

# Application Development for Mobile and Ubiquitous Computing

## Toggl Time Tracker Second Presentation

GroupNo. 9

Team: Annemarie Seidel, Antje Schubotz, Dominik Rupp



## Where has all my time gone?

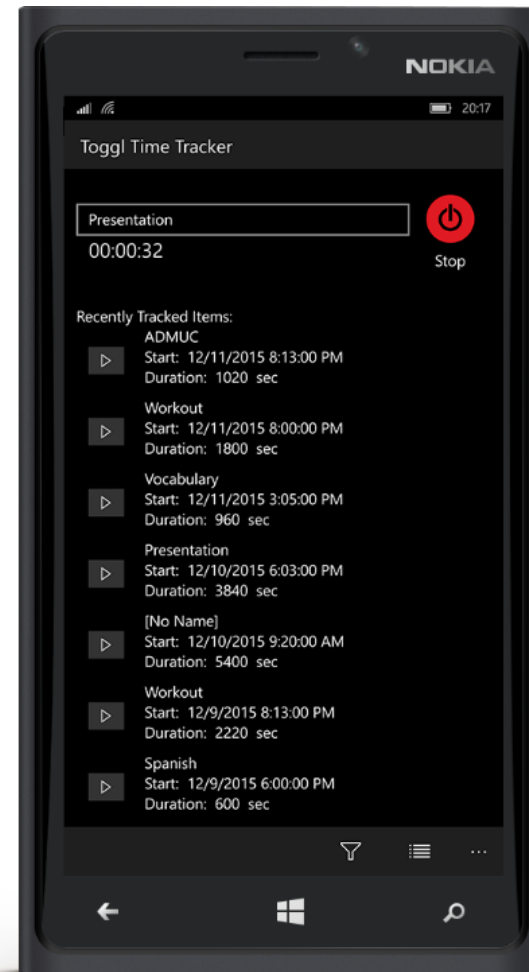
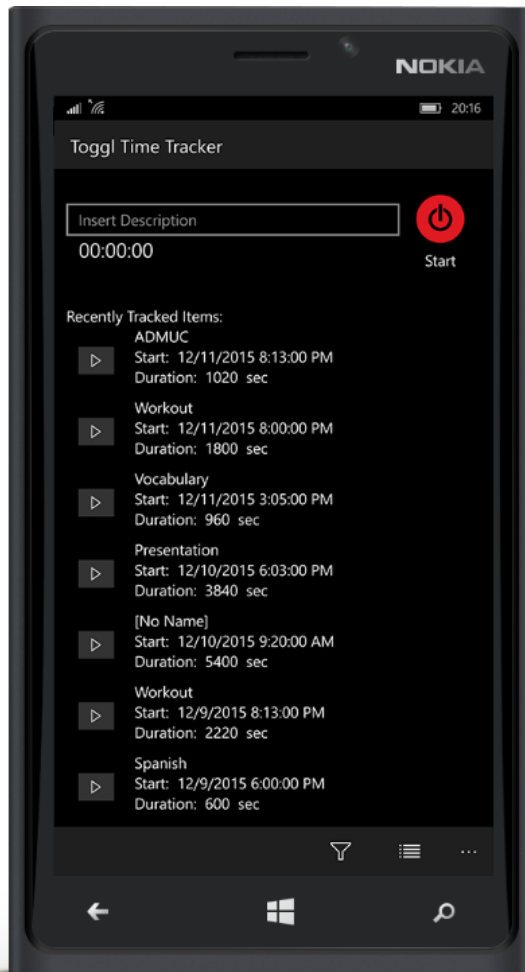
- track your time in projects or at work
- stay on top of things
- save time

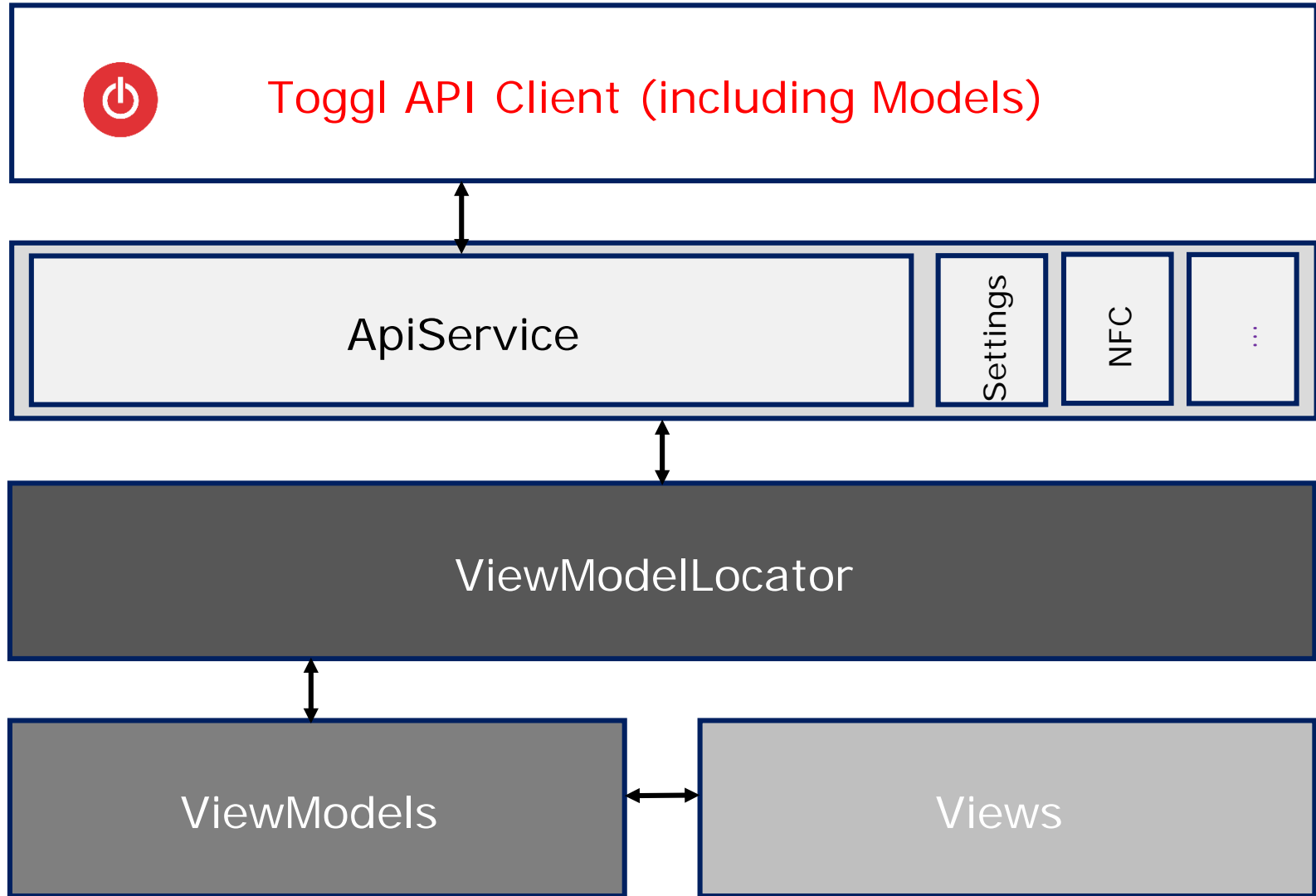


Windows 10

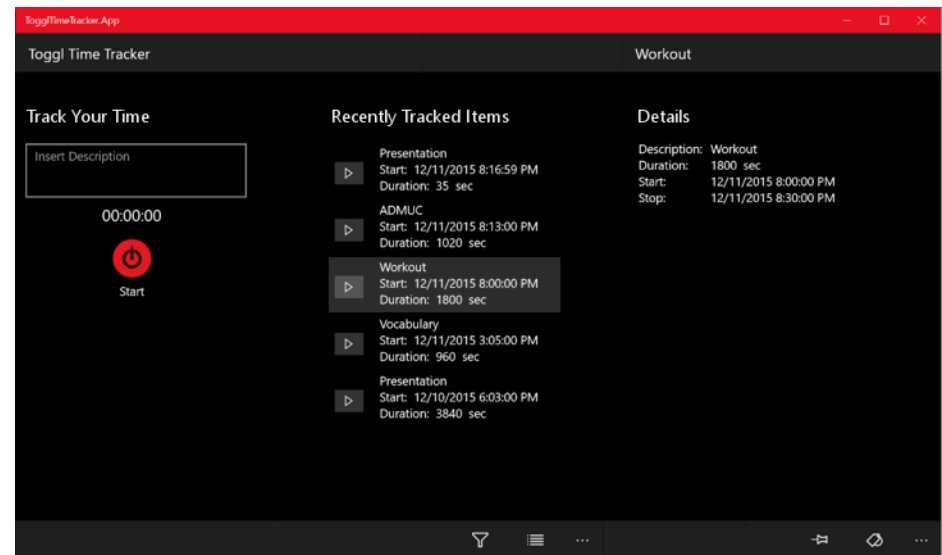
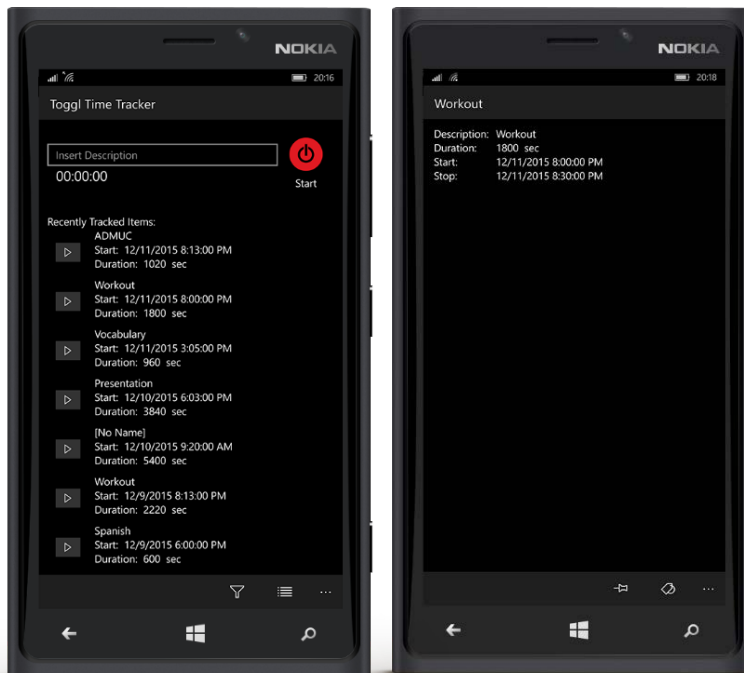






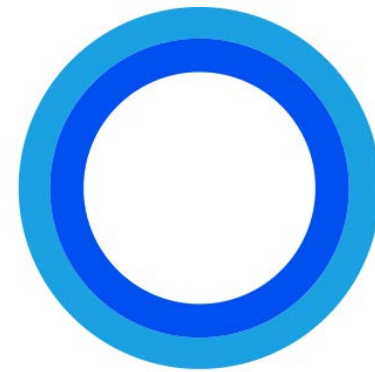


- Form factor challenge
  - mobile first
  - Column Drop/ Re-architect for larger screen sizes



- Offline challenge
  - collect data when offline
  - sync data when back online
    - o every item has id and “last updated on” timestamp
    - o always use most recent data

- Usability challenge
  - voice commands with Cortana
  - “TogglTimeTracker start tracking ‘work!’”

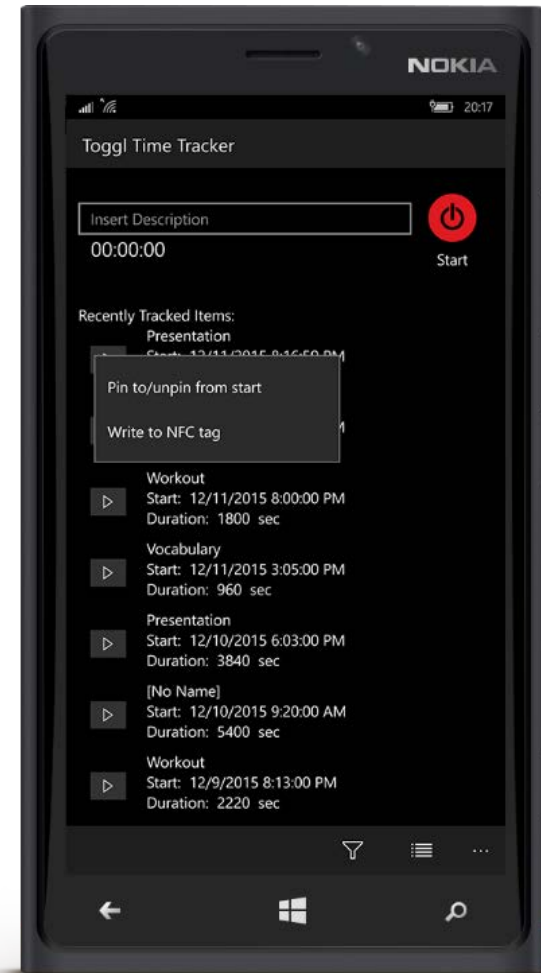


Hi. I'm Cortana.

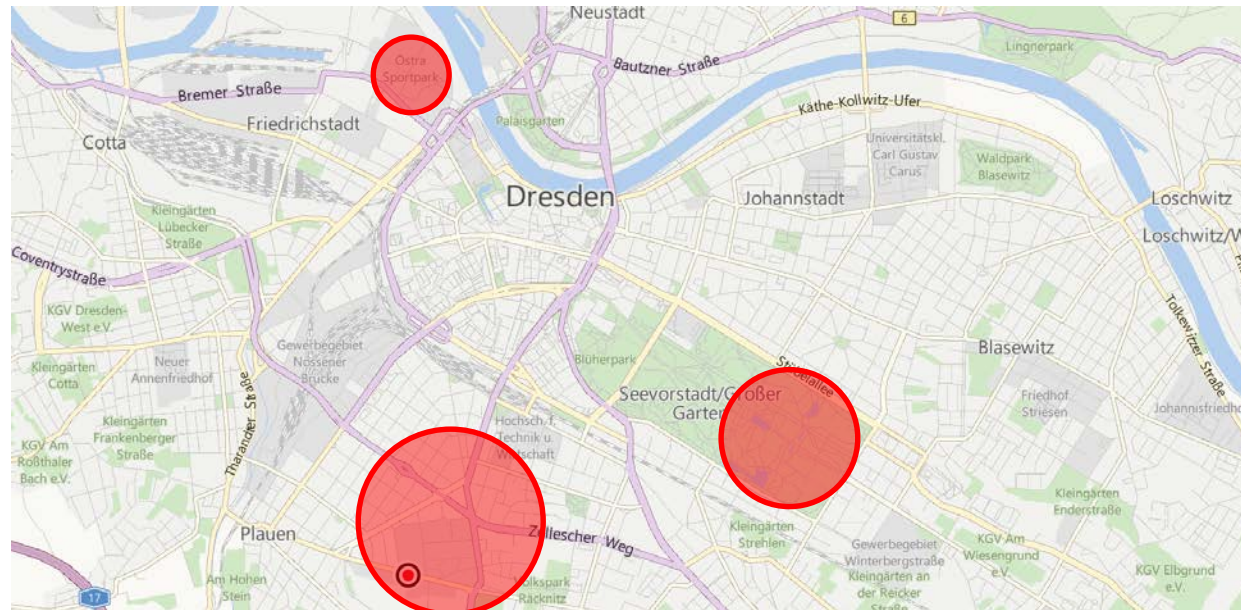




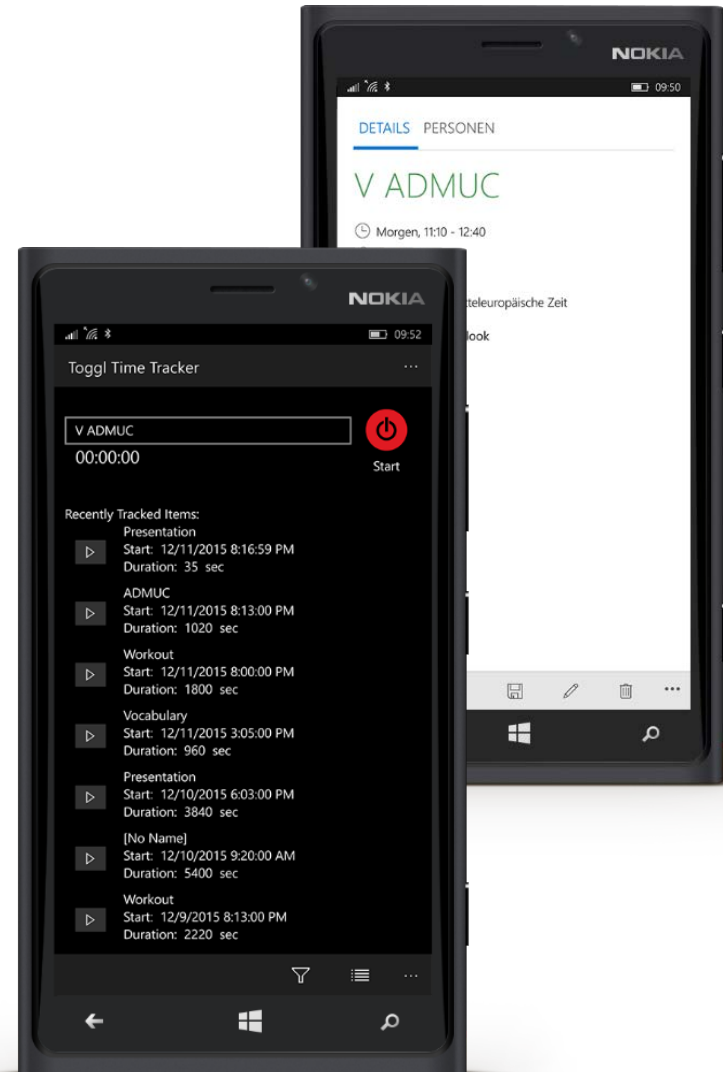
- write time entry on NFC tag
- when triggered: start tracking



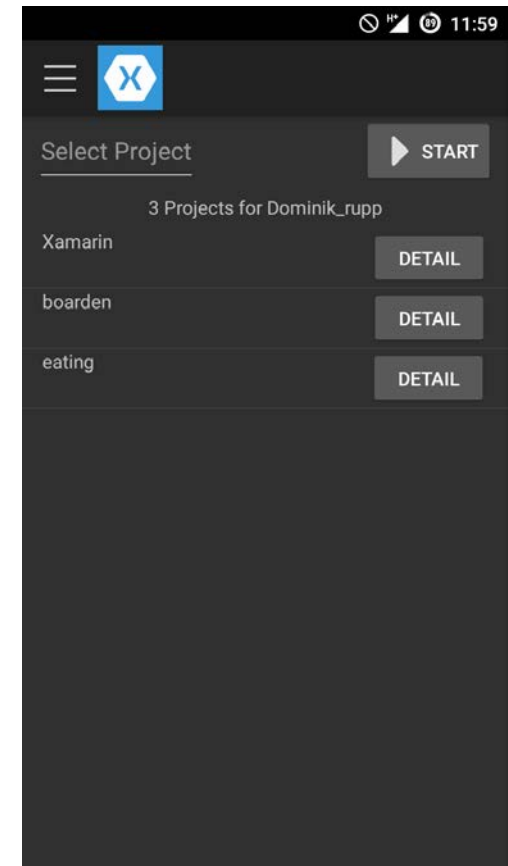
- Geofencing
  - register for change of geofence state
  - when entering in specified geofence: start tracking

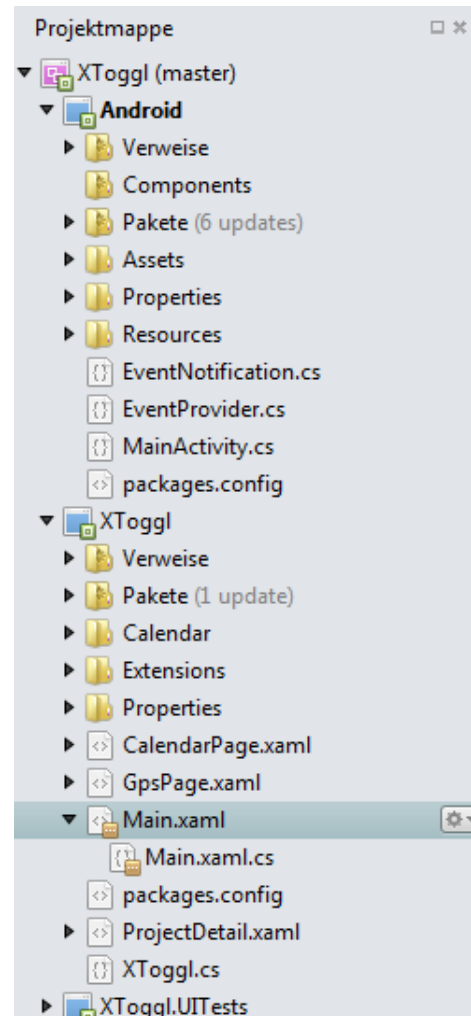


- Calendar connection
  - autofill description with name of current calendar event



- XToggl (Android):
  - Xamarin Forms
  - Dependency Injection Services
  - XLabs (Toast, Location Provider)
  - Toggl API (as DLL)
- Xamarin Studio 5.10 drawbacks:
  - Performance?
  - VCS
- Adaption and Context:
  - Calendar and GPS





```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ContentPage xmlns="http://xamarin.com/schemas/2014/forms" xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml" x:Class="XToggl.ProjectDetail">
3   <ContentPage.Content>
4     <StackLayout>
5
6       <Label x:Name="header" XAlign="Center" ></Label>
7
8       <ListView x:Name="list" VerticalOptions="FillAndExpand">
9         <ListView.ItemTemplate>
10          <DataTemplate>
11            <ViewCell>
12              <StackLayout Padding="15,0" Orientation="Vertical">
13                <Label Text="{Binding Start}" />
14                <Label Text="{Binding Duration}" />
15              </StackLayout>
16            </ViewCell>
17          </DataTemplate>
18        </ListView.ItemTemplate>
19      </ListView>
20
21    </StackLayout>
22  </ContentPage.Content>
23
24  public partial class ProjectDetail : ContentPage
25  {
26    public ProjectDetail (Project p)
27    {
28      InitializeComponent ();
29
30      var timeEntries = App.Toggl.TimeEntry.List (new TimeEntryParams { ProjectId = p.Id });
31      list.ItemsSource = timeEntries;
32      list.ItemSelected += (sender, e) => {
33        ((ListView)sender).SelectedItem = null;
34      };
35      var cnt = timeEntries.Count;
36      header.Text = cnt + " time entr" + (cnt == 1 ? "y" : "ies");
37    }
38  }
```

## ■ Generic Interface (Cross Platform Side)

```
namespace XToggl.Calendar
```

```
{  
    public interface IEventProvider {  
        void Init();  
        IList<Event> GetEventsFromNow();  
        Event GetNextEvent();  
    }  
}
```

Usage (i.e. Main.xaml.cs):

```
var eventProvider = DependencyService.Get<IEventProvider> ();  
eventProvider.Init ();  
_upcomingEvent = eventProvider.GetNextEvent ();
```

## ■ Specific Implementation (Android, iOS, WP)

```
public class EventProvider : IEventProvider  
{  
    private long calendarId;  
    public static ContentResolver ContentResolver;  
  
    public EventProvider ()  
    {  
    }  
  
    #region IEventProvider implementation  
    void IEventProvider.Init ()  
    {  
        // ...  
    }  
}
```

Registration (i.e. MainActivity):

```
DependencyService.Register<EventProvider> ();  
DependencyService.Register<EventNotification> ();  
DependencyService.Register<Geolocator> ();  
  
LoadApplication (new App ());
```

- **30.10.15**: First presentation
- Create app concept
- Implement basic set-up
- Implement functionality iteratively
- **18.12.15**: Second presentation
- Continue implementation (design and functionality)
- Testing, bugfixing and finalizing
- **29.01.15**: Final presentation



Thank you for your attention.