms differently implement some display functions on web page. [8]

The main advantage of mobile HTML5 applications is the ability to update the content and structure of the application without need to release a separate update (build). Modern development of such applications is simplified by libraries, like jQuery Mobile, Sencha Mobile, which provide UI components that are not inferior in performance to their native analogues. These applications do not require any memory space in device. The main disadvantages of HTML5 applications are:

- Lack of secure local data storage
- Basic UI components
- Lack of optimization for a specific device, which usually leads to loss of productivity

- Lack of centralized distribution applications: mobile technology platforms provide application stores only for native products.
- Different degrees of support for HTML5 web-browsers of mobile operating systems
- As all personal databases are saved on a server, users can get access from any device whenever there is internet connection. That is why using such apps with poor internet connection would result in bad user experience.
- Performance is inextricably linked due to browser work and network connection

3.3 Hybrid applications

Hybrid applications come as a result from implementation of the HTML5 application inside the native container, which allows to combine the best sides of both approaches (Native and Web).

The speed of hybrid application will depend completely on the speed of the user's browser. This means hybrid applications will almost never run as fast as a native application runs. The advantage of hybrid applications is that it can be built on a single base, which allows adding new functionalities to multiple versions of application. With native applications, it is required to replicate every new feature that is needed to introduce for each platform. [9]

In that project author decided to make a hybrid application, mainly because of time limit and mobile application needs to be easy to use and do not consist of special operating system gestures and device access. Next it was necessary to decide, what platform is best to implement application. Below is a brief summary of each of the applications. (Table 1) shows the results of the analysis performed in accordance with the selected requirements. All the presented platforms provide for and independent design and development of the mobile application.

3.3.1 Apache Cordova

Apache Cordova is a platform that allows developing mobile applications on different platforms, by embedding the browser in a mobile application. Thus, mobile application is, in fact, a mini-browser that shows one single site – application. [10]

All resources can be placed in the distribution package of the application to accelerate the download or it can be downloaded from the server if necessary.

By default, Cordova provides only basic browser capabilities that are available on this mobile device, but it allows to extend the set of functions available in the browser by using plug-ins. Each plug-in provides a unified interface that can be used from a browser on different platforms. And if the supported CSS / JavaScript functions differ in each browser from the version of the operating system or platform, the Cordova functional tries to provide unified functionality for all supported versions of mobile OS.

Even though this approach has its limitations in the form of the need to consider differences in mobile browsers, it still allows creating high-quality cross-platform applications.

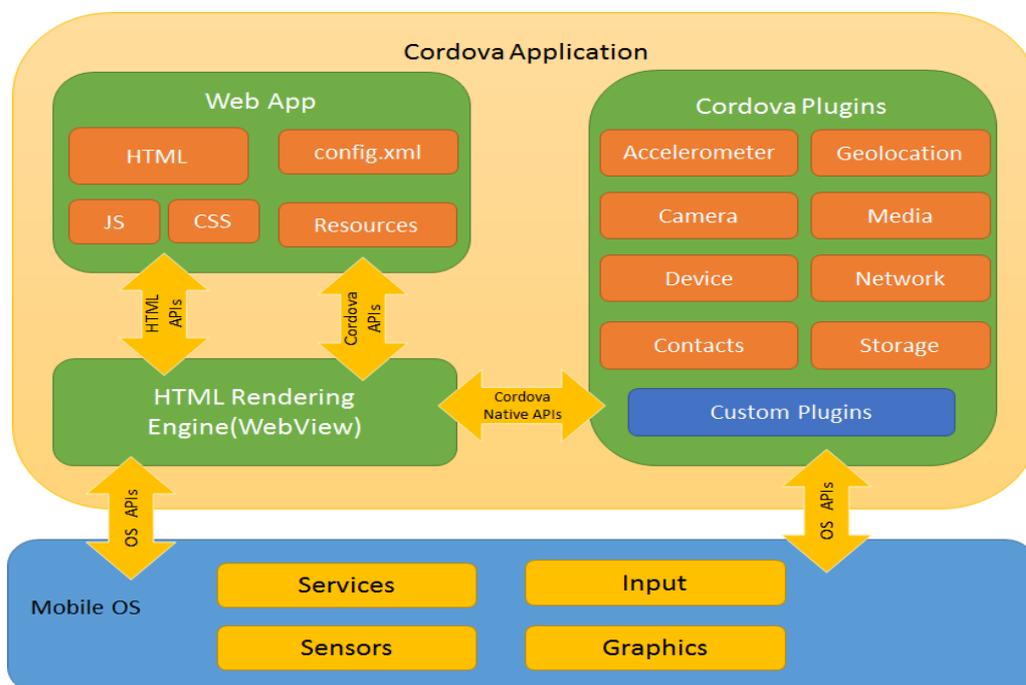


Figure 4 Cordova application architecture [10]

3.3.2 Adobe PhoneGap

PhoneGap appeared around 2009 in startup called Nitobi. This framework has open source code and provides access to its own development environment through the built-in native application component of the Android-based Web View platform. The goal of PhoneGap is to provide the conditions for creating a mobile application using pure web technologies, such as HTML, CSS, JavaScript. At the same time, the developer should have the opportunity to use (if necessary) native code. The output is wrapped application with the ability to interact with system functions (conventional browser wrappers do not have access to the system in whole). [11]

In 2011, Adobe bought Nitobi along with the rights to the PhoneGap brand. Since then, PhoneGap has come to be called Cordova. The project was transferred to the Apache Software Foundation.

At first, the differences between Cordova and PhoneGap were minimal. But Adobe, added its own services to the PhoneGap ecosystem. This is how the PhoneGap Build service was created.

3.3.3 Ionic

Ionic is a framework for creating hybrid applications using HTML, CSS, JavaScript. It comes with a set of components and user interface functions that can be used to create fully functional and attractive mobile applications. Ionic built on the Cordova stack. It is not possible to create mobile applications using only Ionic, because it processes only the user interface part. It should work with Angular, which processes application logic. In addition to the platform and ready-made UI components, Ionic has a thoughtful command line interface (CLI) that allows to generate icons, splash screens, launch the livereload server in the browser with the app for debugging and build applications with short commands in the console. Also, worth mentioning is the strong and detailed documentation with many examples.

	Adobe PhoneGap	Apache Cordova	Ionic
Mobile platforms	iOS Android Windows Phone 7/8 BlackBerry webOS Bada	iOS Android Mac OS X Windows Windows Phone 8 Browser BlackBerry Amazon Fire OS Firefox OS Ubuntu Tizen Bada LG webOS Symbian OS	iOS Android Windows 10 BlackBerry
Technology stack	HTML, CSS, JavaScript	HTML, CSS, JavaScript	HTML, CSS, Angular
Price	1 app free/ \$9.99/month	Free	Starter package is Free. Developer Package \$29/month

Table 1 Hybrid applications properties

3.4 Difference between types of mobile applications

Mobile development is a constantly moving target. Every half year, there is a new mobile operating system, with unique features only accessible with native APIs. The containers bring those to hybrid apps soon thereafter, with the web making huge leaps every few years. The results of the analysis are shown in the following table:

	Native	Web	Hybrid
Graphics	Native APIs	HTML, Canvas	HTML, Canvas
Performance	Fast Wide access to device functionality. Structure, content and visual elements are stored in device memory.	Slow Performance is inextricably linked due to browser work and network connection	Slow Most of data should be loaded from a server.
Distribution	App stores	Web	App stores
Offline storage	Secure file storage	Shared SQL	Secure file storage and shared SQL
Development cost	Usually higher than hybrid or web, if applications are developed for multiple platforms	The lowest cost due to single code base	Low cost, but require high skills for hybrid tools
Connectivity	Online and offline	Online and offline	Online and offline
Device features	Wide access to any device APIs	Only some of device APIs may be used (such as geolocation)	Low-level features (such as gyroscope or accelerometer)
Development skills	Objective C, Java, Swift, .NET (C#)	HTML5, CSS, JavaScript, JS frameworks	HTML5, CSS, JavaScript, Cross platform Mobile Development Frameworks
When to use	Applications with wide requirements, Single platform applications	Applications with limited resources	Multi-platform applications

Table 2 Capabilities of mobile applications

4 Analysis of existing mobile applications

The author found and selected two mobile applications with similar functionality:

- North Dakota Dining
- ClarkDining

4.1 North Dakota Dining

North Dakota Dining is a one-page application that allows user to view the different meal options at the Residence West, and Union Dining Centers at North Dakota State University. All meal information is pulled directly from NDSU Dining website. Author downloaded application and tested it.

This mobile application has following functionality:

- User can change date and see menu 9 days in advance.
- User can choose type of food (breakfast, lunch, dinner).
- User can see two cafeteria menus.
- User can see, which food is vegan and gluten free

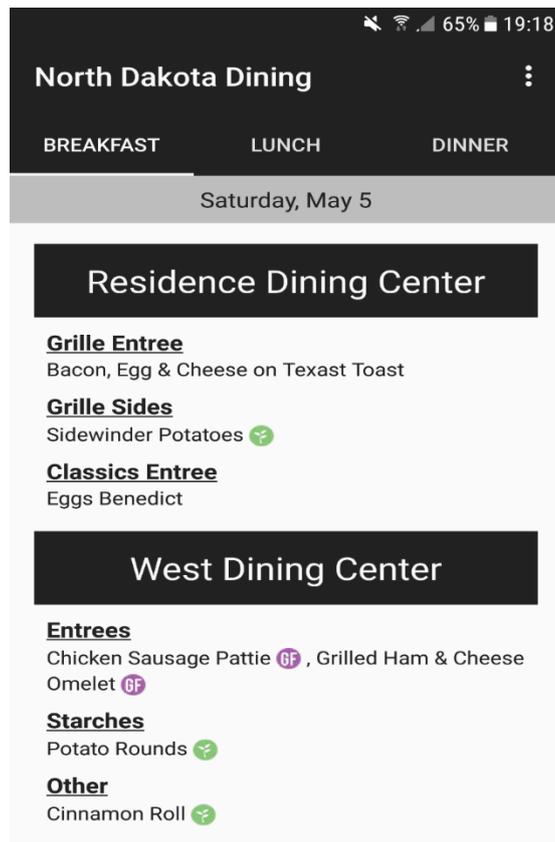


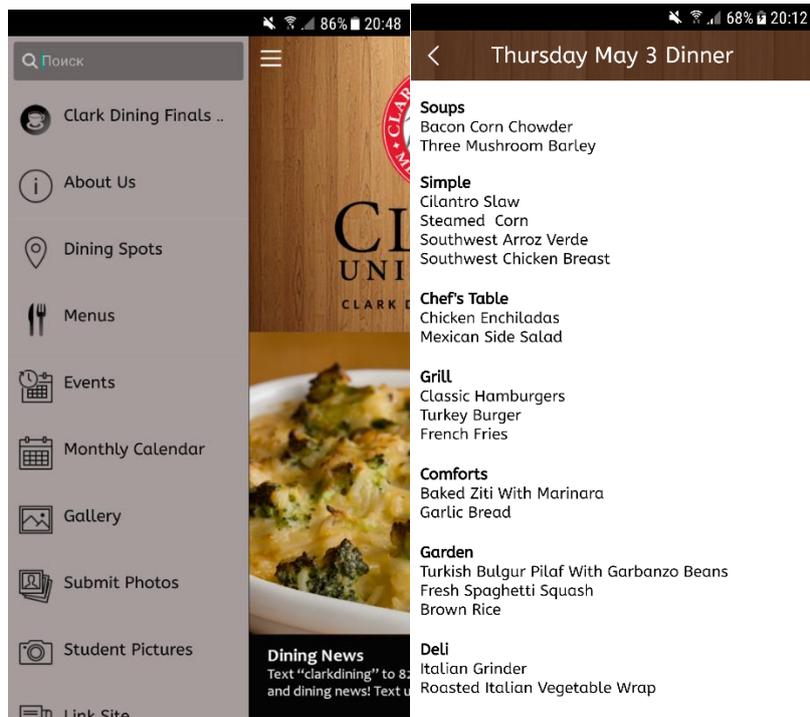
Figure 5 Main menu of North Dakota Dining application

The functionality of this application is not large, but it is enough. It is also worth noting a plain design of application, so that each user will easily find necessary information.

4.2 ClarkDining

ClarkDining mobile application has a much larger functionality than a North Dakota Dining. In the application user can see menu, dining spots, upcoming events, read articles about nutrition plans and allergens, add own cafeteria's food photo, watch other students photos, leave a feedback and also user can get acquainted with their online store.

The menu is designed simply for the user, who needs to select the day and time (breakfast, lunch or dinner), and after that application will show the list of places and their menu.



(a)

(b)

Figure 6 ClarkDining Main menu (a), (b) menu design

5 Mobile Application Implementation

5.1 Parsing and processing data from Facebook

Parser or grabber is a script or program that is used to collect information from websites for subsequent posting on own resources. Initially, parsing was the process of searching for specific information in a large text fragment, as well as breaking up data into meaningful parts. Parsing is used in programming, analytics. It can be useful in any area, where it is possible to work with string data.

In the thesis author used parser for full copying materials from websites with the subsequent placement of these materials on his resource. Work with the parser was carried out according the following algorithm:

- Search for a website with the necessary information.
- Extracting text according to the regular expression, prescribed in the code.
- Formation of text, including analysis of content, sifting of unfit material and saving the page.
- Transition to other website for further data collection.
- Presentation of extracted data in digestible form.

In the mobile application PHP was used as a programming language to write parser. This server language is comfort for creating parsers. It has a built-in library, called libcurl, with the help of which the script connects to any types of servers. PHP supports regular expressions, through them parser processes the data. Regular expressions are described by the Unix syntax, which although considered obsolete, but is widely used due to property of backward compatibility. PHP gets along well with HTML, because it was created for his automatic generation.

The main goal of this thesis was to use cURL in PHP to fetch data from cafeteria Facebook page. The basic idea of using curl in PHP is:

step1: Initialize a curl session using curl_init().

step2: Set option for CURLOPT_URL. This value is the URL which is used for request. Setting option for CURLOPT_RETURNTRANSFER to true or 1, will tell curl to return the string instead of printing it out. Author set option for CURLOPT_SSL_VERIFYPEER to false, because author do not need to verify the authenticity of the peer's certificate and set option for CURLOPT_SSL_VERIFYHOST to 0, so connection succeeds regardless of the names in the certificate.

step3: Execute the curl session using curl_exec ().

step4: Close the curl session.

step5: Decode output and escape special characters.

```
//step1
$ch = curl_init();
//step2
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_URL, $graph_url);
//step3
$graph_data = curl_exec($ch);
//step4
curl_close($ch);
//step5
$fb = json_decode($graph_data);
$fb = escape_string($fb);
```

Figure 7 cURL in PHP

5.2 Parsing and processing data from PDF

Since some of the data was in PDF format, there was a need to extract necessary text from PDF document using PdfParser library. PdfParser is a standalone PHP library that provides various tools to extract data from a PDF file. Key features of this library are: extracting metadata (author, description, keywords), extract text from ordered pages, support for compressed pdf, support of charset encoding. [12]

This library is still under active development and as a result, users must expect BC (Backward compatibility) breaks when using the master version. This library requires

PHP 5.3 version. PdfParser will be automatically downloaded through Composer command line, it just needs to add these lines to “composer.json” file,

```
{
  "require": {
    "smalot/pdfparser": "*"
  }
}
```

Figure 8 composer.json file

And ask for composer to download the bundle by running the command:

```
$ composer update smalot/pdfparser.
```

Figure 9 Command for composer to download the bundle

The extraction of text using PdfParser is not so hard, there is a need to create an instance of the Smalot/PdfParser/Parser class and load the PDF file from relative path.

Daily cafeteria provides 2 TalTech places on website:

- “Akadeemia tee 15A, Tallinn (TTÜ IT maja)”
- “Ehitajate tee 5, Tallinn (TTÜ 6 korpus)”

Kolmapäev 2. mai	
Bataadi - köögivilja püreesupp	1.20
<i>Sweet potato - vegetable puree soup</i>	
Kartuli - sealihasupp spinatiga	2.00
<i>Pork soup with spinach</i>	
Lõhefilee sidruniga	5.00
<i>Salmon with lemon - wine sauce</i>	
Ahjus küpsetatud basiiliku ja küüslaugumaitseline sealiha	4.50
<i>Oven-cooked beef with basil and garlic</i>	
Kanafilee kotlet muna ja paprikaga	4.50
<i>Chicken bit cutlets with egg and paprika</i>	
Sealiha - köögiviljahautis kikerhernestega	3.50
<i>Pork - vegetable stew with chick peas</i>	
Kana - hapukoorekaste (päevapraad)	2.30
<i>Chicken sauce</i>	
Kartuli, roheliste ubade ja tomati hautis	2.90
<i>Potato, green pea and tomato stew</i>	
Neljapäev 3. mai	
Herne-püreesupp	1.20
<i>Sweet pea creamy soup</i>	
Kookose ja karrisupp kanaga	2.00
<i>Coconut and curry soup with chicken</i>	
Ahjukala sooja köögiviljasalatiga	5.00
<i>Baked fish with leek and bell pepper</i>	
Praetud kana poolkoivad	4.20
<i>Baked chicken drumsticks</i>	
Tortilla hakklihaga	4.00
<hr/>	
<i>Burritos with minced meat</i>	
Veiseliha - riisiroog köögiviljadega	3.50
<i>Beef - rice meal with vegetables</i>	
Sealihakaste ürtidega (päevapraad)	2.30
<i>Pork sauce with herbs</i>	

Figure 10 Daily cafeteria menu

```

// Parse pdf file and build necessary objects, there $j is PDF
document number.
$parser = new \Smalot\PdfParser\Parser();
$document = $parser->parseFile('lib/document' . $j . '.pdf');
//Retrieve all pages from the pdf
$pages    = $document->getPages();
//Get first page from PDF document
$page = $pages[0];
//Retrieve all text from page
$content = $page->getText();
//Split string by new line
$array = explode("\n", $content);

```

Figure 11 Parse data from PDF document

Here the author defines array of URL's of pdf documents, fetched from Google Chrome Developer Tools, where variable date is used as document publishing date.

```

$host_array = array(

'http://www.daily.ee/files/copy_of_dn_daily_nadalamenuu_ttu_it_maja_(002).pdf?r='.
$date,

'http://www.daily.ee/files/copy_of_dn_daily_nadalamenuu_ttu_6_korpus.pdf?r='. $date

);

```

Figure 12 Array of PDF documents

5.3 Cordova setup

To begin with the Cordova setup, it is needed to install some components:

- NodeJS and NPM
- Android SDK
- Java SDK
- Cordova

After installing components and creating the application, it is needed to install required platforms.

```

$ cordova platform add ios
$ cordova platform add android

```

Figure 13 Adding platforms to project

After all installations, author need to build and run project using these commands:

```
$ cordova build  
$ cordova run android
```

Figure 14 Commands to build and run Cordova project

After successful installation and running our application, it will show following image on Android Emulator or plugged device.

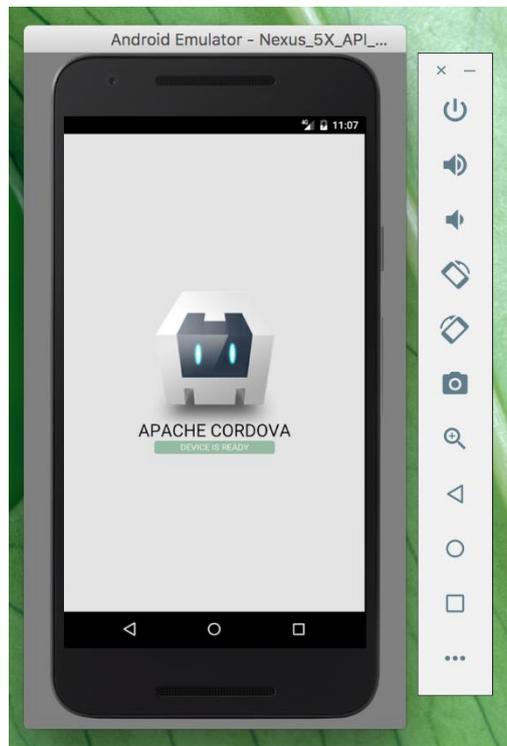


Figure 15 Cordova successful launch

In the package.json file we can look for the platforms, that our application supports, also it shows installed plugins.

```

{
  "name": "eu.fadeyev.dining",
  "version": "1.0.0",
  "displayName": "DiningApp",
  "cordova": {
    "platforms": [
      "browser",
      "ios",
      "android"
    ],
    "plugins": {
      "cordova-plugin-whitelist": {},
      "cordova-plugin-device": {}
    }
  }
}

```

Figure 16 Package.json file

5.4 Mobile application

Using the data from the database, author extracted and converted it to the desired format, separating information about the location, consisting of name, address, image and place coordinates, from the direct information of the menu, price and category.

```

$places[$i] = array(
  'name' => $feeds['name'],
  'address' => $feeds['address'],
  'lat' => $feeds['lat'],
  'lon' => $feeds['lon'],
  'picture' => $feeds['source'],
  'feed' => array()
);

```

Figure 17 Main data about cafeteria

```

$places[($i-1)]['feed'][] = array(
  'est' => trim($feeds['menu_EST'], " \t"),
  'eng' => trim($feeds['menu_ENG'], " \t"),
  'price' => floatval($feeds['price']),
  'category' => $feeds['category'],
  'id' => $feeds['id'],
  'rating' => floatval($feeds['rating']),
  'comments' => $feeds['comments']?:""
);

```

Figure 18 Feeds information

As a result, of this code, author got all data of place and menu and can append it to HTML file.

```
{
  "places": [
    {
      "name": "TTU SOC-maja kohvik",
      "address": "Akadeemia tee 3",
      "lat": "59.3968",
      "lon": "24.6703",
      "picture": "img\\places\\place4.jpg",
      "feed": [
        {
          "est": "Daily offer :Kana-riisiroog",
          "eng": "Chicken-rice dish",
          "price": 2.3,
          "category": "meat",
          "id": "256",
          "rating": "",
          "comments": ""
        }
      ]
    }
  ]
}
```

Figure 19 Feeds response

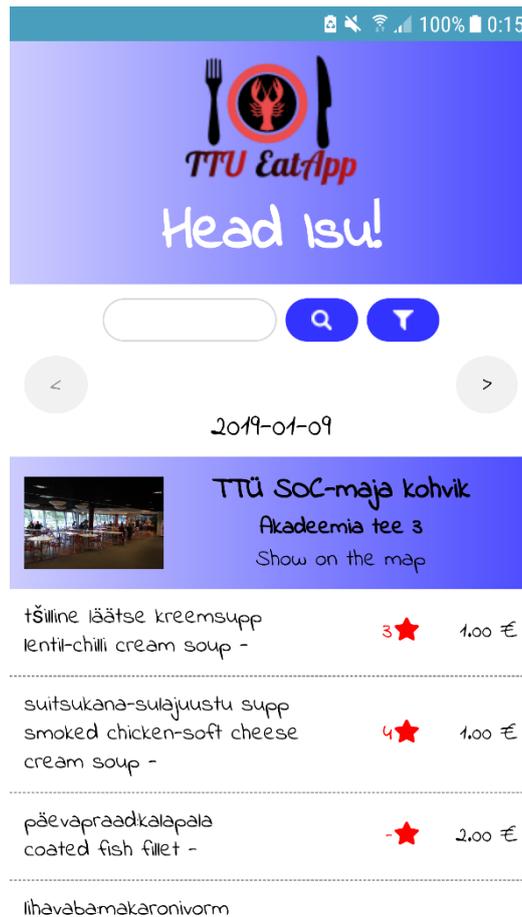


Figure 20 Mobile application main menu

5.4.1 Search

Author use function .search to loop through all feeds to check if there are any text values, that matches the value of the input field.

```
<div class="search">
  <input class="input" name="q" type="text" />
  <button type="button" class="button button-search"><span
class="icon icon-search"></span></button>
</div>
```

Figure 21 HTML for search

```

$feeds = $(p).find('.place-feed-item');
$.each($feeds, function(i, f){
  var $feed = $(f),
  title = $feed.html();
  if (title.toLowerCase().search(
    search.toLowerCase() ) > 0) {
    $place.show();
    $feed.show();
  }
});

```

Figure 22 Implementation of search function

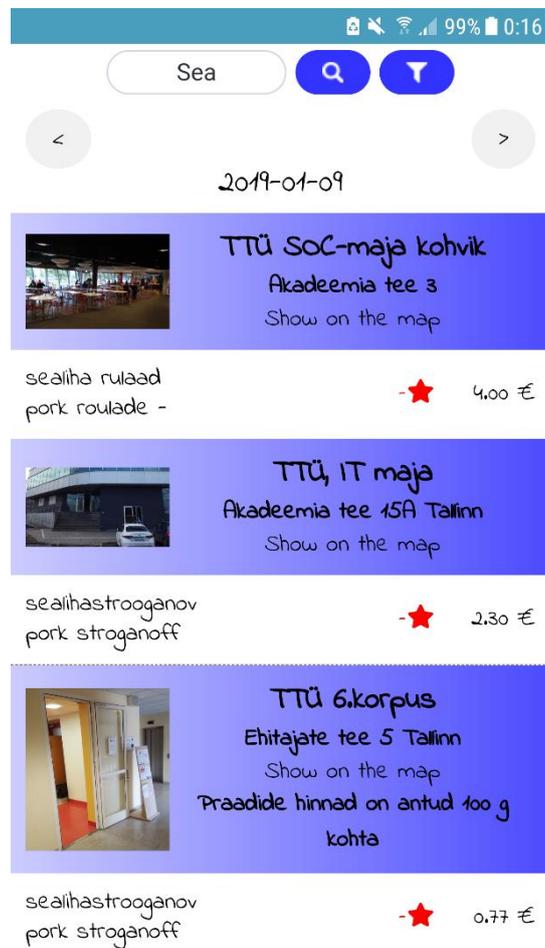


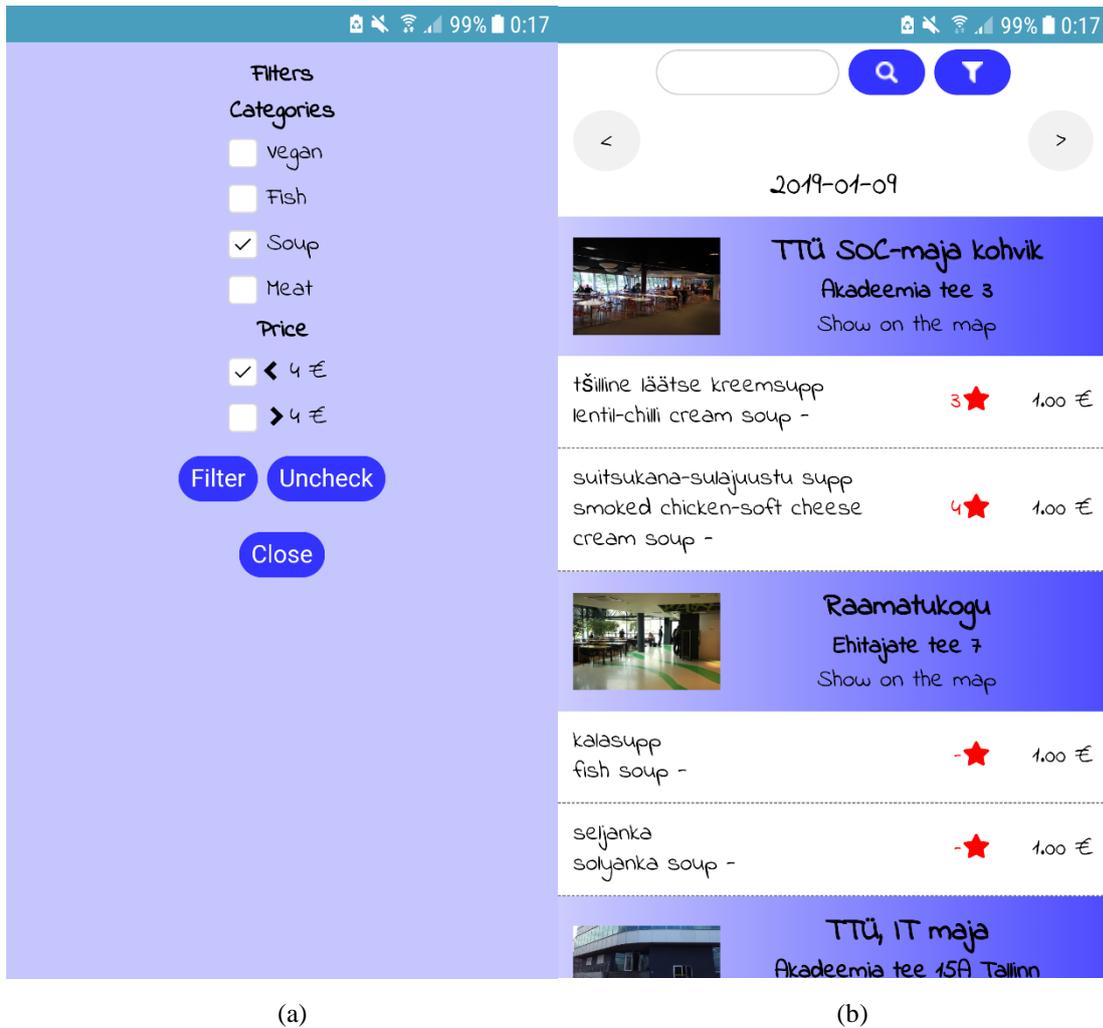
Figure 23 Applying search in application

5.4.2 Filters

Another component of mobile application is search filtering: the ability to allow users to narrow down all menu through a series of filters. As more and more applications specialize in what they offer, and the attempts to place a user in a concrete search

paradigm increase, pure search is becoming less prevalent. The amount of pre-filtering and post-filtering is starting to have greater importance. [13]

As a filter design pattern, was selected “fullscreen filtering”. These filters take up the whole screen, that allows user to show many filters on the display.



(a)

(b)

Figure 24 Applying filter to menu (a), (b) filtered menu

For defining categories, author uses array of key-value pairs, which are created in JSON format, for filtering data. All parsed menu items go through key-value pairs and if menu item match with array value it will get key as a category. If there are no matches between array values and menu items, menu item will be assigned “vegan” category.

```

{
  "meat":
    [
      "karbonaad",
      "veis",
      "pihv",
      ...
    ],
  "fish":
    [
      "lõhe",
      "kala",
      "tursafilee",
      ...
    ],
  "soup":
    [
      "supp",
      "seljanka",
      ...
    ]
}

```

Figure 25 Key-value pairs

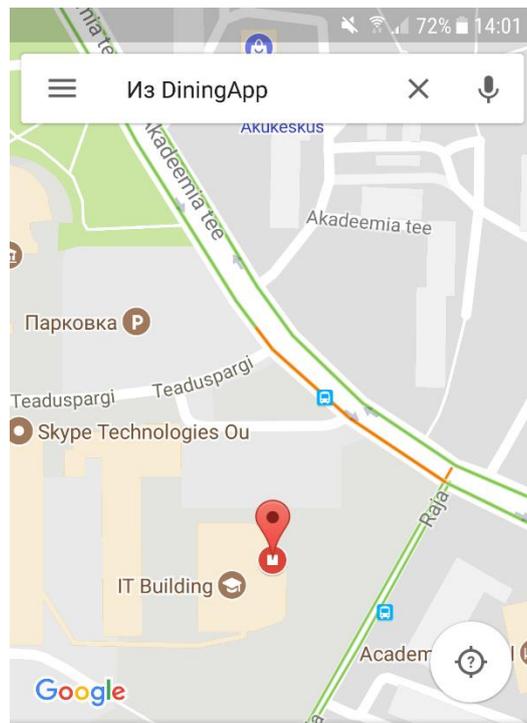
5.4.3 Google maps

For providing users to see cafeteria location, was used geo URI, with the option of opening the geolocation coordinates in the app. This approach allowed the user to use the maps interface, share location with friends, etc. [14]

```

$place_map = $('<a/>', {
  'href' : device.platform == 'iOS' ? 'http://maps.apple.com/?q='+
  geo +'('+ place.name +')' : 'geo:'+ geo +' ?q='+ geo +'('+
  place.name +')',
  'html' : 'Show on the map',
  'data-position' : place.lat + ', ' + place.lon
});

```



TTÜ, IT maja

Из DiningApp

Figure 26 Google maps integration into application

5.4.4 Feedback from user

Every mobile application user can leave a feedback to any menu item. User can leave a rating from 0 to 5 start, according to his opinion, additionally user can leave a comment, so all users can see it.

The rating displayed on the main menu of mobile application is taken as the arithmetic average between ratings from all users for certain menu item.

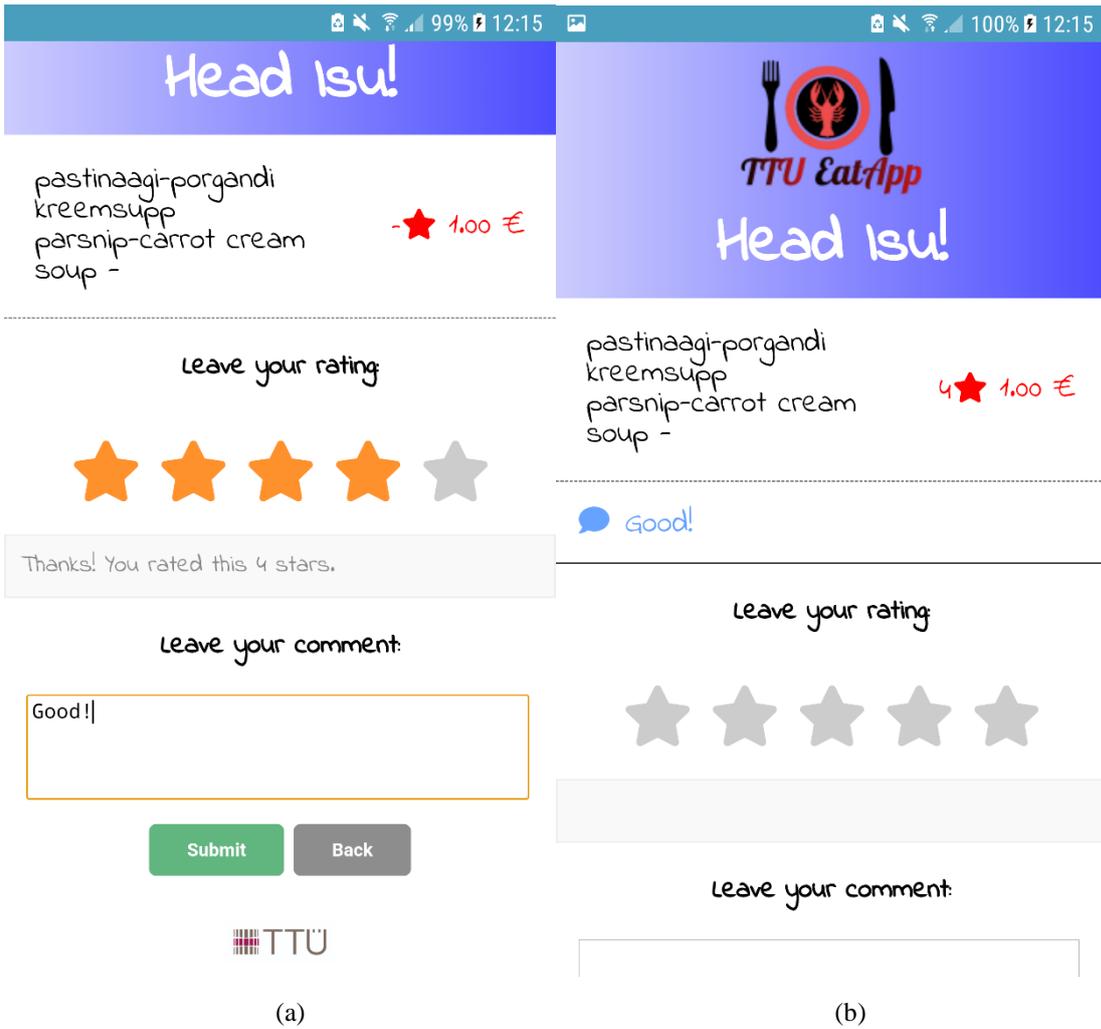


Figure 27 Feedback section before submitting (a), (b) after submitting

6 Summary

The aim of this thesis was to develop a mobile application for Android mobile platform for solving everyday problems of choosing a meal and give possibility to choose proper TalTech cafeteria, using search or any other filters based on meal price, type of meal or dietary preferences.

The author analyzed the types of mobile applications, pros and cons were examined and the hybrid application type for this mobile application was selected because of simple and quick development. After analyzing hybrid applications, author decided to use Apache Cordova mobile development framework, which is free technology and allows to create a mobile application based on standard web technologies, like HTML5, CSS and JavaScript.

The author found existing mobile applications with similar functionality, tested it and took out main application functionalities.

The problem of parsing data from Facebook and websites has been successfully resolved, using Facebook Graph API and PdfParser. The result of parsing was stored in the database and shown in the application itself.

The goal of this work was achieved, and the mobile application was developed and published to Google Play Store.

References

- [1] “Facebook posts page”, [Online]. Available: <https://www.facebook.com/rahvatoitttu/> (15.05.2018)
- [2] “Daily cafeterias list”, [Online]. Available: <http://www.daily.ee/ee/lunch-offers/> (10.05.2018)
- [3] “Web technology for developers: HTML”, [Online] Available: <https://developer.mozilla.org/en-US/docs/Web/HTML> (10.05.2018)
- [4] “Web technology for developers: CSS”, [Online] Available: <https://developer.mozilla.org/en-US/docs/Web/CSS> (10.05.2018)
- [5] “JavaScript Basics”, [Online] Available: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics (10.05.2018)
- [6] Stack Overflow “What is cURL in PHP?”, [Online] Available: <https://stackoverflow.com/questions/3062324/what-is-curl-in-php> (01.05.2018)
- [7] Facebook for developers “API Graph”, [Online] Available: <https://developers.facebook.com/docs/graph-api/overview/> (24.04.2018)
- [8] “Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options”, [Online] Available: https://developer.salesforce.com/page/Native%2C_HTML5%2C_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options (03.01.2019)
- [9] “Native vs. Web vs. Hybrid Apps—What’s the Difference?”, [Online] Available: <https://blog.markgrowth.com/native-vs-web-vs-hybrid-apps-whats-the-difference-1df4c5e4bc50> (25.04.2018)
- [10] “Cordova overview”, [Online] Available: <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>
- [11] “Introduction to PhoneGap Development Benefits”, [Online] Available: <https://www.digitaledge.org/introduction-to-phonegap-development-benefits/>
- [12] “PDF parser”, [Online] Available: <https://pdfparser.org/>
- [13] “6 Best Practices for Mobile App Search Filtering”, [Online] Available: <https://www.raywenderlich.com/155041/6-best-practices-for-mobile-app-search-filtering>
- [14] “Why use geo URI instead of maps”, [Online] Available: <https://blog.bam.tech/developer-news/how-to-add-a-link-to-a-maps-application-in-your-cordova-app>

Appendix 1 – Mobile application in Google Play Store

- Mobile application was uploaded to the Google Play Store with name “TalTech Eat App”, published by Aleksander Fadejev on URL:
<https://play.google.com/store/apps/details?id=eu.fadeyev.dining>