A GetMore Ltd. Mobile Application Innovation

Danil Tkachenko

Bachelor's thesis
2023

Tomas Bata University in Zlín
Faculty of Applied Informatics

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Danil Tkachenko**
Osobní číslo:	**A20448**
Studijní program:	**B0613A140020 Softwarové inženýrství**
Forma studia:	**Prezenční**
Téma práce:	**Inovace mobilní aplikace firmy Getmore s.r.o.**
Téma práce anglicky:	**A GetMore Ltd. Mobile Application Innovation**

Zásady pro vypracování

1. Popište současný stav technologií pro tvorbu mobilních aplikací.
2. Zaměřte se na framework Xamarin a jeho nástupce.
3. Vypracujte návrh rozšíření stávající aplikace o integraci webových řešení společnosti GetMore.
4. Vytvořte klíčové části řešení a tato řešení v práci popište.
5. Zhodnoťte dosažené výsledky a možnosti dalšího rozvoje aplikace.

Forma zpracování bakalářské práce: **tištěná/elektronická**
Jazyk zpracování: **Angličtina**

Seznam doporučené literatury:

1. Xamarin documentation [online]. Dostupné z: https://learn.microsoft.com/en-us/xamarin/.
2. Microsoft.NET documentation [online]. Dostupné z: https://learn.microsoft.com/dotnet/.
3. HERMES, Dan a Nima MAZLOUMI. Building Xamarin.Forms Mobile Apps Using XAML: Mobile Cross-Platform XAML and Xamarin.Forms Fundamentals. California: Apress, 2019. ISBN 978-1-4842-4029-8.
4. SKEET, Jon. C# in depth. Fourth edition. Shelter Island, NY: Manning, [2019]. ISBN 978-1-61729-453-2.
5. MARTIN, Robert C. *Clean Agile: back to basics*. Boston: Pearson, [2020]. Robert C. Martin series. ISBN 978-0-13--578186-9.

Vedoucí bakalářské práce: **Ing. Erik Král, Ph.D.**
Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: **2. prosince 2022**
Termín odevzdání bakalářské práce: **26. května 2023**

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.
①

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 7. prosince 2022

I hereby declare that:

- I understand that by submitting my Bachelor´s Thesis, I agree to the publication of my work according to Law No. 111/1998, Coll., On Universities and on changes and amendments to other acts (e.g. the Universities Act), as amended by subsequent legislation, without regard to the results of the defence of the thesis.
- I understand that my Bachelor´s Thesis will be stored electronically in the university information system and be made available for on-site inspection, and that a copy of the Bachelor´s Thesis will be stored in the Reference Library of the Faculty of Applied Informatics, Tomas Bata University in Zlín, and that a copy shall be deposited with my Supervisor.
- I am aware of the fact that my Bachelor´s Thesis is fully covered by Act No. 121/2000 Coll. On Copyright, and Rights Related to Copyright, as amended by some other laws (e.g. the Copyright Act), as amended by subsequent legislation; and especially, by §35, Para. 3.
- I understand that, according to §60, Para. 1 of the Copyright Act, TBU in Zlín has the right to conclude licensing agreements relating to the use of scholastic work within the full extent of §12, Para. 4, of the Copyright Act.
- I understand that, according to §60, Para. 2, and Para. 3, of the Copyright Act, I may use my work - Bachelor´s Thesis, or grant a license for its use, only if permitted by the licensing agreement concluded between myself and Tomas Bata University in Zlín with a view to the fact that Tomas Bata University in Zlín must be compensated for any reasonable contribution to covering such expenses/costs as invested by them in the creation of the thesis (up until the full actual amount) shall also be a subject of this licensing agreement.
- I understand that, should the elaboration of the Bachelor´s Thesis include the use of software provided by Tomas Bata University in Zlín or other such entities strictly for study and research purposes (i.e. only for non-commercial use), the results of my Bachelor´s Thesis cannot be used for commercial purposes.
- I understand that, if the output of my Bachelor´s Thesis is any software product(s), this/these shall equally be considered as part of the thesis, as well as any source codes, or files from which the project is composed. Not submitting any part of this/these component(s) may be a reason for the non-defence of my thesis.

I herewith declare that:

- I have worked on my thesis alone and duly cited any literature I have used. In the case of the publication of the results of my thesis, I shall be listed as co-author.
- That the submitted version of the thesis and its electronic version uploaded to IS/STAG are both identical.

In Zlín; dated: Danil Tkachenko, v.r.

Student´s Signature

ABSTRAKT

Bakalářská práce se zaměřuje na analýzu současných technologií pro tvorbu mobilních aplikací, porovnává nejpoužívanější platformy a jazyky a zkoumá trendy a budoucí vývoj v oblasti mobilních aplikací. Důraz je kladen na framework Xamarin a jeho nástupce, jejich historii, vývoj a klíčové vlastnosti. V práci je provedeno porovnání Xamarinu s jeho nástupci a vybrán Xamarin framework pro rozšíření aplikace GetMore. Následně je navrženo rozšíření stávající GetMore aplikace o integraci webových řešení společnosti, s analýzou potřeb a požadavků společnosti, představením konceptu integrace a návrhem architektury a rozhraní pro implementaci navrhovaného rozšíření. V rámci implementace klíčových částí řešení je popsán vývoj těchto částí pomocí zvoleného frameworku, integrace webových řešení do stávající aplikace a testování a ladění implementovaných funkcí. Závěr práce se věnuje hodnocení dosažených výsledků a možností dalšího rozvoje aplikace. Je zhodnocena úspěšnost integrace webových řešení do aplikace a diskutována výkonnost, stabilita a uživatelská přívětivost aplikace po rozšíření. Na základě těchto zjištění jsou navrženy možnosti dalšího rozvoje a optimalizace aplikace GetMore.

Klíčová slova: Xamarin, Getmore, iOS, Android

ABSTRACT

The bachelor thesis focuses on the analysis of current technologies for mobile application development, compares the most used platforms and languages and examines trends and future developments in the field of mobile applications. The focus is on the Xamarin framework and its successors, their history, development, and key features. Bachelor's work compares Xamarin with its successors and selects the Xamarin framework to extend the GetMore application. Subsequently, an extension of the existing GetMore application is proposed to integrate the company's web-based solutions, with an analysis of the company's needs and requirements, a pre-presentation of the integration concept, and a design of the architecture and interface for implementing the proposed extension.The implementation of key parts of the solution describes the development of these parts using the chosen framework), the integration of the web solutions into the existing application and the testing and debugging of the implemented features.The thesis concludes with an evaluation of the achieved results and possibilities for further development of the application. The success of integrating web solutions into the application is evaluated and the performance, stability, and user-friendliness of the application after the extension is discussed. Based on these findings, the possibilities of further development and optimization of the GetMore application are proposed.

Keywords: Xamarin, Getmore, iOS, Android

ACKNOWLEDGEMENTS

Acknowledgements, motto and a declaration of honour saying that the print version of the Bachelor's/Master's thesis and the electronic version of the thesis deposited in the IS/STAG system are identical, worded as follows:

I hereby declare that the print version of my Bachelor's/Master's thesis and the electronic version of my thesis deposited in the IS/STAG system are identical.

CONTENTS

INTRODUCTION **10**

I THEORETICAL PART. **11**

1 MOBILE APP DEVELOPMENT TECHNOLOGIES. **12**

1.1 JAVA MOBILE DEVELOPMENT 12

1.2 SWIFT MOBILE DEVELOPMENT 14

1.3 C# MOBILE DEVELOPMENT 16

1.3.1 C# Programming 12

1.3.2 Xamarin 17

1.3.3 How Xamarin works 18

1.3.4 Xamarin.Android 19

1.3.5 Xamarin.iOS 19

1.3.6 MAUI 20

2 IMPORTANT APPLICATION CONTROL. **22**

2.1 XAMARIN.FORMS WEBVIEW CONTROL 22

3 INTEGRATED DEVELOPMENT ENVIRONMENT. **24**

3.1 VISUAL STUDIO 24

II PRACTICAL PART. **26**

4 EXTENSION OF THE EXISTING APPLICATION TO INTEGRATE GETMORE'S WEB SOLUTIONS **27**

4.1 COMPANY GETMORE 27

4.1.1 Getmore Projects 28

4.2 EMBRACING AGILE METHODOLOGIES AT GETMORE 33

4.3 INITIAL STATE GMS MOBILE 35

4.4 GMS MOBILE EXTENSION 44

4.5 OUTLINE DESCRIPTION OF THE MAIN FUNCTIONS GMS MOBILE 47

5 GMS MOBILE APP **49**

5.1 APPLICATION DESIGN 49

5.2 APPLICATION STRUCTURE 56

5.2.1 ApiWrapper for authentication on Async 58

5.2.2 GmsInterfaces 59

5.2.3 Language mutations 60

5.2.4 Notifications 62

6 ACHIEVED RESULTS AND FURTHER DEVELOPMENT **66**

6.1 RESULTS 66

6.2 MIGRATION FROM XAMARIN TO .NET MAUI: A THESIS OVERVIEW 66

REFERENCES **67**

CONCLUSION **68**

LIST OF ABBREVIATIONS **70**

LIST OF FIGURES **72**

INTRODUCTION

The widespread use of smartphones and mobile devices has led to a surge in demand for mobile applications, making it essential for businesses to develop applications that cater to their customers' needs. As a result, mobile application development has become a crucial aspect of modern software engineering, with various platforms and frameworks available to facilitate the process.This bachelor's thesis focuses on the innovation of the GetMore s.r.o. mobile application, specifically in integrating web solutions and utilizing the Xamarin framework and its successors for development purposes. Xamarin is a popular framework for creating cross-platform mobile applications, allowing developers to share code across different platforms, reducing development time and resources.

I. THEORY

1 MOBILE APP DEVELOPMENT TECHNOLOGIES.

The process of creating software for use on mobile devices like smartphones, tablets is termed mobile application development. Typically, this involves designing applications for commonly used operating systems, including Android and iOS. These applications can either come pre-installed on devices, be downloaded from app stores, or be accessed through mobile web browsers.

Mobile app development has experienced rapid growth and now spans several industries such as retail, telecommunications, e-commerce, insurance, and government. The main objective is to satisfy user needs for quick and convenient access to information and transactional capabilities. Today, the primary avenue for both individuals and businesses to access the internet is through mobile devices and their respective applications.

To remain competitive and successful, organizations need to develop mobile applications that cater specifically to the needs of their customers, business partners, and employees. Even though the process of mobile application development may appear challenging, it becomes more navigable when certain crucial guidelines and best practices are adhered to.

After deciding on the most appropriate operating system(s), developers need to take into account the constraints of mobile devices while ensuring their applications can effectively circumnavigate potential distribution obstacles.[1]

1.1 Java mobile development

When it comes to developing mobile applications using Java, the primary framework utilized is the Android SDK (Software Development Kit). The Android SDK comprises a collection of tools, libraries, and documentation necessary for constructing, evaluating, and debugging Android apps. Integrated with Android Studio, the official Integrated Development Environment (IDE) for building Android applications, the Android SDK offers numerous features, such as code editing, debugging, and testing tools, which facilitate Java-based Android app development. By leveraging the Android SDK, developers can create native mobile applications in Java that are tailored to operate efficiently on a wide range of Android devices. [2]

In addition to the basic functionalities offered by the Android SDK, it also provides access to a variety of APIs (Application Programming Interfaces) which developers can use to incorporate diverse features into their apps. For instance, it includes APIs for integrating device hardware functionalities like camera, accelerometer, or GPS, as well as APIs for Google services such as Google Maps or Google Cloud Messaging.

The Android SDK is also deeply tied with the Android Emulator, a virtual device that mimics the behavior and interface of a real Android device. This allows developers to test their applications in a safe, controlled environment before deploying them onto physical devices. Moreover, Android Studio supports instant app deployment, meaning developers can observe the impact of changes in their code in real time.

One of the significant advantages of using Java for Android app development is its robustness and maturity as a programming language. Java has a broad and well-established developer community, providing ample resources, tutorials, and libraries for troubleshooting and learning. This vast ecosystem can be extremely beneficial for both beginners and experienced developers.

Finally, the Android SDK is regularly updated by Google, ensuring it stays relevant and efficient for modern application development. These updates often include performance enhancements, new features, bug fixes, and improvements to the IDE and other tools, allowing developers to stay up-to-date with the latest trends in Android app development. All of these aspects make the Android SDK a highly effective tool for creating sophisticated, user-friendly applications in Java for the Android platform.

This sample of code shows code that after pressing the button will display text "Hello, Android SDK!" on the screen.

```
@Override
  protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      button = findViewById(R.id.button);
      textView = findViewById(R.id.textView);

      button.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View view) {
              textView.setText("Hello, Android SDK!");
          }
      });
  }
```

1.2 Swift mobile development

Swift is an advanced, general-purpose, multi-paradigm, compiled programming language created by Apple Inc. and the open-source community. Introduced in 2014, Swift was designed to replace Apple's earlier programming language, Objective-C, which had remained largely unaltered since the early 1980s and lacked contemporary language features. Swift is compatible with Apple's Cocoa and Cocoa Touch frameworks, and one of its primary design goals was to achieve seamless interoperability with the extensive existing Objective-C codebase for Apple products. Developed using the open-source LLVM compiler framework, Swift has been incorporated into Xcode since its sixth version, released in 2014. On Apple platforms, Swift utilizes the Objective-C runtime library, enabling C, Objective-C, C++, and Swift code to operate within a single program. [3]

Swift has quickly gained popularity due to its modern features, simplicity, and safety compared to Objective-C. With automatic memory management via the ARC (Automatic Reference Counting), Swift eliminates common programming errors such as null pointer dereferencing and memory leaks. It also provides support for functional programming patterns, which leads to more readable and maintainable code.

Swift is well-suited for developing apps on Apple's suite of platforms, including iOS, macOS, watchOS, and tvOS. This wide platform coverage allows developers to write code that can be shared across multiple devices and platforms, reducing both development time and costs. Swift also offers the advantage of faster performance due to its compile-time optimization, leading to more efficient apps.

One of Swift's standout features is its Playgrounds functionality in Xcode, which enables developers to test out a snippet of code without having to run an entire app. This interactive coding environment fosters a learn-as-you-go approach, making Swift more accessible and easier to learn for beginners.

Apple's continuous support and enhancement of Swift, alongside its vibrant open-source community, ensure the language keeps evolving and stays up-to-date with modern programming practices. The community also contributes to a growing library of Swift packages, which can be integrated into any Swift application to provide extra functionality without having to write those features from scratch.

Finally, Swift supports a feature known as "Protocol-Oriented Programming", which emphasizes the use of protocols and protocol extensions over class inheritance, allowing for a more flexible and compositional approach to structuring code. This, alongside other modern features, demonstrates Swift's commitment to facilitating high-quality, efficient, and robust software development on Apple's platforms.

This sample of code shows variable manipulation , and a for loop that will write out all the values from the people array.

```swift
// Variables
var implicitInteger = 70
var implicitDouble = 70.0
var explicitDouble: Double = 70

// Constants
let speedOfLight = 299_792_458
let numberOfApples = 345
let appleSummary = "I have \(numberOfApples) apples."

print("Hello, world")

let people = ["Anna": 67, "Beto": 8, "Jack": 33, "Sam": 25]
for (name, age) in people {
    print("\(name) is \(age) years old.")
}
```

1.3 C# mobile development

For developing mobile applications in C#, the main framework are Xamarin or Maui.

1.3.1 C# programming

C# (articulated as "C-sharp") is a versatile, object-oriented programming (OOP) language conceived by Microsoft Corporation. Launched in 2000 as an integral part of the .NET project, C# has matured into a leading programming language globally, finding utility in a multitude of software development sectors such as web and desktop applications, gaming, and in recent times, mobile application development.

C# was fundamentally engineered to augment productivity during software development. This attribute is clearly observable in its syntax – highly articulate yet uncomplicated, thus making it an appealing choice for novices and seasoned programmers alike.

C# is tightly woven with the .NET framework, a software ecosystem also conceived by Microsoft. This ecosystem furnishes a regulated programming setting where software can be designed, implemented, and executed on platforms operating on Windows. This integration permits developers to architect a diverse array of robust and secure applications, including web-based applications, services, and client-server applications.

A notable attribute of C# is its robust type system, which substantially curtails runtime errors. Coupled with garbage collection and exception handling mechanisms, C# stands out as a remarkably secure programming language. Furthermore, its object-oriented design fosters encapsulation, inheritance, and polymorphism, fundamental principles that empower developers to architect scalable and maintainable code structures.

The debut of .NET Core, a cross-platform iteration of .NET, has broadened C#'s scope beyond a predominantly Windows-focused language. Presently, C# developers have the ability to architect applications that are operable on a multitude of platforms including macOS, Linux, among others.

Lastly, the future trajectory of C# appears optimistic, courtesy of Microsoft's pledge to its consistent enhancement. With each ensuing version, Microsoft introduces novel features, polishes the syntax, and boosts performance, thereby ensuring C# stays contemporary and pertinent amidst the swiftly evolving landscape of software development.

In summation, the adaptability, robustness, and user-friendliness of C# establish it as a pivotal entity in the realm of programming. Its extensive use, from web and desktop applications to mobile apps and game development, underscores its critical function in the software development sector. These attributes render C# as not only an indispensable instrument for modern software development but also an engrossing topic of exploration in the field of computer science.[4]

1.3.2 Xamarin

Xamarin is a cross-platform mobile development framework that allows developers to build applications for iOS, Android, and Windows platforms using C# and the .NET framework. Xamarin was acquired by Microsoft in 2016, and since then, it has become an essential part of the Microsoft development ecosystem. Xamarin uses a single shared codebase to build native user interfaces and access device-specific APIs, reducing the need to write separate code for each platform. This approach significantly reduces development time and effort, making it an attractive choice for mobile app development. Some key features of Xamarin include: [5]

1. Shared Codebase: Xamarin allows developers to share up to 90% of their code across different platforms, making it easier to maintain and update the app in the long run.
2. Native Performance: Xamarin compiles C# code into native code for each platform, ensuring high performance and a native look and feel for the app. Xamarin.Forms:
3. Xamarin.Forms is a UI toolkit that allows developers to create a single, shared user interface for their app across all platforms. This further reduces the amount of platform-specific code needed.

4. Access to Platform-Specific APIs: Xamarin provides bindings to native APIs, which means developers can still access platform-specific features and capabilities when needed.
5. Visual Studio Integration: Xamarin is fully integrated with Visual Studio, Microsoft's flagship IDE, making it easy to develop, debug, and deploy Xamarin apps.

Large Developer Community: Xamarin has a vast and active community of developers, which means there is a wealth of resources, tutorials, and support available online.

1.3.3 How Xamarin works

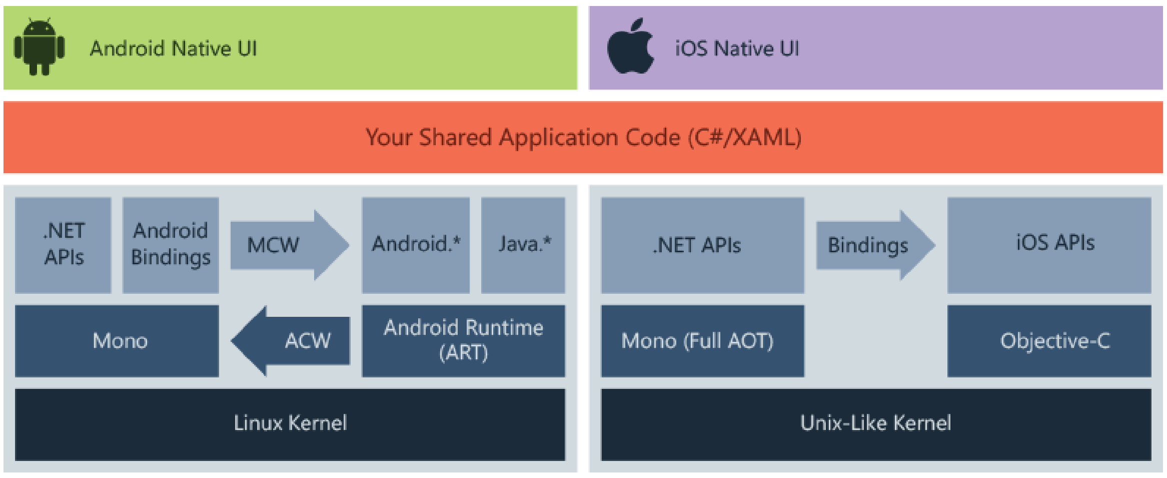


Fig. 1.1: Architecture of a cross-platform Xamarin application [6]

The illustration depicts the comprehensive structure of a cross-platform Xamarin application. Xamarin enables developers to design native user interfaces on each platform while writing shared C# business logic. Generally, Xamarin allows for about 80% of the application code to be shareable across platforms.

Xamarin is built upon the .NET framework, which automatically manages tasks like memory allocation, garbage collection, and seamless interaction with the underlying platforms. [7]

1.3.4 Xamarin.Android

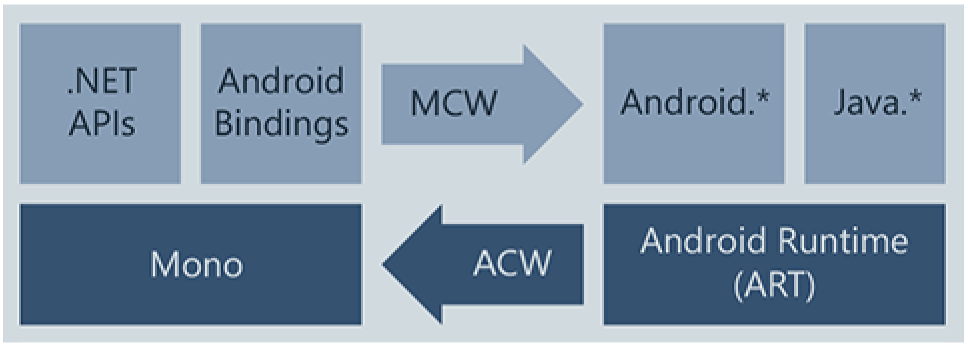


Fig. 1.2: Android architecture of Xamarin application [8]

Xamarin.Android apps are compiled from C# into Intermediate Language (IL), which is then Just-in-Time (JIT) compiled into native assembly upon application launch. Xamarin.Android applications operate within the Mono execution environment, alongside the Android Runtime (ART) virtual machine. Xamarin supplies .NET bindings for Android.* and Java.* namespaces, with the Mono environment calling these namespaces through Managed Callable Wrappers (MCW). Android Callable Wrappers (ACW) are provided to the ART, enabling both environments to execute code within each other.

1.3.5 Xamarin.iOS

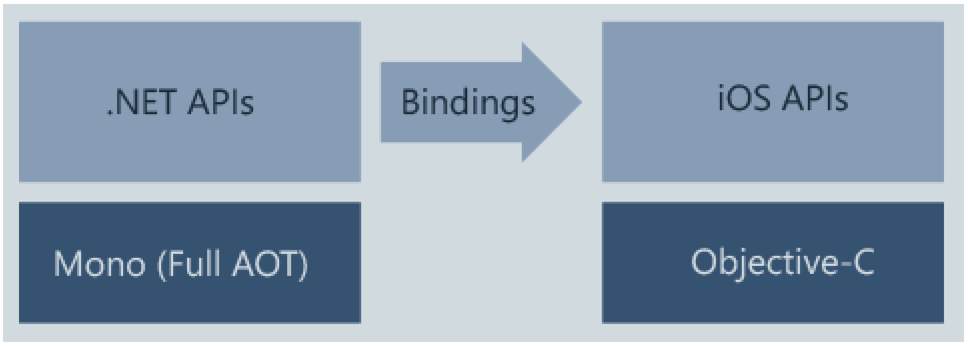


Fig. 1.3: Ios architecture of Xamarin application [9]

Xamarin.iOS apps are entirely Ahead-of-Time (AOT) compiled from C# into native ARM assembly code. Xamarin employs Selectors to make Objective-C accessible to managed C# and Registrars to expose managed C# code to Objective-C. Collectively referred to as "bindings," Selectors and Registrars enable communication between Objective-C and C#.

1.3.6 MAUI

.NET MAUI (Multi-platform App UI) is a cross-platform framework for building native UIs (User Interfaces) for Windows, macOS, iOS, and Android devices. It is an evolution of the Xamarin.Forms framework, which is part of the .NET ecosystem. With the introduction of .NET MAUI, developers can build applications for various platforms using a single code-base, making it easier to maintain and deploy apps.

.NET MAUI works by providing a consistent API (Application Programming Interface) for creating user interfaces across different platforms. It uses the MVVM (Model-View-View-Model) design pattern and supports data binding, which allows developers to create a re-sponsive and testable user interface.

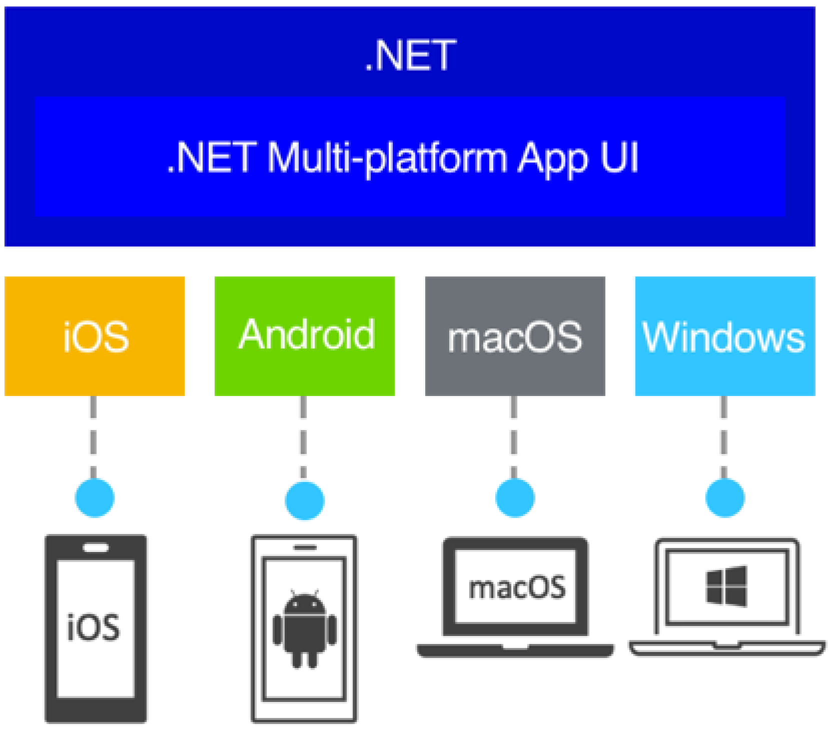


Fig. 1.4: Maui overview [10]

The key features of .NET MAUI include: [11]

- Single codebase:
  Develop applications for multiple platforms using a single codebase, reducing duplication of efforts and maintenance.

- Consistent API:
  Provides a consistent API for creating user interfaces across different platforms, making it easier for developers to adapt to different platforms.
- Native performance:
  Leverages platform-specific capabilities to provide native performance on each platform.
- Extensibility:
  Offers a rich set of controls and components that can be extended and customized as needed.
- UI Controls:
  .NET MAUI includes a wide range of built-in UI controls, such as buttons, labels, text inputs, and more. These controls are designed to work seamlessly across different platforms, adapting to the native UI of each platform. You can also create custom controls or use third-party libraries to extend the available UI components.
- Layouts:
  .NET MAUI provides various layout options to help you design responsive and adaptable user interfaces. Some of the popular layout types include StackLayout, Grid, and FlexLayout, which allow you to arrange UI elements in different ways to suit the needs of your application.
- Styling:
  .NET MAUI can be styled by using CSS (Cascading Style Sheets) or C# code. This allows you to easily customize the appearance of app, and apply consistent styles across different platforms.
- Data Binding:
  .NET MAUI supports data binding, which allows you to connect UI elements to your data models and automatically update the UI when the underlying data changes. This feature simplifies the process of managing the state of your application and helps you build a responsive and dynamic user interface.

2 IMPORTANT APPLICATION CONTROL

2.1 Xamarin.Forms WebView control

The application relies heavily on the WebView component, rendering it essential to comprehend its functionalities.

WebView is a control in the Xamarin.Forms library that allows developers to display web content within a mobile application. It works by embedding a native web view component into the Xamarin.Forms app, providing a way to display HTML content from a URL or a string directly in the app. This enables developers to integrate web pages or web-based features seamlessly into their cross-platform mobile applications.[12]

Key features and capabilities of Xamarin.Forms WebView include:

- Cross-platform compatibility:
  Xamarin.Forms WebView is designed to work across Android, iOS, and the Universal Windows Platform (UWP). This means that developers can use a single WebView control to display web content on various devices and platforms with minimal platform-specific code.
- Navigating to URLs:
  WebView can load and display web content from a URL. This is useful for loading external web pages or local HTML files bundled with the app.
- Displaying HTML strings
  In addition to loading content from URLs, WebView can also display HTML content directly from a string. This enables developers to generate dynamic HTML content within the app and display it in the WebView.
- Navigation events:
  Xamarin.Forms WebView provides events such as Navigating, Navigated, and NavigationFailed, which allow developers to respond to navigation actions, like starting navigation, completing navigation, or handling errors.

- Executing JavaScript:
  WebView supports executing JavaScript code within the displayed web content. This can be useful for interacting with the web content, manipulating the DOM, or communicating between the app and the WebView.
- Controlling navigation:
  Developers can control the WebView's navigation behavior, such as intercepting navigation requests, deciding whether to allow or block certain URLs, or implementing custom logic for handling user interactions with the web content.

3 INTEGRATED DEVELOPMENT ENVIRONMENTS

3.1 Visual studio

Visual Studio, a product of Microsoft, is a well-established Integrated Development Environment (IDE) that has been instrumental for developers worldwide since its launch in 1997. It provides a suite of powerful capabilities and is compatible with an array of programming languages such as C#, C++, Python, JavaScript, and more.

The principal role of Visual Studio is to offer a unified platform for coding, testing, debugging, and deploying applications. Its intuitive interface, along with its smart code prediction tool (IntelliSense), significantly boosts productivity, assisting both beginners and skilled developers in crafting more effective and streamlined code.

A distinguishing feature of Visual Studio is its robust debugging utilities. These facilities help detect and resolve code errors, substantially reducing the time and effort expended in rectifying bugs. Furthermore, the runtime debugging offered by Visual Studio provides developers insights into and rectification of issues that could arise while the application is in operation.

Visual Studio also stands out for its emphasis on collaboration and team work. With integration of Version Control Systems like Git, developers can collaboratively work on the same project while efficiently managing code versions and tracking changes. This functionality proves invaluable in large-scale projects that involve multiple teams and developers.

With Visual Studio, developers are enabled to create applications for various platforms, including Windows, Android, iOS, and the Web. The advent of .NET Core has also ushered in the possibility for developers to use Visual Studio for building cross-platform applications, thereby broadening the scope of their software solutions.

Lastly, Visual Studio seamlessly integrates with Microsoft's cloud service, Azure. This association enables developers to smoothly develop, test, and deploy applications based in the cloud, facilitating a more streamlined workflow in the era dominated by cloud computing.

To sum up, the wide-ranging functionalities of Visual Studio render it an irreplaceable asset in software development. With its support for numerous programming languages, powerful debugging mechanisms, collaboration features, and adaptability to diverse project requirements, Visual Studio stands as a pivotal tool in the world of software development. Thus, it presents a rewarding and engaging area of exploration for scholars in Computer Science and Software Engineering.[13]

II. PRACTICAL PART

4 EXTENSION OF THE EXISTING APPLICATION TO INTEGRATE GETMORE'S WEB SOLUTIONS

4.1 Company Getmore

GETMORE, s.r.o. It is a Czech company that specializes in providing services in Internet marketing, web development, graphic design and other related areas. The main purpose of the company is to help its clients increase their business presence on the Internet in order to attract more customers and increase sales. [14]

In Figure 4.1 we see main site page

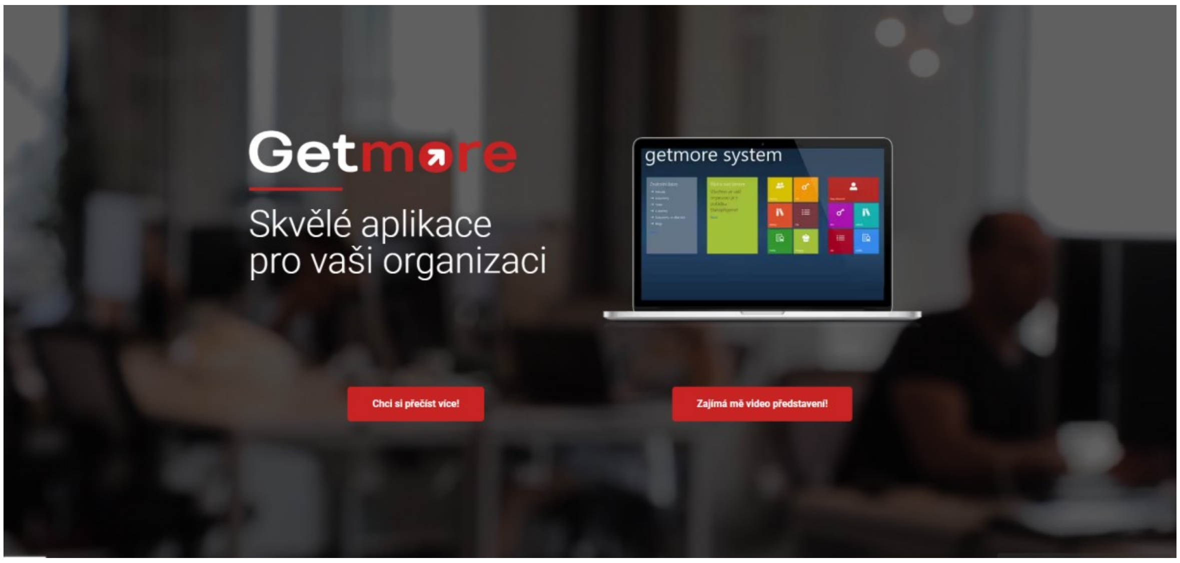

Getmore
Skvělé aplikace pro vaši organizaci
getmore system
Chci si přečíst více!
Zajímá mě video představení!

Fig. 4.1: Website with the company's offerings [15]

Among the services provided by GETMORE, s.r.o. are:

1. Creation and development of websites and online stores.
2. Development and implementation of mobile applications.
3. Creation of graphic design, logos, branding and corporate identity.
4. Contextual advertising management and promotion in social networks.
5. Search Engine Optimization (SEO) and content marketing.
6. Video marketing and video content creation.
7. The company's staff assists clients in developing an effective marketing and web development strategy to achieve their goals and increase profits.

4.1.1 Getmore Projects

Intranet website

This site [16] is office software - for preparing and successfully managing and planning work and team projects, splitting the project into sprints, allocating team and individual capabilities, clear project task lists, automatic notifications.

In Figure 4.1.1 we see main site page

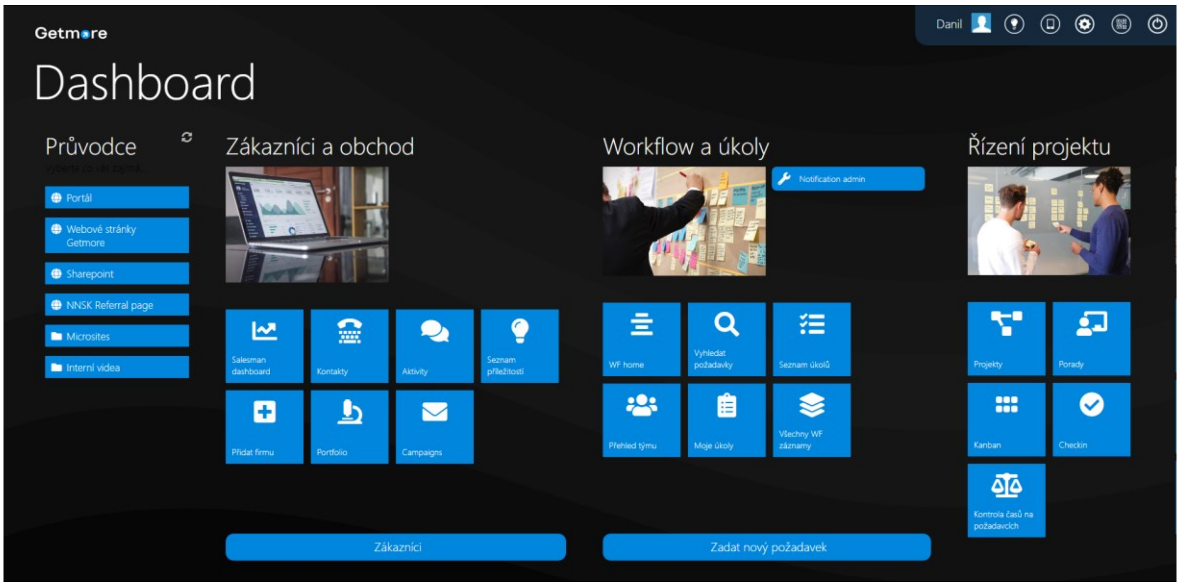
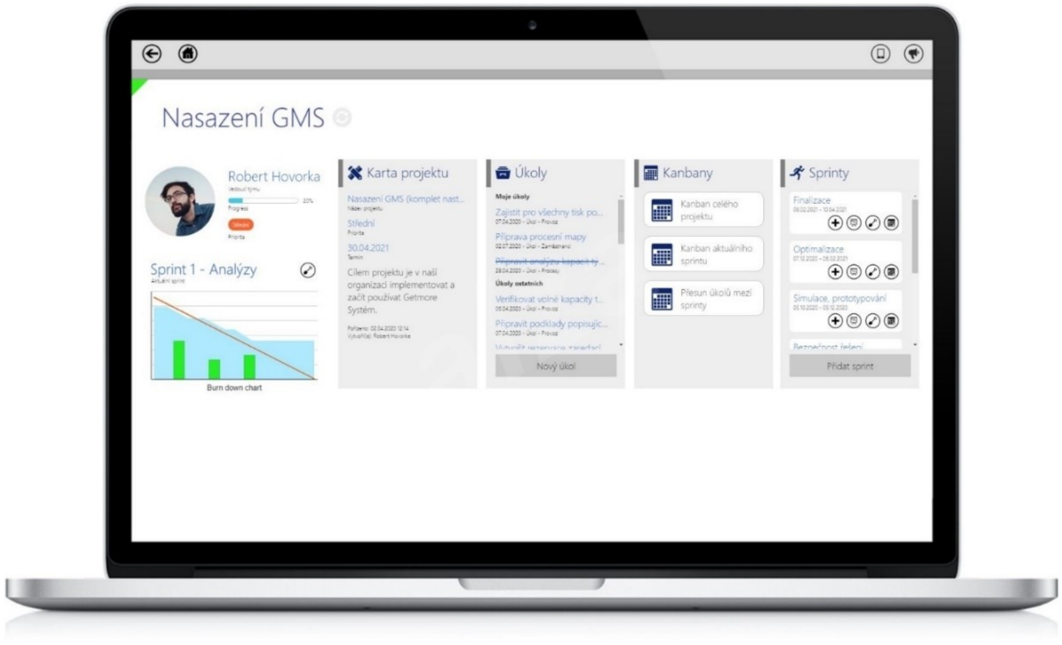


Fig. 4.1.1: *Main Page* getmore.getmore.cz [16]

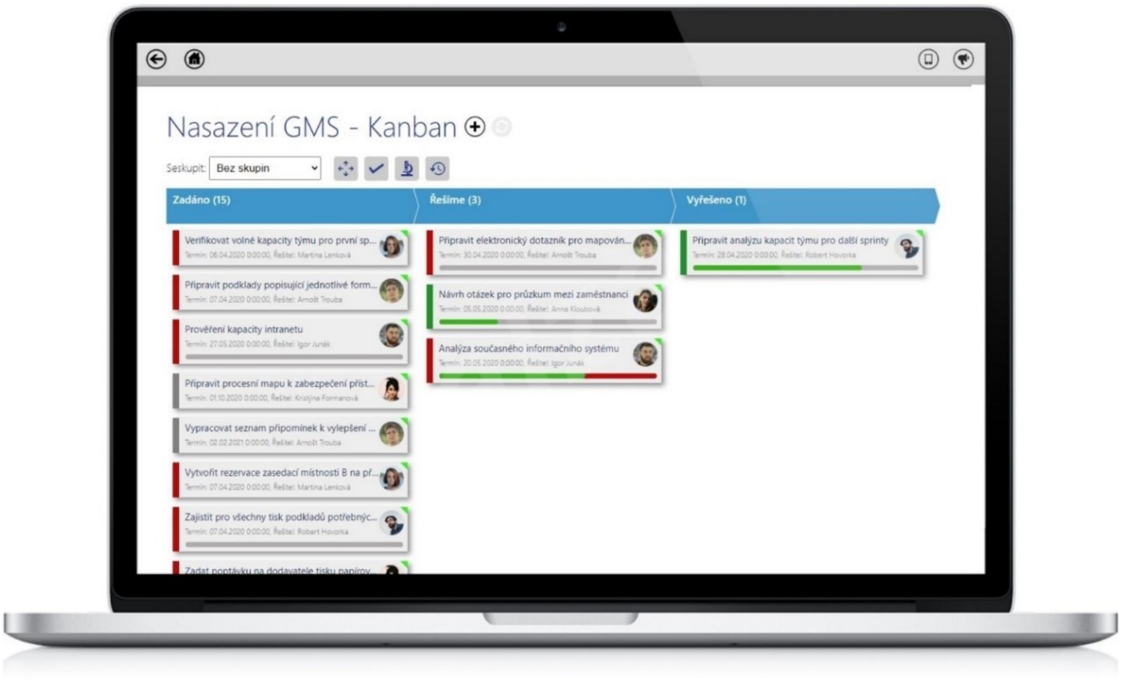
In Figure 4.1.2 we see an example of the profile that each user has on the site.



Cílem projektu je v naší organizaci implementovat a začít používat Getmore Systém.

Fig. 4.1.2: *Profile Page* getmore.getmore.cz [17]

In Figure 4.1.3 we see the kanban page.The page shows the success of workers on completed tasks


Nasazení GMS - Kanban

Seskupit: Bez skupin

Zadáno (15)

- Verifikovat volné kapacity týmu pro první sp... Termín: 06.04.2020 0:00:00, Řešitel: Martina Lenková
- Připravit podklady popisující jednotlivé form... Termín: 07.04.2020 0:00:00, Řešitel: Arnošt Trouba
- Prověření kapacity intranetu Termín: 27.05.2020 0:00:00, Řešitel: Igor Junák
- Připravit procesní mapu k zabezpečení příst... Termín: 01.10.2020 0:00:00, Řešitel: Kristýna Formanová
- Vypracovat seznam připomínek k vylepšení ... Termín: 02.02.2021 0:00:00, Řešitel: Arnošt Trouba
- Vytvořit rezervace zasedací místnosti B na př... Termín: 07.04.2020 0:00:00, Řešitel: Martina Lenková
- Zajistit pro všechny tisk podkladů potřebnýc... Termín: 07.04.2020 0:00:00, Řešitel: Robert Hovorka
- Zadat poptávku na dodavatele tisku papírov...

Řešíme (3)

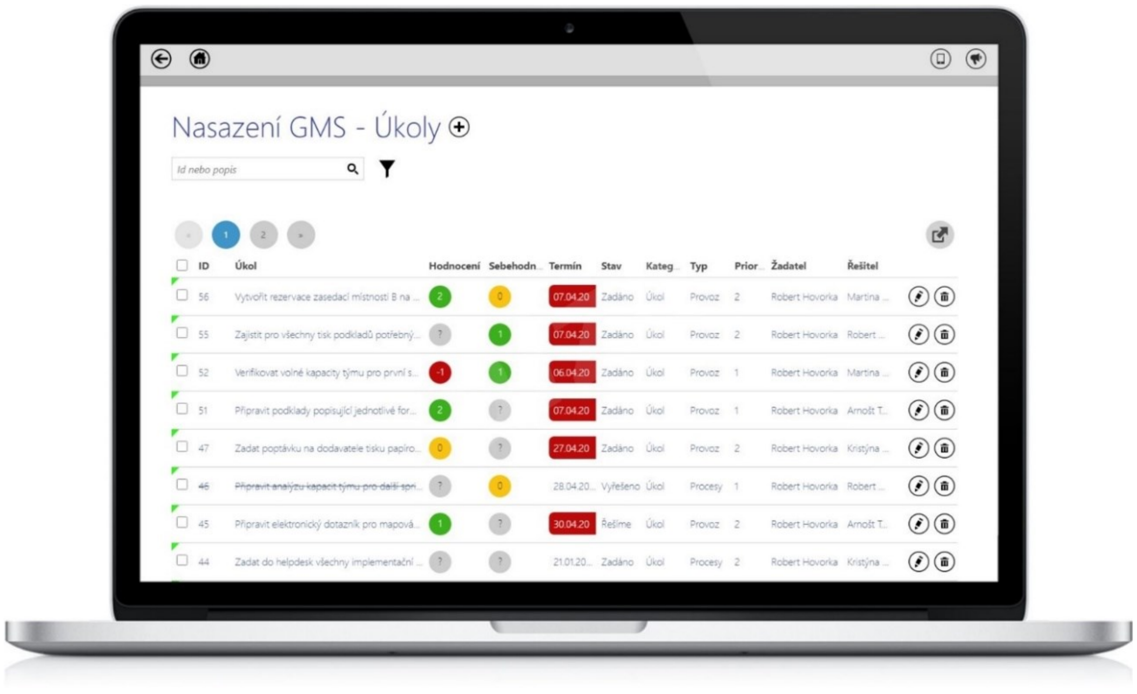
- Připravit elektronický dotazník pro mapován... Termín: 30.04.2020 0:00:00, Řešitel: Arnošt Trouba
- Návrh otázek pro průzkum mezi zaměstnanci Termín: 05.05.2020 0:00:00, Řešitel: Anna Kloubová
- Analýza současného informačního systému Termín: 20.05.2020 0:00:00, Řešitel: Igor Junák

Vyřešeno (1)

- Připravit analýzu kapacit týmu pro další sprinty Termín: 28.04.2020 0:00:00, Řešitel: Robert Hovorka

Fig. 4.1.3: *Kanban Page* getmore.getmore.cz [18]

In Figure 4.1.4 we see the tasks page. Tasks are displayed in a convenient way and with the help of this page the main communication of the team works.


Nasazení GMS - Úkoly

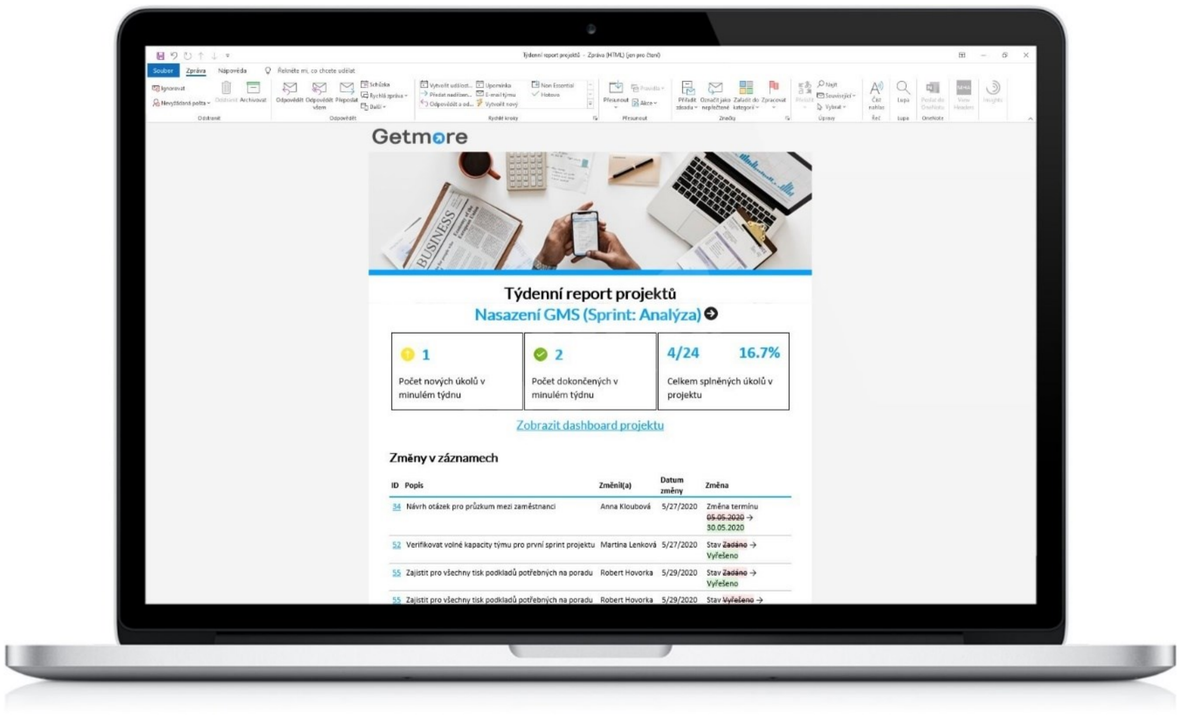
Id nebo popis

ID	Úkol	Hodnocení	Sebehodn...	Termín	Stav	Kateg...	Typ	Prior...	Žadatel	Řešitel
56	Vytvořit rezervace zasedací místnosti B na ...	2	0	07.04.20	Zadáno	Úkol	Provoz	2	Robert Hovorka	Martina ...
55	Zajistit pro všechny tisk podkladů potřebný...	?	1	07.04.20	Zadáno	Úkol	Provoz	2	Robert Hovorka	Robert ...
52	Verifikovat volné kapacity týmu pro první s...	-1	1	06.04.20	Zadáno	Úkol	Provoz	1	Robert Hovorka	Martina ...
51	Připravit podklady popisující jednotlivé for...	2	?	07.04.20	Zadáno	Úkol	Provoz	1	Robert Hovorka	Arnošt T...
47	Zadat poptávku na dodavatele tisku papíro...	0	?	27.04.20	Zadáno	Úkol	Provoz	2	Robert Hovorka	Kristýna ...
~~46~~	~~Připravit analýzu kapacit týmu pro další spri...~~	?	0	28.04.20...	Vyřešeno	Úkol	Procesy	1	Robert Hovorka	Robert ...
45	Připravit elektronický dotazník pro mapová...	1	?	30.04.20	Řešíme	Úkol	Provoz	2	Robert Hovorka	Arnošt T...
44	Zadat do helpdesk všechny implementační ...	?	?	21.01.20...	Zadáno	Úkol	Procesy	2	Robert Hovorka	Kristýna ...

Fig. 4.1.4: *Tasks Page* getmore.getmore.cz [19]

The site also sends important notifications on tasks or company events to a registered email

On the Figure 4.1.5 we can see an example of a notification



Getmore

Týdenní report projektů

Nasazení GMS (Sprint: Analýza)

1 Počet nových úkolů v minulém týdnu	2 Počet dokončených v minulém týdnu	4/24 16.7% Celkem splněných úkolů v projektu

Zobrazit dashboard projektu

Změny v záznamech

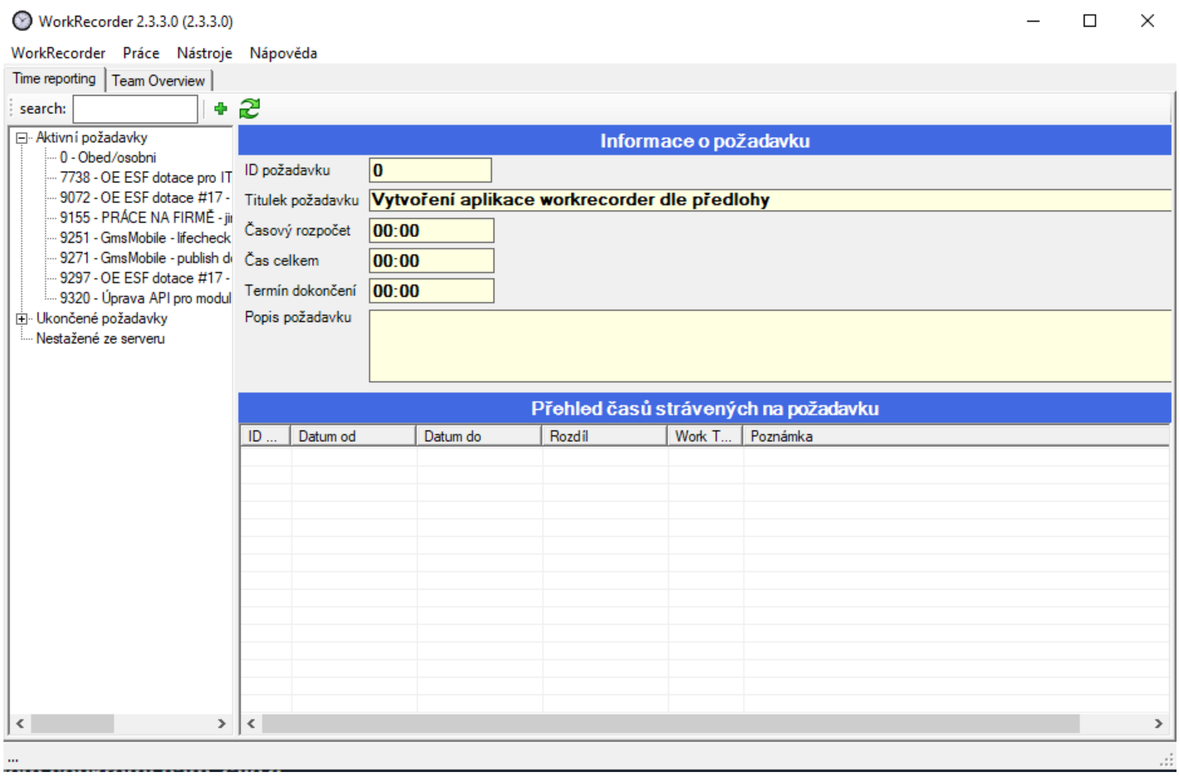
ID	Popis	Změnil(a)	Datum změny	Změna
34	Návrh otázek pro průzkum mezi zaměstnanci	Anna Kloubová	5/27/2020	Změna termínu ~~05.05.2020~~ → 30.05.2020
52	Verifikovat volné kapacity týmu pro první sprint projektu	Martina Lenková	5/27/2020	Stav ~~Zadáno~~ → Vyřešeno
55	Zajistit pro všechny tisk podkladů potřebných na poradu	Robert Hovorka	5/29/2020	Stav ~~Zadáno~~ → Vyřešeno
55	Zajistit pro všechny tisk podkladů potřebných na poradu	Robert Hovorka	5/29/2020	Stav ~~Vyřešeno~~ →

Fig. 4.1.5: *Example of an email message from* getmore.getmore.cz [20]

Application WorkRecorder

This is an application that is needed to record work time and strictly track it then send it to the notary for the payment of wages.

In Figured 4.1.6 we can see the home page of the application.

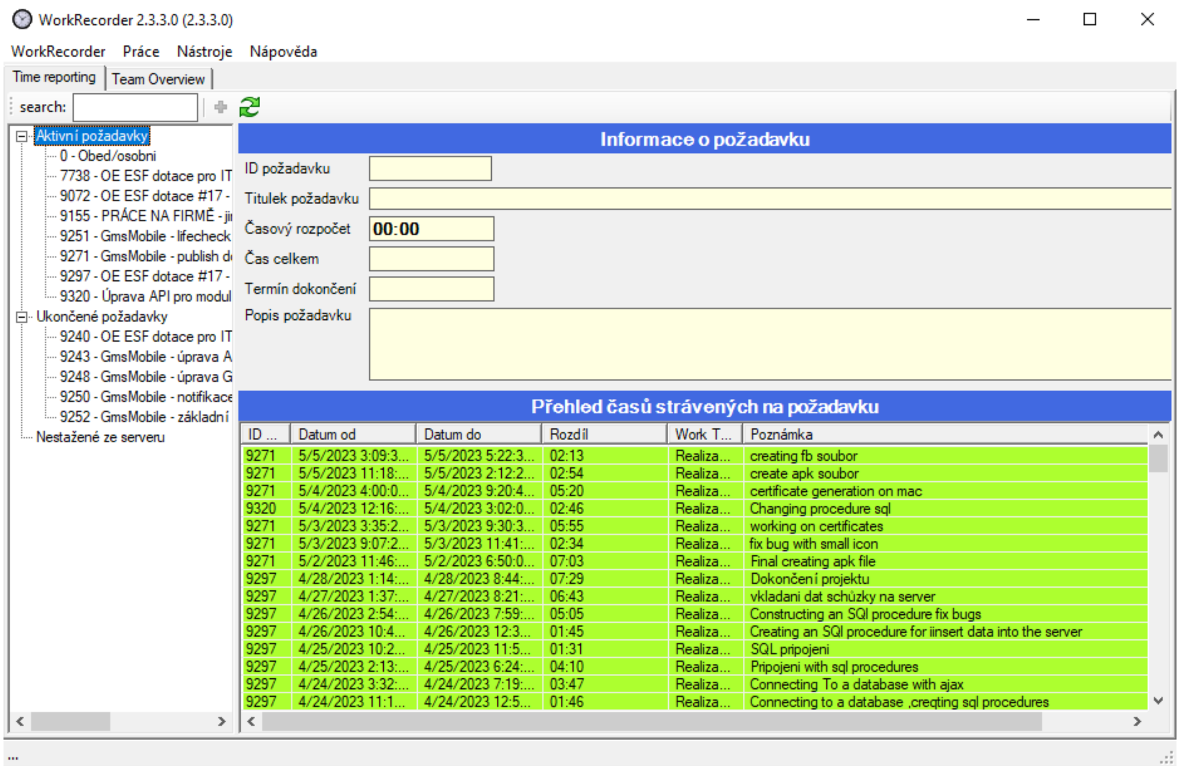


- Aktivní požadavky
  - 0 - Obed/osobni
  - 7738 - OE ESF dotace pro IT
  - 9072 - OE ESF dotace #17 -
  - 9155 - PRÁCE NA FIRMĚ - ji
  - 9251 - GmsMobile - lifecheck
  - 9271 - GmsMobile - publish d
  - 9297 - OE ESF dotace #17 -
  - 9320 - Úprava API pro modul
- Ukončené požadavky
- Nestažené ze serveru

ID požadavku	0
Titulek požadavku	Vytvoření aplikace workrecorder dle předlohy
Časový rozpočet	00:00
Čas celkem	00:00
Termín dokončení	00:00
Popis požadavku	

Fig. 4.1.6: *Main Page* WorkRecorder

In Figure 4.1.7 we see all of our hours that were recorded in the app



- Aktivní požadavky
  - 0 - Obed/osobni
  - 7738 - OE ESF dotace pro IT
  - 9072 - OE ESF dotace #17 -
  - 9155 - PRÁCE NA FIRMĚ - ji
  - 9251 - GmsMobile - lifecheck
  - 9271 - GmsMobile - publish d
  - 9297 - OE ESF dotace #17 -
  - 9320 - Úprava API pro modul
- Ukončené požadavky
  - 9240 - OE ESF dotace pro IT
  - 9243 - GmsMobile - úprava A
  - 9248 - GmsMobile - úprava G
  - 9250 - GmsMobile - notifikace
  - 9252 - GmsMobile - základní
- Nestažené ze serveru

Informace o požadavku

ID požadavku	
Titulek požadavku	
Časový rozpočet	00:00
Čas celkem	
Termín dokončení	
Popis požadavku	

Přehled časů strávených na požadavku

ID ...	Datum od	Datum do	Rozdíl	Work T...	Poznámka
9271	5/5/2023 3:09:3...	5/5/2023 5:22:3...	02:13	Realiza...	creating fb soubor
9271	5/5/2023 11:18:...	5/5/2023 2:12:2...	02:54	Realiza...	create apk soubor
9271	5/4/2023 4:00:0...	5/4/2023 9:20:4...	05:20	Realiza...	certificate generation on mac
9320	5/4/2023 12:16:...	5/4/2023 3:02:0...	02:46	Realiza...	Changing procedure sql
9271	5/3/2023 3:35:2...	5/3/2023 9:30:3...	05:55	Realiza...	working on certificates
9271	5/3/2023 9:07:2...	5/3/2023 11:41:...	02:34	Realiza...	fix bug with small icon
9271	5/2/2023 11:46:...	5/2/2023 6:50:0...	07:03	Realiza...	Final creating apk file
9297	4/28/2023 1:14:...	4/28/2023 8:44:...	07:29	Realiza...	Dokončení projektu
9297	4/27/2023 1:37:...	4/27/2023 8:21:...	06:43	Realiza...	vkladani dat schůzky na server
9297	4/26/2023 2:54:...	4/26/2023 7:59:...	05:05	Realiza...	Constructing an SQl procedure fix bugs
9297	4/26/2023 10:4...	4/26/2023 12:3...	01:45	Realiza...	Creating an SQl procedure for iinsert data into the server
9297	4/25/2023 10:2...	4/25/2023 11:5...	01:31	Realiza...	SQL pripojeni
9297	4/25/2023 2:13:...	4/25/2023 6:24:...	04:10	Realiza...	Pripojeni with sql procedures
9297	4/24/2023 3:32:...	4/24/2023 7:19:...	03:47	Realiza...	Connecting To a database with ajax
9297	4/24/2023 11:1...	4/24/2023 12:5...	01:46	Realiza...	Connecting to a database ,creqting sql procedures

Fig. 4.1.7: R*ecorded hours* WorkRecorder

Official GETMORE site

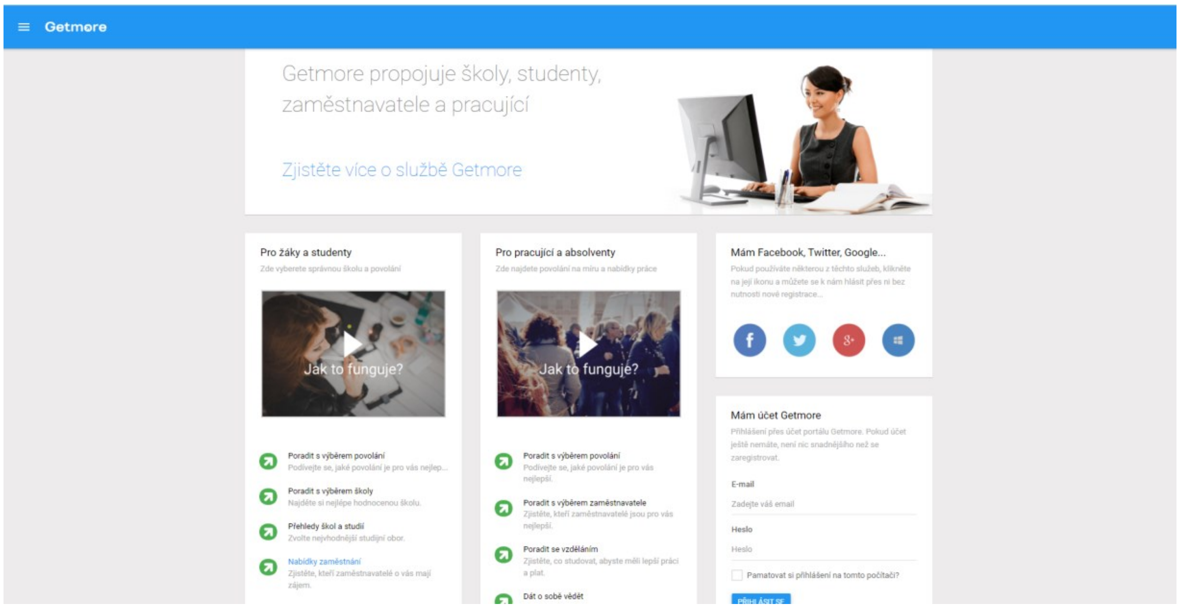
The website [21] is the official site of the company GETMORE, s.r.o. The purpose of this site is to provide information about the company's services and expertise in various areas such as web development, graphic design, internet marketing, and more.

Visitors can learn about the company's history, team members, and the solutions they offer to help businesses grow their online presence and increase sales. The website also showcases the company's portfolio, demonstrating their successful projects and giving potential clients a better understanding of their capabilities and the quality of their work.

Additionally, the website allows users to easily get in touch with the company through a contact form or by providing their phone number and email address. This makes it convenient for potential clients to request additional information, discuss their project requirements, and obtain quotes for the services they need.

In summary, the GETMORE, s.r.o. website serves as an informational and marketing platform for the company, showcasing their expertise, work examples, and providing a convenient way for potential clients to get in touch and learn more about their services.[22]

In Figure 4.1.8 we see main page www.getmore.cz


Getmore

Getmore propojuje školy, studenty, zaměstnavatele a pracující

Zjistěte více o službě Getmore

Pro žáky a studenty
Zde vyberete správnou školu a povolání

Jak to funguje?

Poradit s výběrem povolání
Podívejte se, jaké povolání je pro vás nejlep...

Poradit s výběrem školy
Najděte si nejlépe hodnocenou školu.

Přehledy škol a studií
Zvolte nejvhodnější studijní obor.

Nabídky zaměstnání
Zjistěte, kteří zaměstnavatelé o vás mají zájem.

Pro pracující a absolventy
Zde najdete povolání na míru a nabídky práce

Jak to funguje?

Poradit s výběrem povolání
Podívejte se, jaké povolání je pro vás nejlepší.

Poradit s výběrem zaměstnavatele
Zjistěte, kteří zaměstnavatelé jsou pro vás nejlepší.

Poradit se vzděláním
Zjistěte, co studovat, abyste měli lepší práci a plat.

Dát o sobě vědět

Mám Facebook, Twitter, Google...
Pokud používáte některou z těchto služeb, klikněte na její ikonu a můžete se k nám hlásit přes ni bez nutnosti nové registrace...

Mám účet Getmore
Přihlášení přes účet portálu Getmore. Pokud účet ještě nemáte, není nic snadnějšího než se zaregistrovat.

E-mail
Zadejte váš email

Heslo
Heslo

☐ Pamatovat si přihlášení na tomto počítači?

PŘIHLÁSIT SE

Fig. 4.1.8: Main page www.getmore.cz [22]

4.2 Embracing Agile Methodologies at Getmore

Agile methodologies have become an essential approach to software development, emphasizing adaptability, collaboration, and continuous improvement. By applying Agile principles, companies like Getmore can efficiently develop and deliver high-quality mobile applications in response to rapidly changing market conditions.

At Getmore, Agile practices are integrated into the development process to ensure that the company can quickly respond to user feedback and adapt to the evolving needs of their customer base. The Agile approach emphasizes iterative development, breaking down the project into smaller increments or sprints. This allows Getmore to release frequent updates, continually refine their mobile application, and keep up with the fast-paced technological landscape.

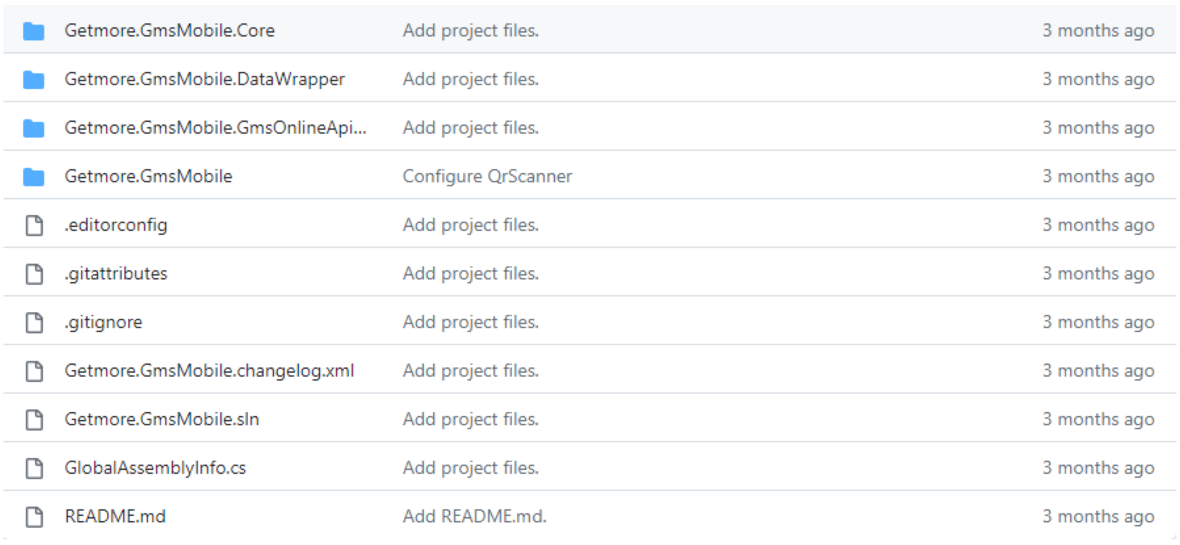
Collaboration is a key component of Agile methodologies, and at Getmore, cross-functional teams work closely together, sharing knowledge and expertise to address complex challenges. Regular communication and feedback loops between developers, designers, product owners, and other stakeholders ensure that all team members have a clear understanding of the project's objectives and progress. [23]

Additionally, Getmore embraces the principle of continuous improvement, consistently evaluating their development processes to identify areas for optimization. Sprint retrospectives are held at the end of each iteration to review performance, discuss lessons learned, and develop action plans for future enhancements.

By incorporating Agile methodologies into their software development process, Getmore can maintain a flexible and adaptive approach, ensuring that their mobile application remains competitive and relevant in an ever-evolving marketplace.

4.3 Initial state GMS MOBILE

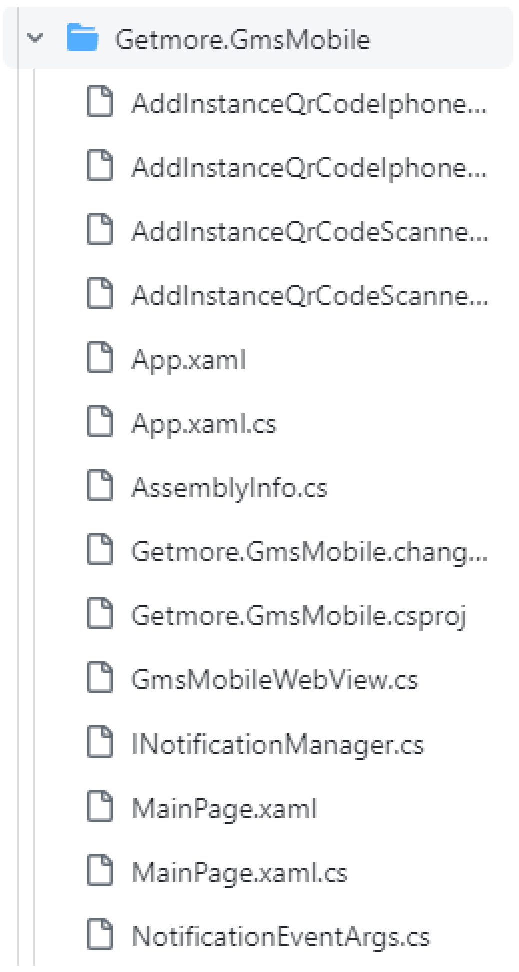
This is the file state of the project which author got when started the project, the application had the WebView module configured and a working Qr cod scanner which only worked on the Android version.



Getmore.GmsMobile.Core	Add project files.	3 months ago
Getmore.GmsMobile.DataWrapper	Add project files.	3 months ago
Getmore.GmsMobile.GmsOnlineApi...	Add project files.	3 months ago
Getmore.GmsMobile	Configure QrScanner	3 months ago
.editorconfig	Add project files.	3 months ago
.gitattributes	Add project files.	3 months ago
.gitignore	Add project files.	3 months ago
Getmore.GmsMobile.changelog.xml	Add project files.	3 months ago
Getmore.GmsMobile.sln	Add project files.	3 months ago
GlobalAssemblyInfo.cs	Add project files.	3 months ago
README.md	Add README.md.	3 months ago

Fig. 4.2: File structure GMS Mobile

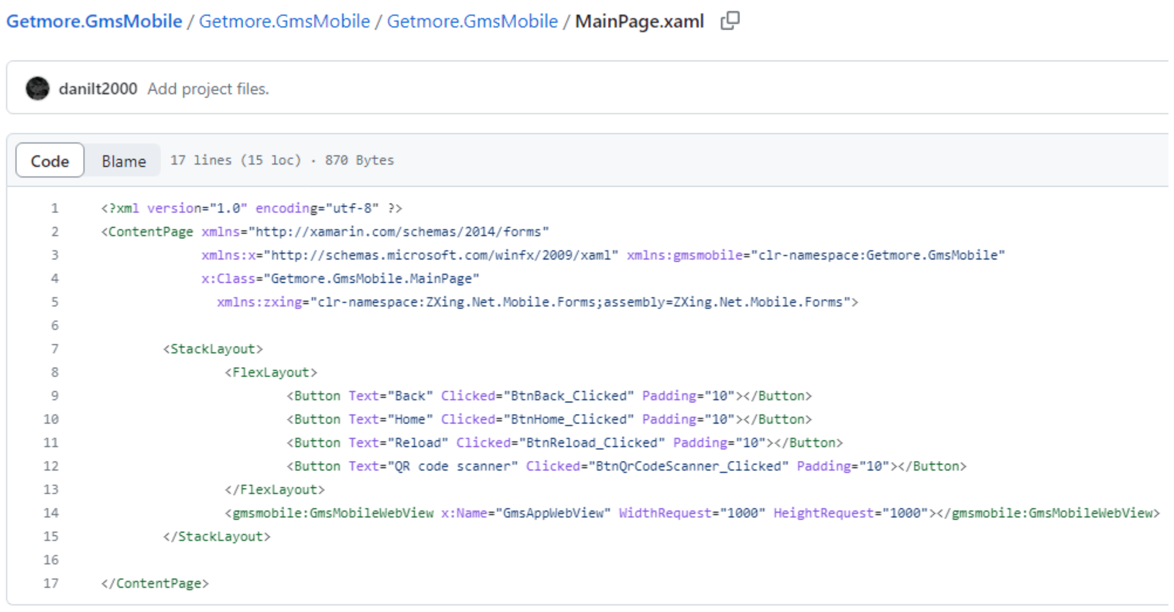
In this Figure 4.3 we see the structure of the Xamarin project.



```
Getmore.GmsMobile
    AddInstanceQrCodeIphone...
    AddInstanceQrCodeIphone...
    AddInstanceQrCodeScanne...
    AddInstanceQrCodeScanne...
    App.xaml
    App.xaml.cs
    AssemblyInfo.cs
    Getmore.GmsMobile.chang...
    Getmore.GmsMobile.csproj
    GmsMobileWebView.cs
    INotificationManager.cs
    MainPage.xaml
    MainPage.xaml.cs
    NotificationEventArgs.cs
```

Fig. 4.3: Xamarin File structure GMS Mobile

On the main page of the application were the following buttons Back Home Reload and QR code Scanner,page was written in xaml.



Getmore.GmsMobile / Getmore.GmsMobile / Getmore.GmsMobile / MainPage.xaml

danilt2000 Add project files.

Code Blame 17 lines (15 loc) · 870 Bytes

```xml
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml" xmlns:gmsmobile="clr-namespace:Getmore.GmsMobile"
             x:Class="Getmore.GmsMobile.MainPage"
               xmlns:zxing="clr-namespace:ZXing.Net.Mobile.Forms;assembly=ZXing.Net.Mobile.Forms">

        <StackLayout>
                <FlexLayout>
                        <Button Text="Back" Clicked="BtnBack_Clicked" Padding="10"></Button>
                        <Button Text="Home" Clicked="BtnHome_Clicked" Padding="10"></Button>
                        <Button Text="Reload" Clicked="BtnReload_Clicked" Padding="10"></Button>
                        <Button Text="QR code scanner" Clicked="BtnQrCodeScanner_Clicked" Padding="10"></Button>
                </FlexLayout>
                <gmsmobile:GmsMobileWebView x:Name="GmsAppWebView" WidthRequest="1000" HeightRequest="1000"></gmsmobile:GmsMobileWebView>
        </StackLayout>

</ContentPage>
```

Fig. 4.4: MainPage xaml GMS Mobile

4.4 GMS Mobile extension

- Mobile apps for Android and iOS.

- Preferred solution based on Xamarin.Forms.

1. Home screen (screen for adding a new instance)

- This is displayed the first time the application is started (or every time it is started if no instance is registered).
- It is also displayed when you click on the "Add another instance..." button, and in this case it also contains a "Back" navigation element to return to the Instance Switching Screen.

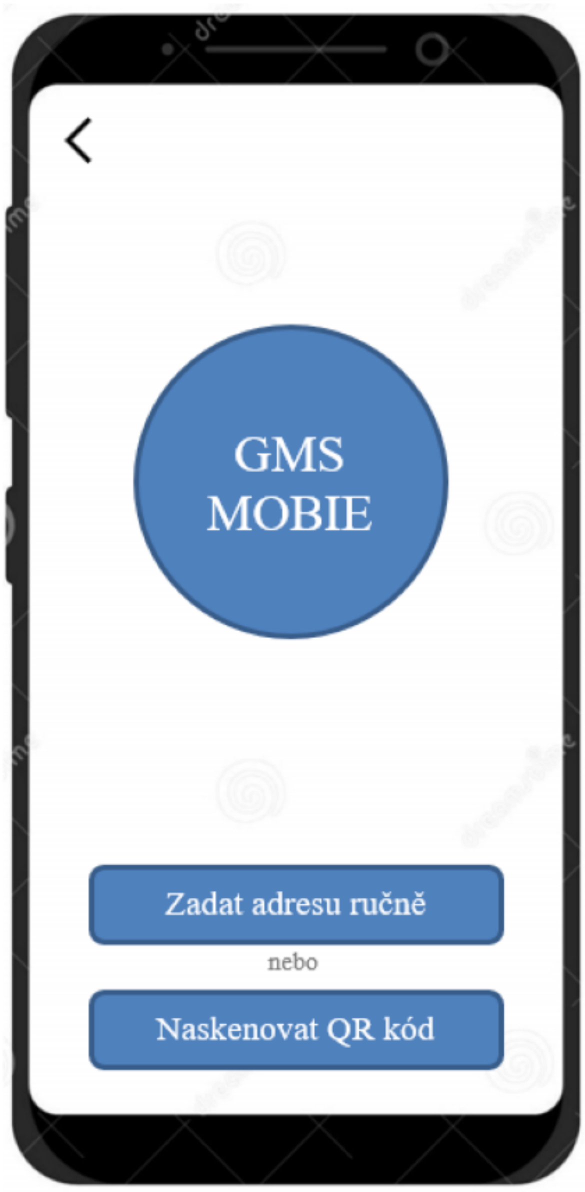
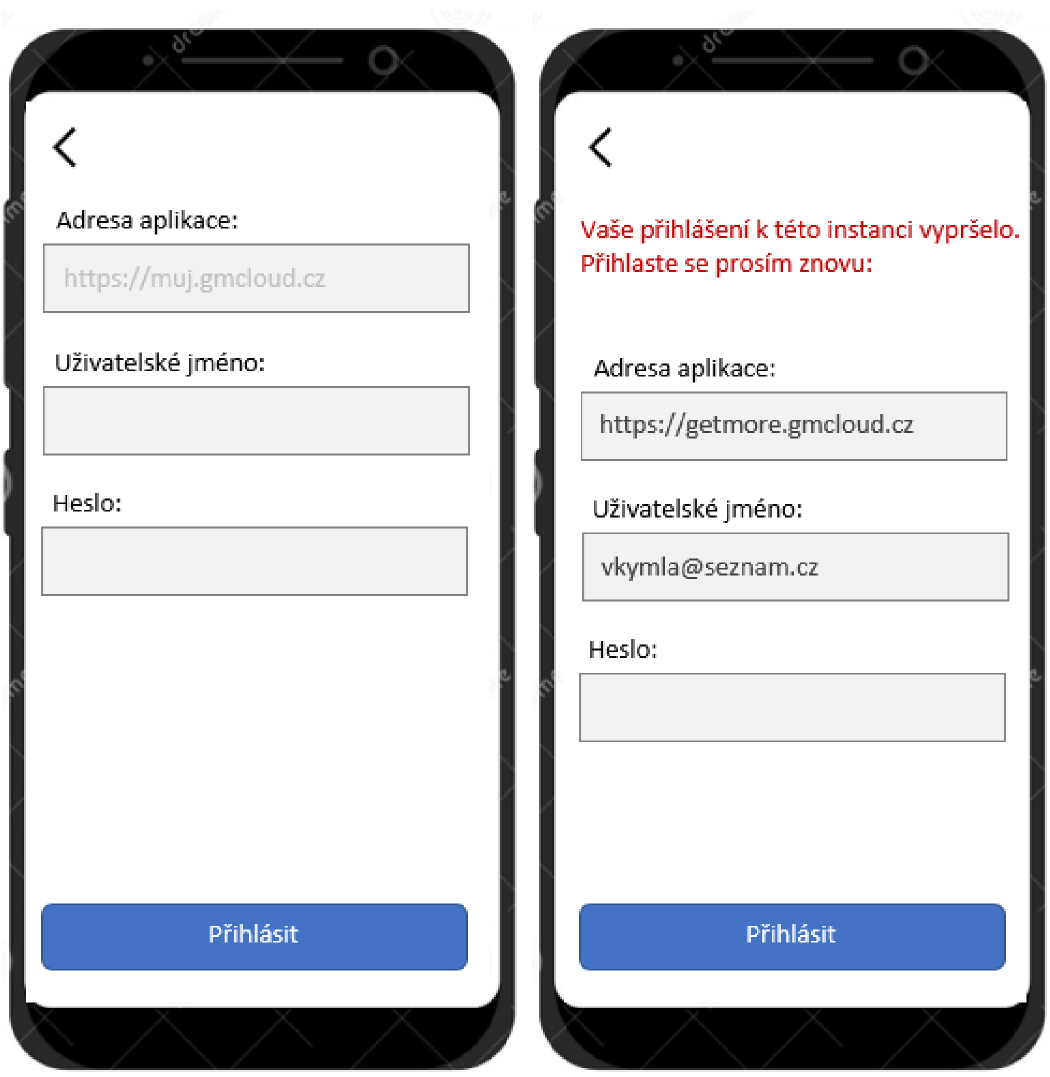


Fig. 4.5: Home screen GMS Mobile

2. Screen for manually adding an instance

- It is displayed when you click on the "**Enter address manually**" button.
- Allows you to enter input values:
    - Application address - validation to a valid URL, mandatory value, placeholder **http://muj.gmcloud.cz**.
    - Username **-** required value.
    - Password **-** required value (password input).
- The header contains a "**Back**" navigation element to return to the Home screen
- After clicking on Login:
    - In case of successful login - navigation to the Main Screen.
    - In case of failure - displays an appropriate error ("**Incorrect password**" etc.).



Vaše přihlášení k této instanci vypršelo. Přihlaste se prosím znovu:

https://getmore.gmcloud.cz

vkymla@seznam.cz

Fig. 4.6: ManulAdd GMS Mobile

3.QR code connection screen

- It is displayed after clicking on the "**Scan QR code**" button.
- Allows to scan a QR from the web app screen - shows the active camera view with the QR code scanner QR kódu.
- The header contains a "**Back**" navigation element to return to the Home screen.
- When a QR code is detected in the viewfinder, a login is attempted:
  - In case of successful login - navigation to Main page.
  - If unsuccessful - hides the view and instead displays an appropriate error.

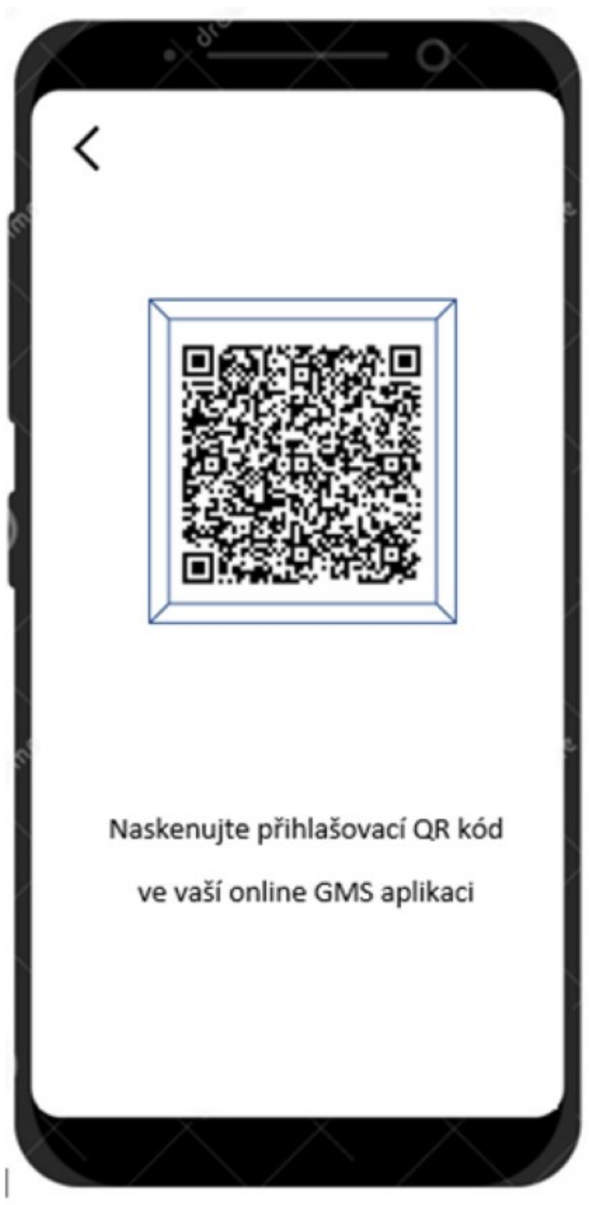


Fig. 4.7: QR code page GMS Mobile

4. Instance switching screen

- It is displayed after clicking on the "**Logout**" option in the application.
- Displays a list of instances. For each instance, it displays:
  - Logo
  - Instance name
  - User name
  - Delete instance icon - click and confirm to remove the instance.
  - Clicking on the instance name navigates to the Main Screen

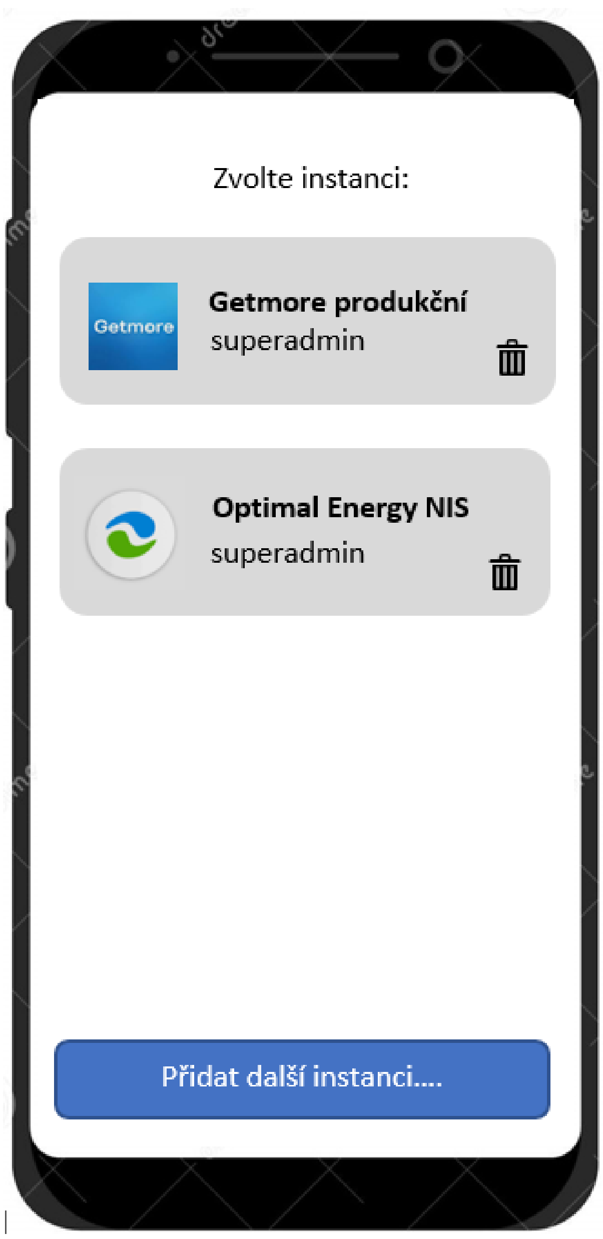
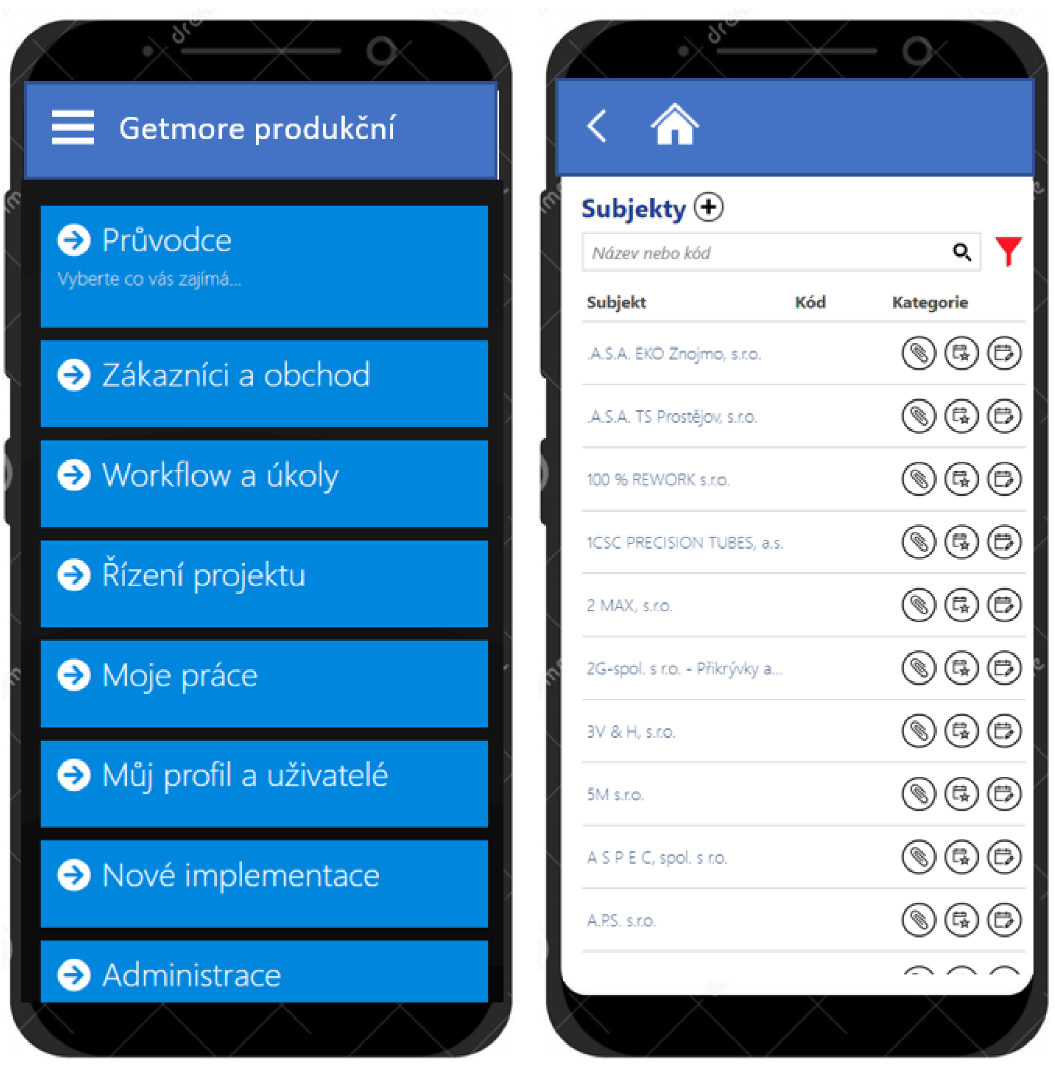


Fig. 4.8: Switch instance Page GMS Mobile

5. Main screen

- It is displayed after selecting an instance on the Instance Switching Screen.
- It is also displayed automatically when the application is started if the application was last terminated with the selected instance.
- It contains a navigation bar in the header.



Subjekt	Kód	Kategorie
.A.S.A. EKO Znojmo, s.r.o.		
.A.S.A. TS Prostějov, s.r.o.		
100 % REWORK s.r.o.		
1CSC PRECISION TUBES, a.s.		
2 MAX, s.r.o.		
2G-spol. s r.o. - Přikrývky a...		
3V & H, s.r.o.		
5M s.r.o.		
A S P E C, spol. s r.o.		
A.P.S. s.r.o.		

Fig. 4.9: Main Page GMS Mobile

6. Main menu of the application

- It appears as a floating menu when you click on "**Hamburger**" on the Main Screen.
- Includes options:
  - Logout - navigates to the Instance Switching Screen.
  - Settings **-** navigates to the Settings screen.
  - About the app **-** navigates to the About screen.

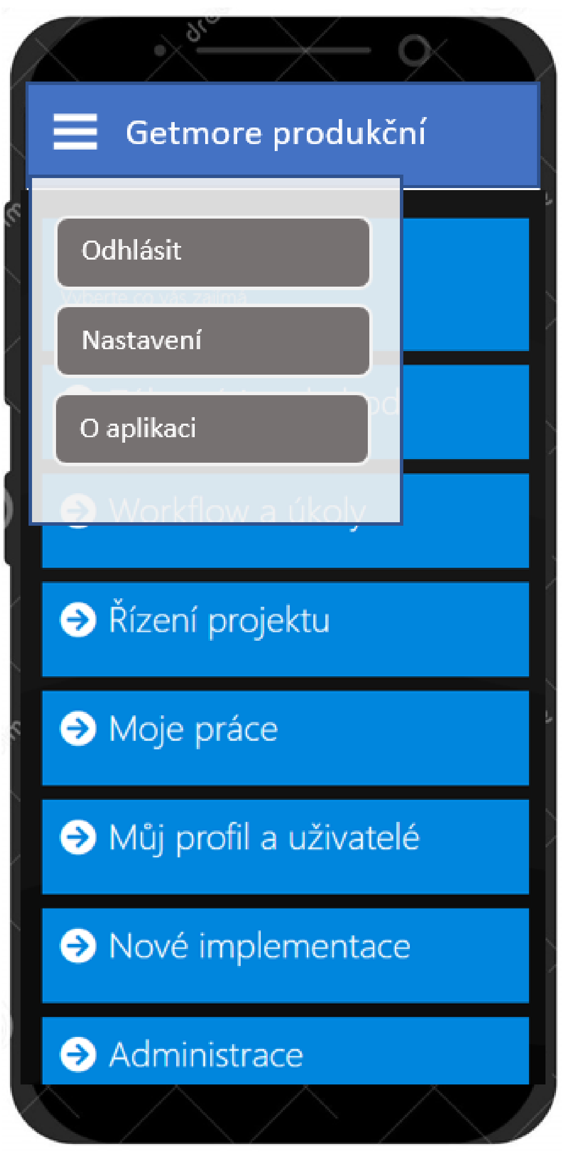


Fig. 4.9.1: Main menu GMS Mobile

7. Settings screen

- It is displayed after clicking on the Settings option in the Main menu of the application.
- Includes a "**Back**" navigation element to return to the Home screen.
- Settings:
  - Enable notifications - enables the display of push notifications.
  - Delete temporary files - performs a complete cleanup of the WebView component.

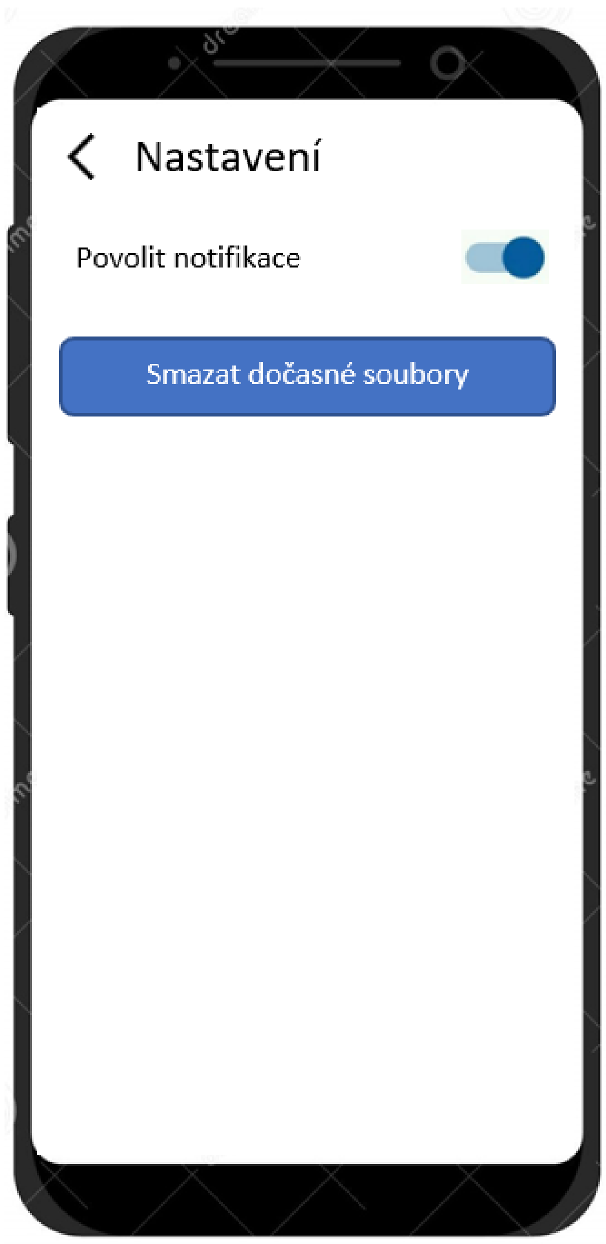
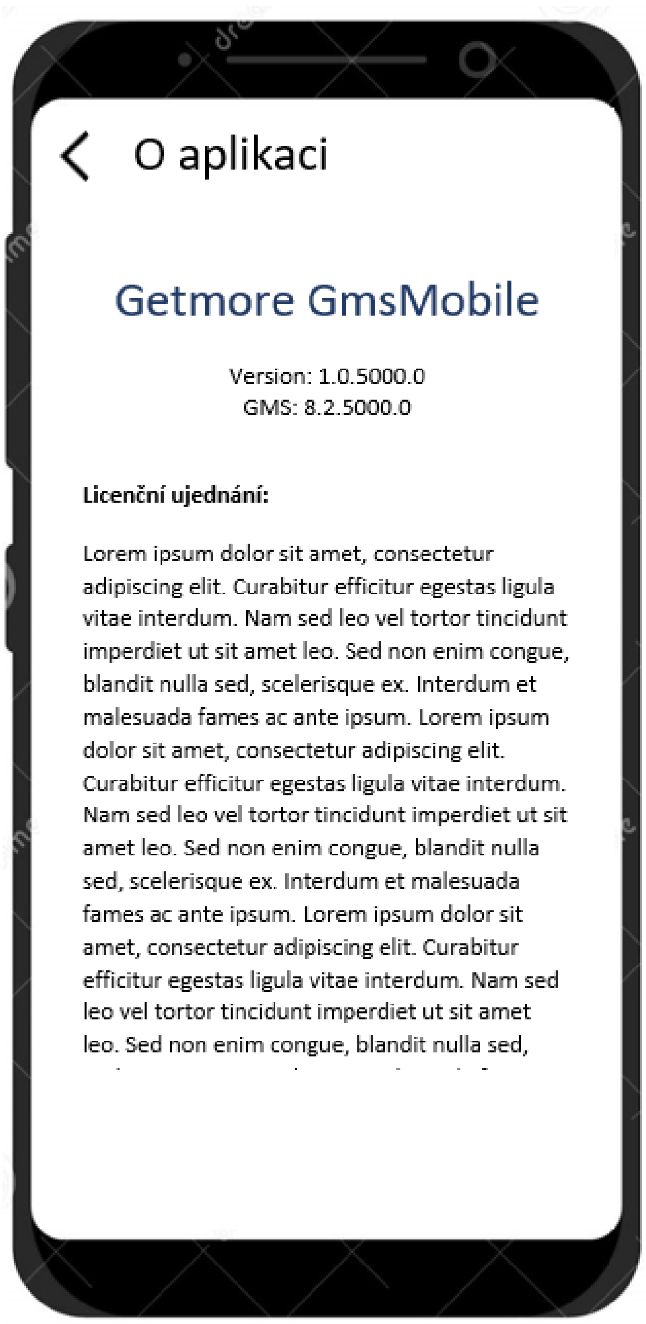


Fig. 4.9.2: Settings Page GMS Mobile

8.About screen

- Displays when you click the About option in the Main menu of the application.
- Includes a "**Back**" navigation element to return to the Main screen.
- Displays static text information:
  - GmsMobile app version
  - Version of the GMS web application
  - Licensing Agreement


O aplikaci

Getmore GmsMobile

Version: 1.0.5000.0
GMS: 8.2.5000.0

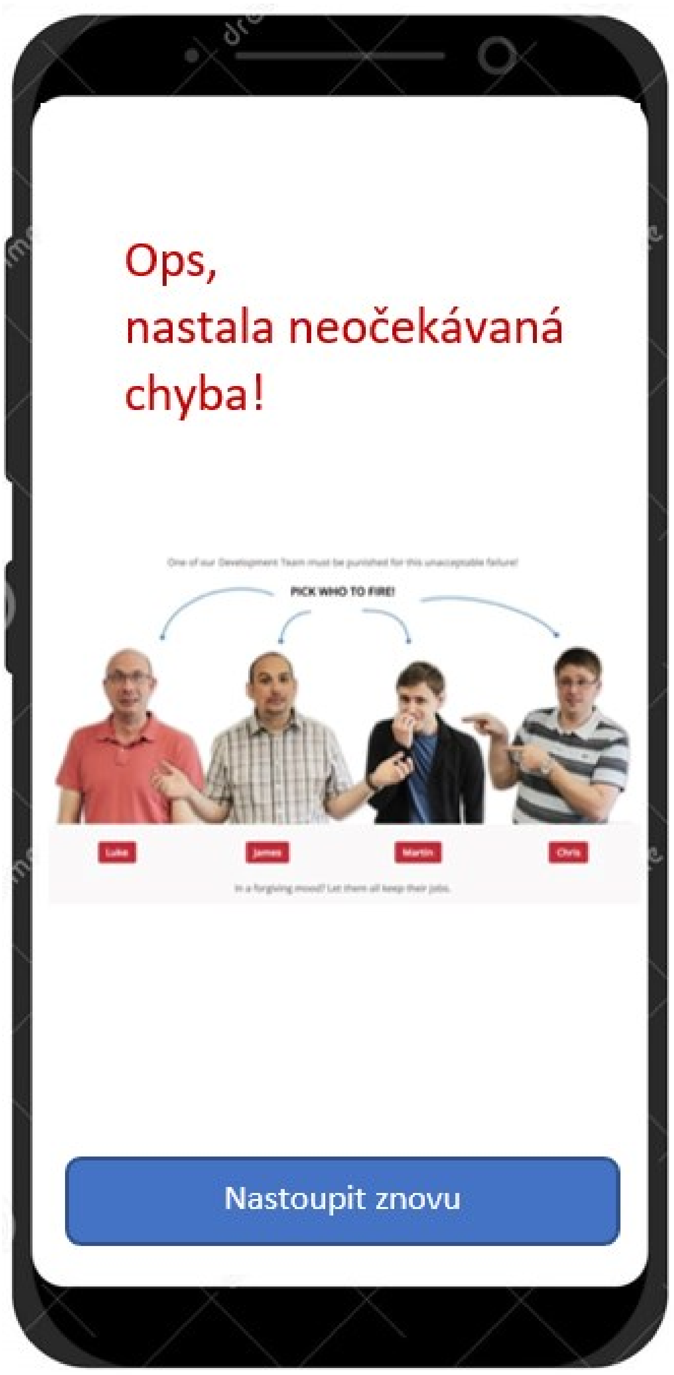
Licenční ujednání:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur efficitur egestas ligula vitae interdum. Nam sed leo vel tortor tincidunt imperdiet ut sit amet leo. Sed non enim congue, blandit nulla sed, scelerisque ex. Interdum et malesuada fames ac ante ipsum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur efficitur egestas ligula vitae interdum. Nam sed leo vel tortor tincidunt imperdiet ut sit amet leo. Sed non enim congue, blandit nulla sed, scelerisque ex. Interdum et malesuada fames ac ante ipsum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur efficitur egestas ligula vitae interdum. Nam sed leo vel tortor tincidunt imperdiet ut sit amet leo. Sed non enim congue, blandit nulla sed,

Fig. 4.9.3: About Page GMS Mobile

9. Error screen

- Used to handle error conditions in the mobile application and inside the web application.
- Displayed ad-hoc as needed.



Ops,
nastala neočekávaná chyba!

One of our Development Team must be punished for this unacceptable failure!

PICK WHO TO FIRE!

In a forgiving mood? Let them all keep their jobs.

Fig. 4.9.4: Error Page GMS Mobile

4.5 Outline description of the main functions GMS Mobile

- These are the main functions that the application should be able to do.

Authentication and login:

- After entering the application address, name and password or scanning the QR code, the data is sent to the API of the respective GMS application.
    - QR code contains: application address, username, temporary login key.
- The GMS application generates a login token, stores it in the database and sends it to the mobile application.
- The token has unlimited validity (however, it has a sliding expiration with a period of e.g. 1 month - i.e. the expiration occurs if it is not used for more than 1 month).
- The mobile app only stores this token (it does not store the password!).
- The login token can be invalidated on the server side (in case of a password change, security incident, etc).
- The login token is tied to a specific device (it is linked to the device ID and cannot be used elsewhere if stolen).

Instances and switching:

- The mobile app remembers the list of registered instances.
- For each instance, it remembers the respective login token.
- The application remembers the last logged instance and automatically logs on to it when the application starts.

Communication between GmsMobile and GMS:

- Mobile application can send messages/commands of predefined type to GMS (to WebView):
    - GoBack – command to navigate back in the navigation queue.
    - GoHome – command to navigate to the main dashboard of the application.
    - NavigateTo – command to navigate to a specific module. (e.g. open the corresponding WF record, etc.)

- GMS can send messages/commands of predefined type to the mobile application:
    - OnNavigate – notifies the mobile app about navigation within the app. (for the ability to edit the bar on the Home screen)
    - LoginRequired – The application session has expired and it is necessary to log in again - the application handles this message automatically and performs a new login using the saved token.
    - Error – notifies the mobile application about the error - the application responds by displaying the Error screen.

Notifications:

- The mobile app displays notifications.
- Notifications are generated on the web application side.
- The mobile app receives notifications from all instances that the user is registered to.
- After clicking on the notification, the user must first be switched to the appropriate instance and then navigate according to the data inside the notification. (the mobile application sends the NavigateTo command).

Support for language mutations:

- Mobile apps must support multiple languages. The choice is made according to the system settings on the device.

Waiting indication:

- The mobile application displays a waiting indication in all necessary cases.(especially when loading a web application into WebView).
- Subsequent waiting inside the web application is handled by the web application GMS.

LifeCheck:

- The mobile app periodically checks if the web app inside Webview is "live".

If the web app does not respond, the mobile app automatically refreshes WebView.

5 GMS MOBILE APP

This chapter describes final state of the application made by the author

5.1 Application design

- Dashboard

In Figure 5.1 we see the final design of the home page.

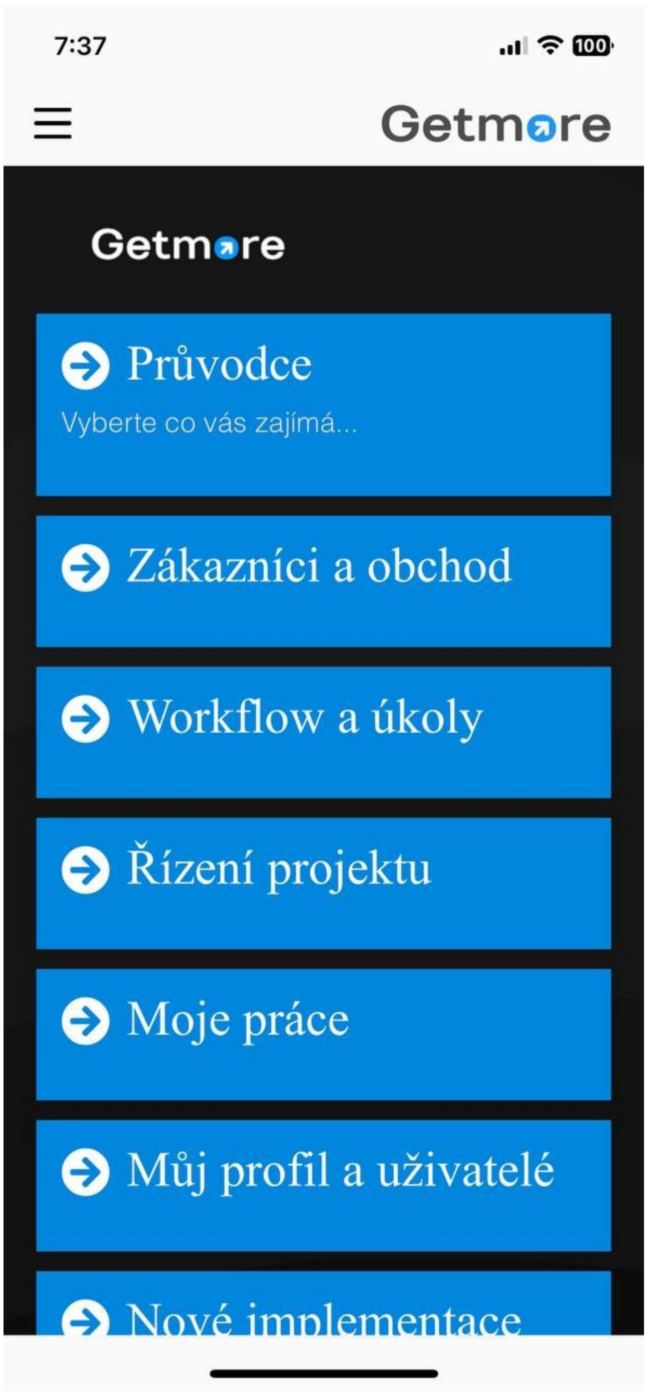


Fig. 5.1: Home page GMS Mobile

- AppShell

In Figure 5.2 we see the AppShell design in which it is possible to open other pages.

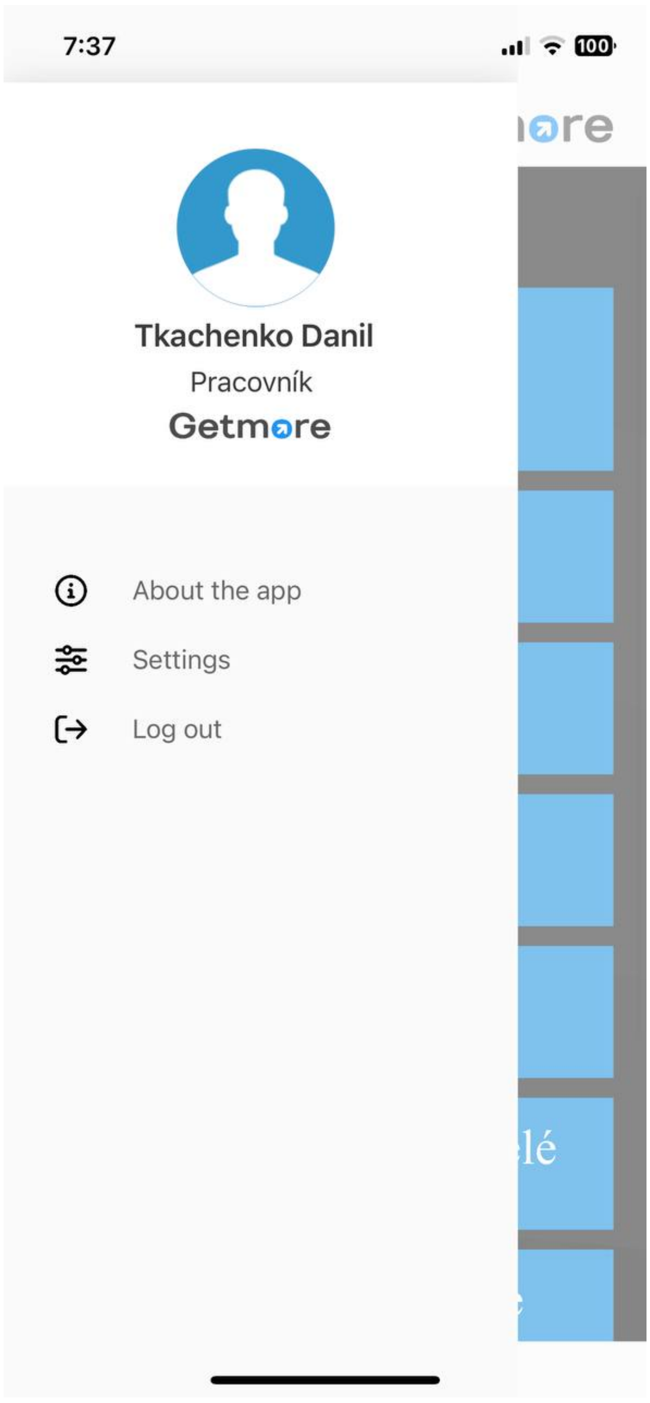
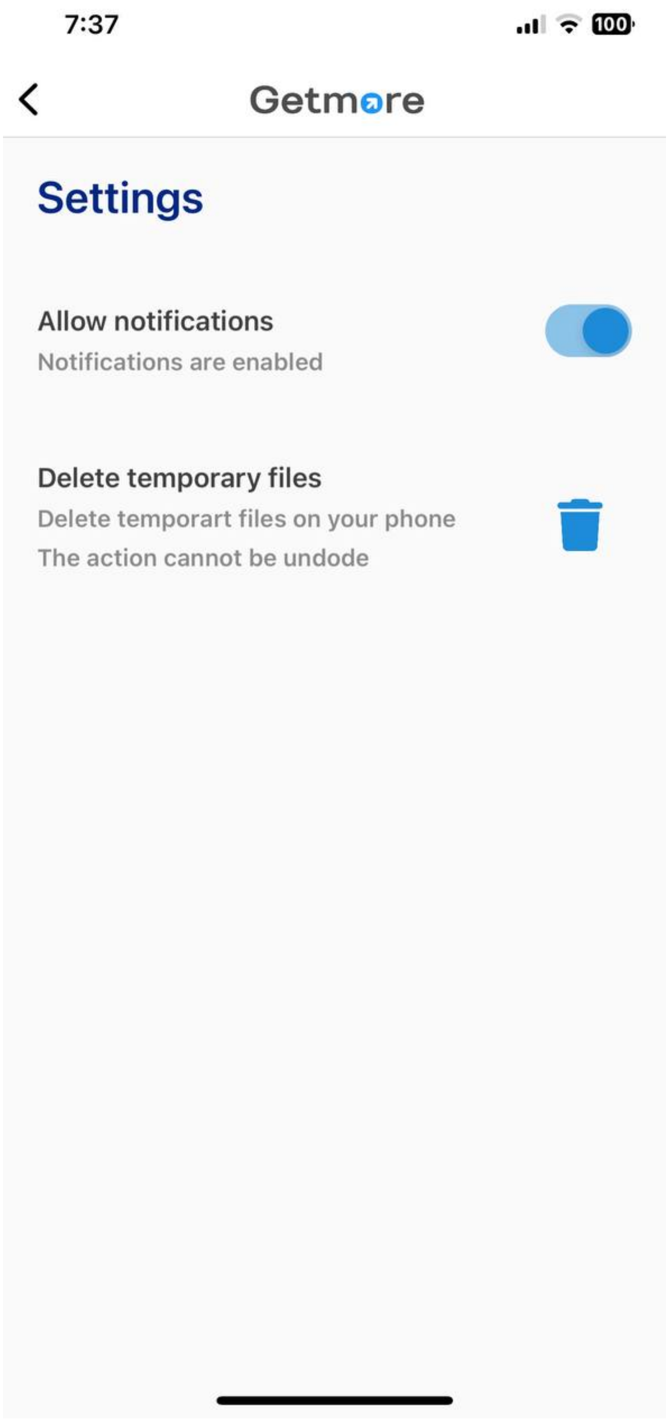


Fig. 5.2: AppShell GMS Mobile

- Settings

In Figure 5.3 we see the settings page where there is an option to turn off the notification as well as a button to remove the old css styles on the WebView.



Settings

Allow notifications
Notifications are enabled

Delete temporary files
Delete temporart files on your phone
The action cannot be undode

Fig. 5.3: Settings Page GMS Mobile

- SwitchAccountPage

In Figure 5.4 we see the final design of the page with the switching instances logic.

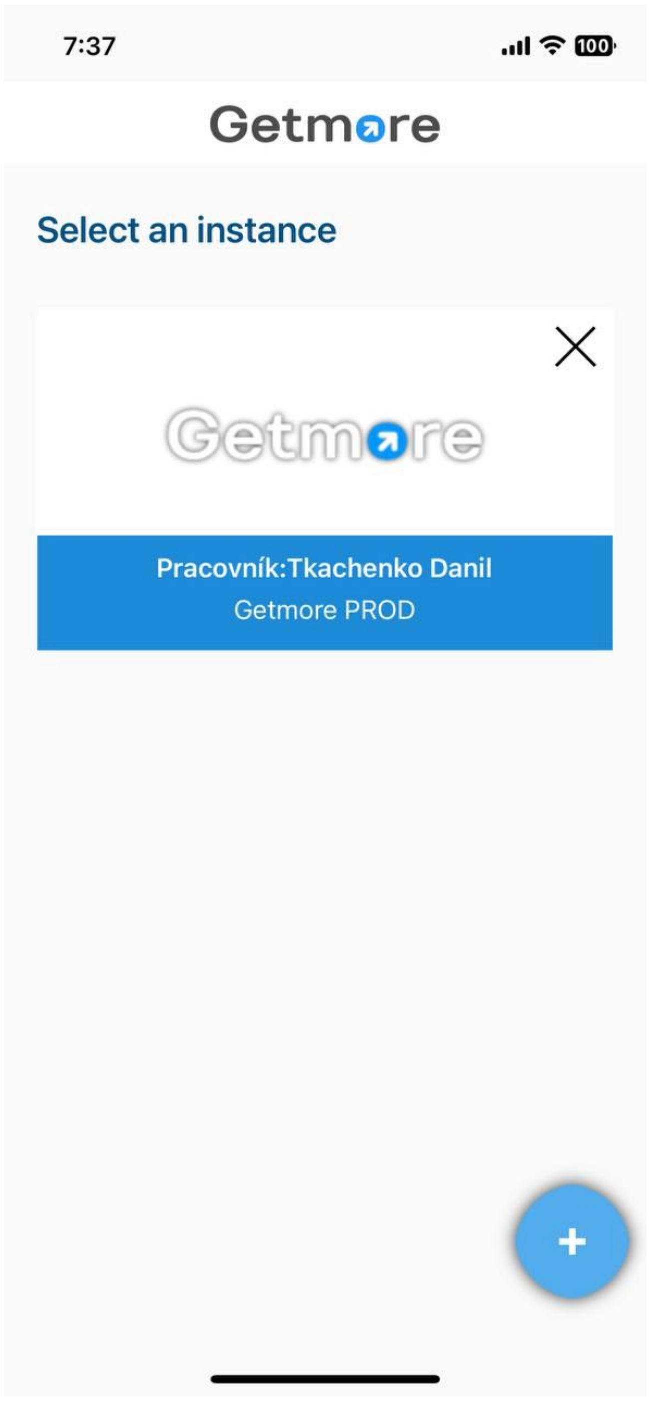


Fig. 5.4: Switch instance Page GMS Mobile

- Onboarding

In Figure 5.5 we see the home page that the user sees for the first time when the application opens.

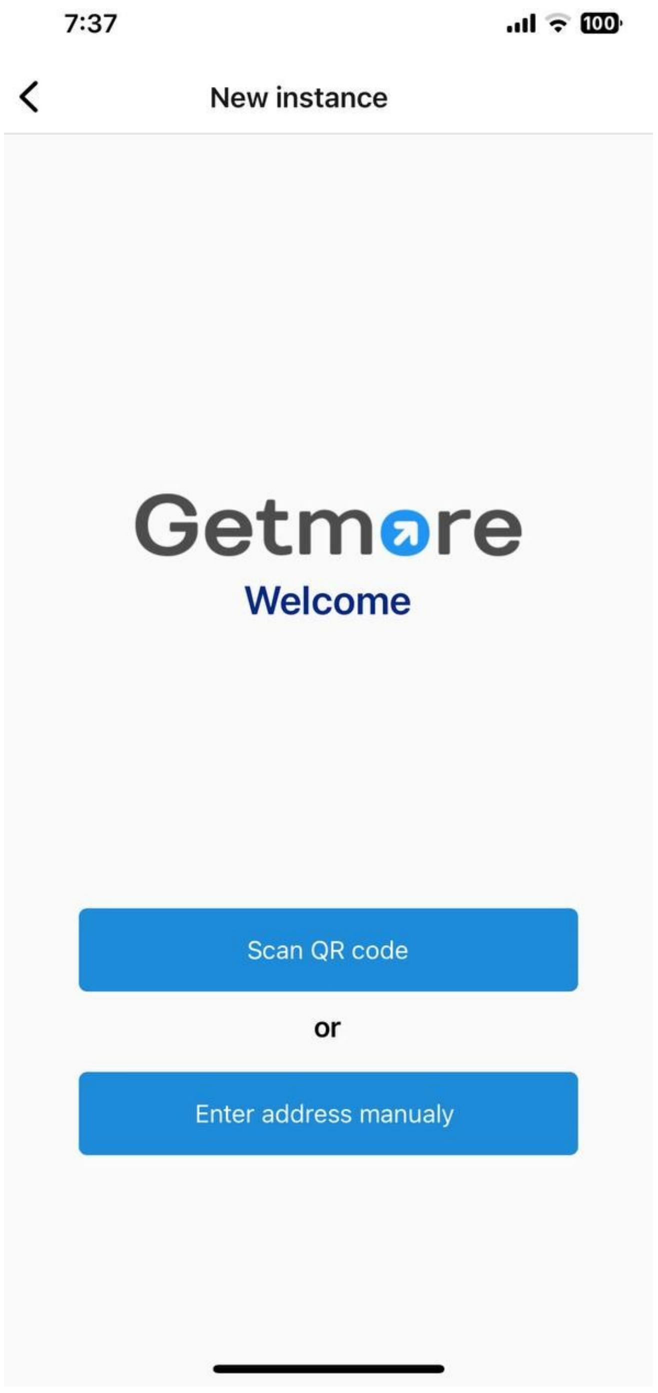


Fig. 5.5: Onboarding Page GMS Mobile

- AddInstanceQrCodeScanner

In Figure 5.6 we see the Qr code screen of the scanner application which redirects us if the scan is successful.

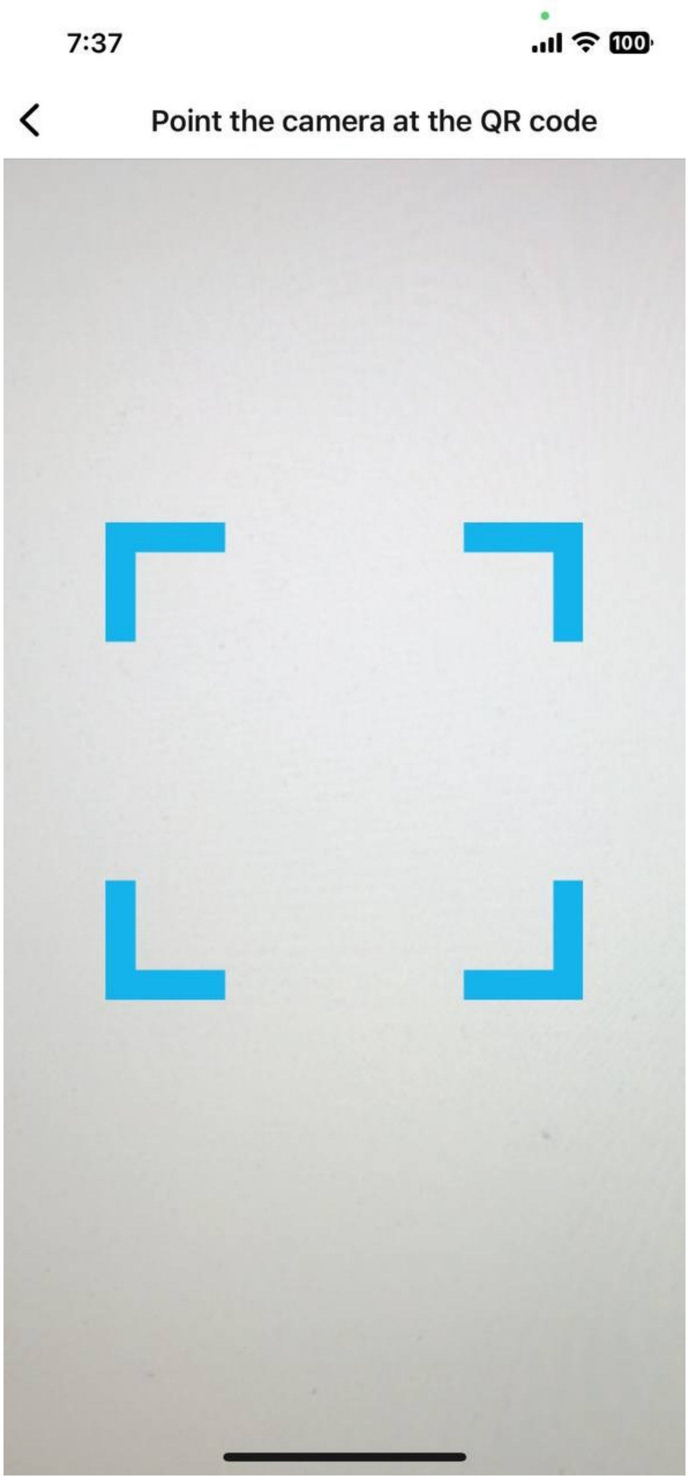


Fig. 5.6: QrCode Page GMS Mobile

- ManualAdd

In Figure 5.7 we see the design of the ManualAdd page where we can enter user data to open the instance.

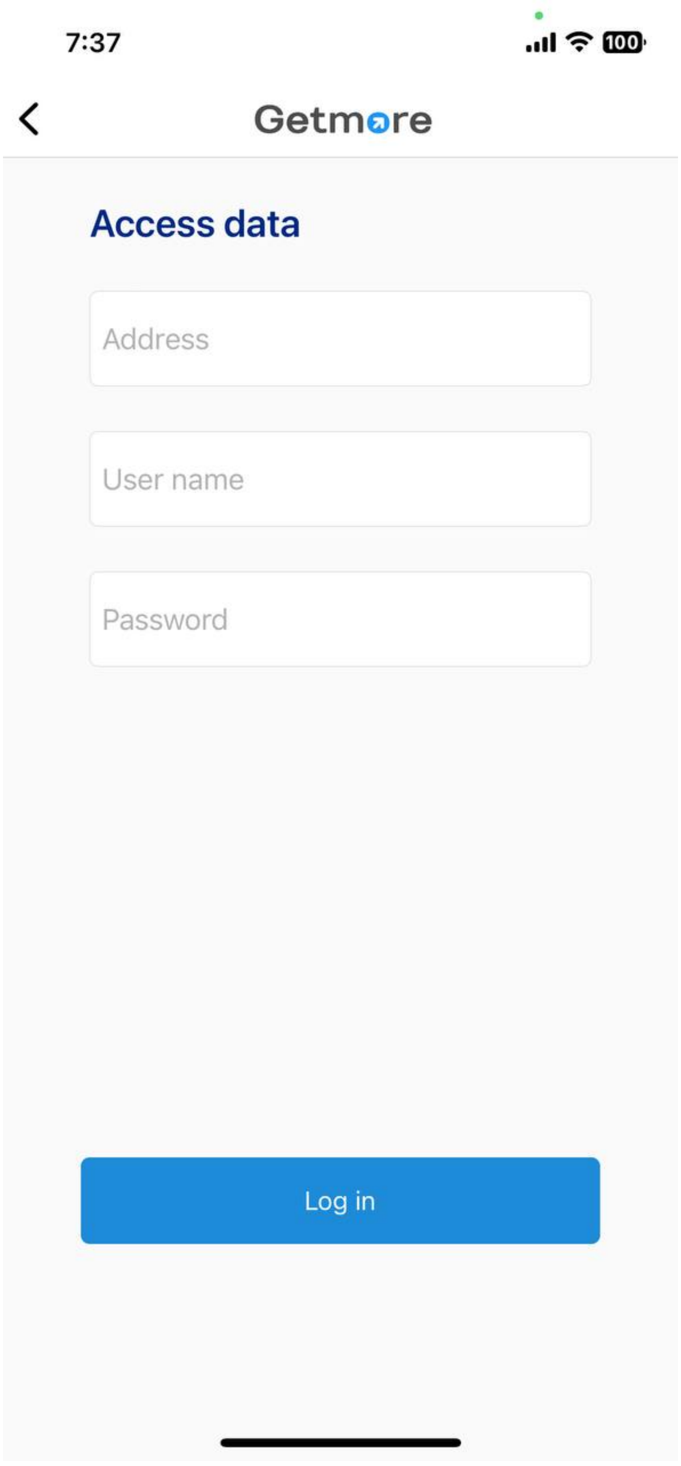
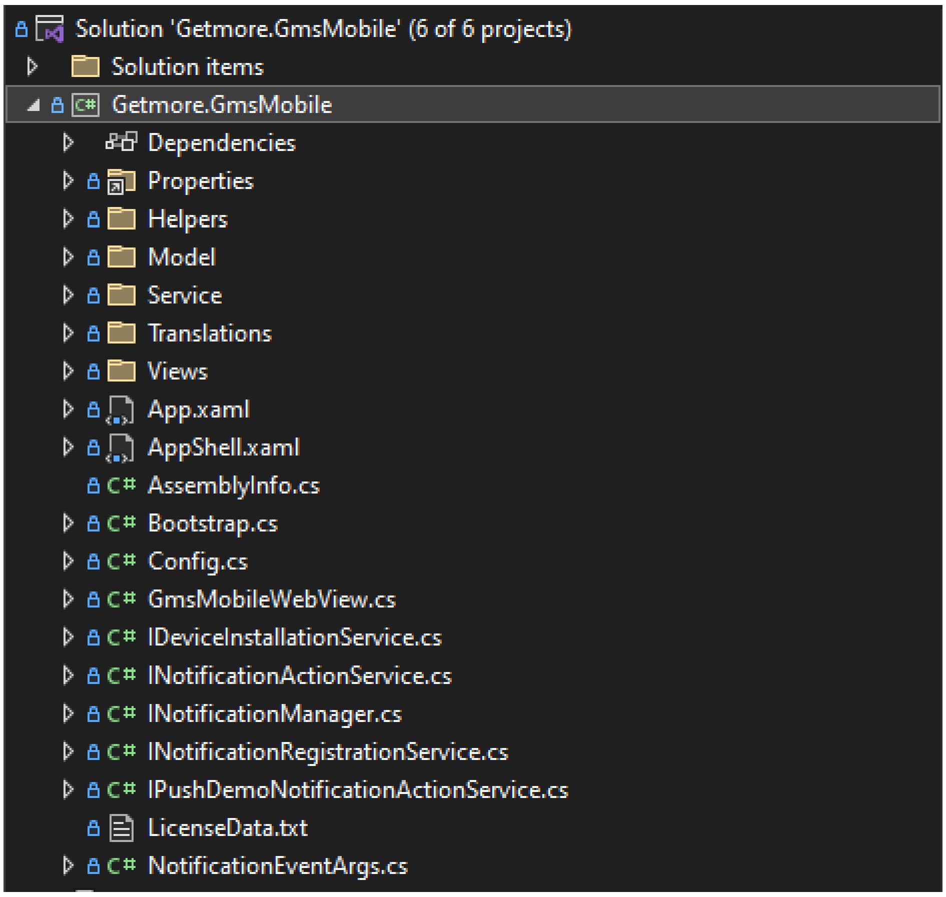


Fig. 5.7: ManulAdd Page GMS Mobile

5.2 Application structure

Finally author received this structure of project, in Figure 5.8 Helpers folder Contains the unimportant classes which usually contain 1 and 2 special functions, Model folder Contains Model classes for the application, Service folder contains different services such as working with permissions or working with notifications, Translations folder contains the logic for translating the application, Views folder is responsible for the logic of the visual part.

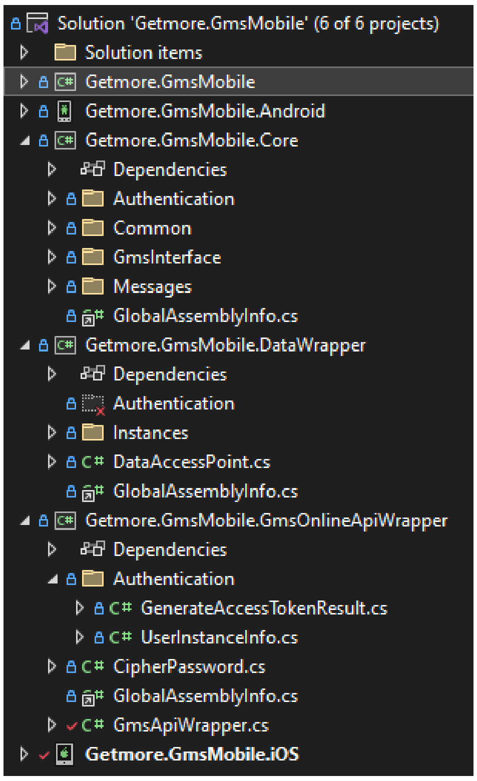


```
Solution 'Getmore.GmsMobile' (6 of 6 projects)
  Solution items
  Getmore.GmsMobile
    Dependencies
    Properties
    Helpers
    Model
    Service
    Translations
    Views
    App.xaml
    AppShell.xaml
    AssemblyInfo.cs
    Bootstrap.cs
    Config.cs
    GmsMobileWebView.cs
    IDeviceInstallationService.cs
    INotificationActionService.cs
    INotificationManager.cs
    INotificationRegistrationService.cs
    IPushDemoNotificationActionService.cs
    LicenseData.txt
    NotificationEventArgs.cs
```

Fig. 5.8: App final Xamarin structure GMS Mobile

In Figure 5.9 we can see the structure of the xamarin project

1. GetmoreGmsMobile is responsible for the structure of the xamarin project.
2. Getmore.GmsMobile is responsible for the Android project structure.
3. Getmore.GmsMobileIos is responsible for the Ios project structure.
4. Getmore.GmsMobile.Core is responsible for the part of the project to which the main Xamarin project will refer to work with Api or Data part of the project.
5. Getmore.GmsMobile.DataWrapper is responsible for working with the phone memory.
6. Getmore.GmsMobileOnlineApiWrapper is responsible for working with Api.



```
Solution 'Getmore.GmsMobile' (6 of 6 projects)
  Solution items
  Getmore.GmsMobile
  Getmore.GmsMobile.Android
  Getmore.GmsMobile.Core
    Dependencies
    Authentication
    Common
    GmsInterface
    Messages
    GlobalAssemblyInfo.cs
  Getmore.GmsMobile.DataWrapper
    Dependencies
    Authentication
    Instances
    DataAccessPoint.cs
    GlobalAssemblyInfo.cs
  Getmore.GmsMobile.GmsOnlineApiWrapper
    Dependencies
    Authentication
      GenerateAccessTokenResult.cs
      UserInstanceInfo.cs
    CipherPassword.cs
    GlobalAssemblyInfo.cs
    GmsApiWrapper.cs
  Getmore.GmsMobile.iOS
```

Fig. 5.9: App final solution structure GMS Mobile

5.2.1 ApiWrapper for authentication on Async

This is the function that makes Api requests to get the AccessTokena that is used in the application for authorization and the subsequent launch of the WebView component.

```
public async Task<GenerateAccessTokenResult> GenerateAccessTokenAsync(
   string userLogin,
   string temporaryAccessTokenHash,
   string deviceId)
{
   bool returnInstanceInfo = true;

   var request = new RestRequest("GmsMobile/Authentication/GenerateAccessToken")
      .AddJsonBody(new
      {
         userLogin,
         temporaryAccessTokenHash,
         deviceId,
         returnInstanceInfo });

   return await this.Client.PostAsync<GenerateAccessTokenResult>(request);
}
```

These method runs asynchronously in the application, which exists in C# with the async keyword.In C#, the async keyword is used to denote asynchronous programming, which is a method of executing tasks concurrently, rather than sequentially.

Asynchronous programming allows a program to perform other tasks while waiting for a long-running operation to complete, such as reading from a file, making a network request, or interacting with a database. This can improve the responsiveness and efficiency of an application, particularly in scenarios involving I/O-bound operations or high-latency tasks.

The async keyword is used in conjunction with the await keyword to define asynchronous methods in C# [24]

5.2.2 GmsInterfaces

All application interfaces are located in the project Getmore.GmsMobile.Core and exist there to facilitate work with Android or Ios parts of the project, as well as to describe the functions.

IFIleService exists to describe a class for handling instance data on the phone

```
    public interface IFileService
            {
                        void CreateInstanceFile(GmsInstanceLoginData newInstance);

                        int DeleteInstanceFromFile(GmsInstanceLoginData tappedGmsInstanceLoginData, List<GmsIn
stanceLoginData> instances);

                        void SerializationInstanceXml(List<GmsInstanceLoginData> CurrentInstancesFromFiles);

                        Task<List<GmsInstanceLoginData>> DeserializeInstanceXml();

                        GmsInstanceLoginData GetActiveInstance();
               }
```

IGmsMobilejsBridge exists to describe one method which on the application side will send certain messages to the site via WebView , the implementation of the method is different on the Android and Ios side

```
     public interface IGmsMobileJsBridge
            {
                        void SendMessageToMobileApp(string messageType, string messageJSON);
            }
```

IRequestPermissonService exists to describe a method that checks the rights to read and create files on Android

```
public interface IRequestPermissionService
    {
            Task<PermissionStatus> CheckAndRequestReadFilePermission();

            Task<PermissionStatus> CheckAndRequestWriteFilePermission();
      }
```

In C#, an interface is a contract that defines a set of abstract methods, properties, events, and/or indexers without providing their implementation. It serves as a blueprint for classes or structs that agree to implement the specified members. Interfaces are useful for designing flexible and extensible software, as they enable the creation of loosely-coupled systems, promoting separation of concerns and adherence to the SOLID principles [24]

5.2.3 Language mutations

Language mutations in the application are provided through ResourceManager logic, in the application at the moment there are Slavic Czech and English, Czech is the main language.

For example in Figure 5.9.1 we can see the structure of the translation files.

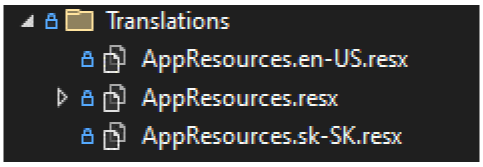
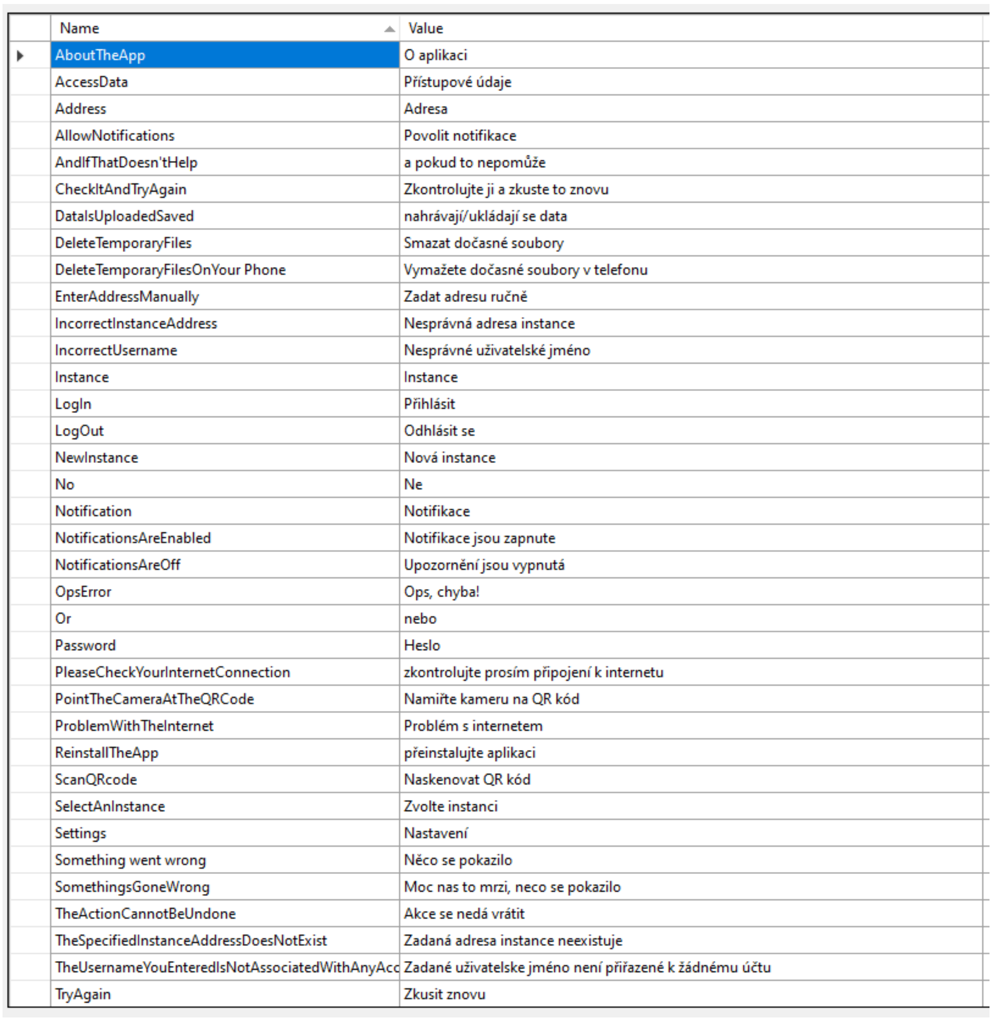


Fig. 5.9.1: Translation folder GMS Mobile

In Figure 5.9.2 we see contents of the czech to english translation file.



Name	Value
AboutTheApp	O aplikaci
AccessData	Přístupové údaje
Address	Adresa
AllowNotifications	Povolit notifikace
AndIfThatDoesn'tHelp	a pokud to nepomůže
CheckItAndTryAgain	Zkontrolujte ji a zkuste to znovu
DataIsUploadedSaved	nahrávají/ukládají se data
DeleteTemporaryFiles	Smazat dočasné soubory
DeleteTemporaryFilesOnYour Phone	Vymažete dočasné soubory v telefonu
EnterAddressManually	Zadat adresu ručně
IncorrectInstanceAddress	Nesprávná adresa instance
IncorrectUsername	Nesprávné uživatelské jméno
Instance	Instance
LogIn	Přihlásit
LogOut	Odhlásit se
NewInstance	Nová instance
No	Ne
Notification	Notifikace
NotificationsAreEnabled	Notifikace jsou zapnute
NotificationsAreOff	Upozornění jsou vypnutá
OpsError	Ops, chyba!
Or	nebo
Password	Heslo
PleaseCheckYourInternetConnection	zkontrolujte prosím připojení k internetu
PointTheCameraAtTheQRCode	Namiřte kameru na QR kód
ProblemWithTheInternet	Problém s internetem
ReinstallTheApp	přeinstalujte aplikaci
ScanQRcode	Naskenovat QR kód
SelectAnInstance	Zvolte instanci
Settings	Nastavení
Something went wrong	Něco se pokazilo
SomethingsGoneWrong	Moc nas to mrzi, neco se pokazilo
TheActionCannotBeUndone	Akce se nedá vrátit
TheSpecifiedInstanceAddressDoesNotExist	Zadaná adresa instance neexistuje
TheUsernameYouEnteredIsNotAssociatedWithAnyAcc	Zadané uživatelske jméno není přiřazené k žádnému účtu
TryAgain	Zkusit znovu

Fig. 5.9.2: CZ Translation File GMS Mobile

In this solution author used Resource files Localization .NET

- Resource Files: In .NET, localization typically involves using resource files to store culture-specific elements. Resource files are XML-based files with the extension .resx, containing key-value pairs for each localizable element. These files are compiled into binary .resources files, which are then embedded into the application's assembly or satellite assemblies. By creating separate resource files for each supported culture, developers can maintain translations and other culture-specific elements in a structured and easily maintainable manner.

- ResourceManager: The ResourceManager class, part of the System.Resources namespace, is used to retrieve culture-specific resources from resource files at runtime. It automatically selects the appropriate resources based on the application's current culture, falling back to a default culture if specific resources are not available. This enables developers to access localized resources programmatically without having to write culture-specific code.[25]

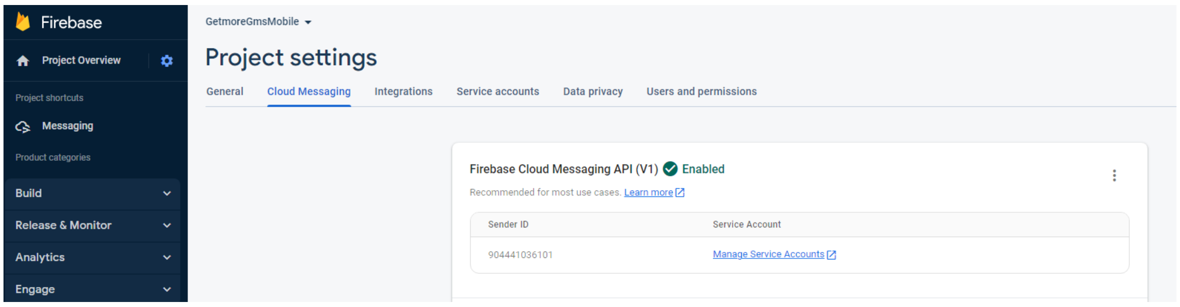
5.2.4 Notifications

To add notifications, Author used Azure Notification Hubs documentation .[26]

The steps I used for implementation.

1. Author created a Firebase project and configured it.

In Figure 5.9.3 we see Firebase project settings.



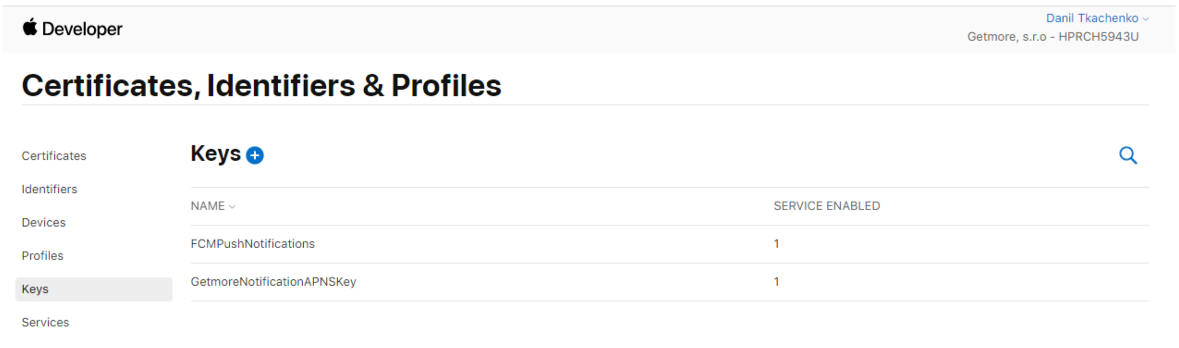
Firebase Cloud Messaging API (V1) Enabled

Recommended for most use cases. Learn more

Sender ID	Service Account
904441036101	Manage Service Accounts

Fig. 5.9.3: Firebase Project

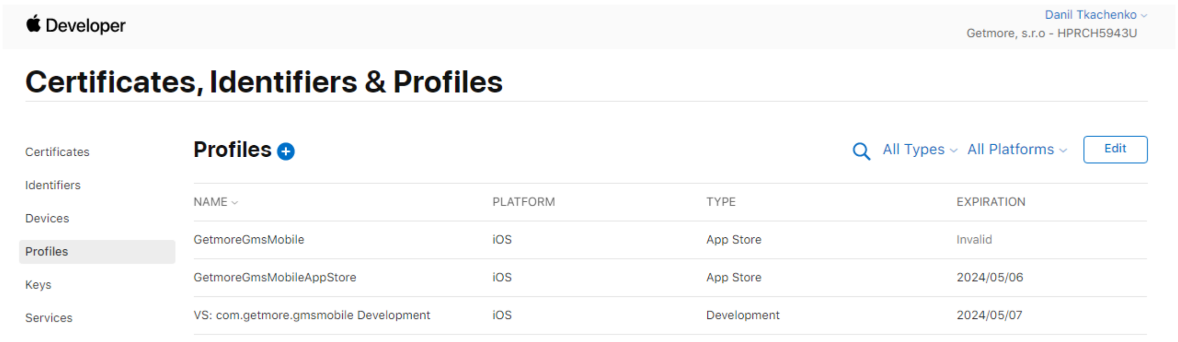
2. Author register Ios app for push notifications in apple developer account and created certificates, profile and key to send notifications.
   In Figure 5.9.4 we see FCM key to receive notifications.



NAME	SERVICE ENABLED
FCMPushNotifications	1
GetmoreNotificationAPNSKey	1

Fig. 5.9.4: GMS Mobile notification keys

In Figure 5.9.5 we see GmsMobile app profile.


Developer

Danil Tkachenko
Getmore, s.r.o - HPRCH5943U

Certificates, Identifiers & Profiles

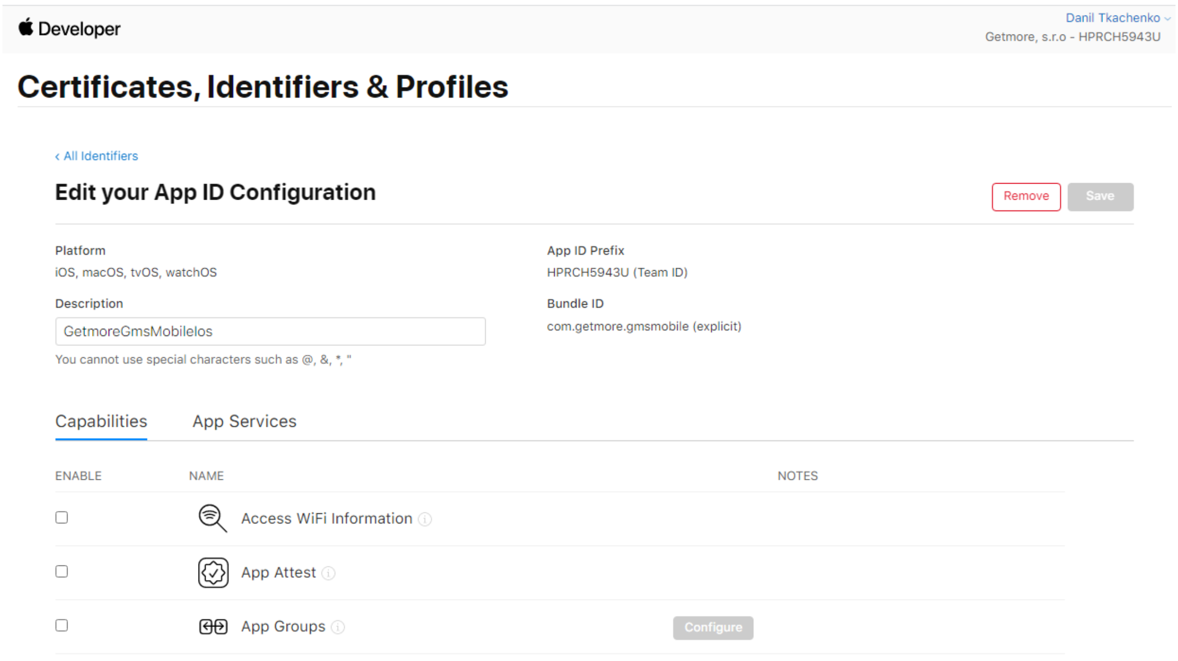
Certificates
Identifiers
Devices
Profiles
Keys
Services

Profiles

NAME	PLATFORM	TYPE	EXPIRATION
GetmoreGmsMobile	iOS	App Store	Invalid
GetmoreGmsMobileAppStore	iOS	App Store	2024/05/06
VS: com.getmore.gmsmobile Development	iOS	Development	2024/05/07

Fig. 5.9.5: GMS Mobile notification profille

In Figure 5.9.6 we see GMS Mobile profille settings.


Developer

Danil Tkachenko
Getmore, s.r.o - HPRCH5943U

Certificates, Identifiers & Profiles

‹ All Identifiers

Edit your App ID Configuration

Platform
iOS, macOS, tvOS, watchOS

App ID Prefix
HPRCH5943U (Team ID)

Description
GetmoreGmsMobileIos
You cannot use special characters such as @, &, *, "

Bundle ID
com.getmore.gmsmobile (explicit)

Capabilities App Services

ENABLE	NAME	NOTES
☐	Access WiFi Information	
☐	App Attest	
☐	App Groups	Configure

Fig. 5.9.6: GMS Mobile profille settings

3. Author added a client to the main getmore web that sends notifications.

```
private readonly NotificationHubClient hub;
internal NotificationHubService()
{
   this.hub = NotificationHubClient.CreateClientFromConnectionString(
      Settings.Default.NotificationHub_ConnectionString,
      Settings.Default.NotificationHub_Name);
}

internal Task SendPlatformNotificationsAsync(gm_notifications_loadNotificationsResult notification, string userDevice, string
clientId)
{
   string androidPayload = CreateNotificationPayloadAndroid(notification, clientId);

   string iOSPayload = CreateNotificationPayloadIos(notification, clientId);

   var sendTasks = new Task[]
   {
            this.hub.SendFcmNativeNotificationAsync(androidPayload, userDevice),

            this.hub.SendAppleNativeNotificationAsync(iOSPayload, userDevice)
   };

   return Task.WhenAll(sendTasks);
}

private static string CreateNotificationPayloadIos(gm_notifications_loadNotificationsResult notification, string clientId)
{
   return JsonSerializer.Serialize(new IosRootJsonTemplate
   {
      Aps = new IosDataJsonTemplate
      {
         Alert = $"{notification.Subject}:{notification.Body}"
      },

      IdUser = notification.IdUser.ToString(),

      ClientId = clientId,

      ReturnUrl = notification.NavigateUrl
   });
}
```

4. Author implemented notification components in Xamarin project to receive notifications

   At first startup, the application registers the device and instantly sends a notification token to the server, which is then used to send notifications to individual users

```
public NotificationRegistrationService(string baseApiUri, string apiKey)
{
        this._client = new HttpClient();

        this._client.DefaultRequestHeaders.Add("Accept", "application/json");

        this._client.DefaultRequestHeaders.Add("apikey", apiKey);

        this._baseApiUrl = baseApiUri;
}


public async Task DeregisterDeviceAsync()
{
        var cachedToken = await SecureStorage.GetAsync(CachedDeviceTokenKey)
           .ConfigureAwait(false);

        if (cachedToken == null)
                return;

        var deviceId = this.DeviceInstallationService?.GetDeviceId();

        if (string.IsNullOrWhiteSpace(deviceId))
                return;

        await this.SendAsync(HttpMethod.Delete, $"{RequestUrl}/{deviceId}")
           .ConfigureAwait(false);

        _ = SecureStorage.Remove(CachedDeviceTokenKey);

        _ = SecureStorage.Remove(CachedTagsKey);

        _ = SecureStorage.SetAsync("IsNotificationOn", "False");
}

public async Task RegisterDeviceAsync(params string[] tags)
{
        var deviceInstallation = this.DeviceInstallationService?.GetDeviceInstallation(tags);

        await this.SendAsync(HttpMethod.Put, RequestUrl, deviceInstallation)
           .ConfigureAwait(false);

        await SecureStorage.SetAsync(CachedDeviceTokenKey, deviceInstallation.PushChannel)
           .ConfigureAwait(false);

        await SecureStorage.SetAsync(CachedTagsKey, JsonConvert.SerializeObject(tags));

        _ = SecureStorage.SetAsync("IsNotificationOn", "True");
}
```

6 ACHIEVED RESULTS AND FURTHER DEVELOPMENT.

6.1 RESULTS

Author rates the application as fully implemented, it solves all the problems it was supposed to solve and shows itself well, in the future I think that the transition of the application on the MAUI framework will be needed.

6.2 Migration from Xamarin to .NET MAUI: A Thesis Overview

Future migration of the project is planned and here are the points that are important for understanding migration.

- Motivation: Migrating from Xamarin.Forms to .NET MAUI offers several benefits, including improved performance, enhanced development experience, and support for additional platforms (e.g., macOS and Windows). Furthermore, .NET MAUI is fully integrated into .NET 7, which simplifies the development process and provides access to the latest .NET features and improvements.
- Preparation: Before starting the migration process, it is essential to analyze the existing Xamarin.Forms project, identify any deprecated controls or libraries, and update them as needed. Additionally, developers should familiarize themselves with the .NET MAUI architecture, controls, and project structure, as well as the latest .NET and C# features, to ensure a smooth migration process.
- Migration Process: The migration from Xamarin.Forms to .NET MAUI can be broken down into the following steps:
- Update the project to the latest Xamarin.Forms version, addressing any breaking changes or deprecated features.
- Create a new .NET MAUI project, following the recommended project structure and organization.
- Migrate the application's code, including views, view models, models, and services, from the Xamarin.Forms project to the new .NET MAUI project.
- Update and adapt platform-specific code, such as renderers or platform effects, to .NET MAUI's new handlers system. [27]

Migrating from Xamarin.Forms to .NET MAUI involves several key steps:

1. Update the Target Framework Moniker (TFM): Xamarin projects use platform-specific TFMs like monoandroid10.0 or xamarinios10. With .NET MAUI, you'll use a single TFM - net6.0, with platform extensions such as net6.0-android or net6.0-ios.
2. Update Namespaces: Many namespaces have changed as part of the consolidation into .NET 6. For example, Xamarin.Forms becomes Microsoft.Maui.Controls.
3. Adopt New Project Structure: .NET MAUI uses the SDK-style project format and simplifies it further by introducing a single project for multiple platforms. This differs from Xamarin.Forms, which typically had separate projects for each platform.
4. Update UI Controls: While many Xamarin.Forms controls and layouts have direct equivalents in .NET MAUI, some have changed. The migration process will likely involve updating UI definitions.
5. Use Maui Startup: The startup configuration in .NET MAUI follows a more streamlined approach compared to the App.xaml and App.xaml.cs files in Xamarin.Forms. This will need to be updated in the migration process.

- Replace Xamarin.Forms controls with their .NET MAUI counterparts, adapting the application's UI as needed.
- Test the application on all target platforms, fixing any issues or discrepancies that may arise.

CONCLUSION

With the evolving landscape of mobile app development, leveraging technologies like .NET Xamarin MAUI is crucial to meet user demands and market trends. The integration of GetMore's web solutions into the existing mobile application will offer a comprehensive and unified platform to the users, enhancing user engagement and satisfaction. Given the flexibility and robustness of these technologies, there are vast possibilities for future enhancements and features, promising a bright future for the application.

REFERENCES

[1] Mobile application development [online].[cit. 2023-05-03]. Dostupné z: https://www.ibm.com/topics/mobile-application-development

[2] Android_SDK. *Wikipedia* [online].[cit. 2023-05-09]. Dostupné z: https://en.wikipedia.org/wiki/Android_SDK

[3] *Swift (programming language)* [online]. [cit. 2023-05-03]. Dostupné z: https://en.wikipedia.org/wiki/Swift_(programming_language).

[4] *C Sharp (programming language)*[online]. [cit. 2023-05-22]. Dostupné z: https://en.wikipedia.org/wiki/C_Sharp_(programming_language).

[5] HERMES, Dan a Nima MAZLOUMI. Building Xamarin.Forms Mobile Apps Using XAML: Mobile Cross-Platform XAML and Xamarin.Forms Fundamentals. California: Apress, 2019. ISBN 978-1-4842-4029-8.

[6] Architecture of a cross-platform Xamarin application [online]. 2022 [cit. 2023-05-03]. Dostupné z: https://learn.microsoft.com/en-us/xamarin/.

[7] Xamarin documentation [online]. 2022 [cit. 2023-05-03]. Dostupné z: https://learn.microsoft.com/en-us/xamarin/.

[8] Android architecture of Xamarin application [online]. 2022 [cit. 2023-05-03]. Dostupné z: https://learn.microsoft.com/en-us/xamarin/.

[9] Ios architecture of Xamarin application [online]. 2022 [cit. 2023-05-03]. Dostupné z: https://learn.microsoft.com/en-us/xamarin/.

[10] Maui overview [online]. 2022 [cit. 2023-05-03]. Dostupné z: https://learn.microsoft.com/en-us/xamarin/.

[11] What-is-maui [online]. 2023 [cit. 2023-05-08]. Dostupné z: https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui

[12] WebView [online]. 2023 [cit. 2023-05-22]. Dostupné z: https://learn.microsoft.com/cs-cz/xamarin/xamarin-forms/user-interface/webview?tabs=windows

[13] Visual Studio [online]. [cit. 2023-05-11]. Dostupné z: https://en.wikipedia.org/wiki/Visual_Studio

[14] Official Getmore company information [online]. [cit. 2023-05-11]. Dostupné z: https://rejstrik-firem.kurzy.cz/25577581/getmore-sro/

[15] Website with the company's offerings [online]. [cit. 2023-05-11]. Dostupné z: https://getmoresystem.cz/

[16] *Main site Getmore* [online]. [cit. 2023-05-11]. Dostupné z: https://getmore.getmore.cz/

[17] Profile page getmore.getmore.cz.[online][cit. 2023-05-11].Dostupné z: https://www.alza.cz/media/getmore-rizeni-agilniho-tymu-elektronicka-licence-d5855425.htm

[18] *Kanban Page* getmore.getmore.cz[online][cit. 2023-05-11].Dostupné z: https://www.alza.cz/media/getmore-rizeni-agilniho-tymu-elektronicka-licence-d5855425.htm

[19] *Tasks Page* getmore.getmore.cz [online][cit. 2023-05-11].Dostupné z: https://www.alza.cz/media/getmore-rizeni-agilniho-tymu-elektronicka-licence-d5855425.htm

[20] *Example of an email message from* getmore.getmore.cz[online][cit. 2023-05-11].Dostupné z https://www.alza.cz/media/getmore-rizeni-agilniho-tymu-elektronicka-licence-d5855425.htm

[21] Official site of the company GETMORE [online]. [cit. 2023-05-11]. Dostupné z: https://www.getmore.cz/

[22] Main page www.getmore.cz/ [online][cit. 2023-05-11]. Dostupné z: https://www.getmore.cz/

[23] ROBERT, MARTIN. MARTIN, Robert C. Clean Agile: back to basics. Boston: Pearson, [2020]. Robert C. Martin series. 2020. ISBN 978-0-13-578186-9.

[24] JON, SKEET. *C# in depth*. Fourth edition. Shelter Island, NY: Manning, 2019. ISBN 978-1-61729-453-2

[25] *Localization in .NET* [online]. [cit. 2023-05-11]. Dostupné z: https://learn.microsoft.com/en-us/dotnet/core/extensions/localization

[26] Push notifications to Xamarin.Forms apps [online]. [cit. 2023-05-11].Dostupné z https://learn.microsoft.com/en-us/azure/developer/mobile-apps/notification-hubs-backend-service-xamarin-forms

[27] Maui migration [online]. [cit. 2023-05-09]. Dostupné z: https://learn.microsoft.com/en-us/dotnet/maui/migration/

LIST OF ABBREVIATIONS

C#	Programming language.
Android SDK	Software development kit.
HTML	Hypertext Markup Language.
IDE	Integrated Development Environment.
MAUI	Multi-Platform App UI.
CSS	Cascading Style Sheets.
DOM	Document Object Model.
JavaScript	Compiled programming language.
UWP	Universal Windows Platform.
Python	Programming Language.
C++	Programming language
Microsoft	Multinational technology corporation.
Java	Programming language.
ARC	Automatic Reference Counting
TFM	Target Framework Moniker.
FCM	Firebase Cloud Messaging.
Getmore	Technological company
Xamarin	Cross-platform development framework
iOS	Apple's mobile operating system
LLVM	Compiler infrastructure project
Android	Google's mobile operating system
Google Cloud Messaging	Push notification service
Objective-C	Object-oriented C language
Xcode	Apple's development environment

Swift	Programming language
macOS	Apple's desktop operating system
watchOS	Apple's watch operating system
tvOS	Apple's TV operating system
Azure	Microsoft's cloud platform
Git	Distributed version control
.NET	Computer software framework.

LIST OF FIGURES

Fig. 1.1: Architecture of a cross-platform Xamarin application

Fig. 1.2: Android architecture of Xamarin application

Fig. 1.3: Ios architecture of Xamarin application

Fig. 1.4: Maui overview

Fig. 4.1: Website with the company's offerings

Fig. 4.1.1: Main Page getmore.getmore.cz

Fig. 4.1.2: Profile Page getmore.getmore.cz

Fig. 4.1.3: Kanban Page getmore.getmore.cz

Fig. 4.1.4: Tasks Page getmore.getmore.cz

Fig. 4.1.5: Example of an email message from getmore.getmore.cz

Fig. 4.1.6: Main Page WorkRecorder

Fig. 4.1.7: Recorded hours WorkRecorder

Fig. 4.1.8: Main page www.getmore.cz

Fig. 4.3: File structure GMS Mobile

Fig. 4.3: Xamarin File structure GMS Mobile

Fig. 4.4: MainPage xaml GMS Mobile

Fig. 4.5: Home screen GMS Mobile

Fig. 4.6: ManulAdd GMS Mobile

Fig. 4.7: QR code page GMS Mobile

Fig. 4.8: Switch instance Page GMS Mobile

Fig. 4.9: Main Page GMS Mobile

Fig. 4.9.1: Main menu GMS Mobile

Fig. 4.9.2: Settings Page GMS Mobile

Fig. 4.9.3: About Page GMS Mobile

Fig. 4.9.4: Error Page GMS Mobile

Fig. 5.1: Home page GMS Mobile

Fig. 5.2: AppShell GMS Mobile

Fig. 5.3: Settings Page GMS Mobile

Fig. 5.4: Switch instance Page GMS Mobile

Fig. 5.5: Onboarding Page GMS Mobile

Fig. 5.6: QrCode Page GMS Mobile

Fig. 5.7: ManulAdd Page GMS Mobile

Fig. 5.8: App final Xamarin structure GMS Mobile

Fig. 5.9: App final solution structure GMS Mobile

Fig. 5.9.1: Translation folder GMS Mobile

Fig. 5.9.2: CZ Translation File GMS Mobile

Fig. 5.9.3: Firebase Project

Fig. 5.9.4: GMS Mobile notification keys

Fig. 5.9.5: GMS Mobile notification profille

Fig. 5.9.6: GMS Mobile profille settings