LAUREA
UNIVERSITY OF APPLIED SCIENCES
Together we are stronger

Designing and developing a CRM mobile application: a company case

Laine, Samuli

2015 Leppävaara

Laurea University of Applied Sciences
Leppävaara

Designing and developing CRM mobile application: A company case

Samuli Laine
Degree Programme in BIT
Bachelor’s Thesis
October, 2015

Laurea University of Applied Sciences
Leppävaara
Business Information Technology

Abstract

Laine, Samuli

Designing and developing CRM mobile application: A company case

Year	2015	Pages	65

Due to the rapid increase in both the number and capabilities of smart devices the overall internet usage among consumers has moved from computers to smart devices. This has created a demand for mobile optimized services. Creating a mobile optimized service or enhancing currently offered service to function with smart devices can offer an advantage over competitors. This was one of the reasons why Evolvit OY decided to improve their existing CRM service by implementing a mobile application.

The objective of this thesis was to design and document a user friendly and functional, platform independent mobile application to act as the front end to the client with the existing web service. Developing the functionalities of the application was a secondary priority. The purpose of the application was to increase the usability of the existing service by introducing it to mobile devices along with PCs. Furthermore the purpose of the application was to potentially increase sales for the company by abling it to offer an improved service.

The design and development lifecycle followed a waterfall development process with some features borrowed from agile development. Design and development tasks were split into phases that were completed by following the order dictated by the waterfall model. All the relevant framework of reference and methods used in the design and development are explained.

Key words: design, development, mobile

Laurea-ammattikorkeakoulu
Laurea
Business Information Technology

Tiivistelmä

Laine, Samuli

Designing and developing CRM mobile application: A company case

Vuosi	2015	Sivumäärä	65

Älypuhelimien ja tablettien teknologian nopea kehittyminen sekä yleistyminen on saanut internetin käyttäjät siirtymään tietokoneilta älylaitteille. Tämä on luonut kysyntää mobiililaitteille optimoiduille palveluille. Mobiili optimoitujen palveluiden kehittäminen tai nykyisten palveluiden päivittäminen tukemaan mobiilialustoja voi olla suuri kilpailullinen etu yrityksille. Tämä oli yksi pääsyistä sille, miksi Evolvit OY päätti päivittää CRM-palvelunsa tukemaan kaikkia suuria mobiilikäyttöjärjestelmiä.

Tämän opinnäytetyön päätavoitteena oli suunnitella käyttäjäystävällinen mobiilisovellus laajentamaan nykyistä web-sovellusta, sekä suunnitella ja dokumentoida toivotut toiminnot. Toissijainen tavoite opinnäytetyölle oli aloittaa sovelluksen kehitysprosessi. Luodun sovelluksen toivottiin lisäävän nykyisen palvelun käytettävyyttä mahdollistamalla käyttö mobiililaitteilla. Sovelluksen lopullinen tavoite on lisätä palvelun käyttäjämääriä parempilaatuisen palvelun ansiosta.

Suunnittelu, sekä kehitys prosessi pohjautuivat vesiputous malliin, johon on integroitu ketteriä kehitysmenetelmiä. Kaikki yksittäiset suunnittelu ja kehitys tehtävät lajiteltiin vaihaisiin vesiputousmallin mukaisesti ja suoritettiin vesiputousmallin mukaisessa järjestyksessä. Kaikki metodien ja suunnittelu ja kehitys valintojen ymmärtämiseksi tarvittava tietopohja käydään läpi opinnäytetyössä.

Sovelluksen kehitysvaihetta ei keretty suorittamaan loppuun opinnäytetyön aikana. Tämän takia sovelluksen testaamista tai julkaisemista ei ole sisällytetty tähän opinnäytetyöhön. Sovelluksen testaamattomuuden, sekä keskeneräisyyden vuoksi dataa käyttäjien palautteesta tai mahdollisista vaikutuksista solvellusken käyttäjämääriin tai vaikutusta myyntiin ei ole kerätty.

Asiasanat: kehitys, mobiili, suunnittelu,

Table of contents

1 Introduction

The rapid increase in the number of smart devices such as mobile phones and tablets has created a demand for mobile optimized services. According to a study released by comScore smart devices have surpassed desktops in popularity (Figure 1). While most of the companies currently offer a website for their customers, investing into implementing a mobile optimized service has a potential to grow their business.

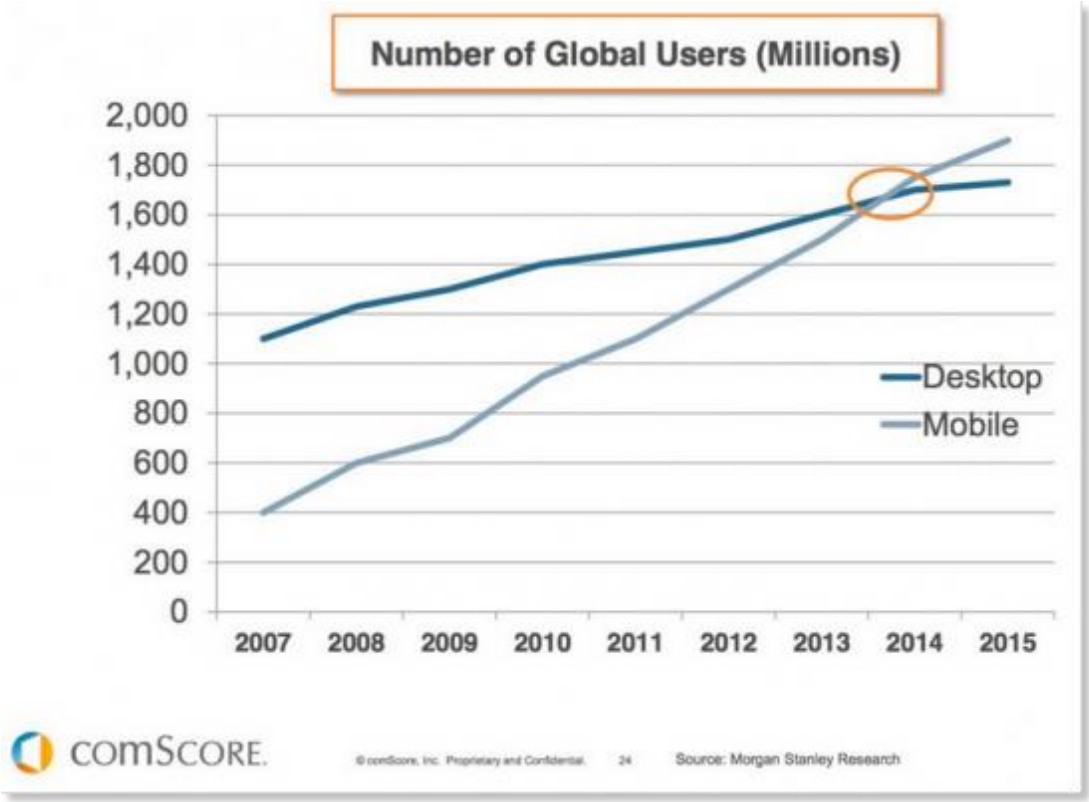


Figure 1 Mobile usage

Evolvit OY has implemented customer relation management (CRM) for web service their customers. Evolvit OY is planning to expand their existing service by implementing the CRM service as a mobile application for each of the major platforms (iOS, Android and Windows Phone 8). Evolvit OY has implemented a mobile CRM application in the past, but the usability and overall user experience has been negative. Therefore Evolvit OY wishes emphasis on user experience in the new application.

This paper will cover the CRM mobile application design and development process thorough its lifecycle. It is divided into six main sections. The first section covers the basics of the projects such as background and introduction of the client as well as scope and limitations. The second section will discuss the methodology and the research approach used in the thesis project. Third main section covers all the knowledge and concepts needed for the project. The fourth section covers the actual design and development of the CRM application. The two final sections wrap up the paper with client evaluation and conclusion.

1.1 Company background

Evolvit OY is a small Finnish company with two main business areas, financial solutions and business information solutions. The Evolvit OY was established 2007 and is located in Espoo and Tampere. Currently Evolvit OY employs 75 personnel. Estimated turnover for the year 2014 is 7M€ with EBIT (earnings before interest and taxes) of 16% (Pers. com, 2015). Evolvit OY wishes to offer a mobile application for their clients along with the existing web interface in order to offer a more comprehensive service, hopefully leading into increased sales.

1.2 Project background

Evolvit OY has developed a CRM mobile application in the past. The application had issues with both design and implementation which lead into application having performance issues as well as negative overall user experience. Due to this user experience and the application design are of utmost importance in the new application.

1.3 Objectives

The objective of the thesis is to design and begin the development process of a user friendly and well-functioning mobile application for all of the major platforms: Android, Windows Phone 8 (WP8) and iOS. According to the commisioner’s wishes out of the three major platforms, iOS was the main focus in development.

The most essential tasks were designing the application and describing and documenting functionality of all the main features. Application development tasks were secondary priorities.

Application was to be designed and developed according to the mobile development best practices and the data gathered research methods such as wireframing and benchmarking. Additionally wishes of the commisioner were taken into account unless the wishes collided with either best practices or the research data.

1.4 Project scope and limitations

The paper concentrates mainly on the design process. Development process is included up to a point it reached at the beginning of writing the paper. Due to development being uncomplete at the time any effects to number of users, user feedback or effect on Evolvit OY's sales cannot be confirmed.

The mobile application is designed and developed exclusively for the Evolvit OY's existing CRM web interface. Requirements or the design and development priorities may not be similar for each CRM application.

In the development tasks all the Microsoft technologies were preferred by the commissioner which affected / limited choosing of the development languages and frameworks.

2 Methodology

The research for design and development is done mostly by using qualitative methods. The research methods include structured interviews with the commisioner for gathering requirement and feedback, benchmarking and literature reviews. Additionally due to high amount of visual and functional design, "wireframing" can be counted as an additional method. Data gathered with aforementioned methods is analyzed and results taken into account in designing and developing the application, resulting into a better product.

2.1 Research approach

The research approach for this thesis project was constructive research approach. Constructive research approach focuses into producing new artefacts, such as diagrams or plans, or in case of this project wireframes and documents describing functionalities inside the application, and thus bring new knowledge (Metodix, n.d.). This research was chosen because the goal of application design and development is solving existing problem faced in real world, a goal that is shared with constructive research approach.

Data created by using constructive research approach was analyzed and used in the design and development of the application.

2.2 Development process

Majority of the software development follow either Agile or Waterfall development methodologies. A development methodology is a framework for development work, which an engineering team follows in order to build a given product (MaRS, 2009).

For this thesis project I choose the waterfall model as the base development methodology because of its simplicity. In waterfall model development is split into steps and the steps are executed in order. Figure 2 Waterfall modelvisualizes general steps in waterfall methodology. However, traditional waterfall model alone is very rigid because each step has to be completed from start to finish before moving onto the next one, and cannot be revisited later on (Westfall, 2008).

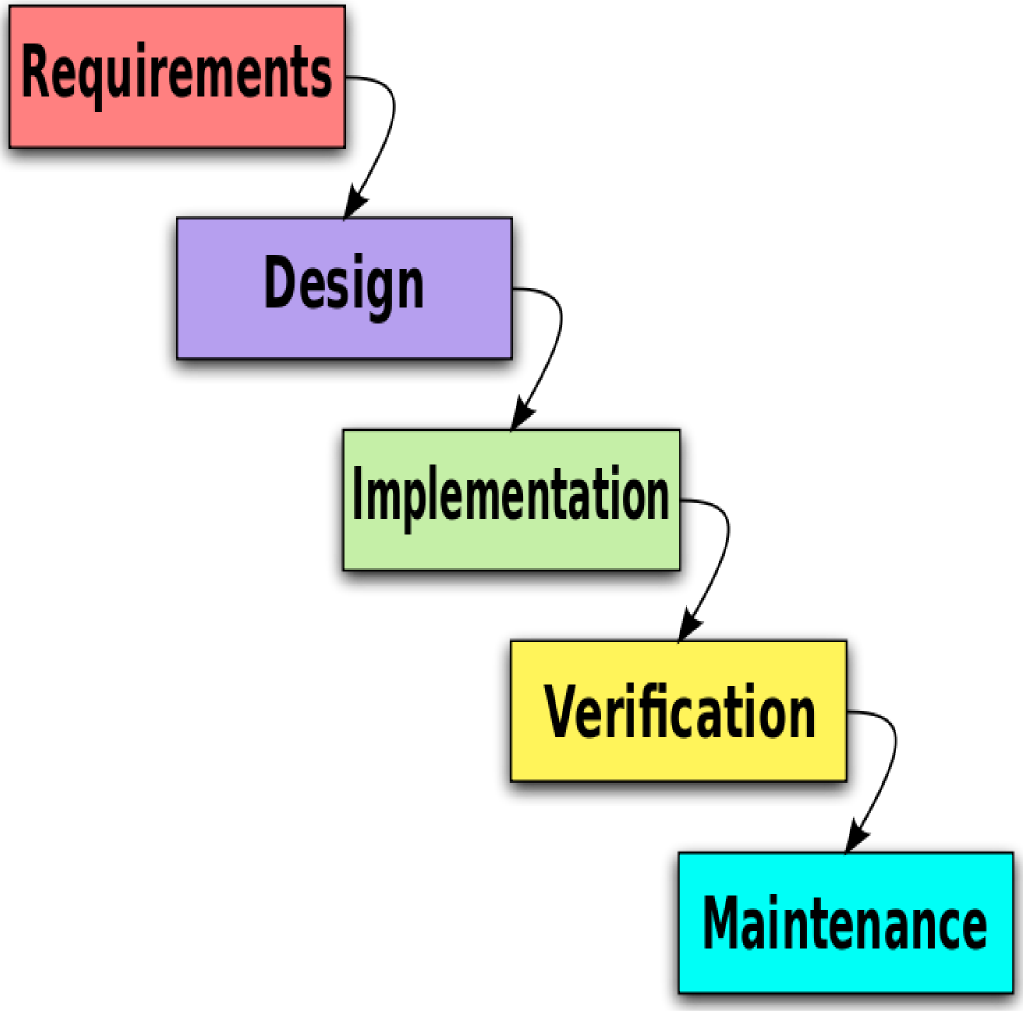


Figure 2 Waterfall model

Because of the limitations waterfall models sets I chose to combine it with selected features from agile development methodology. In agile development methodology all required tasks are listed and executed in order of importance (MaRS, 2009). In agile development priority tasks can be changed in fast pace if necessary, making the development process flexible.

In the end combining these two methodologies together created a clear and easy to execute development lifecycle for the application. This approach made it possible to go back into already completed tasks in case new data was acquired through research methods or customer feedback.

2.3 Research methods

During the thesis project various different research methods were used in order to collect and analyze data related to design and development tasks. As briefly mentioned research methods included structured interviews, wireframing, benchmarking and literature reviews.

Structured interviews were conducted at the beginning and at the end of the project while non-structured email conversations were held continuously thorough the project lifecycle. First structured interview was used to gain fundamental understanding of the project goals, the commissioner’s wishes as well as indicate the priority tasks and agree upon project timetable. Second structured interview was used to gather feedback and evaluation from the customer at the end of the project, and discuss about the future plans considering the application. Frequent unstructured email conversations were used as a more casual and convenient way to gather commissioner’s approval or feedback on everyday tasks instead of arranging a structured meeting.

For one of the requirements from commisioner was to do the designing and development according to the relevant best practices I had to study and analyze application development books and websites. Data from the other research methods, mainly wireframing and benchmarking was analyzed and used to justify design and development choices made during the project. Generally each step of development process included data gathering and analyzing.

Research method wireframing is a process of creating visual presentations of the application. “Wireframes are simple block diagrams that show the placement of elements in a user interface and demonstrate the intended layout and functionality of a solution“(Wulf, 2012). Wireframes are mostly created only at the beginning of the development process as a way to pitch designs and gather feedback from the stakeholders early on during the project. Generally wireframes display user interface (UI) elements and how they are positioned and/or will function. Figure 3 visualizes a mobile application wireframe. Various wireframes were created during the project which were analyzed and sent to the commisioner to see which he found most appealing.

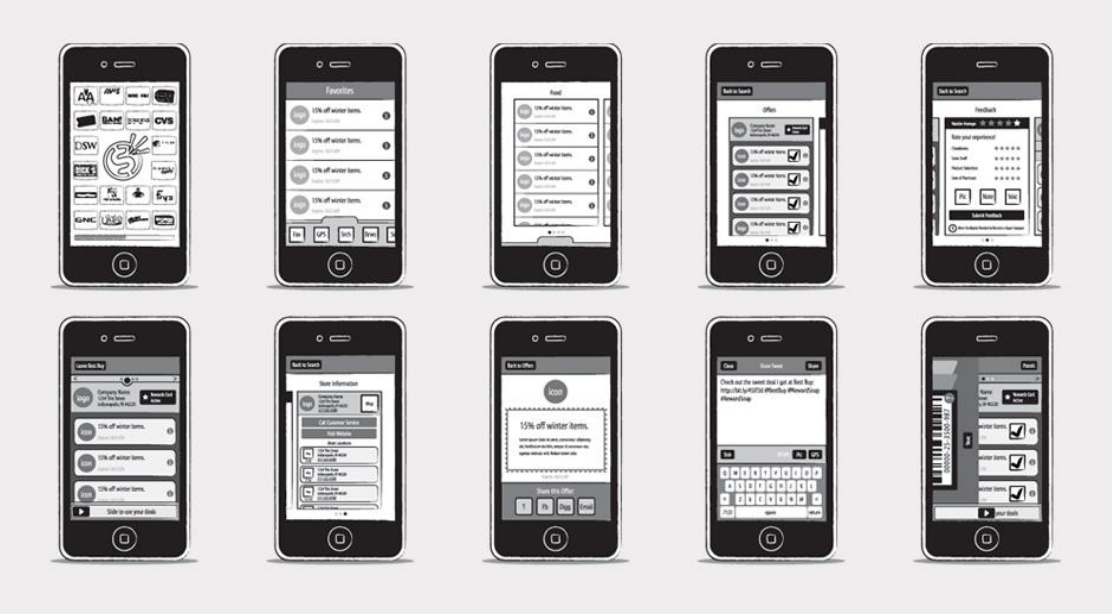


Figure 3 Wireframe

Benchmarking is a research methods that allows comparing your service with existing similar service that is already considered to be great. The purpose of benchmarking study is compare products for sake of learning for achieving improvements (Andersen, 1996). Benchmarking is done in order to indicate what makes a similar service great and whether it can be implemented in your service. The commisioner provided me with several application with appealing designs that I could take a look into. In addition I choose 3 of the highest ranked and most downloaded CRM applications from the google play store as comparison. Benchmarking was used to compare the selected applications to see how the similar features were implemented and how they were visually presented. Based on the comparison a list of features was created.

3 Knowledge base and theoretical foundation

In this section concepts that are necessary to understand the project are explained. Firstly supporting concepts such as what is CRM are explained followed with more technical concepts. The section is split into following three subsections: general, design related and development related.

3.1 CRM

Customer Relationship Management (CRM) is a strategic approach to managing customer relationships resulting to increased shareholder value. CRM is generally associated with applying business information solutions, such as this CRM mobile application, to implement or improve relationship marketing strategies.

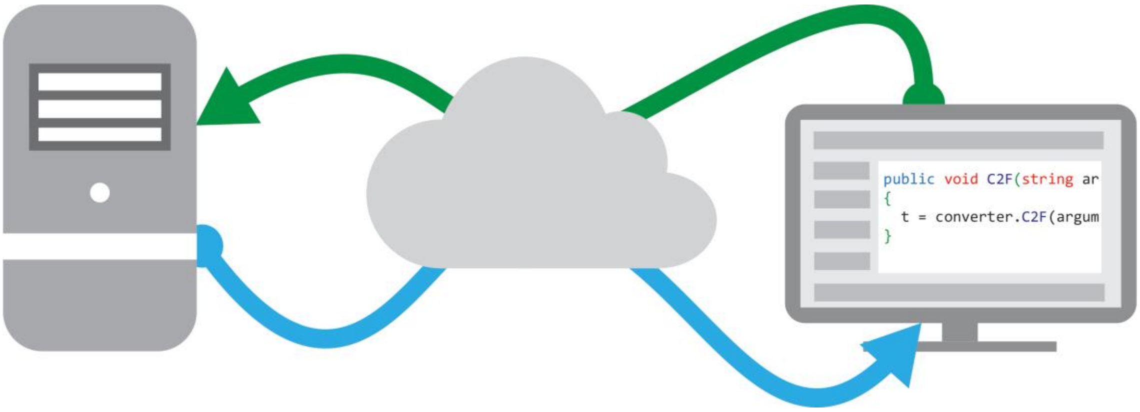
Increasing power and reduced cost of technology has made it possible for companies to concentrate on creating and maintaining long-term relations between customers and the service provider. (Payne, 2012). CRM is a major element for company strategy because improved technology makes customer-focused marketing possible. “New technologies now enable companies to target chosen market segments, micro-segments or individual customers more precisely “ making companies marketing more efficient (Payne, 2012). Targeted marketing CRM allows has a potential for increasing the number and profitability of the transactions between consumer and service provider.

3.2 Web services

Because of the nature of this project the need to understand web services is paramount. The commisioner has existing web service, on top of which they wished to expand by implementing mobile application.

Web service is a piece of software which encapsulates business logic so that it can be reused or utilized by users or other connected software. Web services range from major services such as CRM or data storage management to a minor services such as displaying weather or a bus schedule on the mobile phone (Rouse, 2007).

Web services communicate with client applications and other services through a stack of internet standards such as The Hypertext Transfer Protocol (HTTP), Web Service Definition Language (WSDL) and Simple Object Access Protocol (SOAP) (Baresi, 2007). Implementing web services allows companies to offer various and customizable services for their clients by combining elements from different services together. Figure 4 illustrates common client – service connection where mobile client can either push data to the service or pull data from the service using internet protocols.



```
public void C2F(string ar
{
  t = converter.C2F(argum
}
```

Figure 4 Web service

In this project the application works as a front-end client which is pulling and pushing data from the existing CRM service using WSDL protocol.

3.3 UX

In this paper user experience (UX) design was not a priority task. However in order to create a functional application that is user friendly at the same time understanding basics of UX and UX design was necessary.
User experience (UX) focuses on obtaining a deep understanding of services target audience, what aspects they need and what they do or don’t value in a particular product, system or service. The essence of UX design is ensuring that users find value in the service they are using (Usablitity.Gov, n.d). User experience can help differentiate components within a service that are appealing to users from the components that need improvement and reasons behind both.

In software development user experience is often confused with or perceived meaning the same as user interface (UI). However, as Figure 5 illustrates various different aspects, other than visual design, of a service can have an effect on user experience.

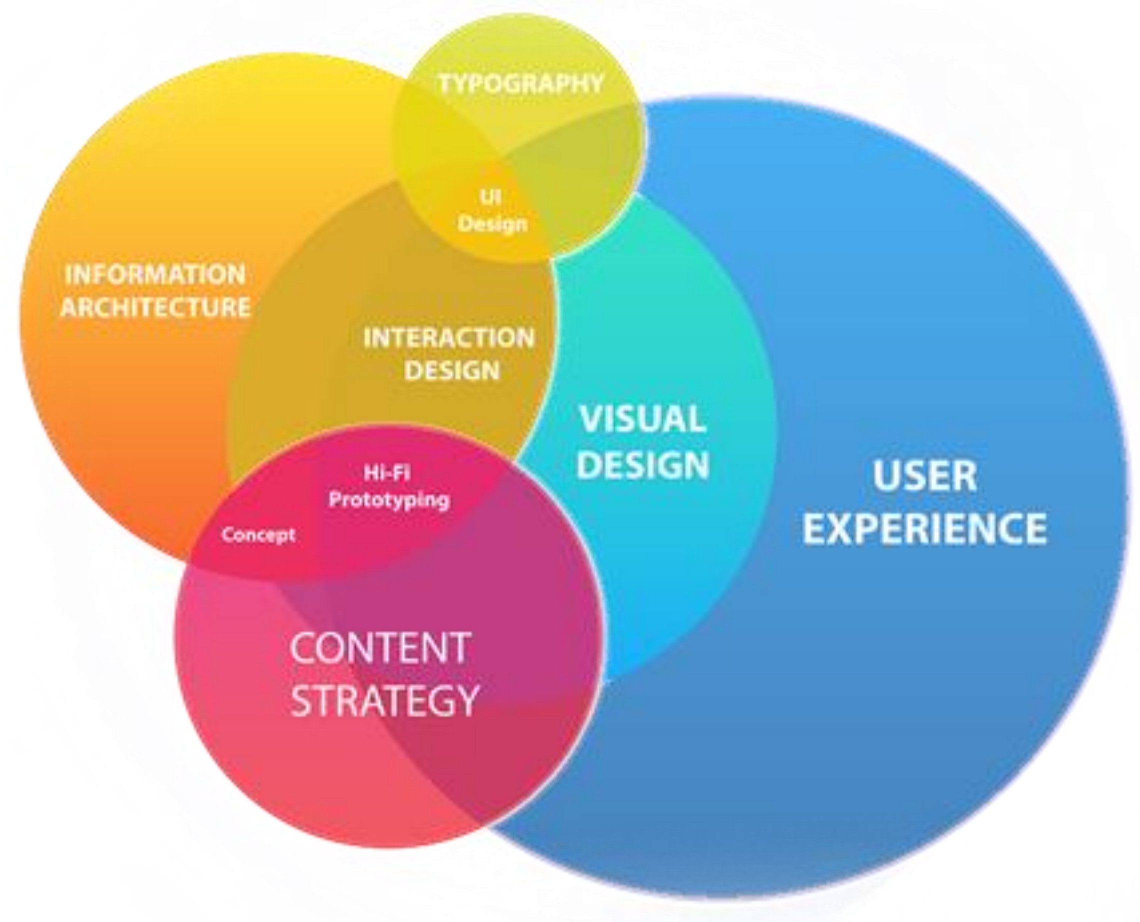


Figure 5 User experience

3.4 Design related concepts

This section covers design related concepts that are directly tied into this paper. This is to give reader deeper understanding of the concepts before they are repeatedly used later on in the following sections. In addition to the ones listed below application designing holds numerous other concepts, irrelevant to this paper. While wireframing and benchmarking can be counted as design related concepts they were explained earlier and therefore excluded from this section.

3.4.1 Functionality design

Functionality design is a process of designing software, or any service components in a way they accomplish a task they are intended to perform in the best way available (Wax, 2008). Especially in the software development there are multiple different ways to reach desired results, which makes choosing the appropriate one even more essential. The designer has to consider various elements such as target audience, how the service is used by users and over all purpose of the service.

3.4.2 Functional specification

A functional specification is a document which is created in order to capture product's intended capabilities, appearance, dependencies, and interactions with users and describe them in detail. The document does not go deep into how the software is implemented but focuses on describing the software solely from users' point of view (Dooley, 2011). Functional specification document is created at the beginning of the software development process and is used as a reference point by developers during development.

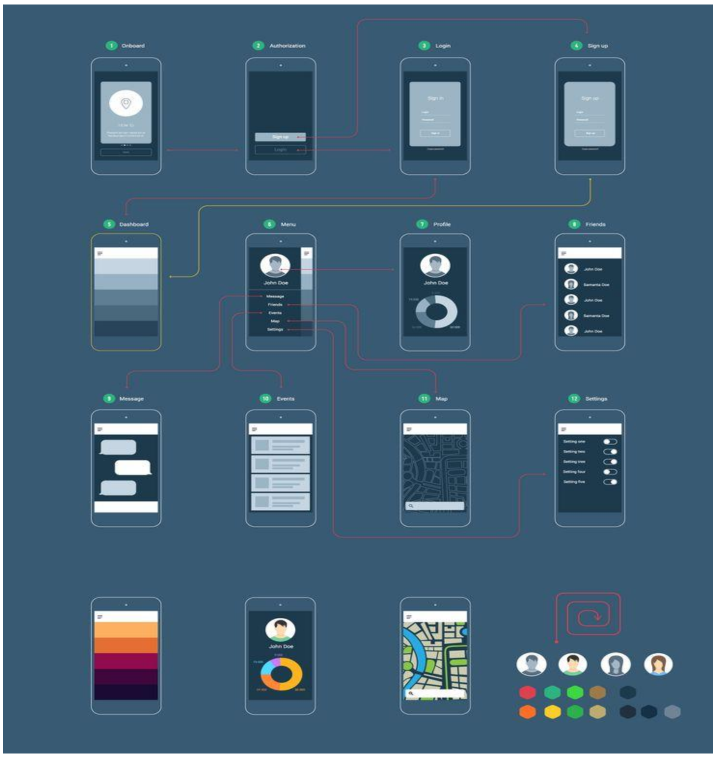


Figure 6 Functional specification

Typically functional specification document is a collection of interactive screens and dialogs (Figure 6). These would demonstrate the visual appearance of the user interface and describe each of the possible user input actions and the program response actions (Rouse, 2007).

3.4.3 User interface design

User interface (UI) designing is a process of creating user interfaces for software, such websites and mobile applications with the focus on maximizing the user experience. Goal of UI design is ensuring that the interface has elements that are easy to access, understand, and use to accomplish a task they are intended to perform. UI combines concepts from interaction and visual design, as well as information architecture (Usability.Gov, n.d.).

Even without introducing extra guidelines for user interface design it has generally a lot of problems which require finding a right balance. Typical problems encountered could for example be creating UI to be both easy to use and multifunctional, or lightweight but contains huge amounts of data (Johnson, 2010). Optimal user interfaces are consistent in their styles and simple, making it user friendly (Usability.Gov, n.d.).

3.4.4 Customer journey

Customer journeys are a service design tool. Customer journeys are created from user's perspective to provide a concise high-level overview of all the factors that have an effect on user experience. Customer journey is a structured visualization of service user's experience. Customer journey maps all the touchpoints (points where user communicates with the service) (Figure 7) detailing their interactions (Stickdorn, 2012).

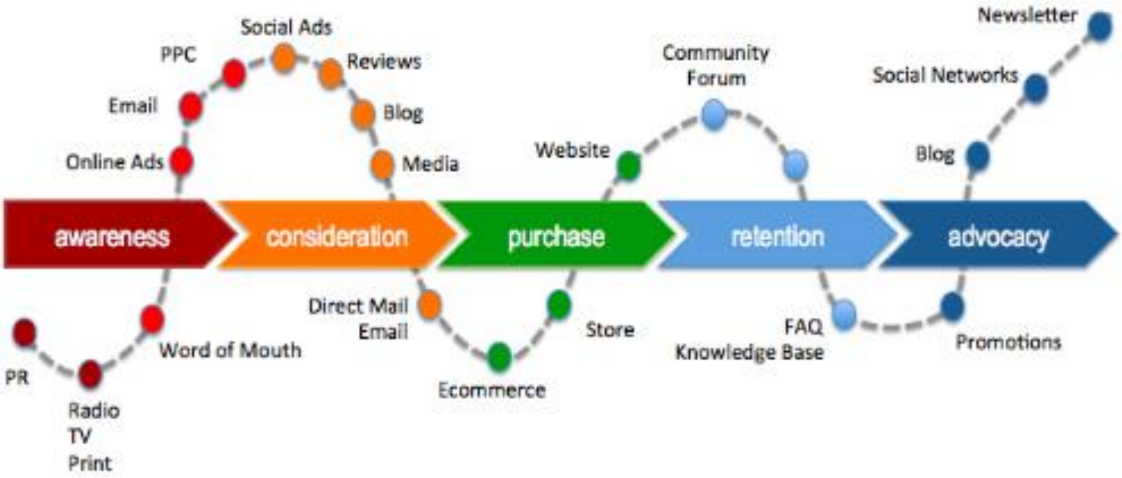


Figure 7 Customer journey

Identifying the touchpoints is crucial for customer journey. Touchpoints can be for example human interactions, or pressing a button inside an application. Touchpoints in this project consist mostly out of the latter example. It is essential to visualize different user types going through the customer journey and identify how their experience differs in order to get more comprehensive understanding of user's experience (Stickdorn, 2012).

3.5 Development related concepts

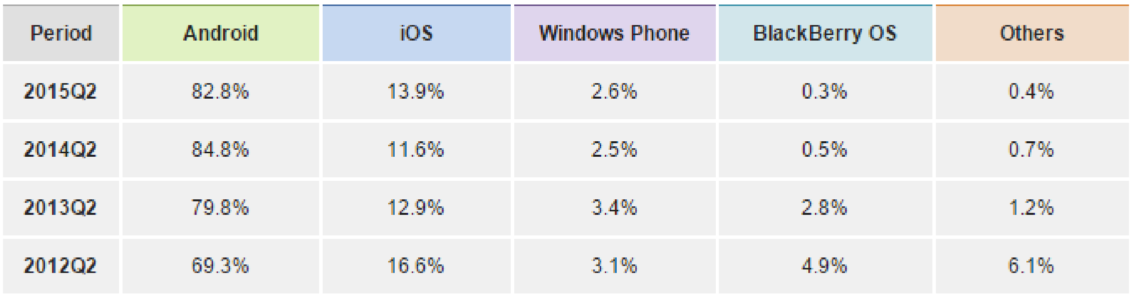
The following section covers software development related concepts in orders. This is to give reader deeper understanding of the concepts before they are repeatedly used later on in the following sections. Software development is a wide topic that includes for example multiple different programming languages, various tools and frameworks and back-ends (such as servers and web services). The following sections only covers the concepts directly related to this paper.

3.5.1 Software development process

Software development process, also known as software development life cycle is a division of software development into phases. Each software no matter of the extent or the size of a development team has a lifecycle. Generally software development projects fall into either traditional plan-driven models, such as waterfall model or newer agile development models (Dooley, 2011). Plan-driven development models are strict and follow the lifecycle phase by phase in order. Agile development is more flexible model allowing tasks to be completed in order of importance or multiple tasks at the same time. Traditionally plan-driven approach requires a larger amount of documentation making it less common with small-scale projects (Dooley, 2011).

3.5.2 Mobile platforms

Out of all the mobile operating systems (OS) three can be picked up as major platforms. In year 2015 Android, iOS and Windows Phone devices shared over 99% of user base (Table 1). While Android is by far the most popular platform amongst the users with over 82% of the users, releasing applications for iOS platform remains as the most lucrative option in the eyes of companies (Richter, 2013). According to Richter platforms other than iOS and android are struggling for relevance and cannot compete for companies' interest.



Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

Source: IDC, Aug 2015

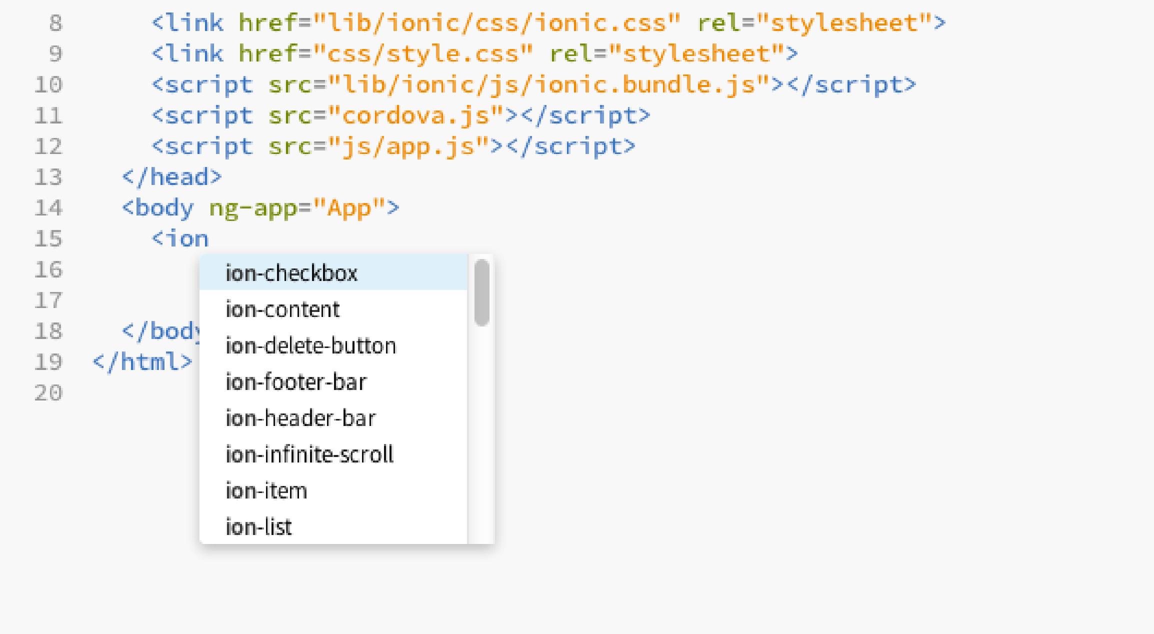
Table 1 mobile OS statistics

3.5.3 Development frameworks

A software development framework is a reusable software environment that provides particular functionality that can be selectively changed by user-written code in order to facilitate software development process. Development frameworks function as a base of a project on top of which developers start building the features of an application (Porebski et al, 2011).

Additionally software frameworks may include support programs such as code libraries, tools, and application programming interfaces (APIs).

Figure 8 demonstrates functionalities a framework can offer. In this case instead of choosing from general HTML5 elements, framework offers a variety of predefined and styled elements from which programmers can choose and therefore making the development faster and the components more consistent.

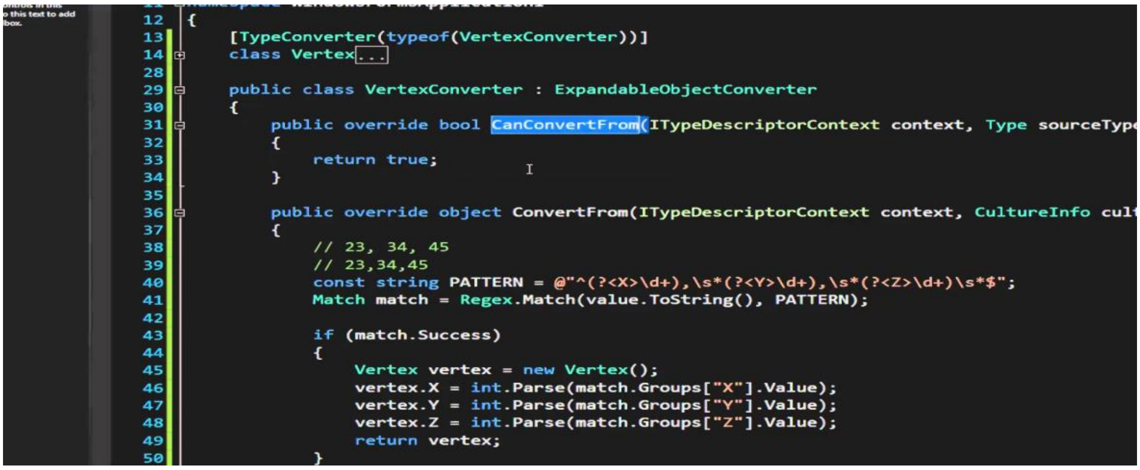


```
    <link href="lib/ionic/css/ionic.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">
    <script src="lib/ionic/js/ionic.bundle.js"></script>
    <script src="cordova.js"></script>
    <script src="js/app.js"></script>
  </head>
  <body ng-app="App">
    <ion


  </body
</html>

```

Figure 8 Framework

3.5.4 C#

C# (C sharp) is a modern object-oriented programming language developed by Microsoft. C# incorporates elements from different modern programming languages such as C, pascal and Java (Petzold, 2010). C# is mainly used to implement the functionality of the software (Figure 9). C# can also be used to create UI elements up to a point, however markup languages such as HTML5 and XAML are more efficient for that aim. Because C# is developed by Microsoft it is designed to work together with other Microsoft technologies such as .NET and has spread widely among Microsoft-oriented companies, organizations and individual developers (Nakov, S.2013).



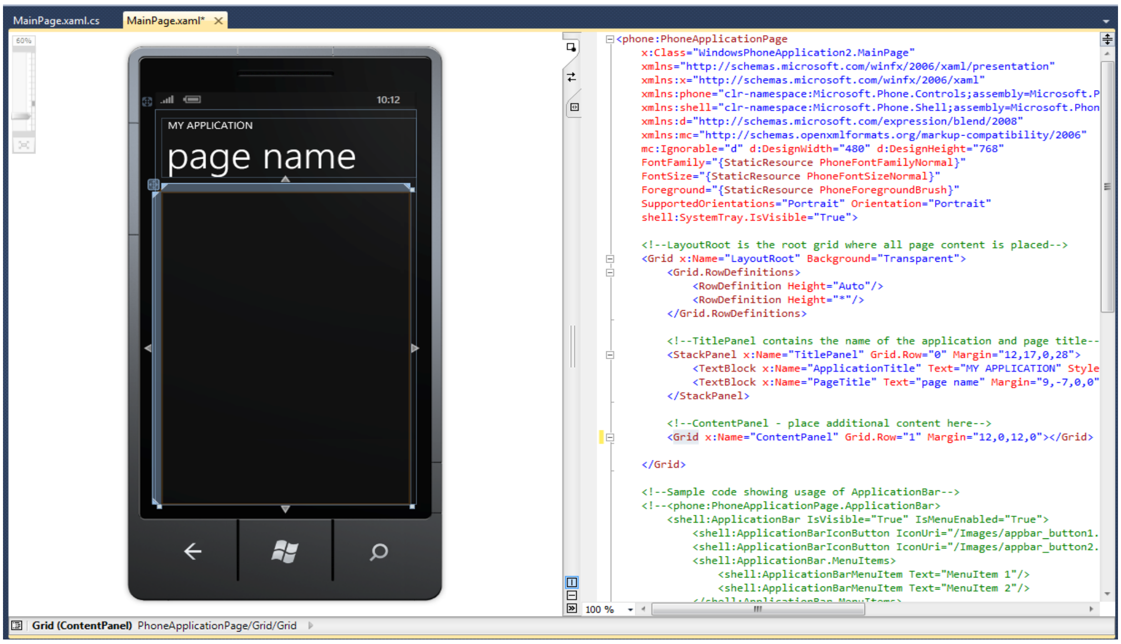
```
{
    [TypeConverter(typeof(VertexConverter))]
    class Vertex...

    public class VertexConverter : ExpandableObjectConverter
    {
        public override bool CanConvertFrom(ITypeDescriptorContext context, Type sourceTyp
        {
            return true;
        }

        public override object ConvertFrom(ITypeDescriptorContext context, CultureInfo cul
        {
            // 23, 34, 45
            // 23,34,45
            const string PATTERN = @"^(?<X>\d+),\s*(?<Y>\d+),\s*(?<Z>\d+)\s*$";
            Match match = Regex.Match(value.ToString(), PATTERN);

            if (match.Success)
            {
                Vertex vertex = new Vertex();
                vertex.X = int.Parse(match.Groups["X"].Value);
                vertex.Y = int.Parse(match.Groups["Y"].Value);
                vertex.Z = int.Parse(match.Groups["Z"].Value);
                return vertex;
            }
```

Figure 9 C# script

In this project C# was selected as the main programming language due to commisioner's wishes and the fact that the selected development framework runs with C# code base. Additionally C# allows to connect application with .dll (dynamic link library) file, which was essentially necessary for this thesis.

3.5.5 XAML

Extensible Application Markup Language (XAML) is a declarative markup language developed by Microsoft. XAML and markup languages in general are the programming languages defining the visual presentation (UI elements) of an application (Microsoft, n.d). Figure 10 visualizes a simple XAML script for one page application.



```
<phone:PhoneApplicationPage
    x:Class="WindowsPhoneApplication2.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.P
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phon
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="768"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
    shell:SystemTray.IsVisible="True">

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>

        <!--TitlePanel contains the name of the application and page title--
        <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
            <TextBlock x:Name="ApplicationTitle" Text="MY APPLICATION" Style
            <TextBlock x:Name="PageTitle" Text="page name" Margin="9,-7,0,0"
        </StackPanel>

        <!--ContentPanel - place additional content here-->
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0"></Grid>

    </Grid>

    <!--Sample code showing usage of ApplicationBar-->
    <!--<phone:PhoneApplicationPage.ApplicationBar>
        <shell:ApplicationBar IsVisible="True" IsMenuEnabled="True">
            <shell:ApplicationBarIconButton IconUri="/Images/appbar_button1.
            <shell:ApplicationBarIconButton IconUri="/Images/appbar_button2.
            <shell:ApplicationBar.MenuItems>
                <shell:ApplicationBarMenuItem Text="MenuItem 1"/>
                <shell:ApplicationBarMenuItem Text="MenuItem 2"/>
```

Figure 10 Xaml

Depending on the size and complexity of a projects the nature of XAML may wary. Simple, or small scale application may benefit out of having a single master XAML script containing all the elements of the application. However once the size of projects grow having multiple XAML scripts, each containing a small portion of the UI elements can prove to be better for performance and consistency in the components. As example of the latter one script could control button elements, another one text input elements and third one navigation elements such as side menus or header bars. In this project XAML was chosen due to commisioner's wishes and the framework requirements.

3.5.6 WSDL

Web Service Definition Language (WSDL) is XML format used to define web services in a standard way. WSDL elements contain a description of the data and operations that are to be performed on the data, typically using one or more XML schemas, to be passed to the Web service so that both the sender and the receiver understand the data being exchanged (Newman, 2002). In order for web service and client (such as mobile application) to be able to communicate with one another they both need to have an access to same WSDL files in order to be able to understand each other (Figure 11).

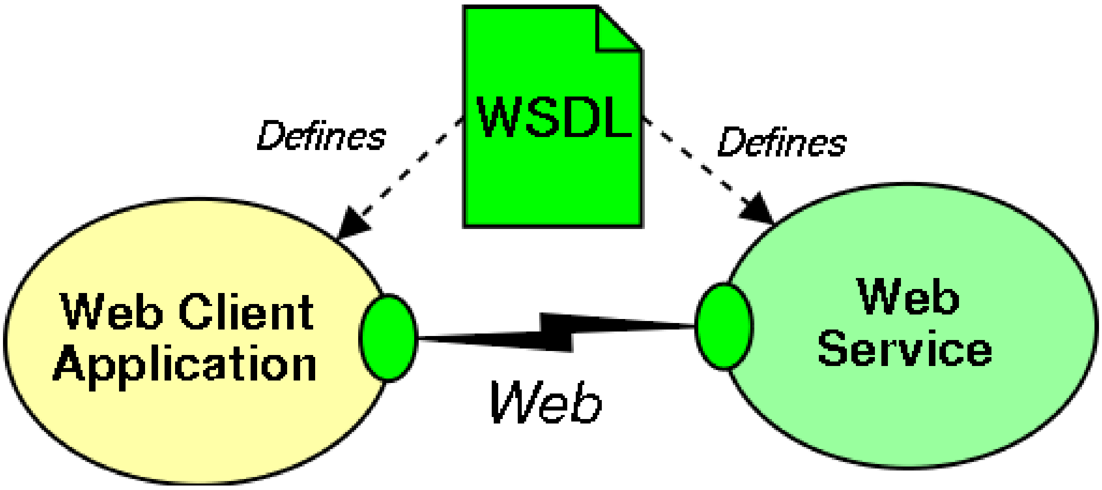


Figure 11 WSDL

WSDL was developed by Microsoft, Ariba and IBM. Ever since being accepted by The World Wide Web Consortium (W3C) WSDL has had a broad support and has been one of the most common web-service related technologies (Newman, 2002).

4 Practical implementation

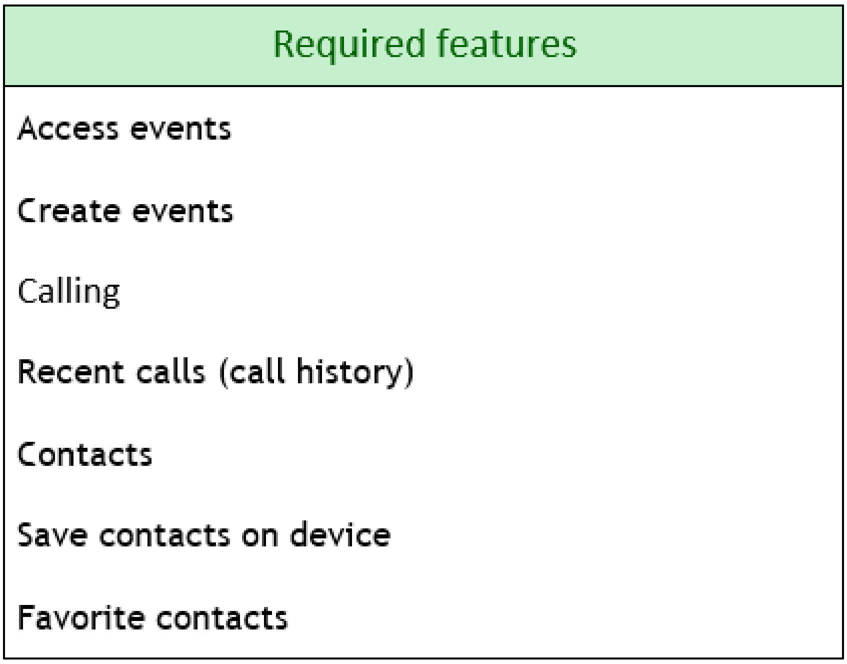
The following section will cover the designing and development of the CRM application. I will explain the course of the project in chronological order following software development lifecycle, starting from planning up to the implementation. Traditionally the final step in software development lifecycle would be testing, however the development was not finished during the project and thusly any large scale testing was not performed.

This thesis project was started as my second internship as part of my bachelor's degree. Due to this majority of design and development tasks were concentrated between January and May of 2015. Some development tasks as well as documentation of the project, namely this paper, was done during the summer and autumn of 2015.

4.1 Gathering customer requirements

Thesis project started by having a kickoff meeting with the commisioner at the beginning of year 2015. The main purpose of this meeting was to get an overview of project goals, familiarize myself with existing service and gather the customer requirements for the application. Customer requirements were gathered by interviewing Evolvit OY representative who would oversee the thesis project. The interview can be found in Appendix 1. In addition to the required features other focus areas were discussed in the meeting such as priority platforms and main target audience of the application. After the meeting as well as thorough the project more information about requirements was gathered using email conversations. According to the commisioner the main purpose of the application was to improve the existing service by offering clients a mobile application alongside the web service (pers.com, 2015).

The application was to be designed and developed according to development best practices. Using Microsoft technologies was desired by the commisioner due to his and his colleagues' prior expertise with technologies Microsoft such as C# making the maintenance of the application or further development easy. To this goal developing according to the best practices was essential. Implementing the application based on modules (individual features as separate 'blocks') makes introducing new features as well as troubleshooting and overall the currently existing ones more convenient.



Required features
Access events
Create events
Calling
Recent calls (call history)
Contacts
Save contacts on device
Favorite contacts

Table 2 Customer requirements

The table above (Table 2) contains the list of priority features for the application (Pers.com, 2015). Mostly desired features are fairly simple, concentrating around manipulating data pulled from web service and pushing it back. Out of the main functions only calling function calls hardware (phone or tablet) functionalities. Naturally ‘saving contacts on device’ function uses devices storage as well as existing call list service. Additional features would be added or changes to the current priority features would be made during the development in case there is time or extra resources to spend.

4.2 Research

Research was an essential element of each step within the project lifecycle. Even though I had prior knowledge and experience with the methodologies and technologies used in the project additional research was required.

At the beginning of the project research tasks consisted out getting an overview of the methodologies and selecting the relevant ones to be used further along the project lifecycle. After choosing the waterfall model as development method I had to research the strong points and weaknesses of the model and how to overcome them by combining it with features from agile development. Before starting the planning of the project research consisted out of the flow of software development process, project management and case studies of similar application development projects.

During the designing and development I carried out task related research before starting each individual task. Both design and development tasks were to be performed according to the relevant best practices, making indicating and selecting them an essential research tasks. According the customer requirements the essence of the application design was user friendliness. Therefore I had to research all the individual tasks such as benchmarking from that perspective.

During development I had to compare the required features with the inbuilt features the selected framework offered. If the required functionality was included into the framework I had to look into documentation of the framework and find a way to implement the feature into the application. However if the feature was included into the framework I had to research an alternative way to achieve the sought feature and incorporate it into the application.

Even though any large scale testing was not performed during the project I still had to do some prior research preparing for it. Testing was supposed to be carried out in two phases before the launch. Firstly as closed beta testing with a small selected group of users from the client company. In second stage the target user group was to be expanded to cover some of the intended customers. Feedback from the testing was to be analyzed and used to fix existing problems and improve the application further.

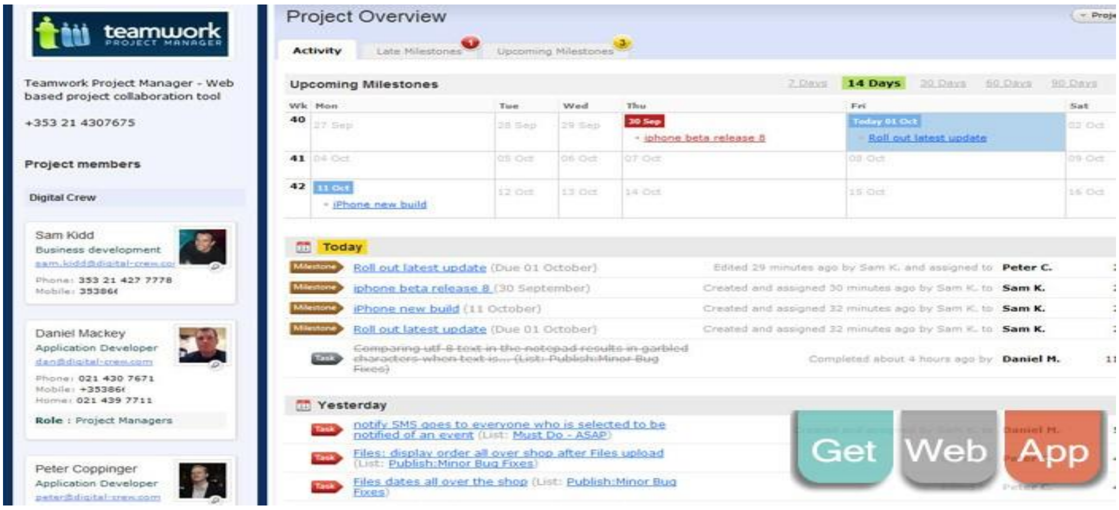
4.3 Planning

After gathering the requirements and project goals the next step of the project was to create a project plan. Project plan was created as a way to better visualize the flow of the project as well as documenting each stage and keeping track of tasks and deadlines. Secondary benefit from creating a project plan was to give the commisioner an idea about the vision I had for the project as well as the timetable I plan to advance. Table 3 visualizes the initial timetable for this thesis project. Tasks are planned to be executed in chronological order, loosely following waterfall model which was chosen as main development methodology.

Month	January				February				March				April				May			
Week	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Initiation	🟧	🟧																		
Planning		🟧	🟧																	
Research	🟧	🟧			🟧	🟧		🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧		🟧	🟧		
Choosing a framework			🟧																	
functionality design			🟧	🟧																
Design:																				
Benchmarking					🟧	🟧														
Wireframing					🟧	🟧														
UX design						🟧	🟧													
functional specification							🟧													
Development:																				
Login								🟧	🟧											
Developing layouts									🟧	🟧	🟧									
Navigation									🟧	🟧	🟧									
Lists											🟧	🟧	🟧							
Calling funtion													🟧							
Events													🟧	🟧	🟧					
Contacts															🟧	🟧				
Save contacts															🟧	🟧				
Testing																				
Closed launch																	🟧	🟧	🟧	
"Friend launch"																				🟧
Public launch																				

Table 3 Project timetable

For managing the project I used online project management tool TeamworkPM (teamwork.com). TeamworkPM offers a clear web interface for managing a project (Figure 12). With TeamworkPM I could create and manage tasks and milestones easily and display them in clear format. Using an online tool for project management made it possible to share the environment with the commisioner as well as thesis supervisor making it more convenient for them to follow the process.



Teamwork Project Manager - Web based project collaboration tool

+353 21 4307675

Project members

Digital Crew

Sam Kidd
Business development
sam.kidd@digital-crew.co
Phone: 353 21 427 7778
Mobile: 35386

Daniel Mackey
Application Developer
dan@digital-crew.com
Phone: 021 430 7671
Mobile: +35386
Home: 021 439 7711
Role : Project Managers

Peter Coppinger
Application Developer
peter@digital-crew.com

Upcoming Milestones

Wk	Mon	Tue	Wed	Thu	Fri	Sat
40	27 Sep	28 Sep	29 Sep	30 Sep • iphone beta release 8	Today 01 Oct • Roll out latest update	02 Oct
41	04 Oct	05 Oct	06 Oct	07 Oct	08 Oct	09 Oct
42	11 Oct • iPhone new build	12 Oct	13 Oct	14 Oct	15 Oct	16 Oct

Today

Milestone Roll out latest update (Due 01 October) Edited 29 minutes ago by Sam K. and assigned to Peter C.

Milestone iphone beta release 8 (30 September) Created and assigned 30 minutes ago by Sam K. to Sam K.

Milestone iPhone new build (11 October) Created and assigned 32 minutes ago by Sam K. to Sam K.

Milestone Roll out latest update (Due 01 October) Created and assigned 32 minutes ago by Sam K. to Sam K.

Task ~~Comparing utf-8 text in the notepad results in garbled characters when text is... (List: Publish:Minor Bug Fixes)~~ Completed about 4 hours ago by Daniel M.

Yesterday

Task notify SMS goes to everyone who is selected to be notified of an event (List: Must Do - ASAP) Daniel M.

Task Files: display order all over shop after Files upload (List: Publish:Minor Bug Fixes)

Task Files dates all over the shop (List: Publish:Minor Bug Fixes)

Figure 12 TeamworkPM

Following subsections cover other elements relevant in planning and initiating the project before beginning with the design and development tasks.

4.3.1 Choosing a development framework

As stated in the previous section a framework is a base of a project on top of which developers start building the features of an application. Because of this fact selecting the correct framework for the development was essential. Generally there are numerous of solid and well-functioning frameworks available for mobile application development. However because the customer wanted the application to be platform independent (available for all the major platforms) and to be developed using Microsoft technologies (Pers.com, 2015) available frameworks were filtered down into a few.

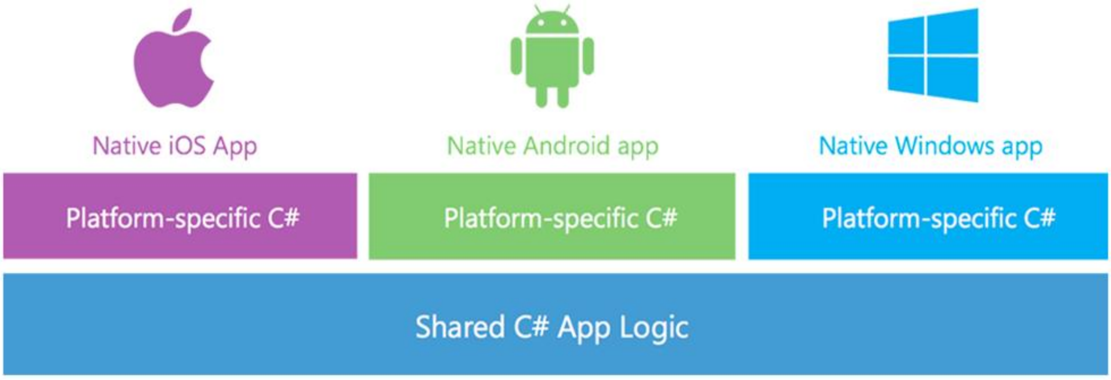


Figure 13 Xamarin

Out of the remaining frameworks I selected Xamarin. Xamarin is a powerful framework built on top of old monodevelop framework. Xamarin allows applications to share a singular C# logic (Figure 13). Generally this covers 70-95% of the logic of the application On top of the shared codebase developers can platform specific C# scripts. This allows Xamarin an access to multiple platform-specific plugins and gives developer an opportunity to incorporate native iOS, Windows or Android features in to the application enabling native-like performance. For example native features could include platform specific UI elements or page transition (navigation) animations.

On the negative side Xamarin is a fairly new framework. This means that it does not cover the same amount of features as older, over time established and improved frameworks. Being a new framework Xamarin also does not have a comprehensive documentation, which in turn created challenges during the feature specific research.

4.3.2 Choosing a development tool

After choosing a framework I had to select a development tool. For the actual development process having a specific development tool is not required but having one that supports the framework makes the development faster and more straightforward. Development tools offer features such as code hinting and logs to help locating and fixing errors. Without purchasing either business or enterprise license to Xamarin, the only fully supported development tool was Xamarin studio and was therefore chosen.

4.4 Functionality designing

This section talks about designing and describing the required features and how they function. I am concentrating solely on the main features listed by the commisioner. This is because development is still under progress and extra features are not planned and documented properly. The complete functionality design document can be found from Appendix 3.

Before wireframing or any development tasks could be started understanding of the features had to be obtained. The functionality design included benchmarking features from similar mobile application. After studying already well established applications the process of planning out the development of features became straightforward.

I included both layout based and purely functional features into functionality design phase. Planning out how the main navigation and the major UI elements function is beneficial once the development begins. Planning out the functionality of each feature made it possible to start development related research early on in the project lifecycle. All functionality design choices were based on the data gathered from benchmarking or commissioner’s feedback.

4.5 Choosing thr design

This section covers the design phase of the project lifecycle. The design phases included benchmarking similar CRM applications. The commisioner provided ‘Sugarcrm’ as a reference point for benchmarking. Other than that I selected ‘Zoho CRM’ as an additional benchmarking target based on the reviews and ratings in respected app stores. Benchmarking was done to compare both visual design choices and implementation of features and imitate the best ones in this project. After benchmarking several wireframes were created in order to see which layout Mr. Saarikivi preferred for the application. Based on the benchmarking results decisions regarding to main navigation structure in the application as well as main layout elements were made and included into the wireframes. Once the layouts were chosen the next choice was selecting overall color scheme. Base on the customer requirements (Appendix 1) color scheme was to be similar to the one in the web service. Final step of the design was putting everything together into a visual presentation, functional specification. Functional specification was used to give Mr. Saarikivi an idea what the finished product could look like and how the UI elements function together.

4.5.1 Benchmarking

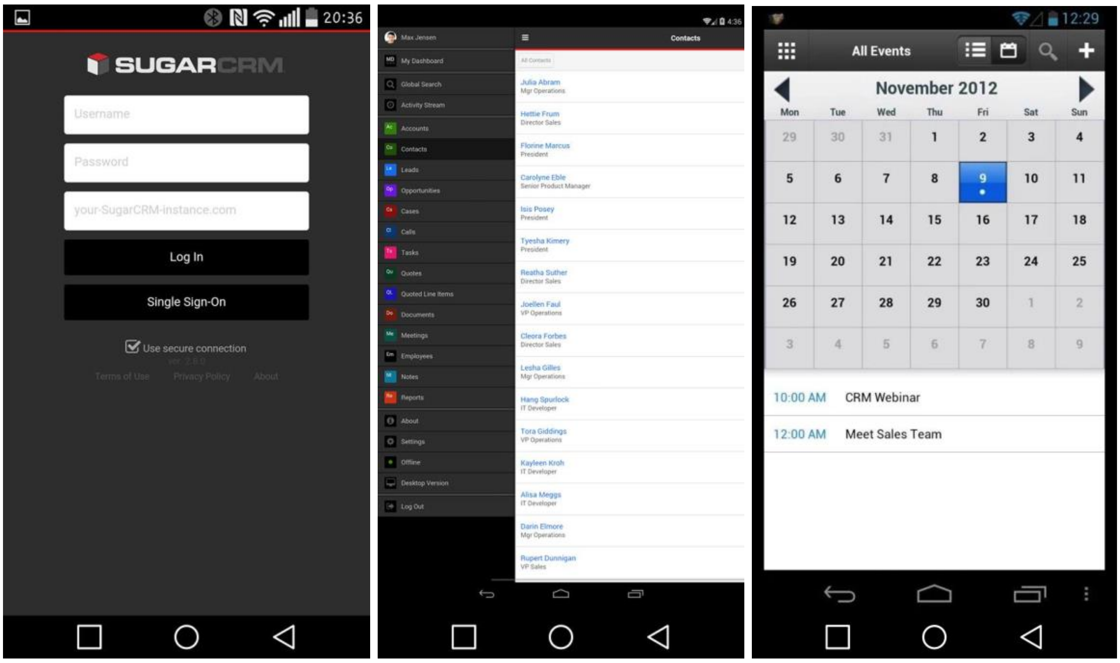
The main goal of benchmarking is to identify the commonly used and well-functioning solutions from various similar applications and incorporate them into your own product. Once benchmarking is done and desired features are found they need to be modified in order for them to fit to the theme of your software product. In this project the essence of benchmarking was to investigate if features desired by the commissioner were incorporated into other CRM applications and how they were implemented. Secondary point of interest in the benchmarking process was to see how desired features were displayed. Table 4 below presents the benchmarking results regarding to both layout elements and functional features.

Feature	Findings
Login	Only Username & password or username, password and instance reference
Main navigation	Side menu master - detail navigation
Information display elements	List with name and varying information such as position in company, image or company name.
Call	Executed from contact details, email, text messages as additional contact methods
Contactslist	List with name and position in company, can be sorted by last modified, title and name
Contact details	Different contact methods as different buttons, image, company, employer/ supervisor, activities, follow & favorite
Modify contacts	Text input fields with previous values as default values. Saving updates the values and lists updated at the same time
Favorites	Displayed in separate list
Call list	Inside contact, held and scheduled
Activities create	On separate page, different kinds of activities on different pages: meetings, tasks, campaigns etc. Different fields for different tasks

Activities view	Inside contact details, held and scheduled and on separate page
Save contacts	Not implemented or through hardware phonebook

Table 4 Benchmarking features

As briefly mentioned before, two CRM mobile applications were used as a reference point while doing the benchmarking. An application ‘Sugarcrm’ was suggested by the commissioner while the second ‘Zoho crm’ was picked based on downloads and ratings. Sample screenshots from the applications can be found from below (Figure 14 benchmarking screens). Both applications, but especially Sugarcrm had a large impact on the design of the application since it had most of the desired features incorporated in a clear and well-structured fashion. Based on the benchmarking data initial decisions regarding the layouts and presentation of features could be made. Features gathered from the benchmarking were categorized and then included into wireframes based on the referenced applications layout solutions.



Mon	Tue	Wed	Thu	Fri	Sat	Sun
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

10:00 AM CRM Webinar

12:00 AM Meet Sales Team

Figure 14 benchmarking screens

4.5.2 Wireframing

Wireframing was started on parallel to the benchmarking. Based on the customer requirements and data gathered from benchmarking several wireframes were created. Designed wireframes were presented to the commissioner in order to see which one he preferred. Master-detail layout was chosen as the main layout and navigation for the application. The layout can be seen in the Figure 15 master-detail layout below.

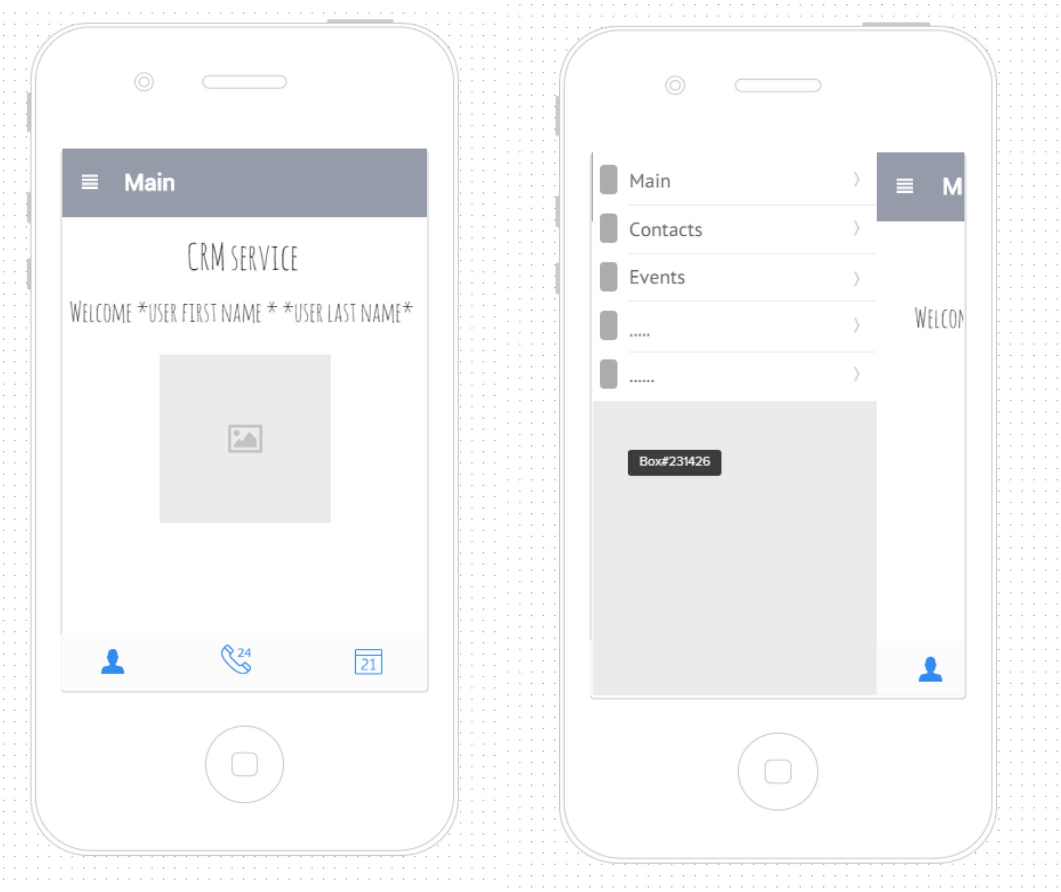


Figure 15 master-detail layout

In master-detail layout a view is divided into two main sections – master and detail. The master section of the view is generally a side menu bar with varying content. In this application side menu held the main navigation. Clicking on a list item in the side menu calls a separate script that holds UI elements and functionalities for a certain page and displays that in the detail section. For example if user would like to navigate into ‘contacts’ page they would open the side menu, tap on ‘contacts’ and contact page would be loaded into the detail view. The detail section holds the main content of the page. While master view stays constant in every screen of the application, content in the details view changes on each page. Each detail view contains individual UI elements and sub navigation elements in order to complete the function it was created for. Master page is accessed from details page by swiping right or tapping the icon top left in the header.

Master-detail was considered as the best layout solution for this project due to its user friendliness, effective display of information and the fact that the layout is easy to expand once new features are introduced.

Figure 16 below holds a wireframe of the login and sign up page. In the end the sign up section was not implemented as a primary tasks for the reason getting a session key in order to login to the application contacting the sales staff of the company was required. The login page contains three text-input fields where users set their login credentials.

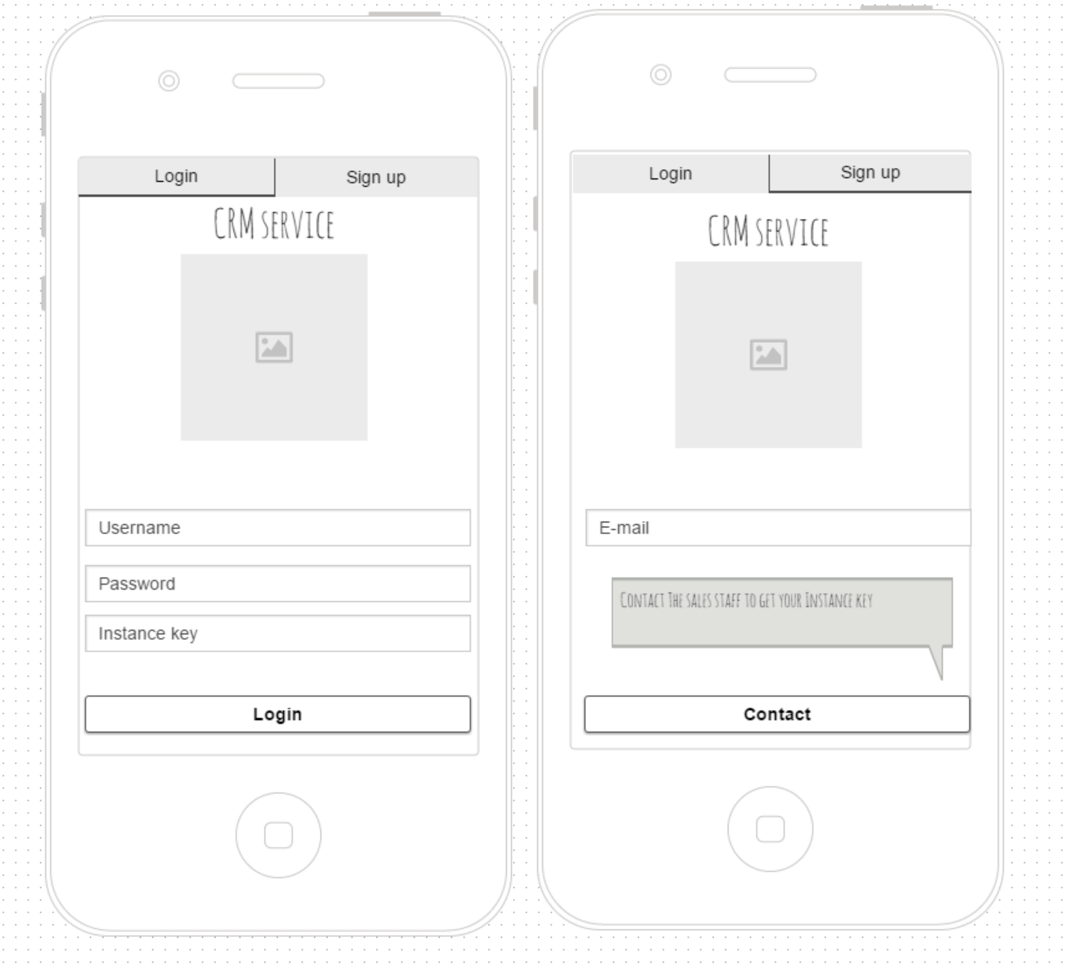


Figure 16 Start screen wireframe

Wireframes were used as guidelines during the development of the application. Final layouts differ from the created wireframes. More wireframe illustrations can be found from the Appendix 2 Wireframes

4.5.3 Choosing a color scheme

As mentioned earlier, the commissioner wished the application to use similar color scheme as the web service. During the bench marking I looked into referenced applications color scheme. General trend of color schemes was to use a one bright color combined with greyscale colors from white to black. I located the color hex values from customer's web site and displayed in a Table 5 below.

Name	Hex	Color
Grey	#7a7a7a	▨
White	#FFFFFF	⬜
Black	#000000	⬛
Green	#68ae5c	🟩
Turquoise	#339999	🟦

Table 5 Colors

Based on the table, the selected colors follow the trends found out with benchmarking fairly well.

4.5.4 Functional specification

The final stage of designing the application was creating a functional specification document. Functional specification document is a visual presentation of the application from user's point of view. In order to create an adequate presentation of the application I combined all the data gathered by previously mentioned sources: benchmarking, wireframing, color-scheme and customer requirements and created 'more specific wireframe' like document. As a prototyping tool I chose JustInMind.

Creating a functional specification document was important in this project as an emphasis was on user friendliness and user experience. Creating the document helped both me and Mr. Saarikivi to visualize the application from user's point of view and determine whether some feature is user friendly or if it still has to be improved. Functional specification document also holds the navigation structure, covering all the navigation elements used inside the application.

As the functional specification document includes data from all the other design methods it is final and the most realistic visualization of the final application design, before getting user feedback during testing. Functional specification document can be found from the Appendix 4.

4.6 Developing the features

This section talks about development phase of the lifecycle. In this thesis development tasks were of secondary importance after design related tasks. Therefore I will not go into much detail about the actual development work. This section covers the development up to a point it reached before I started documenting the process - this paper. The development is currently in progress and therefore not all required features are yet implemented.

4.6.1 Development process

The development process was planned as follows. After setting up the framework and development tool, the first tasks was to build the navigation structure and required UI elements. Second task was to connect the application with the .dll file and through that web service. After connection with web service had been formed, the features dependent on the web service were to be created. Started with login and afterwards the various lists and modifying the data. After that the features requiring hardware functionalities were to be developed and lastly all the UI elements were to be styled according to the designs.

Building the navigation structure was fairly straight forward tasks. Xamarin framework has a built in master – detail layout structure with semi-adequate documentation. Master detail page was built with C# script to function as the base of the application. Once users select a page they wish to navigate into from the side menu, based on the selection another C# script containing page specific elements was to be loaded to populate the current detail section of master – detail page.

All the page layouts were developed using C#, while XAML was later on used to give pages a style. After building the navigation structure the next task was to create the layouts of the pages. Pages were created mostly by using internal layout elements, but for such elements as the calendar additional code library was required. As mentioned earlier the main data holding element is a list view. After I created a list view it could be reused in multiple pages.

Login page was a tabbed page outside the main navigation. Once the application was launched the login page would be the first screen to be seen. Login page, as the others, is a C# page containing input fields for credentials. Additional background script is handling the data input and once user taps ‘login’ button the background script pushes data into the web service and checks if user with entered name and password exist within the defined instance. If user exists web service would send positive response and the login would be successful leading the user to main page of the application. If however the result were to be negative a popup would be launched instead to ask user to check the entered credentials.

Main page of the application is fairly simple. It contains only static information about the application and welcome message. On the bottom of the page a button bar holds quick navigation buttons that can be used instead of the side menu.

Contact list page is a tabbed page. Tabs can be switched between all contacts and favorite contacts. All contacts page displays all the contacts while favorite contacts displays only the contacts the user has marked as favorites. Contacts are displayed in a list that can be sorted by name, company, or last modified. List items hold an image of a contact, name and company they're working with. Tapping a contact opens a contact details page.

Contact details page holds additional information about the selected contact such as position in the company, contact details and notes. From this page user can save the contact into their device or initiate phone calls with the contact by pressing a designated buttons. In order to get access to devices phonebook an additional code library had to be added into Xamarin. Pressing the call button initiates the hardware function and launches the phone dialer from user's phone. From the phone dialer the user has an additional way to add the contact into his or her device. From contact details page user can select to modify the contact information.

My profile page works with same principle as the contact details page. The only difference is that the information displayed is related to the user himself and the call and save contact functions are disabled.

Modifying user data page is initiated from the contact details page. The modify page holds input fields for all the information the profile displays. In the input fields current values are used as default values. When user modifies a field and taps 'save' new information is pushed into a web service replacing the current information. The next time a contacts page is loaded the information is updated into the list view.

Calls page is another page holding a list view. This list logs all the recent calls the user has initiated or received. The list is sorted by time. The list is formatted in a similar fashion as the contact details with a difference that the call timestamp was added to the list item. From the list user has a possibility to initiate another call with the contact by pressing a designated button inside the list item.

The events page is another tabbed page. Tabs are 'create event' and 'view events'. The view events tab holds a calendar view and a list view. Calendar view was implemented by external code library. Selecting a date from the calendar gets all the events scheduled for that day from the web service and displays them in a list below. Selecting an event from the list opens a more detailed view with more information about the event. Create events page is a page with multiple input fields where user has to define required information regarding the event he wants to create. After the fields are filled and user taps 'create' the application creates a new event to the web service with details specified by the user.

4.6.2 Issues

During the development process I encountered issues of different magnitude. I will be concentrating only on the major issues and disregarding the smaller issues that were generally resolved after some research.

The only major issue I faced in the development so far was connecting the frontend application with the backend web service. Xamarin framework has an issue recognizing the .dll file even if it is set up following the documentation and best practices. The issue is currently being researched. While the issue remains the application was developed and tested by using 'dummy', made up data to populate the fields and manipulate.

5 Evaluation

Due to the fact that the development process is not finished and the application is not available for the end users, the possible influence on sales or the number of end users could not be determined and used to evaluate the project. Because of this the evaluation was done by having a discussion with the commissioner Mr. Saarikivi, on collecting his feedback on the project and the next steps of the project.

The main emphasis on the project was designing a user friendly mobile CRM application according to the best practices as well as documenting the incorporated functionalities in the application. Mr. Saarikivi was satisfied with the designs and the documentation produced during this thesis project. The customer feedback can be found from Appendix 5. All in all despite the fact that development not being finished by the end of the project, the project itself was deemed success by Mr. Saarikivi.

6 Conclusion

I believe that the designs and documentations will provide essentially useful once the development process is continued by the Evolvit OY. Furthermore once the application is launched it has a potential to reach the goal of increasing sales and is guaranteed to help offering the clients more comprehensive CRM service. The development process so far can serve as a base of the application upon which the remaining features are built.

References

Andersen, B & Pettersen, P. 1996. The benchmarking handbook. London: Chapman & Hall.

Baresi, L & Nitto, E. 2007. Test and analysis of web services. Dordrecht: Springer

ComScore. 15 Apr 2015. Mobile internet usage skyrockets in past 4 years to overtake desktop as most used digital platform. Accessed 20 August 2015. https://www.comscore.com/Insights/Blog/Mobile-Internet-Usage-Skyrockets-in-Past-4-Years-to-Overtake-Desktop-as-Most-Used-Digital-Platform

Dooley, J. 2011. Software development and professional practice. Dordrecht: Springer

Johnson, J. 2010. Designing with the mind in mind: simple guide to understanding user interface design rules. Elsevier Science

MaRS, 06 Dec 2009. Product development: using agile methodology for software development. Accessed 29 July 2015. http://www.marsdd.com/mars-library/product-development-using-agile-methodology-for-software-development/

MaRS DISCOVERY, 06 Dec 2009. Product development: the waterfall methodology (model) in software development. Accessed 29 July 2015. http://www.marsdd.com/mars-library/product-development-the-waterfall-methodology-model-in-software-development/

McWherter, J & Gowell, S. 2012. Professional mobile application development. Wiley

Metodix. No Date. What is a constructive research approach? Accessed 29 July 2015. http://www.metodix.com/en/sisallys/01_menetelmat/02_metodiartikkelit/lukka_const_research_app/02_mita_konst_tut_tark

Microsoft. No date. XAML. Accessed 30 August2015. https://msdn.microsoft.com/en-us/library/cc189054(v=vs.95)

Nakov, S. 2013. Fundamentals of computer programming with C#. Sofia: Svetlin Nakov & Co.

Newman, E. 2002. Understanding web services: XML, WSDL, SOAP, and UDDI. Addison-Wesley Professional

Payne, A. 2012. Handbook for CRM: achieving excellence in customer management. Taylor and Francis

Petzold, C. 2010. Programming Microsoft® Windows® with C#. Microsoft Press

Porebski, B, Przystalski, K & Nowak, L. 2011. Building PHP applications with symfony, CakePHP, and Zend framework. Wiley

Richter, F. 2013. iOS remains the favored platform for enterprise apps. Accessed 01 September 2015. http://www.statista.com/chart/1509/mobile-platform-survey/

Rouse, M. 2007. Web services (application services) definition. Accessed 29 August 2015. http://searchsoa.techtarget.com/definition/Web-services.

Rouse, M. 2007. Functional specification definition. Accessed 30 August 2015. http://searchsoftwarequality.techtarget.com/definition/functional-specification

Saarikivi, A. 2015. Commissioner. Evolvit OY. Interview 11.02.2015. Personal communication.

Saarikivi, A. 2015. Commissioner. Evolvit OY. Emails to the author. 01.11.2015 – 01.10.2015. Personal communication.

SmartInsigths. No date. Mobile marketing statistics 2015. Accessed 20 August 2015. http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/

Stickdorn, M. 2012. This is service design thinking: basics-tools-cases. BIS Publishers

TechCrunch. 21 Aug 2014. Majority of digital media consumption now takes place in mobile Apps. Accessed 20 August 2015. http://techcrunch.com/2014/08/21/majority-of-digital-media-consumption-now-takes-place-in-mobile-apps/

Usability.Gov. No date. User experience basics. Accessed 28 August 2015.

http://www.usability.gov/what-and-why/user-experience.html

Usability.Gov. No date. User interface design basics. Accessed 29 August 2015.

http://www.usability.gov/what-and-why/user-interface-design.html

Wax, A. 2008. 7 Essential guidelines for functional design. Accessed 28 August 2015.

http://www.smashingmagazine.com/2008/08/7-essential-guidelines-for-functional-design/

Westfall, L. 2009. The certified software quality engineer handbook. Milwaukee: ASQ Quality Press

Wulf, A. 2012. Wireframes: a great way to starting development projects. Accessed 27 August 2015. http://www.infoq.com/articles/wireframes-start-development-projects

Figures

Tables

Appendixes

Appendix 1 Customer requirements

Thesis kickoff meeting

Interviewer: Samuli Laine
Interviewee: Antti Saarikivi, Evolvit OY

Topic: Customer requirements
Date: 11.02.2015

1. What's the main focus group for the application, should they be taken into consideration during the development?

   Mainly sales divisions of the company clients

2. Main purpose of the application?

   Offer better service for clients and mobile option to the existing service.

3. Key features of the application

   - Calling
   - Events (Access and create)
   - Contacts
   - Recent calls
   - Save contacts on device
   - Favorite contacts

4. Most important things to take into consideration during the development?

   User friendliness

5. Platforms to concentrate the most on?

   iOS

6. Any other wishes / ideas?

No.

7. Who takes care of publishing and maintenance?

Evolvit.

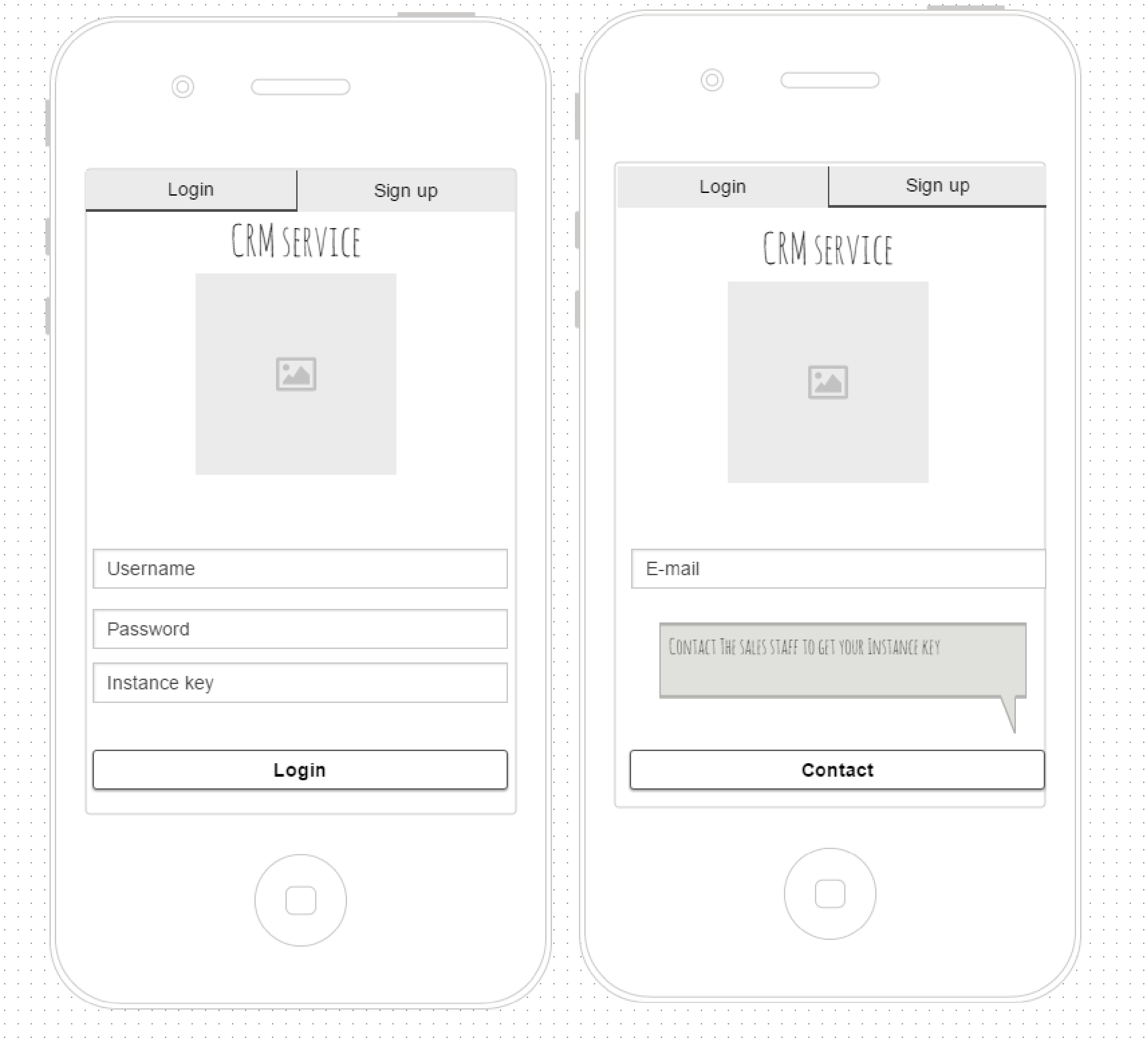
8. Color Scheme

Follow the sites color

Appendix 2 Wireframes

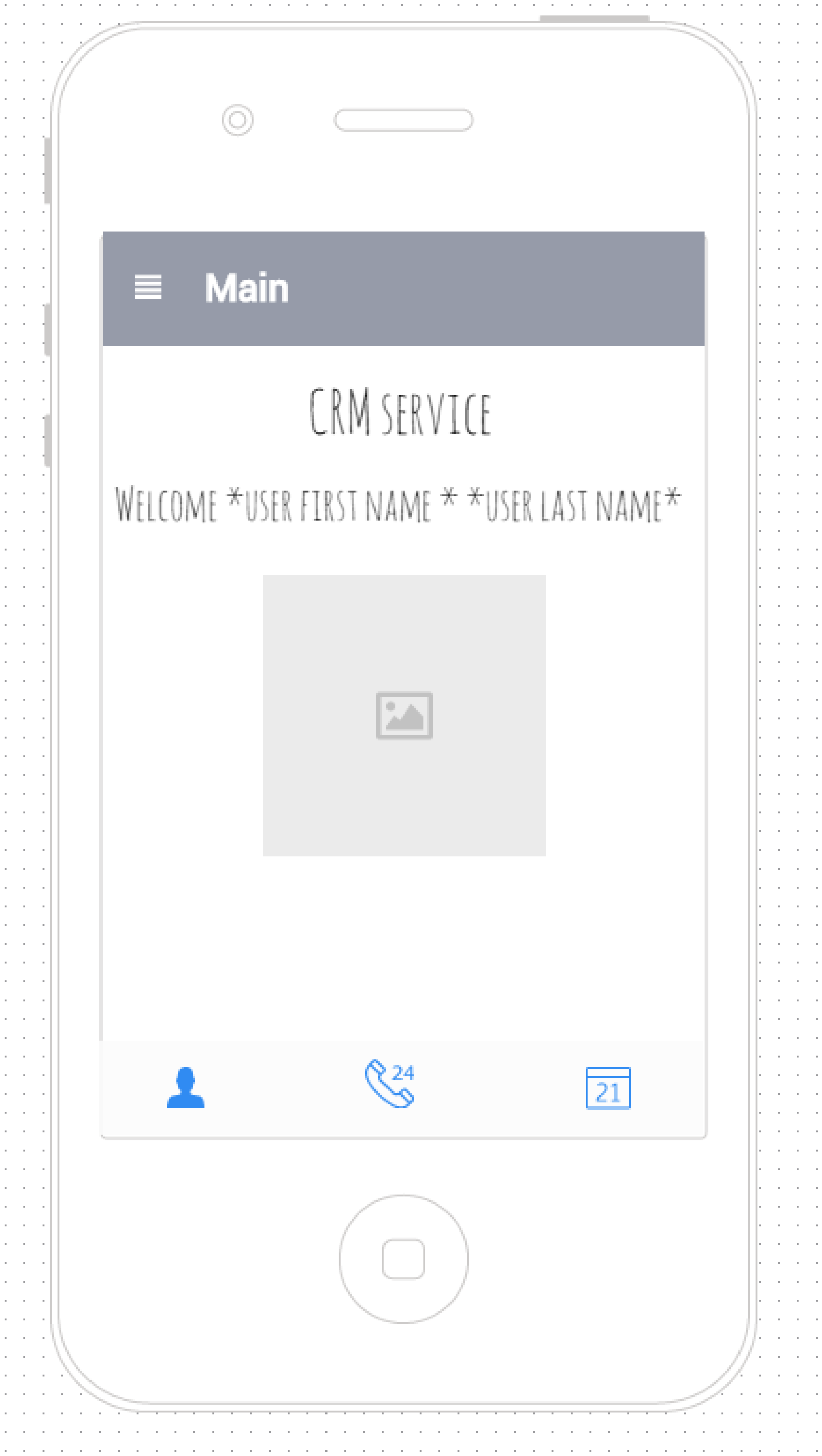
Login / Sign up page

Authentication fields, login and sign up tabs. CRM service name and logo.



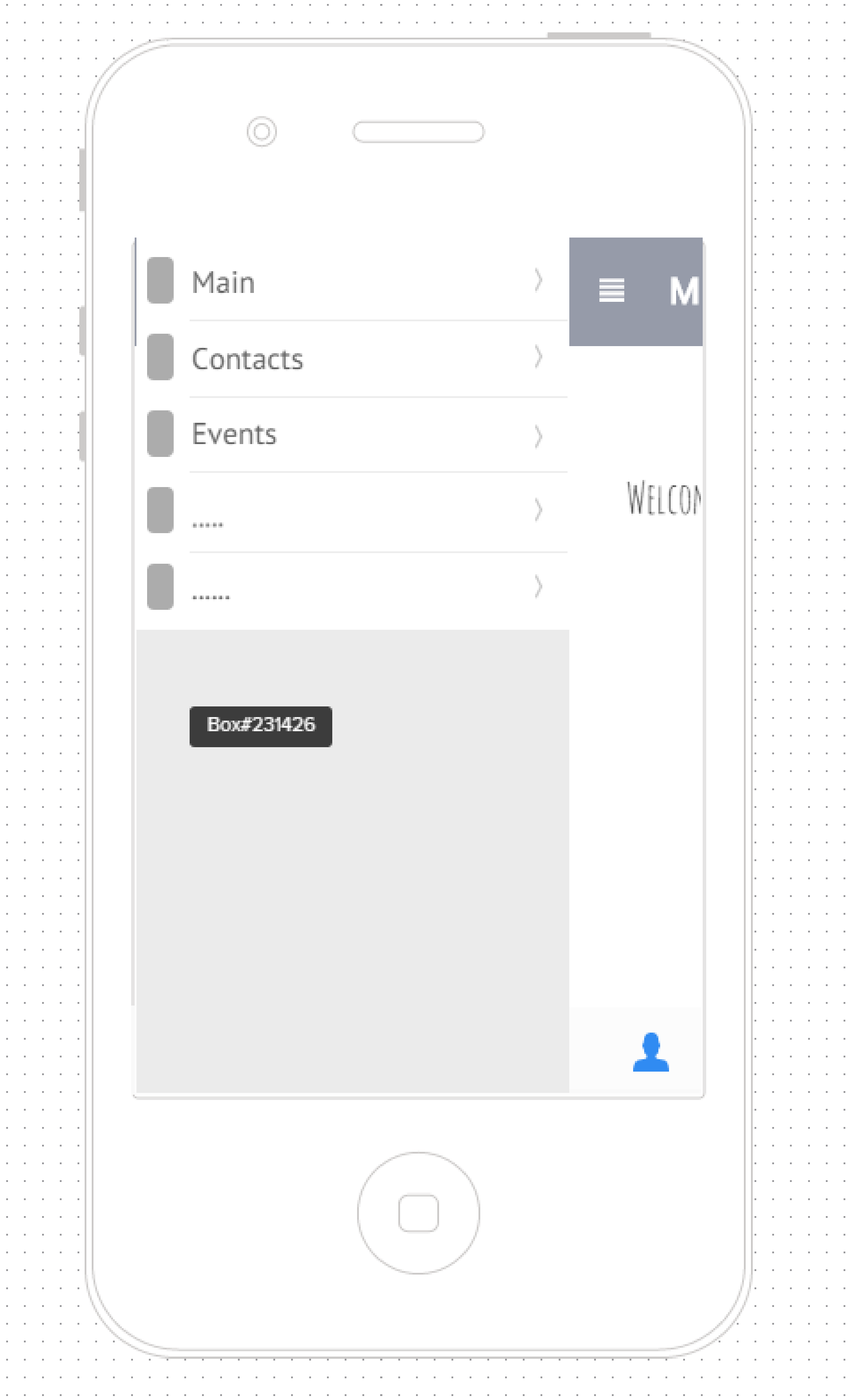
Main page

Side menu, welcome user, logo, CRM-service name, quick navigation buttons



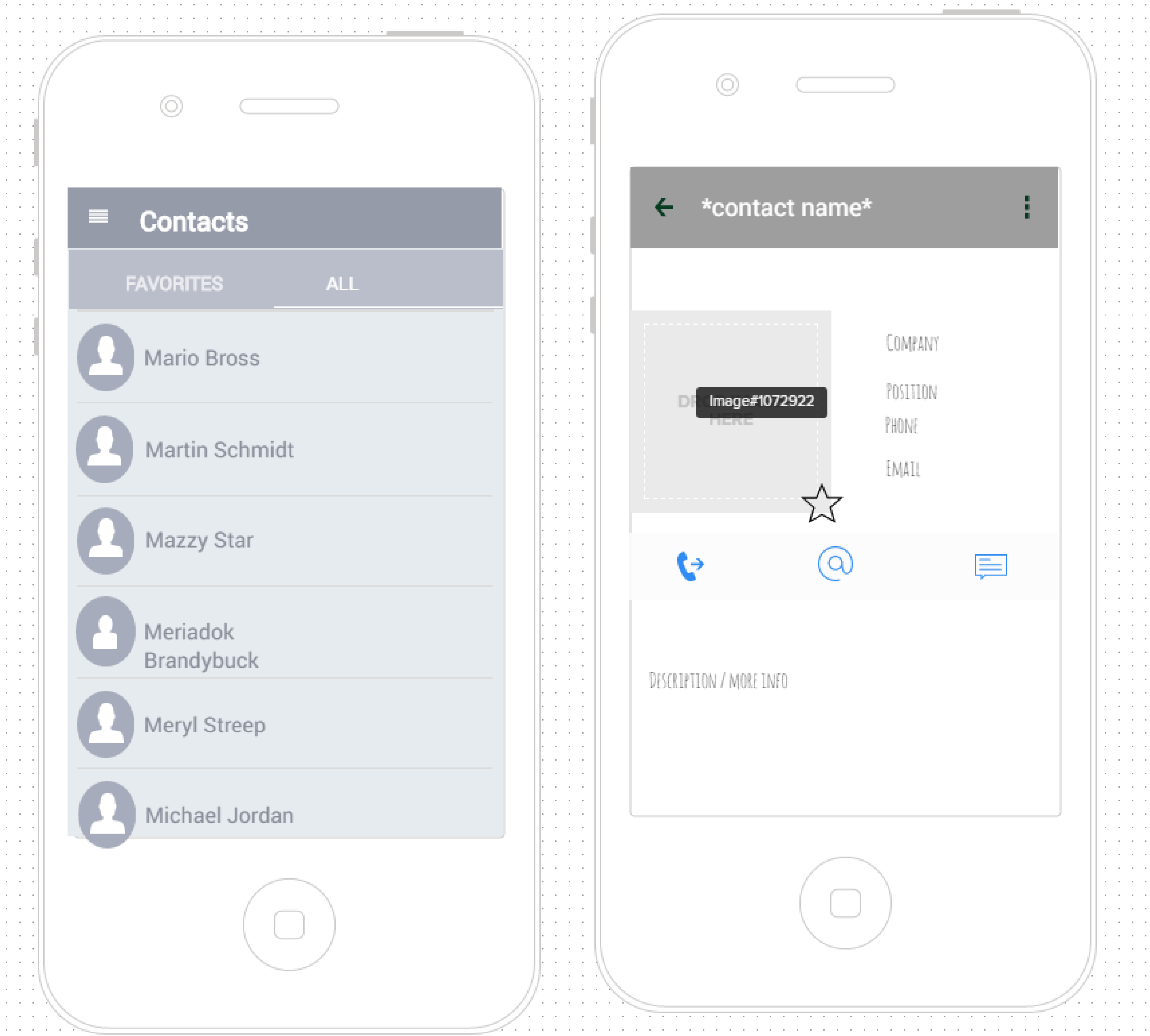
Side menu navigation

Side menu open, pushes content out of the screen. Listed with name of the page and icon.



Contact lists, all and favorite and contact details

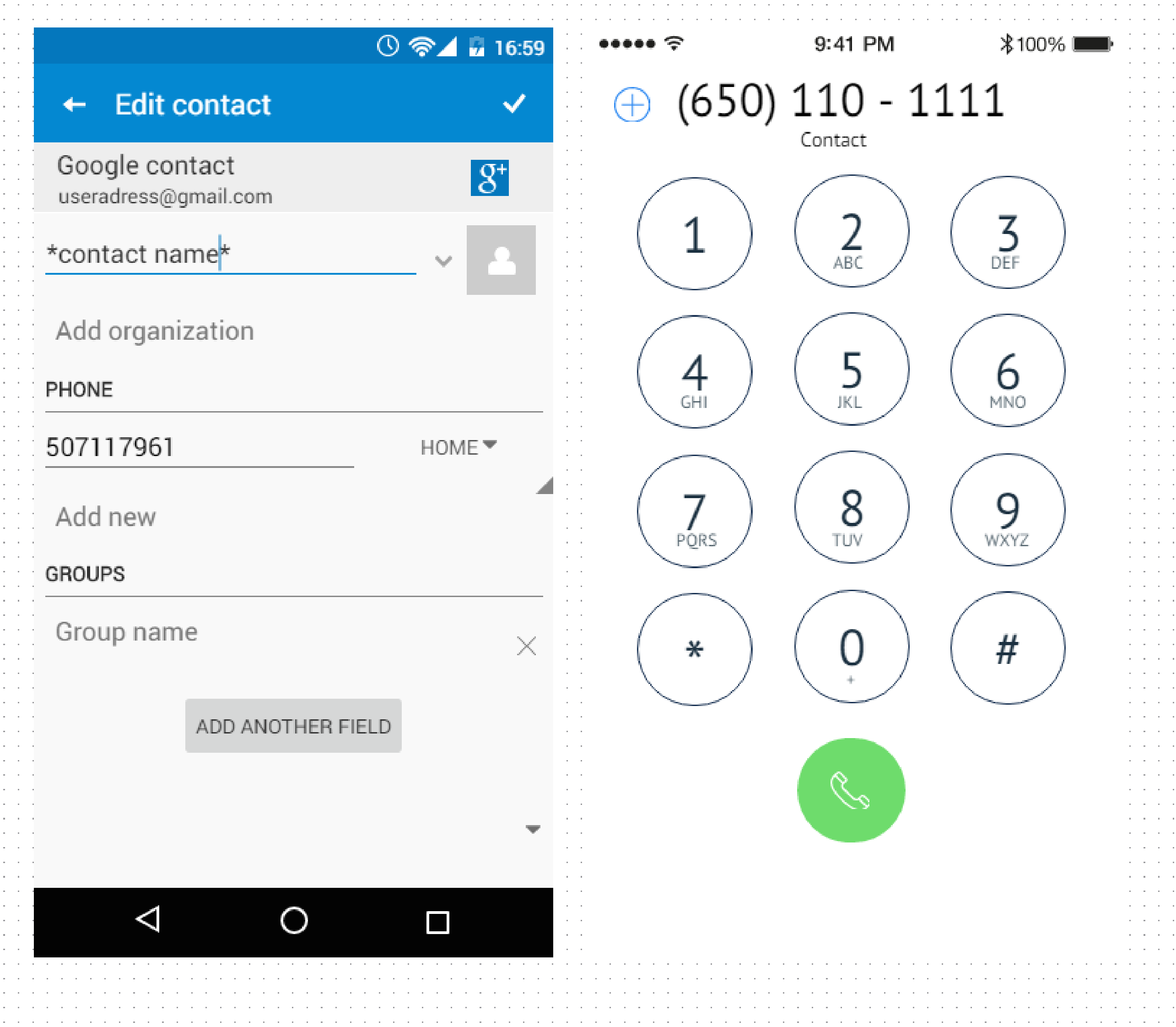
Contacts listed based on being favorite. List displays name and image. Tapping list item opens contact details page



Contact details page contains most of the features in the application. Making calls, setting contact as favorite and modifying contacts is done from here. Additional data about the selected contact is located into this pages as well

Modify contacts & Call

Hardware call functionality initiated. Edit contact contains input fields with previous values as default.



16:59

Edit contact

Google contact
useradress@gmail.com

contact name

Add organization

PHONE

507117961 HOME

Add new

GROUPS

Group name

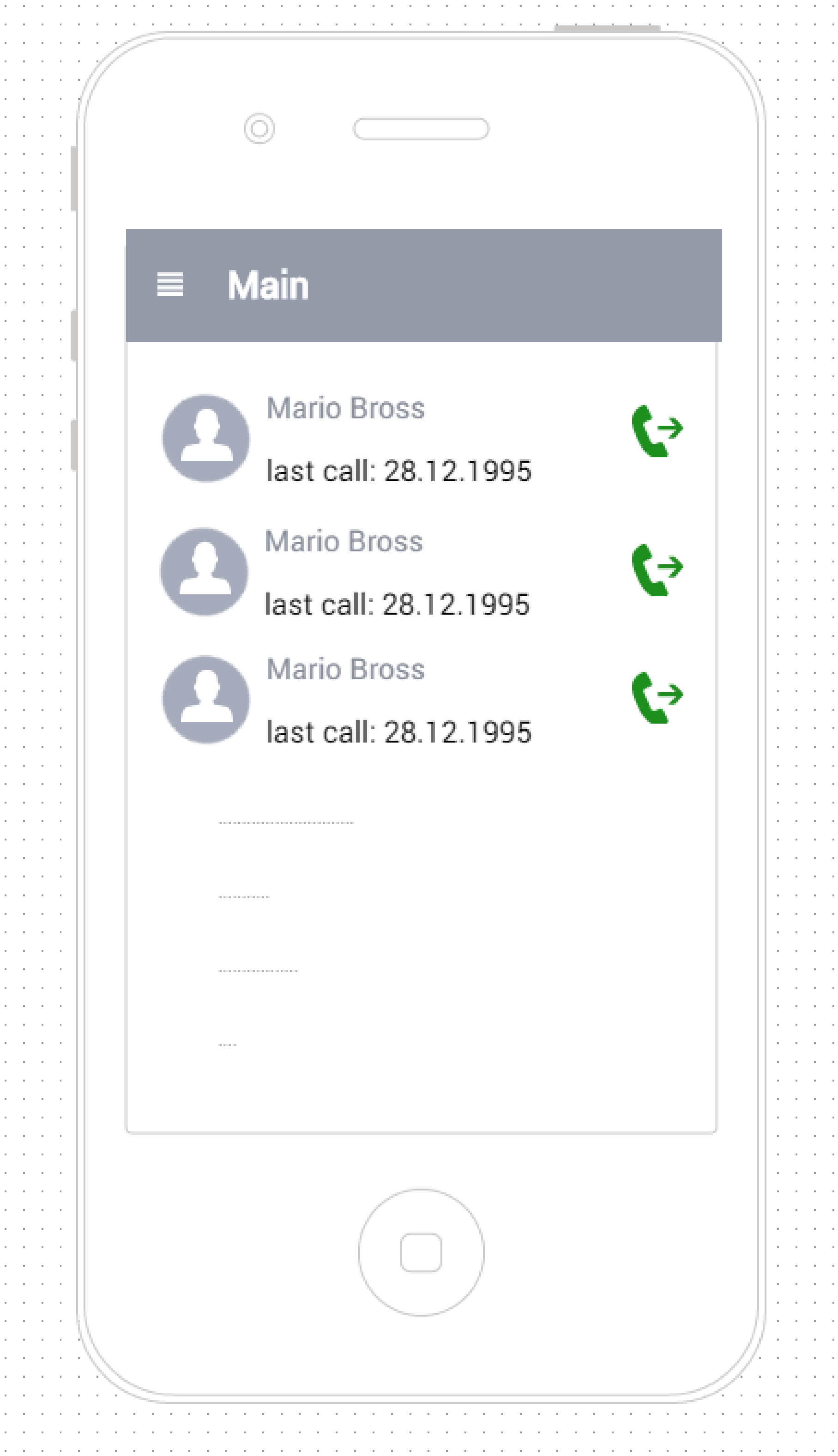
ADD ANOTHER FIELD

9:41 PM 100%

(650) 110 - 1111
Contact

Call list

Recent calls are listed into a list. Selecting an item in the list initiates the call function all over.



Main

Mario Bross

last call: 28.12.1995

Mario Bross

last call: 28.12.1995

Mario Bross

last call: 28.12.1995

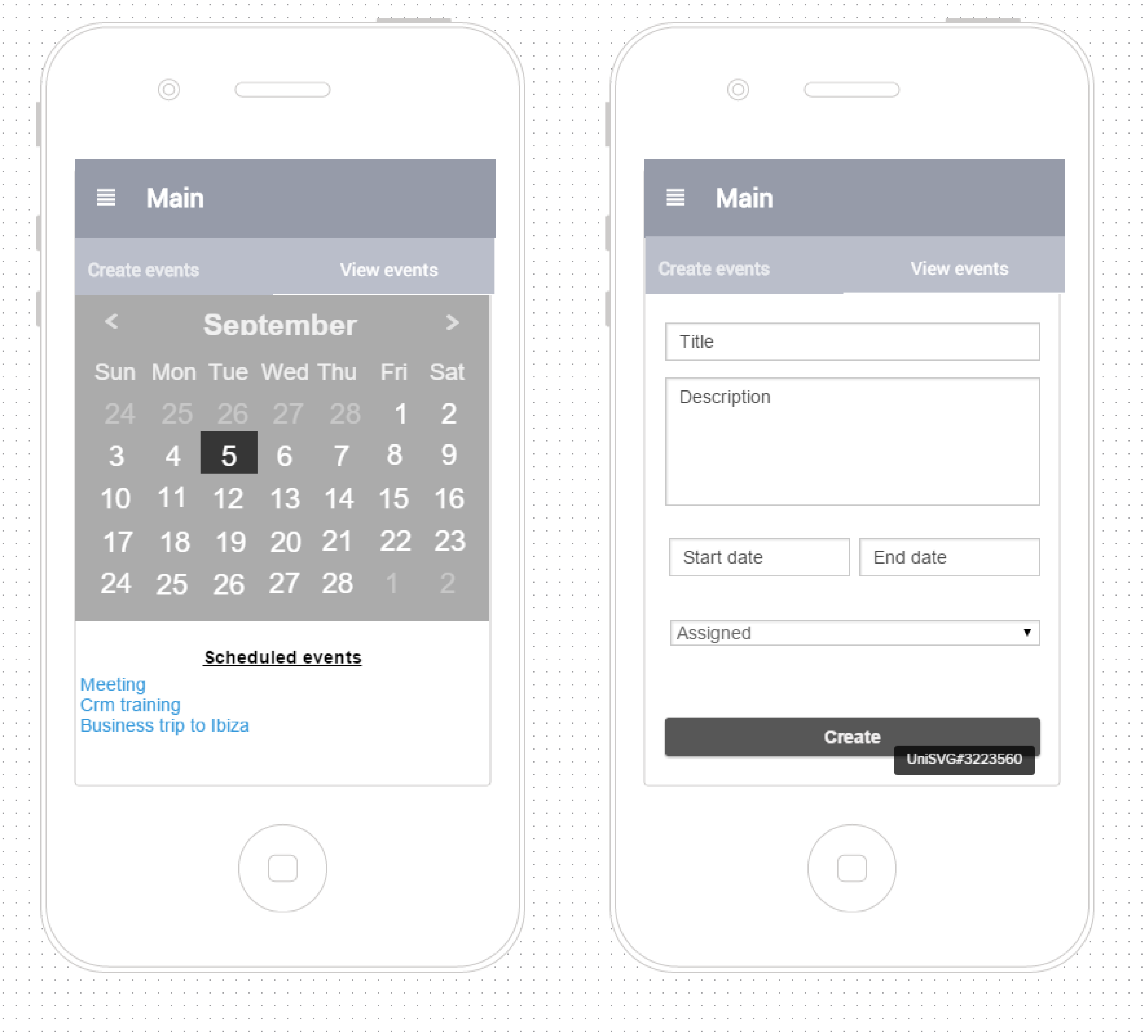
Events

View events:

Calendar, selecting a date shows scheduled events in a list underneath.
Clicking on event opens a page with more information.

Create event:

Page with input fields, tapping ‘create’ button will create a new event with values from the fields.



Sun	Mon	Tue	Wed	Thu	Fri	Sat
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2

Scheduled events
Meeting
Crm training
Business trip to Ibiza

Appendix 3 Functional design

Functionality design

Introduction

In this document I am describing the main functionalities of the CRM application. CRM application is a cross platform customer relation management application created for Evolvit OY. The application is created on top of existing CRM web service.

Majority of the desired features of the application function with a connection with the web service. The connection is formed by using a .dll (dynamic link library) file provided by the client. WSDL defines the communication between the client and the backend service.

Key features of the application:

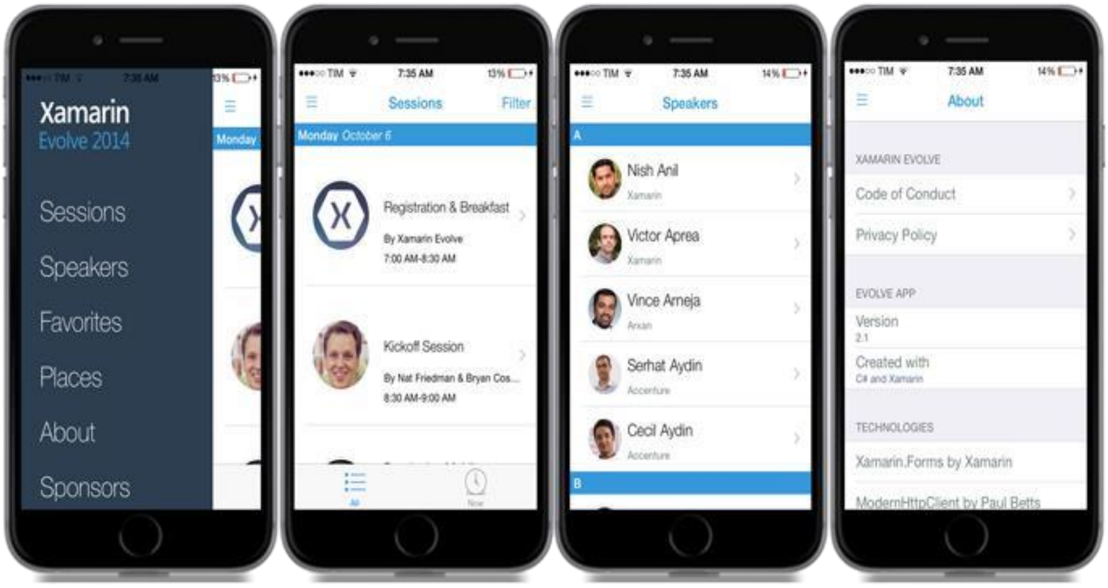
- Calling
- Events (Access and create)
- Contacts
- Recent calls
- Save contacts on device
- Favorite contacts

Layout related

In this sections I will describe the navigation and layout related functionalities.

Main navigation structure

As the primary navigation a master – detail layout was chosen. Master detail functions by having 2 major UI elements on the same page, a side menu view and a main content area (Figure 1). Side menu (master page) contains all the navigation elements of the main navigation as a list. Selecting an item for the list changes the content of the detail (main content area) with either other content or whole another layout page.



Master – detail layout is aside from the tabs layout is considered as the most powerful and common layouts. This means it is familiar for the users and user friendly. The layout is also easily expanded when new features are introduced.

In addition to the main navigation each individual page has its' own navigation elements depending on its needs. Those will be covered further in the following section.

List views

Because of the nature of required features list views were the most common layout element in this project. Most of the features evolved around displaying and or modifying existing data. To this end the most efficient way was to display the data on custom designed lists. Once user has selected the item of the list which he wants to view or modify, a page specific navigation function takes the user into another page with more details and possibilities to modify data. Section later on will cover populating list views with data in greater detail.

<u>Login</u>

In order to access any data within the application users have to be authenticated. Therefore the first thing users see when they open the application is the login screen. This CRM application uses three fields in order to authenticate the user.

Firstly user has to submit an instance key. Instance key is provided by the seller once user starts using CRM service. Instance key is used as a link to a certain section within the service that contains exclusively data relevant to the current user (clients, events, contacts etc.). By doing this the integrity of information is ensured. Only the ones with the instance key get to access their own data. Calendar features can be achieved through Xamarin.Mobile library provided by Xamarin developers. Xamarin.Mobile is a library that exposes a single set of APIs for accessing common mobile device functionality across iOS, Android and Windows platforms.
After submitting the instance key, traditional username – password login is initiated. Entered username and password are pushed through the .dll file into the instance in the web service defined by instance key and if username and the password match with an existing account a positive response is sent. Positive response allows user to move past the login page into main page of the application while negative response launches a popup notification asking users to check their credentials or contact the service provider.

Call

Call function is one of the only features within the application that doesn’t require pushing data into the web service. Call function is initiated by user clicking a call button in the application. Once the call function is initiated the application will fetch the phone number of the person user is trying to reach and sets in into a variable.

```
var url = new NSUrl ("tel:" + *Phone number reference *);
```

“tel” in front of the number reference lets the application know that the following set of numbers is a phone number and should be threated accordingly.

```
UIApplication.SharedApplication.OpenUrl (url);
```

Once the number is set a phone specific hardware function is called with the reference to the number and phone dialer is launched.

Populating listviews

Populating list views with data functions through the .dll file. Every time a page is loaded lists are populated with desired data, generally sorted by person in each element.

Contacts

Contact list of the application displays ALL of the available contacts in the service instance. Contacts are displayed in a simple but informative list view that gives a quick overview and the basic info of the contact.

List
-- List item
--- User image
--- User name
--- User Company

Above is an example of how a list can be formatted.

Contacts in the list can be filtered alphabetically or by company (also alphabetically). List items can be tapped in order to navigate to another page that contains more information about the contact. Additional information can contain for example contact details, position in company and more information the person. This data is kept in the web service and is accessed through the .dll file and displayed in the client application. The detail page also contains the features to modify contacts and making a phone call or adding contact to your favorites. If contact is modified new data is pushed into the web service and the next time the contact list page is opened the list is automatically updated.

Recent calls

After performing a phone call details of the user is marked as your recent contact. This can be done either marking the contact as recent forever or for X duration (for example 30 days) and is removed afterwards if call is not repeated. Once the “Recent Calls” page is opened from the client, out of all the contacts the ones that are not marked as recent contact are filtered out and the rest are pushed into a list view.
Recent calls list view is formatted:
- List
-- List item
--- User image
--- User name
--- Called date / called x time ago
Tapping an item in the list initiates another phone call to the contact using the same make call feature.

Favorite contacts

Favorite contacts works on the same principle as the recent calls list. The contacts that cannot be found from your favorites are filtered and the rest are displayed in a list. Favorites list is formatted and has access to exactly the same functions as the contacts list.

Events

Creating and viewing events is the only feature not using a list view in this application. Even though viewing events works on the same principle as viewing contacts. Events can include for example conferences, meetings or business trips. Event data is pulled from the web service into the client and displayed in a calendar format. Meaning the days that contain events are marked on different color and once they are tapped another page with more detailed information about the nature of the events is displayed.
Creating events works on a same principle as modifying contacts. ‘Create activity’ page contains input fields for information such as: event name, date, duration, description and location. Once the required fields are filled and user taps ‘create activity’ button, a new event is

created into the web service with the values from the client. Once the events page is loaded the next time users can see the created event.

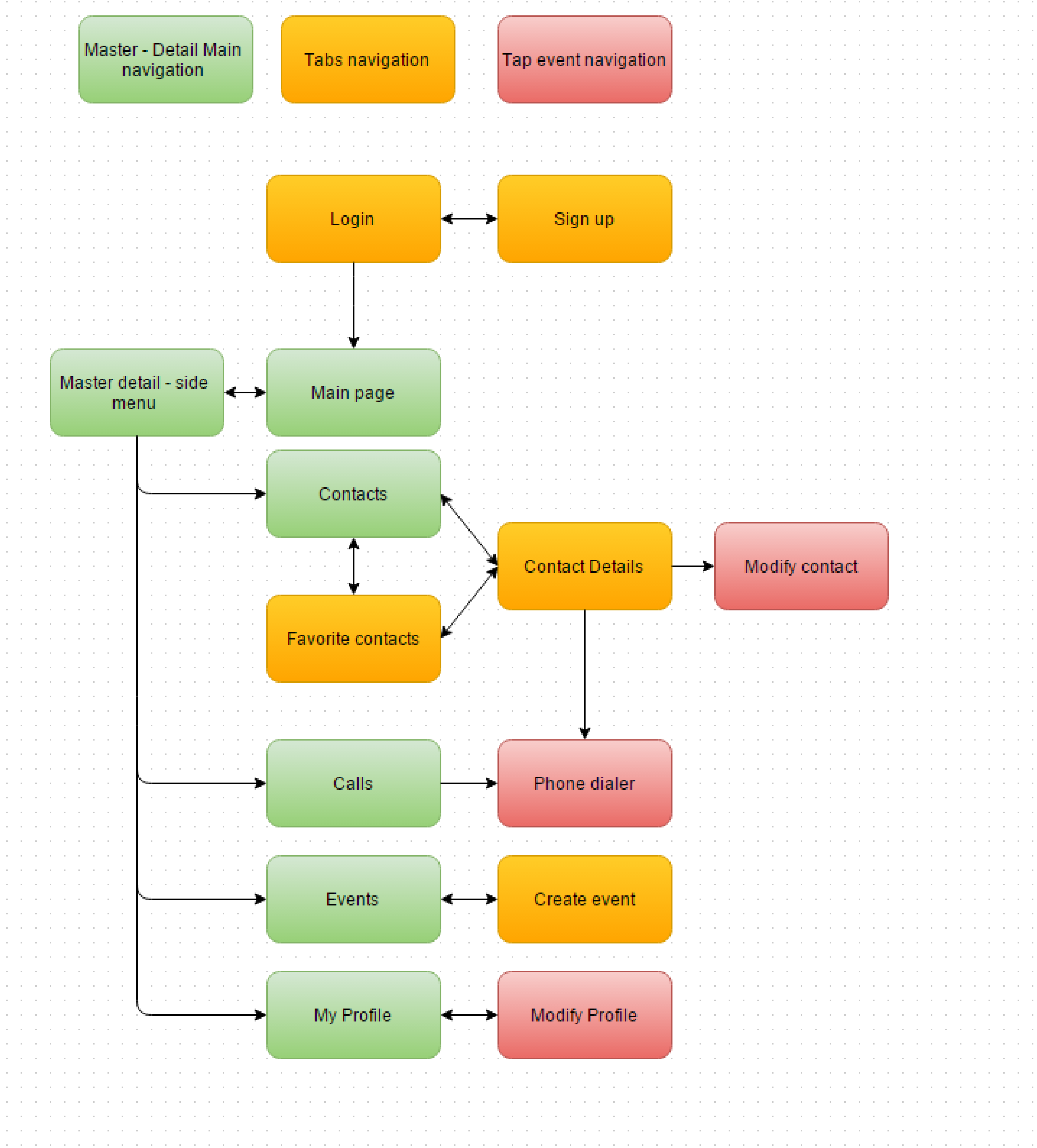
Save contacts on device

In the application there are several ways to save contact data into your hardware device. One is functioning purely on phones individual hardware functions. This is achieved by initiating a phone call with desired user. Once the call is initiated and phone dialer is opened the phone dialer has embedded function to save contacts into phonebook of your device. Secondary way to access and modify the phone book of your device from the application is through Xamarin.Mobile library provided by Xamarin developers. Xamarin.Mobile is a library that exposes a single set of APIs for accessing common mobile device functionality across iOS, Android and Windows platforms.
Once the library has been included to the project phonebook can be accessed by adding 'using Xamarin.Contacts' into the script where contacts are required. Accessing phonebook requires READ_CONTACTS permissions on Android (is to be set into the manifest file).
Contacts can be added into the phonebook by creating a new contact and assigning relevant values from the web service through client. Example of getting contacts below.

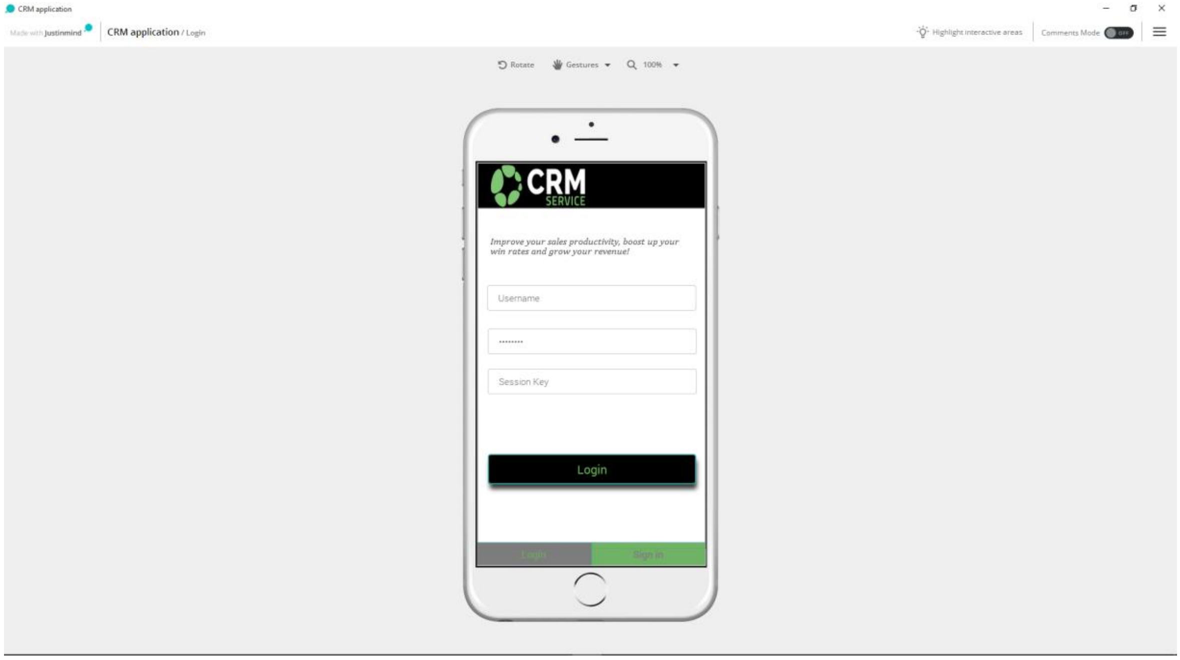
```
using Xamarin.Contacts; // ...  var book = new
Xamarin.Contacts.AddressBook (); //
new AddressBook (this); on Android if (!await
book.RequestPermission()) {
    Console.WriteLine ("Permission denied by user or manifest");
return;
}
foreach (Contact contact in book.OrderBy (c => c.LastName)) {
    Console.WriteLine ("{0} {1}", contact.FirstName, contact.LastName);
}
```

Appendix 4 Functional specification

Navigation Tree



Login Page & Sign up page

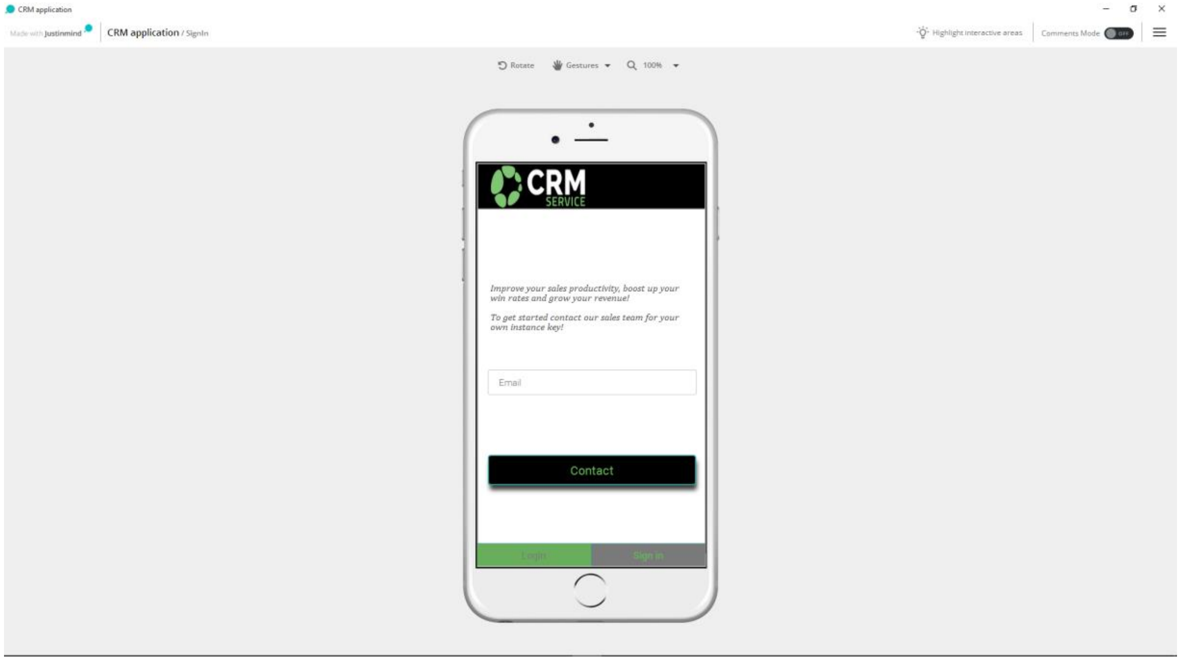


Improve your sales productivity, boost up your win rates and grow your revenue!

Username

·········

Session Key



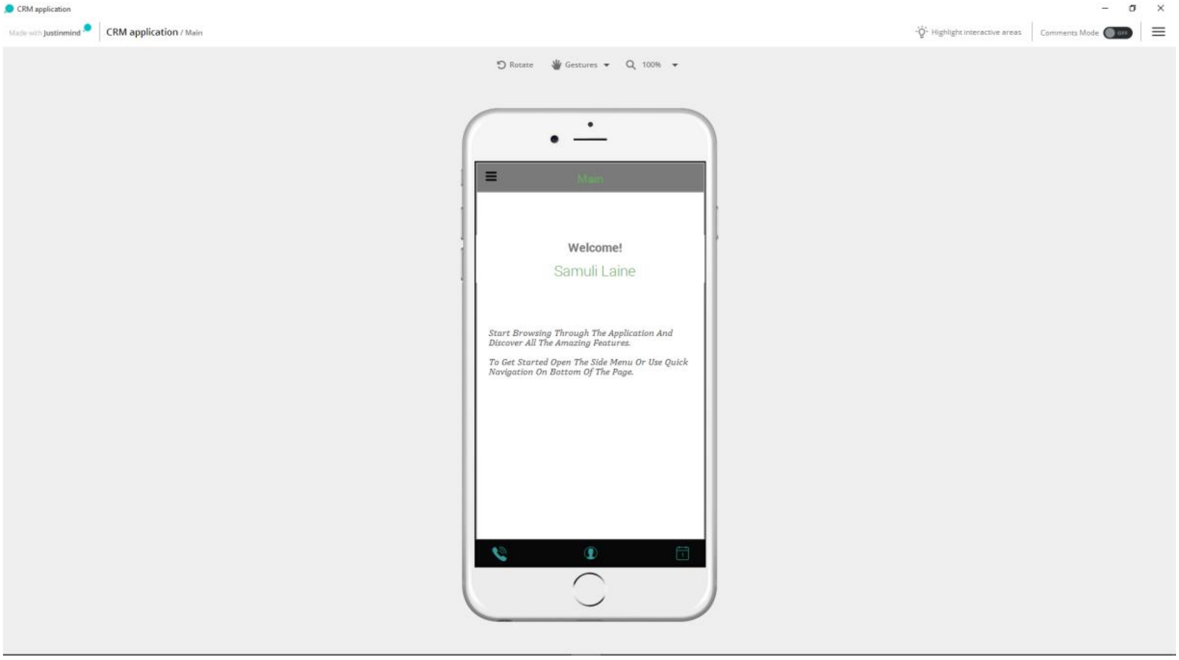
Improve your sales productivity, boost up your win rates and grow your revenue!

To get started contact our sales team for your own instance key!

Email

Main page master - detail

Detail – main page



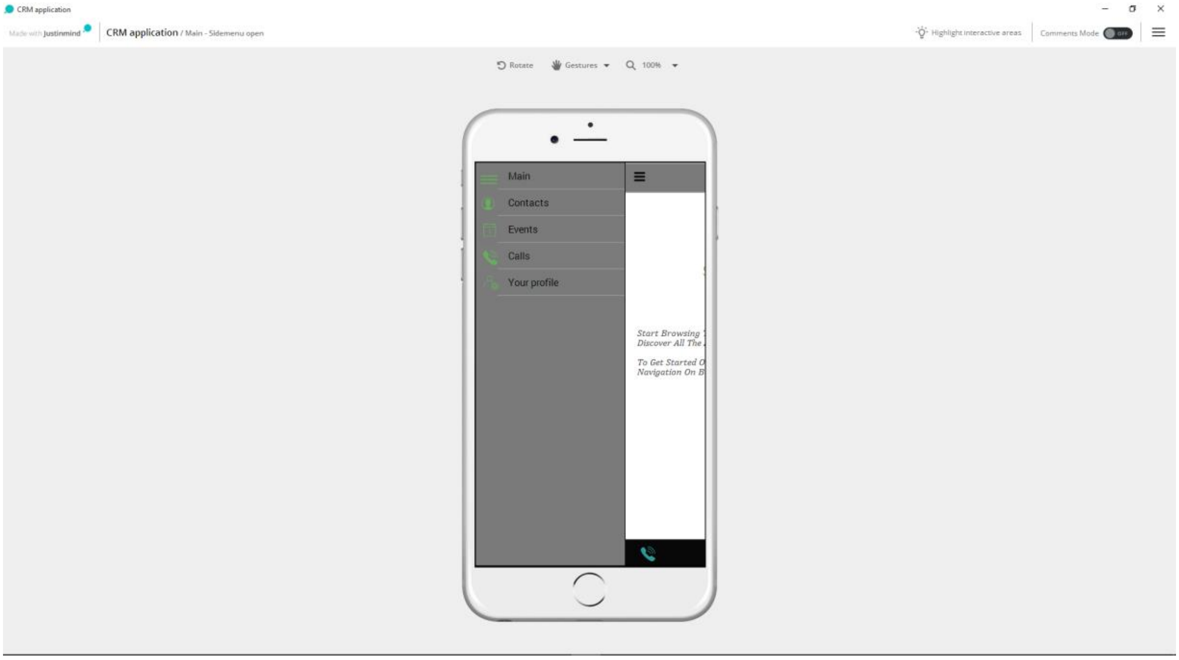
Welcome!

Samuli Laine

Start Browsing Through The Application And Discover All The Amazing Features.

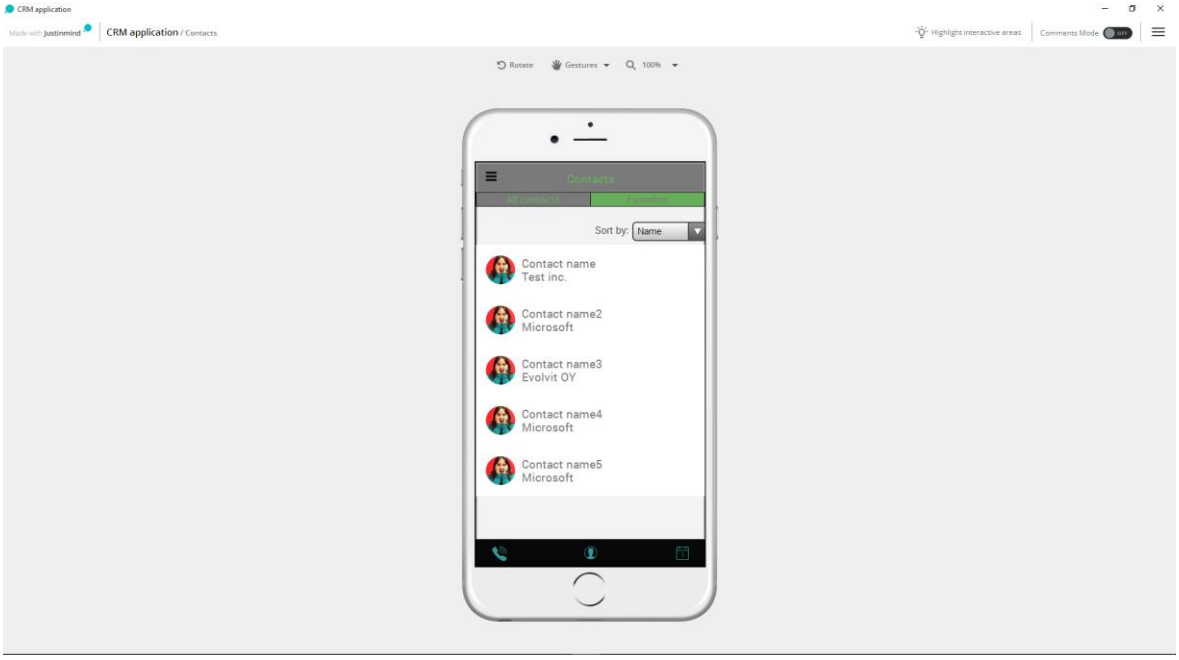
To Get Started Open The Side Menu Or Use Quick Navigation On Bottom Of The Page.

Master



Contacts

All



Sort by: Name

Contact name
Test inc.

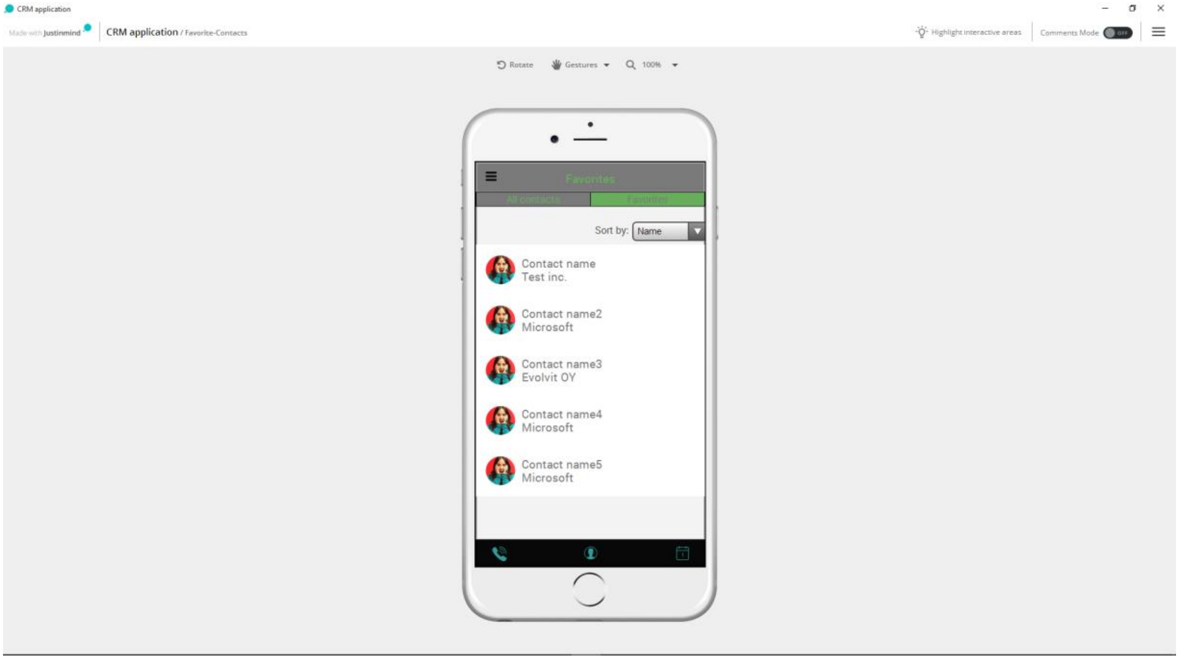
Contact name2
Microsoft

Contact name3
Evolvit OY

Contact name4
Microsoft

Contact name5
Microsoft

Favorites



Sort by: Name

Contact name
Test inc.

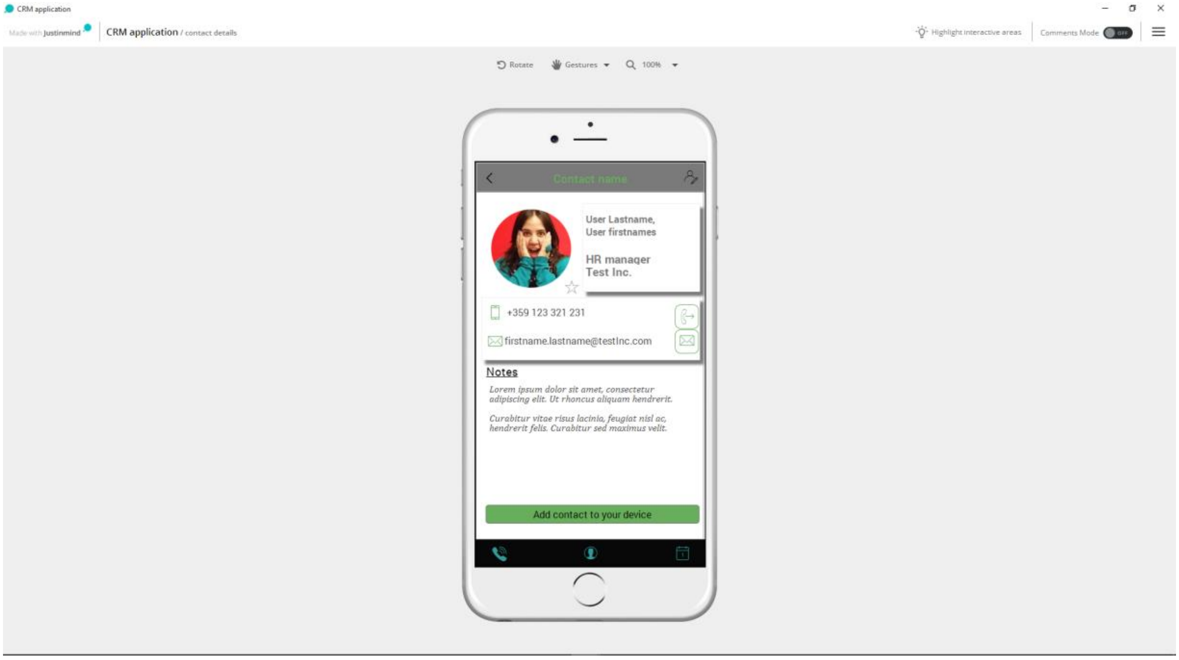
Contact name2
Microsoft

Contact name3
Evolvit OY

Contact name4
Microsoft

Contact name5
Microsoft

details


CRM application
Made with Justinmind
CRM application / contact details
Highlight interactive areas
Comments Mode OFF
Rotate
Gestures
100%
Contact name
Add contact to your device

User Lastname,
User firstnames

HR manager
Test Inc.

+359 123 321 231

firstname.lastname@testInc.com

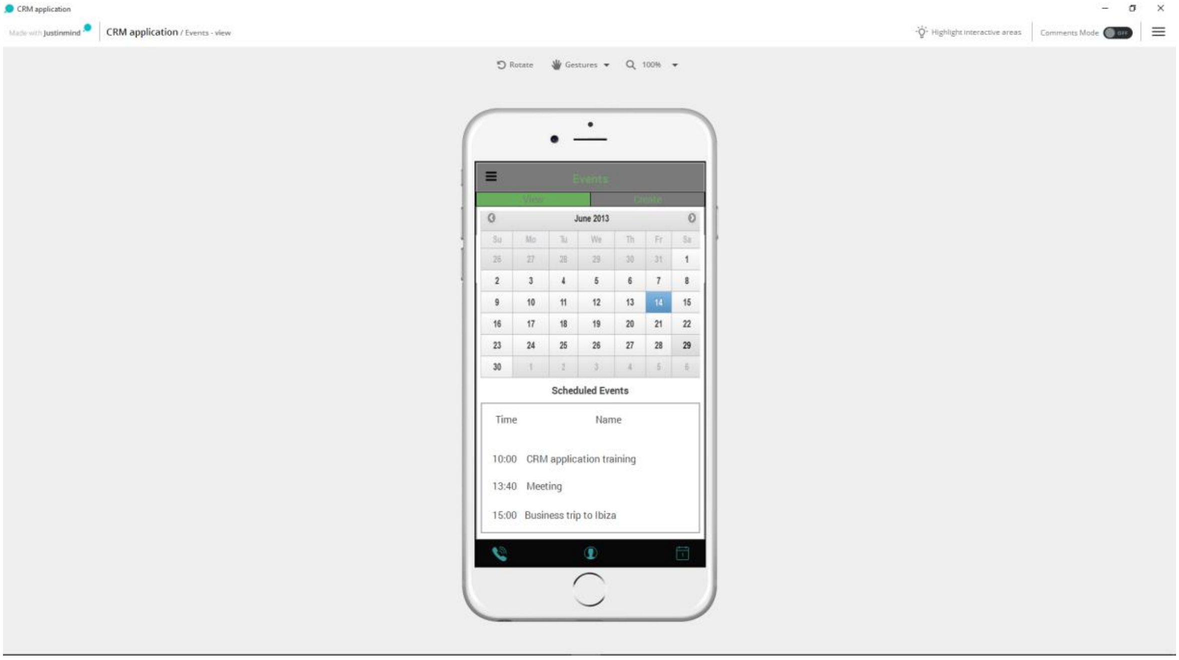
Notes

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut rhoncus aliquam hendrerit.

Curabitur vitae risus lacinia, feugiat nisl ac, hendrerit felis. Curabitur sed maximus velit.

Events

view



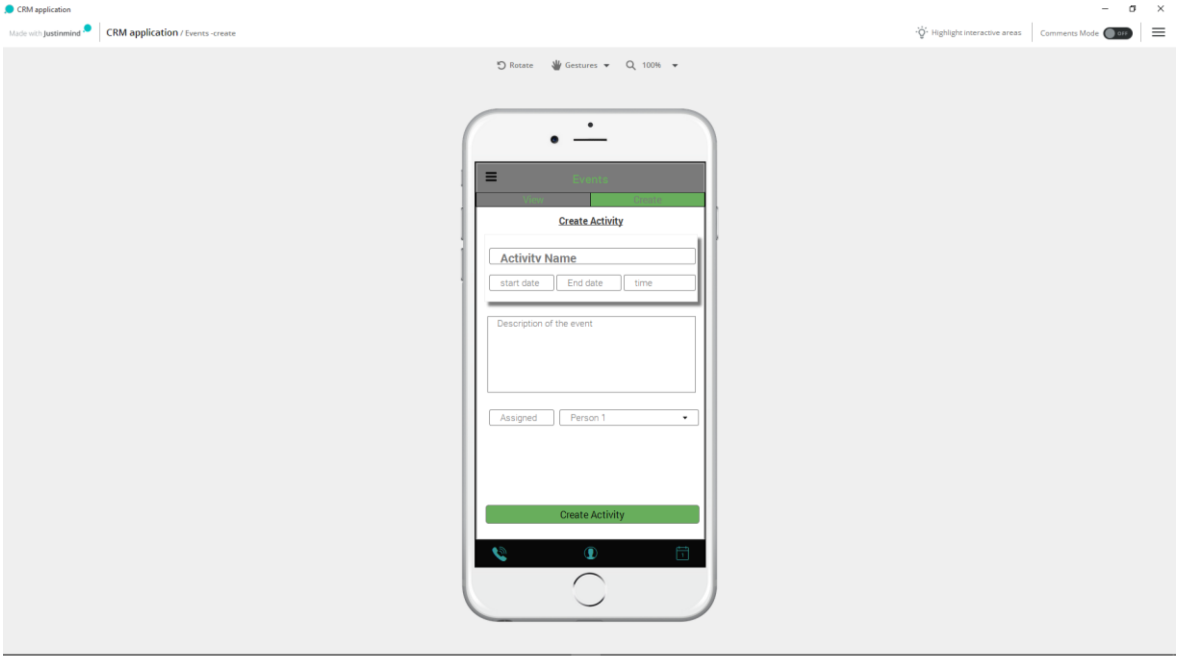
June 2013

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Scheduled Events

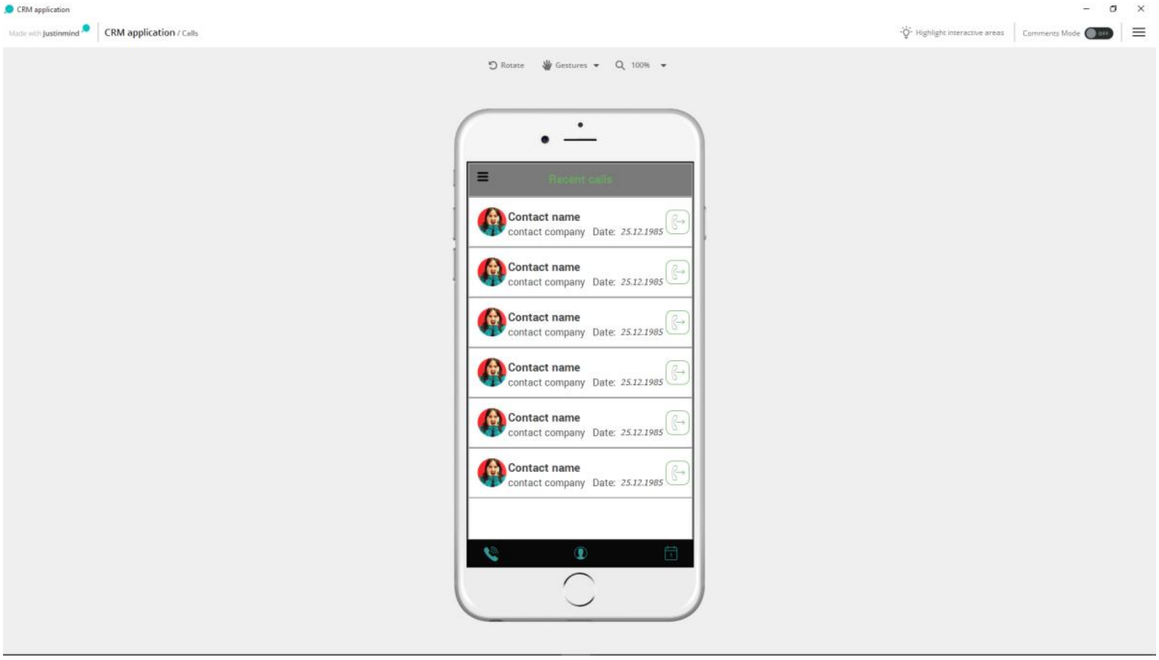
Time	Name
10:00	CRM application training
13:40	Meeting
15:00	Business trip to Ibiza

create

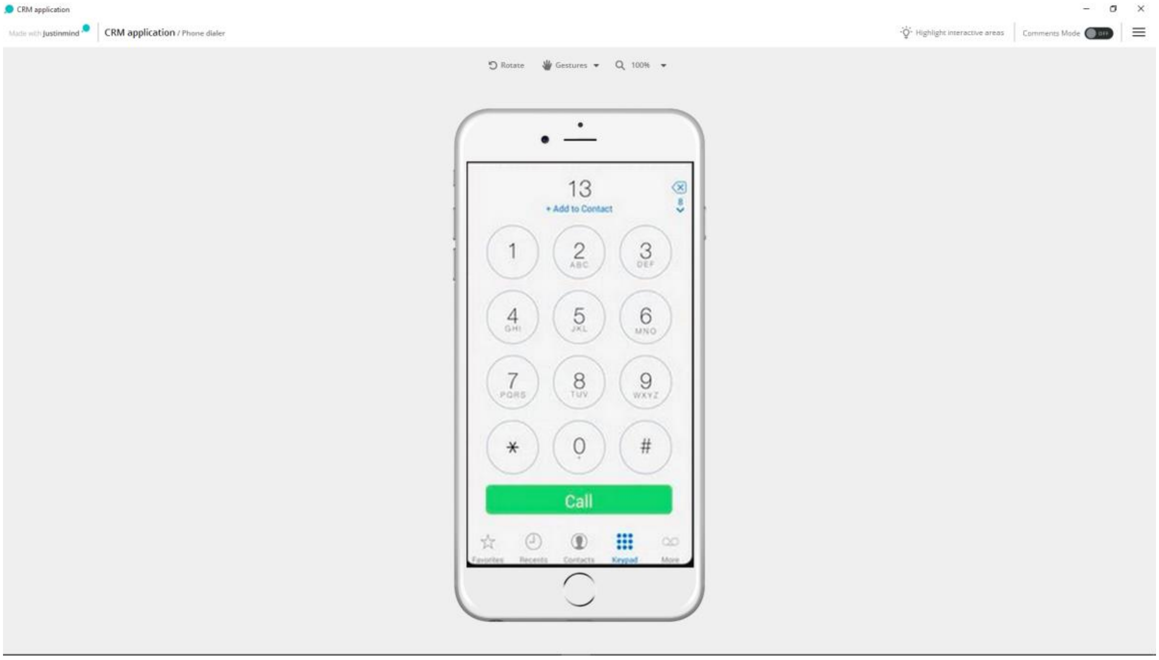


Calls

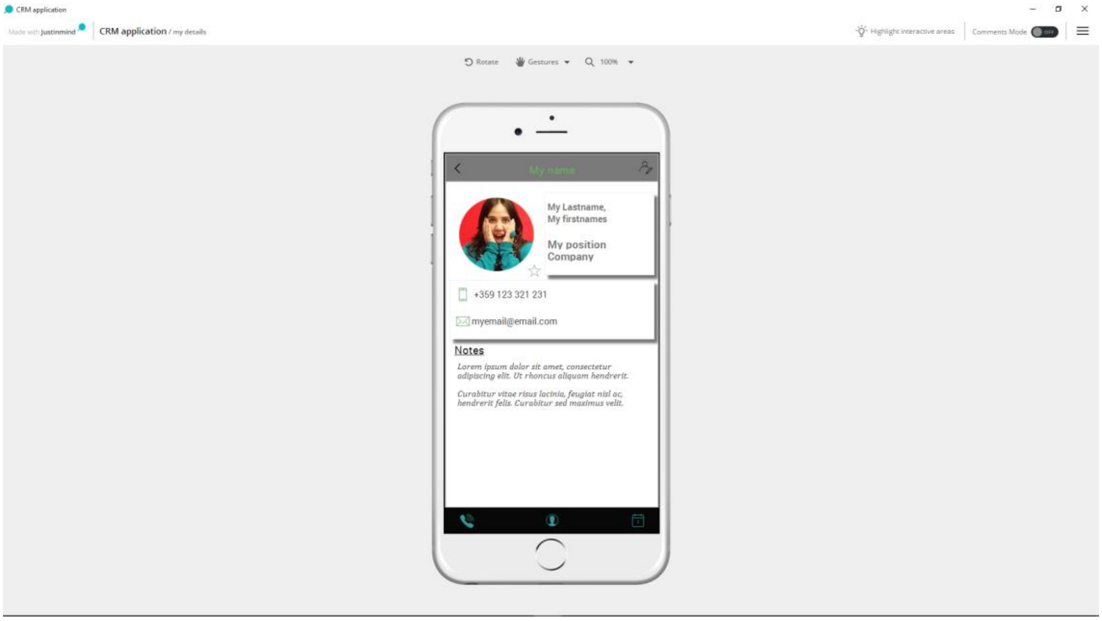
Recent calls


CRM application
Made with Justinmind
CRM application / Calls
Highlight interactive areas
Comments Mode
OFF
Rotate
Gestures
100%
Recent calls
Contact name
contact company Date: 25.12.1985
Contact name
contact company Date: 25.12.1985
Contact name
contact company Date: 25.12.1985
Contact name
contact company Date: 25.12.1985
Contact name
contact company Date: 25.12.1985
Contact name
contact company Date: 25.12.1985

Phone dialer



My profile



My name

My Lastname,
My firstnames

My position
Company

+359 123 321 231

myemail@email.com

Notes

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut rhoncus aliquam hendrerit.

Curabitur vitae risus lacinia, feugiat nisl ac, hendrerit felis. Curabitur sed maximus velit.

Appendix 5 Feedback

Design and development process feedback

Interviewer: Samuli Laine

Interviewee: Antti Saarikivi

Date: 13.10.2015

Evaluation:

Working in this project with Samuli was fairly easy because Samuli had good knowledge in mobile development and very good attitude. Samuli showed that he knew what he was doing from the beginning. Samuli had good ideas about the design and the code generated in this project was smooth and well organized. Unfortunately we were not able to finish this project with Samuli, because of technical reasons that were not in Samuli's reach to fix.

Otherwise we are happy with Samuli's work and design. From this project we got good model and ideas for our future mobile application that will surely be taken in consideration when we will start developing.

Antti Saarikivi
040 519 5354
IT-Consultant
Evolvit