



PROJECT DOSSIER
2015 SOFA GTL
TREEWATCH PROJECT



Jan Kerkenhoff
Martijn Bonajo
Max van der Linden
René Karoff
Ron Gebauer

2016-01-17

Document information

Document name: Final Report
Document owner: TreeWatch
Company/Organisation: Fleuren Baarlo
Contact person: Max van der Linden, Group leader
Date: 2016-01-17
Place: Fontys University of Applied Science Venlo
Authors:
Jan Kerkenhoff
j.kerkenhoff@student.fontys.nl
2197164

Martijn Bonajo
m.bonajo@student.fontys.nl
2213297

Max van der Linden
max.vanderlinden@student.fontys.nl
2209349

René Karoff
r.karoff@student.fontys.nl
2198664

Ron Gebauer
r.gebauer@student.fontys.nl
2153294

Contents

Document information	1
1 Introduction	2
2 Handover Document	4
3 Project Plan	45
4 Quality Management Plan	64
5 Configuration Plan	78
6 Corporate Identity	90
7 Native vs. Hybrid	96
8 Class Diagram	102
9 Geofence Class Diagram	107
10 Entity Relationship Diagram	109
11 Meetings	111
12 Minutes	122
13 Scrum	140
14 Final Presentation	187
15 Important Code	200

Chapter 1

Introduction

This is the project dossier for the SoftwareFactory project of the GreenTech Lab also known as Tree Watch. As part of this document all the documents produced during the runtime of the project are included. Below you will find a brief explanation of what each included document is about.

The Project Plan is about the project itself. The initiation and the planning of the project in general are described. This document has been written by VAA ICT and was already produced before the start of the SoFa project. The document being produced by VAA is also the reason that it is written in Dutch

The Configuration Plan is the document created by the configuration manager. It shows dependencies, version control, change control and conventions.

The Corporate Identity document gives insight into how the TreeWatch company should profile itself. With for instance what color code and what icons.

The Class Diagram gives an overview of what classes are used within the app and how they are related to each other.

The Entity Relationship Diagram document explains the structure of the database which is used by the app.

The Native vs Hybrid document was produced to be able to decide if we were to develop each app separately (being android and IOS) or one hybrid app. The Native vs hybrid document is basically about weighing the pros and cons of each method of development.

The Handover Document is like a final report. It describes everything that anyone would need to know about the project in a summarized way.

The Meetings document is a bundle of all the notes which have been taken of the customer meetings. It gives insight into what was discussed during these meetings, who was present and when it took place.

The Minutes document is a bundle of the minutes which were taken every week. It shows presence, decisions and what to do for that week.

The Quality Management Plan is a plan about how to ensure the quality of the documents and products produced within the SoFa.

The Scrum plan contains all the planning and results of the development process.

The Final Presentation chapter contains the slides of the final presentation so that future groups can use the same style when presenting.

Chapter 2

Handover Document



HANOVER DOCUMENT 2015 SOFA GTL TREEWATCH PROJECT



Jan Kerkenhoff
Martijn Bonajo
Max van der Linden
René Karoff
Ron Gebauer

2016-01-17

Document information

Document name: Handover document

Document owner: TreeWatch

Company/Organisation: Fleuren Baarlo

Contact person: Max van der Linden, Group leader

Date: 2016-01-17

Place: Fontys University of Applied Science Venlo

Authors: Jan Kerkenhoff

j.kerkenhoff@student.fontys.nl

2197164

Martijn Bonajo

m.bonajo@student.fontys.nl

2213297

Max van der Linden

max.vanderlinden@student.fontys.nl

2209349

René Karoff

r.karoff@student.fontys.nl

2198664

Ron Gebauer

r.gebauer@student.fontys.nl

2153294

Acronyms

API Application program interface. II, 31, *Glossary*: Application program interface

app Application. IV, 2, 3, 6, 7, 15, 17, 19, 23, 24

GUI Graphical User Interface. II, 9, 24, *Glossary*: Graphical User Interface

IDE Integrated development environment. II, *Glossary*: Integrated development environment

KML Keyhole Markup Language. II, 20, *Glossary*: Keyhole Markup Language

SOFA Software Factory. II, 32, *Glossary*: Software Factory

XML Extensible Markup Language. II, 20, *Glossary*: Extensible Markup Language

Glossary

Agile Software Development refers to approaches in software development process, which should improve transparency and flexibility and result in a rapid onset of the developed systems, in order to minimize risks in the development process. The core idea is to keep the sub-processes as simple and thus moveable (= agile) Siepermann. 32

Application program interface is a software intermediary that makes it possible for application programs to interact with each other and share data. It's often an implementation of REST that exposes a specific software functionality while protecting the rest of the application. II, 31

Extensible Markup Language is a markup language to display hierarchic structured data in text files. II, 20

Geofencing is a feature in a software program that uses the global positioning system (GPS) to define geographical boundaries. A geofence is a virtual barrier. 20

Graphical User Interface is a user interface that includes graphical elements, such as windows, icons and buttons. II, 9

Integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development. An

IDE normally consists of a source code editor, build automation tools and a debugger. Most modern IDEs have an intelligent code completion. II

Keyhole Markup Language is a markup language based on XML used for geospatial data. II, 20

map tiles is part of a way maps are displayed digitally, each tile corresponds to a specific area of the map. Every map consists of multiple tiles that contain information for every zoom level. 19

Nuget is open source package manager for .Net projects developed by microsoft Homepage. 17

overlay is a way to display additional data on top of something else like a map. Can be a image or a shape or something else. 17, 20

polygon is a plane figure that is bounded by a finite chain of straight line segments closing in a loop to form a closed chain or circuit. 18

renderer component of Xamarin Forms, responsible to create the actual representation for the data for the device. 17

Scrum is a management and control process that cuts through complexity to focus on building software that meets business needs. Management and teams are able to get their hands around the requirements and technologies, never let go, and deliver working software, incrementally and empiricallyscrum.org. 32

Software Factory is the last project type module in the curriculum. To make it as real life as possible, a real customer is involved. The students work in groups, each group having its own customer and is using product specific technologiesEngineering. II, 32

Xamarin Forms is a framework developed by Xamarin Inc in 2014-05-28 to rapidly build crossplatform apps. It can be compiled to iOS, Android and Windows Phone from a shared code base. 17, 31, 32

Xamarin Studio, a standalone IDE for mobile app development, was released in February 2013 as part of Xamarin 2.0 and is based on the open source project MonoDevelop. In addition to a debugger, Xamarin Studio includes code completion in C#, an Android UI builder for creating user interfaces without XML, and integration with Xcode Interface Builder for iOS app design. 32

List of Figures

1	Milestones of the Whole Project	8
2	Sign in screen	15
3	What are you coming to do screen	15
4	Default screen with the map	16
5	Map screen with menu open	16
6	Map screen with an overlay	16
7	Screen where you can see to do list	16
8	Screen with the history of tasks	16
9	Screen where you can change settings	16
10	Block overlay of field Grotto	18
11	Heatmap inside of the iOS Application (app)	19
12	Geofencing class diagram	22

List of Tables

2	List of Stakeholders	4
3	Risks	10
4	Priorities & User Stories	14
5	Group roles	23
6	Document Review	25

Contents

Document information	I
Acronyms	II
Glossary	II
List of Figures	IV
List of Tables	IV
1 Introduction	1
1.1 The Customer	1
2 Project definition	2
2.1 Problem description	2
2.2 Scope	2
2.2.1 In Scope	3
2.2.2 Out of Scope	3
2.3 Stakeholders	3
3 Planning	5
3.1 Roles	5
3.2 Phases	6
3.3 Deliverables	6
3.4 Milestones	8
3.5 Projectplanning	9
3.6 Risk	9
4 Project analysis and prioritization	11
4.1 Planning and the desired results of the analysis phase	11
4.2 Results and their discussion within the Milestones	11
5 Requirements	13
6 Design	15
7 Implementation	17
7.1 Displaying a Map inside Xamarin Forms	17

7.2	Showing Overlays on the Map	17
7.3	Creating and Displaying HeatMaps	19
7.4	Importing KML files into the app	20
7.5	Geofencing	20
8	Quality management	23
8.1	Organization and Responsibilities	23
8.2	Quality Planning	23
8.2.1	Define Project Quality	24
8.2.2	Measure Project Quality	24
8.3	Quality Assurance	24
8.3.1	Analyze Project Quality	24
8.3.2	Improve Project Quality	24
8.4	Quality Control	26
9	Manual	27
9.1	Credentials	27
9.2	General setup	27
9.3	Getting the project running	27
9.4	Running on a Device	28
9.4.1	iOS	28
9.4.2	Android	28
10	Conclusion	29
11	Advice	30
12	Reflection	31
12.1	Max van der Linden	31
12.2	Martijn Bonajo	31
12.3	Rene Karoff	31
12.4	Jan Kerkenhoff	32
12.5	Ron Gebauer	32

1 Introduction

This document is about a project called Software Factory which has to be done during the 7th semester at Fontys University of Applied Science Venlo. The project brings Software Engineering and Business Informatics Students together to develop a solution for a real customer. In this case the customer is called Fleuren.

One goal of this project is to let the students act as the SoFa was their own company. This means the group works independent from the lecturers. Only the process coach keeps track of what the students do. In meetings which were scheduled on a weekly base, the process coach got information about the current status and if there were any problems.

1.1 The Customer

Fleuren is a tree nursery using high precision farming to grow fruit trees. After two or three years, depending on the variety, the trees are sold to companies which then harvest the fruits. Precision farming, in case of Fleuren, means that the trees are planted automatically by a tractor in rows of six trees of the same kind, this is then called a block.

2 Project definition

Fleuren has given the SoFa TreeWatch group the assignment to create a multi-platform app. The app should provide Fleuren with insight in what has been done to their fields and should give an overview of what still needs to be done to the fields. Using the app Fleuren should then also be able to make long term decisions based on the data shown. For instance when a certain field produces fruit trees of better quality it should be possible to look into the history of that field and compare that history to the history of a field which has produced lower quality trees. By looking at the differences Fleuren should be able to spot patterns in how to improve the quality of its trees.

TreeWatch project group will work on the project for five months. During this time the TreeWatch project group will try to fulfill as much requirements as possible. After the five month period VAA ICT Consultancy will take over and finish the project. The take over means that it will be important for the TreeWatch project group to properly document all the code.

2.1 Problem description

The main problem of Fleuren is that they started to notice that the quality of some fruit trees decreased due to human error. For instance one block of trees would get sprayed twice because something went wrong when registering which blocks have been sprayed. This is an unnecessary waste of resources.

Another problem Fleuren has, is that there is no clear overview of what happens when they change one of the steps within the production process. Therefore it is hard to improve the production and make decisions which affect long term production.

2.2 Scope

This section is about what is in scope of this project and what needs to be done afterwards by the VAA ICT Consultancy after they took over the project.

2.2.1 In Scope

This document is about analysis, design and implementation of a functional prototype of a mobile app running on iOS and Android.

2.2.2 Out of Scope

Not inside of the scope is anything which is needed to provide the app with data from external sources. This means servers or web services are not created during this project.

2.3 Stakeholders

This paragraph contains an overview of the most important stakeholder information of the stakeholders of this project. An overview will also be given to show the importance and influence of each stakeholder. In order to improve communication contact information is added. To clarify who benefits most from the success of the project, numbers have been added. The bigger the number, the higher the importance/influence of the stakeholder.

Stakeholder	Important notes	Profit from project success	Influence on project success	Contact information
TreeWatch group	Main developers	7	9	gtl@fontysvenlo.org
Han Fleuren	Owner of Fleuren Baarlo	8	4	directie@fleuren.nl
Yannick Smedts	Project leader of Fleuren	7	7	planning@fleuren.nl
Randy Wilbrink	Consultant of VAA	5	7	rwilbrink@vaa.com

Stakeholder	Important notes	Profit from project success	Influence on project success	Contact information
Jan Jacobs	Coach	3	5	jan.jacobs@fontys.nl

Table 2: List of Stakeholders

3 Planning

This chapter is about the overall planning of the TreeWatch project, which means it contains the roles each group member was elected for. Furthermore the overall planning contains the different project phases, the deliverables and milestones as well as the time-planning.

3.1 Roles

The following roles were assigned to the group members by the group. Every group member was asked to give his preferred role and since there were no conflicts everyone got the role which he preferred. Each role had its own responsibilities which are described in this subchapter.

Project Manager

This role was given to Max van der Linden as he studies Business Informatics which is a good basis for this role. The project manager was responsible for the communication with the customer and the planning of the project.

Quality Manager

The Quality Manager role was assigned to René Karoff. The Quality Manager was responsible for defining the quality standards which have to be met by all implemented features e.g. code-coverage in testing or coding style guides. Furthermore he took care of the requirements and that everything implemented matched them.

Scrum Master

The Scrum Master role was chosen by Ron Gebauer. The Scrum Master was had the task to set up the scrum environment and to solve problems. This means that, if there were problems which kept the group or a group member from working on the project he was about to solve it.

Configuration Manager

The Configuration Manager had to make sure that the development environment ran smooth. This means e.g. solve issues with the used IDE or used servers.

Main Software Engineer

The Main Software Engineer was responsible for the implementation. He made sure that coding was done. If someone struggled with an issue too long, he tried to help him out.

3.2 Phases

Since this project was done using scrum, the typical phases as analysis, design and implementation as used in the waterfall methodology do not fit to us. But still the project could be broken down to analysis, design, implementation and documenting sprints. Each sprint took two weeks.

Analysis sprints

During the analysis sprints the project was set up and the user stories were written and validated by the customer.

Design sprints

During the design sprints the first impression of a design was created using mockups.

Implementation sprints

These were used to implement the functional prototype. The implementation sprints were planned as the most time consuming sprints.

Documenting sprint

At the end of the project there will be a documenting sprint, which is used only to document the everything what was done, so the VAA ICT Consultancy is able to work on the application after the project was finished by the SoFa....

After every sprint a sprint-meeting with the customer was scheduled to give an update to him, clarify any misunderstandings and receive possible new ideas from the customer. With this meeting the customer was able to take part at the development of the app and could influence it e.g. when he wanted another feature with a higher priority than others. The customer also could introduce totally new ideas which should be implemented first. This is the main strength of an agile methodology like scrum.

3.3 Deliverables

This section describes the deliverables which had to be handed over to Fontys and/or the customer.

User Stories

The User Stories had to be defined by the TreeWatch group and the customer had to prioritise them.

Personal development plan

This document is about how every group member wants to improve during the project.

Personal competence plan

The Personal Competence Plan is a reworked personal development plan ex-

tended to s.m.a.r.t. details.

Mockup-Design

The Mockup-Design will be created by the TreeWatch group and provided to the customer to give him an idea of how the app could look like.

Functional prototype

The prototype is most time consuming deliverable and milestone.

Handover

This is the document which will be needed to start working on the app directly afterwards we finished the project. It contains what was done and what needs to be done to finalise the application.

3.4 Milestones

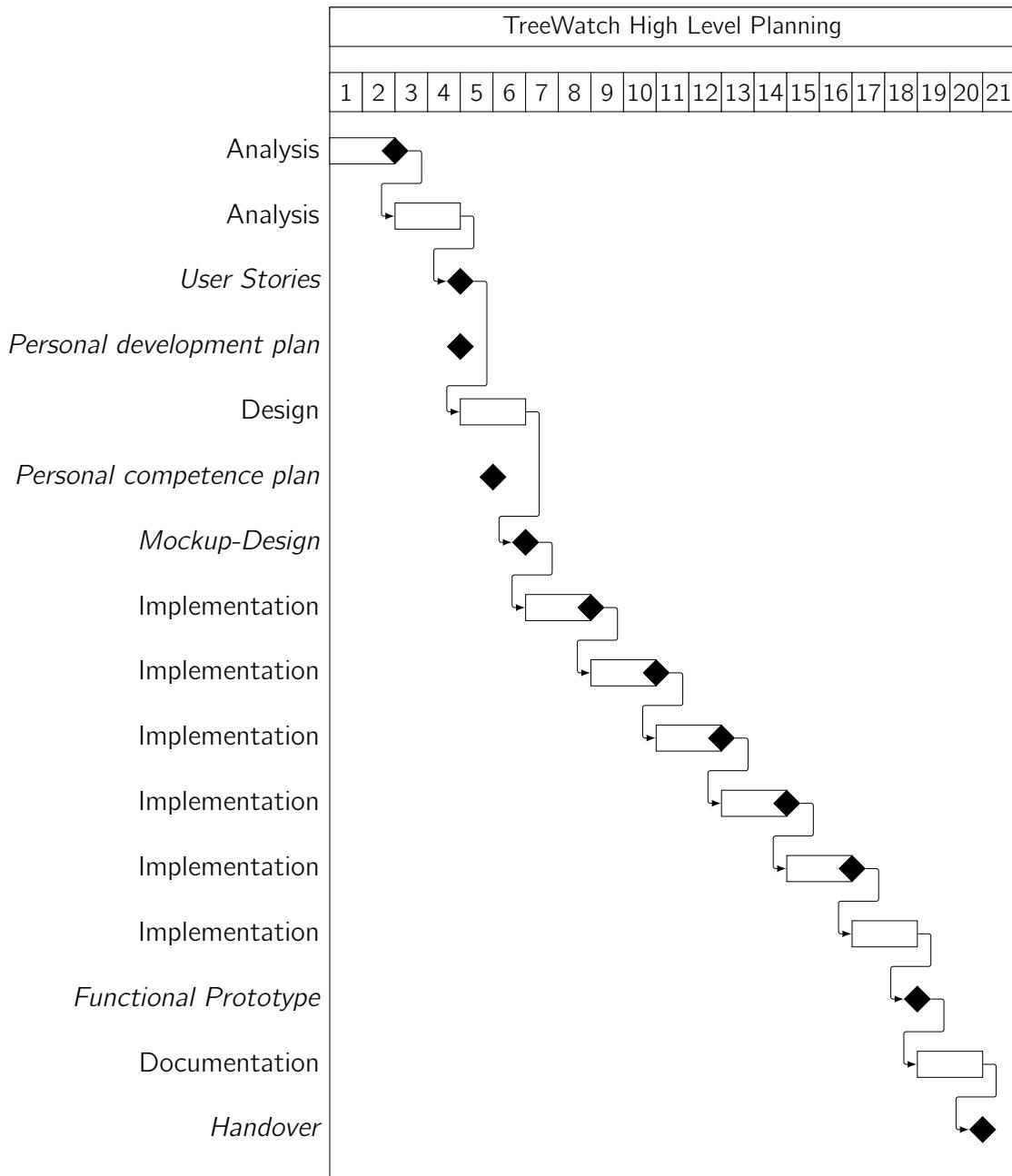


Figure 1: Milestones of the Whole Project

As shown in the figure 1, every two weeks there is a milestones. These milestones are the sprint meetings in which the customer gets to see the progress. Furthermore all the deliverables described in the chapter before are also milestones.

3.5 Projectplanning

As shown in figure 1 the SoFa group had 20 weeks of working time, it was planned that one sprint is two weeks long. Furthermore it was planned that the analysis would take two sprints in which the project was set up, the user stories were defined and prioritised. The design phase was planned to take only one sprint, since after a first design the group was able to start implementing anyways even if there were changes to the Graphical User Interface (GUI). The implementation, which was the main part of the project was planned to take six sprints and the documenting should take one sprint.

3.6 Risk

The following table contains the biggest risks the project will be facing. The severity and probability are defined for every known risk. These are graded on a scale of one to five with one having the lowest severity and five the highest. Also an action is defined which will be executed when a problem occurs. If the severity and probability are multiplied a multiplier is calculated. This multiplier helps determine which risks pose the most danger to the project.

ID	Risk	Severity	Probability	Multiplier	Action
1	Possible conflict of interest	4	4	16	Speak to involved parties to clear the problem and define direction of the project.
2	Xamarin licenses not available	2	4	8	Develop natively for both apps.
3	Customer not satisfied	4	2	8	Talk with the customer and define what the deficits are. Then communicate to VAA what needs to be changed.

ID	Risk	Severity	Probability	Multiplier	Action
3	No extra Mac-Books available	2	3	6	Let the group members who do not have a MacBook only program android.

Table 3: Risks

4 Project analysis and prioritization

Content of this chapter is everything that happens in the beginning of the project. What contains talks between us and the company for which the project is. Furthermore, out thinking about the project in the beginning, the analysis of what the user wants to get at the end and final the mockup planning of the menu structure.

4.1 Planning and the desired results of the analysis phase

The analysis phase started on 2015-08-31 and proceeded over a period of 2 milestones that were each 2 weeks long. The end of the stage was therefore the 2015-09-25.

In the first milestone the requirements of the application were examined. From the results various Epics and User Stories have been created, which have been prioritized by our contact person in Fleuren Baarlo and accordingly brought into order.

The second milestone was launched with finding ways of development and the currently used systems. After the systems were determined, various guidelines had to be created to determine how to be developed. Following mock-ups are created to represent the structure of the surface.

4.2 Results and their discussion within the Milestones

The results of the first milestone are in Table 4 of the 5. Section. While the results of the second milestone, which have been expanded in the design phase, can be found in Section 6.

At the beginning of the project a discussion with Yannick Smedts took place, our future contact person within Fleuren Baarlo, about their wishes to the application being created.

During the next few days some questions arose that were answered and discussed during a further meeting, what was extended with a visit to Van Den Borne aardapelen.

Finally, a last talk was organized in Fleuren Baarlo to clarify the recent ambiguities and thus successfully complete the analysis phase.

The talks took place as follows:

2015-08-31 Fontys University of Applied Science Venlo

2015-09-09 Fleuren Baarlo headquarter and on a field

2015-09-17 Van Den Borne aardappelen

2015-09-25 Fleuren Baarlo headquarter

5 Requirements

The requirements definition was started by setting a meeting with Fleuren. Here Fleuren explained what their problem was and what they wanted to see as functionality of the app. Based on that meeting a set of user stories was created. These user stories were then grouped in epics. The epics were then sent back to Fleuren to have them prioritized and checked for completeness.

The checked and prioritized user stories were then put in a table to show which functionalities needed to be implemented first and which ones were optional. The table of user stories can be found below. Within the table the word system is to be defined as the combination of the App and the database containing the data.

Priority	User Story
1	<p><i>As a user</i>, I want the ability to overlay the visualizations on top of the field.</p> <p><i>As a user</i>, I want to have my field represented graphically in the application.</p>
2	<p><i>As a user</i>, I want the application to register as much data as possible, automatically.</p> <p><i>As a user</i>, I want my GPS data entered into the system.</p> <p><i>As a user</i>, I want my weather data entered into the system.</p> <p><i>As a user</i>, I want my soil data gathered into the system.</p> <p><i>As a user</i>, I want my system data entered into the system.</p> <p><i>As a user</i>, I want to be able to take pictures of the trees.</p> <p><i>As a user</i>, I want to be able to upload pictures and tag them to a tree inside a certain row/block.</p> <p><i>As a user</i>, I want a system which visualizes the available data for the fields/blocks/rows/trees.</p> <p><i>As a user</i>, I want to correlate the available data in different views so that I can see the relations between different datasets.</p>
3	<p><i>As a user</i>, I want to digitize the kwekerij schrift (Nursery Script).</p>

Priority User Story

4	<p><i>As a user</i>, I want the application to contain different forms to transmit different information about the actual state of the trees on the field (e.g. brown leaves, thin stems...).</p> <p><i>As a user</i>, I want to be able to see a chronological ordering of events that happened on trees in a certain row/block's history.</p> <p><i>As a user</i>, I want to be able to see where and how long I worked on a field.</p> <p><i>As a user</i>, I want to be able to specify what work I am doing on a field/block/row/tree.</p>
5	<p><i>As a user</i>, I want the application to analyze the data of the trees and give hints and solutions for problems.</p>
6	<p><i>As a user</i>, I want to have a system that manages my customers, their orders and the amount of trees that are growing or are fully grown for them.</p>

Table 4: Priorities & User Stories

6 Design

This chapter is about the design of the TreeWatch project. The design part is necessary to give the customer an idea of what the application will look like and so that the customer is able to give feedback. The design also gives the group members an idea of what the application should look like when it is finished. This way the group members can work towards a visual goal.

First the user should sign in, so that data is only visible to an authenticated user, as can be seen in figure 2.

The main goal of the application is to show all the fields on a map, with overlays of certain data. This is visualised in figure 4 to figure 6.

Fleuren asked to have what they call "What are you coming to do here" as part of the application. As soon as an user enters a field the app will show a screen with the question "What are you coming to do" as seen in figure 3.

Finally there are three more screens, namely the to do list in figure 7, the history of tasks screen in figure 8 and lastly the settings screen in figure 9.



Figure 2: Sign in screen

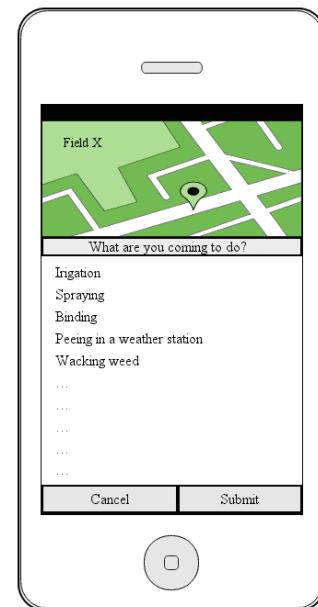


Figure 3: What are you coming to do screen

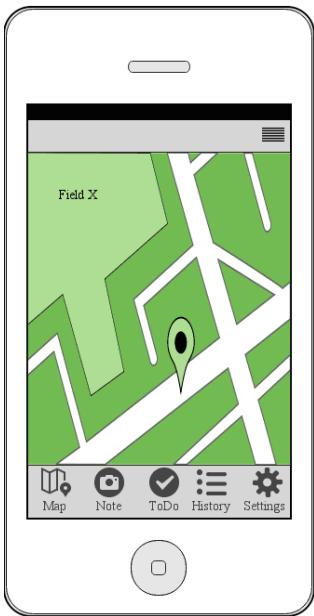


Figure 4: Default screen with the map

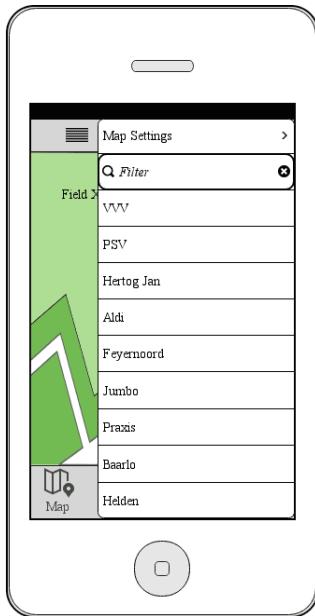


Figure 5: Map screen with the menu open

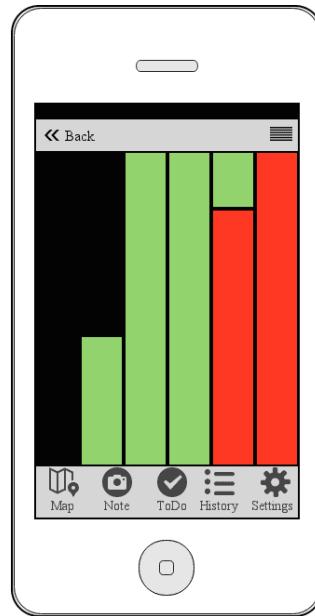


Figure 6: Map screen with an overlay

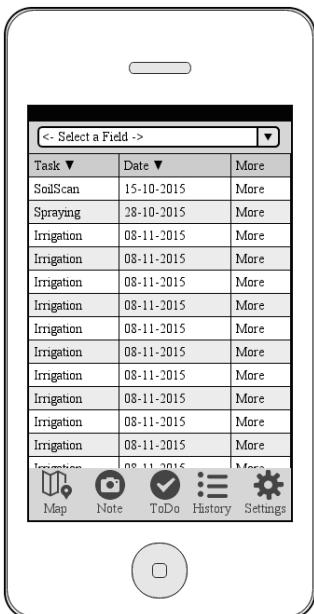


Figure 7: Screen where you can see to do list

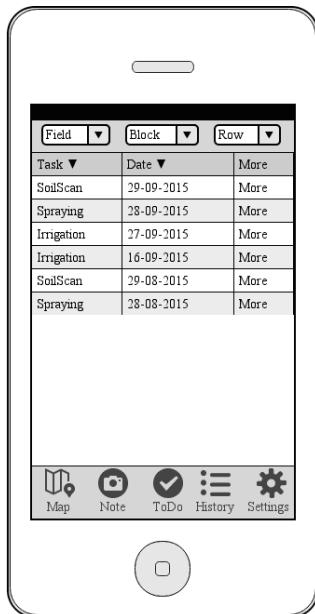


Figure 8: Screen with the history of tasks

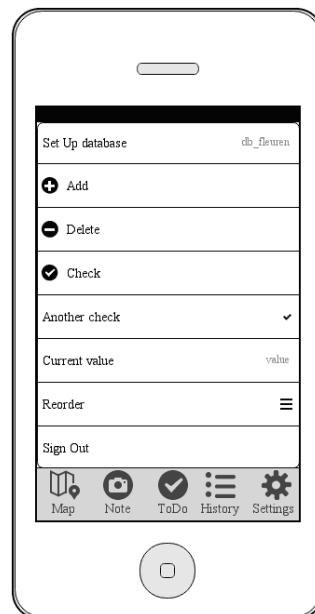


Figure 9: Screen where you can change settings

7 Implementation

This section gives you additional insight on part of the implementation and discusses challenges and problems encountered. Also gives information on the approaches to solve this issues.

7.1 Displaying a Map inside Xamarin Forms

One of the first things that was implemented in the app, after getting the basic structure of the app, was showing a map. The first implementation was really simple and just showed a map, like the default map app on a phone. Afterwards additional functionality was added.

Displaying a map in Xamarin Forms normally is really simple, as you can see on the Xamarin Forms Maps Xamarin-Inc. (a) site, using the Xamarin Forms map component is really simple. Just add a Nuget package to the project and then display a map with a few lines of code. This is fine if you don't need to extend it, but in our case we wanted to add additional functionality to it.

After determining that something more powerful was needed, different solutions to display the map were considered. Existing solutions like OsmSharp SharpSoftware, which provided a good looking map based on open street map, were considered. But they were focused on routing and didn't offer any of the features our app needed like adding custom overlays to a map. Also there are different solutions available that offer extensive server side rendering of maps containing your own data. A popular solution is Mapbox MapBox, but using a server backend for the app was out of scope for the project so a solution was chosen that would work on the device.

The only usable way we found was to implement our own custom map renderer on each platform. This renderer is overriding the default map renderer on each platform and then adds the custom functionality on top of it. This approach also meant that each feature related to the map needed to be implemented two times. Once for android and once for iOS, which mean an increase in development time. This solution worked well until we added extensive data to the map, when performance hits were noticeable.

7.2 Showing Overlays on the Map

After the map was implemented the next challenge we tackled was showing an overlay of the fields on the map. A field consist of two different parts, the outer boundary of the field and also many blocks on the field. The blocks represent six rows of Trees

of a specific species. But both are represented on the map as polygons, consisting of the points defining the outer boundaries of the field or block. All blocks of the same species should be easily recognisable so they are shown in the same color.

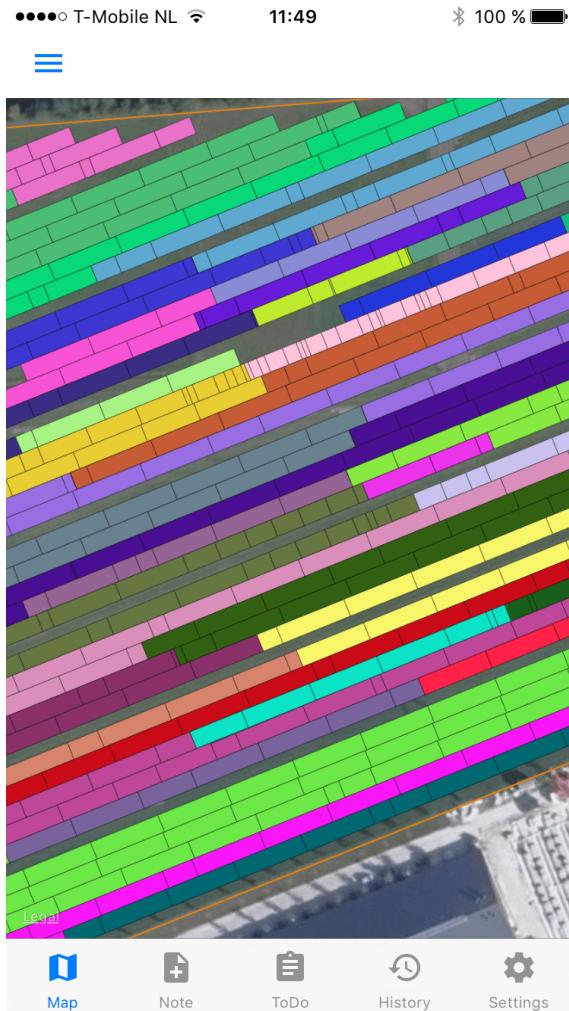


Figure 10: Block overlay of field Grotto

As you can see in figure 10 a field consists of a lot individual blocks, the amount depends on the size of the field but is somewhere in the several hundreds. This created some serious performance issues for the map. The map gets really unresponsive and slow after adding so many polygons to it. To reduce the stress on the map on iOS we added our own polygon implementation that could contain many single polygons. This reduced the work needed by the map enormously, since the polygons for one field are drawn in one single draw and not every polygon on his own. This made the

map usable again but also introduce a delay when the polygons need to be shown by the map. But this is the only working workaround for the performance problems. Further performance improvements could only be achieved by moving the map drawing to a server and only getting pre rendered map tiles to display from it.

7.3 Creating and Displaying HeatMaps

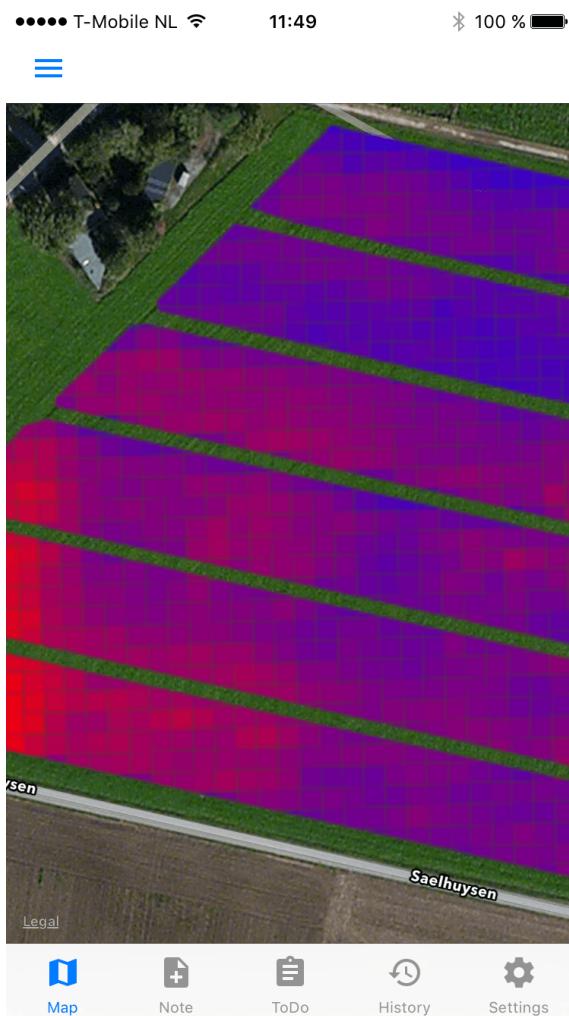


Figure 11: Heatmap inside of the iOS app

Another feature that was implemented for the map was the display of heat maps, these should show data like a soil scan that is not bound to specific blocks but more to specific points on the field.

The first try of implementing heat maps included displaying an image overlays created from the data and putting that on top of the map. For this approach we found a working Objective C Library called LFHeatMap Polak. To use it in our project we created a C# Binding for it. This was really easy to do following the tutorial on the Xamarin site Xamarin-Inc. (b). But this created heat maps based on single weighted points. Since we only got heat map data from Fleuren that contained of polygons with data, we looked for another solution.

As you can see in figure 11 we implemented the heat map as just a bunch of polygons. The colors of the polygons were calculated based on their data. The color moves from solid blue for the lowest values to solid red for the highest values. The heat map displayed consists of about 4000 different polygons. The approach therefore is more a proof of concept since the drawing on the map takes several seconds and doesn't even work on android.

7.4 Importing KML files into the app

Fleuren supplied us with various datasets from their fields, to test our implementation with real data, especially the amount of data, we implemented a mechanism to read these files. The data we got was mostly shapefiles, since there is no easy way to read these we converted them to Keyhole Markup Language (KML) files. KML is basically just a version of Extensible Markup Language (XML) which is designed for geospatial data. To convert the files we used the .Net xml parsing library.

7.5 Geofencing

The Geofencing part of the application is based on the plugin which can be found at <https://github.com/domaven/xamarin-plugins/tree/master/Geofence>. This plugin has most of the functionality that is required in the project, however the plugin is outdated and doesn't work anymore.

To make the most out of Xamarin, most of the work should be done in the Forms application, this way code is reused for both Android and iOS. The class diagram on the next page shows what classes are in the Forms application.

The IGeofence and IGeofenceStore are interfaces that should be implemented in a platform specific way, because they contain the actual geofences and their platform specific way of working. The IGeofenceStore interface has some methods that can be shared between the platforms, therefor an abstract class BaseGeofenceStore is created, that has these methods already implemented.

The CrossGeofence class is responsible for creating the actual platform specific im-

lementation of the IGeofence interface. This is done by using DependencyService to get the actual platform specific implementation.

The platform specific code calls the methods in the CrossGeofenceListener class inside the Forms project, this way state changes can be handled in a universal way, instead of platform specific.

<<interface>> IGeofence <p>Regions : IReadOnlyDictionary<string, GeofenceCircularRegion> GeofenceResults : IReadOnlyDictionary<string, GeofenceResult> IsMonitoring : bool LastKnownLocation : GeofenceLocation</p> <p>StartMonitoring(region : GeofenceCircularRegion) : void StartMonitoring(regions : IList <GeofenceCircularRegion>) : void StopMonitoring(id : string) : void StopMonitoring(ids : IList<string>) : void StopMonitoringAllRegions() : void</p>	<<interface>> IGeofenceListener <p>OnMonitoringStarted(id : string) : void OnMonitoringStopped() : void OnMonitoringStopped(id : string) : void OnRegionStateChanged(result : GeofenceResult) : void OnError(error: string) : void</p>	<<interface>> IGeofenceStore <p>GetAll() : Dictionary<string, GeofenceCircularRegion> Get(string id) : GeofenceCircularRegion Save(GeofenceCircularRegion region) : void RemoveAll() : void Remove(string id) : void</p>
<<abstract>> BaseGeofenceStore <p>#GetFieldKey(string id, string fieldName) : string</p>		
GeofenceLocation <p>Latitude : double Longitude : double Date : DateTime Accuracy : double</p>		

CrossGeofence <p>Implementation : IGeofence GeofenceListener : IGeofenceListener IsInitialized : bool Current : IGeofence</p> <p>Initialize(GeofencePriority geofencePriority, float smallesDisplacement) : void CreateGeofence() : IGeofence</p>	CrossGeofenceListener <p>OnMonitoringStarted(- string region) : void OnMonitoringStopped(- string region) : void OnMonitoringStopped() : void OnError(string error) : void OnRegionStateChanged(GeofenceResult result) : void</p>	GeofenceCircularRegion <p>Id : String Latitude : double Longitude : double Radius : double NotifyOnEntry : bool NotifyOnStay : bool NotifyOnExit : bool</p>
--	--	--

Figure 12: Geofencing class diagram

8 Quality management

This chapter describes the necessary information needed to manage the project quality from the project planning to the delivery to the customer. Within this document the quality policies, procedures, roles, responsibilities and authorities are defined. At the highest level quality management involves planning, doing, checking, and acting to improve project quality standards. Project quality management is therefore split into three categories: quality planning, quality assurance and quality control.

8.1 Organization and Responsibilities

Name	Role	Quality Responsibility
Max van der Linden	Project Manager	External communication, Auditing
Martijn Bonajo	Configuration Manager	Infrastructure, source files, Software engineering documentation, auditing
Ron Gebauer	Scrum Master	Scrum planning, auditing
Rene Karoff	Quality Manager	Ensure use of quality guidelines, auditing
Jan Kerkenhoff	Main Engineer	Main Code Master, auditing

Table 5: Group roles

8.2 Quality Planning

Since this project is about programming a mobile business app, it is highly important that the system has a minimum of bugs/errors. A good documentation is necessary too, not only because this app will be developed by another SoFa group after our semester is over, but the customer should understand the system which is developed by us.

8.2.1 Define Project Quality

To ensure to quality of the written code it is mandatory to develop every module test-driven, therefore unit-testing is introduced. The written documents will be checked for its grammar, writing style and content.

8.2.2 Measure Project Quality

Most of the code of this app was tested, but since we have a lot of GUI related code which we weren't able to run unit tests with, we were not able to achieve the former stated goal of a code coverage of 90%.

Furthermore the GUI was tested using the Xamarin Testcloud which was included in our Academic license which represents the Xamarin Business license. Since we had the Business license, we were allowed to use the Xamarin Testcloud for one hour per month. But since we had 5 licenses this expanded to 5 hours per month.

In table 6 it is described who reviewed which document and with what rank it was approved.

The code of the app will be tested. Therefore it is a goal to get at least 90% code coverage, better 100%. Each evaluation criterion of the written documents will be ranked from “–“ to “++” (“–“ too bad – “++” excellent).

8.3 Quality Assurance

Since the code is tested, this will show the group if the quality goal is achieved. Furthermore the written documents will be audited by at least one group member and the project quality manager.

8.3.1 Analyze Project Quality

The tool to measure the code coverage (**which still needs to be defined**) will show the programmer which code is still uncovered, and the writer of a document will receive feedback of the auditing persons, so he can improve on his writing too.

8.3.2 Improve Project Quality

As stated in the project management plan, proper requirements were created to increase the project quality. This was done during two sprints, where the customer received the user stories, prioritized them and gave feedback to the group in the first

Document name	Author	Reviewer 1	Rank 1	Reviewer 2	Rank 2
User Stories	Max van der Linden	René Karoff	++	Martijn Bonajo	+
Personal development plan	Personal	Group work	René Karoff	++	Max van der Linden
competence plan	Mockup	Design			++
Functional Prototype	Handover				

Table 6: Document Review

sprint. During the 2nd sprint the user stories were reworked due to the remarks the customer gave to the group.

8.4 Quality Control

At the end of the project each deliverable will be audited again, by every team member to ensure the project quality.

9 Manual

9.1 Credentials

To continue working on this project you need the following credentials:

- Access to the Github repository
- License for Xamarin
- Optional: Google Account that contains the API Key

9.2 General setup

If you want to checkout the project read access to the repository and a trial xamarin license is fine.

Important

If you do not have access to the Google account that belongs to the API key used, then you need to exchange the GoogleMaps API Key in the 'AndroidManifest.xml', which is explained in detail later on.

9.3 Getting the project running

- Install Xamarin Studio <https://xamarin.com/download>
- Install Xcode and the latest iOS SDK
- Install Xamarin Android Player <https://xamarin.com/android-player>
- Download an android device in Xamarin Android Player
- Install Google Play Services for the emulator <https://university.xamarin.com/resources/how-to-install-google-play-on-android-emulator>
- Checkout git repository
- Open project in Xamarin Studio
- Add your android app fingerprint to the Google Maps API key:

How to determine your MD5 or SHA1 signature https://developer.xamarin.com/guides/android/deployment,_testing,_and_metrics/MD5_SHA1/

This https://developers.google.com/maps/documentation/android-api/signup#get_an_android_api_key explains how you can add a fingerprint to an existing key or how to create a new key and add it to the app.

If you want to exchange the key, the android manifest is located at `Droid/Properties/AndroidManifest.xml`

- You can now run the project from Xamarin Studio on an emulator

9.4 Running on a Device

9.4.1 iOS

- Get a valid development certificate for iOS development
- Get a valid provisioning profile for the app including the device you want to run on
- Connect the device
- Select the device inside of Xamarin Studio
- Deploy the app to the device

9.4.2 Android

- Connect the device
- Select the device inside of Xamarin Studio
- Deploy the app to the device

10 Conclusion

The TreeWatch app has come a long way but still has a long way to go. A lot of the basic functionality has been implemented. The app can already be used to visualize where all the fields are located on the map and is also able to show overlays such as the block data and biomass.

VAA ICT Consultancy will need to continue the development of the app in order to make it practically usable. This means that a web based database needs to be added which contains all the info. For now the information about the fields is still stored locally on the device. Furthermore also the history and todo parts of the app should be implemented.

11 Advice

Because of performance issues that occur when the app has to calculate the overlays of the fields and the blocks, it is recommended that the server should draw all the polygons and then send it as an image to the phone. Then the phone should be able to overlay this image on the correct position. This would greatly improve the speed of the app.

12 Reflection

12.1 Max van der Linden

As the project leader of the SoFa TreeWatch group I learned a lot about what it means to lead a project. This meant that i had to make sure that everyone follows the rules and sticks to the deadlines. One of the tasks of the project leader was also to be the main contacting point to and from the customer. From this i learned a lot about communicating in a professional way.

All in all in my opinion a good and useful product form which a lot was learned.

12.2 Martijn Bonajo

I learned a lot from this project. For me it was the first time creating a multi platform application. This meant learning how to work with Xamarin, using MVVM on the Xamarin Forms and programming to platform specific Application program interface (API)s.

As configuration manager I was responsible for making sure everyone could run all the software that was needed during the project.

12.3 Rene Karoff

During this project my role was the project quality manager. This means that my responsibility was to define the quality standards and to make sure that the defined standards were met during the project. This went well most of the time, but since I study Business Informatics I am not as experienced in coding as the Software Engineering students, I wasn't always able to push the group to the test-driven way of developing the features. As the project quality manager i put myself in the role of reviewing every document and implemented user story. During the next project I would define that rule different, as that the reviewer switches. This should lead to a higher quality since more eyes will see more errors.

All in all my quality management skills really improved during this project and I was also able to contribute code to the project which improved my coding skills too.

The experience of working for a real customer was really nice, especially because we had a customer where it was pleasant to work with. The customer was able to define his needs in a understandable way and because of that I also learned something about precision farming.

12.4 Jan Kerkenhoff

I learned a lot about cross platform development and also about C# development. It was my first time developing a mobile app and also never did work on anything cross platform. I gained a lot of knowledge about Xamarin and Xamarin Forms. As the main developer is was responsible for solving all programming related challenges, pick out the frameworks and libraries we used and helping others out if they were stuck. Also kept an eye on the Test Driven Development process together with René. This thought me a lot about writing tests and challenges when doing TDD.

12.5 Ron Gebauer

In this project I learn a bunch of stuff. As an example I worked the first time with Xamarin Studio and was a part in the creation of an cross platform application. Furthermore I learned a lot from our customer, which creates an really interesting atmosphere inside the meetings. In addition to that I got more and more confirm with Scrum, which was our Agile Software Development method.

Finally I will say that this Software Factory (SOFA) one of the most interesting and helping modules was in my study.

References

- D. M. Siepermann. Agile softwareentwicklung. Version 5. [Online]. Available: <http://wirtschaftslexikon.gabler.de/Archiv/381707695/agile-softwareentwicklung-v5.html>
- scrum.org. A better way of building software. Webpage. [Online]. Available: <http://www.scrum.org/>
- F. V. D. S. Engineering, *Module description for SOFA Scrum project for a real software customer*, 2012th ed., Fontys Hogeschool Techniek en Logistiek. [Online]. Available: <http://sofa.fontysvenlo.org/pdffiles/md-sofa-en.pdf>
- Xamarin-Inc. Xamarin forms maps developer guides. [Online]. Available: <https://developer.xamarin.com/guides/cross-platform/xamarin-forms/user-interface/map/> Accessed: 2015-12-18
- SharpSoftware. Osmsharp website. [Online]. Available: <http://www.osmsharp.com/> Accessed: 2015-12-18
- MapBox. Mapbox website. [Online]. Available: <https://www.mapbox.com/> Accessed: 2015-12-18
- G. Polak. Lfheatmaps heatmap generator library. [Online]. Available: <https://github.com/gpolak/LFHeatMap/> Accessed: 2015-12-18
- Xamarin-Inc., "Walkthrough: Binding an objective-c library." [Online]. Available: https://developer.xamarin.com/guides/ios/advanced_topics/binding_objective-c/Walkthrough_Binding_objective-c_library/

Chapter 3

Project Plan



PLAN VAN AANPAK

Perceel Online App

Abstract

Ontwikkeling van een App voor Smartphones en Tablets die het mogelijk maakt om in het veld te kunnen beschikken over actuele en historische data van de teelt en het perceel zelf.

De App maakt het mogelijk om informatie te raadplegen over bijvoorbeeld gewasbeschermingsmiddelen en waarnemingen en bewerkingen op een perceel te registreren.

Randy Wilbrink
RWilbrink@vaa.com



PLAN VAN AANPAK



Auteur : Randy Wilbrink
In opdracht van : Boomkwekerij Fleuren
Datum : 21 juli 2015
Referentie : Innovatieprogramma Boomkwekerij
Versie : 0.1



Rietbeemdenborch 10
5241 LG Rosmalen

T +31 (0) 73 523 32 32
F +31 (0) 73 523 32 33

W www.vaa.com
E vaa@vaa.com



INHOUDSOPGAVE

INHOUDSOPGAVE	4
1 ALGEMEEN.....	6
1.1 Versie Informatie	6
2 VOORWOORD.....	6
3 MANAGEMENT SAMENVATTING.....	6
4 INTRODUCTIE.....	6
5 AANLEIDING.....	7
5.1 Accordering en bijstelling	7
6 PROJECTOMSCHRIJVING	7
6.1 Projectomgeving	8
6.2 Doelstelling project	8
6.3 Op te leveren producten en diensten	8
6.4 Eisen en beperkingen	9
7 AANPAK 9	
7.1 Vooronderzoeksfase.....	9
7.2 Plan van Aanpak.....	10
7.3 User Stories.....	10
7.4 Use Cases	10
7.5 Aanpassingen doen aan registratieplatform	10
7.6 Realisatie van de App	11
7.7 Test & rework fase	11
7.8 Opleverfase.....	11
7.9 Communicatiefase	12
7.10 Documentatiefase	12
8 KNELPUNTEMANALYSE	12



9 PLANNEN 13

9.1 Communicatieplan	13
9.1.1 Doelgroep	13
9.1.2 Doelstelling	13
9.1.3 Boodschap	14
9.1.4 Strategie	14
9.1.5 Communicatiemiddelen	14
9.1.6 Actieplannen.....	14
9.2 Financieel plan	15
10 KWALITEITSBORGING.....	15
10.1 Productkwaliteit.....	15
10.2 Proceskwaliteit	15
10.3 Voorgestelde maatregelen	16
10.4 Maatregelen ter verificatie en validatie.....	16
11 OVERIGE PLANNEN	16
12 BIJLAGEN 17	
12.1 Bijlage 1: Begrippen en definities	17
12.2 Bijlage 2: Projectplanning.....	18



1 Algemeen

1.1 VERSIE INFORMATIE

Versie	Datum	Omschrijving	Auteur	Gefiatteerd (door)
0.1	21-7-2015	Uitgebreide versie van het PvA	Randy Wilbrink	

2 Voorwoord

Het Innovatieprogramma Boomkwekerij is een van de laatste projecten waarvoor via het Productschap Tuinbouw sectorgelden beschikbaar zijn gesteld. Binnen het innovatieprogramma kunnen kleinschalige reeds bestaande innovatieprojecten in aanmerking komen voor subsidie om de laatste stap naar de praktijk te zetten.

3 Management samenvatting

Uit een ronde-tafel brainstormsessie met branchegenoten komt naar voren dat er onder hen grote behoefte bestaat om in het veld via een Mobile Device (Smartphone of Tablet) te kunnen beschikken over actuele en historische data van de teelt en het perceel dat tevens de mogelijkheid biedt waarnemingen te registreren.

Hiermee wil men:

- a) Bij vele verschillende teelten/rassen/partijen per perceel, effectief personeel/loonwerker/teeltadviseur van derden naar de juiste partij/ras/teelt sturen binnen het perceel om hierop een bewerking uit te voeren. (navigeren)
- b) Gegevens tonen van eerdere bewerkingen zoals een GBM toediening. As-Applied informatie van sputtachine als kaartlaag over het perceel (informeren)
- c) Tijdens het scouteren naar aantastingen een GBM actie plannen en zien welk middel actueel is toegelaten en onder welke randvoorwaarden. (informeren/plannen)
- d) Tonen van informatie uit bodem en gewas zoals bodembemonstering, bodemscans, opbrengstmetingen, satellietbeelden etc...
- e) Vastleggen van waarnemingen op betreffende locatie in het veld (scouteren)

De innovatie bestaat uit het ontwikkelen van een Smartphone App die bovengenoemde functionaliteiten biedt. De App is door iedereen gratis te downloaden en wordt gekoppeld aan een gebruikersaccount op een bestaand registratieplatform waarmee synchronisatie plaatsvindt. Dit platform is gebaseerd op open standaarden en maakt gegevensuitwisseling mogelijk met derden zoals telersverenigingen, adviesorganisaties en bedrijfsmanagement systemen (BMS'en). De basisversie is hierbij gratis toegankelijk, aan specifieke functionaliteiten is een verdienmodel gekoppeld voor onderhoud en ondersteuning van de App.

4 Introductie

Naar aanleiding van informatie uit eerdere gesprekken met o.a. Boomkwekerij Fleuren in het kader van de Kenniscirkel Precisielandbouw Boomkwekerij in de jaren 2011-2012 is na het openstellen van het Innovatieprogramma Boomkwekerij opnieuw contact opgenomen met een aantal boomkwekers die aan deze kenniscirkel hebben deelgenomen. Uit dit hernieuwde contact kwam de behoefte duidelijk naar voren om binnen het Innovatieprogramma Boomkwekerij een oplossing uit te werken voor enkele problemen waar de bedrijven in de uitvoering van hun werkzaamheden tegenaan lopen.



Omdat een projectaanmelding voor het Innovatieprogramma Boomkwekerij volgens de voorwaarden van het innovatieprogramma ingediend moet worden door een ondernemer in de boomkwekerijsector is Boomkwekerij Fleuren BV in overleg aangesteld als aanvrager van het projectvoorstel.

5 Aanleiding

Het bedrijf loopt bij de uitvoering van haar dagelijkse werkzaamheden tegen een aantal problemen aan die te maken hebben met de planning en toepassing van Gewasbeschermingsmiddelen (GBM), het inzichtelijk maken van oorzaak-gevolg als het gaat om de detectie van (teelt)problemen in het veld en het eenduidig sturen van eigen medewerkers of met name derden (loonwerkers, adviseurs) naar een specifieke partij plantgoed binnen een perceel. Daarnaast is het doeltreffend scouten en vastleggen van waarnemingen in het veld een wens die vele telers gaarne gerealiseerd willen zien.

De belangrijkste problemen:

- 1) Een belangrijk kenmerk van het bedrijf is dat er gewerkt wordt met vele verschillende rassen/partijen binnen een perceel. Deze rassen/partijen zijn vaak klein tot zeer klein in aantal en daarmee ook in oppervlakte. Een groot risico doet zich voor in de situatie dat er bespuitingen moeten worden uitgevoerd op alleen een select gedeelte van het perceel (partij of ras). Een derde partij kan zich vergissen in de locatie en bespuit de verkeerde partij/ras. Dit moet voorkomen kunnen worden en ook traceerbaar zijn.
- 2) Traceerbaarheid voor problemen die zich later in de teelt voordoen is zeer wenselijk. Bijvoorbeeld een bespuiting die met iets lagere dosering uitgevoerd wordt om uit te komen met de oppervlakte van het perceel. Later blijkt er toch een aantasting op te treden. Men wil graag kunnen herleiden wat de dosering is geweest op welke locatie en aanvullend bijvoorbeeld kunnen zien wat de instellingen waren ten aanzien van druppelgrootte, gebruikte spuitdruk etc. Moderne spuitmachines produceren een logbestand met GPS coördinaten dat nu niet wordt gebruikt in de praktijk.
- 3) Momenteel is er geen actueel overzicht beschikbaar dat oproepbaar is in het veld over welk GBM is toegelaten in de huidige teelt. Het plannen van een GBM actie n.a.v. geconstateerde aantastingen in het veld vraagt een hoop stappen die nu op kantoor moeten worden uitgevoerd om te checken a.d.v. regelgeving en reeds toegepaste hoeveelheden van een GBM in de teelt. Dit leidt tot fouten en potentieel strijdige acties met regelgeving wanneer een GBM actie niet zorgvuldig gepland wordt.

5.1 ACCORDERING EN BIJSTELLING

De kerntekst van dit Plan van Aanpak wordt vóór 24 augustus voorgelegd ter beoordeling en goedkeuring aan de Raad voor de Boomkwekerij (per e-mail aan info@raadvoordeboomkwekerij.nl). Na een definitieve beoordelingsronde in september 2015 volgt het bericht of de projectaanvraag wel of niet is gehonoreerd.

6 Projectomschrijving

Binnen dit project wordt er een App ontwikkeld, geschikt voor Smartphone en Tablet die gelinkt wordt aan een gebruikersaccount op het bestaande registratieplatform 1xregisteren.nl. De App bevat gratis basisinformatie die door iedere praktijkterler gebruikt kan worden. In combinatie met het commerciële gebruikersaccount bij 1xregisteren.nl wordt er aanvullende bedrijfsspecifieke functionaliteit aan de App toegevoegd en heeft er synchronisatie plaats van bedrijfsgegevens tussen platform en App.

Zowel het platform 1xregisteren.nl als de nieuw te ontwikkelen App maken gebruik van open standaarden voor gegevensuitwisseling met derden zoals overheden, telersverenigingen, adviesorganisaties en bedrijfsmanagementsystemen. Door het gebruik van web-services is het platform voorbereid op het aansluiten van toekomstige nieuwe informatiebronnen.

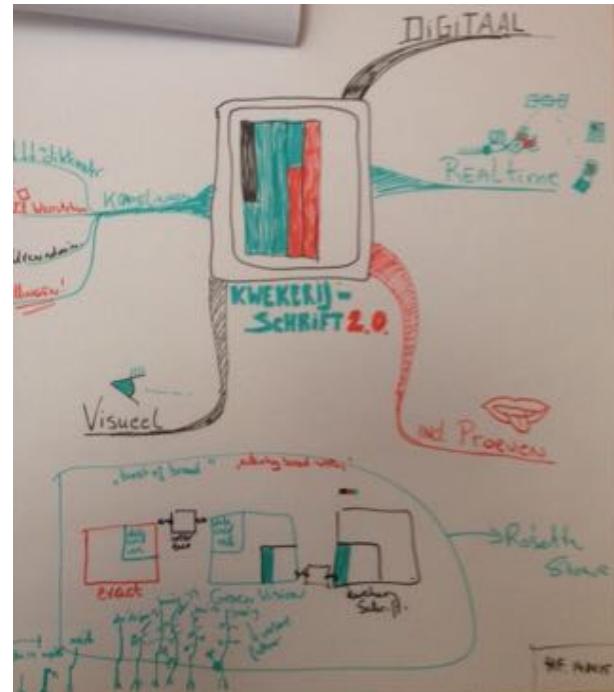
6.1 PROJECTOMGEVING

De App richt zich op het ondersteunen van de primaire bedrijfsprocessen op het boomkwekerijbedrijf door het ontsluiten en registreren van voor de teelt cruciale gegevens in het veld via de smartphone of tablet.

Cruciale informatie is afkomstig uit diverse bronnen en is op de productielocatie, in het veld, slecht te raadplegen. Informatie uit digitale bronnen is op kantoor prima te raadplegen maar niet of nauwelijks in het veld. Dit komt omdat veel informatiebronnen hun gegevens slecht of niet kunnen ontsluiten naar in het veld bruikbare apparaten zoals smartphones of tablets.

Omgekeerd geldt dat waarnemingen en bewerkingen in het veld slechts via papier naar kantoor getransporteerd worden en aldaar optioneel worden overgenomen in een digitale bron. Met alle risico's op gegevensverlies en gemis aan overzicht van dien.

Figuur 1 laat het landschap zien waarbinnen de ontwikkeling van deze App een rol speelt.



Figuur 1 Landschap voor de te ontwikkelen app

6.2 DOELSTELLING PROJECT

Om duurzaam en milieubewust om te gaan met gewasbeschermingsmiddelen en traceerbaar acties uit te voeren, evenals het vermijden van fouten door het missen van cruciale informatie wil een boomkweker in het veld:

- Teeltkundige beslissingen kunnen onderbouwen aan de hand van het inzien van gegevens van een historische bewerking die is uitgevoerd op het perceel, zoals de geregistreerde dosering door de computer van een spuitmachine, maar ook de uitslag van een bodembemonstering of -scan.
- Aan de hand van geconstateerde aantastingen in een perceel een gewasbeschermingsactie kunnen inplannen en daarbij ter plaatse kunnen inzien welk gewasbeschermingsmiddel actueel is toegelaten en onder welke voorwaarden.
- Kunnen vastleggen van waarnemingen en registreren van uitgevoerde acties op de betreffende locatie.
- Bij vele verschillende teelten/rassen/partijen per perceel, effectief personeel/loonwerker/teelt-adiseur van derden naar de juiste partij/ras/teelt sturen binnen het perceel om hierop een bewerking uit te voeren.

6.3 OP TE LEVEREN PRODUCTEN EN DIENSTEN

Binnen het project wordt een App ontwikkeld voor Smartphone en Tablet devices die draaien op iOS of Android. De App wordt kosteloos beschikbaar gesteld via de Apple Appstore en de Google PlayStore.

Met deze App kan:

- een niet geregistreerde gebruiker:



- per teelt actuele informatie inzien welk gewasbeschermingsmiddel is toegelaten in zijn teelt en onder welke voorwaarden. Hierbij heeft deze gebruiker ook toegang tot actuele etiketinformatie waaronder dosering en veiligheidsinformatie.
- na het koppelen van de niet-geregistreerde App aan zijn e-mail adres, via dit e-mailadres gemachtigd worden door een geregistreerde gebruiker om perceelskaarten in te zien met daarbinnen informatie over eventuele aanwezige teelten, rassen of partijen. De eigen geografische locatie is hierbij zichtbaar op deze kaart.
- een bij 1xregistreren.nl geregistreerde gebruiker kan alle functionaliteit gebruiken die een niet geregistreerde gebruiker kan gebruiken, zoals hierboven is omschreven. Daarnaast kan een geregistreerde gebruiker:
 - teeltkundige beslissingen onderbouwen aan de hand van het inzien van gegevens van een historische gewasbeschermingsactie die is uitgevoerd op het perceel. Hierbij kan de gebruiker binnen de App een kaartlaag over het perceel projecteren die de geregistreerde (As-Applied) dosering van een spuitmachine-computer inzichtelijk maakt. Deze As-Applied gegevens vallen binnen het huidige groeiseizoen en zijn geupload via 1xregistreren.nl;
 - waarnemingen vastleggen en uitgevoerde acties registreren op de huidige geografische locatie. Deze waarnemingen hebben de vorm van een notitie die gekoppeld is aan de geolocatie. Uitgevoerde acties worden gekozen uit een lijst van acties die de gebruiker kan definiëren voor zijn bedrijf binnen zijn account-instellingen op 1xregistreren.nl;
 - een derde machtigen via het e-mailadres van die derde om een of meerdere perceelskaarten in te zien met daarbinnen aangegeven eventueel aanwezige teelten, rassen of partijen.
 - een geregistreerde derde machtigen via het e-mailadres van die geregistreerde derde om uitgevoerde bewerkingen op een perceel van gebruiker te registreren.
 - Een machtiging ontvangen van een geregistreerde derde om door gebruiker uitgevoerde bewerkingen op een perceel van die derde te registreren.

Bestandsformaten van As-applied gegevens die kunnen worden geupload op 1xregistreren.nl zijn van het formaat: txt, csv en iso-xml

De App wordt volledig nieuw ontwikkeld op basis van bovenstaande functionaliteiten maar sluit aan op bestaande functionaliteit van het online registratieplatform 1xregistreren.nl

6.4 EISEN EN BEPERKINGEN

De onder 6.3 genoemde functionaliteit wordt uiterlijk opgeleverd op 30 juli 2017. Aan de Raad voor de Boomkwekerij wordt voor 30 september 2017 een beschrijvend eindverslag geleverd.

7 Aanpak

Binnen het project zijn de volgende fases te beschrijven om te komen tot het gewenste resultaat. Daarnaast wordt een indicatie gegeven van de tijdsperiode waarin de fase voltooid wordt.

7.1 VOORONDERZOEKSFASE

Binnen de vooronderzoeksfase worden de behoeften van de boomkwekerijsector onderzocht om zo te komen tot de juiste probleemstelling en het kunnen inventariseren van behoeften en prioriteiten. Concreet wordt deze fase ingevuld door het voeren van ronde-tafel-gesprekken met branchegeleden binnen de (fruit)boomkwekerij en vaste plantenteelt sector. Door middel van deze ronde-tafel-gesprekken worden de behoeften en prioriteiten uit de sector geïnventariseerd.

Op basis van de gesprekken wordt een probleemstelling gedefinieerd en een requirementsanalyse opgesteld.

De vooronderzoeksfase loopt van week 20/2015 tot en met week 43/2015



7.2 PLAN VAN AANPAK

Nadat de probleemstelling helder is wordt een Plan van Aanpak uitgeschreven. In het Plan van Aanpak wordt de brug geslagen tussen het afgebakende resultaat en de inrichting van het project, door middel van beantwoording van de "hoe"-vraag. Doel is om door middel van het Plan van Aanpak overeenstemming te verkrijgen over de te volgen weg, om te komen tot het gewenste resultaat.

Per eindresultaat wordt aangegeven welke activiteiten zullen worden uitgevoerd en eventueel welke tussenresultaten worden opgeleverd. Tevens wordt hierbij ingegaan op het waarom van de gekozen oplossing. Daarbij wordt verwezen naar de geformuleerde eisen en beperkingen ten aanzien van proces, resultaat en kwaliteit. Binnen het Plan van Aanpak wordt het project gefaseerd waarbij de fasering wordt gericht op de realisatie van op zich zelf staande functionaliteiten, die na samenvoeging in de laatste projectfase als één geheel de oplossing vormen. Indien nodig kan de samenhang gevisualiseerd worden in de vorm van een eenvoudig netwerkplan zonder kwantitatieve gegevens. Conform de structuur en fasering wordt dit hoofdstuk in paragrafen opgedeeld.

Het plan van aanpak wordt geschreven van week 30/2015 tot en met week 34/2015

7.3 USER STORIES

Op basis van de requirementsanalyse worden de User Stories uitgewerkt. Een User Story is een korte beschrijving (story) van wat een gebruiker (user) wil. User Stories worden gebruikt bij het ontwikkelen van software of producten. Een User Story zijn enkele zinnen gewone spreektaal van de (computer)gebruiker waarin staat wat de gebruiker doet of moet doen, als onderdeel van z'n werk. Het is de belangrijkste manier waarop de gebruiker invloed heeft op de functionaliteit van het te ontwikkelen systeem of product.

De User Stories worden geschreven van week 44/2015 tot en met week 48/2016.

7.4 USE CASES

Vervolgens worden de User Stories uitgewerkt tot Use Cases. Een Use Case beschrijft een systeem vanuit het gebruikersperspectief. Het beschrijft de actor, de initiator van de interactie, en het systeem zelf als een opeenvolging van eenvoudige stappen. Actoren kunnen iets of iemand zijn, die bestaat buiten het te bestuderen systeem, en die deelneemt in de opeenvolgende activiteiten in een dialoog met het systeem om een bepaald doel te bereiken. Actoren kunnen eindgebruikers, andere systemen of hardware (apparatuur) zijn. Elke Use Case is een complete serie van zogenaamde "events", beschreven vanuit het standpunt van de actor.

De Use Cases worden voorgelegd aan de opdrachtgever ter akkoordverkrijging. De Use Cases dienen als basis voor de ontwikkeling door de systeemontwikkelaars. Wanneer de Use Cases akkoord bevonden zijn kan er worden begonnen met de systeemontwikkeling in de realisatiefase.

De Use Cases worden geschreven van week 3/2016 tot en met week 7/2016.

7.5 AANPASSINGEN DOEN AAN REGISTRATIEPLATFORM

Binnen het bestaande registratieplatform moeten er een aantal aanpassingen worden gedaan om de specifieke functionaliteit te ondersteunen die binnen de App wordt ontwikkeld. In het bijzonder gaat het hier om:

- Het toevoegen van de geografische ligging aan een perceel in de vorm van een kaart;
- Het toevoegen van de geografische ligging aan een teelt in de vorm van een zone binnen de perceelskaart;



- Het toevoegen van functionaliteit om machtigingen te kunnen uitgeven aan een derde om perceelsgegevens te kunnen inzien;
- Het toevoegen van functionaliteit om machtigingen te kunnen aannemen van een derde om perceelsgegevens van die derde te kunnen inzien;
- Het toevoegen van functionaliteit om de geografische ligging van teelten afkomsting van RTK-GPS systemen op plantmachines te kunnen inlezen;
- Het toevoegen van functionaliteit om registratiebestanden (As-Applied) van sputtmachines te kunnen inlezen.

De aanpassingen aan het registratieplatform worden gerealiseerd in week 8/2016 tot en met week 21/2016.

7.6 REALISATIE VAN DE APP

Het realiseren van een mobiele App die geschikt is voor installatie op Smartphones en Tablets die draaien onder het besturingssysteem iOS of Android in de volgende volgorde:

- Toevoegen van functionaliteit om informatie over gewasbeschermingsmiddelen op te vragen en deze te kunnen opzoeken per gewas;
- Toevoegen van functionaliteit om te kunnen communiceren met het platform 1xregistreren.nl en daarbij aan de gebruiker gekoppelde informatie te kunnen inlezen;
- Toevoegen van functionaliteit om de geografische locatie van een perceel op een kaart te kunnen zien;
- Toevoegen van functionaliteit om de geografische locatie van teelten, rassen of partijen binnen een perceel op een kaart te kunnen zien;
- Toevoegen van functionaliteit om waarnemingen in de vorm van notities, gekoppeld aan de huidige geografische locatie van de gebruiker te kunnen koppelen aan een teelt, ras of partij;
- Toevoegen van functionaliteit om een kaartlaag met As-Applied gegevens vanaf het platform te kunnen inlezen en weer te geven binnen de app, over het perceel heen;

De realisatie van de App zelf wordt voorzien van week 22/2016 tot en met week 46/2016.

7.7 TEST & REWORK FASE

In de testfase wordt de op te leveren functionaliteit systematisch doorlopen om vast te kunnen stellen of de op te leveren functionaliteit voldoet aan de gestelde eisen en doelstellingen. Hierbij wordt achtereenvolgens getest:

- Algemene functionaliteit van de App;
- Communicatie tussen App en platform;
- Verstrekking en inwinning van machtigingen;
- Het inzien van gegevens van derden.

De test en rework fase staat gepland van week 47/2016 tot en met week 13/2017

Onder rework worden activiteiten uitgevoerd om bevindingen gedaan tijdens de systeemtests die niet voldoen aan de gestelde doelstellingen te herstellen.

7.8 OPLEVERFASE

In de opleverfase wordt de App in de Apple App Store en in de Google PlayStore geplaatst.

De oplevering van de App wordt voorzien in week 14 en 15 van 2017.



7.9 COMMUNICATIEFASE

Binnen de communicatiefase worden een aantal activiteiten uitgevoerd om de introductie van de Perceel Online App te communiceren. Op hoofdlijnen worden hieronder de volgende activiteiten verstaan:

- Toevoegen van een sectie over de Perceel Online App op de homepage van 1xregistreren.nl;
- Uitschrijven van een persbericht;
- Het verkrijgen van extra media-aandacht door diverse journalisten van vakbladen te benaderen voor het schrijven van een artikel over de Perceel Online App.
- Het geven van een aantal demonstraties op studieclub-avonden en regionale boomkwekerij-vakbeurzen.

De communicatiefase loopt van week 3/2017 tot en met week 27/2017.

7.10 DOCUMENTATIEFASE

Voor 30 september 2017 wordt er een beschrijvend eindverslag opgeleverd aan de Raad voor de Boomkwekerij. Dit verslag voorziet een samenvatting, het projectplan, een beschrijving van het verloop van het project en de resultaten uit het communicatieplan. Daarnaast wordt de projectbegroting en eindafrekening vermeld en wordt een toelichting gegeven op de eindafrekening.

De documentatiefase is de laatste fase die voorzien is binnen het project en loopt van week 17/2017 tot en met week 21/2017.

Bijlage 2 toont de grafische overzichtsweergave van de projectplanning.

8 Knelpuntenanalyse

De doelstelling is het ontwikkelen en realiseren van een App voor actueel inzicht in toegelaten gewasbeschermingsmiddelen per gewas en het registreren en inzichtelijk maken van uitgevoerde handelingen in het perceel. Daarbij is:

1. De bouw afhankelijk van de vorderingen door een ICT bedrijf;
2. De werking afhankelijk van een goed functionerend internet;
3. De inhoud afhankelijk van
 - De actualiteit van de infobronnen over (toelatingen van) gewasbeschermingsmiddelen;
 - De wijze en nauwkeurigheid van registreren door de betrokkenen (personeelsleden en derden);
4. Het te realiseren marktaandeel afhankelijk van de acceptatie door de potentiële gebruikers.

Bovengenoemde afhankelijkheden brengen risico's op knelpunten met zich mee. Deze knelpunten zijn (behorend bij de nummers):

1. Vertragingen in de oplevering waardoor overige doelstellingen niet meer worden gehaald;
2. Het niet functioneren van de oplossing door het gebrek aan een werkende internetverbinding;
3. Een niet actuele inhoud waardoor de oplossing in de praktijk niet voldoet aan de verwachtingen;
4. Het marktaandeel achterblijft en daardoor minder actieve gebruikers de App gebruiken. Hierdoor komt de doorontwikkeling en up to date houden van de App in gevaar.

Deze knelpunten kunnen worden ondervangen door het nemen van de volgende maatregelen bij elke respectievelijk punt:

1. De App te laten ontwikkelen volgens de Agile Scrum methodiek waarbij initiatiefnemer als productowner een actieve positie inneemt binnen het ontwikkeltraject en invloed uit kan oefenen op het ontwikkelproces door het bewaken en stellen van prioriteiten.



2. Dit knelpunt ligt buiten de invloedssfeer van de betrokkenen. In de informatiesfeer bij de App wordt het advies meegegeven te kiezen voor een betrouwbare (mobiele internet) provider die een goede dekking garandeert in het werkgebied van de gebruiker.
3. Het kiezen van een bewezen en geschikte partij als aanbieder van het gekoppelde registratieplatform, zo mogelijk een die belang heeft bij zo actueel mogelijke informatie over gewasbeschermingsmiddelen vanwege overige producten die deze partij aanbiedt. Het meeleveren van een gemakkelijk leesbare gebruikershandleiding gaat gebruikers wijzen op het belang van een tijdige en correcte registratie.
4. Uitvoering geven aan het onder 9.1 vermelde communicatieplan borgt dat er voldoende inspanning geleverd wordt ten aanzien van het vermarkten van de App.

9 Plannen

In het hoofdstuk plannen wordt de resultante vastgelegd van het evenwicht tussen activiteiten, tijd, geld en middelen teneinde de opdracht te kunnen uitvoeren. De verschillende paragrafen worden als volgt ingevuld:

9.1 COMMUNICATIEPLAN

De Perceel Online App is de digitale teelt assistent bij uitstek. De Perceel Online App bundelt per gewas essentiële informatie over gewasbeschermingsmiddelen samen met gegevens over de teeltregistratie en biedt de mogelijkheden in het veld de huidige bewerking te registreren. Alle gegevens worden online bewaard en zijn vanaf iedere plek via een internetverbinding inzichtelijk. Ook kan men uitvoerders van werk op het perceel machtigen gegevens in te zien en zelfs te registreren.

De Perceel Online App is ontwikkeld op verzoek van een groep boomtelers door 1xregisteren.nl. De leden van deze groep boomtelers, vertegenwoordigd door Boomkwekerij Fleuren BV zijn de trekker van het product en verantwoordelijk om van dit plan een succes te maken.

9.1.1 Doelgroep

De primaire doelgroepen voor de Perceel Online App zijn:

- Kwekers van boomkwekerijproducten in de vollegrond;
- Kwekers van fruitgewassen in de vollegrond;
- Kwekers van sierteeltgewassen in de vollegrond.

De secundaire doelgroepen voor de Perceel Online App zijn:

- Telers van groentegewassen in de vollegrond;
- Telers van tuinbouwgewassen in de vollegrond;
- Telers van akkerbouwgewassen in de vollegrond.

In eerste instantie wordt er op de primaire doelgroep gericht.

Intermediairs

- De groep initiatiefnemers voor deze App als enthousiaste gebruikers;
- Leveranciers van gewasbeschermingsmiddelen en hulpstoffen;
- Accountants;
- Adviseurs en teeltbegeleiders.

9.1.2 Doelstelling

De kwekers van boomkwekerijproducten, fruitgewassen en sierteeltgewassen in de vollegrond:

- Voor 1 januari 2018 kent 80% van deze groep het bestaan van deze App en heeft weet van enkele voordelen die het gebruik van deze App met zich meebrengt;



- Voor 1 juli 2018 ziet 50% van deze groep het gebruik van deze App als een meerwaarde;
- Voor 1 januari 2019 is 20% van deze groep een actief gebruiker van deze App.

Van de telers van groentegewassen, tuinbouwgewassen en akkerbouwgewassen in de vollegrond:

- Kent voor juli 2018 80% van deze groep het bestaan van de App en heeft weet van enkele voordelen die het gebruik van deze App met zich meebrengt;
- Ziet voor 1 januari 2019 50% van deze groep het gebruik van deze App als een meerwaarde;
- Voor 1 juli 2019 is 20% van deze groep een actief gebruiker van deze App.

De intermediairs:

- Weten voor 1 oktober 2017 wat de Perceel Online App inhoudt en wat de voordelen van het gebruik ervan zijn;
- Vinden voor 1 december 2017 de Perceel Online App een aanvulling voor de primaire doelgroep;
- Brengen vanaf 1 december 2017 de Perceel Online App actief onder de aandacht bij de primaire doelgroep.

9.1.3 Boodschap

De Perceel Online App geeft een actueel inzicht in mijn teelt en via deze app kan ik effectief plannen en registreren. Het inzicht in toegelaten gewasbeschermingsmiddelen per teelt is een geweldige vervanging voor het huidige uitzoekwerk dat gepaard gaat met het plannen van een gewasbeschermingsactie.

9.1.4 Strategie

De intermediairs fungeren als ambassadeurs richting de primaire doelgroep. Deze intermediairs worden ondersteund door media aandacht voor de Perceel Online App. De media aandacht draait vooral om het verspreiden van het gebruiksgemak en de meerwaarde vanuit het perspectief van de huidige gebruiker. Een eerste aanzet vanuit het gebruikersperspectief wordt gegeven door de initiatiefnemers als enthousiaste gebruikers.

9.1.5 Communicatiemiddelen

- Informatie/introductiefilmpje;
- Artikelen in gewasnieuwsbrieven (waar onder testimonials);
- Artikelen in de vakpers;
- Nieuwsbrieven andere producten 1xregistreren.nl;
- Banner op website 1xregistreren.nl;
- Lezingen en presentaties voor studiegroepen;
- Demonstraties op vak gerelateerde beurzen en bijeenkomsten;
- Factsheets uitdelen op va gerelateerde beurzen en bijeenkomsten;
- Persberichten;
- Georganiseerde bijeenkomst;
- Direct mailing.

9.1.6 Actieplannen

Fase 1: Januari t/m oktober 2017

- Informatie/introductiefilmpje gereed;
- Uitschrijven van persberichten naar de vakpers met link naar introductiefilmpje en testimonial van initiatiefnemer;
- Bijeenkomst voor intermediairs (toeleveranciers en adviseurs) organiseren ter introductie van de App;
- intermediairs (toeleveranciers en adviseurs)
- Uitschrijven van persberichten naar toeleveranciers en adviseurs met link naar introductiefilmpje en testimonial van initiatiefnemer met de vraag deze informatie in hun nieuwsbrief op te nemen;



- Telefonisch contact met de vakpers om toelichting gegeven op het persbericht;
- Telefonisch contact met de toeleveranciers en adviseurs om toelichting te geven op het verzoek;
- Aanwezigheid op een toonaangevende (landelijke) vakbeurs of op een aantal regionale vakbeurzen.
- Banner op website 1xregistreren.nl

Fase 2: oktober 2017 t/m juli 2018

- Persbericht over enthousiaste eerste gebruikers naar de vakpers;
- Lezingen en presentaties voor studiegroepen;
- Demonstraties op vak gerelateerde beurzen en bijeenkomsten;
- Factsheets uitdelen op va gerelateerde beurzen en bijeenkomsten;
- Nieuwsbrieven andere producten 1xregistreren.nl;

Fase 3: augustus 2018 t/m juli 2019

- Lezingen en presentaties voor studiegroepen;
- Aanwezigheid op een toonaangevende (landelijke) vakbeurs of op een aantal regionale vakbeurzen.
- Demonstraties op vak gerelateerde beurzen en bijeenkomsten;
- Factsheets uitdelen op va gerelateerde beurzen en bijeenkomsten;
- Vermelding in nieuwsbrieven van de intermediairs.

9.2 FINANCIËEL PLAN

In deze paragraaf wordt inzicht gegeven in de kosten (mensen, middelen en overig) van het project. Aangegeven worden de resources die in de planning zijn opgenomen, de hiervoor gehanteerde tarieven en de hieruit resulterende verwachte kosten.

10 Kwaliteitsborging

Dit hoofdstuk geeft inzicht in de relatie tussen de voorgestelde maatregelen en de door de opdrachtgever gestelde eisen ten aanzien van de kwaliteit. Hiernaast worden maatregelen getroffen om onderkende risico's uit te sluiten of de gevolgen te minimaliseren, en de cruciale succesfactoren te beïnvloeden. Als uitgangspunt worden de door de opdrachtgever gestelde kwaliteitseisen gehanteerd. Deze worden verbijzonderd naar de te stellen kwaliteitseisen per product. De voorgestelde maatregelen in het proces zijn een vertaling van deze vastgestelde productkwaliteitseisen. Naast maatregelen in het proces om te voldoen aan de kwaliteitseisen per product worden additioneel maatregelen getroffen voor de kwaliteit van de tussenproducten of het proces zelf. Laatstgenoemde wordt ontleend aan ondermeer de vereiste kwaliteit van besturing of het minimaliseren van risico's. Alle maatregelen zijn in het proces ingebouwd en zijn dus elders in het plan van aanpak opgenomen als activiteit, inrichtingsaspect of voorwaarde. Dit hoofdstuk geeft het totaaloverzicht van de invulling van het kwaliteitsaspect. De paragrafen worden als volgt ingevuld:

10.1 PRODUCTKWALITEIT

Eisen per product per kwaliteitsattribuut voorzien van weging en acceptatiecriteria. Relatie met de gestelde eisen aan, en acceptatiecriteria van, het projectresultaat

10.2 PROCESKWALITEIT

Eisen te stellen aan het proces.

Voorbeelden hiervan zijn:

- vakbekwaamheid



- gebruik van (systeem)ontwikkelmethode
- procedures
- gebruik van methode voor projectmanagement;
- uitbesteding en inkoop

Controle achteraf is mogelijk door verificatie en validatie

10.3 VOORGESTELDE MAATREGELEN

Maatregelen in het proces met per maatregel de relatie naar de eisen.

Voorbeelden hiervan zijn:

- opleidingsplan
- gebruik van methode voor systeemontwikkeling
- testplan
- gebruik van Managing Projects als methode voor projectmanagement

10.4 MAATREGELEN TER VERIFICATIE EN VALIDATIE

Voorbeelden hiervan zijn;

- audits
- reviews

Bovenstaande, mogelijk lange en droge opsomming van, relaties kunnen visueel meer inzichtelijk worden gemaakt door deze op te nemen in een matrix.

11 Overige plannen

In dit hoofdstuk worden alle plannen opgenomen die niet op tijd, geld en middelen zijn gericht. De invulling is afhankelijk van de projectbehoefte.

Voorbeelden:

- communicatieplan
- documentatieplan
- configuratiebeheerplan
- beveiligingsplan



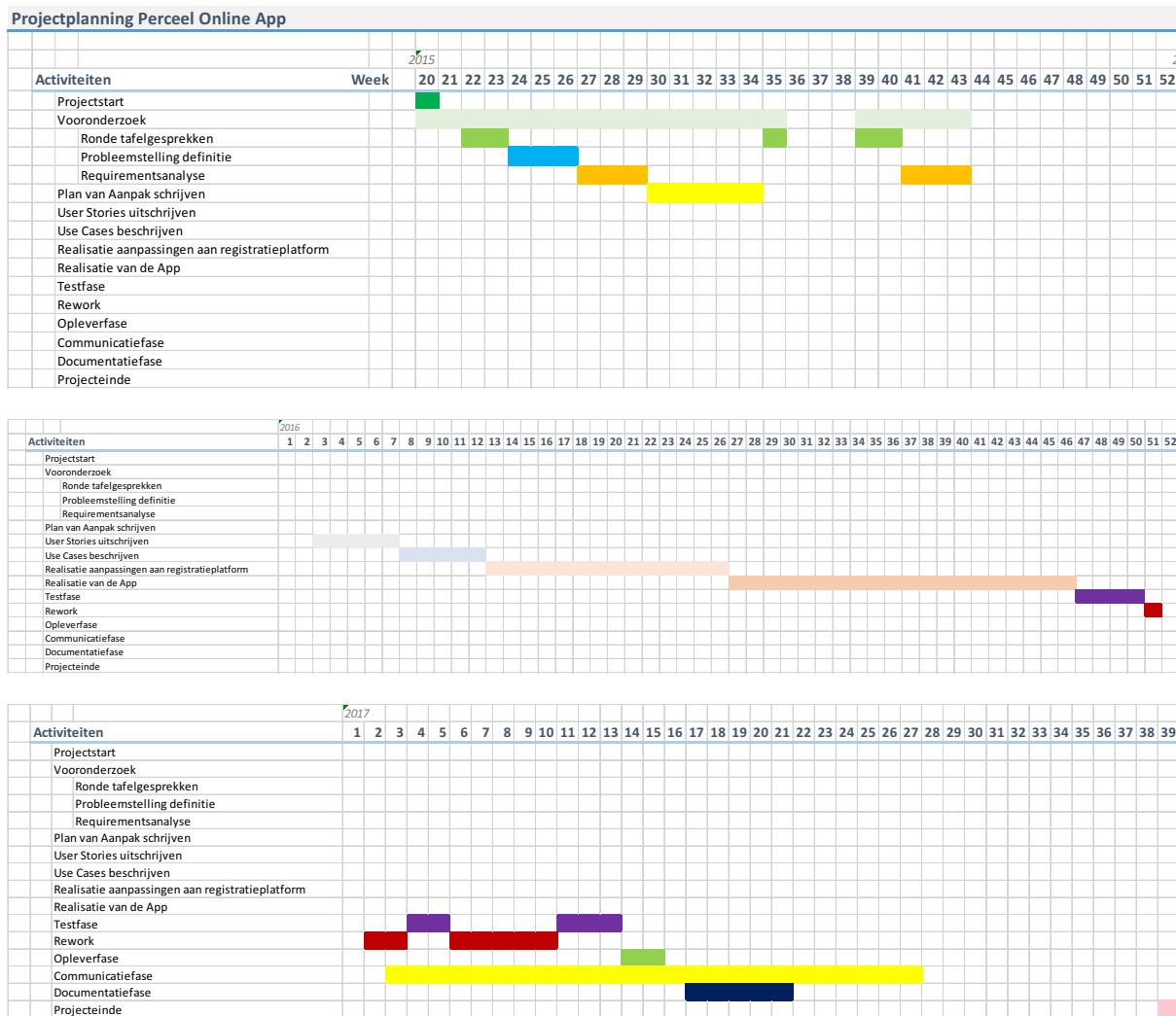
12 Bijlagen

In dit hoofdstuk wordt verwezen naar de relevante standaards en projectprocedures. In het voorkomend geval zal verwezen worden naar reeds bestaande c.q. gebruikelijke bedrijfsstandaards. Voorwaarde is wel dat deze gedocumenteerd zijn.

In de bijlagen worden ook Begrippen en definities opgenomen om begripsverwarring te voorkomen. De begrippenlijst hoeft niet uitputtend te zijn, alleen de gehanteerde begrippen in het Plan van Aanpak komen hiervoor in aanmerking.

12.1 BIJLAGE 1: BEGRIPPEN EN DEFINITIES

12.2 BIJLAGE 2: PROJECTPLANNING



Chapter 4

Quality Management Plan



QUALITY MANAGEMENT PLAN 2015 SOFA GTL TREEWATCH PROJECT



Jan Kerkenhoff
Martijn Bonajo
Max van der Linden
René Karoff
Ron Gebauer

2016-01-17

Document information

Document name: Quality Management Plan
Document owner: TreeWatch
Company/Organisation: Fleuren Baarlo
Contact person: Max van der Linden, Group leader
Date: 2016-01-17
Place: Fontys University of Applied Science Venlo
Authors:
j.kerkenhoff@student.fontys.nl
2197164

Martijn Bonajo
m.bonajo@student.fontys.nl
2213297

Max van der Linden
max.vanderlinden@student.fontys.nl
2209349

René Karoff
r.karoff@student.fontys.nl
2198664

Ron Gebauer
r.gebauer@student.fontys.nl
2153294

Revision history

Editor	Date	Note	Status
Rene Karoff	2015-09-04	Initial	Draft
Max van der Linden	2015-09-04	Review	Draft
Rene Karoff	2015-09-24	Update	Draft
Rene Karoff	2015-10-02	Update	v0.1
Max van der Linden	2015-10-02	Review	v0.1
Rene Karoff	2015-10-16	Update	v0.2
Rene Karoff	2015-11-13	Update	v0.3
Rene Karoff	2015-12-10	Update	Final
Max van der Linden	2015-12-10	Review	Final

List of Figures

1	Review Process	7
---	----------------	---

List of Tables

1	Group roles	2
3	All Reviewed Documents	9

Contents

Document information	I
Revision history	II
List of Figures	II
List of Tables	II
1 Introduction	1
2 Organization and Responsibilities	2
3 Quality Planning	3
3.1 Define Project Quality	3
3.2 Measure Project Quality	3
4 Quality Assurance	4
4.1 Writing Guidelines	4
4.2 Coding convention	4
5 Analyze Project Quality	4
6 Improve Project Quality	6
7 Quality Control	7

1 Introduction

This quality management plan describes the necessary information needed to manage the project quality from the project planning to the delivery to the customer. Within this document the quality policies, procedures, roles, responsibilities and authorities are defined.

2 Organization and Responsibilities

The SoFa Group consists of five people. Since the SoFa Group should act as a company roles and responsibilities have to be defined.

The following table shows every group member with his role and the responsibilities.

Name	Role	Quality Responsibility
Max van der Linden	Project Manager	Project Planning, External communication, Auditing
Martijn Bonajo	Configuration Manager	Configuration Plan, Infrastructure, Approve Packages and Pushes to Master, Auditing
Ron Gebauer	Scrum Master	Scrum planning, Auditing
Rene Karoff	Quality Manager	Quality Management Plan, Ensure use of quality guidelines, Auditing
Jan Kerkenhoff	Main Engineer	Main Code-Master, Auditing

Table 1: Group roles

Besides his role, the group will work together on several tasks (e.g. User Stories, Design or Implementation).

3 Quality Planning

This chapter is about the defined project quality standards and how the project standards are measured. The defined standards include coding style guides as well as documentation style guides.

3.1 Define Project Quality

Since this project is about programming a mobile business application, it is highly important that the system has a minimum of bugs/errors. A good documentation is necessary too, not only because this application will be developed by VAA ICT Consultancy after the SoFa is finished, but the customer should understand the system which is developed. To ensure that the quality of the written code it matches the expectation of our customer it is mandatory to develop every module test-driven, therefore unit-testing is introduced. The written documents will be checked for its grammar, writing style and content.

Furthermore there are several design constraints which every member of the group has to follow when creating a document, these constraints are defined in the chapter Quality assurance.

3.2 Measure Project Quality

During the implementation every implemented feature needs to be reviewed by another group member, the written code is reviewed and if there is any effect on the GUI, it is also to be reviewed.

In the Appendix is a table which contains the description which document was reviewed by whom and what grade (from ~~-~~ to ~~+~~) it got when it was approved. The code of the application should be tested. Therefore it is a goal to get a code coverage 80%.

4 Quality Assurance

Since the code is tested, this will show the group if the quality goal is achieved. Furthermore the written documents will be audited by at least one group member. This chapter also contains the Writing guidelines as well as the used coding convention.

4.1 Writing Guidelines

General Each document which will be created shall be created using LaTeX

Template Every document which will be created shall use a Template containing:

 Titlepage

 Document information

Font size The Font size for default text is 12pt, LaTeX scales the headings according to the default text size.

Date format The Date in every documents to be written in according to ISO 8601
 use `\today` or `\printdate{dd.mm.yyyy}`

Tables & Figures Tables and Figures shall be marked as what they are

 use `\caption{Guidelines}` `\label{tab:Guidelines}`

Images Images inside a document shall be readable when printed, if images are to big, they have to be in the appendix

Documents to be delivered Every document which is sent to anyone, will be exported as a .PDF-File

Citations Citations are to be done with the IEEE standard which is also used within the Fontys Internship reports and the Graduation reports

4.2 Coding convention

To keep the code readable and understandable it is important that every programmer uses the same coding convention which is hereby defined as the official Microsoft C# Coding Conventions. [Microsoft Corporation(2015)]

5 Analyze Project Quality

Since this project is this application is implemented with Xamarin, the group wasn't able to find a suitable possibility to measure the code coverage, however within Xamarin it should be possible somehow to introduce code coverage measurements since it uses the Mono-framework but due to several forums (e.g. [Unknown(2015)]) It would take a lot of effort and consume a lot of time which the group prefers to spend on the solution. Furthermore the writer of a document will receive feedback

of the auditing person, so he can improve on his writing too.

6 Improve Project Quality

To improve the quality of code the following tools were used:

Pair programming Writing code with the 4-eyes-principle helps preventing errors especially with complicated algorithms.

Feature-driven development Every developer pair implements one feature and only if this feature works correctly the next feature will be implemented.

Test-driven development In order to have the code working properly, the code should be tested using unit-tests.

7 Quality Control

To verify that the quality of every document and every feature matches the introduced standard they were reviewed by at least one group member, therefore the review process shown in Figure 1 was introduced. In the review process the term 'document' is defined as either a .PDF-document or a feature.

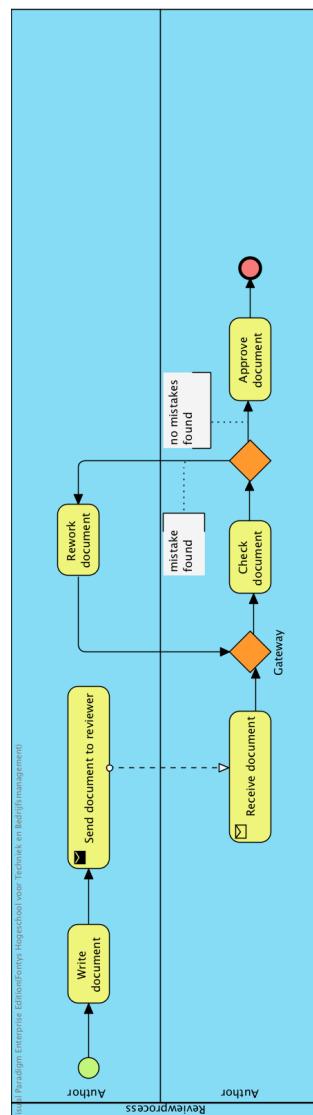


Figure 1: Review Process

References

- [Microsoft Corporation(2015)] Microsoft Corporation. (2015) C# coding conventions. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ff926074.aspx> Accessed: 2015-01-08
- [Unknown(2015)] Unknown. (2015) What do you use for code coverage.

Appendix

Reviewed Documents

Document	Author	Reviewer	Grade	Remark	Approval
Native vs Hybrid	Rene Karoff & Martijn Bonajo	Jan Kerkenhoff	+		yes
SCRUM Planning	Ron Gebauer	Rene Karoff	+		yes
Quality Management Plan	Rene Karoff	Max van der Linden	-	Add reviewed documents	no
Quality Management Plan	Rene Karoff	Max van der Linden	+		yes
Configuration Plan	Martijn Bonajo	Jan Kerkenhoff	-	Add Git-Structure	no
Configuration Plan	Martijn Bonajo	Jan Kerkenhoff	+		yes
ER Diagram	Rene Karoff	Martijn Bonajo	-	Spelling Error	No
ER Diagram	Rene Karoff	Martijn Bonajo	++		Yes
User Stories	Max van der Linden	Group & Customer	++		yes
Mockups	Groupwork	Customer	0	No grade given	yes

Table 3: All Reviewed Documents

Chapter 5

Configuration Plan



CONFIGURATION REPORT

2015 SOFA GTL

TREEWATCH PROJECT



Martijn Bonajo

2016-01-17

Document information

Document name: Configuration report
Document owner: TreeWatch
Company/Organisation: Fleuren Baarlo
Contact person: Max van der Linden, Group leader
Date: 2016-01-17
Place: Fontys University of Applied Science Venlo
Authors:
Martijn Bonajo
m.bonajo@student.fontys.nl
2213297

René Karoff
r.karoff@student.fontys.nl
2198664

Revision history

Editor	Date	Note	Status
Martijn Bonajo	2015-09-01	Initial	Draft
René Karoff	2015-09-02	-	Review
Martijn Bonajo	2015-10-09	Update	Proposal
Martijn Bonajo	2015-10-16	Update	v0.1
Martijn Bonajo	2015-11-03	Update	v0.2
Martijn Bonajo	2015-11-17	Update	v0.3
Martijn Bonajo	2016-01-08	Update	Final
Rene Karoff	2016-01-08	Review	Final

Glossary

Appveyor is a hosted, distributed continuous integration service used to build and test projects hosted at GitHub on a Microsoft Windows virtual machine.. 2

git is a widely-used version control system for software development. It is a distributed revision control system with an emphasis on speed,[7] data integrity, and support for distributed, non-linear workflows. 2

github is a Web-based Git repository hosting service. It offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features. 2

PascalCase is the practice of writing compound words or phrases such that each word or abbreviation begins with a capital letter. 4

Contents

Document information	I
Revision history	I
Glossary	II
Glossary	IV
1 Introduction	1
1.1 Purpose	1
1.2 Scope	1
2 Hardware	2
3 Software	2
3.1 Version control	2
3.2 Baselines	2
3.3 Change control	2
4 Environments	3
4.1 GIT structure	3
5 Conventions	4
6 Software dependencies	4
6.1 Google maps	6

Glossary

Appveyor is a hosted, distributed continuous integration service used to build and test projects hosted at GitHub on a Microsoft Windows virtual machine.. 2

git is a widely-used version control system for software development. It is a distributed revision control system with an emphasis on speed,[7] data integrity, and support for distributed, non-linear workflows. 2

github is a Web-based Git repository hosting service. It offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features. 2

PascalCase is the practice of writing compound words or phrases such that each word or abbreviation begins with a capital letter. 4

1 Introduction

1.1 Purpose

The purpose of this document is to give rules to the coding standard, naming conventions, layout standard that will be used in this project.

1.2 Scope

This document is about:

- Giving naming conventions for all documents.
- Environment configuration
- Software configuration
- Deployment pipeline

This document will not contain:

- Detailed content of files
- Folder structure

2 Hardware

The hardware used in this project are the laptops of the participants and a couple of testing devices. These devices are not necessary for the deployment of the application and will therefore be omitted from this document.

3 Software

3.1 Version control

All source code, this includes the test code, will be stored under version control. The team decided to use git as the version control system. For this project it was decided that it should be open source, so this meant we could use a public repository on github.

3.2 Baselines

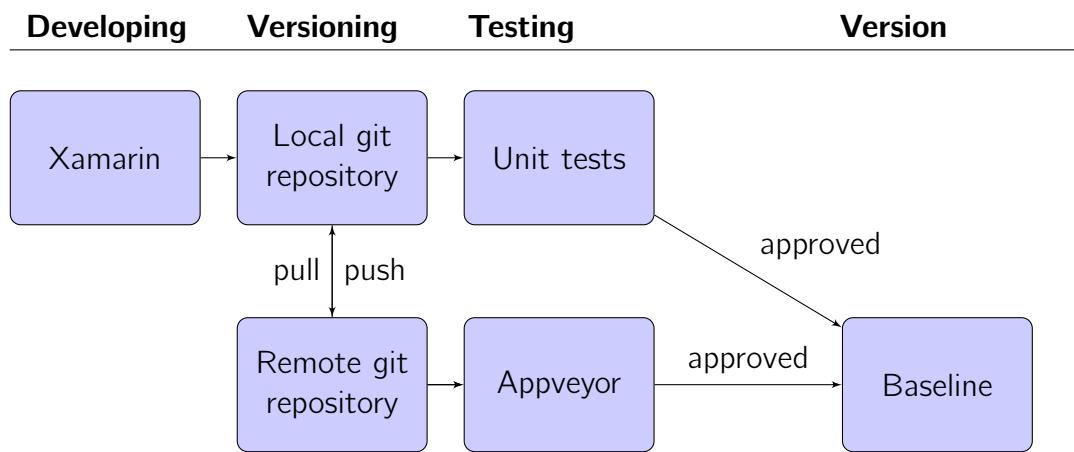
At the beginning of the project a baseline for the software was established. The baseline can be changed by new dependencies or by new versions of the software. All changes to the baseline have to be approved by the configuration manager.

Changes to the baseline should pass the automatic build test from Appveyor on github and manual testing before it can be added as a new baseline.

3.3 Change control

All features should be developed on their own branch on github. This is to make sure the number of conflicts stays to a minimum. When a feature is done it should first be brought up to date with the current baseline(if it is not already). Before a feature can become a new baseline it should first be approved by the quality manager. Then it should pass the automatic build by Appveyor. After this the new feature is merged with the current baseline and becomes the new baseline.

4 Environments

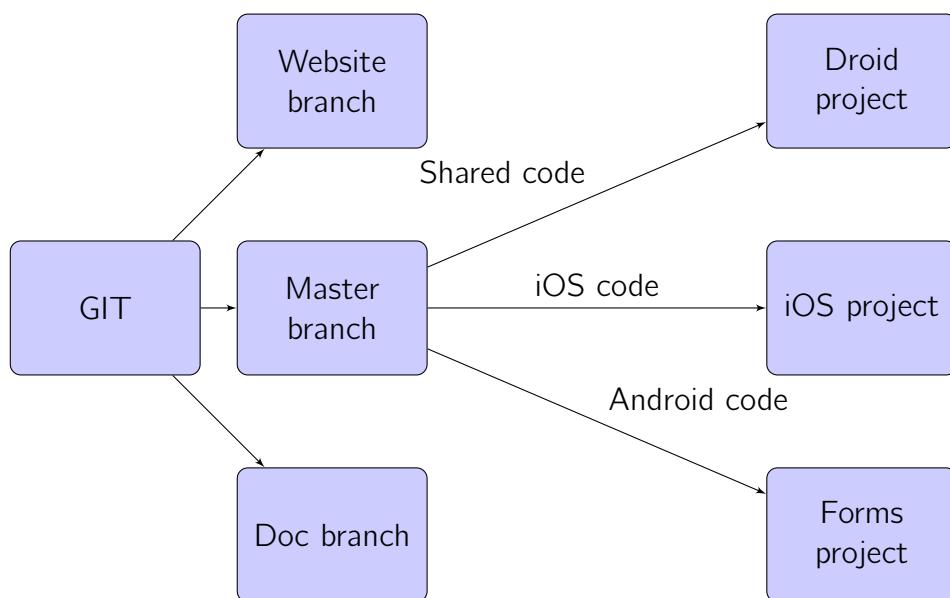


4.1 GIT structure

All documentation can be found in the doc branch.

The project can be in the master branch.

The website can be found in the website branch.



5 Conventions

Filename conventions:

Images PascalCase

Latex All lowercase letters

For programming in c# the developer should adhere to the official coding standards of Microsoft which can be found here: <https://msdn.microsoft.com/en-us/library/ff926074.aspx>.

6 Software dependencies

This section contains the software dependencies used in the project. The listed dependencies are managed by the specific packages.config file for each platform. Every time a new version of a dependency is released it should be tested if it works with the current baseline before adding it to the baseline.

Dependencies for Forms application:

Dependency	Version
Xamarin.Forms	2.0.0.6490
Xamarin.Forms.Maps	2.0.0.6490
ExifLib.PCL	1.0.1
Newtonsoft.Json	7.0.1
SQLite.Net.Core-PCL	3.1.1
SQLite.Net-PCL	3.1.1
SQLiteNetExtensions	1.3.0

Dependencies for Android application:

Dependency	Version
Xamarin.Forms	2.0.0.6490
Xamarin.Forms.Maps	2.0.0.6490
ExifLib.PCL	1.0.1
Newtonsoft.Json	7.0.1
SQLite.Net.Core-PCL	3.1.1
SQLite.Net-PCL	3.1.1
SQLiteNetExtensions	1.3.0
Xamarin.Android.Support.Design	23.0.1.3
Xamarin.Android.Support.v4	23.0.1.3
Xamarin.Android.Support.v7.AppCompat	23.0.1.3
Xamarin.Android.Support.v7.CardView	23.0.1.3
Xamarin.Android.Support.v7.MediaRouter	23.0.1.3

Dependencies for iOS application:

Dependency	Version
Xamarin.Forms	2.0.0.6490
Xamarin.Forms.Maps	2.0.0.6490
ExifLib.PCL	1.0.1
Newtonsoft.Json	7.0.1
SQLite.Net.Core-PCL	3.1.1
SQLite.Net-PCL	3.1.1
SQLiteNetExtensions	1.3.0
Xamarin.TestCloud.Agent	0.17.0

6.1 Google maps

The Google maps needs an api key and fingerprints to work on Android. In order to get the Google maps to work, you need to follow the following steps:

- Determine your MD5 or SHA1 signature, as shown here.
- Add fingerprint to existing key or create a new key and add fingerprint to it. As shown here.
- Replace the key in the AndoirdManifest.xml, which can be found at Droid/Properties/AndroidManifest.xml.

Chapter 6

Corporate Identity



CORPORATE IDENTITY
2015 SOFA GTL
TREEWATCH PROJECT



Max van der Linden

2016-01-16

Document information

Document name: Corporate Identity
Document owner: Max van der Linden
Company/Organisation: Fleuren Baarlo
Contact person: Max van der Linden, Group leader
Date: 2016-01-16
Place: Fontys University of Applied Science Venlo
Authors: Max van der Linden
max.vanderlinden@student.fontys.nl
2209349

Contents

Document information

1	Corporate Identity	1
1.1	Logo's	1
1.1.1	Company Logo	1
1.1.2	App Logo	1
1.2	Design Guidelines	2

1 Corporate Identity

As part of the SoFa it is important to act as a professional group. This acting professional also means that the group should feel as a company working for a real customer, making the project as realistic as possible. In order to achieve this a corporate identity needed to be created. The GTL SoFa project group decided to call itself TreeWatch since the app provides Fleuren with the ability to watch over their trees. The name of the group is also used as the app name. This meant that logos and a design guideline had to be created. The logo, app icon and home style can be found in the next images. When continuing the project it is recommended to keep using the home style in reporting and emailing.

1.1 Logo's

1.1.1 Company Logo



1.1.2 App Logo



1.2 Design Guidelines



Lettertype:
Devangani Sangam MN - Regular

Colors CMYK:



35 60 80 25



90 30 95 30



79 00 70 01



61 00 69 01

Design by Johannes Engelhardt 2015

Chapter 7

Native vs. Hybrid



NATIVE VS. HYBRID
2015 SOFA GTL
TREEWATCH PROJECT



René Karoff

2016-01-17

Document Information

Document name: Native vs. Hybrid

Document owner: TreeWatch

Company/Organisation: Fleuren Baarlo

Contact person: Max van der Linden, Group leader

Date: 2015-09-09

Place: Fontys University of Applied Science Venlo

Author:
René Karoff
r.karoff@student.fontys.nl
2198664

Contents

Document information

1 Native vs. Hybrid Application Development	1
1.1 Decision Criteria:	1
1.2 Hybrid	1
1.2.1 Pro's	1
1.2.2 Con's	1
1.3 Native	1
1.3.1 Pro's	1

1 Native vs. Hybrid Application Development

1.1 Decision Criteria:

How important speed and performance would be, to you. If you would like your app to include any device-specific features If you want your app deployed to more than one platform

1.2 Hybrid

1.2.1 Pro's

- One Codebase
- iOS developer Account still needed to deploy the App to the iOS AppStore
- Developed in HTML5, CSS3 and Javascript
- Depending on the SDK - Full Access to the API

1.2.2 Con's

- SDK-depending fee's
- iOS developer Account still needed to deploy the App to the iOS AppStore
- Developed in HTML5, CSS3 and Javascript
- Support for multiple browsers
- Apps performance dependent on the devices Browser capabilities

1.3 Native

1.3.1 Pro's

- Full Access to any API
- Optimised use of hardware resources

- Many code-snippets to be found online
- Two codebases to take care of
- Two different programming languages (Java & Objective-C or Swift)
- iOS developer Account

Chapter 8

Class Diagram

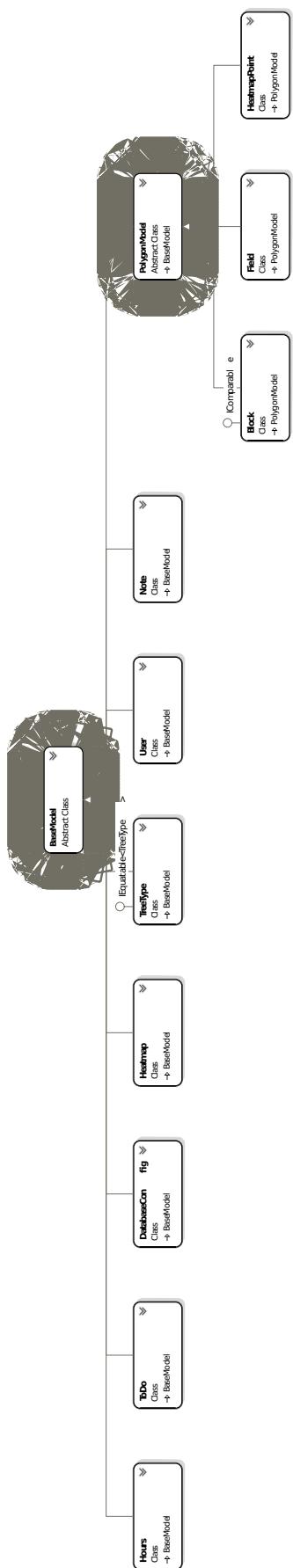


Figure 1: Class Diagramm Forms Part 1

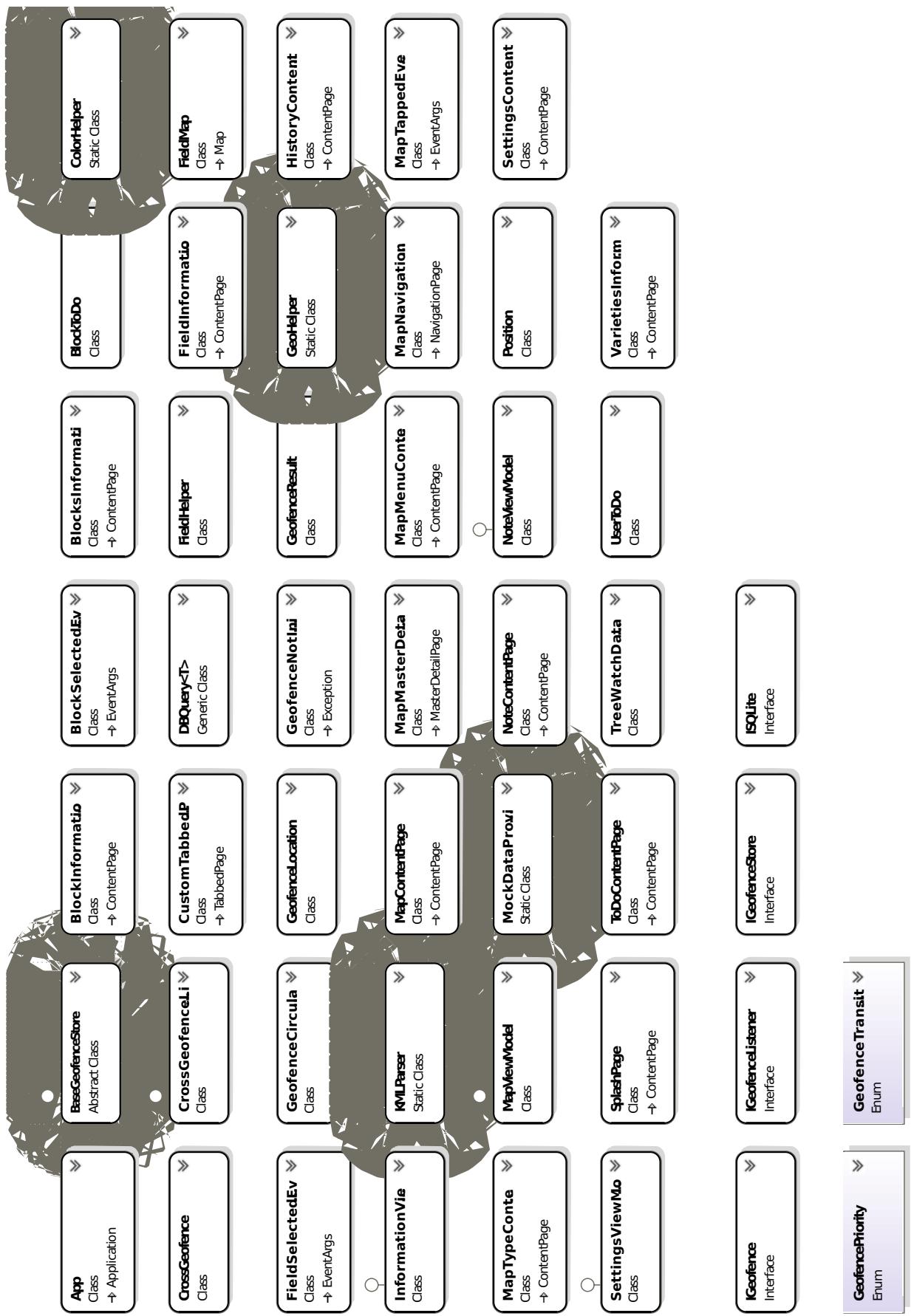


Figure 2: Class Diagramm Forms Part 2

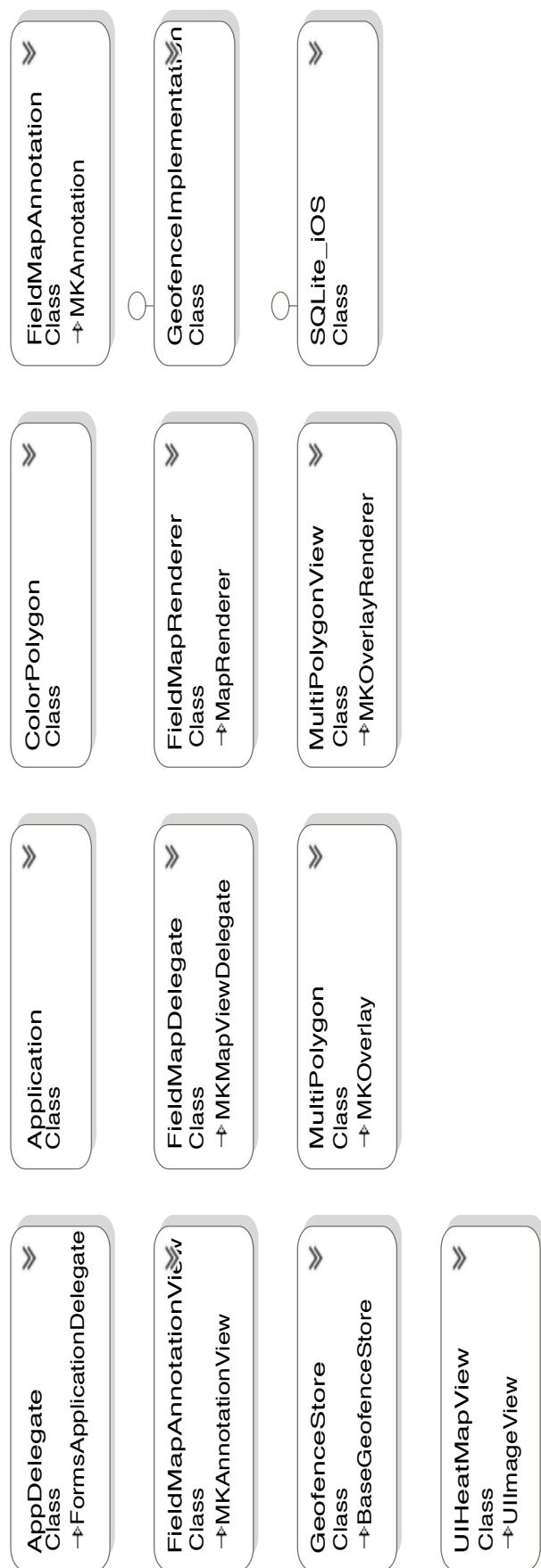


Figure 3: Class Diagramm iOS

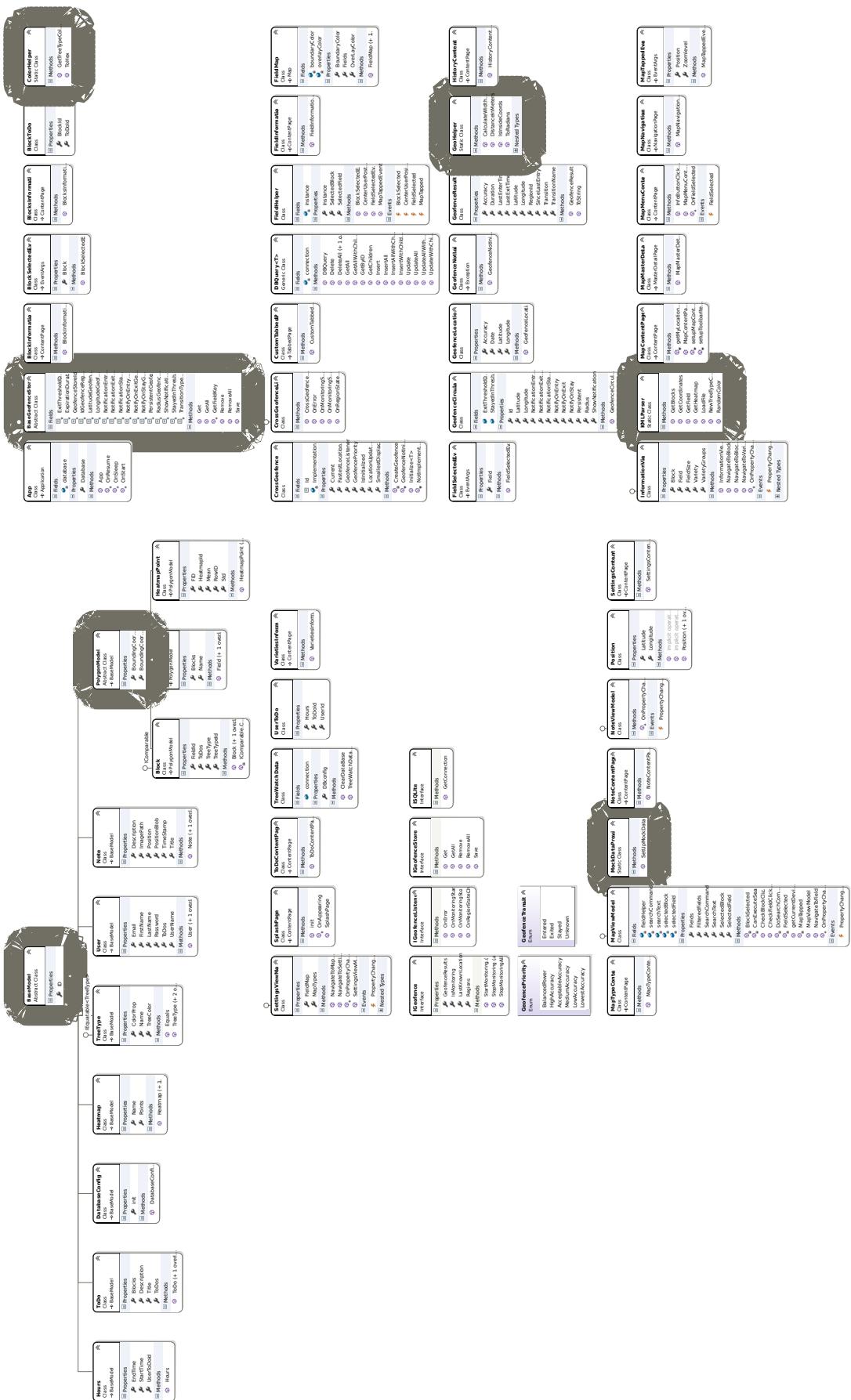


Figure 4: Class Diagramm Forms

Chapter 9

Geofence Class Diagram

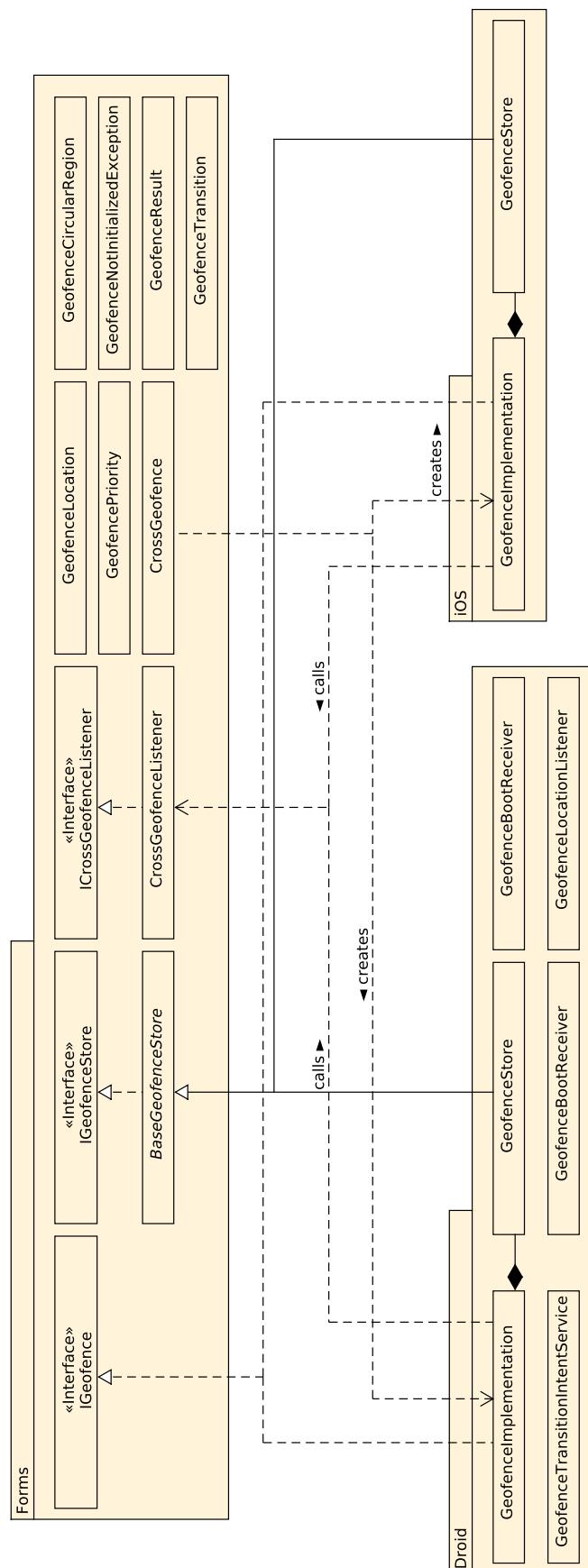


Figure 1: ¹ Geofence diagram

Chapter 10

Entity Relationship Diagram

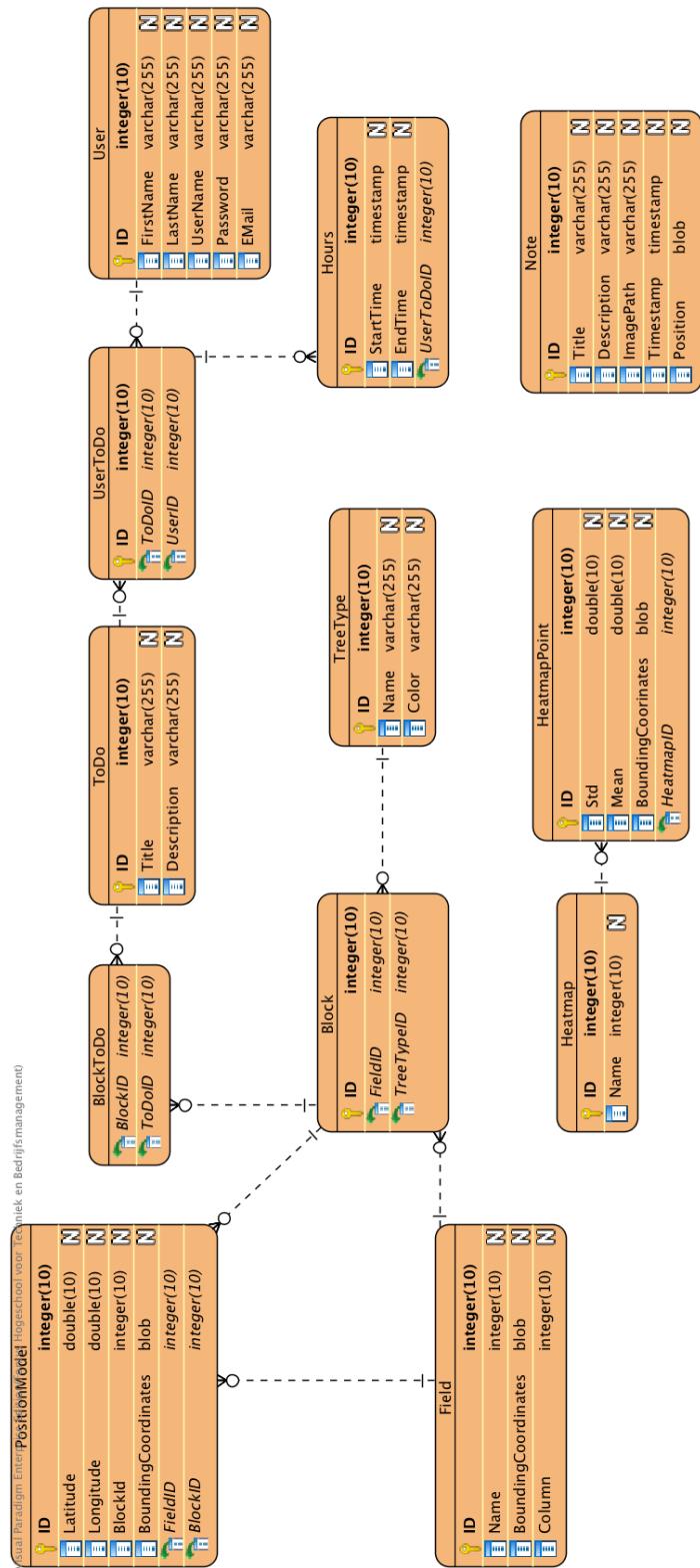


Figure 1: ER Diagram SQLite

Chapter 11

Meetings



CUSTOMER MEETINGS
2015 SOFA GTL
TREEWATCH PROJECT



Max van der Linden

2016-01-16

Document information

Document name: Customer Meetings
Document owner: Max van der Linden
Company/Organisation: Fleuren Baarlo
Contact person: Max van der Linden, Group leader
Date: 2016-01-16
Place: Fontys University of Applied Science Venlo
Authors: Max van der Linden
max.vanderlinden@student.fontys.nl
2209349

Contents

Document information	1
1 Customer meeting (2015-09-09)	1
2 Customer meeting (2015-09-17)	3
3 Customer meeting (2015-09-25)	4
4 Customer meeting (2015-10-16)	5
5 Customer meeting (2015-11-24)	6
6 Customer meeting (2015-12-08)	7

1 Customer meeting (2015-09-09)

Date: 2015-09-09

Present: Yannick Smedts, Jan Jacobs, René, Jan, Martijn, Ron, Max

Location: Fontys

Topics discussed:

Initial project meeting, Talk about customer requirements and get insight into project

Notes:

Van Aaken Ict Solutions, zal later het project later helpen (8 weken)

Fleuren heeft 100 hectare fruitbomen per jaar, 2 jaar groeien om naar fruitboeren te gaan. 45 mensen werken per jaar voor Fleuren.

Iedereen moet dezelfde informatie krijgen over welke dingen al zijn gebeurd met de bomen (water hebben gekregen, besproeid zijn) Er moet een mobiele applicatie komen die informatie geeft over de geschiedenis van de boom/blok/soort.

Verder dienen bodem scans, drone foto's en andere precisie agrocultuur middelen geïntegreerd worden.

De data wordt verzameld en moet zichtbaar worden gemaakt voor de collega's in het veld.

Dus een applicatie die kan helpen bij het maken van beslissingen.

Data moet ook aangevuld kunnen worden op locatie.

Gps opgeslagen en hoe lang je bezig bent met bijvoorbeeld irrigatie.

Connected farm scout

Connected farm field

Crop-er

+31638752419 > Whatsapp Group

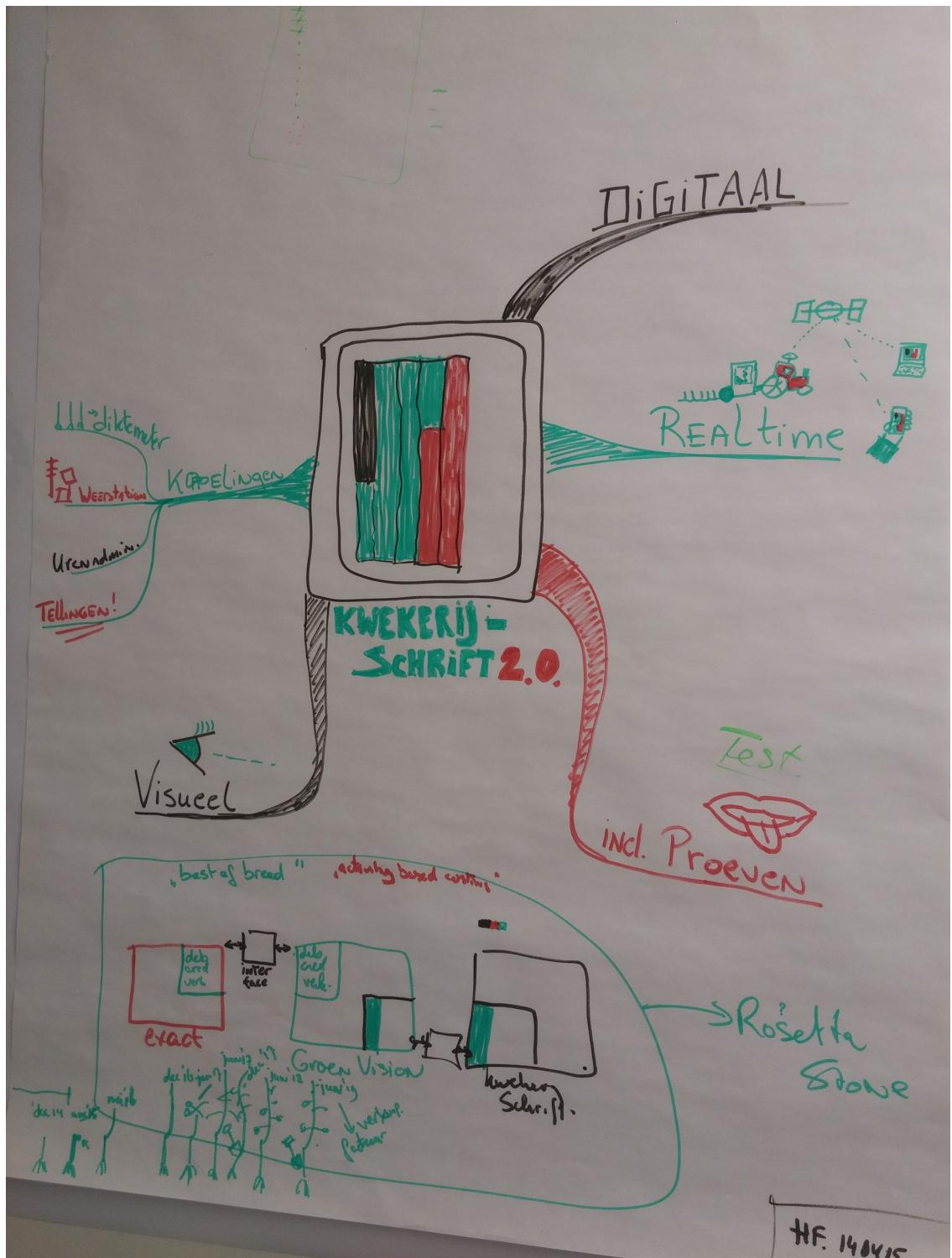
planning@fleuren.net

Yannick neemt contact op met Randy voor een demo van het programma wat we willen gebruiken.

Elke 2 weken communicatie over status

Tussendoor met kritieke punten

2015-09-17 tot 2015-10-17 = Yannick will be on holiday



2 Customer meeting (2015-09-17)

Date: 2015-09-17

Present: Jacob van den Boorne, Randy Wilbrink, René, Jan, Martijn, Max

Location: Reusel

Topics discussed:

How does the system at van den Boorne aardappelen work.

Notes:

Jacob is very far with the implementation of precision farming using modern technologies.

He can generate overviews of the fields and add different overlays on a map.

In the system he can also see weather data and soil scans.

The only thing really missing from the Crop-r app is the geocaching.

It also turned out that a project plan had already been made by VAA. We asked randy to send it to us. He can also see pictures which have been taken from the tractor when it was driving down the fields. Jacob seems to be interested in having students work for him maybe in a coming sofa?

The system shown by Jacob seems to be the exact thing we are supposed to create during the SoFa.

3 Customer meeting (2015-09-25)

Date: 2015-09-25

Present: Han Fleuren, Randy Wilbrink, René, Jan, Martijn, Max, Ron

Location: Fleuren Baarlo

Topics discussed:

How to continue with the project since it already exists for 90% as crop-r

Notes:

The meeting was organized to see how to continue with the project after seeing that Jacob van den Borne was already using an app with almost all the required functionality.

We cleared up the possible conflict of interest and asked Randy to provide us with technical information. We also clarified who the customer is and what the goals are for each party participating in the project. And why crop-r is not a problem but more an idea of in which direction our application should go. Further more Han gave some motivational speeches to get the team a bit more focussed.

4 Customer meeting (2015-10-16)

Date: 2015-10-16

Present: Randy Wilbrink (skype), René, Jan, Martijn, Max, Ron

Location: Fontys

Topics discussed:

Demo and status update

Notes:

Attending Persons

- * Randy Wilbrink
- * Ron Gebauer
- * Jan Kerkenhoff
- * Rene Karoff
- * Martijn Bonajo
- * Max van der Linden

Agenda

- * Show and discuss demo
- * Show and discuss mockups
- * General feedback
- * Whats next?

Discussion

- * There was a discussion about what icons should be used, this however did not seem to have a high priority.
- * When taking a picture on the field the accuracy should be shown and stored.
- * When taking a picture on the field the coordinates should be stored.
- * Randy pointed out that he was satisfied with the progress so far.
- * For the next sprint the focus will be on overlaying fields on the map and finalizing the map menu.
- * Small meetings like this are more efficient through Skype since it does not involve traveling.

Action points

- * Continue the development of the app with focus on overlaying the fields on the map
- * Ask Yannick for GPS data

5 Customer meeting (2015-11-24)

Date: 2015-11-24

Present: Yannick Smedts, René, Jan, Martijn, Max, Ron

Location: Fontys

Topics discussed:

Demo and status update

Notes:

Yannick has just returned from Holiday. This meeting is to catch him up with the status of the project.

He was really satisfied and surprised with the progress we had made so far.

6 Customer meeting (2015-12-08)

Date: 2015-12-08

Present: Yannick Smedts (skype), Han Fleuren (skype), Randy Wilbrink (skype),

René, Jan, Martijn, Max, Ron

Location: Fontys

Topics discussed:

Demo and status update

Notes:

Show the last touches to the development of the app.

All stakeholders were very satisfied with the result.

Received applause from Yannick and Han

Decision was made to do a 2 week document sprint in stead of 1 because Han, Yannick and Randy were already satisfied with the implementation of the app.

Han told us that a golf ball has 320 dimples and weighs 45 grams.

Chapter 12

Minutes



MINUTES
2015 SOFA GTL
TREEWATCH PROJECT



Max van der Linden

2016-01-16

Document information

Document name: Minutes
Document owner: Max van der Linden
Company/Organisation: Fleuren Baarlo
Contact person: Max van der Linden, Group leader
Date: 2016-01-16
Place: Fontys University of Applied Science Venlo
Authors: Max van der Linden
max.vanderlinden@student.fontys.nl
2209349

Contents

Document information	1
1 Minutes SoFa (2015-09-01)	1
2 Minutes SoFa (2015-09-08)	2
3 Minutes SoFa (2015-09-15)	3
4 Minutes SoFa (2015-09-22)	4
5 Minutes SoFa (2015-09-29)	5
6 Minutes SoFa (2015-10-06)	6
7 Minutes SoFa (2015-10-13)	7
8 Minutes SoFa (2015-11-03)	8
9 Minutes SoFa (2015-11-10)	9
10 Minutes SoFa (2015-11-17)	10
11 Minutes SoFa (2015-11-24)	11
12 Minutes SoFa (2015-12-01)	12
13 Minutes SoFa (2015-12-08)	13
14 Minutes SoFa (2015-12-15)	14

1 Minutes SoFa (2015-09-01)

Date: 2015-09-01

Present: Ron, René, Jan, Martijn, Max

Absent: -

Late: -

Decisions made:

Defined some rules and roles and set up communication network Everyone agreed to the rules and roles

ToDo:

Write down personal goals

Argumentation for mobile or web app

Create overall planning

Create and send questions to Yannick

Define rules and roles

Create group website

Project plan

Configuration management plan

Quality management plan

2 Minutes SoFa (2015-09-08)

Date: 2015-09-08

Present: Ron, René, Jan, Martijn, Max

Absent: -

Late: -

Decisions made:

Received reply from Yannick stating that we should discuss the questions during the meeting on Wednesday 2015-09-09)

ToDo:

Write down personal goals

Argumentation for mobile or web app

Create overall planning

Create group website

Analyse order

Project plan

Corporate identity

Look into multi platform development

Configuration management plan

Quality management plan

3 Minutes SoFa (2015-09-15)

Date: 2015-09-15

Present: Ron, René, Jan, Martijn, Max

Absent: -

Late: -

Decisions made:

ToDo:

Analyse order

Project plan

Look into multi platform development

Configuration management plan

Quality management plan

4 Minutes SoFa (2015-09-22)

Date: 2015-09-22

Present: Ron, René, Jan, Martijn, Max

Absent: -

Late: -

Decisions made:

No SRS since user stories will be created

ToDo:

Create user stories

Project plan

Prepare for customer meeting

Look into multi platform development

Configuration management plan

Quality management plan

5 Minutes SoFa (2015-09-29)

Date: 2015-09-29

Present: Ron, René, Jan, Martijn, Max

Absent: -

Late: -

Decisions made:

ToDo:

Create user stories

Project plan

Create mockups

Get mockup approval

Look into multi platform development

Configuration management plan

Quality management plan

6 Minutes SoFa (2015-10-06)

Date: 2015-10-06

Present: Ron, René, Jan, Martijn, Max

Absent: -

Late: -

Decisions made:

ToDo:

Project plan

Setup Xamarin

Setup Github

Start development

Configuration management plan

Quality management plan

7 Minutes SoFa (2015-10-13)

Date: 2015-10-13

Present: Ron, René, Jan, Martijn, Max

Absent: -

Late: -

Decisions made:

ToDo:

Project plan

App development

Configuration management plan

Quality management plan

8 Minutes SoFa (2015-11-03)

Date: 2015-11-03

Present: René, Jan, Martijn, Max

Absent: Ron (ill)

Late: -

Decisions made:

ToDo:

Project plan

App development

Create questionnaires

Work on personal development plan

Configuration management plan

Quality management plan

9 Minutes SoFa (2015-11-10)

Date: 2015-11-10

Present: Ron, René, Jan, Martijn, Max

Absent: -

Late: -

Ron had to leave one hour earlier for work.

Decisions made:

ToDo:

App development

Create database

Work on personal development plan

10 Minutes SoFa (2015-11-17)

Date: 2015-11-17

Present: Ron, René, Jan, Martijn, Max

Absent: -

Late: -

Decisions made:

ToDo:

App development

Prepare customer meeting

11 Minutes SoFa (2015-11-24)

Date: 2015-11-24

Present: Ron, René, Jan, Martijn, Max

Absent: -

Late: -

Decisions made:

ToDo:

App development

12 Minutes SoFa (2015-12-01)

Date: 2015-12-01

Present: Ron, Martijn, Max

Absent: -

Late: René, Jan

Decisions made:

ToDo:

App development

Prepare customer meeting

13 Minutes SoFa (2015-12-08)

Date: 2015-12-08

Present: Ron, René, Jan, Martijn, Max

Absent: -

Late: -

Decisions made:

ToDo:

App development

Prepare for final customer meeting

14 Minutes SoFa (2015-12-15)

Date: 2015-12-15

Present: Ron, René, Jan, Martijn, Max

Absent: -

Late: -

Decisions made:

Use 2 week documenting sprint in stead of 1

ToDo:

Create final report

Chapter 13

Scrum



SCRUM
2015 SOFA GTL
TREEWATCH PROJECT



Jan Kerkenhoff
Martijn Bonajo
Max van der Linden
René Karoff
Ron Gebauer

2016-01-17

Document information

Document name: Scrum
Document owner: TreeWatch
Company/Organisation: Fleuren Baarlo
Contact person: Max van der Linden, Group leader
Date: 2016-01-17
Place: Fontys University of Applied Science Venlo
Authors:
Ron Gebauer
r.gebauer@student.fontys.nl
2153294

Contents

Document information	1
1 Scrum Planning	1
1.1 Sprint Planning	1
1.1.1 Sprint Planning Meeting	1
1.1.2 Daily Scrum Meeting	1
1.1.3 Sprint Review Meeting	2
2 User Stories	3
3 Cumulative Flow	6
4 Working Time	7
4.1 Time by Person	7
4.2 Complete Timesheet	8

1 Scrum Planning

The Scrum planning took place at the beginning of the project, which happened between the second and third week of September. During the planning the criteria and time course were set.

The results of the planning were shown inside the following sub sections. Furthermore you will find all the User Stories, the Cumulative Flow and the working hours, consists of a complete overview and a detailed description, in the following sections.

1.1 Sprint Planning

For a Sprint a Time-Box of six calendar days was set, which are 3 days per week. Each Sprint begins with a Kick-Off, the Sprint Planning Meeting, and ends with a Sprint Review Meeting. Furthermore Non-Team-Members, standing for Product Owner and other Stakeholders, had to be disposal for questions during the Sprint's. Emerging new or changed requirements are still only included into a new starting Sprint.

1.1.1 Sprint Planning Meeting

A Sprint Planning Meeting has a Time-Box of two hours and was used to define the Sprint Backlog for each sprint.

The meeting starts by specifying a Sprint Goal and whose important Items decided by the Product Owner. The chosen items were discussed in a discourse with the whole project team to reduce or expand them.

Toward the end of that meeting the Scrum Team sits together alone, creates and assigns the necessary tasks to achieve the objective.

1.1.2 Daily Scrum Meeting

At the beginning of each working day the Scrum Master inquired about the current status of a task and the further planned course.

1.1.3 Sprint Review Meeting

The Sprint Review Meeting took place at the beginning of each Sprint Planning Meeting and had a Time-Box of one hour.

2 User Stories

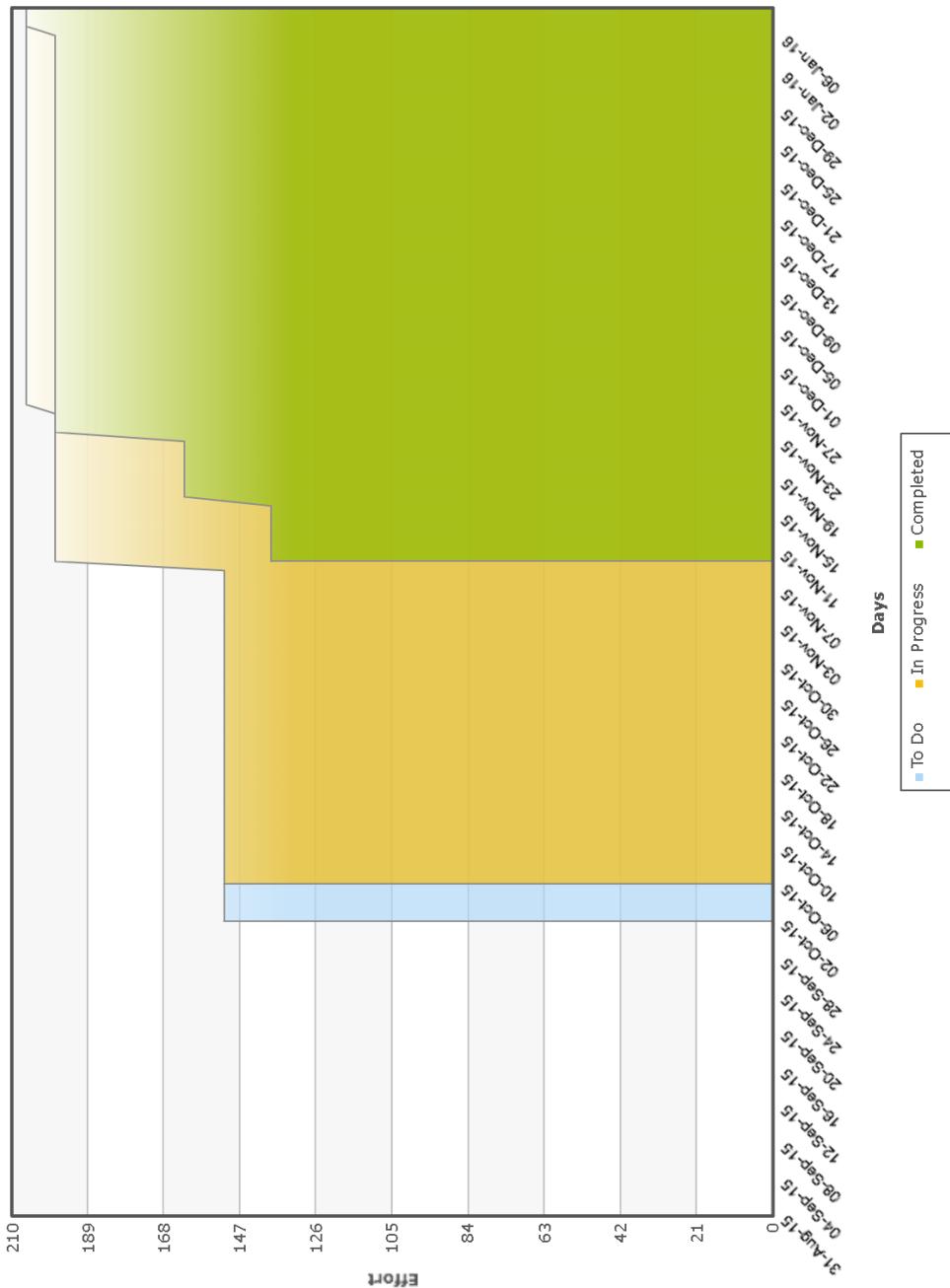
Name	Entity State	Business Value
As a user, I want to correlate the available data in different views so that I can see the relations between different datasets.	Open	Nice To Have
As a user, I want the application to contain different forms to transmit different information about the actual state of the trees on the field (e.g. brown leaves, thin stems...).	Open	Nice To Have
As a user, I want to digitize the kwekerij schrift (Nursery Script)	Open	Nice To Have
As a user, I want the application to analyze the data of the trees and give hints and solutions for problems.	Open	Nice To Have
As a user, I want the application to register as much data as possible, automatically.	Open	Nice To Have
As a user, I want to have a system that manages my customers, their orders and the amount of trees that are growing or are fully grown for them.	Open	Nice To Have
As a user, I want to have an application on my portable devices so that it is possible for me to check all information everywhere.	In Progress	Great

Name	Entity State	Business Value
Data entry	In Progress	Great
As a user, I want my GPS data entered into the system.	Open	Great
As a user, I want my weather data entered into the system.	Open	Good
As a user, I want my soil data gathered into the system.	Open	Nice To Have
As a user, I want my stem data entered into the system.	Open	Nice To Have
As a user, I want the ability to fill out forms within the application.	Open	Good
As a user, I want to be able to take pictures of the trees.	In Progress	Good
Logging	Open	Great
As a user, I want to be able to specify what work I am doing on a field/block/row/tree.	Open	Great
Main functionality	In Progress	Must Have
As a user, I want to have my field represented graphically in the application.	Done	Must Have
As a user, I want to be able to see a chronological ordering of events that happened on trees in a certain row/block's history.	Open	Good
As a user, I want to be able to see where and how long I worked on a field.	Open	Great

Name	Entity State	Business Value
As a user, I want to have overlays for my fields that show collected data.	Open	Must Have
As a user, I want to get my data from a data storage where everything is inside.	Done	Must Have
As a user, I want to have a lot of GPS functionality inside the map.	Open	Nice To Have
As a user, I want to be able to overlay the field with various data.	Open	Nice To Have
As a user, I want to be able to filter and search inside the blocks of a field.	Done	Nice To Have

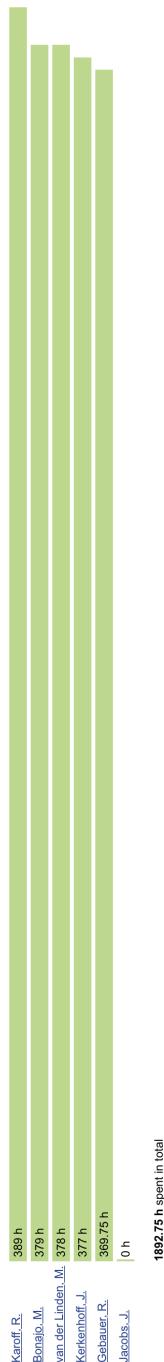
Table 2: User Stories

3 Cumulative Flow



4 Working Time

4.1 Time by Person



4.2 Complete Timesheet

User Last Name	Spent	Role	Date	Activity
Bonajo	3,00	Project Manager	2015-08-31	Initialmeeting
Gebauer	3,00	Project Manager	2015-08-31	Initialmeeting
Karoff	3,00	Project Manager	2015-08-31	Initialmeeting
Kerkenhoff	3,00	Project Manager	2015-08-31	Initialmeeting
van der Linden	3,00	Project Manager	2015-08-31	Initialmeeting
Bonajo	3,00	Project Manager	2015-09-01	Configuration management plan
van der Linden	7,00		2015-09-01	Create personal goal
Kerkenhoff	8,00		2015-09-01	Create Personalgoals
Gebauer	4,00		2015-09-01	Create Scrum Platform
Gebauer	4,00		2015-09-01	Create the project plan
van der Linden	1,00		2015-09-01	Customer contact
Bonajo	5,00		2015-09-01	Personal goals
Karoff	5,00	Project Manager	2015-09-01	Quality management plan
Karoff	3,00		2015-09-01	Write down personal goals

User Last Name	Spent	Role	Date	Activity	
Bonajo	8,00		2015-09-03	Argumentation mobile/web app	mobile/web app
Karoff	4,00		2015-09-03	Argumentation bapp vs native	we-
van der Linden	7,00		2015-09-03	Create overall	plan-
Gebauer	2,00		2015-09-03	Create Scrum	Plat-
Gebauer	4,00		2015-09-03	Create the project	plan
van der Linden	1,00		2015-09-03	Customer contact	
Kerkenhoff	8,00		2015-09-03	Develop Group	web-
Gebauer	2,00		2015-09-03	Personal goals	
Karoff	4,00	Project Manager	2015-09-03	Quality	management
Karoff	4,00	Quality Manager	2015-09-03	plan	
Gebauer	3,00	Project Manager	2015-09-04	Analysis of the received order	
Bonajo	8,00		2015-09-04	Argumentation mobile/web app	mobile/web app
Karoff	8,00		2015-09-04	Argumentation bapp vs native	we-
van der Linden	4,00		2015-09-04	Create overall	plan-

User Last Name	Spent	Role	Date	Activity
van der Linden	4,00		2015-09-04	Create personal goal
Gebauer	4,00		2015-09-04	Create the project plan
Kerkenhoff	8,00		2015-09-04	Develop Group website
Gebauer	1,00		2015-09-04	Personal goals
Karoff	8,00	Quality Manager	2015-09-04	
Bonajo	1,00	Project Manager	2015-09-07	Weekmeeting
Gebauer	1,00	Project Manager	2015-09-07	Weekmeeting
Karoff	1,00	Project Manager	2015-09-07	Weekmeeting
Kerkenhoff	1,00	Project Manager	2015-09-07	Weekmeeting
van der Linden	1,00	Project Manager	2015-09-07	Weekmeeting
Gebauer	3,00	Project Manager	2015-09-08	Analysis of the received order
Kerkenhoff	8,00	Project Manager	2015-09-08	Corporate Identity
van der Linden	5,00		2015-09-08	Look into multi platform dev
Gebauer	1,00		2015-09-08	Meeting preparation

User Last Name	Spent	Role	Date	Activity
van der Linden	2,00		2015-09-08	Prepare questions
Bonajo	8,00		2015-09-08	Research Multiplatform
Gebauer	4,00		2015-09-08	Rework personal goals
van der Linden	1,00		2015-09-08	setup Meeting
Karoff	8,00	Developer	2015-09-08	
Bonajo	4,00	Project Manager	2015-09-09	Customer meeting
Gebauer	4,00	Project Manager	2015-09-09	Customer meeting
Karoff	4,00	Project Manager	2015-09-09	Customer meeting
Kerkenhoff	4,00	Project Manager	2015-09-09	Customer meeting
van der Linden	4,00	Project Manager	2015-09-09	Customer meeting
Bonajo	8,00	Project Manager	2015-09-10	Analysis of the received order
Gebauer	8,00	Project Manager	2015-09-10	Analysis of the received order
Karoff	8,00	Project Manager	2015-09-10	Analysis of the received order
van der Linden	8,00	Project Manager	2015-09-10	Analysis of the received order

User Last Name	Spent	Role	Date	Activity
Kerkenhoff	8,00	Project Manager	2015-09-10	Corporate Identity
Bonajo	4,00	Project Manager	2015-09-11	Analysis of the received order
Karoff	4,00	Project Manager	2015-09-11	Analysis of the received order
van der Linden	2,00	Project Manager	2015-09-11	Analysis of the received order
Kerkenhoff	4,00	Project Manager	2015-09-11	Corporate Identity
van der Linden	1,00		2015-09-11	Prepare questions
van der Linden	1,00		2015-09-11	setup Meeting
Bonajo	1,00	Project Manager	2015-09-14	Weekmeeting
Gebauer	1,00	Project Manager	2015-09-14	Weekmeeting
Karoff	1,00	Project Manager	2015-09-14	Weekmeeting
Kerkenhoff	1,00	Project Manager	2015-09-14	Weekmeeting
van der Linden	1,00	Project Manager	2015-09-14	Weekmeeting
Karoff	4,00		2015-09-15	Analyse conflict of interest
Bonajo	8,00	Project Manager	2015-09-15	Analysis of the received order

User Last Name	Spent	Role	Date	Activity
Gebauer	8,00	Project Manager	2015-09-15	Analysis of the received order
Karoff	4,00	Project Manager	2015-09-15	Analysis of the received order
van der Linden	4,00	Project Manager	2015-09-15	Analysis of the received order
van der Linden	4,00		2015-09-15	Conflict of interest resolution
Kerkenhoff	2,00	Project Manager	2015-09-15	Corporate Identity
Kerkenhoff	6,00	Quality Manager	2015-09-15	Improve website look and feel
Bonajo	8,00	Project Manager	2015-09-17	Customer meeting
Karoff	8,00	Project Manager	2015-09-17	Customer meeting
Kerkenhoff	8,00	Project Manager	2015-09-17	Customer meeting
van der Linden	8,00	Project Manager	2015-09-17	Customer meeting
Bonajo	8,00	Project Manager	2015-09-18	Analysis of the received order
Gebauer	8,00	Project Manager	2015-09-18	Analysis of the received order
Karoff	8,00	Project Manager	2015-09-18	Analysis of the received order

User Last Name	Spent	Role	Date	Activity
Kerkenhoff	4,00	Quality Manager	2015-09-18	Analysis of the received order
van der Linden	6,00	Project Manager	2015-09-18	Analysis of the received order
van der Linden	2,00		2015-09-18	Conflict of interest resolution
Kerkenhoff	4,00	Quality Manager	2015-09-18	Research testing for web apps
Bonajo	1,00	Project Manager	2015-09-21	Weekmeeting
Gebauer	1,00	Project Manager	2015-09-21	Weekmeeting
Karoff	1,00	Project Manager	2015-09-21	Weekmeeting
Kerkenhoff	1,00	Project Manager	2015-09-21	Weekmeeting
van der Linden	1,00	Project Manager	2015-09-21	Weekmeeting
Bonajo	8,00	Project Manager	2015-09-22	Analysis of the received order
Gebauer	2,00	Project Manager	2015-09-22	Analysis of the received order
Karoff	8,00	Project Manager	2015-09-22	Analysis of the received order
van der Linden	8,00	Project Manager	2015-09-22	Analysis of the received order

User Last Name	Spent	Role	Date	Activity
Kerkenhoff	8,00	Quality Manager	2015-09-22	look into mobile app with Cordova
Gebauer	6,00		2015-09-22	Talk about the meeting last Friday
Bonajo	8,00	Project Manager	2015-09-24	Analysis of the received order
Gebauer	4,00	Project Manager	2015-09-24	Analysis of the received order
Karoff	4,00	Project Manager	2015-09-24	Analysis of the received order
Kerkenhoff	8,00	Quality Manager	2015-09-24	Analysis of the received order
van der Linden	8,00	Project Manager	2015-09-24	Analysis of the received order
Gebauer	4,00		2015-09-24	Plan meetings talk for Friday
Karoff	4,00	Project Manager	2015-09-24	Quality management plan
Bonajo	5,00	Project Manager	2015-09-25	Analysis of the received order
Gebauer	3,00	Project Manager	2015-09-25	Analysis of the received order
Karoff	5,00	Project Manager	2015-09-25	Analysis of the received order
van der Linden	5,00	Project Manager	2015-09-25	Analysis of the received order

User Last Name	Spent	Role	Date	Activity
Gebauer	2,00	Developer	2015-09-25	Create protocol
Bonajo	3,00	Project Manager	2015-09-25	Customer meeting
Gebauer	3,00	Project Manager	2015-09-25	Customer meeting
Karoff	3,00	Project Manager	2015-09-25	Customer meeting
Kerkenhoff	3,00	Project Manager	2015-09-25	Customer meeting
van der Linden	3,00	Project Manager	2015-09-25	Customer meeting
Kerkenhoff	5,00	Quality Manager	2015-09-25	look into mobile app with Cordova
Bonajo	1,00	Project Manager	2015-09-28	Weekmeeting
Gebauer	1,00	Project Manager	2015-09-28	Weekmeeting
Karoff	1,00	Project Manager	2015-09-28	Weekmeeting
Kerkenhoff	1,00	Project Manager	2015-09-28	Weekmeeting
van der Linden	1,00	Project Manager	2015-09-28	Weekmeeting
Bonajo	8,00	Project Manager	2015-09-29	Analysis of the received order

User Last Name	Spent	Role	Date	Activity
Gebauer	5,00	Project Manager	2015-09-29	Analysis of the received order
Kerkenhoff	8,00	Developer	2015-09-29	Analysis of the received order
van der Linden	8,00	Project Manager	2015-09-29	Analysis of the received order
Karoff	8,00		2015-09-29	Create Mockups
Gebauer	3,00		2015-09-29	Read into Xamarin
Bonajo	8,00	Project Manager	2015-10-01	Analysis of the received order
Gebauer	8,00	Project Manager	2015-10-01	Analysis of the received order
Kerkenhoff	8,00	Developer	2015-10-01	Analysis of the received order
van der Linden	8,00	Project Manager	2015-10-01	Analysis of the received order
Karoff	8,00		2015-10-01	Create Mockups
Bonajo	8,00	Project Manager	2015-10-02	Analysis of the received order
Gebauer	8,00	Project Manager	2015-10-02	Analysis of the received order
Kerkenhoff	8,00	Developer	2015-10-02	Analysis of the received order
van der Linden	8,00	Project Manager	2015-10-02	Analysis of the received order

User Last Name	Spent	Role	Date	Activity
Karoff	6,00		2015-10-02	Create Mockups
Karoff	2,00	Project Manager	2015-10-02	Quality management plan
Bonajo	1,00	Project Manager	2015-10-05	Weekmeeting
Gebauer	1,00	Developer	2015-10-05	Weekmeeting
Karoff	1,00	Project Manager	2015-10-05	Weekmeeting
Kerkenhoff	1,00	Project Manager	2015-10-05	Weekmeeting
van der Linden	1,00	Project Manager	2015-10-05	Weekmeeting
Gebauer	1,00		2015-10-06	Create App Icons
Gebauer	2,00	Developer	2015-10-06	Create Bottom Bar
Karoff	0,25	Quality Manager	2015-10-06	Create Bottom Bar
Gebauer	2,00	Developer	2015-10-06	Create Mainframe
Karoff	5,50	Quality Manager	2015-10-06	Create Mainframe
Kerkenhoff	7,00	Developer	2015-10-06	Create Mainframe
Karoff	1,00	Developer	2015-10-06	Create Map View
Kerkenhoff	1,00	Developer	2015-10-06	Create Map View
Gebauer	1,00		2015-10-06	Get application running

User Last Name	Spent	Role	Date	Activity
Bonajo	2,50		2015-10-06	Help others setup
Gebauer	1,00		2015-10-06	Install Xamarin
Gebauer	1,00		2015-10-06	Planning this week
van der Linden	1,00		2015-10-06	Send Update
Karoff	1,25		2015-10-06	Set up Xamarin
van der Linden	7,00		2015-10-06	Set Up Xamarin
Bonajo	5,50		2015-10-06	Setup Xamarin
Gebauer	8,00	Developer	2015-10-08	Create Bottom Bar
Karoff	1,50	Developer	2015-10-08	Create Mainframe
Karoff	4,00	Developer	2015-10-08	Create Map View
Kerkenhoff	8,00	Developer	2015-10-08	Create Map View
Bonajo	3,00		2015-10-08	Help others setup
Karoff	2,50		2015-10-08	Set up Xamarin
van der Linden	8,00		2015-10-08	Setup Mac
Bonajo	5,00		2015-10-08	Setup xamarin on macbook
van der Linden	1,00	Developer	2015-10-09	Add Fields to Map View
Bonajo	2,00	Project Manager	2015-10-09	Configuration management plan
Gebauer	2,00	Developer	2015-10-09	Create Bottom Bar
Gebauer	1,50	Developer	2015-10-09	Create Bottom Bar

User Last Name	Spent	Role	Date	Activity
Karoff	0,50	Quality Manager	2015-10-09	Create Bottom Bar
Gebauer	1,50	Developer	2015-10-09	Create Mainframe
Karoff	5,00	Quality Manager	2015-10-09	Create Mainframe
Karoff	0,50	Quality Manager	2015-10-09	Create Map Menu and add Searchbar, Field-list
Karoff	1,50	Quality Manager	2015-10-09	Create Map View
Kerkenhoff	8,00	Developer	2015-10-09	Create Map View
Bonajo	6,00		2015-10-09	Create popup menu
van der Linden	7,00	Developer	2015-10-09	Create PopUp Menu
Gebauer	3,00	Developer	2015-10-09	Create Top Bar
Karoff	0,50	Quality Manager	2015-10-09	Create Top Bar
Bonajo	1,00	Project Manager	2015-10-12	Weekmeeting
Gebauer	1,00	Developer	2015-10-12	Weekmeeting
Karoff	1,00	Project Manager	2015-10-12	Weekmeeting
Kerkenhoff	1,00	Project Manager	2015-10-12	Weekmeeting
van der Linden	1,00	Project Manager	2015-10-12	Weekmeeting

User Last Name	Spent	Role	Date	Activity
Bonajo	4,00		2015-10-13	Added android emulator on mac
Gebauer	0,50	Developer	2015-10-13	Create Mainframe
Bonajo	2,00	Developer	2015-10-13	Create Map Menu and add Searchbar, Field-list
Gebauer	0,50	Scrum Master	2015-10-13	Create Map Menu and add Searchbar, Field-list
Gebauer	0,20	Scrum Master	2015-10-13	Create Map Menu and add Searchbar, Field-list
Gebauer	1,00	Scrum Master	2015-10-13	Create Map Menu and add Searchbar, Field-list
Gebauer	2,80	Scrum Master	2015-10-13	Create Map Menu and add Searchbar, Field-list
Gebauer	3,00	Scrum Master	2015-10-13	Create Map Menu and add Searchbar, Field-list
van der Linden	7,00	Developer	2015-10-13	Create Map Menu and add Searchbar, Field-list
Karoff	4,00	Developer	2015-10-13	Create Map View
Karoff	0,50	Quality Manager	2015-10-13	Create Map View

User Last Name	Spent	Role	Date	Activity
Kerkenhoff	8,00	Developer	2015-10-13	Create Map View
Bonajo	2,00		2015-10-13	Help others setup android emulator
Karoff	3,50	Developer	2015-10-13	Research which Test-Driven-Methods will be used
van der Linden	1,00		2015-10-13	send meeting invite
Karoff	1,50	Quality Manager	2015-10-15	Create Mainframe
van der Linden	8,00	Developer	2015-10-15	Create Mainframe
Bonajo	6,00	Developer	2015-10-15	Create Map Menu and add Searchbar, Field-list
Gebauer	4,00	Scrum Master	2015-10-15	Create Map Menu and add Searchbar, Field-list
Gebauer	4,00	Scrum Master	2015-10-15	Create Map Menu and add Searchbar, Field-list
Karoff	2,50	Developer	2015-10-15	Create Map View
Bonajo	2,00		2015-10-15	Help others setup android emulator
Karoff	4,00	Quality Manager	2015-10-15	Research which Test-Driven-Methods will be used

User Last Name	Spent	Role	Date	Activity
Kerkenhoff	8,00	Developer	2015-10-15	Research which Test-Driven-Methods will be used
Bonajo	1,00	Developer	2015-10-16	Configuration management plan
Gebauer	5,25	Developer	2015-10-16	Create Mainframe
Karoff	1,50	Quality Manager	2015-10-16	Create Mainframe
Bonajo	4,00	Developer	2015-10-16	Create Map Menu and add Searchbar, Field-list
Gebauer	0,25	Scrum Master	2015-10-16	Create Map Menu and add Searchbar, Field-list
Gebauer	0,50	Scrum Master	2015-10-16	Create Map Menu and add Searchbar, Field-list
Karoff	2,50	Quality Manager	2015-10-16	Create Map View
Bonajo	2,00	Project Manager	2015-10-16	Customer meeting
Gebauer	2,00	Project Manager	2015-10-16	Customer meeting
Karoff	2,00	Project Manager	2015-10-16	Customer meeting
Kerkenhoff	2,00	Project Manager	2015-10-16	Customer meeting

User Last Name	Spent	Role	Date	Activity
van der Linden	2,00	Project Manager	2015-10-16	Customer meeting
van der Linden	4,00	Developer	2015-10-16	Fix emulator issues
Bonajo	1,00	Developer	2015-10-16	Help others setup android emulator
Karoff	2,00	Quality Manager	2015-10-16	Quality management plan
Kerkenhoff	6,00	Quality Manager	2015-10-16	Research which Test-Driven-Methods will be used
van der Linden	2,00	Developer	2015-10-16	Setup sprint meeting
Gebauer	2,00	Developer	2015-10-23	Create new project version to us an assembly.
Gebauer	6,00	Developer	2015-10-23	Improve issue that android simulator does not work fine.
Bonajo	1,00	Project Manager	2015-11-02	Weekmeeting
Gebauer	1,00	Developer	2015-11-02	Weekmeeting
Karoff	1,00	Project Manager	2015-11-02	Weekmeeting
Kerkenhoff	1,00	Project Manager	2015-11-02	Weekmeeting
van der Linden	1,00	Project Manager	2015-11-02	Weekmeeting

User Last Name	Spent	Role	Date	Activity
Karoff	8,00	Project Manager	2015-11-03	As a user, I want to have overlays for my fields that show collected data.
Bonajo	2,00	Developer	2015-11-03	Configuration management plan
Bonajo	6,00	Developer	2015-11-03	Create Map Menu and add Searchbar, Field-list
Kerkenhoff	8,00	Developer	2015-11-03	Create Map Menu and add Searchbar, Field-list
van der Linden	5,00	Developer	2015-11-03	Create questionnaire
van der Linden	3,00	Developer	2015-11-03	Start Note page
Karoff	1,00	Quality Manager	2015-11-05	As a user, I want to have overlays for my fields that show collected data.
Karoff	7,00	Quality Manager	2015-11-05	As a user, I want to have overlays for my fields that show collected data.
Bonajo	8,00	Developer	2015-11-05	Create Map Menu and add Searchbar, Field-list
Gebauer	1,50	Scrum Master	2015-11-05	Create Map Menu and add Searchbar, Field-list

User Last Name	Spent	Role	Date	Activity
Kerkenhoff	8,00	Developer	2015-11-05	Create Map Menu and add Searchbar, Field-list
Gebauer	2,13	Developer	2015-11-05	Create note view and make pictures with comment
Gebauer	0,17	Scrum Master	2015-11-05	Improve code and set the icons in right way
Gebauer	1,00	Scrum Master	2015-11-05	Improve code and set the icons in right way
Gebauer	2,50	Scrum Master	2015-11-05	Improve code and set the icons in right way
Gebauer	0,70	Developer	2015-11-05	Open In-Call Status Bar ... change size of window content inside iOS
van der Linden	8,00	Developer	2015-11-05	Try to get project running
Kerkenhoff	8,00	Developer	2015-11-06	Add Fields to Map View
Karoff	8,00	Quality Manager	2015-11-06	As a user, I want to have overlays for my fields that show collected data.
van der Linden	3,00	Developer	2015-11-06	Create ER diagram
Bonajo	8,00	Developer	2015-11-06	Create & design Annotations in both devices

User Last Name	Spent	Role	Date	Activity
Gebauer	3,00	Developer	2015-11-06	Create Map Menu and add Searchbar, Fieldlist
Gebauer	3,00	Scrum Master	2015-11-06	Create note view and make pictures with comment
van der Linden	2,00		2015-11-06	Fix Android Listmenu bug
Gebauer	2,00	Developer	2015-11-06	Open In-Call Status Bar ... change size of window content inside iOS
van der Linden	3,00	Developer	2015-11-06	Start Note page
Bonajo	1,00	Project Manager	2015-11-09	Weekmeeting
Gebauer	1,00	Developer	2015-11-09	Weekmeeting
Karoff	1,00	Project Manager	2015-11-09	Weekmeeting
Kerkenhoff	1,00	Project Manager	2015-11-09	Weekmeeting
van der Linden	1,00	Project Manager	2015-11-09	Weekmeeting
Karoff	1,00	Quality Manager	2015-11-10	Add Fields to Map View
Gebauer	2,00	Developer	2015-11-10	Blocks are not rectangles
Karoff	6,00	Developer	2015-11-10	Create data storage

User Last Name	Spent	Role	Date	Activity
Karoff	1,00	Quality Manager	2015-11-10	Create Map Menu and add Searchbar, Field-list
Gebauer	0,25	Scrum Master	2015-11-10	Create note view and make pictures with comment
Gebauer	0,75		2015-11-10	Create Questionnaire
van der Linden	5,00	Project Manager	2015-11-10	Database
van der Linden	3,00	Project Manager	2015-11-10	Fix api packages
Bonajo	8,00	Developer	2015-11-10	Make them clickable
Kerkenhoff	8,00	Developer	2015-11-10	Make them clickable
Gebauer	3,00	Scrum Master	2015-11-10	Plan the new Release
Gebauer	2,00	Scrum Master	2015-11-10	resolve issue with android virtual machine
Bonajo	4,00	Developer	2015-11-12	Create & design Annotations in both devices
Karoff	8,00	Developer	2015-11-12	Create data storage
van der Linden	8,00	Project Manager	2015-11-12	Fix api packages
Bonajo	3,00	Developer	2015-11-12	Helped database design
Kerkenhoff	8,00	Developer	2015-11-12	Make them clickable

User Last Name	Spent	Role	Date	Activity
Gebauer	2,00	Scrum Master	2015-11-12	Plan the new Release
Gebauer	0,12	Developer	2015-11-12	Remove hardcoded colors -> into database table?
Bonajo	1,00	Developer	2015-11-12	Search bar is not clear after click on field
Gebauer	5,88	Developer	2015-11-12	Search bar is not clear after click on field
Bonajo	3,00	Configuration Manager	2015-11-13	2 Instances of View-Model
Gebauer	1,00	Developer	2015-11-13	2 Instances of View-Model
Gebauer	0,50	Developer	2015-11-13	2 Instances of View-Model
Karoff	6,00	Developer	2015-11-13	Create data storage
Bonajo	5,00	Developer	2015-11-13	Make them clickable
Gebauer	0,12	Developer	2015-11-13	Make them clickable
Gebauer	1,00	Developer	2015-11-13	Make them clickable
Gebauer	0,50	Developer	2015-11-13	Make them clickable
Gebauer	1,00	Developer	2015-11-13	Make them clickable
Gebauer	1,38	Developer	2015-11-13	Make them clickable
Kerkenhoff	8,00	Developer	2015-11-13	Make them clickable

User Last Name	Spent	Role	Date	Activity
van der Linden	8,00	Project Manager	2015-11-13	Make them clickable
Karoff	2,00	Developer	2015-11-13	Quality management plan
Gebauer	2,50	Scrum Master	2015-11-13	update targetprozess
Bonajo	1,00	Project Manager	2015-11-16	Weekmeeting
Gebauer	1,00	Scrum Master	2015-11-16	Week meeting
Karoff	1,00	Project Manager	2015-11-16	Weekmeeting
Kerkenhoff	1,00	Developer	2015-11-16	Weekmeeting
van der Linden	1,00	Project Manager	2015-11-16	Weekmeeting
van der Linden	8,00	Developer	2015-11-17	Add additional informations into the field
Karoff	1,00	Quality Manager	2015-11-17	Blocks are not rectangles
Bonajo	1,00	Developer	2015-11-17	Configuration management plan
Gebauer	6,00	Developer	2015-11-17	Create & design Annotations in both devices
Karoff	1,00	Quality Manager	2015-11-17	Create data storage

User Last Name	Spent	Role	Date	Activity
Karoff	4,00	Developer	2015-11-17	Create data storage
Kerkenhoff	8,00	Developer	2015-11-17	Create data storage
Bonajo	7,00	Developer	2015-11-17	Make them clickable
Gebauer	2,00	Developer	2015-11-17	Make them clickable
Karoff	2,00	Developer	2015-11-17	Remove hard-coded colors -> into database table?
van der Linden	4,00	Developer	2015-11-19	Add additional informations into the field
van der Linden	4,00	Developer	2015-11-19	Add different blocks to fields
Gebauer	1,50	Developer	2015-11-19	Change Navigation to tabbedbar
Kerkenhoff	8,00	Developer	2015-11-19	Create data storage
Gebauer	2,00	Developer	2015-11-19	Finalize design of detail information view (settings view iOS)
Gebauer	2,00	Developer	2015-11-19	Finalize design of detail information view (settings view iOS)
Gebauer	1,50	Developer	2015-11-19	Finalize design of detail information view (settings view iOS)
Gebauer	1,00	Developer	2015-11-19	Finalize design of detail information view (settings view iOS)

User Last Name	Spent	Role	Date	Activity
Bonajo	3,00	Developer	2015-11-19	Helped database design
Karoff	8,00	Developer	2015-11-19	Remove hard-coded colors -> into database table?
Bonajo	5,00	Developer	2015-11-19	Zoom to a block by clicking it.
Karoff	2,50	Developer	2015-11-20	Add a loading screen to the application.
Karoff	0,50	Quality Manager	2015-11-20	Add a loading screen to the application.
Kerkenhoff	3,00	Developer	2015-11-20	Add a loading screen to the application.
Karoff	4,50	Developer	2015-11-20	Change field polygons to outline
Gebauer	3,00	Developer	2015-11-20	Change Navigation to tabbedbar
Gebauer	2,00	Developer	2015-11-20	Change Navigation to tabbedbar
Karoff	0,50	Quality Manager	2015-11-20	Create & design Annotations in both devices
Kerkenhoff	5,00	Developer	2015-11-20	Create data storage
van der Linden	8,00	Project Manager	2015-11-20	Final group dossier

User Last Name	Spent	Role	Date	Activity
Gebauer	1,00	Developer	2015-11-20	Finalize design of detail information view (settings view iOS)
Gebauer	2,00	Developer	2015-11-20	Finalize design of detail information view (settings view iOS)
Bonajo	8,00	Developer	2015-11-20	Zoom to a block by clicking it.
Bonajo	1,00	Project Manager	2015-11-23	Weekmeeting
Gebauer	1,00	Developer	2015-11-23	Weekmeeting
Karoff	1,00	Project Manager	2015-11-23	Weekmeeting
Kerkenhoff	1,00	Project Manager	2015-11-23	Weekmeeting
van der Linden	1,00	Project Manager	2015-11-23	Weekmeeting
Gebauer	0,75	Developer	2015-11-24	Add a info button to MapMenu.
Bonajo	2,00	Project Manager	2015-11-24	Customer meeting
Gebauer	2,00	Project Manager	2015-11-24	Customer meeting
Karoff	2,00	Project Manager	2015-11-24	Customer meeting
Kerkenhoff	2,00	Project Manager	2015-11-24	Customer meeting

User Last Name	Spent	Role	Date	Activity
van der Linden	2,00	Project Manager	2015-11-24	Customer meeting
Karoff	6,00	Developer	2015-11-24	Pair programing
van der Linden	6,00	Project Manager	2015-11-24	Pin and zooming
Kerkenhoff	6,00	Developer	2015-11-24	Remove pins and annotations from map
Gebauer	3,00	Developer	2015-11-24	Remove Warnings in Xamarin projects.
Gebauer	0,25	Developer	2015-11-24	Remove Warnings in Xamarin projects.
Gebauer	1,00	Developer	2015-11-24	Remove Warnings in Xamarin projects.
Bonajo	6,00	Developer	2015-11-24	Take photo/select from album
Gebauer	1,00	Developer	2015-11-26	Add a info button to MapMenu.
Gebauer	5,00	Developer	2015-11-26	Add a info button to MapMenu.
Kerkenhoff	8,00	Developer	2015-11-26	Added CI
Gebauer	5,00	Developer	2015-11-26	If clicked on selected file, does not move back to it.
Karoff	8,00	Developer	2015-11-26	Pair-programing
van der Linden	8,00	Project Manager	2015-11-26	Pin and zooming

User Last Name	Spent	Role	Date	Activity
Bonajo	6,00	Developer	2015-11-26	Set start position to actual position.
Bonajo	2,00		2015-11-26	Take photo/select from album
Gebauer	6,00	Developer	2015-11-27	Add button (go to my position) on the map
Kerkenhoff	8,00	Developer	2015-11-27	Debugging api key issues
van der Linden	8,00	Project Manager	2015-11-27	Final group dossier
Bonajo	2,00	Developer	2015-11-27	If clicked on selected file, does not move back to it.
Karoff	6,00	Developer	2015-11-27	Pair-programming
Karoff	1,00	Quality Manager	2015-11-27	Remove pins and annotations from map
Bonajo	6,00	Developer	2015-11-27	Set start position to actual position.
Karoff	1,00	Quality Manager	2015-11-27	Zoom to a block by clicking it.
Bonajo	1,00	Project Manager	2015-11-30	Weekmeeting
Gebauer	1,00	Developer	2015-11-30	Weekmeeting
Karoff	1,00	Project Manager	2015-11-30	Weekmeeting

User Last Name	Spent	Role	Date	Activity
Kerkenhoff	1,00	Project Manager	2015-11-30	Weekmeeting
van der Linden	1,00	Project Manager	2015-11-30	Weekmeeting
Karoff	4,00	Quality Manager	2015-12-01	Add real fields and their blocks into database.
Bonajo	8,00	Developer	2015-12-01	Geofence
van der Linden	8,00	Project Manager	2015-12-01	Geofencing
Kerkenhoff	8,00		2015-12-01	Helping Rene with KML Parsing (Pair-programming)
Karoff	4,00	Quality Manager	2015-12-01	Pair-programming
Gebauer	8,00	Developer	2015-12-01	refactoring
Karoff	8,00	Developer	2015-12-03	Add real fields and their blocks into database.
Gebauer	8,00	Developer	2015-12-03	filtering blocks, overlay choosing
Bonajo	8,00	Developer	2015-12-03	Geofence
van der Linden	8,00	Project Manager	2015-12-03	Geofencing
Kerkenhoff	8,00		2015-12-03	Helping Rene with KML Parsing (Pair-programming)

User Last Name	Spent	Role	Date	Activity
Gebauer	6,00	Developer	2015-12-04	As a user, I want to be able to filter and search inside the blocks of a field.
Gebauer	4,00	Developer	2015-12-04	Finalize Application for next Release
Bonajo	8,00	Developer	2015-12-04	Geofence
van der Linden	8,00	Project Manager	2015-12-04	Geofencing
Kerkenhoff	8,00		2015-12-04	Helping Rene with KML Parsing (Pair-programming)
Karoff	8,00	Quality Manager	2015-12-04	Pair-programing
Gebauer	4,00	Developer	2015-12-04	refactoring
Bonajo	1,00	Project Manager	2015-12-07	Weekmeeting
Gebauer	1,00	Developer	2015-12-07	Weekmeeting
Karoff	1,00	Project Manager	2015-12-07	Weekmeeting
Kerkenhoff	1,00	Project Manager	2015-12-07	Weekmeeting
van der Linden	1,00	Project Manager	2015-12-07	Weekmeeting

User Last Name	Spent	Role	Date	Activity
Gebauer	0,50	Developer	2015-12-08	As a user, I want to be able to filter and search inside the blocks of a field.
Gebauer	0,75	Developer	2015-12-08	As a user, I want to be able to filter and search inside the blocks of a field.
Gebauer	1,50	Developer	2015-12-08	As a user, I want to be able to filter and search inside the blocks of a field.
Gebauer	3,00	Developer	2015-12-08	Create a menu button on map page to select the overlays on the field.
Bonajo	2,00	Project Manager	2015-12-08	Customer meeting
Gebauer	2,00	Project Manager	2015-12-08	Customer meeting
Karoff	2,00	Project Manager	2015-12-08	Customer meeting
Kerkenhoff	2,00	Project Manager	2015-12-08	Customer meeting
van der Linden	2,00	Project Manager	2015-12-08	Customer meeting
Bonajo	6,00	Developer	2015-12-08	Geofence

User Last Name	Spent	Role	Date	Activity
van der Linden	6,00	Project Manager	2015-12-08	Geofencing
Karoff	6,00	Developer	2015-12-08	Overlay the field with shape files
Kerkenhoff	6,00	Developer	2015-12-08	Overlay the field with shape files
Gebauer	2,00		2015-12-08	Work on block filtering and searching
Karoff	6,50	Developer	2015-12-10	Add real fields and their blocks into database.
Gebauer	3,50	Developer	2015-12-10	Change view of blocks from all blocks to variety
Karoff	0,50	Quality Manager	2015-12-10	Change view of blocks from all blocks to variety
Gebauer	2,00	Developer	2015-12-10	Create a menu button on map page to select the overlays on the field.
Bonajo	8,00	Developer	2015-12-10	Geofence
van der Linden	8,00	Project Manager	2015-12-10	Geofencing
Gebauer	2,50	Developer	2015-12-10	Look for code analysis
Kerkenhoff	8,00	Developer	2015-12-10	Overlay the field with shape files

User Last Name	Spent	Role	Date	Activity
Karoff	1,00	Developer	2015-12-10	Quality management plan
Karoff	8,00	Quality Manager	2015-12-11	Add real fields and their blocks into database.
Bonajo	8,00	Developer	2015-12-11	Geofence
van der Linden	8,00	Project Manager	2015-12-11	Geofencing
Gebauer	5,00	Developer	2015-12-11	Look for code analysis
Kerkenhoff	8,00	Developer	2015-12-11	Overlay the field with shape files
Gebauer	3,00	Developer	2015-12-11	Work on block filtering and searching
Bonajo	1,00	Project Manager	2015-12-14	Weekmeeting
Karoff	1,00	Project Manager	2015-12-14	Weekmeeting
Kerkenhoff	1,00	Project Manager	2015-12-14	Weekmeeting
van der Linden	1,00	Project Manager	2015-12-14	Weekmeeting
Bonajo	5,00	Project Manager	2015-12-15	Customer meeting
Karoff	5,00	Project Manager	2015-12-15	Customer meeting

User Last Name	Spent	Role	Date	Activity
Kerkenhoff	5,00	Project Manager	2015-12-15	Customer meeting
van der Linden	5,00	Project Manager	2015-12-15	Customer meeting
Karoff	3,00	Quality Manager	2015-12-15	Final group dossier
Kerkenhoff	3,00	Project Manager	2015-12-15	Final group dossier
van der Linden	3,00	Project Manager	2015-12-15	Final group dossier
Bonajo	3,00	Developer	2015-12-15	Merging
Bonajo	1,00	Developer	2015-12-17	Final group dossier
Gebauer	8,00	Developer	2015-12-17	Final group dossier
Karoff	8,00	Quality Manager	2015-12-17	Final group dossier
Kerkenhoff	8,00	Project Manager	2015-12-17	Final group dossier
van der Linden	8,00	Project Manager	2015-12-17	Final group dossier
Bonajo	7,00	Developer	2015-12-17	Geofencing
Bonajo	3,00	Developer	2015-12-18	Final group dossier
Gebauer	10,00	Developer	2015-12-18	Final group dossier
Karoff	8,00	Quality Manager	2015-12-18	Final group dossier

User Last Name	Spent	Role	Date	Activity
Kerkenhoff	8,00	Project Manager	2015-12-18	Final group dossier
van der Linden	8,00	Project Manager	2015-12-18	Final group dossier
Bonajo	5,00	Developer	2015-12-18	Geofencing
Bonajo	1,00	Project Manager	2016-01-04	Weekmeeting
Gebauer	1,00	Developer	2016-01-04	Weekmeeting
Karoff	1,00	Project Manager	2016-01-04	Weekmeeting
Kerkenhoff	1,00	Project Manager	2016-01-04	Weekmeeting
van der Linden	1,00	Project Manager	2016-01-04	Weekmeeting
Bonajo	8,00	Project Manager	2016-01-05	Final group dossier
Gebauer	8,00	Developer	2016-01-05	Final group dossier
Karoff	8,00	Quality Manager	2016-01-05	Final group dossier
Kerkenhoff	8,00	Project Manager	2016-01-05	Final group dossier
van der Linden	8,00	Project Manager	2016-01-05	Final group dossier
Bonajo	5,00	Project Manager	2016-01-07	Configuration management plan

User Last Name	Spent	Role	Date	Activity
van der Linden	6,00	Project Manager	2016-01-07	Damage control
Bonajo	4,00	Project Manager	2016-01-07	Final group dossier
Gebauer	12,00	Developer	2016-01-07	Final group dossier
Karoff	8,00	Quality Manager	2016-01-07	Final group dossier
Kerkenhoff	8,00	Project Manager	2016-01-07	Final group dossier
van der Linden	3,00	Project Manager	2016-01-07	Final group dossier
Bonajo	1,00	Project Manager	2016-01-08	Configuration management plan
van der Linden	8,00	Project Manager	2016-01-08	Damage control
Bonajo	8,00	Project Manager	2016-01-08	Final group dossier
Gebauer	8,00	Developer	2016-01-08	Final group dossier
Karoff	8,00	Project Manager	2016-01-08	Final group dossier
Kerkenhoff	8,00	Project Manager	2016-01-08	Final group dossier

1892,75

476 Entries

Table 4: Timesheet

Chapter 14

Final Presentation



SoFa Project GreenTechLab

Boomkwekerij
Fleuren



Introduction

- Fleuren Baarlo
- Project members and roles
- Assignment



Contents

- Project definition
- Planning
- Analysis
- User stories
- Design
- Implementation
- Live demo
- Conclusion
- Questions?



Project definition

- Problem description
- Stakeholders



Planning

Scrum

- Analysis sprints (2 Sprints)
- Design sprints (1 Sprint)
- Implementation sprints (6 Sprints)
- Documentation sprints (1 Sprint)

Each sprint took 2 weeks.



Analysis

- Meeting Fleuren
- Meeting van den Boorne
- User stories



User Stories

- 18 User stories in total
- Only 2 stories with highest priority:
 - As a User, I want the ability to overlay the visualisations on top of the field.
 - As a User, I want to have my field represented graphically in the application.
- User stories regarding Geofencing, History view and a ToDo-List where prioritized from 2 to 6



Design

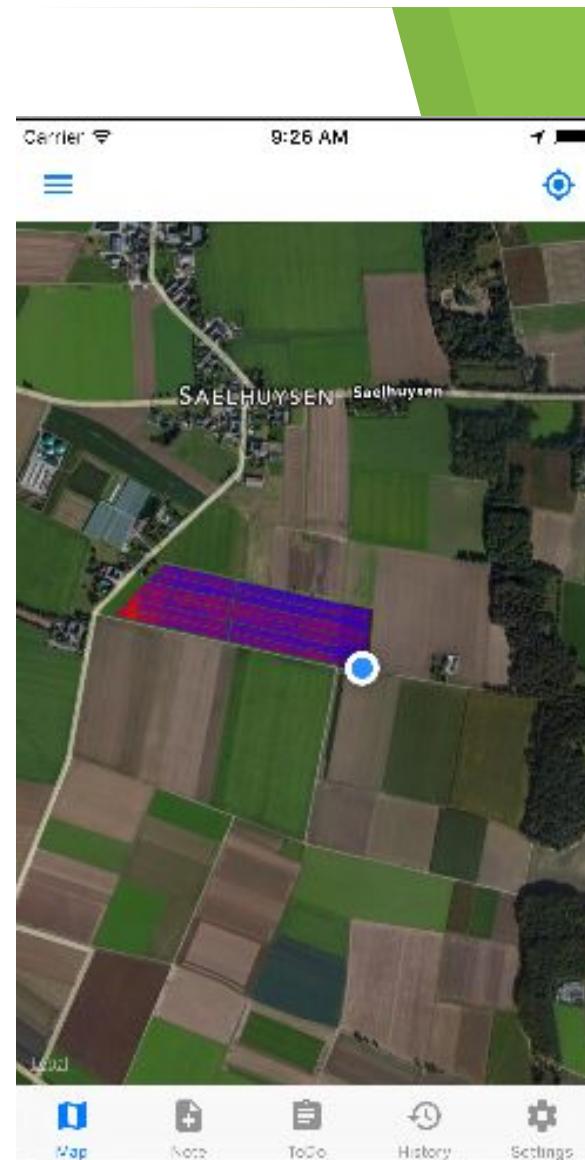
Mockups





Implementation

- Multi platform
- Xamarin
- License problem
- Setup App
- Setup Map
- Add overlays (performance issues)
- Add geofencing





Live demo



Conclusion

- Learned a lot
- Good teamwork
- Golffball





Questions?



Chapter 15

Important Code

Contents

1	TreeWatch/View/CustomTabPage.cs	3
2	TreeWatch/View/Map/MapMasterDetailPage.cs	4
3	TreeWatch/View/Map/MapNavigationPage.cs	5
4	TreeWatch/View/Map/MapContentPage.xaml	6
5	TreeWatch/View/Map/MapContentPage.xaml.cs	7
6	TreeWatch/View/Map/.../MapMenuContentPage.xaml	10
7	TreeWatch/View/Map/.../MapMenuContentPage.xaml.cs	12
8	TreeWatch/View/Map/.../VarietiesInformationContentPage.xaml	14
9	TreeWatch/View/Map/.../VarietiesInformationContentPage.xaml.cs	16
10	TreeWatch/View/Map/.../FieldInformationContentPage.xaml	17
11	TreeWatch/View/Map/.../FieldInformationContentPage.xaml.cs	19
12	TreeWatch/View/Map/.../BlocksInformationContentPage.xaml	20
13	TreeWatch/View/Map/.../BlocksInformationContentPage.xaml.cs	22
14	TreeWatch/View/Map/.../BlockInformationContentPage.xaml	23
15	TreeWatch/View/Map/.../BlockInformationContentPage.xaml.cs	24
16	TreeWatch/View/Note/NoteContentPage.xaml	25
17	TreeWatch/View/Note/NoteContentPage.xaml.cs	27
18	TreeWatch/View/ToDo/ToDoContentPage.xaml	28
19	TreeWatch/View/ToDo/ToDoContentPage.xaml.cs	29
20	TreeWatch/View/History/HistoryContentPage.xaml	30
21	TreeWatch/View/History/HistoryContentPage.xaml.cs	31
22	TreeWatch/View/Settings/SettingsContentPage.xaml	32
23	TreeWatch/View/Settings/SettingsContentPage.xaml.cs	34
24	TreeWatch/View/Settings/Map/MapTypeContentPage.xaml	35
25	TreeWatch/View/Settings/Map/MapTypeContentPage.xaml.cs	36

26	TreeWatch/Database/TreeWatchDatabase.cs	37
27	TreeWatch/Database/BlockToDo.cs	39
28	TreeWatch/Model/Map/Field/Block.cs	40
29	TreeWatch/Model/Note.cs	41
30	iOS/Geofence/GeofenceImplementation.cs	42
31	Droid/Geofence/GeofenceImplementation.cs	58
32	Droid/Geofence/GeofenceBroadcastReceiver.cs	71
33	Droid/Geofence/GeofenceLocationListener.cs	72
34	Droid/Geofence/GeofenceTransitionsIntentService.cs	74
35	TreeWatch/Utils/GeoHelper.cs	80
36	iOS/Map/FieldMapRenderer.cs	84
37	iOS/Map/MultiPolygonView.cs	92
38	Droid/Map/FieldMapRenderer.cs	95
39	UITests/Tests.cs	102
40	TreeWatch.Tests/GeoHelperTest.cs	106
41	appveyor.yaml	109
42	TreeWatch/Utils/KMLParser.cs	110
43	documenttemplate.tex	115
44	TreeWatch/ViewModels/Map/MapViewModel.cs	119

1 TreeWatch/View/CustomTabPage.cs

```
22 namespace TreeWatch
23 {
24     using Xamarin.Forms;
25
26     /// <summary>
27     /// A Custom tabbed page inherit TabbedPage
28     /// </summary>
29     public class CustomTabPage : TabbedPage
30     {
31         /// <summary>
32         /// Initializes a new instance of the <see
33         → cref="TreeWatch.CustomButton"/> class.
34         /// </summary>
35         public CustomTabPage()
36         {
37             this.StyleId = "TabPageView";
38
39             // map tab page
40             var mapMasteDetailPage = new MapMasterDetailPage();
41             Children.Add(mapMasteDetailPage);
42
43             // note tab page
44             Children.Add(new NoteContentPage());
45
46             // todo tab page
47             Children.Add(new ToDoContentPage());
48
49             // history tab page
50             Children.Add(new HistoryContentPage());
51
52             // settings tab page
53             Children.Add(new
54             → SettingsContentPage(mapMasteDetailPage.Detail));
55
56         }
57     }
```

2 TreeWatch/View/Map/MapMasterDetailPage.cs

```
22  namespace TreeWatch
23  {
24      using Xamarin.Forms;
25
26      /// <summary>
27      /// Map master detail page.
28      /// </summary>
29      public class MapMasterDetailPage : MasterDetailPage
30      {
31          /// <summary>
32          /// Initializes a new instance of the <see
33           cref="TreeWatch.MapMasterDetailPage"/> class.
34          /// </summary>
35          public MapMasterDetailPage()
36          {
37              // set testing style id
38              this.StyleId = "MapMasterDetailPage";
39
39          // set view model for all pages
40          var mapViewModel = new MapViewModel();
41
42          // Create the master page with the ListView.
43          var mapMenuContentPage = new
44          MapMenuContentPage(mapViewModel);
45          mapMenuContentPage.FieldSelected += (sender, e) =>
46          {
47              IsPresented = false;
48          };
49          this.Master = mapMenuContentPage;
50
50          this.Detail = TargetPlatform.iOS == Device.OS ? new
51          MapNavigationPage(new MapContentPage(mapViewModel)) as Page : new
52          MapContentPage(mapViewModel);
53
53          // configuration of this page
54          this.MasterBehavior = MasterBehavior.Popover;
55          this.Title = this.Detail.Title;
56          if (Detail.Icon != null)
57          {
58              this.Icon = this.Detail.Icon;
59          }
60      }
61  }
```

3 TreeWatch/View/Map/MapNavigationPage.cs

```
22  namespace TreeWatch
23  {
24      using Xamarin.Forms;
25
26      /// <summary>
27      /// Map navigation page.
28      /// </summary>
29      public class MapNavigationPage : NavigationPage
30      {
31          /// <summary>
32          /// Initializes a new instance of the <see
33          ↪ cref="TreeWatch.MapNavigationPage"/> class.
34          /// </summary>
35          /// <param name="root">The content page that should be the root
36          ↪ page.</param>
37          public MapNavigationPage(Page root)
38              : base(root)
39          {
40              // set style id for testing
41              this.StyleId = "MapNavigationPage";
42
43              // set default values
44              this.Title = "Map";
45              if (root.Icon != null)
46              {
47                  this.Icon = root.Icon;
48              }
49      }
```

4 TreeWatch/View/Map/MapContentPage.xaml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage
3      xmlns      ="http://xamarin.com/schemas/2014/forms"
4      xmlns:x    ="http://schemas.microsoft.com/winfx/2009/xaml"
5      x:Class    ="TreeWatch.MapContentPage"
6      xmlns:local="clr-namespace:TreeWatch;assembly=TreeWatch"
7      StyleId    ="MapView">
8      <ContentPage.Icon>
9          <OnPlatform
10             x:TypeArguments="FileImageSource"
11             iOS          ="Icons/MapTabBarIcon.png" />
12      </ContentPage.Icon>
13      <ContentPage.Content>
14          <local:FieldMap
15             x:Name="fieldMap" />
16      </ContentPage.Content>
17  </ContentPage>
```

5 TreeWatch/View/Map/MapContentPage.xaml.cs

```
22 namespace TreeWatch
23 {
24     using Xamarin.Forms;
25     using Xamarin.Forms.Maps;
26
27     /// <summary>
28     /// Map content page.
29     /// </summary>
30     public partial class MapContentPage : ContentPage
31     {
32         /// <summary>
33         /// Initializes a new instance of the <see
34         → cref="TreeWatch.MapContentPage"/> class.
35         /// </summary>
36         /// <param name="mapViewModel">Map view model.</param>
37         public MapContentPage(MapViewModel mapViewModel)
38         {
39             this.InitializeComponent();
40
41             if (TargetPlatform.Android == Device.OS)
42             {
43                 this.Title = "Map";
44             }
45
46             // add view model
47             this.BindingContext = mapViewModel;
48
49             this.SetupMapView();
50
51             this.SetupToolbarItems();
52         }
53
54         /// <summary>
55         /// Gets my location toolbar item.
56         /// </summary>
57         /// <returns>The my location toolbar item.</returns>
58         internal static ToolbarItem GetMyLocationToolbarItem()
59         {
60             var myLocationToolBarItem = new ToolbarItem();
61
62             // Set clicked event
63             myLocationToolBarItem.Clicked += (sender, e) =>
64             → FieldHelper.Instance.CenterUserPostionEvent();
```

```

64          // Design Toolbar Item
65          myLocationToolBarItem.Icon = Device.OS ==
66          ↳ TargetPlatform.iOS ? "Icons/MyLocationIcon.png" :
67          ↳ "MyLocationIcon.png";
68
69          // Set style id for ui testing
70          myLocationToolBarItem.StyleId = "MMyLocationButton";
71
72
73          /// <summary>
74          /// Setups the map content view.
75          /// </summary>
76          internal void SetupMapView()
77          {
78              // Get current position
79              var currentLocation =
80              ↳ MapViewModel.GetCurrentDevicePosition();
81
82              // Jump to the current location inside the map
83              ↳ fieldMap.MoveToRegion(MapSpan.FromCenterAndRadius(currentLocation,
84              ↳ Distance.FromKilometers(1)));
85
86              // Get binding context of this view
87              var viewModel = BindingContext as MapViewModel;
88
89              // Add all fields into the map
90              fieldMap.Fields = viewModel.Fields;
91
92              // Change Map type to hybrid
93              fieldMap.MapType = MapType.Hybrid;
94
95              // set binding context of field map view to the same of
96              ↳ this view
97              fieldMap.BindingContext = this.BindingContext;
98          }
99
100         /// <summary>
101         /// Setups the toolbar items.
102         /// </summary>
103         internal void SetupToolbarItems()
104         {
105             if (TargetPlatform.iOS == Device.OS)
106             {
107                 ToolbarItems.Insert(0, GetMyLocationToolbarItem());

```

```
105 }  
106 }  
107 }  
108 }
```

6 TreeWatch/View/Map/MapMenu/MapMenuContentPage.xaml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage
3      xmlns      ="http://xamarin.com/schemas/2014/forms"
4      xmlns:x     ="http://schemas.microsoft.com/winfx/2009/xaml"
5      x:Class    ="TreeWatch.MapMenuContentPage"
6      xmlns:local="clr-namespace:TreeWatch;assembly=TreeWatch"
7      Title      ="Map Menu"
8      StyleId    ="MapMenuView">
9      <ContentPage.Icon>
10     <OnPlatform
11         x:TypeArguments="FileImageSource"
12             iOS          ="Icons/HamburgerMenuIcon.png"
13             Android      ="HamburgerMenuIcon.png" />
14     </ContentPage.Icon>
15     <ContentPage.Padding>
16         <OnPlatform
17             x:TypeArguments="Thickness"
18             iOS          ="0, 20, 0, 0" />
19     </ContentPage.Padding>
20     <ContentPage.Content>
21         <StackLayout
22             VerticalOptions  ="CenterAndExpand"
23             HorizontalOptions="FillAndExpand"
24             Orientation      ="Vertical"
25             Spacing          ="15">
26             <SearchBar
27                 Placeholder  ="Search"
28                 Text          ="{Binding SearchText}"
29                 SearchCommand=" {Binding SearchCommand}"
30                 StyleId      ="MMSearchBar" />
31             <ListView
32                 x:Name      ="fieldView"
33                 ItemsSource = "{Binding FilteredFields}"
34                 SelectedItem=" {Binding SelectedField}"
35                 StyleId    ="MMLListView">
36                 <ListView.Header>
37                     <StackLayout>
38                         </StackLayout>
39                 </ListView.Header>
40                 <ListView.ItemTemplate>
41                     <DataTemplate>
42                         <ViewCell
43                             StyleId="MMLVField">
```

```

44      <StackLayout
45          Orientation="Horizontal"
46          Padding      ="20,0">
47          <Label
48              Text          ="{Binding Name}"
49              TextColor      ="#000000"
50              VerticalOptions="Center" />
51          <Button
52              Clicked
53                  => "InfoButtonClicked"
54                  CommandParameter ="{Binding Name}"
55                  HorizontalOptions="EndAndExpand"
56                  VerticalOptions  ="Center"
57                  WidthRequest     ="50">
58                  <Button.BackgroundColor>
59                      <OnPlatform
60                          x:TypeArguments="Color"
61                          Android
62                          => "Transparent" />
63                  </Button.BackgroundColor>
64                  <Button.Image>
65                      <OnPlatform
66                          x:TypeArguments="FileImageSource"
67                          iOS
68                          => "Icons/InfoIcon.png"
69                          Android
70                          => "InfoIcon.png" />
71                  </Button.Image>
72          </StackLayout>
73      </ViewCell>
74  </DataTemplate>
75  </ListView.ItemTemplate>
76  </ListView>
77  </StackLayout>
78  </ContentPage.Content>
79 </ContentPage>

```

7 TreeWatch/View/Map/MapMenu/MapMenuContentPage.xaml.cs

```
22 namespace TreeWatch
23 {
24     using System;
25     using Xamarin.Forms;
26
27     /// <summary>
28     /// Map menu content page.
29     /// </summary>
30     public partial class MapMenuContentPage : ContentPage
31     {
32         /// <summary>
33         /// Initializes a new instance of the <see
34         /// cref="TreeWatch.MapMenuContentPage"/> class.
35         /// </summary>
36         /// <param name="mapViewModel">Map view model.</param>
37         public MapMenuContentPage(MapViewModel mapViewModel)
38         {
39             this.InitializeComponent();
40
41             this.BindingContext = mapViewModel;
42
43             fieldView.ItemTapped += this.OnFieldSelected;
44
45         /// <summary>
46         /// Occurs when field selected.
47         /// </summary>
48         public event EventHandler FieldSelected;
49
50         /// <summary>
51         /// Infos the button clicked.
52         /// </summary>
53         /// <param name="sender">The sender of clicked event.</param>
54         /// <param name="e">Event arguments.</param>
55         public void InfoButtonClicked(object sender, EventArgs e)
56         {
57             foreach (Field field in fieldView.ItemsSource)
58             {
59                 if (field.Name.Equals((sender as
60                 Button).CommandParameter))
61                 {
62                     MapViewModel.NavigateToField(field);
63                 }
64             }
65         }
66     }
67 }
```

```
63         }
64     }
65
66     /// <summary>
67     /// Raises the field selected event.
68     /// </summary>
69     /// <param name="sender">The sender of tapped event.</param>
70     /// <param name="e">Item tapped event arguments.</param>
71     protected virtual void OnFieldSelected(object sender,
72     ItemTappedEventArgs e)
73     {
74         if (this.FieldSelected != null)
75         {
76             this.FieldSelected(this, EventArgs.Empty);
77             FieldHelper.Instance.FieldSelectedEvent(e.Item as
78             Field);
79         }
80     }

```

8 TreeWatch/View/Map/DetailedInformation/ VarietiesInformationContentPage.xaml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage
3      xmlns ="http://xamarin.com/schemas/2014/forms"
4      xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
5      x:Class="TreeWatch.VarietiesInformationContentPage"
6      StyleId="VarietyInformationView"
7      Padding="0, 0, 0, 0">
8      <ContentPage.Content>
9          <ListView
10             ItemsSource      ="{Binding VarietyGroups}"
11             SelectedItem     ="{Binding Variety}"
12             StyleId          ="VIVListView"
13             VerticalOptions="FillAndExpand"
14             x:Name           ="varietiesView">
15             <ListView.Header>
16                 <StackLayout>
17                     </StackLayout>
18             </ListView.Header>
19             <ListView.ItemTemplate>
20                 <DataTemplate>
21                     <ViewCell
22                         StyleId="VIVLVVariety">
23                         <StackLayout
24                             Orientation="Horizontal"
25                             Padding      ="30, 0">
26                             <RelativeLayout
27                                 HorizontalOptions="Start"
28                                 VerticalOptions  ="Center">
29                                 <Label
30                                     Text          ="{Binding Variety}"
31                                     TextColor     ="Black"
32                                     VerticalOptions="Center" />
33                                 <Label
34                                     Opacity      ="0.75"
35                                     Text          ="{Binding Variety}"
36                                     TextColor     ="{Binding Color}"
37                                     VerticalOptions="Center" />
38                             </RelativeLayout>
39                             <Label
40                                 HorizontalOptions="EndAndExpand"
41                                 Text          ="{Binding Count}"
42                                 TextColor     ="#8E8F93"
43                                 VerticalOptions="Center" />

```

```
44             <Image
45                 VerticalOptions="Center">
46                 <Image.Source>
47                     <OnPlatform
48                         x:TypeArguments="ImageSource"
49                         iOS
50                         ="Icons/ArrowRightIcon.png"
51                         Android
52                         ="ArrowRightIcon.png" />
53                     </Image.Source>
54                 </Image>
55             </StackLayout>
56         </ViewCell>
57     </DataTemplate>
58 </ListView.ItemTemplate>
59 </ListView>
</ContentPage.Content>
</ContentPage>
```

9 TreeWatch/View/Map/DetailedInformation/ VarietiesInformationContentPage.xaml.cs

```
22 namespace TreeWatch
23 {
24     using Xamarin.Forms;
25
26     /// <summary>
27     /// Varieties information content page.
28     /// </summary>
29     public partial class VarietiesInformationContentPage : ContentPage
30     {
31         /// <summary>
32         /// Initializes a new instance of the <see
33         <code href="TreeWatch.VarietiesInformationContentPage" /> class.
34         /// </summary>
35         /// <param name="informationViewModel">Information view
36         <code>model.</code>
37         public VarietiesInformationContentPage(InformationViewModel
38             informationViewModel)
39         {
40             this.InitializeComponent();
41
42             // set view model
43             this.BindingContext = informationViewModel;
44
45             varietiesView.ItemTapped += (sender, e) =>
46             <code>(this.BindingContext as InformationViewModel).NavigateToBlocks();
47         }
48     }
49 }
```

10 TreeWatch/View/Map/DetailedInformation/FieldInformationContentPage.xaml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage
3      xmlns      ="http://xamarin.com/schemas/2014/forms"
4      xmlns:x    ="http://schemas.microsoft.com/winfx/2009/xaml"
5      x:Class    ="TreeWatch.FieldInformationContentPage"
6      xmlns:local="clr-namespace:TreeWatch;assembly=TreeWatch"
7      Title      ="{Binding Field.Name}"
8      StyleId    ="FieldInformationView"
9      Padding    ="0, 0, 0, 0">
10     <ContentPage.Content>
11         <TableView
12             Intent  ="Menu"
13             StyleId="FIVTableView">
14             <TableRoot>
15                 <TableSection>
16                     <ViewCell
17                         StyleId="FIVTVSize">
18                         <StackLayout
19                             Orientation="Horizontal"
20                             Padding    ="20,0">
21                             <Label
22                                 Text      ="Size"
23                                 TextColor  ="Black"
24                                 VerticalOptions="Center" />
25                             <Label
26                                 HorizontalOptions="EndAndExpand"
27                                 Text      ="{Binding FieldSize}"
28                                 TextColor  ="#8E8F93"
29                                 VerticalOptions="Center" />
30                         </StackLayout>
31                     </ViewCell>
32                 </TableSection>
33                 <TableSection
34                     Title="Variety">
35                     <ViewCell
36                         StyleId="FIVTVVarietyLink"
37                         x:Name  ="varieties">
38                         <StackLayout
39                             Orientation="Horizontal"
40                             Padding    ="20,0">
41                             <Label
42                                 Text      ="Varieties"
43                                 TextColor  ="#000000"
```

```

44             VerticalOptions="Center" />
45             <Label
46                 HorizontalOptions="EndAndExpand"
47                 Text           ="{Binding
48                 ↵   VarietyGroups.Count}"
49
50             <Image
51                 VerticalOptions="Center">
52                 <Image.Source>
53                     <OnPlatform
54                         x:TypeArguments="ImageSource"
55                         iOS
56                         ↵   ="Icons/ArrowRightIcon.png"
57                         Android
58                         ↵   ="ArrowRightIcon.png" />
59                     </Image.Source>
60                 </Image>
61             </StackLayout>
62         </ViewCell>
63     </TableSection>
64 </TableRoot>
65 </TableView>
66 </ContentPage.Content>
67 </ContentPage>

```

11 TreeWatch/View/Map/DetailedInformation/FieldInformationContentPage.xaml.cs

```
22 namespace TreeWatch
23 {
24     using Xamarin.Forms;
25
26     /// <summary>
27     /// Field information content page.
28     /// </summary>
29     public partial class FieldInformationContentPage : ContentPage
30     {
31         /// <summary>
32         /// Initializes a new instance of the <see
33         cref="TreeWatch.FieldInformationContentPage"/> class.
34         /// </summary>
35         /// <param name="informationViewModel">Information view
36         model.</param>
37         public FieldInformationContentPage(InformationViewModel
38         informationViewModel)
39         {
40             this.InitializeComponent();
41
42             // set view model
43             this.BindingContext = informationViewModel;
44
45             NavigationPage.SetBackButtonTitle(this,
46             informationViewModel.Field.Name);
47
48             varieties.Tapped += (sender, e) => (this.BindingContext as
49             InformationViewModel).NavigateToVarieties();
50         }
51     }
52 }
```

12 TreeWatch/View/Map/DetailedInformation/BlocksInformationContentPage.xaml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage
3      xmlns ="http://xamarin.com/schemas/2014/forms"
4      xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
5      x:Class="TreeWatch.BlocksInformationContentPage"
6      StyleId="BlocksInformationView"
7      Padding="0, 0, 0, 0">
8      <ContentPage.Content>
9          <ListView
10             ItemsSource      ="{Binding Variety}"
11             SelectedItem    ="{Binding Block}"
12             StyleId         ="BIVListView"
13             VerticalOptions="FillAndExpand"
14             x:Name          ="blockView">
15             <ListView.Header>
16                 <StackLayout>
17                     </StackLayout>
18             </ListView.Header>
19             <ListView.ItemTemplate>
20                 <DataTemplate>
21                     <ViewCell
22                         StyleId="BIVLVBBlock">
23                         <StackLayout
24                             Orientation="Horizontal"
25                             Padding     ="30, 0">
26                             <RelativeLayout
27                                 HorizontalOptions="Start"
28                                 VerticalOptions  ="Center">
29                                 <Label
30                                     Text          ="{Binding
31                                     TreeType.Name}"
32                                     TextColor     ="Black"
33                                     VerticalOptions="Center" />
34                                 <Label
35                                     Opacity      ="0.75"
36                                     Text          ="{Binding
37                                     TreeType.Name}"
38                                     TextColor     ="{Binding
39                                     TreeType.ColorProp}"
40                                     VerticalOptions="Center" />
41                             </RelativeLayout>
42                             <Image
43                                 HorizontalOptions="EndAndExpand"
```

```
41                     VerticalOptions = "Center">
42             <Image.Source>
43                 <OnPlatform
44                     x:TypeArguments="ImageSource"
45                         iOS
46                         ↳ "Icons/ArrowRightIcon.png"
47                         Android
48                         ↳ "ArrowRightIcon.png" />
49             </Image.Source>
50         </Image>
51     </StackLayout>
52     </ViewCell>
53 </DataTemplate>
54 </ListView.ItemTemplate>
55 </ListView>
</ContentPage.Content>
</ContentPage>
```

13 TreeWatch/View/Map/DetailedInformation/BlocksInformationContentPage.xaml.cs

```
22 namespace TreeWatch
23 {
24     using Xamarin.Forms;
25
26     /// <summary>
27     /// Blocks information content page.
28     /// </summary>
29     public partial class BlocksInformationContentPage : ContentPage
30     {
31         /// <summary>
32         /// Initializes a new instance of the <see
33         /// cref="TreeWatch.BlocksInformationContentPage"/> class.
34         /// </summary>
35         /// <param name="informationViewModel">Information view
36         /// model.</param>
37         public BlocksInformationContentPage(InformationViewModel
38             informationViewModel)
39         {
40             this.InitializeComponent();
41
42             // set view model
43             this.BindingContext = informationViewModel;
44
45             NavigationPage.SetBackButtonTitle(this, "Blocks");
46
47             blockView.ItemTapped += (sender, e) => (this.BindingContext
48                 as InformationViewModel).NavigateToBlock();
49         }
50     }
51 }
```

14 TreeWatch/View/Map/DetailedInformation/BlockInformationContentPage.xaml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage
3      xmlns ="http://xamarin.com/schemas/2014/forms"
4      xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
5      x:Class="TreeWatch.BlockInformationContentPage"
6      Title  ="{Binding Block.TreeType.Name}"
7      StyleId="BlockInformationView"
8      Padding="0, 0, 0, 0">
9      <ContentPage.Content>
10     <TableView
11         Intent ="Menu"
12         StyleId="BIVTableView">
13         <TableRoot>
14             <TableSection>
15                 <ViewCell
16                     StyleId="BIVTVBlockID">
17                     <StackLayout
18                         Orientation="Horizontal"
19                         Padding      ="20,0">
20                         <Label
21                             Text          ="Database-ID"
22                             TextColor      ="Black"
23                             VerticalOptions="Center" />
24                         <Label
25                             HorizontalOptions="EndAndExpand"
26                             Text          ="{Binding Block.ID}"
27                             TextColor      ="#8E8F93"
28                             VerticalOptions="Center" />
29                     </StackLayout>
30                     </ViewCell>
31                 </TableSection>
32             </TableRoot>
33         </TableView>
34     </ContentPage.Content>
35 </ContentPage>
```

15 TreeWatch/View/Map/DetailedInformation/ BlockInformationContentPage.xaml.cs

```
22 namespace TreeWatch
23 {
24     using Xamarin.Forms;
25
26     /// <summary>
27     /// Block information content page.
28     /// </summary>
29     public partial class BlockInformationContentPage : ContentPage
30     {
31         /// <summary>
32         /// Initializes a new instance of the <see
33         cref="TreeWatch.BlockInformationContentPage"/> class.
34         /// </summary>
35         /// <param name="informationViewModel">Information view
36         model.</param>
37         public BlockInformationContentPage(InformationViewModel
38             informationViewModel)
39         {
40             this.InitializeComponent();
41
42             // set view model
43             this.BindingContext = informationViewModel;
44         }
45     }
46 }
```

16 TreeWatch/View/Note/NoteContentPage.xaml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage
3      xmlns      ="http://xamarin.com/schemas/2014/forms"
4      xmlns:x     ="http://schemas.microsoft.com/winfx/2009/xaml"
5      x:Class     ="TreeWatch.NoteContentPage"
6      xmlns:local ="clr-namespace:TreeWatch;assembly=TreeWatch"
7      Title       ="Note"
8      StyleId     ="NoteView"
9      BindingContext="{local:NoteViewModel}">
10     <ContentPage.Padding>
11         <OnPlatform
12             x:TypeArguments="Thickness"
13             iOS           ="0,20,0,0" />
14     </ContentPage.Padding>
15     <ContentPage.Icon>
16         <OnPlatform
17             x:TypeArguments="FileImageSource"
18             iOS           ="Icons/NoteTabBarIcon.png" />
19     </ContentPage.Icon>
20     <ContentPage.Resources>
21         <ResourceDictionary>
22             <local:NoteViewModel
23                 x:Key="noteViewModel" />
24         </ResourceDictionary>
25     </ContentPage.Resources>
26     <ContentPage.Content>
27         <ScrollView>
28             <StackLayout
29                 HorizontalOptions="FillAndExpand"
30                 VerticalOptions  ="FillAndExpand"
31                 Padding        ="20">
32                 <Entry
33                     Placeholder="Title"
34                     Keyboard      ="Default" />
35                 <Image
36                     Source        ="camera.png"
37                     HeightRequest  ="250"
38                     VerticalOptions="StartAndExpand" />
39             <Editor
40                 Text          ="Description"
41                 VerticalOptions="FillAndExpand" />
42         </ScrollView>
43     </ContentPage.Content>
44 </ContentPage>
```

```
42             <StackLayout
43                 HorizontalOptions="CenterAndExpander"
44                 and"
45                 VerticalOptions="End"
46                 Orientation="Horizontal"
47                 Spacing="20">
48                 <Button
49                     Text="Save" />
50                 <Button
51                     Text="Cancel" />
52             </StackLayout>
53         </ScrollView>
54     </ContentPage.Content>
55 </ContentPage>
```

17 TreeWatch/View/Note/NoteContentPage.xaml.cs

```
22 namespace TreeWatch
23 {
24     using Xamarin.Forms;
25
26     /// <summary>
27     /// Note content page.
28     /// </summary>
29     public partial class NoteContentPage : ContentPage
30     {
31         /// <summary>
32         /// Initializes a new instance of the <see
33         ↳ cref="TreeWatch.NoteContentPage"/> class.
34         /// </summary>
35         public NoteContentPage()
36         {
37             this.InitializeComponent();
38         }
39     }
```

18 TreeWatch/View/ToDo/ToDoContentPage.xaml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage
3      xmlns = "http://xamarin.com/schemas/2014/forms"
4      xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
5      x:Class="TreeWatch.ToDoContentPage"
6      Title  = "ToDo"
7      StyleId="TodoView">
8      <ContentPage.Padding>
9          <OnPlatform
10             x:TypeArguments="Thickness"
11             iOS          = "0,20,0,0" />
12         </ContentPage.Padding>
13         <ContentPage.Icon>
14             <OnPlatform
15                 x:TypeArguments="FileImageSource"
16                 iOS          = "Icons/ToDoTabBarIcon.png" />
17             </ContentPage.Icon>
18             <ContentPage.Content>
19                 <ScrollView>
20                     <StackLayout
21                         VerticalOptions  = "CenterAndExpand"
22                         HorizontalOptions= "CenterAndExpand"
23                         Orientation      = "Vertical"
24                         Spacing          = "15">
25                     </StackLayout>
26                 </ScrollView>
27             </ContentPage.Content>
28         </ContentPage>
```

19 TreeWatch/View/ToDo/ToDoContentPage.xaml.cs

```
22  namespace TreeWatch
23  {
24      using Xamarin.Forms;
25
26      /// <summary>
27      /// To do content page inherits ContentPage
28      /// </summary>
29      public partial class ToDoContentPage : ContentPage
30      {
31          /// <summary>
32          /// Initializes a new instance of the <see
33          ↳ cref="TreeWatch ToDoContentPage"/> class.
34          /// </summary>
35          public ToDoContentPage()
36          {
37              this.InitializeComponent();
38          }
39      }
```

20 TreeWatch/View/History/ HistoryContentPage.xaml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage
3      xmlns ="http://xamarin.com/schemas/2014/forms"
4      xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
5      x:Class="TreeWatch.HistoryContentPage"
6      Title  ="History"
7      StyleId="HistoryView">
8      <ContentPage.Padding>
9          <OnPlatform
10             x:TypeArguments="Thickness"
11             iOS          ="0,20,0,0" />
12      </ContentPage.Padding>
13      <ContentPage.Icon>
14          <OnPlatform
15              x:TypeArguments="FileImageSource"
16              iOS          ="Icons/HistoryTabBarIcon.png" />
17      </ContentPage.Icon>
18      <ContentPage.Content>
19          <ScrollView>
20              <StackLayout
21                  VerticalOptions  ="CenterAndExpand"
22                  HorizontalOptions="CenterAndExpand"
23                  Orientation      ="Vertical"
24                  Spacing         ="15">
25                  </StackLayout>
26          </ScrollView>
27      </ContentPage.Content>
28  </ContentPage>
```

21 TreeWatch/View/History/ HistoryContentPage.xaml.cs

```
22 namespace TreeWatch
23 {
24     using Xamarin.Forms;
25
26     /// <summary>
27     /// History content page.
28     /// </summary>
29     public partial class HistoryContentPage : ContentPage
30     {
31         /// <summary>
32         /// Initializes a new instance of the <see
33         ↳ cref="TreeWatch.HistoryContentPage"/> class.
34         /// </summary>
35         public HistoryContentPage()
36         {
37             this.InitializeComponent();
38         }
39     }
```

22 TreeWatch/View/Settings/SettingsContentPage.xaml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage
3      xmlns ="http://xamarin.com/schemas/2014/forms"
4      xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
5      x:Class="TreeWatch.SettingsContentPage"
6      Title  ="Settings"
7      StyleId="SettingsView">
8      <ContentPage.Padding>
9          <OnPlatform
10             x:TypeArguments="Thickness"
11             iOS          ="0,20,0,0" />
12         </ContentPage.Padding>
13         <ContentPage.Icon>
14             <OnPlatform
15                 x:TypeArguments="FileImageSource"
16                 iOS          ="Icons/SettingsTabBarIcon.png" />
17         </ContentPage.Icon>
18         <ContentPage.Content>
19             <TableView
20                 Intent ="Settings"
21                 StyleId="SVTableView">
22                 <TableRoot>
23                     <TableSection
24                         Title="Map">
25                         <ViewCell
26                             StyleId="SVTVMapType"
27                             x:Name  ="mapType">
28                             <StackLayout
29                                 Orientation="Horizontal"
30                                 Padding    ="20,10,20,10">
31                                 <Label
32                                     Text          ="Map Type"
33                                     TextColor     ="#000000"
34                                     VerticalOptions="Center" />
35                                 <Label
36                                     HorizontalOptions="EndAndExpand"
37                                     Text          ="{Binding
38                                     <FieldMap.MapType}">
39                                     TextColor     ="#8E8F93"
40                                     VerticalOptions="Center" />
41                                     <Image
42                                         VerticalOptions="Center">
43                                         <Image.Source>
```

```

43             <OnPlatform
44                 x:TypeArguments="ImageSource"
45                 iOS
46                     => "Icons/ArrowRightIcon.png"
47                     Android
48                     => "ArrowRightIcon.png" />
49             </Image.Source>
50         </Image>
51     </StackLayout>
52 </ViewCell>
53 <SwitchCell
54     On      = "{Binding FieldMap.IsShowingUser}"
55     StyleId="SVTCurrentPosition"
56     Text    = "Current Position" />
57 </TableSection>
58 <TableSection
59     Title="Note">
60         <TextCell
61             Text  = "Not Supported"
62             Detail="Comming Soon" />
63 </TableSection>
64 <TableSection
65     Title="ToDo">
66         <TextCell
67             Text  = "Not Supported"
68             Detail="Comming Soon" />
69 </TableSection>
70 <TableSection
71     Title="History">
72         <TextCell
73             Text  = "Not Supported"
74             Detail="Comming Soon" />
75 </TableSection>
76     </TableRoot>
77 </TableView>
78 </ContentPage.Content>
79 </ContentPage>

```

23 TreeWatch/View/Settings/SettingsContentPage.xaml.cs

```
22 namespace TreeWatch
23 {
24     using Xamarin.Forms;
25
26     /// <summary>
27     /// Settings content page inherits ContentPage
28     /// </summary>
29     public partial class SettingsContentPage : ContentPage
30     {
31         /// <summary>
32         /// Initializes a new instance of the <see
33         <code href="TreeWatch.SettingsContentPage"/> class.
34         /// </summary>
35         /// <param name="mapPage">Map page.</param>
36         public SettingsContentPage(Page mapPage)
37         {
38             this.InitializeComponent();
39
39             // set view model
40             if (Device.OS == TargetPlatform.iOS)
41             {
42                 this.BindingContext = new SettingsViewModel((mapPage as
43                 MapNavigationPage).CurrentPage as MapContentPage);
44             }
45             else if (Device.OS == TargetPlatform.Android)
46             {
47                 this.BindingContext = new SettingsViewModel(mapPage as
48                 MapContentPage);
49             }
50
51             mapType.Tapped += (sender, e) => (BindingContext as
52             SettingsViewModel).NavigateToMapType();
53         }
54     }
55 }
```

24 TreeWatch/View/Settings/Map/MapTypeContentPage.xaml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ContentPage
3      xmlns ="http://xamarin.com/schemas/2014/forms"
4      xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
5      x:Class="TreeWatch.MapTypeContentPage"
6      StyleId="MapTypeView">
7      <ContentPage.Content>
8          <ListView
9              ItemsSource="{Binding MapTypes}"
10             StyleId      ="MTVListView"
11             x:Name       ="mapTypes">
12             <ListView.Header>
13                 <StackLayout>
14                     </StackLayout>
15             </ListView.Header>
16             <ListView.ItemTemplate>
17                 <DataTemplate>
18                     <TextCell
19                         CommandParameter="{Binding Name}"
20                         StyleId          ="MTVLVMapType"
21                         Text              ="{Binding Name} " />
22                 </DataTemplate>
23             </ListView.ItemTemplate>
24         </ListView>
25     </ContentPage.Content>
26 </ContentPage>
```

25 TreeWatch/View/Settings/Map/MapTypeContentPage.xaml.cs

```
22 namespace TreeWatch
23 {
24     using Xamarin.Forms;
25
26     /// <summary>
27     /// Map type content page.
28     /// </summary>
29     public partial class MapTypeContentPage : ContentPage
30     {
31         /// <summary>
32         /// Initializes a new instance of the <see
33         cref="TreeWatch.MapTypeContentPage"/> class.
34         /// </summary>
35         /// <param name="settingsViewModel">Settings view model.</param>
36         public MapTypeContentPage(SettingsViewModel settingsViewModel)
37         {
38             this.InitializeComponent();
39
39             // set view model
40             this.BindingContext = settingsViewModel;
41
42             mapTypes.ItemTapped += (sender, e) => (this.BindingContext
43             as SettingsViewModel).NavigateToSettings(e.Item);
44         }
45     }
46 }
```

26 TreeWatch/Database/TreeWatchDatabase.cs

```
22  namespace TreeWatch
23  {
24      using SQLite.Net;
25
26      using Xamarin.Forms;
27
28      /// <summary>
29      /// TreeWatch database.
30      /// </summary>
31      public class TreeWatchDatabase
32      {
33          /// <summary>
34          /// The connection.
35          /// </summary>
36          public readonly SQLiteConnection Connection;
37
38          /// <summary>
39          /// Initializes a new instance of the <see
→  cref="TreeWatch.TreeWatchDatabase"/> class.
40          /// </summary>
41          public TreeWatchDatabase()
42          {
43              this.Connection =
→ DependencyService.Get<ISQLite>().GetConnection();
44              this.Connection.CreateTable<ToDo>();
45              this.Connection.CreateTable<Block>();
46              this.Connection.CreateTable<Field>();
47              this.Connection.CreateTable<User>();
48              this.Connection.CreateTable<Hours>();
49              this.Connection.CreateTable<Note>();
50              this.Connection.CreateTable<UserToDo>();
51              this.Connection.CreateTable<BlockToDo>();
52              this.Connection.CreateTable<TreeType>();
53              this.Connection.CreateTable<DatabaseConfig>();
54              this.Connection.InsertOrIgnore(new DatabaseConfig());
55          }
56
57          /// <summary>
58          /// Gets or sets the DBconfig.
59          /// </summary>
60          /// <value>The DBconfig.</value>
61          public DatabaseConfig DBconfig
62          {
63              get
```

```
64      {
65          return this.Connection.Get<DatabaseConfig>(1);
66      }
67
68      set
69      {
70          this.Connection.Update(value);
71      }
72  }
73
74  /// <summary>
75  /// Clears the database.
76  /// </summary>
77  public void ClearDataBase()
78  {
79      this.Connection.DeleteAll<ToDo>();
80      this.Connection.DeleteAll<Block>();
81      this.Connection.DeleteAll<Field>();
82      this.Connection.DeleteAll<User>();
83      this.Connection.DeleteAll<Hours>();
84      this.Connection.DeleteAll<Note>();
85      this.Connection.DeleteAll<UserToDo>();
86      this.Connection.DeleteAll<BlockToDo>();
87      this.Connection.DeleteAll<TreeType>();
88  }
89 }
90 }
```

27 TreeWatch/Database/BlockToDo.cs

```
22  namespace TreeWatch
23  {
24      using SQLiteNetExtensions.Attributes;
25
26      /// <summary>
27      /// DatabaseHelperclass for mapping Todos to Blocks
28      /// </summary>
29      public class BlockToDo
30      {
31          /// <summary>
32          /// Gets or sets the block identifier.
33          /// </summary>
34          /// <value>The block identifier.</value>
35          [ForeignKey(typeof(Block))]
36          public int BlockId { get; set; }
37
38          /// <summary>
39          /// Gets or sets todo identifier.
40          /// </summary>
41          /// <value>To do identifier.</value>
42          [ForeignKey(typeof(ToDo))]
43          public int ToDoId { get; set; }
44      }
45  }
```

28 TreeWatch/Model/Map/Field/Block.cs

```
/// <summary>
/// Gets or sets the field ID.
/// </summary>
/// <value>The field identifier.</value>
[ForeignKey(typeof(Field))]
public int FieldId { get; set; }

/// <summary>
/// Gets or sets the List of Todos.
/// </summary>
/// <value>To dos.</value>
[ManyToMany(typeof(BlockToDo))]
public List<ToDo> ToDos { get; set; }

/// <summary>
/// Gets or sets the tree type identifier.
/// Only used for SQLite. Should not be called in any way
→ outside of SQLite methods.
/// </summary>
/// <value>The tree type identifier.</value>
[ForeignKey(typeof(TreeType))]
public int TreeTypeId { get; set; }

/// <summary>
/// Gets or sets the type of the tree.
/// </summary>
/// <value>The type of the tree.</value>
[OneToOne(CascadeOperations = CascadeOperation.CascadeRead |
→ CascadeOperation.CascadeInsert)]
public TreeType TreeType { get; set; }
```

29 TreeWatch/Model/Note.cs

```
/// <summary>
/// Gets or sets the position.
/// </summary>
/// <value>The position.</value>
[TextBlob("PositionBlob")]
public Position Position { get; set; }

/// <summary>
/// Gets or sets the position BLOB.
/// Only used for SQLite. Should not be called in any way
→ outside of SQLite methods.
/// </summary>
/// <value>The position BLOB.</value>
public string PositionBlob { get; set; }
```

30 iOS/Geofence/GeofenceImplementation.cs

```
184     /// <summary>
185     /// Checks if has passed to stayed state
186     /// </summary>
187     /// <returns>The if stayed.</returns>
188     /// <param name="regionId">Region identifier.</param>
189     public static async Task CheckIfStayed(string regionId)
190     {
191         if
192             (CrossGeofence.Current.GeofenceResults.ContainsKey(regionId) &&
193             CrossGeofence.Current.Regions.ContainsKey(regionId) &&
194             CrossGeofence.Current.Regions[regionId].NotifyOnStay &&
195             CrossGeofence.Current.GeofenceResults[regionId].Transition ==
196             GeofenceTransition.Entered && Math.Abs(CrossGeofence.Current.Region_
197             s[regionId].StayedInThresholdDuration.TotalMilliseconds) >
198             float.Epsilon)
199             {
200                 await Task.Delay((int)CrossGeofence.Current.Regions[reg_
201             ionId].StayedInThresholdDuration.TotalMilliseconds);
202
203                 if
204                     (CrossGeofence.Current.GeofenceResults[regionId].LastExitTime ==
205                      null && CrossGeofence.Current.GeofenceResults[regionId].Transition
206                      != GeofenceTransition.Stayed)
207                     {
208
209                         CrossGeofence.Current.GeofenceResults[regionId].Transition =
210                         GeofenceTransition.Stayed;
211
212                         CrossGeofence.GeofenceListener.OnRegionStateChanged_
213                         (CrossGeofence.Current.GeofenceResults[regionId]);
214
215                         if
216                             (CrossGeofence.Current.Regions[regionId].ShowNotification)
217                             {
218
219                                 GeofenceImplementation.CreateNotification(ViewAction,
220                                 string.IsNullOrEmpty(CrossGeofence.Current.Regions[regionId].Notifi_
221                                 cationStayMessage) ?
222                                 CrossGeofence.Current.GeofenceResults[regionId].ToString() :
223                                 CrossGeofence.Current.Regions[regionId].NotificationStayMessage);
224
225                             }
226                         }
227                     }
228                 }
229             }
230         }
```

```
208     /// <summary>
209     /// Availables for monitoring.
210     /// </summary>
211     /// <returns><c>true</c>, if for monitoring was availabled,
212     <c>false</c> otherwise.</returns>
213     public bool AvailableForMonitoring()
214     {
215         bool retVal = false;
216         this.RequestAlwaysAuthorization();
217
218         if (!CLLocationManager.LocationServicesEnabled)
219         {
220             string message = string.Format("{0} - {1}",
221             CrossGeofence.Id, "You need to enable Location Services");
222             System.Diagnostics.Debug.WriteLine(message);
223             CrossGeofence.GeofenceListener.OnError(message);
224         }
225         else if (CLLocationManager.Status ==
226             CLAuthorizationStatus.Denied || CLLocationManager.Status ==
227             CLAuthorizationStatus.Restricted)
228         {
229             string message = string.Format("{0} - {1}",
230             CrossGeofence.Id, "You need to authorize Location Services");
231             System.Diagnostics.Debug.WriteLine(message);
232             CrossGeofence.GeofenceListener.OnError(message);
233         }
234         else if
235             (CLLocationManager.IsMonitoringAvailable(typeof(CLRegion)))
236         {
237             var settings =
238                 UIUserNotificationSettings.GetSettingsForTypes(
239                     UIUserNotificationType.Alert
240                     | UIUserNotificationType.Badge
241                     | UIUserNotificationType.Sound,
242                     new NSSet());
243                 UIApplication.SharedApplication.RegisterUserNotificatio
244             nSettings(settings);
245
246             retVal = true;
247         }
248         else
249         {
250             string message = string.Format("{0} - {1}",
251             CrossGeofence.Id, "Not available for monitoring");
252             System.Diagnostics.Debug.WriteLine(message);
253             CrossGeofence.GeofenceListener.OnError(message);
254         }
255     }
256 }
```

```

246     }
247
248     return retVal;
249 }
250
251     /// <summary>
252     /// Starts monitoring region
253     /// </summary>
254     /// <param name="region">Region to monitor</param>
255     public void StartMonitoring(GeofenceCircularRegion region)
256     {
257         if (this.AvailableForMonitoring())
258         {
259             if (!this.regions.ContainsKey(region.Id))
260             {
261                 this.regions.Add(region.Id, region);
262             }
263             else
264             {
265                 this.regions[region.Id] = region;
266             }
267
268             GeofenceStore.SharedInstance.Save(region);
269
270             if (this.Regions.Count > 20 &&
271             this.locationManager.MonitoredRegions.Count == 20)
272             {
273                 this.RecalculateRegions();
274             }
275             else
276             {
277                 this.AddRegion(region);
278             }
279
280             this.locationManager.StartMonitoringSignificantLocationChanges();
281         }
282
283     /// <summary>
284     /// Start monitoring regions
285     /// </summary>
286     /// <param name="regions"> Regions to monitor</param>
287     public void StartMonitoring(IList<GeofenceCircularRegion>
288     regions)
289     {
290         if (this.AvailableForMonitoring())

```

```

290     {
291         foreach (var region in regions)
292         {
293             if (!this.regions.ContainsKey(region.Id))
294             {
295                 this.regions.Add(region.Id, region);
296             }
297             else
298             {
299                 this.regions[region.Id] = region;
300             }
301
302             GeofenceStore.SharedInstance.Save(region);
303         }
304
305         if (this.Regions.Count > 20 &&
306             ↳ this.locationManager.MonitoredRegions.Count == 20)
307         {
308             this.RecalculateRegions();
309         }
310         else
311         {
312             foreach (var region in regions)
313             {
314                 this.AddRegion(region);
315             }
316
317             ↳ this.locationManager.StartMonitoringSignificantLocationChanges();
318         }
319     }
320
321     /// <summary>
322     /// Stops monitoring all regions
323     /// </summary>
324     public void StopMonitoringAllRegions()
325     {
326         if (this.AvailableForMonitoring())
327         {
328             GeofenceStore.SharedInstance.RemoveAll();
329
330             foreach (CLCircularRegion region in
331                 ↳ this.locationManager.MonitoredRegions)
332             {
333                 this.locationManager.StopMonitoring(region);
334             }
335         }
336     }

```

```

334
335
    ↵ this.locationManager.StopMonitoringSignificantLocationChanges();
    ↵ this.regions.Clear();
    ↵ this.geofenceResults.Clear();
    ↵ CrossGeofence.GeofenceListener.OnMonitoringStopped();
339
    ↵ }
340
}
341
    /// <summary>
343    /// Stops monitoring region
344    /// </summary>
345    /// <param name="regionIdentifier">Region to stop
    ↵ monitoring</param>
346    public void StopMonitoring(string regionIdentifier)
347    {
348        if
    ↵ (CLLocationManager.IsMonitoringAvailable(typeof(CLRegion)))
349        {
350            this.RemoveRegionMonitoring(regionIdentifier);
351
            CrossGeofence.GeofenceListener.OnMonitoringStopped(regi
    ↵ onIdentifier);
353
            if (this.regions.Count == 0)
355        {
356
            CrossGeofence.GeofenceListener.OnMonitoringStopped();
    ↵ }
358
        }
359
    }
360
    /// <summary>
362    /// Stop monitoring regions
363    /// </summary>
364    /// <param name="regionIdentifiers">Region to stop
    ↵ monitoring</param>
365    public void StopMonitoring(IList<string> regionIdentifiers)
366    {
367        if (this.AvailableForMonitoring())
368        {
369            foreach (string regionIdentifier in regionIdentifiers)
370            {
371                this.StopMonitoring(regionIdentifier);
372            }
373        }
374    }

```

```

375
376     /// <summary>
377     /// Get current 20 monitored regions.
378     /// </summary>
379     /// <param name="regions">List of all regions</param>
380     /// <returns>current 20 monitored regions</returns>
381     public IList<GeofenceCircularRegion>
382         GetCurrentRegions(IList<GeofenceCircularRegion> regions)
383     {
384         IList<GeofenceCircularRegion> nearestRegions;
385
386         if (regions.Count > 20)
387         {
388             if (this.locationManager.Location != null)
389             {
390                 IEnumerable<GeofenceCircularRegion> newRegions =
391                     regions.OrderBy(r => GeofenceImplementation.CalculateDistance(this.]
392                     locationManager.Location.Coordinate.Latitude,
393                     this.locationManager.Location.Coordinate.Longitude, r.Latitude,
394                     r.Longitude)).Take(20);
395
396                 nearestRegions = newRegions.ToList();
397             }
398             else
399             {
400                 nearestRegions = regions.Take(20).ToList();
401             }
402         }
403
404         return nearestRegions;
405     }
406
407     /// <summary>
408     /// Calculates distance between two locations
409     /// </summary>
410     /// <param name="lat1">Latitude first point</param>
411     /// <param name="lon1">Longitude first point</param>
412     /// <param name="lat2">Latitude second point</param>
413     /// <param name="lon2">Longitude second point</param>
414     /// <returns>The Distance</returns>
415     private static double CalculateDistance(double lat1, double
416         lon1, double lat2, double lon2)
417     {

```

```

416         const double R = 6372.8; // In kilometers
417         double lat = Math.PI * (lat2 - lat1) / 180.0;
418         double lon = Math.PI * (lon2 - lon1) / 180.0;
419         lat1 = Math.PI * lat1 / 180.0;
420         lat2 = Math.PI * lat2 / 180.0;
421
422         double a = (Math.Sin(lat / 2) * Math.Sin(lat / 2)) +
423             (Math.Sin(lon / 2) * Math.Sin(lon / 2) * Math.Cos(lat1) *
424             Math.Cos(lat2));
425
426         return (R * 2 * Math.Asin(Math.Sqrt(a))) * 1000; // meters
427     }
428
429     /// <summary>
430     /// Dids the start monitoring for region.
431     /// </summary>
432     /// <param name="sender">Sender wo fired the event.</param>
433     /// <param name="e">Event args.</param>
434     private static void DidStartMonitoringForRegion(object sender,
435         CLRegionEventArgs e)
436     {
437         CrossGeofence.GeofenceListener.OnMonitoringStarted(e.Region,
438             e.Identifier);
439
440         /// <summary>
441         /// Creates the notification.
442         /// </summary>
443         /// <param name="title">Title of the notification.</param>
444         /// <param name="message">Message of the notification.</param>
445         private static void CreateNotification(string title, string
446             message)
447         {
448             var notification = new UILocalNotification();
449
450             notification.AlertAction = title;
451             notification.AlertBody = message;
452             notification.HasAction = true;
453
454             notification.SoundName =
455                 UILocalNotification.DefaultSoundName;
456             #if __UNIFIED__
457                 UIApplication.SharedApplication.PresentLocalNotificationNow(
458                     notification);
459             #else
460                 UIApplication.SharedApplication.PresentLocationNotification(
461                     Now(notification));

```

```

455         #endif
456     }
457
458     /// <summary>
459     /// Sets the last known location.
460     /// </summary>
461     /// <param name="location">Location to set.</param>
462     private void SetLastKnownLocation(CLLocation location)
463     {
464         if (location != null)
465         {
466             if (this.lastKnownGeofenceLocation == null)
467             {
468                 this.lastKnownGeofenceLocation = new
469                 GeofenceLocation();
470             }
471
472             this.lastKnownGeofenceLocation.Latitude =
473                 location.Coordinate.Latitude;
474             this.lastKnownGeofenceLocation.Longitude =
475                 location.Coordinate.Longitude;
476             this.lastKnownGeofenceLocation.Accuracy =
477                 location.HorizontalAccuracy;
478             DateTime referenceDate =
479                 TimeZone.CurrentTimeZone.ToLocalTime(new DateTime(2001, 1, 1, 0, 0,
480                 0));
481             referenceDate.AddSeconds(location.Timestamp.SecondsSince
482                 eReferenceDate);
483
484             this.lastKnownGeofenceLocation.Date = referenceDate;
485         }
486     }
487
488     /// <summary>
489     /// Locationses the updated.
490     /// </summary>
491     /// <param name="sender">Sender who fired the event.</param>
492     /// <param name="e">Event arguemnts</param>
493     private void LocationsUpdated(object sender,
494         CLLocationsUpdatedEventArgs e)
495     {
496         CLLocation lastLocation = e.Locations[e.Locations.Length -
497         1];
498
499         this.SetLastKnownLocation(lastLocation);
500     }

```

```

492         if (this.Regions.Count > 20 &&
493             this.locationManager.MonitoredRegions.Count == 20)
494             {
495                 this.RecalculateRegions();
496             }
497             {
498                 // Check any current monitored regions not in loaded
499                 persistent regions and stop monitoring them
500                 foreach (CLCircularRegion region in
501                     this.locationManager.MonitoredRegions)
502                     {
503                         if (!this.Regions.ContainsKey(region.Identifier))
504                         {
505                             this.locationManager.StopMonitoring(region);
506                             string message = string.Format("{0} - Stopped
507                             monitoring region {1} wasn't in persistent loaded regions",
508                             CrossGeofence.Id, region.Identifier);
509                             System.Diagnostics.Debug.WriteLine(message);
510                         }
511                     }
512
513             System.Diagnostics.Debug.WriteLine(string.Format("{0} -
514             {1}: {2}, {3}", CrossGeofence.Id, "Location update",
515             lastLocation.Coordinate.Latitude,
516             lastLocation.Coordinate.Longitude));
517         }
518
519         /// <summary>
520         /// Gets called after State is determined.
521         /// </summary>
522         /// <param name="sender">Sender who fired the event.</param>
523         /// <param name="e">Event arguements</param>>
524         private void DidDetermineState(object sender,
525             CLRegionStateDeterminedEventArgs e)
526             {
527                 switch (e.State)
528                 {
529                     case CLRegionState.Inside:
530                         System.Diagnostics.Debug.WriteLine(string.Format("{0} - {1}",
531                         CrossGeofence.Id, "StartedRegion: " + e.Region));
532                         this.OnRegionEntered(e.Region);
533                         break;
534                     case CLRegionState.Outside:
535                         break;
536                 }
537             }

```

```

528         default:
529             string message = string.Format("{0} - {1}" ,
530             CrossGeofence.Id, "Unknown region state");
531             System.Diagnostics.Debug.WriteLine(message);
532             break;
533         }
534     }
535
536     /// <summary>
537     /// Get Called on Failure.
538     /// </summary>
539     /// <param name="sender">Sender who fired the event.</param>
540     /// <param name="e">Event arguemnts</param>
541     private void OnFailure(object sender, NSErrorEventArgs e)
542     {
543         if (this.IsMonitoring)
544         {
545             this.StopMonitoringAllRegions();
546         }
547
548         CrossGeofence.GeofenceListener.OnError(e.Error.LocalizedDes
549         cription);
550     }
551
552     /// <summary>
553     /// Get Called on RegionEntered.
554     /// </summary>
555     /// <param name="sender">Sender who fired the event.</param>
556     /// <param name="e">Event arguemnts</param>
557     private void RegionEntered(object sender, CLRegionEventArgs e)
558     {
559         this.OnRegionEntered(e.Region);
560     }
561
562     /// <summary>
563     /// Get Called on RegionEntered.
564     /// </summary>
565     /// <param name="region">Region entered.</param>
566     private async void OnRegionEntered(CLRegion region)
567     {
568         if (this.GeofenceResults.ContainsKey(region.Identifier) &&
569             this.GeofenceResults[region.Identifier].Transition ==
570             GeofenceTransition.Entered)
571         {
572             return;
573         }
574     }

```

```

571         if (!this.geofenceResults.ContainsKey(region.Identifier))
572     {
573         this.geofenceResults.Add(
574             region.Identifier,
575             new GeofenceResult
576             {
577                 RegionId = region.Identifier
578             });
579     }
580
581     if (this.LastKnownLocation != null)
582     {
583         this.geofenceResults[region.Identifier].Latitude =
584             this.LastKnownLocation.Latitude;
585         this.geofenceResults[region.Identifier].Longitude =
586             this.LastKnownLocation.Longitude;
587         this.geofenceResults[region.Identifier].Accuracy =
588             this.LastKnownLocation.Accuracy;
589     }
590     else
591     {
592         this.geofenceResults[region.Identifier].Latitude =
593             region.Center.Latitude;
594         this.geofenceResults[region.Identifier].Longitude =
595             region.Center.Longitude;
596         this.geofenceResults[region.Identifier].Accuracy =
597             region.Radius;
598     }
599
600     this.geofenceResults[region.Identifier].LastEnterTime =
601         DateTime.Now;
602     this.geofenceResults[region.Identifier].LastExitTime = null;
603     this.geofenceResults[region.Identifier].Transition =
604         GeofenceTransition.Entered;
605
606     if (region.NotifyOnEntry)
607     {
608         CrossGeofence.GeofenceListener.OnRegionStateChanged(this,
609             s.geofenceResults[region.Identifier]);
610
611         if (this.Regions.ContainsKey(region.Identifier) &&
612             this.Regions[region.Identifier].ShowNotification)
613         {

```

```

604
    ↵ GeofenceImplementation.CreateNotification(ViewAction, string.IsNullOrEmpty |
    ↵ OrEmpty(this.Regions[region.Identifier].NotificationEntryMessage) ?
    ↵ this.GeofenceResults[region.Identifier].ToString() :
    ↵ this.Regions[region.Identifier].NotificationEntryMessage);
    ↵ }
606    }
607
608    // Checks if device has stayed asynchronously
609    await
    ↵ GeofenceImplementation.CheckIfStayed(region.Identifier);
610    }
611
612    /// <summary>
613    /// Get called when Region is left
614    /// </summary>
615    /// <param name="sender">Sender who fired the event.</param>
616    /// <param name="e">Event arguemnts</param>
617    private void RegionLeft(object sender, CLRegionEventArgs e)
618    {
619        if (!this.GeofenceResults.ContainsKey(e.Region.Identifier))
    ↵ || this.GeofenceResults[e.Region.Identifier].Transition != 
    ↵ GeofenceTransition.Exited)
620        {
621            this.OnRegionLeft(e.Region);
622        }
623    }
624
625    /// <summary>
626    /// Get called when Region is left
627    /// </summary>
628    /// <param name="region">Region thats left.</param>
629    private void OnRegionLeft(CLRegion region)
630    {
631        if (!this.geofenceResults.ContainsKey(region.Identifier))
632        {
633            this.geofenceResults.Add(
634                region.Identifier,
635                new GeofenceResult
636                {
637                    RegionId = region.Identifier
638                });
639        }
640
641        if (this.LastKnownLocation != null)
642        {

```

```
643             this.geofenceResults[region.Identifier].Latitude =
644             ↵ this.LastKnownLocation.Latitude;
645             this.geofenceResults[region.Identifier].Longitude =
646             ↵ this.LastKnownLocation.Longitude;
647             this.geofenceResults[region.Identifier].Accuracy =
648             ↵ this.LastKnownLocation.Accuracy;
649         }
650         else
651         {
652             this.geofenceResults[region.Identifier].Latitude =
653             ↵ region.Center.Latitude;
654             this.geofenceResults[region.Identifier].Longitude =
655             ↵ region.Center.Longitude;
656             this.geofenceResults[region.Identifier].Accuracy =
657             ↵ region.Radius;
658         }
659
660         this.geofenceResults[region.Identifier].LastExitTime =
661             ↵ DateTime.Now;
662         this.geofenceResults[region.Identifier].Transition =
663             ↵ GeofenceTransition.Exited;
664
665         CrossGeofence.GeofenceListener.OnRegionStateChanged(this.geofenceResults[region.Identifier]);
666
667         if (this.Regions[region.Identifier].ShowNotification)
668         {
669             GeofenceImplementation.CreateNotification(ViewAction,
670             ↵ string.IsNullOrEmpty(this.Regions[region.Identifier].NotificationExitMessage) ? this.GeofenceResults[region.Identifier].ToString() :
671             ↵ this.Regions[region.Identifier].NotificationExitMessage);
672         }
673
674         /// <summary>
675         /// Recalculates the regions.
676         /// </summary>
677         private void RecalculateRegions()
678         {
679             IList<GeofenceCircularRegion> tmpRegions =
680             ↵ this.Regions.Values.ToList();
681
682             // Stop all monitored regions
683             foreach (CLCircularRegion region in
684             ↵ this.locationManager.MonitoredRegions)
685             {
686                 this.locationManager.StopMonitoring(region);
687             }
688         }
689     }
690 }
```

```

676     }
677
678     IList<GeofenceCircularRegion> nearestRegions =
679     ↵ this.GetCurrentRegions(tmpRegions);
680
681     foreach (GeofenceCircularRegion region in nearestRegions)
682     {
683         this.AddRegion(region);
684     }
685
686     string message = string.Format("{0} - {1}",
687     ↵ CrossGeofence.Id, "Restarted monitoring to nearest 20 regions");
688     System.Diagnostics.Debug.WriteLine(message);
689 }
690
691     /// <summary>
692     /// Adds the region.
693     /// </summary>
694     /// <param name="region">Region to add.</param>
695     private void AddRegion(GeofenceCircularRegion region)
696     {
697         CLRegion addingRegion;
698
699         addingRegion = UIDevice.CurrentDevice.CheckSystemVersion(7,
700         ↵ 0)
701         ? new CLCircularRegion(new
702             CLLocationCoordinate2D(region.Latitude, region.Longitude),
703             (region.Radius >
704             this.locationManager.MaximumRegionMonitoringDistance) ?
705             this.locationManager.MaximumRegionMonitoringDistance :
706             region.Radius, region.Id)
707             : new CLRegion(new
708                 CLLocationCoordinate2D(region.Latitude, region.Longitude),
709                 (region.Radius >
710                 this.locationManager.MaximumRegionMonitoringDistance) ?
711                 this.locationManager.MaximumRegionMonitoringDistance :
712                 region.Radius, region.Id);
713
714         addingRegion.NotifyOnEntry = region.NotifyOnEntry ||
715         region.NotifyOnStay;
716         addingRegion.NotifyOnExit = region.NotifyOnExit;
717         this.locationManager.StartMonitoring(addingRegion);
718         this.locationManager.RequestState(addingRegion);
719     }
720
721     /// <summary>
722     /// Removes the region from monitoring.

```

```

709     ///</summary>
710     ///<param name="regionIdentifier">Region identifier.</param>
711     private void RemoveRegionMonitoring(string regionIdentifier)
712     {
713         if (this.regions.ContainsKey(regionIdentifier))
714         {
715             this.regions.Remove(regionIdentifier);
716         }
717
718         if (this.geofenceResults.ContainsKey(regionIdentifier))
719         {
720             this.geofenceResults.Remove(regionIdentifier);
721         }
722
723         GeofenceStore.SharedInstance.Remove(regionIdentifier);
724
725         var region = this.GetRegion(regionIdentifier);
726         if (region != null)
727         {
728             this.locationManager.StopMonitoring(region);
729         }
730     }
731
732     ///<summary>
733     /// Gets the region.
734     ///</summary>
735     ///<returns>The region.</returns>
736     ///<param name="identifier">Identifier of the region to
737     ← get.</param>
738     private CLRegion GetRegion(string identifier)
739     {
740         CLRegion region = null;
741         foreach (CLRegion r in
742             this.locationManager.MonitoredRegions)
743         {
744             if (r.Identifier.Equals(identifier,
745             StringComparison.OrdinalIgnoreCase))
746             {
747                 region = r;
748                 break;
749             }
750         }
751
752     }
753
754     ///<summary>

```

```

753     /// Requests the always authorization.
754     /// </summary>
755     private void RequestAlwaysAuthorization()
756     {
757         CLAuthorizationStatus status = CLLocationManager.Status;
758         if (status == CLAuthorizationStatus.AuthorizedWhenInUse ||
759             status == CLAuthorizationStatus.Denied)
760         {
761             string title = (status == CLAuthorizationStatus.Denied)
762             ? "Location services are off" : "Background location is not
763             enabled";
764             const string Message = "To use background location you
765             must turn on 'Always' in the Location Services Settings";
766
767             var alertView = new UIAlertView(title, Message, null,
768             "Cancel", "Settings");
769
770             alertView.Clicked += (sender, buttonArgs) =>
771             {
772                 if (buttonArgs.ButtonIndex == 1)
773                 {
774                     // Send the user to the Settings for this app
775                     var settingsUrl = new
776                     NSUrl(UIApplication.OpenSettingsUrlString);
777
778                     UIApplication.SharedApplication.OpenUrl(settingsUrl);
779                 }
780             };
781
782         }
783     }

```

31 Droid/Geofence/GeofenceImplementation.cs

```
184     /// <summary>
185     /// Gets the geofence transition pending intent.
186     /// </summary>
187     /// <value>The geofence transition pending intent.</value>
188     private PendingIntent GeofenceTransitionPendingIntent
189     {
190         get
191         {
192             // If the PendingIntent already exists
193             if (this.modeGeofencePendingIntent == null)
194             {
195                 //var intent = new Intent(Application.Context,
196                 //typeof(GeofenceBroadcastReceiver));
197                 //intent.SetAction(string.Format("{0}.{1}",
198                 //Application.Context.PackageName, GeoReceiverAction));
199                 var intent = new Intent(string.Format("{0}.{1}",
200                 Application.Context.PackageName, GeoReceiverAction));
201                 Console.Error.WriteLine("{0}.{1}",
202                 Application.Context.PackageName, GeoReceiverAction);
203                 //var intent = new Intent(Application.Context,
204                 //typeof(GeofenceTransitionsIntentService));
205                 this.modeGeofencePendingIntent =
206                 PendingIntent.GetBroadcast(Application.Context, 0, intent,
207                 PendingIntentFlags.UpdateCurrent);
208             }
209
210             return this.modeGeofencePendingIntent;
211         }
212     }
213
214     /// <summary>
215     /// Starts the monitoring.
216     /// </summary>
217     /// <param name="region">Region that should be
218     /// monitored.</param>
219     public void StartMonitoring(GeofenceCircularRegion region)
220     {
221         /*if (IsMonitoring && mGoogleApiClient.IsConnected)
222         {
223             Android.Gms.Location.LocationServices.GeofencingApi.Remove
224             eGeofences(mGoogleApiClient,
225             GeofenceTransitionPendingIntent).SetResultCallback(this);
226         }*/
227     }
```

```

218         if (!this.modeRegions.ContainsKey(region.Id))
219         {
220             this.modeRegions.Add(region.Id, region);
221         }
222
223         this.RequestMonitoringStart();
224     }
225
226     /// <summary>
227     /// Starts the monitoring.
228     /// </summary>
229     /// <param name="regions">Regions list.</param>
230     public void StartMonitoring(IList<GeofenceCircularRegion>
231     regions)
232     {
233         /* if (IsMonitoring && mGoogleApiClient.IsConnected)
234         {
235             Android.Gms.Location.LocationServices.GeofencingApi.Remove
236             eGeofences(mGoogleApiClient,
237             GeofenceTransitionPendingIntent).SetResultCallback(this);
238         }*/
239
240         foreach (var region in regions)
241         {
242             if (!this.modeRegions.ContainsKey(region.Id))
243             {
244                 this.modeRegions.Add(region.Id, region);
245             }
246         }
247
248     /// <summary>
249     /// Stops the monitoring.
250     /// </summary>
251     /// <param name="regionIdentifier">Region identifier.</param>
252     public void StopMonitoring(string regionIdentifier)
253     {
254         this.StopMonitoring(new List<string> { regionIdentifier });
255     }
256
257     /// <summary>
258     /// Stops the monitoring.
259     /// </summary>
260     /// <param name="regionIdentifiers">Region identifiers.</param>
261     public void StopMonitoring(IList<string> regionIdentifiers)

```

```

262     {
263         this.modeRequestedRegionIdentifiers = regionIdentifiers;
264
265         if (this.IsMonitoring &&
266             this.modeGoogleApiClient.IsConnected)
267         {
268             this.RemoveGeofences(regionIdentifiers);
269         }
270         else
271         {
272             // If not connection then connect
273             if (!this.modeGoogleApiClient.IsConnecting)
274             {
275                 this.modeGoogleApiClient.Connect();
276             }
277
278             // Request to add geofence regions once connected
279             this.CurrentRequestType = RequestType.Delete;
280         }
281
282     /// <summary>
283     /// Stops monitoring all geofence regions
284     /// </summary>
285     public void StopMonitoringAllRegions()
286     {
287         if (this.IsMonitoring &&
288             this.modeGoogleApiClient.IsConnected)
289         {
290             this.RemoveGeofences();
291         }
292         else
293         {
294             // If not connection then connect
295             if (!this.modeGoogleApiClient.IsConnecting)
296             {
297                 this.modeGoogleApiClient.Connect();
298             }
299
300             // Request to add geofence regions once connected
301             this.CurrentRequestType = RequestType.Clear;
302         }
303
304     /// <summary>
305     /// Raises the connection failed event.
306     /// </summary>

```

```

307     /// <param name="result">Connection result.</param>
308     public void OnConnectionFailed(ConnectionString result)
309     {
310         int errorCode = result.ErrorCode;
311         string message = string.Format("{0} - {1} {2}",
312             CrossGeofence.Id, "Connection to Google Play services failed with
313             error code ", errorCode);
314         System.Diagnostics.Debug.WriteLine(message);
315         CrossGeofence.GeofenceListener.OnError(message);
316     }
317
318     /// <summary>
319     /// Raises the connected event.
320     /// </summary>
321     /// <param name="connectionHint">Connection hint.</param>
322     public void OnConnected(Bundle connectionHint)
323     {
324         Android.Locations.Location location =
325             Android.Gms.Location.LocationServices.FusedLocationApi.GetLastLocat
326             ion(this.modeGoogleApiClient);
327         this.SetLastKnownLocation(location);
328
329         switch (this.CurrentRequestType)
330         {
331             case RequestType.Add:
332                 this.AddGeofences();
333                 this.StartLocationUpdates();
334                 break;
335             case RequestType.Clear:
336                 this.RemoveGeofences();
337                 break;
338             case RequestType.Delete:
339                 if (this.modeRequestedRegionIdentifiers != null)
340                 {
341                     this.RemoveGeofences(this.modeRequestedRegionIdentifiers);
342                 }
343                 break;
344             default:
345                 this.CurrentRequestType = RequestType.Default;
346             }
347
348             /// <summary>
349             /// Raises the result event.
350             /// </summary>

```

```

349     ///<param name="result">Result object.</param>
350     public void OnResult(Java.Lang.Object result)
351     {
352         var res = result.JavaCast<IResult>();
353
354         int statusCode = res.Status.StatusCode;
355         //int statusCode = 0;
356         string message = string.Empty;
357
358         switch (statusCode)
359         {
360             case CommonStatusCodes.SuccessCache:
361             case CommonStatusCodes.Success:
362                 if (this.CurrentRequestType == RequestType.Add)
363                 {
364                     message = string.Format("{0} - {1}",
365                     CrossGeofence.Id, "Successfully added Geofence.");
366
367                     foreach (GeofenceCircularRegion region in
368                     this.Regions.Values)
369                     {
370
371                     CrossGeofence.GeofenceListener.OnMonitoringStarted(region.Id);
372
373                     }
374
375                     break;
376             case CommonStatusCodes.Error:
377                 message = string.Format("{0} - {1}",
378                     CrossGeofence.Id, "Error adding Geofence.");
379                 break;
380             case
381                 Android.Gms.Location.GeofenceStatusCodes.GeofenceTooManyGeofences:
382                 message = string.Format("{0} - {1}",
383                     CrossGeofence.Id, "Too many geofences.");
384                 break;
385             case Android.Gms.Location.GeofenceStatusCodes.GeofenceT
386             ooManyPendingIntents:
387                 message = string.Format("{0} - {1}",
388                     CrossGeofence.Id, "Too many pending intents.");
389                 break;

```

```

386             case
387     ↳  Android.Gms.Location.GeofenceStatusCodes.GeofenceNotAvailable:
388         message = string.Format("{0} - {1}",
389         ↳  CrossGeofence.Id, "Geofence not available.");
390         break;
391     }
392
393     System.Diagnostics.Debug.WriteLine(message);
394
395     if (statusCode != CommonStatusCodes.Success && statusCode
396     ↳  != CommonStatusCodes.SuccessCache && this.IsMonitoring)
397     {
398         this.StopMonitoringAllRegions();
399
400         if (!string.IsNullOrEmpty(message))
401         {
402             CrossGeofence.GeofenceListener.OnError(message);
403         }
404     }
405
406
407     /// <summary>
408     /// Raises the connection suspended event.
409     /// </summary>
410     /// <param name="cause">Cause int.</param>
411     public void OnConnectionSuspended(int cause)
412     {
413         string message = string.Format("{0} - {1} {2}",
414         ↳  CrossGeofence.Id, "Connection to Google Play services suspended
415         ↳  with error code ", cause);
416         System.Diagnostics.Debug.WriteLine(message);
417         CrossGeofence.GeofenceListener.OnError(message);
418     }
419
420     /// <summary>
421     /// Raises the monitoring removal event.
422     /// </summary>
423     internal void OnMonitoringRemoval()
424     {
425         if (this.modeRegions.Count == 0)
426         {
427             CrossGeofence.GeofenceListener.OnMonitoringStopped();
428
429             this.StopLocationUpdates();
430
431             this.modeGoogleApiClient.Disconnect();
432         }
433     }

```

```
428 }
429
430     /// <summary>
431     /// Starts the location updates.
432     /// </summary>
433     internal void StartLocationUpdates()
434     {
435         var modeLocationRequest = new
436             Android.Gms.Location.LocationRequest();
437             modeLocationRequest.SetInterval(CrossGeofence.LocationUpdat_]
438             esInterval == 0 ? 30000 :
439             CrossGeofence.LocationUpdatesInterval);
440             modeLocationRequest.SetFastestInterval(CrossGeofence.Fastes_]
441             tLocationUpdatesInterval == 0 ? 5000 :
442             CrossGeofence.FastestLocationUpdatesInterval);
443             string priorityType = "Balanced Power";
444             switch (CrossGeofence.GeofencePriority)
445             {
446                 case GeofencePriority.HighAccuracy:
447                     priorityType = "High Accuracy";
448                     modeLocationRequest.SetPriority(Android.Gms.Locatio_]
449                     n.LocationRequest.PriorityHighAccuracy);
450                     break;
451                 case GeofencePriority.LowAccuracy:
452                     priorityType = "Low Accuracy";
453                     modeLocationRequest.SetPriority(Android.Gms.Locatio_]
454                     n.LocationRequest.PriorityLowPower);
455                     break;
456                 case GeofencePriority.LowestAccuracy:
457                     priorityType = "Lowest Accuracy";
458                     modeLocationRequest.SetPriority(Android.Gms.Locatio_]
459                     n.LocationRequest.PriorityNoPower);
460                     break;
461                 default:
462                     modeLocationRequest.SetPriority(Android.Gms.Locatio_]
463                     n.LocationRequest.PriorityBalancedPowerAccuracy);
464                     break;
465             }
466
467             System.Diagnostics.Debug.WriteLine(string.Format("{0} - "
468             {1}: {2}", CrossGeofence.Id, "Priority set to", priorityType));
469             ////(Regions.Count == 0) ?
470             (CrossGeofence.SmallestDisplacement==0?50
471             :CrossGeofence.SmallestDisplacement): Regions.Min(s =>
472             (float)s.Value.Radius)
473             if (CrossGeofence.SmallestDisplacement > 0)
474             {
```

```

462             modeLocationRequest.SetSmallestDisplacement(CrossGeofence_
463             .SmallestDisplacement);
464             System.Diagnostics.Debug.WriteLine(string.Format("{0} - 
465             {1}: {2} meters", CrossGeofence.Id, "Location smallest displacement
466             set to", CrossGeofence.SmallestDisplacement));
467         }
468
469         /// <summary>
470         /// Sets the last known location.
471         /// </summary>
472         /// <param name="location">Location object.</param>
473         internal void SetLastKnownLocation(Android.Locations.Location
474             location)
475         {
476             if (location != null)
477             {
478                 if (this.lastKnownGeofenceLocation == null)
479                 {
480                     this.lastKnownGeofenceLocation = new
481                         GeofenceLocation();
482                 }
483
484                 this.lastKnownGeofenceLocation.Latitude =
485                     location.Latitude;
486                 this.lastKnownGeofenceLocation.Longitude =
487                     location.Longitude;
488                 double seconds = location.Time / 1000;
489                 this.lastKnownGeofenceLocation.Date = new
490                     DateTime(1970, 1, 1, 0, 0, 0,
491                     DateTimeKind.Local).AddSeconds(seconds);
492             }
493         }
494
495         /// <summary>
496         /// Stops the location updates.
497         /// </summary>
498         internal void StopLocationUpdates()
499         {
500             Android.Gms.Location.LocationServices.FusedLocationApi.Remo
501             veLocationUpdates(this.modeGoogleApiClient,
502             GeofenceLocationListener.SharedInstance);
503         }

```

```

496
497     /// <summary>
498     /// Adds the geofence result.
499     /// </summary>
500     /// <param name="identifier">Identifier string.</param>
501     internal void AddGeofenceResult(string identifier)
502     {
503         this.modeGeofenceResults.Add(
504             identifier,
505             new GeofenceResult
506             {
507                 RegionId = identifier,
508                 Transition = GeofenceTransition.Unknown
509             });
510     }
511
512     /// <summary>
513     /// Requests the monitoring start.
514     /// </summary>
515     private void RequestMonitoringStart()
516     {
517         // If connected to google play services then add regions
518         if (this.modeGoogleApiClient.IsConnected)
519         {
520             this.AddGeofences();
521         }
522         else
523         {
524             // If not connection then connect
525             if (!this.modeGoogleApiClient.IsConnecting)
526             {
527                 this.modeGoogleApiClient.Connect();
528             }
529
530             // Request to add geofence regions once connected
531             this.CurrentRequestType = RequestType.Add;
532         }
533     }
534
535     /// <summary>
536     /// Adds the geofences.
537     /// </summary>
538     private void AddGeofences()
539     {
540         try
541         {

```

```
542         var geofenceList = new
543             List<Android.Gms.Location.IGeofence>();
544             var regions = this.Regions.Values;
545             foreach (GeofenceCircularRegion region in regions)
546             {
547                 int transitionTypes = 0;
548
549                 if (region.NotifyOnStay)
550                 {
551                     transitionTypes |=
552                         Android.Gms.Location.Geofence.GeofenceTransitionDwell;
553
554                 if (region.NotifyOnEntry)
555                 {
556                     transitionTypes |=
557                         Android.Gms.Location.Geofence.GeofenceTransitionEnter;
558
559                 if (region.NotifyOnExit)
560                 {
561                     transitionTypes |=
562                         Android.Gms.Location.Geofence.GeofenceTransitionExit;
563
564                 if (transitionTypes != 0)
565                 {
566                     geofenceList.Add(new
567                         Android.Gms.Location.GeofenceBuilder()
568                             .SetRequestId(region.Id)
569                             .SetCircularRegion(region.Latitude,
570                                 region.Longitude, (float)region.Radius)
571                             .SetLoiteringDelay((int)region.StayedInThreshold);
572
573                     // SetNotificationResponsiveness(mNotificationResponsiveness)
574
575                     .SetExpirationDuration(Android.Gms.Location.Geofence.NeverExpire)
576                     .SetTransitionTypes(transitionTypes)
577                     .Build());
578
579
580                 if (GeofenceStore.SharedInstance.Get(region.Id)
581                     == null)
582                 {
583                     GeofenceStore.SharedInstance.Save(region);
584                 }
585             }
586         }
587     }
588 }
```

```

579     →  CrossGeofence.GeofenceListener.OnMonitoringStarted(region.Id);
580         }
581     }
582
583     if (geofenceList.Count > 0)
584     {
585         Android.Gms.Location.GeofencingRequest request =
586         →  new Android.Gms.Location.GeofencingRequest.Builder().SetInitialTrig
587         →  ger(Android.Gms.Location.GeofencingRequest.InitialTriggerEnter).Add
588         →  Geofences(geofenceList).Build();
589
590         Android.Gms.Location.LocationServices.GeofencingApi
591         →  .AddGeofences(this.modeGoogleApiClient, request,
592         →  this.GeofenceTransitionPendingIntent).SetResultCallback(this);
593
594         this.CurrentRequestType = RequestType.Default;
595     }
596 }
597 catch (Java.Lang.Exception ex1)
598 {
599     string message = string.Format("{0} - Error: {1}",
600     →  CrossGeofence.Id, ex1);
601     System.Diagnostics.Debug.WriteLine(message);
602     CrossGeofence.GeofenceListener.OnError(message);
603 }
604
605
606 /// <summary>
607 /// Removes the geofences.
608 /// </summary>
609 private void RemoveGeofences()
610 {
611     GeofenceStore.SharedInstance.RemoveAll();
612     this.modeRegions.Clear();
613     this.modeGeofenceResults.Clear();
614     Android.Gms.Location.LocationServices.GeofencingApi.RemoveG
615     →  eofences(this.modeGoogleApiClient,
616     →  this.GeofenceTransitionPendingIntent).SetResultCallback(this);
617     this.StopLocationUpdates();

```

```

616         this.modeGoogleApiClient.Disconnect();
617         CrossGeofence.GeofenceListener.OnMonitoringStopped();
618     }
619
620     /// <summary>
621     /// Initializes the google AP.
622     /// </summary>
623     private void InitializeGoogleAPI()
624     {
625         int queryResult = GoogleApiAvailability.Instance.IsGooglePlayServicesAvailable(Application.Context);
626
627         if (queryResult == ConnectionResult.Success)
628         {
629             if (this.modeGoogleApiClient == null)
630             {
631                 this.modeGoogleApiClient = new
632                     GoogleApiClient.Builder(Application.Context).AddApi(Android.Gms.Location.LocationServices.API).AddConnectionCallbacks(this).AddOnConnectionFailedListener(this).Build();
633                 string message = string.Format("{0} - {1}",
634                     CrossGeofence.Id, "Google Play services is available.");
635                 System.Diagnostics.Debug.WriteLine(message);
636             }
637             if (!this.modeGoogleApiClient.IsConnected)
638             {
639                 this.modeGoogleApiClient.Connect();
640             }
641         }
642         else
643         {
644             string message = string.Format("{0} - {1}",
645                     CrossGeofence.Id, "Google Play services is unavailable.");
646             System.Diagnostics.Debug.WriteLine(message);
647             CrossGeofence.GeofenceListener.OnError(message);
648         }
649
650     /// <summary>
651     /// Removes the geofences.
652     /// </summary>
653     /// <param name="regionIdentifiers">Region identifiers.</param>
654     private void RemoveGeofences(IList<string> regionIdentifiers)
655     {
656         foreach (string identifier in regionIdentifiers)
657         {

```

```

657         // Remove this region from regions dictionary and
658         ↵ results
659             this.RemoveRegion(identifier);
660
661             // Remove from persistent store
662             GeofenceStore.SharedInstance.Remove(identifier);
663
664             // Notify monitoring was stopped
665         ↵ CrossGeofence.GeofenceListener.OnMonitoringStopped(identifier);
666         }
667
668             // Stop Monitoring
669             Android.Gms.Location.LocationServices.GeofencingApi.RemoveG
670             ↵ eofences(this.modeGoogleApiClient,
671             ↵ regionIdentifiers).SetResultCallback(this);
672
673             // Check if there are still regions
674             this.OnMonitoringRemoval();
675         }
676
677             /// <summary>
678             /// Removes the region.
679             /// </summary>
680             /// <param name="regionIdentifier">Region identifier.</param>
681             private void RemoveRegion(string regionIdentifier)
682             {
683                 if (this.modeRegions.ContainsKey(regionIdentifier))
684                 {
685                     this.modeRegions.Remove(regionIdentifier);
686                 }
687
688                 if (this.modeGeofenceResults.ContainsKey(regionIdentifier))
689                 {
690                     this.modeGeofenceResults.Remove(regionIdentifier);
691                 }
692             }

```

32 Droid/Geofence/GeofenceBroadcastReceiver.cs

```
22  namespace TreeWatch.Droid
23  {
24      using Android.App;
25      using Android.Content;
26      using Android.Support.V4.Content;
27
28      /// <summary>
29      /// GeofenceBootReceiver class
30      /// Receive geofence updates
31      /// </summary>
32      [BroadcastReceiver]
33      [IntentFilter(new[] { "nl.gtl.treewatch.ACTION_RECEIVE_GEOFENCE" })]
34      public class GeofenceBroadcastReceiver : WakefulBroadcastReceiver
35      {
36          /// <Docs>The Context in which the receiver is running.</Docs>
37          /// <summary>
38          /// This method is called when the BroadcastReceiver is
39          → receiving an Intent
40              /// broadcast.
41              /// </summary>
42              /// <param name="context">Context object.</param>
43              /// <param name="intent">Intent object.</param>
44          public override void OnReceive(Context context, Intent intent)
45          {
46              System.Diagnostics.Debug.WriteLine(string.Format("{0} - "
47              → {1}", CrossGeofence.Id, "Region State Change Received"));
48              var serviceIntent = new Intent(context,
49              → typeof(GeofenceTransitionsIntentService));
50
51              → serviceIntent.AddFlags(ActivityFlags.IncludeStoppedPackages);
52              → serviceIntent.ReplaceExtras(intent.Extras);
53              → serviceIntent.SetAction(intent.Action);
54              → WakefulBroadcastReceiver.StartWakefulService(context,
55              → serviceIntent);
56
57              this.ResultCode = Result.Ok;
58          }
59      }
60  }
```

33 Droid/Geofence/GeofenceLocationListener.cs

```
22  namespace TreeWatch.Droid
23  {
24      using Android.Gms.Location;
25
26      /// <summary>
27      /// Geofence location listener.
28      /// </summary>
29      public class GeofenceLocationListener : Java.Lang.Object,
30      ↪ ILocationListener
31      {
32          /// <summary>
33          /// The shared instance.
34          /// </summary>
35          private static GeofenceLocationListener sharedInstance = new
36          ↪ GeofenceLocationListener();
37
38          /// <summary>
39          /// Prevents a default instance of the <see
40          ↪  cref="TreeWatch.Droid.GeofenceLocationListener"/> class from being
41          ↪ created.
42          /// </summary>
43          private GeofenceLocationListener()
44          {
45          }
46
47          /// <summary>
48          /// Gets the shared instance.
49          /// </summary>
50          /// <value>The shared instance.</value>
51          public static GeofenceLocationListener SharedInstance
52          {
53              get { return sharedInstance; }
54          }
55
56          /// <summary>
57          /// Raises the location changed event.
58          /// </summary>
59          /// <param name="location">Location object.</param>
60          void
61          ↪ ILocationListener.OnLocationChanged(Android.Locations.Location
62          ↪ location)
63          {
64              // Location Updated
```

```
59         System.Diagnostics.Debug.WriteLine(string.Format("{0} - "
60     ↵ {1}: {2}, {3}", CrossGeofence.Id, "Location Update",
61     ↵ location.Latitude, location.Longitude));
62         ((GeofenceImplementation)CrossGeofence.Current).SetLastKnown_
63     ↵ nLocation(location);
64         /*
65         GeofenceCircularRegion region = null;
66         CrossGeofence.Current.Regions.TryGetValue("Fontys", out
67     ↵ region);
68         System.Diagnostics.Debug.WriteLine(region.Radius);
69     ↵ */
70     ↵ }
71 }
72 }
```

34 Droid/Geofence/ GeofenceTransitionsIntentService.cs

```
22 namespace TreeWatch.Droid
23 {
24     using System;
25     using System.Collections.Generic;
26     using Android.App;
27     using Android.Content;
28     using Android.Media;
29     using Android.OS;
30     using Android.Support.V4.App;
31
32     /// <summary>
33     /// GeofenceTransitionsIntentService class
34     /// Service that handles geofence events
35     /// </summary>
36     [Service]
37     public class GeofenceTransitionsIntentService : IntentService
38     {
39         /// <summary>
40         /// The notification max identifier.
41         /// </summary>
42         private const int NotificationMaxId = 6;
43
44         /// <summary>
45         /// The notification identifier.
46         /// </summary>
47         private static int notificationId;
48
49         /// <summary>
50         /// Creates the notification.
51         /// </summary>
52         /// <param name="title">Title string.</param>
53         /// <param name="message">Message string.</param>
54         public static void CreateNotification(string title, string
55             message)
56         {
57             try
58             {
59                 NotificationCompat.Builder builder;
60                 Context context = Android.App.Application.Context;
61
62                 Intent resultIntent = context.PackageManager.GetLaunchI
63                 ntentForPackage(context.PackageName);
```

```
63 // Create a PendingIntent; we're only using one
64 → PendingIntent (ID = 0):
65     const int PendingIntentId = 0;
66     PendingIntent resultPendingIntent =
67     PendingIntent.getActivity(context, PendingIntentId, resultIntent,
68     PendingIntentFlags.OneShot);
69
70     // Build the notification
71     builder = new NotificationCompat.Builder(context)
72         .SetAutoCancel(true) // dismiss the notification
73     ← from the notification area when the user clicks on it
74         .SetContentIntent(resultPendingIntent) // start up
75     ← this activity when the user clicks the intent.
76         .SetContentTitle(title) // Set the title
77
78     .SetSound(RingtoneManager.GetDefaultUri(RingtoneType.Notification))
79         .SetSmallIcon(context.ApplicationInfo.Icon) // This
80     ← is the icon to display
81         .SetContentText(message); // the message to display.
82
83     var notificationManager =
84     context.GetSystemService(Context.NotificationService) as
85     NotificationManager;
86
87     if (notificationId >= NotificationMaxId)
88     {
89         notificationId = 0;
90     }
91
92     notificationManager.Notify(notificationId++,
93     builder.Build());
94     }
95     catch (Java.Lang.Exception ex)
96     {
97         System.Diagnostics.Debug.WriteLine(string.Format("{0} - {1}",
98         CrossGeofence.Id, ex));
99     }
100     catch (Exception ex1)
101     {
102         System.Diagnostics.Debug.WriteLine(string.Format("{0} - {1}",
103         CrossGeofence.Id, ex1));
104     }
105 }
106
107 /// <Docs>To be added.</Docs>
108 /// <remarks>To be added.</remarks>
109 /// <summary>
```

```

98     /// Raises the handle intent event.
99     /// </summary>
100    /// <param name="intent">Intent object.</param>
101   protected override void OnHandleIntent(Intent intent)
102   {
103       Context context = Android.App.Application.Context;
104       Bundle extras = intent.Extras;
105       Android.Gms.Location.GeofencingEvent geofencingEvent =
106       ↳ Android.Gms.Location.GeofencingEvent.FromIntent(intent);
107
108       if (geofencingEvent.HasError)
109       {
110           string errorMessage = Android.Gms.Location.GeofenceStatu
111           ↳ usCodes.GetStatusCodeString(geofencingEvent.ErrorCode);
112           string message = string.Format("{0} - {1}",
113           ↳ CrossGeofence.Id, errorMessage);
114           System.Diagnostics.Debug.WriteLine(message);
115           CrossGeofence.GeofenceListener.OnError(message);
116       }
117
118       // Get the transition type.
119       int geofenceTransition = geofencingEvent.GeofenceTransition;
120
121       // Get the geofences that were triggered. A single event
122       // can trigger multiple geofences.
123       IList<Android.Gms.Location.IGeofence> triggeringGeofences =
124       ↳ geofencingEvent.TriggeringGeofences;
125
126       GeofenceTransition geoTransition;
127
128       ((GeofenceImplementation)CrossGeofence.Current).CurrentRequestType
129       = GeofenceImplementation.RequestType.Update;
130
131       foreach (Android.Gms.Location.IGeofence geofence in
132       ↳ triggeringGeofences)
133       {
134           if (!CrossGeofence.Current.GeofenceResults.ContainsKey(
135           ↳ geofence.RequestId))
136           {
137               ((GeofenceImplementation)CrossGeofence.Current).Add(
138               ↳ GeofenceResult(geofence.RequestId));
139           }
140
141           /// geofencingEvent.TriggeringLocation.Accuracy

```

```
133     ↵     CrossGeofence.Current.GeofenceResults[geofence.RequestId].Latitude
134     ↵     = geofencingEvent.TriggeringLocation.Latitude;
135
136     ↵     CrossGeofence.Current.GeofenceResults[geofence.RequestId].Longitude
137     ↵     = geofencingEvent.TriggeringLocation.Longitude;
138
139     ↵     CrossGeofence.Current.GeofenceResults[geofence.RequestId].Accuracy
140     ↵     = geofencingEvent.TriggeringLocation.Accuracy;
141
142     ↵     double seconds =
143     ↵     geofencingEvent.TriggeringLocation.Time / 1000;
144     ↵     DateTime resultDate = new DateTime(1970, 1, 1, 0, 0, 0,
145     ↵     DateTimeKind.Utc).AddSeconds(seconds).ToLocalTime();
146
147     ↵     ////DateTime resultDate = DateTime.Now;
148
149     ↵     switch (geofenceTransition)
150     ↵     {
151     ↵         ↵         case
152     ↵         ↵         Android.Gms.Location.Geofence.GeofenceTransitionEnter:
153     ↵             ↵             geoTransition = GeofenceTransition.Entered;
154     ↵             ↵             CrossGeofence.Current.GeofenceResults[geofence.
155     ↵             ↵             RequestId].LastEnterTime =
156     ↵             ↵             resultDate;
157     ↵             ↵             CrossGeofence.Current.GeofenceResults[geofence.]
158     ↵             ↵             RequestId].LastExitTime =
159     ↵             ↵             null;
160     ↵             ↵             break;
161     ↵         ↵         case
162     ↵         ↵         Android.Gms.Location.Geofence.GeofenceTransitionExit:
163     ↵             ↵             geoTransition = GeofenceTransition.Exited;
164     ↵             ↵             CrossGeofence.Current.GeofenceResults[geofence.
165     ↵             ↵             RequestId].LastExitTime =
166     ↵             ↵             resultDate;
167     ↵             ↵             break;
168     ↵         ↵         case
169     ↵         ↵         Android.Gms.Location.Geofence.GeofenceTransitionDwell:
170     ↵             ↵             geoTransition = GeofenceTransition.Stayed;
171     ↵             ↵             break;
172     ↵         ↵         default:
173     ↵             ↵             string message = string.Format("{0} - {1}",
174     ↵             ↵             CrossGeofence.Id, "Invalid transition type");
175     ↵             ↵             System.Diagnostics.Debug.WriteLine(message);
176     ↵             ↵             geoTransition = GeofenceTransition.Unknown;
177     ↵             ↵             break;
178     ↵     }
179
180 }
```

```
162
163             System.Diagnostics.Debug.WriteLine(string.Format("{0} - "
164             → Transition: {1}", CrossGeofence.Id, geoTransition));
165             if (CrossGeofence.Current.GeofenceResults[geofence.Requ
166             → estId].Transition !=
167             → geoTransition)
168             {
169                 CrossGeofence.Current.GeofenceResults[geofence.Requ
170             → estId].Transition =
171             → geoTransition;
172
173             CrossGeofence.GeofenceListener.OnRegionStateChanged [
174             → (CrossGeofence.Current.GeofenceResults[geofence.RequestId]);
175
176             if
177             → (CrossGeofence.Current.Regions.ContainsKey(geofence.RequestId) &&
178             → CrossGeofence.Current.Regions[geofence.RequestId].ShowNotification)
179             {
180                 string message = CrossGeofence.Current.Geofence [
181             → Results[geofence.RequestId].ToString();
182
183                 if
184             → (CrossGeofence.Current.Regions.ContainsKey(geofence.RequestId))
185             {
186                 switch (geoTransition)
187                 {
188                     case GeofenceTransition.Entered:
189                         message =
190                         string.IsNullOrEmpty(CrossGeofence.Current.Regions[geofence.Request
191                         → Id].NotificationEntryMessage) ? message :
192                         CrossGeofence.Current.Regions[geofence.RequestId].NotificationEntry [
193                         → Message;
194                         break;
195                     case GeofenceTransition.Exited:
196                         message =
197                         string.IsNullOrEmpty(CrossGeofence.Current.Regions[geofence.Request
198                         → Id].NotificationExitMessage) ? message :
199                         CrossGeofence.Current.Regions[geofence.RequestId].NotificationExitM [
200                         → essage;
201                         break;
202                     case GeofenceTransition.Stayed:
203                         message =
204                         string.IsNullOrEmpty(CrossGeofence.Current.Regions[geofence.Request
205                         → Id].NotificationStayMessage) ? message :
206                         CrossGeofence.Current.Regions[geofence.RequestId].NotificationStayM [
207                         → essage;
208                         break;
```

```
187                         }
188                     }
189
190                     CreateNotification(context.ApplicationInfo.Load_]
191             ↳ Label(context.PackageManager),
192             ↳ message);
193             }
194         }
195     }
196 }
```

35 TreeWatch/Utils/GeoHelper.cs

```
22  namespace TreeWatch
23  {
24      using System;
25      using System.Collections.Generic;
26
27      /// <summary>
28      /// Helper class for geo related functions.
29      /// </summary>
30      public static class GeoHelper
31      {
32          /// <summary>
33          /// Determines if a position is inside the polygon defined by
34          /// the coordinates.
35          /// </summary>
36          /// <returns><c>true</c> is inside the polygon defined by the
37          /// coordinates; otherwise, <c>false</c>.</returns>
38          /// <param name="coordinates">coordinates of polygon.</param>
39          /// <param name="position">Position which should be tested if
40          /// in.</param>
41          public static bool IsInsideCoords(List<Position> coordinates,
42          Position position)
43          {
44              int i, j;
45              int nvert = coordinates.Count;
46
47              bool inside = false;
48
49              if (coordinates.Count < 3)
50              {
51                  return inside;
52              }
53
54              for (i = 0, j = nvert - 1; i < nvert; j = i++)
55              {
56                  if (((coordinates[i].Latitude > position.Latitude) !=
57                      (coordinates[j].Latitude > position.Latitude))
58                      && (position.Longitude < ((coordinates[j].Longitude -
59                          coordinates[i].Longitude) * (position.Latitude -
60                          coordinates[i].Latitude) / ((coordinates[j].Latitude -
61                          coordinates[i].Latitude) + coordinates[i].Longitude))))
62                  {
63                      inside = !inside;
64                  }
65              }
66          }
67      }
68  }
```

```

58
59         return inside;
60     }
61
62     /// <summary>
63     /// Calculates the bounding box for a list of coordinates.
64     /// </summary>
65     /// <returns>The bounding box.</returns>
66     /// <param name="boundingCoordinates">List of
67     ← coordinates.</param>
68     public static BoundingBox CalculateBoundingBox(List<Position>
69     ← boundingCoordinates)
70     {
71         if (boundingCoordinates.Count < 2)
72         {
73             return new BoundingBox { Width = 0d, Height = 0d };
74
75         double smallestLongitude = boundingCoordinates[0].Longitude;
76         double biggestLongitude = smallestLongitude;
77         double smallestLatitude = boundingCoordinates[0].Latitude;
78         double biggestLatitude = smallestLatitude;
79
80         for (int i = 1; i < boundingCoordinates.Count; i++)
81         {
82             smallestLongitude =
83                 Math.Min(boundingCoordinates[i].Longitude, smallestLongitude);
84             biggestLongitude =
85                 Math.Max(boundingCoordinates[i].Longitude, biggestLongitude);
86
87             smallestLatitude =
88                 Math.Min(boundingCoordinates[i].Latitude, smallestLatitude);
89             biggestLatitude =
90                 Math.Max(boundingCoordinates[i].Latitude, biggestLatitude);
91         }
92
93         double width = biggestLongitude - smallestLongitude;
94         double height = biggestLatitude - smallestLatitude;
95         var center = new Position(smallestLatitude + (height *
96             0.5), smallestLongitude + (width * 0.5));
97
98         return new BoundingBox
99         {
100             Width = width,
101             Height = height,
102             Center = center,

```

```

97             WidthInMeters = DistanceInMeters(smallestLatitude,
98     ↵     smallestLongitude, smallestLatitude, biggestLongitude),
99             HeightInMeters = DistanceInMeters(smallestLatitude,
100    ↵     smallestLongitude, biggestLatitude, smallestLongitude)
101         );
102     }
103
104     /**
105      * Calculates the distances between two coordinates the in
106      * meters.
107     */
108     /**
109      * The in meters.
110     */
111     /**
112      * Latitude of coordinate 1.
113      * Longitude of coordinate 1.
114      * Latitude of coordinate 2.
115      * Longitude of coordinate 2.
116      */
117     public static double DistanceInMeters(double latitude1, double
118     ↵     longitude1, double latitude2, double longitude2)
119     {
120         const double EarthRadius = 6371000; // meters
121         double doubleLatitude = ToRadians(latitude2 - latitude1);
122         double doubleLongitude = ToRadians(longitude2 - longitude1);
123         double a = (Math.Sin(doubleLatitude / 2) *
124             Math.Sin(doubleLatitude / 2)) +
125                 (Math.Cos(ToRadians(latitude1)) *
126                 Math.Cos(ToRadians(latitude2)) * Math.Sin(doubleLongitude / 2) *
127                 Math.Sin(doubleLongitude / 2));
128         double c = 2 * Math.Atan2(Math.Sqrt(a), Math.Sqrt(1 - a));
129         float dist = (float)(EarthRadius * c);
130
131         return dist;
132     }
133
134     /**
135      * Converts angle in degrees to radians
136      */
137     /**
138      * Angle as double.
139      */
140     public static double ToRadians(double angle)
141     {
142         return (Math.PI / 180) * angle;
143     }
144
145     /**
146      * HelperStruct to return a box.
147      */
148     public struct BoundingBox

```

```

137  {
138      ///<summary>
139      /// The width.
140      ///</summary>
141      public double Width;
142
143      ///<summary>
144      /// The height.
145      ///</summary>
146      public double Height;
147
148      ///<summary>
149      /// The center.
150      ///</summary>
151      public Position Center;
152
153      ///<summary>
154      /// The width in meters.
155      ///</summary>
156      public double WidthInMeters;
157
158      ///<summary>
159      /// The height in meters.
160      ///</summary>
161      public double HeightInMeters;
162  }
163 }
164 }
```

36 iOS/Map/FieldMapRenderer.cs

```
22  using TreeWatch;
23  using TreeWatch.iOS;
24  using Xamarin.Forms;
25
26  [assembly: ExportRenderer(typeof(FieldMap), typeof(FieldMapRenderer))]
27  [assembly: System.Diagnostics.CodeAnalysis.SuppressMessage("StyleCop.CS",
28    "SA1300:ElementMustBeginWithUpperCaseLetter",
29    MessageId = "Ctl", Scope = "namespace", Target = "Assembly name",
30    Justification = "Auto generated name")]
31
32  // Analysis disable once InconsistentNaming
33  namespace TreeWatch.iOS
34  {
35      using System;
36      using System.Collections.Generic;
37      using System.Linq;
38      using CoreGraphics;
39      using CoreLocation;
40      using MapKit;
41      using ObjCRuntime;
42      using UIKit;
43      using Xamarin.Forms.Maps.iOS;
44      using Xamarin.Forms.Platform.iOS;
45
46      /// <summary>
47      /// iOS specific MapRender
48      /// </summary>
49      public class FieldMapRenderer : MapRenderer
50      {
51          /// <summary>
52          /// The mercator radius.
53          /// </summary>
54          private const double MercatorRadius = 85445659.44705395;
55
56          /// <summary>
57          /// The max google zoom levels.
58          /// </summary>
59          private const int MaxGoogleLevels = 20;
60
61          /// <summary>
62          /// The map view.
63          /// </summary>
64          private MKMapView mapView;
```

```
63     ///<summary>
64     /// The FieldMap containing data for the map.
65     ///</summary>
66     private FieldMap myMap;
67
68     ///<summary>
69     /// The tap gesture.
70     ///</summary>
71     private UITapGestureRecognizer tapGesture;
72
73     ///<summary>
74     /// The field helper.
75     ///</summary>
76     private FieldHelper fieldHelper;
77
78     ///<summary>
79     /// Initializes a new instance of the <see
80     ↵ cref="TreeWatch.iOS.FieldMapRenderer"/> class.
81     ///</summary>
82     public FieldMapRenderer()
83     {
84         this.tapGesture = new
85             UITapGestureRecognizer(this.MapTapped);
86         this.tapGesture.NumberOfTapsRequired = 1;
87         this.fieldHelper = FieldHelper.Instance;
88         this.fieldHelper.FieldSelected += this.FieldSelected;
89         this.fieldHelper.CenterUserPosition +=
90             this.CenterOnUserPosition;
91     }
92
93     ///<summary>
94     /// Converts the coordinates.
95     ///</summary>
96     ///<returns>The coordinates.</returns>
97     ///<param name="coordinates">Coordinates of a position.</param>
98     public static CLLocationCoordinate2D[]
99         ConvertCoordinates(List<Position> coordinates)
100    {
101        var points = new CLLocationCoordinate2D[coordinates.Count];
102        var i = 0;
103        foreach (var pos in coordinates)
104        {
105            points[i] = new CLLocationCoordinate2D(pos.Latitude,
106            pos.Longitude);
107            i++;
108        }
109    }
110}
```

```

105         return points;
106     }
107
108     /// <summary>
109     /// Gets called when a field is selected.
110     /// </summary>
111     /// <param name="sender">Sender who fired the event.</param>
112     /// <param name="e">Event Arguments.</param>
113     protected void FieldSelected(object sender,
114         FieldSelectedEventArgs e)
115     {
116         if (this.mapView != null)
117         {
118             var widthHeight =
119                 GeoHelper.CalculateBoundingBox(e.Field.BoundingCoordinates);
120             var center = widthHeight.Center;
121             var coords = new
122                 CLLocationCoordinate2D(center.Latitude, center.Longitude);
123             var span = new MKCoordinateSpan(widthHeight.Width *
124                 1.1, widthHeight.Height * 1.1);
125             this.mapView.Region = new MKCoordinateRegion(coords,
126                 span);
127         }
128     }
129
130     /// <summary>
131     /// Get called when Map is displayed.
132     /// </summary>
133     /// <param name="e">Event Arguments.</param>
134     protected override void
135         OnElementChanged(ElementChangedEventArgs<View> e)
136     {
137         base.OnElementChanged(e);
138
139         if (e.OldElement == null)
140         {
141             this.mapView = Control as MKMapView;
142             this.mapView.ShowsUserLocation = true;
143             this.mapView.DidUpdateUserLocation +=
144                 this.SetUserPositionOnce;
145             this.mapView.AddGestureRecognizer(this.tapGesture);
146
147             /* Read LFHeatMap project first
148             * Found at https://github.com/TreeWatch/LFHeatMaps
149             * Code:
150             * mapView.RegionChanged += ChangeRegion;
151             */
152     }

```

```

145
146         this.myMap = e.NewElement as FieldMap;
147         this.mapView.OverlayRenderer = this.GetOverlayRender;
148
149         this.AddFields();
150     }
151 }
152
153     /// <summary>
154     /// Sets the user position once.
155     /// </summary>
156     /// <param name="sender">Sender who fired the event.</param>
157     /// <param name="e">Event Arguments.</param>
158     protected void SetUserPositionOnce(object sender,
159     MKUserLocationEventArgs e)
160     {
161         this.mapView.DidUpdateUserLocation -=
162         this.SetUserPositionOnce;
163         this.mapView.SetCenterCoordinate(e.UserLocation.Location.Co_
164         ordinate,
165         true);
166     }
167
168     /// <summary>
169     /// Centers on the user position.
170     /// </summary>
171     /// <param name="sender">Sender who fired the event.</param>
172     /// <param name="e">Event Arguments.</param>
173     protected void CenterOnUserPosition(object sender, EventArgs e)
174     {
175         if (this.mapView != null &&
176             this.mapView.UserLocation.Location != null)
177         {
178             this.mapView.SetCenterCoordinate(this.mapView.UserLocat_
179             ion.Location.Coordinate,
180             true);
181         }
182         else
183         {
184             this.mapView.DidUpdateUserLocation +=
185             this.SetUserPositionOnce;
186         }
187     }
188
189     /// <summary>
190     /// Calculates the zoomlevel.
191     /// </summary>

```

```

184     ///<returns>The level.</returns>
185     ///<param name="mapView">Map view.</param>
186     private static double ZoomLevel(MKMapView mapView)
187     {
188         var longitudeDelta = mapView.Region.Span.LongitudeDelta;
189         var mapWidthInPixels = mapView.Bounds.Size.Width;
190
191         double zoomScale = longitudeDelta * MercatorRadius *
192             Math.PI / (180.0 * mapWidthInPixels);
193         double zoomLevel = MaxGoogleLevels - Math.Log(zoomScale,
194             2.0);
195         if (zoomLevel < 0)
196         {
197             zoomLevel = 0;
198         }
199
200         return zoomLevel;
201     }
202
203     /* TODO Readd LFHeatMap project first
204      * Found at https://github.com/TreeWatch/LFHeatMaps
205      * Code:
206      void ChangeRegion(object sender, MKMapViewChangeEventArgs e){
207
208         foreach (var item in mapView.Subviews)
209         {
210             var heatMap = item as UIHeatMapView;
211             if (heatMap != null)
212                 heatMap.RefreshHeatMap(mapView);
213         }
214     */
215     ///<summary>
216     ///Gets the overlay render for a overlay.
217     ///</summary>
218     ///<returns>The overlay render.</returns>
219     ///<param name="m">The Mapview.</param>
220     ///<param name="o">The Overlay.</param>
221     private MKOverlayRenderer GetOverlayRender(MKMapView m,
222         IMKOverlay o)
223     {
224         var overlay = Runtime.GetNSObject(o.Handle) as MKPolygon;
225         if (overlay != null)
226         {
227             var polygon = overlay;
228             var polygonRenderer = new MKPolygonRenderer(polygon);

```

```
229         if (polygon.Title == "Field")
230     {
231         polygonRenderer.FillColor =
232             this.myMap.OverLayColor.ToUIColor();
233         polygonRenderer.StrokeColor =
234             this.myMap.BoundaryColor.ToUIColor();
235         polygonRenderer.LineWidth = 1;
236     }
237
238     return polygonRenderer;
239 }
240
241     if (o is MultiPolygon)
242     {
243         return new MultiPolygonView(o);
244     }
245
246     return null;
247 }
248
249 /// <summary>
250 /// Adds the fields to the map.
251 /// </summary>
252 private void AddFields()
253 {
254     var connection = new TreeWatchDatabase();
255     foreach (var field in this.myMap.Fields)
256     {
257         var query = new DBQuery<Field>(connection);
258         var blockPolygons = new List<ColorPolygon>();
259         query.GetChildren(field);
260         if (field.Blocks.Count != 0)
261         {
262             foreach (var block in field.Blocks)
263             {
264                 if (block.BoundingCoordinates.Count != 0 &&
265                     block.BoundingCoordinates.Count >= 3)
266                 {
267                     var blockPoints =
268                     ConvertCoordinates(block.BoundingCoordinates);
269                     var blockPolygon =
270                     (ColorPolygon)MKPolygon.FromCoordinates(blockPoints);
271                     blockPolygon.FillColor =
272                     block.TreeType.ColorProp.ToCGColor();
273                     blockPolygons.Add(blockPolygon);
274                 }
275             }
276         }
277     }
278 }
```

```

269 }
270
271         var blockMultiPolygon = new
272             MultiPolygon(blockPolygons);
273
274         this.mapView.AddOverlay(blockMultiPolygon);
275     }
276
277     if (field.BoundingCoordinates.Count != 0 &&
278         field.BoundingCoordinates.Count >= 3)
279     {
280         var points =
281             ConvertCoordinates(field.BoundingCoordinates);
282         var polygon = MKPolygon.FromCoordinates(points);
283         polygon.Title = "Field";
284         this.mapView.AddOverlay(polygon);
285     }
286 }
287
288 var query2 = new DBQuery<HeatMap>(connection);
289 var heatmaps = query2.GetAllWithChildren();
290 var heatmap = heatmaps[0];
291
292     this.AddHeatMap(heatmap.Points);
293 }
294
295 /// <summary>
296 /// Adds a HeatMap to the map.
297 /// </summary>
298 /// <param name="points">Points for the Heatmap.</param>
299 private void AddHeatMap(List<HeatmapPoint> points)
300 {
301     var polygons = new List<ColorPolygon>();
302
303     var max = points.Max(r => r.Mean);
304     var min = points.Min(r => r.Mean);
305
306     var difference = max - min;
307
308     foreach (var item in points)
309     {
310         var singlepolygon = (ColorPolygon)MKPolygon.FromCoordinates(
311             ConvertCoordinates(item.BoundingCoordinates));
312         var red = (((item.Mean - min) / difference) * 245) +
313             10) / 255;
314         var color = Color.FromRgb(red, 0, 1 - red);
315         singlepolygon.FillColor = color.ToCGColor();

```

```

311             singlepolygon.DrawOutlines = false;
312             polygons.Add(singlepolygon);
313         }
314
315         var heatmap = new MultiPolygon(polygons);
316
317         this.mapView.AddOverlay(heatmap);
318
319         /* Showing a 'Real' heatmap using just points
320          * current Version is using multiple polygons
321          * TODO Readd LFHeatMap project first
322          * Found at https://github.com/TreeWatch/LFHeatMaps
323          * Code :
324          var positions = new List<Position>();
325          var weights = new List<Double>();
326
327          foreach (var item in points)
328          {
329
330              foreach (var pos in item.BoundingCoordinates) {
331                  positions.Add(pos);
332                  weights.Add(item.Mean);
333              }
334          }
335          var view = new UIHeatMapView(positions, weights, mapView);
336          mapView.AddSubview(view); */
337      }
338
339      /// <summary>
340      /// Gets called when the Map is clicked.
341      /// </summary>
342      /// <param name="sender">Sender who fired the event.</param>
343      private void MapTapped(UIDevice sender)
344      {
345          CGPoint pointInView = sender.LocationInView(this.mapView);
346          CLLocationCoordinate2D touchCoordinates =
347          this.mapView.ConvertPoint(pointInView, this.mapView);
348
349          FieldHelper.Instance.MapTappedEvent(new
350          Position(touchCoordinates.Latitude, touchCoordinates.Longitude),
351          ZoomLevel(this.mapView));
352      }
353  }

```

37 iOS/Map/MultiPolygonView.cs

```
22 [assembly: System.Diagnostics.CodeAnalysis.SuppressMessage("StyleCop.CS1
  ↳ harp.NamingRules", "SA1300:ElementMustBeginWithUpperCaseLetter",
  ↳ MessageId = "Ctl", Scope = "namespace", Target = "Assembly name",
  ↳ Justification = "Auto generated name")]
23
24 // Analysis disable once InconsistentNaming
25 namespace TreeWatch.iOS
26 {
27     using System;
28     using CoreGraphics;
29     using MapKit;
30
31     /// <summary>
32     /// View for a multipolygon.
33     /// Handles custom drawing of all polygons inside a multipolygon.
34     /// </summary>
35     public class MultiPolygonView : MKOverlayRenderer
36     {
37         /// <summary>
38         /// The polygon overlay.
39         /// </summary>
40         private IMKOverlay polygonOverlay;
41
42         /// <summary>
43         /// Initializes a new instance of the <see
44         ↳ cref="TreeWatch.iOS.MultiPolygonView"/> class.
45         /// </summary>
46         /// <param name="overlay">Overlay to be displayed.</param>
47         public MultiPolygonView(IMKOverlay overlay)
48         {
49             this.polygonOverlay = overlay;
50         }
51
52         /// <summary>
53         /// Draws the map rectangle.
54         /// </summary>
55         /// <param name="mapRect">Map rectangle.</param>
56         /// <param name="zoomScale">Zoom scale.</param>
57         /// <param name="context"> Graphics context.</param>
58         public override void DrawMapRect(MKMapRect mapRect, nfloat
  ↳ zoomScale, CGContext context)
59         {
60             base.DrawMapRect(mapRect, zoomScale, context);
61             var multiPolygons = (MultiPolygon)this.polygonOverlay;
```

```

61     foreach (var item in multiPolygons.Polygons)
62     {
63         var path = new CGPath();
64         this.InvokeOnMainThread(() =>
65         {
66             path = PolyPath(item.Polygon);
67         });
68         if (path != null)
69         {
70             context.SetFillColor(item.FillColor);
71             context.BeginPath();
72             context.AddPath(path);
73             context.DrawPath(CGPathDrawingMode.EOFill);
74             if (item.DrawOutlines)
75             {
76                 context.BeginPath();
77                 context.AddPath(path);
78                 context.StrokePath();
79             }
80         }
81     }
82 }
83
84 /// <summary>
85 /// Gets the poly path for a polygon.
86 /// </summary>
87 /// <returns>The path.</returns>
88 /// <param name="polygon">Polygon to get the path for.</param>
89 public CGPath PolyPath(MKPolygon polygon)
90 {
91     var path = new CGPath();
92
93     foreach (var item in polygon.InteriorPolygons)
94     {
95         var interiorPath = this.PolyPath(item);
96         path.AddPath(interiorPath);
97     }
98
99     var relativePoint = PointForMapPoint(polygon.Points[0]);
100    path.MoveToPoint(relativePoint);
101    foreach (var point in polygon.Points)
102    {
103        path.AddLineToPoint(this.PointForMapPoint(point));
104    }
105
106    return path;
107 }

```

108 }

109 }

38 Droid/Map/FieldMapRenderer.cs

```
22  using TreeWatch;
23  using TreeWatch.Droid;
24  using Xamarin.Forms;
25
26  [assembly: ExportRenderer(typeof(FieldMap), typeof(FieldMapRenderer))]
27
28  namespace TreeWatch.Droid
29  {
30      using System;
31      using System.Collections.Generic;
32      using Android.Gms.Maps;
33      using Android.Gms.Maps.Model;
34      using Xamarin.Forms.Maps.Android;
35      using Xamarin.Forms.Platform.Android;
36
37      /// <summary>
38      /// Field map renderer.
39      /// </summary>
40      public class FieldMapRenderer : MapRenderer, IOnMapReadyCallback
41      {
42          /// <summary>
43          /// The map view.
44          /// </summary>
45          private MapView mapView;
46
47          /// <summary>
48          /// My map.
49          /// </summary>
50          private FieldMap myMap;
51
52          /// <summary>
53          /// The field helper.
54          /// </summary>
55          private FieldHelper fieldHelper;
56
57          /// <summary>
58          /// Initializes a new instance of the <see
59          → cref="TreeWatch.Droid.FieldMapRenderer"/> class.
60          /// </summary>
61          public FieldMapRenderer()
62          {
63              this.fieldHelper = FieldHelper.Instance;
64              this.fieldHelper.FieldSelected += this.FieldSelected;
65          }
66
67          /// <summary>
68          /// Gets the field helper.
69          /// </summary>
70          public FieldHelper FieldHelper
71          {
72              get { return fieldHelper; }
73          }
74
75          /// <summary>
76          /// Gets the map view.
77          /// </summary>
78          public MapView MapView
79          {
80              get { return mapView; }
81          }
82
83          /// <summary>
84          /// Handles the OnMapReady event.
85          /// </summary>
86          void IOnMapReadyCallback.OnMapReady(MapView mapView)
87          {
88              this.mapView = mapView;
89
90              // Set the map type to hybrid
91              mapView.SetMapType(MapView.MapType.Hybrid);
92
93              // Set the camera position
94              mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
95
96              // Set the map type to hybrid
97              mapView.SetMapType(MapView.MapType.Hybrid);
98
99              // Set the camera position
100             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
101
102             // Set the map type to hybrid
103             mapView.SetMapType(MapView.MapType.Hybrid);
104
105             // Set the camera position
106             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
107
108             // Set the map type to hybrid
109             mapView.SetMapType(MapView.MapType.Hybrid);
110
111             // Set the camera position
112             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
113
114             // Set the map type to hybrid
115             mapView.SetMapType(MapView.MapType.Hybrid);
116
117             // Set the camera position
118             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
119
120             // Set the map type to hybrid
121             mapView.SetMapType(MapView.MapType.Hybrid);
122
123             // Set the camera position
124             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
125
126             // Set the map type to hybrid
127             mapView.SetMapType(MapView.MapType.Hybrid);
128
129             // Set the camera position
130             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
131
132             // Set the map type to hybrid
133             mapView.SetMapType(MapView.MapType.Hybrid);
134
135             // Set the camera position
136             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
137
138             // Set the map type to hybrid
139             mapView.SetMapType(MapView.MapType.Hybrid);
140
141             // Set the camera position
142             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
143
144             // Set the map type to hybrid
145             mapView.SetMapType(MapView.MapType.Hybrid);
146
147             // Set the camera position
148             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
149
150             // Set the map type to hybrid
151             mapView.SetMapType(MapView.MapType.Hybrid);
152
153             // Set the camera position
154             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
155
156             // Set the map type to hybrid
157             mapView.SetMapType(MapView.MapType.Hybrid);
158
159             // Set the camera position
160             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
161
162             // Set the map type to hybrid
163             mapView.SetMapType(MapView.MapType.Hybrid);
164
165             // Set the camera position
166             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
167
168             // Set the map type to hybrid
169             mapView.SetMapType(MapView.MapType.Hybrid);
170
171             // Set the camera position
172             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
173
174             // Set the map type to hybrid
175             mapView.SetMapType(MapView.MapType.Hybrid);
176
177             // Set the camera position
178             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
179
180             // Set the map type to hybrid
181             mapView.SetMapType(MapView.MapType.Hybrid);
182
183             // Set the camera position
184             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
185
186             // Set the map type to hybrid
187             mapView.SetMapType(MapView.MapType.Hybrid);
188
189             // Set the camera position
190             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
191
192             // Set the map type to hybrid
193             mapView.SetMapType(MapView.MapType.Hybrid);
194
195             // Set the camera position
196             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
197
198             // Set the map type to hybrid
199             mapView.SetMapType(MapView.MapType.Hybrid);
200
201             // Set the camera position
202             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
203
204             // Set the map type to hybrid
205             mapView.SetMapType(MapView.MapType.Hybrid);
206
207             // Set the camera position
208             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
209
210             // Set the map type to hybrid
211             mapView.SetMapType(MapView.MapType.Hybrid);
212
213             // Set the camera position
214             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
215
216             // Set the map type to hybrid
217             mapView.SetMapType(MapView.MapType.Hybrid);
218
219             // Set the camera position
220             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
221
222             // Set the map type to hybrid
223             mapView.SetMapType(MapView.MapType.Hybrid);
224
225             // Set the camera position
226             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
227
228             // Set the map type to hybrid
229             mapView.SetMapType(MapView.MapType.Hybrid);
230
231             // Set the camera position
232             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
233
234             // Set the map type to hybrid
235             mapView.SetMapType(MapView.MapType.Hybrid);
236
237             // Set the camera position
238             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
239
240             // Set the map type to hybrid
241             mapView.SetMapType(MapView.MapType.Hybrid);
242
243             // Set the camera position
244             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
245
246             // Set the map type to hybrid
247             mapView.SetMapType(MapView.MapType.Hybrid);
248
249             // Set the camera position
250             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
251
252             // Set the map type to hybrid
253             mapView.SetMapType(MapView.MapType.Hybrid);
254
255             // Set the camera position
256             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
257
258             // Set the map type to hybrid
259             mapView.SetMapType(MapView.MapType.Hybrid);
260
261             // Set the camera position
262             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
263
264             // Set the map type to hybrid
265             mapView.SetMapType(MapView.MapType.Hybrid);
266
267             // Set the camera position
268             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
269
270             // Set the map type to hybrid
271             mapView.SetMapType(MapView.MapType.Hybrid);
272
273             // Set the camera position
274             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
275
276             // Set the map type to hybrid
277             mapView.SetMapType(MapView.MapType.Hybrid);
278
279             // Set the camera position
280             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
281
282             // Set the map type to hybrid
283             mapView.SetMapType(MapView.MapType.Hybrid);
284
285             // Set the camera position
286             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
287
288             // Set the map type to hybrid
289             mapView.SetMapType(MapView.MapType.Hybrid);
290
291             // Set the camera position
292             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
293
294             // Set the map type to hybrid
295             mapView.SetMapType(MapView.MapType.Hybrid);
296
297             // Set the camera position
298             mapView.SetCameraPosition(new CameraPosition(0, 0, 10, 0));
299
299 }
```

```

65
66     ///<summary>
67     ///Occurs when map ready.
68     ///</summary>
69     public event EventHandler MapReady;
70
71     ///<summary>
72     ///Gets the map.
73     ///</summary>
74     ///<value>The map.</value>
75     public new GoogleMap Map
76     {
77         get;
78         private set;
79     }
80
81     ///<summary>
82     ///Raises the map ready event.
83     ///</summary>
84     ///<param name="googleMap">Google map.</param>
85     public void OnMapReady(GoogleMap googleMap)
86     {
87         this.Map = googleMap;
88         this.AddFields();
89         this.Map.MapClick += this.MapClicked;
90         this.Map.MyLocationEnabled = true;
91         this.Map.UiSettings.MyLocationButtonEnabled = true;
92         this.Map.MyLocationChange += this.SetUserPositionOnce;
93         var handler = this.MapReady;
94         if (handler != null)
95         {
96             handler(this, EventArgs.Empty);
97         }
98     }
99
100    ///<summary>
101    ///Fields the selected.
102    ///</summary>
103    ///<param name="sender">Sender object.</param>
104    ///<param name="e">E event.</param>
105    public void FieldSelected(object sender, FieldSelectedEventArgs
106    ←   e)
107    {
108        if (this.Map != null)
109        {
110            var builder = new LatLngBounds.Builder();

```

```
110 var whc =
111     GeoHelper.CalculateBoundingBox(e.Field.BoundingCoordinates);
112     double width = whc.Width * 0.95;
113     double height = whc.Height * 0.59;
114     builder.Include(new LatLng(whc.Center.Latitude - width,
115     whc.Center.Longitude - height));
116     builder.Include(new LatLng(whc.Center.Latitude + width,
117     whc.Center.Longitude + height));
118     var bounds = builder.Build();
119
120     this.Map.MoveCamera(CameraUpdateFactory.NewLatLngBounds(bounds, 0));
121 }
122 }
123
124 /// <summary>
125 /// Raises the element changed event.
126 /// </summary>
127 /// <param name="e">E event.</param>
128 protected override void
129 OnElementChanged(ElementChangedEventArgs<View> e)
130 {
131     base.OnElementChanged(e);
132
133     if (e.OldElement == null)
134     {
135         this.mapView = Control as MapView;
136         this.mapView.GetMapAsync(this);
137
138         this.myMap = e.NewElement as FieldMap;
139     }
140 }
141
142 /// <summary>
143 /// Sets the user position once.
144 /// </summary>
145 /// <param name="sender">Sender object.</param>
146 /// <param name="e">E event.</param>
147 protected void SetUserPositionOnce(object sender,
148 GoogleMap.MyLocationChangeEventArgs e)
149 {
150     this.Map.MyLocationChange -= this.SetUserPositionOnce;
151     var latlng = new LatLng(e.Location.Latitude,
152     e.Location.Longitude);
153     this.Map.MoveCamera(CameraUpdateFactory.NewLatLng(latlng));
154 }
155
156 /// <summary>
```

```

150     /// Gets the polygon.
151     /// </summary>
152     /// <returns>The polygon.</returns>
153     /// <param name="coordinates">Coordinates iterable.</param>
154     /// <param name="color">Android Color.</param>
155     private static PolygonOptions GetPolygon(Java.Lang.IIterable
156     ↪ coordinates, Android.Graphics.Color color)
157     {
158         var polygonOptions = new PolygonOptions();
159         polygonOptions.InvokeFillColor(color);
160         polygonOptions.InvokeStrokeWidth(1);
161         polygonOptions.InvokeStrokeColor(Color.Black.ToAndroid());
162         polygonOptions.AddAll(coordinates);
163
164         return polygonOptions;
165     }
166
167     /// <summary>
168     /// Gets the polygon.
169     /// </summary>
170     /// <returns>The polygon.</returns>
171     /// <param name="coordinates">Coordinates iterable.</param>
172     /// <param name="fillColor">Fill color.</param>
173     /// <param name="boundaryColor">Boundary color.</param>
174     private static PolygonOptions GetPolygon(Java.Lang.IIterable
175     ↪ coordinates, Android.Graphics.Color fillColor,
176     ↪ Android.Graphics.Color boundaryColor)
177     {
178         var polygonOptions = new PolygonOptions();
179         polygonOptions.InvokeFillColor(fillColor);
180         polygonOptions.InvokeStrokeWidth(4);
181         polygonOptions.InvokeStrokeColor(boundaryColor);
182         polygonOptions.AddAll(coordinates);
183
184         return polygonOptions;
185     }
186
187     /// <summary>
188     /// Converts the coordinates.
189     /// </summary>
190     /// <returns>The coordinates.</returns>
191     /// <param name="coordinates">Coordinates list.</param>
192     private static Java.Util.ArrayList
193     ↪ ConvertCoordinates(List<Position> coordinates)
194     {
195         var coords = new Java.Util.ArrayList();
196         foreach (var pos in coordinates)
197     }

```

```

193         {
194             coords.Add(new LatLng(pos.Latitude, pos.Longitude));
195         }
196
197         return coords;
198     }
199
200     /// <summary>
201     /// Markers the clicked.
202     /// </summary>
203     /// <param name="sender">Sender object.</param>
204     /// <param name="e">E event.</param>
205     private static void MarkerClicked(object sender,
206     ↪ GoogleMap.MarkerEventArgs e)
207     {
208         Android.Util.Log.Verbose("Sender", sender.ToString());
209         e.Marker.ShowInfoWindow();
210     }
211
212     /// <summary>
213     /// Infos the window clicked.
214     /// </summary>
215     /// <param name="sender">Sender object.</param>
216     /// <param name="e">E event.</param>
217     private void InfoWindowClicked(object sender,
218     ↪ GoogleMap.InfoWindowEventArgs e)
219     {
220         Android.Util.Log.Verbose("Sender", sender.ToString());
221         Marker marker = e.Marker;
222         Field field = null;
223         foreach (Field f in this.myMap.Fields)
224         {
225             if (f.Name.Equals(e.Marker.Title))
226             {
227                 field = f;
228                 break;
229             }
230             if (field != null)
231             {
232                 var navigationPage =
233                 ↪ (NavigationPage)Application.Current.MainPage;
234
235                 navigationPage.PushAsync(new
236                 ↪ FieldInformationContentPage(new InformationViewModel(field)));
237             }

```

```

236     }
237
238     /// <summary>
239     /// Adds the fields.
240     /// </summary>
241     private void AddFields()
242     {
243         foreach (var field in this.myMap.Fields)
244         {
245             var connection = new TreeWatchDatabase();
246             var query = new DBQuery<Field>(connection);
247             query.GetChildren(field);
248             if (field.Blocks.Count != 0)
249             {
250                 foreach (var block in field.Blocks)
251                 {
252                     if (block.BoundingCoordinates.Count != 0 &&
253                         block.BoundingCoordinates.Count >= 3)
254                         this.Map.AddPolygon(GetPolygon(
255
256                         < FieldMapRenderer.ConvertCoordinates(block.BoundingCoordinates),
257
258                         < block.TreeType.ColorProp.ToAndroid()));
259                     }
260
261                     if (field.BoundingCoordinates.Count != 0 &&
262                         field.BoundingCoordinates.Count >= 3)
263                     {
264                         this.Map.AddPolygon(GetPolygon(
265
266                         < FieldMapRenderer.ConvertCoordinates(field.BoundingCoordinates),
267                         this.myMap.OverLayColor.ToAndroid(),
268                         this.myMap.BoundaryColor.ToAndroid());
269                     }
270
271         /// <summary>
272         /// Maps the clicked.
273         /// </summary>
274         /// <param name="sender">Sender object.</param>
275         /// <param name="e">E event.</param>
276         private void MapClicked(object sender,
277                         GoogleMap.MapClickEventArgs e)

```

```
277     {
278         FieldHelper.Instance.MapTappedEvent(new
279             Position(e.Point.Latitude, e.Point.Longitude),
280             this.Map.CameraPosition.Zoom);
281     }
```

39 UITests/Tests.cs

```
22  namespace TreeWatch.UITests
23  {
24      using System.Linq;
25      using NUnit.Framework;
26      using Xamarin.UITest;
27      using Xamarin.UITest.Queries;
28
29      /// <summary>
30      /// Tests class.
31      /// </summary>
32      [TestFixture(Platform.Android)]
33      [TestFixture(Platform.iOS)]
34      public class Tests
35      {
36          /// <summary>
37          /// The app.
38          /// </summary>
39          private IApp app;
40
41          /// <summary>
42          /// The platform.
43          /// </summary>
44          private Platform platform;
45
46          /// <summary>
47          /// Initializes a new instance of the <see
48          ↵      cref="TreeWatch.UITests.Tests"/> class.
49          /// </summary>
50          /// <param name="platform">Platform object.</param>
51          public Tests(Platform platform)
52          {
53              this.platform = platform;
54          }
55
56          /// <summary>
57          /// Befores the each test.
58          /// </summary>
59          [SetUp]
60          public void BeforeEachTest()
61          {
62              this.app = AppInitializer.StartApp(this.platform);
63          }
64          /// <summary>
```

```

65  /// Maps are displayed.
66  /// </summary>
67  [Test]
68  public void MapDisplayed()
69  {
70      this.app.Tap("Map");
71      AppResult[] results = this.app.WaitForElement(c =>
72          c.Marked("MapView"));
73      this.app.Screenshot("Map screen");
74      Assert.IsTrue(results.Any());
75  }
76
77  /// <summary>
78  /// MapMenu is displayed.
79  /// </summary>
80  [Test]
81  public void MapMenuDisplayed()
82  {
83      this.app.Tap("Map");
84      this.app.WaitForElement(c => c.Marked("MapView"));
85
86      this.app.Tap("MapMenuView");
87  }
88
89  /// <summary>
90  /// Histories are displayed.
91  /// </summary>
92  [Test]
93  public void HistoryDisplayed()
94  {
95      this.app.Tap("History");
96      AppResult[] results = this.app.WaitForElement(c =>
97          c.Marked("HistoryView"));
98      this.app.Screenshot("History screen");
99      Assert.IsTrue(results.Any());
100 }
101
102  /// <summary>
103  /// Todos are displayed.
104  /// </summary>
105  [Test]
106  public void TodoDisplayed()
107  {
108      this.app.Tap("ToDo");
109      AppResult[] results = this.app.WaitForElement(c =>
110          c.Marked("TodoView"));
111      this.app.Screenshot("Todo screen");

```

```

109             Assert.IsTrue(results.Any());
110         }
111
112         /// <summary>
113         /// Settings are displayed.
114         /// </summary>
115         [Test]
116         public void SettingsDisplayed()
117         {
118             if (this.platform == Platform.Android)
119             {
120                 this.app.Tap("History"); // Workaround for Android to
121                 ← make Settingstab visible
122             }
123
124             this.app.Tap("Settings");
125             AppResult[] results = this.app.WaitForElement(c =>
126                 ← c.Marked("SettingsView"));
127             this.app.Screenshot("Settings screen");
128             Assert.IsTrue(results.Any());
129
130             /// <summary>
131             /// Notes are displayed.
132             /// </summary>
133             [Test]
134             public void NoteDisplayed()
135             {
136                 this.app.Tap("Note");
137                 AppResult[] results = this.app.WaitForElement(c =>
138                     ← c.Marked("NoteView"));
139                 this.app.Screenshot("Note screen");
140                 Assert.IsTrue(results.Any());
141
142             /// <summary>
143             /// The master detail menu is displayed.
144             /// </summary>
145             [Test]
146             public void MapMasterDetailMenuIsDisplayed()
147             {
148                 this.app.Tap("Map");
149                 AppResult[] results = this.app.WaitForElement(c =>
150                     ← c.Marked("MapMasterDetailPage"));
151                 this.app.Screenshot("MapMasterDetailPage screen");
152                 this.app.Tap("Menu");
153                 Assert.IsTrue(results.Any());

```

152 }

153 }

154 }

40 TreeWatch.Tests/GeoHelperTest.cs

```
22 // LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
23 // ARISING FROM,
24 // OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
25 // DEALINGS IN
26 #endregion
27 namespace TreeWatch.Tests
28 {
29     using System.Collections.Generic;
30     using NUnit.Framework;
31
32     /// <summary>
33     /// Geo helper test.
34     /// </summary>
35     [TestFixture]
36     public class GeoHelperTest
37     {
38         /// <summary>
39         /// Tests the position is within rectangle.
40         /// </summary>
41         [Test]
42         public void TestPositionIsWithinRectangle()
43         {
44             var fieldcoords = new List<Position>();
45
46             fieldcoords.Add(new Position(51.39202, 6.04745));
47             fieldcoords.Add(new Position(51.39202, 6.05116));
48             fieldcoords.Add(new Position(51.38972, 6.05116));
49             fieldcoords.Add(new Position(51.38972, 6.04745));
50
51             var posInsideOne = new Position(51.39143, 6.04817);
52             var posInsideTwo = new Position(51.39024, 6.04982);
53             var posOutsideOne = new Position(51.39220, 6.04683);
54             var posOutsideTwo = new Position(51.38949, 6.05168);
55
56             Assert.IsTrue(GeoHelper.IsInsideCoords(fieldcoords,
57             posInsideOne));
58             Assert.IsTrue(GeoHelper.IsInsideCoords(fieldcoords,
59             posInsideTwo));
60             Assert.IsFalse(GeoHelper.IsInsideCoords(fieldcoords,
61             posOutsideOne));
62             Assert.IsFalse(GeoHelper.IsInsideCoords(fieldcoords,
63             posOutsideTwo));
64         }
65     }
66 }
```



```

94     {
95         var positionOne = new Position(51.39082462, 6.050752777);
96         var positionTwo = new Position(51.3904837, 6.04767631);
97
98
99     ↳ Assert.AreEqual(GeoHelper.DistanceInMeters(positionOne.Latitude,
100         positionOne.Longitude,
101             positionTwo.Latitude, positionTwo.Longitude), 216.8);
102
103     ↳ Assert.AreEqual(GeoHelper.DistanceInMeters(positionTwo.Latitude,
104         positionTwo.Longitude,
105             positionOne.Latitude, positionOne.Longitude), 216.8);
106
107     ↳ Assert.AreEqual(GeoHelper.DistanceInMeters(positionOne.Latitude,
108         positionOne.Longitude,
109             positionTwo.Latitude, positionTwo.Longitude),
110                 GeoHelper.DistanceInMeters(positionTwo.Latitude,
111                     positionTwo.Longitude,
112                         positionOne.Latitude, positionOne.Longitude));
113
114     /// <summary>
115     /// Tests conversion to radians.
116     /// </summary>
117     [Test]
118     public void TestToRadians()
119     {
120         Assert.AreEqual(Math.Round(GeoHelper.ToRadians(10.0), 10),
121             1.5707963268);
122         Assert.AreEqual(Math.Round(GeoHelper.ToRadians(180.0), 10),
123             3.1415926536);
124         Assert.AreEqual(Math.Round(GeoHelper.ToRadians(-90.0), 10),
125             -1.5707963268);
126     }
127 }
128 }
```

41 appveyor.yaml

```
1  version: 0.3.{build}
2  pull_requests: do_not_increment_build_number: true
3  branches:
4      except:
5          - website
6  os: Visual Studio 2015
7  init:
8      - cmd: choco install android-sdk
9  xamarin:
10     email: r.karoff@student.fontys.nl
11     password:
12         secure: Y2svZY8jnGPTZONLl19lKQ==
13     android: true
14     ios: true
15 before_build:
16     - cmd: nuget restore TreeWatch.sln -PackagesDirectory .\packages
17 build:
18     project: TreeWatch.sln
19     verbosity: minimal
20 test:
21     assemblies: TreeWatch.Tests.dll
```

42 TreeWatch/Utils/KMLParser.cs

```
22  namespace TreeWatch
23  {
24      using System;
25      using System.Collections.Generic;
26      using System.Globalization;
27      using System.IO;
28      using System.Linq;
29      using System.Reflection;
30      using System.Xml.Linq;
31
32      /// <summary>
33      /// Get Information from KML file.
34      /// </summary>
35      public static class KMLParser
36      {
37          /// <summary>
38          /// Generates a new unique Color.
39          /// </summary>
40          /// <returns>The tree type color.</returns>
41          /// <param name="treeTypes">Tree types.</param>
42          public static string NewTreeTypeColor(List<TreeType> treeTypes)
43          {
44              var colors = new List<string>();
45              foreach (var item in treeTypes)
46              {
47                  colors.Add(item.TreeColor);
48              }
49
50              string color;
51              do
52              {
53                  color = ColorHelper.RandomColor();
54              }
55              while (colors.Contains(color));
56
57              return color;
58          }
59
60          /// <summary>
61          /// Gets blocks from a KML file.
62          /// </summary>
63          /// <returns>The blocks.</returns>
64          /// <param name="kml">Path to the KML file.</param>
65          /// <param name="treeTypes">Existing Tree types.</param>
```

```

66     public static List<Block> GetBlocks(string kml, List<TreeType>
67     treeTypes)
68     {
69         var allTreeTypes = new List<TreeType>();
70         allTreeTypes.AddRange(treeTypes);
71
72         var xml = XDocument.Parse(kml);
73
74         var ns = xml.Root.Name.Namespace;
75
76         var blocks = xml.Descendants(ns + "Placemark");
77         var resultList = new List<Block>();
78
79         foreach (var item in blocks)
80         {
81             var resultBlock = new Block();
82             resultBlock.BoundingCoordinates = new List<Position>();
83             var treetypeName = item.Descendants(ns +
84             "SimpleData").FirstOrDefault();
85
86             var treetypeNameString = treetypeName != null ?
87             treetypeName.Value : "NOTDEFINED";
88
89             if (allTreeTypes.All(treeType => treeType.Name !=
90             treetypeNameString))
91             {
92                 var gayTreeType = new TreeType(treetypeNameString,
93                 NewTreeTypeColor(allTreeTypes));
94                 resultBlock.TreeType = gayTreeType;
95                 allTreeTypes.Add(gayTreeType);
96             }
97             else
98             {
99                 resultBlock.TreeType =
100                 allTreeTypes[allTreeTypes.IndexOf(new
101                 TreeType(treetypeNameString))];
102             }
103
104             var cords = item.Descendants(ns + "coordinates");
105
106             resultBlock.BoundingCoordinates = GetCoordinates(cords);
107
108             resultList.Add(resultBlock);
109         }
110
111         return resultList;
112     }

```

```

106
107     /// <summary>
108     /// Gets the field from a KML file.
109     /// </summary>
110     /// <returns>The field.</returns>
111     /// <param name="kml">Kml file.</param>
112     public static Field GetField(string kml)
113     {
114         var resultField = new Field();
115         resultField.BoundingCoordinates = new List<Position>();
116         var xml = XDocument.Parse(kml);
117
118         var ns = xml.Root.Name.Namespace;
119
120         var cords = xml.Descendants(ns + "coordinates");
121
122         resultField.BoundingCoordinates = GetCoordinates(cords);
123
124         return resultField;
125     }
126
127     /// <summary>
128     /// Gets the heatmap rom a KML file.
129     /// </summary>
130     /// <returns>The heatmap.</returns>
131     /// <param name="kml">Kml file.</param>
132     public static HeatMap GetHeatmap(string kml)
133     {
134         var xml = XDocument.Parse(kml);
135         var ns = xml.Root.Name.Namespace;
136
137         var points = xml.Descendants(ns + "Placemark");
138         var heatmap = new HeatMap();
139         heatmap.Points = new List<HeatmapPoint>();
140
141         foreach (var item in points)
142         {
143             var resultPoint = new HeatmapPoint();
144             var cords = item.Descendants(ns + "coordinates");
145
146             resultPoint.BoundingCoordinates = GetCoordinates(cords);
147
148             var descendants = item.Descendants(ns +
149             "SimpleData").ToList();
150             resultPoint.RowID =
151             Convert.ToDouble(descendants[1].Value, new CultureInfo("en-US"));

```

```

150             resultPoint.FID =
151     ↵     Convert.ToDouble(descendants[2].Value, new CultureInfo("en-US"));
152             resultPoint.Mean =
153     ↵     Convert.ToDouble(descendants[3].Value, new CultureInfo("en-US"));
154             resultPoint.Std =
155     ↵     Convert.ToDouble(descendants[4].Value, new CultureInfo("en-US"));
156
157             heatmap.Points.Add(resultPoint);
158         }
159
160     }
161
162     /// <summary>
163     /// Loads the file.
164     /// </summary>
165     /// <returns>The file.</returns>
166     /// <param name="resourcename">Name of the ressource.</param>
167     public static string LoadFile(string resourcename)
168     {
169         var assembly = typeof(KMLParser).GetTypeInfo().Assembly;
170         Stream stream =
171     ↵     assembly.GetManifestResourceStream("TreeWatch." + resourcename);
172         string text;
173         using (var reader = new StreamReader(stream))
174         {
175             text = reader.ReadToEnd();
176         }
177
178         return text;
179     }
180
181     /// <summary>
182     /// Gets the coordinates from XML.
183     /// </summary>
184     /// <returns>The coordinates.</returns>
185     /// <param name="coords">The list of XML elements.</param>
186     public static List<Position>
187     ↵     GetCoordinates(IEnumerable< XElement > coords)
188     {
189         var listOfCords = coords.First().Value.Trim().Split(' ');
190         var posList = new List<Position>();
191
192         foreach (var cord in listOfCords)
193         {
194             var longitude = Convert.ToDouble(cord.Split(',') [0] ,
195     ↵     new CultureInfo("en-US"));

```

```
191         var latitude = Convert.ToDouble(cord.Split(',') [1], new
192             CultureInfo("en-US"));
193             var pos = new Position(latitude, longitude);
194             posList.Add(pos);
195         }
196         return posList;
197     }
198 }
199 }
```

43 documenttemplate.tex

```
22 \usepackage[authoryear]{natbib}
23 \usepackage{enumitem}
24 %\bibliographystyle{fontysIEEtranN}
25
26 \pagestyle{fancy}
27 \fancyhf{}
28 \lhead{\includegraphics[width=0.2\linewidth]{TreewatchLogo.pdf}}
29 \rhead{DOCUMENTNAME}
30 \rfoot{Page \thepage}
31
32 % alternate row colors for all tables
33 \definecolor{lightGrey}{rgb}{0.95,0.95,0.95}
34
35 \let\oldtable\table
36 \let\endoldtable\endtable
37 \renewenvironment{table}{\rowcolors{2}{lightGrey}{}\oldtable}{\endoldtable}
38 \let\oldtabular\tabular
39 \let\endoldtabular\endtabular
40 \renewenvironment{tabular}{\rowcolors{2}{lightGrey}{}\oldtabular}{\endoldtabular}
41 \let\oldtabularx\tabularx
42 \let\endoldtabularx\endtabularx
43 \renewenvironment{tabularx}{\rowcolors{2}{lightGrey}{}\oldtabularx}{\endoldtabularx}
44
45 \let\oldlongtable\longtable
46 \let\endoldlongtable\endlongtable
47 \renewenvironment{longtable}{\rowcolors{2}{lightGrey}{}\oldlongtable}{\endoldlongtable}
48
49 \graphicspath{./././././img/}
50
51 \renewcommand{\arraystretch}{1.5}
52
53 \DeclareDocumentCommand{\newdualentry}{ O{} O{} m m m } {
54   \newglossaryentry{gls-#3}{name={#5},text={#5\glsadd{#3}},
```

```

57     description={#6},#1
58 }
59 \makeglossaries
60 \newacronym[see={[Glossary:]\{gls-#3\},#2}\{#3}\{#4}\{#5\glsadd{gls-#3}\}
61 }
62
63
64 \newlist{SMART}{description}{2}
65 \setlist[SMART]{leftmargin=12em,style=nextline}
66
67 \newlist{STAR}{description}{2}
68 \setlist[STAR]{leftmargin=8em,style=nextline}
69
70
71 \%loadglsentries{../glossary.tex}
72
73 \setmarginsrb{3 cm}{2.5 cm}{3 cm}{2.5 cm}{1 cm}{1.5 cm}{1 cm}{1.5 cm}
74
75 \%makeglossaries
76 \begin{document}
77
78 %%%%%%%%%%%%%%%Preface of the report
79
80 \pagenumbering{roman} % Roman numerals for page counter
81
82
83 % Title Page
84 \begin{titlingpage}
85     \begin{center}
86         \begin{minipage}{\linewidth}
87             \centering
88             %University logo
89             \includegraphics[width=0.3\linewidth]{FontysLogo.pdf}
90             \par
91             \vspace{3cm}
92             %Thesis title
93             {\uppercase
94                 {\Large DOCUMENTNAME\ \ AUTHOR \ 2015 Sofa GTL \ \
95                 \ TreeWatch Project \ %Change
96                 \ Document title and Name
97                 \par
98                 \vspace{3cm}}}
99                 \includegraphics[width=0.5\linewidth]{TreewatchLogo.pdf}
100                 \par
101                 \vspace{2cm}
102                 %Author's name

```



```

127
128      %Delete comments where you need it
129
130      \%printglossary[type=\acronymtype]
131      \%
132      \%printglossary
133      \%
134      \%listoffigures
135      \%addcontentsline{toc}{section}{\listfigurename}
136      \%listoftables
137      \%addcontentsline{toc}{section}{\listtablename}
138      \%pagebreak
139
140      \tableofcontents
141      \clearpage
142
143      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
144      \%
145      \%pagenumbering{arabic}
146
147      \section{Introduction}
148      %Write the Introduction with the purpose of the document HERE
149
150      %Content belongs HERE
151
152      \%
153      \%clearpage
154
155      \%bibliography{biblist}
156
157      \end{document}

```

44 TreeWatch/ViewModels/Map/MapViewModel.cs

```
22  namespace TreeWatch
23  {
24      using System;
25      using System.Collections.Generic;
26      using System.Collections.ObjectModel;
27      using System.ComponentModel;
28      using System.Linq;
29      using System.Runtime.CompilerServices;
30
31      using System.Windows.Input;
32      using Xamarin.Forms;
33
34      /// <summary>
35      /// Map view model.
36      /// </summary>
37      public class MapViewModel : INotifyPropertyChanged
38      {
39          /// <summary>
40          /// The search text.
41          /// </summary>
42          private string searchText = string.Empty;
43
44          /// <summary>
45          /// The search command.
46          /// </summary>
47          private Command searchCommand;
48
49          /// <summary>
50          /// The selected field.
51          /// </summary>
52          private Field selectedField;
53
54          /// <summary>
55          /// The selected block.
56          /// </summary>
57          private Block selectedBlock;
58
59          /// <summary>
60          /// The field helper.
61          /// </summary>
62          private FieldHelper fieldHelper;
63
64          /// <summary>
```

```

65     ///> Initializes a new instance of the <see cref="TreeWatch.MapViewModel"/> class.
66     ///> </summary>
67     public MapViewModel()
68     {
69         this.fieldHelper = FieldHelper.Instance;
70         this.fieldHelper.MapTapped += this.MapTapped;
71         this.fieldHelper.FieldSelected += this.FieldSelected;
72         this.fieldHelper.BlockSelected += this.BlockSelected;
73         this.Fields = new ObservableCollection<Field>(new
74             DBQuery<Field>(App.Database).GetAll());
75         this.selectedField = new Field("Dummy", new
76             List<Position>(), new List<Block>());
77
78         foreach (Field f in this.Fields)
79         {
80             var whc =
81                 GeoHelper.CalculateBoundingBox(f.BoundingCoordinates);
82             double rad = (whc.WidthInMeters > whc.HeightInMeters) ?
83                 whc.WidthInMeters : whc.HeightInMeters;
84             rad *= 0.5;
85             CrossGeofence.Current.StartMonitoring(new
86                 GeofenceCircularRegion(f.Name, whc.Center.Latitude,
87                 whc.Center.Longitude, rad)
88             {
89                 NotifyOnStay = false,
90                 StayedInThresholdDuration =
91                     TimeSpan.FromMinutes(10)
92             });
93         }
94
95         ///> <summary>
96         ///> Occurs when property changed.
97         ///> </summary>
98         public event PropertyChangedEventHandler PropertyChanged;
99
100        ///> <summary>
101        ///> Gets or sets the selected field.
102        ///> </summary>
103        ///> <value>The selected field.</value>
104        public Field SelectedField
105        {
106            get
107            {
108                return this.selectedField;
109            }
110        }

```

```

104
105         set
106     {
107         if (value != null &&
108             !value.Name.Equals(this.selectedField.Name))
109         {
110             this.selectedField = value;
111             this.SearchText = string.Empty;
112         }
113     }
114 }
115
116     /// <summary>
117     /// Gets or sets the selected block.
118     /// </summary>
119     /// <value>The selected block.</value>
120     public Block SelectedBlock
121     {
122         get
123     {
124         return this.selectedBlock;
125     }
126
127         set
128     {
129         if (value != null && value.ID != this.selectedBlock.ID)
130         {
131             this.selectedBlock = value;
132
133             this.fieldHelper.BlockSelectedEvent(this.selectedBlock);
134         }
135     }
136
137     /// <summary>
138     /// Gets or sets the search text.
139     /// </summary>
140     /// <value>The search text.</value>
141     public string SearchText
142     {
143         get
144     {
145         return this.searchText;
146     }
147

```

```

148     set
149     {
150         if (this.searchText != value)
151         {
152             this.searchText = value ?? string.Empty;
153             this.OnPropertyChanged("SearchText");
154
155             if (this.SearchCommand.CanExecute(null))
156             {
157                 this.SearchCommand.Execute(null);
158             }
159         }
160     }
161 }
162
163     /// <summary>
164     /// Gets the search command.
165     /// </summary>
166     /// <value>The search command.</value>
167     public ICommand SearchCommand
168     {
169         get
170         {
171             this.searchCommand = this.searchCommand ?? new
172             Command(this.DoSearchCommand, MapViewModel.CanExecuteSearchCommand);
173
174             return this.searchCommand;
175         }
176     }
177
178     /// <summary>
179     /// Gets the filtered fields.
180     /// </summary>
181     /// <value>The filtered fields.</value>
182     public ObservableCollection<Field> FilteredFields
183     {
184         get
185         {
186             var filteredFields = new ObservableCollection<Field>();
187
188             if (this.Fields != null)
189             {
190                 List<Field> entities = this.Fields.Where(x =>
191                     x.Name.ToLower().Contains(this.searchText.ToLower())).ToList();
192
193                 if (entities != null && entities.Any())
194                 {

```

```

192                     filteredFields = new
193             ObservableCollection<Field>(entities);
194         }
195
196         return filteredFields;
197     }
198 }
199
200 /// <summary>
201 /// Gets the fields.
202 /// </summary>
203 /// <value>The fields.</value>
204 public ObservableCollection<Field> Fields
205 {
206     get;
207     private set;
208 }
209
210 /// <summary>
211 /// Navigates to field.
212 /// </summary>
213 /// <param name="field">The field which information should be
214     shown.</param>
215     public static void NavigateToField(Field field)
216     {
217         var navigationPage =
218             (NavigationPage)Application.Current.MainPage;
219
220         var informationViewModel = new InformationViewModel(field);
221         navigationPage.PushAsync(new
222             FieldInformationContentPage(informationViewModel));
223     }
224
225 /// <summary>
226 /// Gets the current device position.
227 /// </summary>
228 /// <returns>The current device position.</returns>
229 public static Position GetCurrentDevicePosition()
230 {
231     // Todo: make this not static
232     var pos = new Position();
233     pos.Latitude = 51.39202;
234     pos.Longitude = 6.04745;
235
236     return pos;
237 }

```

```

235
236     ///<summary>
237     /// Checks the field clicked.
238     ///</summary>
239     ///<returns>The field clicked.</returns>
240     ///<param name="touchPos">Touch position.</param>
241     public Field CheckFieldClicked(Position touchPos)
242     {
243         foreach (Field field in this.Fields)
244         {
245             if (GeoHelper.IsInsideCoords(field.BoundingCoordinates,
246             ↵ touchPos))
247             {
248                 return field;
249             }
250         }
251         return null;
252     }
253
254     ///<summary>
255     /// Checks the block clicked.
256     ///</summary>
257     ///<returns>The block clicked.</returns>
258     ///<param name="touchPos">Touch position.</param>
259     public Block CheckBlockClicked(Position touchPos)
260     {
261         foreach (Block block in this.SelectedField.Blocks)
262         {
263             if (GeoHelper.IsInsideCoords(block.BoundingCoordinates,
264             ↵ touchPos))
265             {
266                 return block;
267             }
268         }
269         return null;
270     }
271
272     ///<summary>
273     /// Raises the property changed event.
274     ///</summary>
275     ///<param name="propertyName">Property name.</param>
276     protected virtual void OnPropertyChanged([CallerMemberName]
277     ↵ string propertyName = null)
278     {
279         PropertyChangedEventHandler handler = this.PropertyChanged;

```

```

279             if (handler != null)
280             {
281                 handler(this, new
282                 PropertyChangedEventArgs(propertyName));
283             }
284         }
285
286         /// <summary>
287         /// Determines if can execute search command.
288         /// </summary>
289         /// <returns><c>true</c> if can execute search command;
290         /// otherwise, <c>false</c>.</returns>
291         private static bool CanExecuteSearchCommand()
292         {
293             return true;
294         }
295
296         /// <summary>
297         /// Maps the tapped.
298         /// </summary>
299         /// <param name="sender">The sender of the tapped event inside
300         /// the map.</param>
301         /// <param name="e">The arguments of the map tapped
302         /// event.</param>
303         private void MapTapped(object sender, MapTappedEventArgs e)
304         {
305             var tappedField = this.CheckFieldClicked(e.Position);
306             if (tappedField != null)
307             {
308                 this.SelectedField = tappedField;
309             }
310
311             if (e.Zoomlevel > 15)
312             {
313                 var tappedBlock = this.CheckBlockClicked(e.Position);
314                 if (tappedBlock != null)
315                 {
316                     this.SelectedBlock = tappedBlock;
317                     var navigationPage =
318                     (NavigationView)Application.Current.MainPage;
319
320                     var informationViewModel = new
321                     InformationViewModel(this.SelectedField, this.SelectedBlock);
322                     navigationPage.PushAsync(new
323                     BlockInformationContentPage(informationViewModel));
324                 }
325             }
326         }

```

```

319     }
320
321     ///<summary>
322     /// Fields the selected.
323     ///</summary>
324     ///<param name="sender">The sender of the selected
325     field.</param>
326     ///<param name="e">The arguments of the selected field
327     event.</param>
328     private void FieldSelected(object sender,
329     FieldSelectedEventArgs e)
330     {
331         this.SelectedField = e.Field;
332     }
333
334     ///<summary>
335     /// Blocks the selected.
336     ///</summary>
337     ///<param name="sender">The sender of the selection.</param>
338     ///<param name="e">The arguments of the selection
339     event.</param>
340     private void BlockSelected(object sender,
341     BlockSelectedEventArgs e)
342     {
343         this.SelectedBlock = e.Block;
344     }
345
346     ///<summary>
347     /// Dos the search command.
348     ///</summary>
349     private void DoSearchCommand()
350     {
351         // Refresh the list, which will automatically apply the
352         search text
353         this.OnPropertyChanged("FilteredFields");
354     }
355 }

```