

# Designing an application for focussed surveys

Sander Bloemen

University of Twente

PO Box 217, 7500 AE

Enschede

the Netherlands

s.p.r.bloemen@student.utwente.nl

## ABSTRACT

As companies grow bigger and direct communication between management and employees at the work floor reduces, it gets more difficult for the management to get an accurate overview of which regions might need some extra attention. Opinion polling amongst employees has been proven useful in this kind of field, but harder to achieve when the company grows bigger. Especially in companies which are divided in various regions/workplaces and where the employees feel more connected with the place they are working than the company they actually work for, which is often the case in cleaning companies such as Asito. This research will be focus on the development of a new application which is capable of doing focussed surveys. The development will be done using a design science methodology.

## Keywords

Survey, Mobile application, Data gathering, Xamarin

## 1. INTRODUCTION

Staff wellbeing has always been proven important for almost all disciplines of business. Especially if the main asset of the company are people, which is the case for a cleaning company. The 3 most common aspects of staff wellbeing is job satisfaction, job stress and job motivation. [1] This research will focus mainly on job satisfaction, while this is most frequently measured with surveys. [1] But the less employees feel connected to the company they work for, the lower the change they will actually fill in any survey their employer gives them. This is also a huge problem for Asito. It is known that the employees feel much more connected to the object they are working for, than they feel connected to Asito. E.g. Cleaning staff who always work at the University of Twente, feel more loyal to the university than they feel to Asito. This phenomenon was mentioned by several headquarter employees of Asito during interviews (See Table 1) and is mainly caused by the fact that the employees move along with an object when it shifts to another cleaning organization.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

29<sup>th</sup> Twente Student Conference on IT, July. 6<sup>th</sup>, 2018, Enschede, The Netherlands. Copyright 2018, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.

In this research paper if referenced to an object, this means a working place where several cleaners work their shifts. E.g the Horst tower at the University of Twente is considered an object and the Carré together with the Nanolab are considered one object (25 and 20 cleaners per object respectively). In this example the client is the University of Twente. The cleaning staff

are the employees who work for Asito, but feel like they work for the University of Twente.

## 2. PROBLEM STATEMENT

Currently the response rate of Asito's employee surveys (in 2018) is below the average of 33%[2]. This response rate includes the responses of all of Asito's employees, not only the cleaning staff. Asito would like this rate to be increased, especially under the cleaning staff. To do this, Asito would like the surveys to be more narrowed, so that less, but much more valuable questions can be asked at specific workplaces instead of the whole company.

During several interviews with employees of Asito's headquarter (See Table 1), their main problem became quite clear: Asito's current opinion polling is done by an external company, which receives a below average response rate as stated above. By law, this opinion poll needs to be anonymized in such a way that the answers cannot be traced back to individuals. For the law to have an 'anonymous' survey, there need to be at least 10 responses before 1 response is being considered 'anonymous'. For Asito this anonymisation means that the value/usefulness of the data gathered from the survey has dropped significantly. This is due to the fact that the responses cannot be traced back to specific objects. Without the ability to narrow the responses down to an object level, it becomes very hard to conduct knowledge from this data. A good example for this would be the following: The survey data showed that quite a few cleaners indicated that they did not have the sufficient materials to do their job efficiently. However due to the anonymity Asito does not have the insight which objects had the highest "insufficient materials" score and just replacing the materials of all objects is financially not possible.

### 2.1 Research Goals

The goal of this research paper is to find a new manner of doing a staff wellbeing survey by designing a new mobile application. This application should be capable of asking questions at object level, while at the same time persuades users to keep using this application.

In order to achieve the goals stated in the previous section, the following research questions will be addressed:

1. What should a survey application for Asito look like?
2. How can the data gathered be anonymized but still be focussed (more) narrow than regions?
3. How do we get employees to fill in the survey? or
4. Which functionalities could be added in the future to get more employees to use the application or make it indispensable?

### 2.2 Known Difficulties

Getting the employees to use the application is expected to have to most problems, due to the target population of the application. After some conversations I have had with some of the headquarter and on side employees of Asito, the following difficulties with the target population came up:

1. As mentioned before, Asito's cleaning personnel feels little connection with Asito.
2. A substantial number of Asito's cleaning personnel does not speak the native (Dutch) language.
3. A small part of the cleaning personnel are functionally illiterate or even fully illiterate.

These are important difficult properties of the target population which need to be taken into account the whole process.

## 2.3 Methods

This research by design will primarily be executed according Wieringa's Design Science Methodology. [3]

The two parts of design science, design and investigation, correspond to two kinds of research problems in design science, namely, design problems and knowledge questions. Design problems call for a change in the real world and require an analysis of actual or hypothetical stakeholder goals, which will be further allocated in the next section. Knowledge questions, by contrast, do not call for a change in the world, but ask for knowledge about the world as it is. [3] The engineering cycle is a well know concept in the design science world. The main idea of this cycle is that developers test every artifact before going on to the next artifact. (See Figure 1) In other words, the software engineer will repeat the whole engineering cycle for every new or changed functionality/appearance.

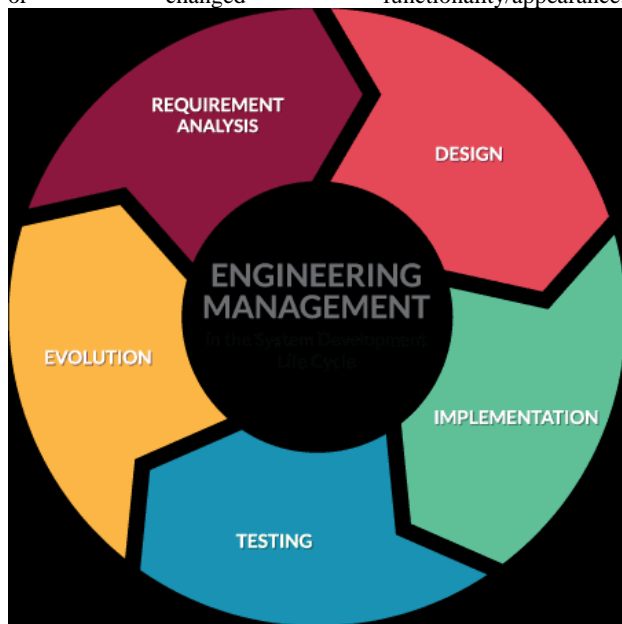


Figure 1: Engineering Cycle

Source: <https://online.uark.edu/programs/master-science-engineering-management.php>

Table 1. Interviewees Asito

Name	Function	Location	Date
------	----------	----------	------

Alberts, Mert	CIO	Asito Headquarters	17-05-2018
Belshof, Jose	Projectleider	Asito Office UT(Spiegel)	21-06-2018
Meulekamp, Mark	Application Architect .NET	Asito Headquarters	17-05-2018
Tibben, Roy	Business development manager	Asito Headquarters	18-05-2018
Tibben, Roy	Business development manager	Asito Headquarters	25-05-2018

## 3. RELATED WORK

In Mason et al.[4] is explained how this survey information can be used to derive new strategies to improve customer and staff satisfaction/wellbeing. It illustrates how survey data can be used to generate new business cases. It is not particularly used in this research, however it could add use cases after a survey is done.

Jeanes et al.[5] explains the importance of staff wellbeing and how only asking your company's staff, will already improve the employees feeling of involvement and adherence.

The paper: 'Towards a decision-making structure for selecting a research design in empirical software engineering' [6] is very useful for this research, as it describes several methods for software design projects and for which reasons a method should be chosen. The same is done by Baker N, but is focussed on socio-technical systems.[7]

Baker N [8] describes how performance measurement can be used to give feedback to employees and how this can help the perform better in the future. The paper is a great example of an use case derived from survey data

## 4. PROBLEM INVESTIGATION

The main task of the new application will be useful data gathering. Useful means in the context of this paper: back-traceable to specific objects. It is important that the data will be stored in such a way that it is anonymous, while at the same time back-traceable to the object. When searching through the literature and the internet, the most common way of doing staff surveys is to send a list of questions to every employee. For this research I would like to try a whole different approach of doing a survey, due to the previously mentioned target population difficulties and after several (informal) interviews with employees in all kind of departments within Asito. A few major points almost always came up during those interviews.

- Employees who have nothing to do with the data that is gathered are very suspicious at the anonymity of the survey.
- After a survey is done, no feedback is given about the outcomes of the survey.

The first point seems quite understandable because of the fact that at the start of the current survey, full personal information is asked. Employees have to trust this external company to anonymize their responses, before submitting it back to Asito.

The second point should be taken into account in future research. Due to time constraints this research will focus on acquiring data first. The paper: “Employee feedback technologies in the human performance system” could help in the development of a feedback system. [7]

## 5. TREATMENT DESIGN AND VALIDATION

Instead of asking a whole list of questions, this new approach will only ask one question each day. Asito’s cleaning employees generally start their shift at a designated area inside of the objects they are working on, further referred to as “object foyer”. This seemed like a good opportunity to solve the issue of anonymity suspiciousness. Instead of asking employees for their full personal information, the app will only ask for which object they work.

### 5.1 The Application

The application itself will be made as simple as possible, so it does not need any explanation to operate it. The only data it needs is an object code as described above and an answer to the question. Acquiring the object code will be done by a QR code hanging in the object foyers. This QR code will represent the object and is then scanned by the employees when starting their shift. After the QR code is scanned, the application will send an object code to the server and a question will be sent back. The application will show this question and ask for a response. This response will then be sent back to the server. With this concept the data will not consist of any personal information of the employees, so it does not need to be anonymized afterwards. Also, due to the lack of personal information related questions, the application appears to be much anonymous than a ‘regular’ survey. When introducing the application to the staff, there will be special attention in showing the potential users the output of the application and thus proving its anonymity. This could theoretically eradicate the problem of suspiciousness towards the anonymity. The code for the application is added in the appendix. When the application is started, the opening screen will be shown. (See Figure 2) The opening screen has quite some empty space left. This is done on purpose to leave enough space for future developments, without having to change the opening screen too much. After the “SCAN” button is pressed, the scanning page will open as shown in Figure 3. After the QR code is scanned, the question page will be shown with the question associated with the scanned object on top. The desired response button can be clicked and this will lead the user to the final page.

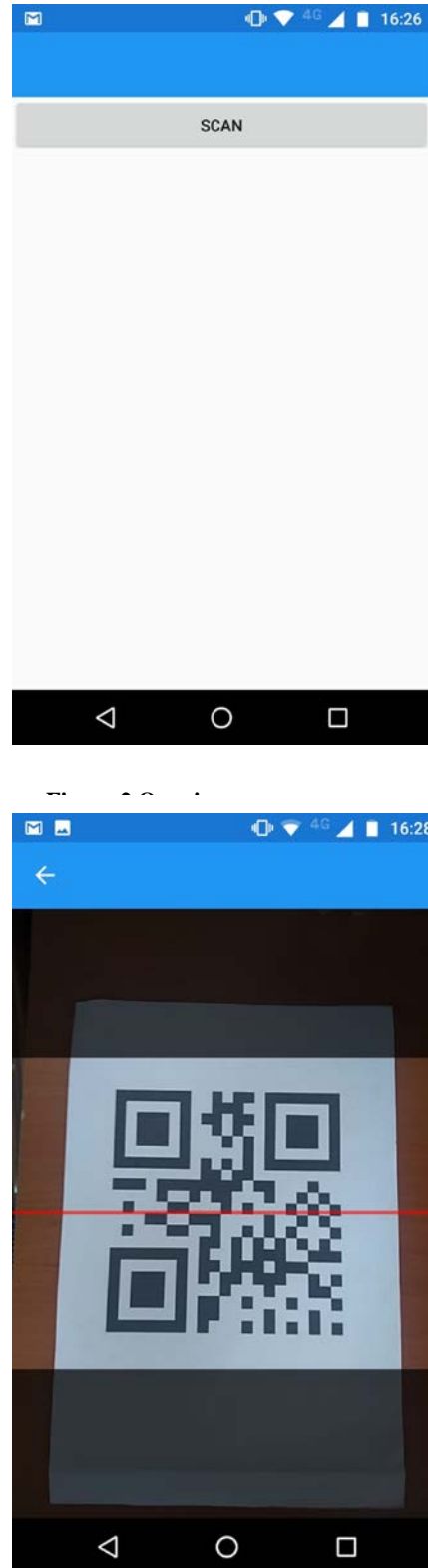


Figure 3 Scanning QR-code

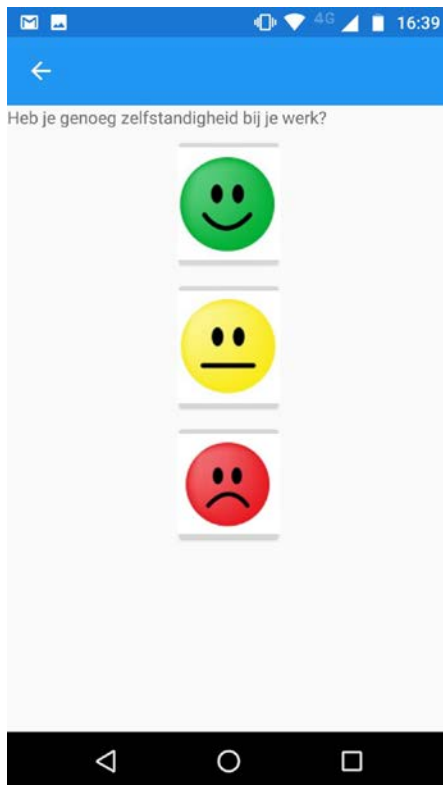


Figure 4 Question page

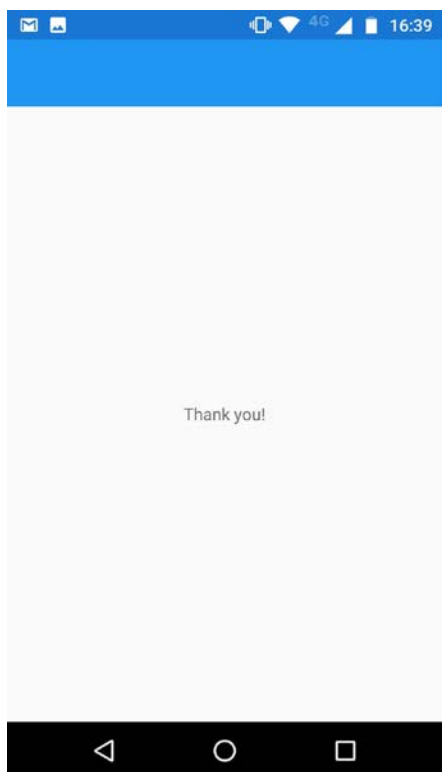


Figure 5 Final page

## 5.2 The Server

For the application to be able to function and test it in the real world, it needs to be connected to a server. This API can be very

basic as it only needs answer question requests and to keep record of the responses. These responses will be saved directly under the question into an XML file stored on the server. The code for the server and the XML output is added in the appendix.

## 6. TREATMENT IMPLEMENTATION

For the implementation of the application and server a whole variety of programming languages and frameworks are possible. During several conversations with ICT Asito employees also working on mobile application, it became clear that Asito uses Xamarin forms to implement their applications.

Xamarin is a cross-platform development tool. It helps developers to write native iOS, Android and Universal Windows apps using the C# programming language. With the use of xamarin developers can create native applications for all previously mentioned operating systems while writing the code only once. This is very time-saving and thus money-saving for Asito, as they only have to write the application once. (Table 1-3)

For the purpose of compatibility and usability for Asito, I decided to do the same, even though I had no experience with Xamarin or even C# at all.

The server is a Raspberry Pi3 running Windows 10 IoT. Because the mobile application was written in C#, I decided it might be best to write the API for the server also in C#.

The actual implementation was planned to be done by using the application on the workflow. The idea for the first test was to equip 2 object foyers on the campus of the University of Twente with QR codes. The staff working on those objects would be informed and asked of they would like to participate voluntarily. It is important that participation will be voluntarily in this test, while participating in the survey is also voluntarily. During the setup of this test, some problems occurred which forced me to change the evaluation which will be discussed in the next section.

## 7. IMPLEMENTATION EVALUATION

The actual evaluation has been proven to be a lot more difficult than assumed when developing the pilot test. Few unexpected problems occurred during the evaluation, which unfortunately led to the conclusion that a functional test could not yet be done. Firstly, due to time constraints it was not possible to release the application into the Apple iStore and Google Play store on time. Secondly, the foyer located in the basement had very bad to no Wifi and cellular reception. So eventually I had to take a step back and instead I went for an interview with the 3 project leaders of the selected test objects (1 of the test objects is a combination of 2 objects which makes 3 in total) and showed them the application/procedure. Afterwards I asked them for their input and thoughts. The following results were derived from this interview:

- Anonymity is a mayor improvement. Even though none of the project leaders had noteworthy IT knowledge, they could all confirm the anonymity of the XML output.
- The fact that the use of mobile phones during work time is prohibited should be kept in mind during future developments or should be reconsidered. The second option having less support due to the reason that in

their experience mobile phones are the number one distractor and therefore decrease efficiency.

- QR codes could be replaced with an object code. This would eliminate the constraint of having to be in the foyer to use the application. However it would also eliminate the visual 'reminder' hanging on the wall.
- Implementing some kind of reward system would probably increase the response rate, however this is just an assumption based on their experience.

## 8. FUTURE DEVELOPMENTS

In the future the application could be extended to make it much more attractive to actually use it for the cleaning staff. This should especially be in the corner of feedback. One of the ideas which would fit perfectly to the outcomes of the problem investigation is the idea of giving feedback back to the respondents through the application. This could be in the form of displaying to outcome of yesterday's question or the amount of people who responded at yesterday's question. This second option may seem odd, but is proven to have a positive effect on user participation. [9]

Another potentially valuable option would be to add some kind of "question to the headquarter" option, in which the cleaning personnel could easily ask question directly to the main office without the need to get through the bureaucracy. Question could be directly asked to someone in Asito's headquarters or show up the next day as a question to everyone.

Also mentionable is that the application could be linked to a clocking systems so Asito could get rid of manual time registration papers and at the same time adding value to the application or even making it indispensable.

## 9. DISCUSSION AND CONCLUSION

Due to the time constraint of this project a lot of ideas/additions to the application could not be implemented. It might also be a good idea for the next time to not choose for a framework/programming language I have no experience in. On the other hand would it be quite risky to build something in a programming language which the company it is made for does not use.

It is important that in future development ease of use should have high priority in the appearance of the application. Anonymity is also an important aspect as it has to be obviously anonymous. Asking for personal information at the beginning of a survey is not beneficial for anonymity and makes respondents suspicious, which prevents employees from filling out the survey.

While this research could potentially establish a whole new way of doing employee wellbeing surveys, much has yet to be tested.

Especially on the social and persuasion fields. The main advice I would like to give to Asito is that you should focus on what the users would like to have instead of what would only be nice for Asito. As long as the users like or see the application as useful, they will keep on answering questions.

## 10. ACKNOWLEDGMENTS

Special thanks to Berend Alberts from Asito/University of Twente for his excellent accompaniment and input during this research. I would also like to thank the Asito employees mentioned in this paper for their time and quick response to every question I had during the process.

## 11. REFERENCES

- [1] Zaheera KMA, Kumar CBS, Significance of an employee and team satisfaction for augmentation and successfulness of an organization, *International Journal of Mechanical Engineering and Technology* (2018;9(4) 771-779).
- [2] <https://surveyanyplace.com/average-survey-response-rate/>
- [3] Wieringa R.J, *Design Science Methodology for Information Systems and Software Engineering* (2014)
- [4] Mason CS, Chang ASF et al., Strategic Use of Employee Opinion Surveys: Using a Quasi-Linkage Approach to Model the Drivers of Organisational Effectiveness, *Australian Journal of Management* (2005;30(1) 127-143)
- [5] Jeanas A, Hall TB, Motivation and job satisfaction of cleaning staff in the NHS: A pilot study, *Journal of infection prevention* (2012;13(2) 55-64)
- [6] Song M, Song H, Methodology of user interfaces design based on Android, 2nd International Conference on Multimedia Technology, Article number 6002076, (2011;408-411)
- [7] Baxter G, Socio-technical systems: From design methods to systems engineering, *Interacting with Computers* (2011;23(1)4-17)
- [8] Baker N, Employee feedback technologies in the human performance system, *Human Resource Development International* (2010;13(4)477-485)
- [9] Ham J, Midden C, *Ambient Persuasive Technology Needs Little Cognitive Effort: The Differential Effects of Cognitive Load on Lightning Feedback Versus Factual Feedback*, Persuasive (Berlin, Heidelberg:Springer-Verlag, 2010).

## APPENDIX

### A. APPLICATION CODE

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:Asito_QR"
    x:Class="Asito_QR.MainPage">

    <StackLayout>
        <Button Text="Scan" Clicked="Button_Clicked"/>
        <Label Text="" x:Name="mycode" TextColor="Red" HorizontalOptions="Center"/>
        <ActivityIndicator x:Name="MyActivityIndicator" HeightRequest="200" WidthRequest="200" IsEnabled="True"
IsRunning="False"></ActivityIndicator>
    </StackLayout>

</ContentPage>

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;
using Xamarin.Forms;
using ZXing.Net.Mobile.Forms;

namespace Asito_QR
{
    public partial class MainPage : ContentPage
    {
        public MainPage()
        {
            InitializeComponent();
        }

        private async void Button_Clicked(Object sender, EventArgs e)
        {
            var scan = new ZXingScannerPage();
            await Navigation.PushAsync(scan);
            scan.OnScanResult += (result) =>
            {
                Device.BeginInvokeOnMainThread(async () =>
                {
                    await Navigation.PopAsync();
                    mycode.Text = "One moment please..";
                    MyActivityIndicator.IsRunning = true;
                    string url = ("http://85.150.15.63:8081/?" + result);

                    using (HttpClient client = new HttpClient())
                    {
                        using (HttpResponseMessage response = client.GetAsync(url).Result)
                        {
                            using (HttpContent content = response.Content)
                            {
                                string res = content.ReadAsStringAsync().Result;
                                mycode.Text = res;
                                await Navigation.PushAsync(new QuestionPage(res, result.ToString()));
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

    });

    };

    }
}

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="Asito_QR.QuestionPage">
    <ContentPage.Content>
        <StackLayout>
            <Label x:Name="question" Text="question"></Label>
            <Button x:Name="Happy" HorizontalOptions="Center" Clicked="Happy_Clicked" ></Button>
            <Button x:Name="Meh" HorizontalOptions="Center" Clicked="Meh_Clicked" ></Button>
            <Button x:Name="Bad" HorizontalOptions="Center" Clicked="Bad_Clicked" ></Button>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

```

```

namespace Asito_QR
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class QuestionPage : ContentPage
    {
        private readonly string target;
        public QuestionPage (String qes, String target)
        {
            InitializeComponent ();
            Happy.Image = "happy.png";
            Meh.Image = "meh.png";
            Bad.Image = "bad.png";
            question.Text = qes.Substring(0, (qes.Length-14)).Remove(0, 58);
            this.target = target;
        }
        private void Meh_Clicked(object sender, EventArgs e)
        {
            SentResponse("2", target);
            Navigation.PushAsync(new Page1());
        }
        private void Happy_Clicked(object sender, EventArgs e)
        {
            SentResponse("1", target);
            Navigation.PushAsync(new Page1());
        }

        private void Bad_Clicked(object sender, EventArgs e)
        {
            SentResponse("3", target);
            Navigation.PushAsync(new Page1());
        }

        private void SentResponse(String e, String target)
        {

```

```

string url = ("http://85.150.15.63:8081/" + target + "/" + e);

using (HttpClient client = new HttpClient())
{
    using (HttpResponseMessage response = client.GetAsync(url).Result)
    {
        using (HttpContent content = response.Content)
        {
            }
        }
    }
}
}

```

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="Asito_QR.Page1">
    <ContentPage.Content>
        <StackLayout>
            <Label Text="Thank you!"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />
        </StackLayout>
    </ContentPage.Content>
</ContentPage>

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

```

```

namespace Asito_QR
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class Page1 : ContentPage
    {
        protected override bool OnBackButtonPressed()
        {
            Navigation.PushAsync(new MainPage());
            return true;
        }
        public Page1 ()
        {
            InitializeComponent ();
            NavigationPage.SetHasBackButton(this, false);
        }
    }
}

```

## B. SERVER CODE

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net.Http;
using Windows.ApplicationModel.Background;
using Windows.Networking.Sockets;
using Windows.System.Threading;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.Storage.Streams;

```



```
using System.IO;
using System.Diagnostics;
```

// The Background Application template is documented at <http://go.microsoft.com/fwlink/?LinkID=533884&clcid=0x409>

```
namespace AsitoRPI
{
    public sealed class StartupTask : IBackgroundTask
    {
        private static BackgroundTaskDeferral _Deferral = null;

        public async void Run(IBackgroundTaskInstance taskInstance)
        {
            _Deferral = taskInstance.GetDeferral();

            var webserver = new MyWebserver();

            await ThreadPool.RunAsync(workItem =>
            {
                webserver.Start();
            });

            internal class MyWebserver
            {
                private const uint BufferSize = 8192;

                public async void Start()
                {
                    var listener = new StreamSocketListener();
                    var XmlHandler = new XmlHandler();
                    //XmlHandler.MainAsync();

                    await listener.BindServiceNameAsync("8081");

                    listener.ConnectionReceived += async (sender, args) =>
                    {
                        var request = new StringBuilder();

                        using (var input = args.Socket.InputStream)
                        {
                            var data = new byte[BufferSize];
                            Windows.Storage.Streams.IBuffer buffer = data.AsBuffer();
                            var dataRead = BufferSize;

                            while (dataRead == BufferSize)
                            {
                                await input.ReadAsync(
                                    buffer, BufferSize, InputStreamOptions.Partial);
                                request.Append(Encoding.UTF8.GetString(
                                    data, 0, data.Length));
                                dataRead = buffer.Length;
                            }
                        }

                        //string query = GetQuery(request);
                        string query = await XmlHandler.RunQuery(GetQuery(request));

                        using (var output = args.Socket.OutputStream)
                        {
                            using (var response = output.AsStreamForWrite())
                            {
                                var html = Encoding.UTF8.GetBytes(
                                    $"<html><head><title>Background Message</title></head><body>{query}</body></html>");
                                using (var bodyStream = new MemoryStream(html))
```

```

        {
            var header = $"HTTP/1.1 200 OK\r\nContent-Length: {bodyStream.Length}\r\nConnection: close\r\n\r\n";
            var headerArray = Encoding.UTF8.GetBytes(header);
            await response.WriteAsync(headerArray,
                                    0, headerArray.Length);
            await bodyStream.CopyToAsync(response);
            await response.FlushAsync();
        }
    }
};
}

private static string GetQuery(StringBuilder request)
{
    var requestLines = request.ToString().Split(' ');

    var url = requestLines.Length > 1
        ? requestLines[1] : string.Empty;

    var uri = new Uri("http://localhost" + url);
    var query = uri.Query;
    Debug.WriteLine("Query is: " + query);
    return query;
}

}
}

```

```

using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Xml;
using System.Xml.Linq;
using Windows.Storage;

```

```

namespace AsitoRPi
{
    class XmlHandler
    {
        public static async Task MainAsync()
        {
            var wfile = await ApplicationData.Current.LocalFolder.CreateFileAsync("Responses.xml");
            Stream wstream = await wfile.OpenStreamForWriteAsync();

            XDocument doc = new XDocument(new XElement("Objects"));

            doc.Root.Add(new XElement("Object",
                new XAttribute("ID", "00001"),
                new XElement("Question", "Do you like your job?"),
                new XElement("Responses", new XElement("Response", "Yes"), new XElement("Response", "No"), new
XElement("Response", "Yes"))));
            doc.Root.Add(new XElement("Object",
                new XAttribute("ID", "00002"),
                new XElement("Question", "Question 2"),
                new XElement("Responses", new XElement("Response", "Yes"), new XElement("Response", "No"))));

            doc.Save(wstream);
            wstream.Close();
        }
    }
}

```

```

    }

    public async Task<String> RunQuery(String e)
    {
        var q = e.Split("/");

        if (q.Count() == 1)
        {
            return await GetQuestionAsync(e);
        }

        if (q.Count() == 2)
        {
            String target = q[0].Remove(0, 1);
            String ans = q[1];
            WriteResponse(ans, target);
            String res = "Done";
            return res;
        }

        else
        {
            String res = "Error";
            return res;
        }
    }

    public async Task<string> GetQuestionAsync(String e)
    {
        // var wfile = await ApplicationData.Current.LocalFolder.GetFilesAsync("Responses.xml");
        // Stream wstream = await wfile.OpenStreamForWriteAsync();
        // XDocument wxmlFile = XDocument.Load(rstream);

        e = e.Remove(0, 1);
        Debug.WriteLine(e);

        try
        {
            var file = await ApplicationData.Current.LocalFolder.GetFilesAsync("Responses.xml");
            Stream rstream = await file.OpenStreamForReadAsync();
            var rxmlFile = XDocument.Load(rstream);
            rstream.Close();

            String xmlString = rxmlFile.Root.Elements("Object").Where(sel => sel.Attribute("ID").Value ==
e).FirstOrDefault().Element("Question").Value.ToString();
            return xmlString;
        }

        catch (Exception ex)
        {
            Debug.WriteLine(ex);
            String xmlString = "Not Found.";
            return xmlString;
        }
    }

    public async void WriteResponse(String res, String targ)
    {

```

```

var file = await ApplicationData.Current.LocalFolder.GetFilesAsync("Responses.xml");
Stream rstream = await file.OpenStreamForReadAsync();
var rxmlFile = XDocument.Load(rstream);
rstream.Close();
Debug.WriteLine("Response: " + res);
Debug.WriteLine("Target: " + targ);

try
{
    var add = new XElement("Response", res);
    rxmlFile.Root.Elements("Object").Where(sel => sel.Attribute("ID").Value ==
targ).FirstOrDefault().Element("Responses").Add(add);
    Stream wstream = await file.OpenStreamForWriteAsync();
    rxmlFile.Save(wstream);
    wstream.Close();
}

catch (Exception ex)
{
    Debug.WriteLine(ex);
}
}
}

```

## C. XML OUTPUT RESPONSES

```

<?xml version="1.0" encoding="utf-8"?>
<Objects>
  <Object ID="00001">
    <Question>Heb je genoeg zelfstandigheid bij je werk?</Question>
    <Responses>
      <Response>1</Response>
      <Response>1</Response>
      <Response>3</Response>
      <Response>2</Response>
      <Response>2</Response>
      <Response>1</Response>
      <Response>1</Response>
      <Response>3</Response>
      <Response>2</Response>
      <Response>1</Response>
      <Response>1</Response>
      <Response>2</Response>
      <Response>1</Response>
      <Response>3</Response>
      <Response>2</Response>
      <Response>1</Response>
      <Response>1</Response>
      <Response>1</Response>
    </Responses>
  </Object>
  <Object ID="00002">

```

```
<Question>Krijg je voldoende aandacht van je leidinggevende?</Question>
<Responses>
<Response>1</Response>
<Response>1</Response>
<Response>1</Response>
<Response>3</Response>
<Response>3</Response>
<Response>1</Response>
<Response>3</Response>
<Response>2</Response>
<Response>1</Response>
<Response>1</Response>
<Response>1</Response>
<Response>1</Response>
<Response>2</Response>
<Response>2</Response>
<Response>1</Response>
<Response>1</Response>
<Response>1</Response>
</Responses>
</Object>
</Objects>
```