TECHNISCHE UNIVERSITÄT DRESDEN

Toggl on Xamarin

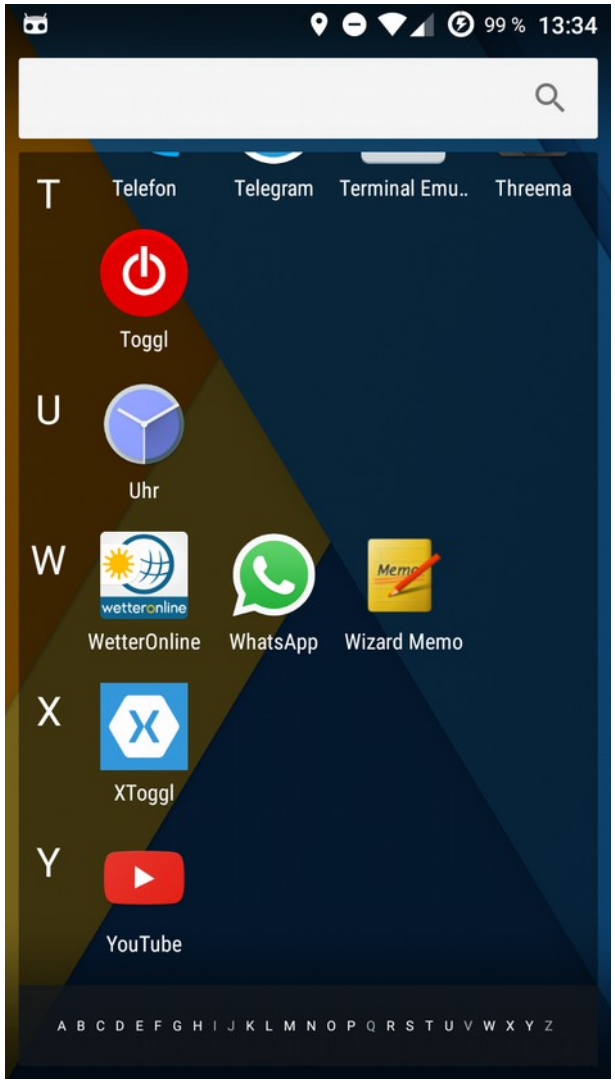
Dresden, 28. Jan 2016

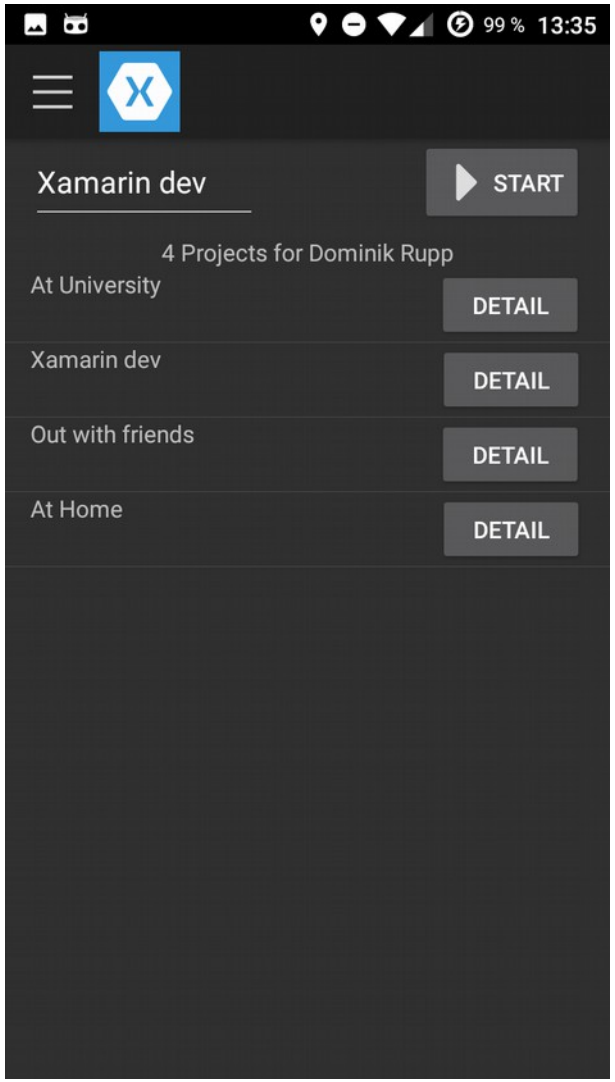
Use Cases

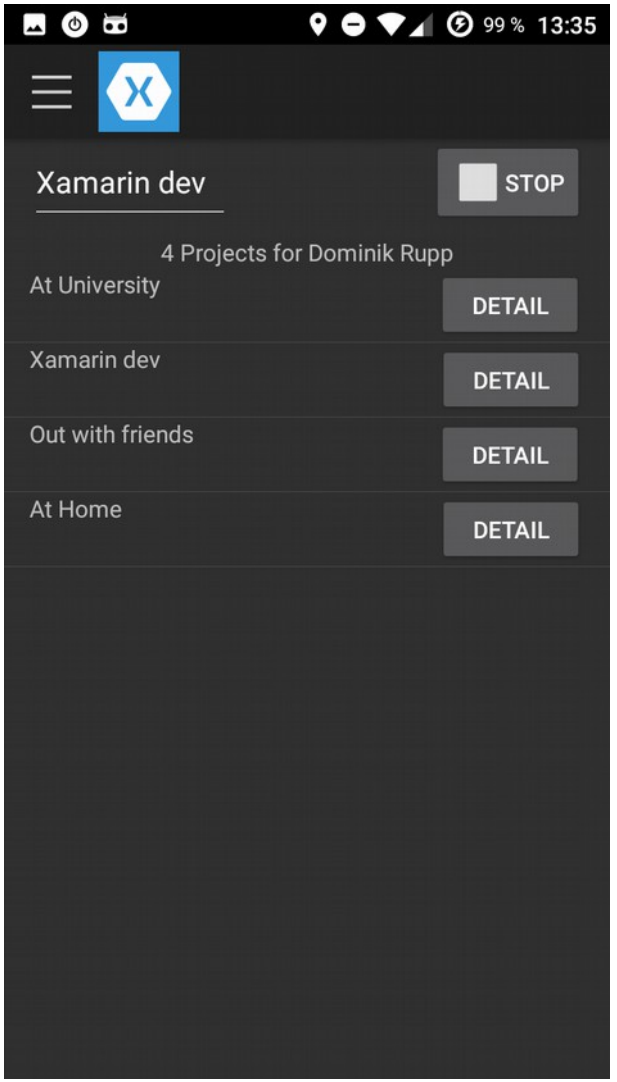
- List Projects
- Select project and see details (time entries)
- Choose project and start timer
- Stop active timer (special case: timer already running)
- Link project to gps position
- Link project to calendar event

TECHNISCHE UNIVERSITÄT DRESDEN

Screenshots

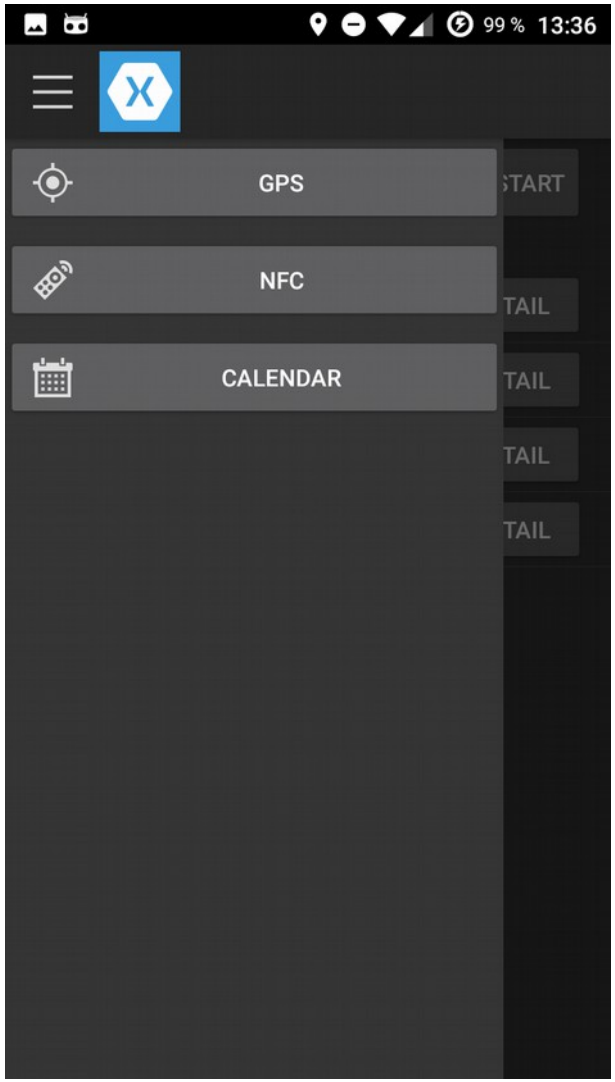


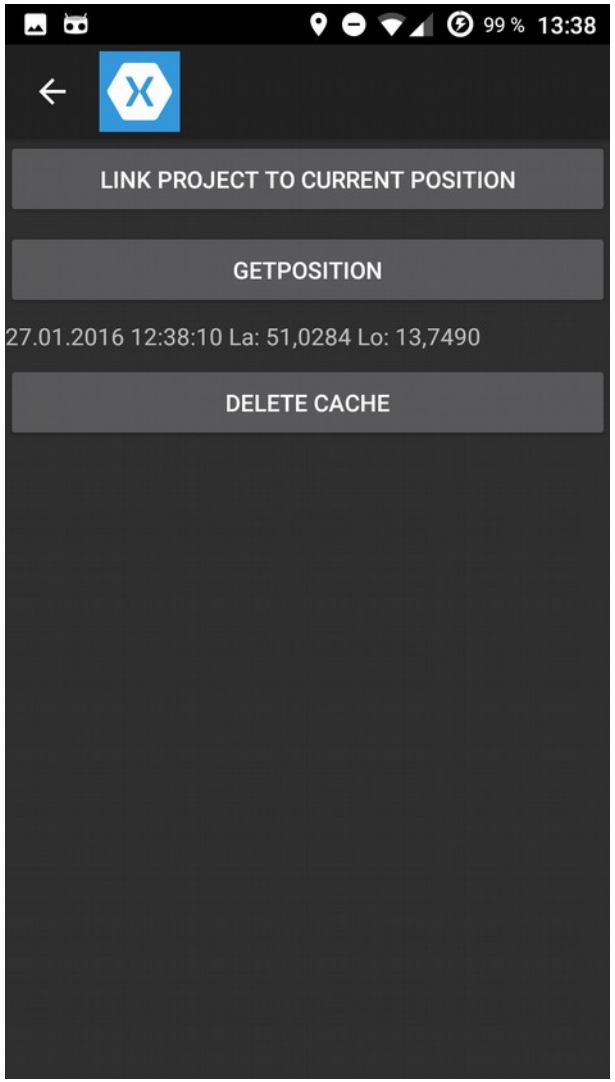


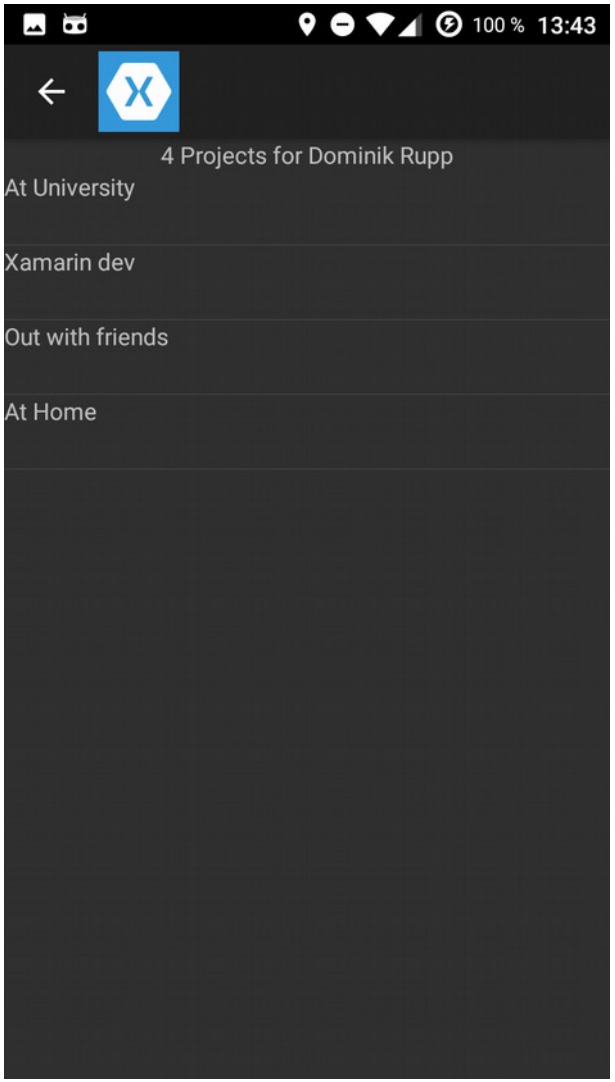


TECHNISCHE UNIVERSITÄT DRESDEN

Screenshots II

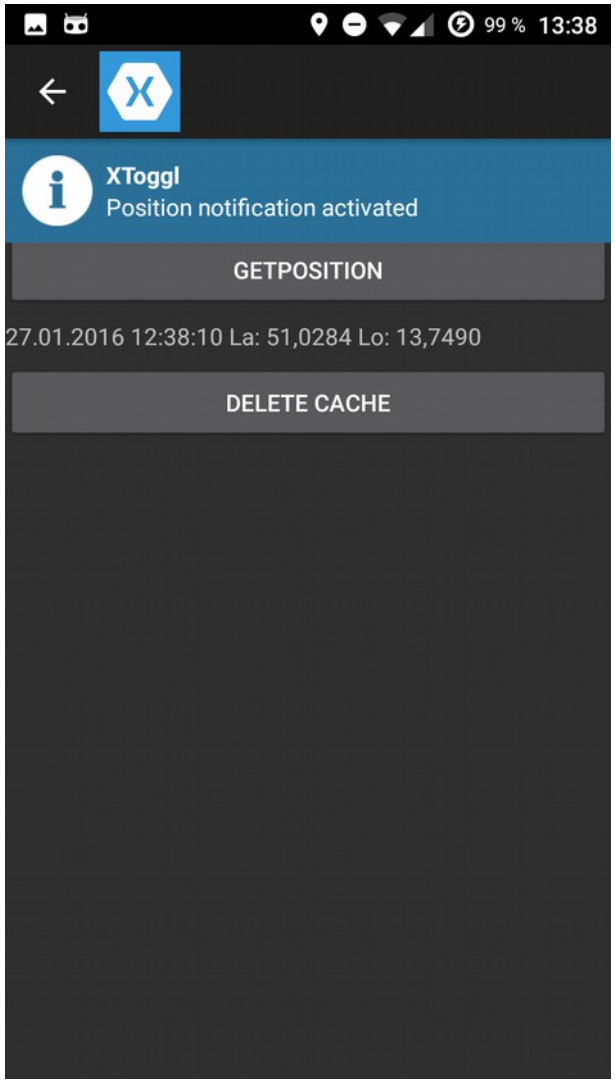


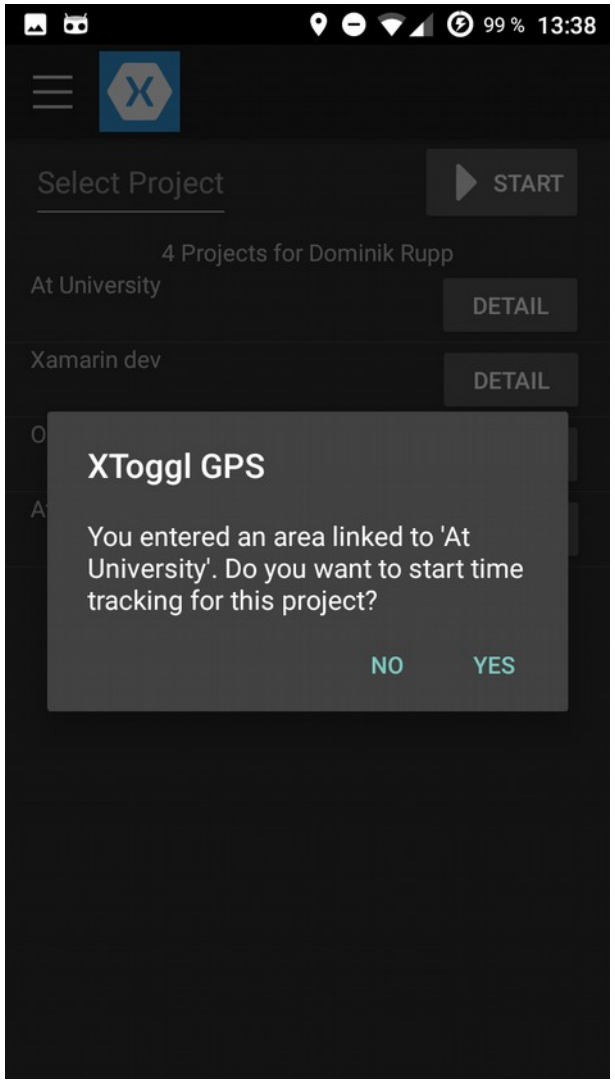


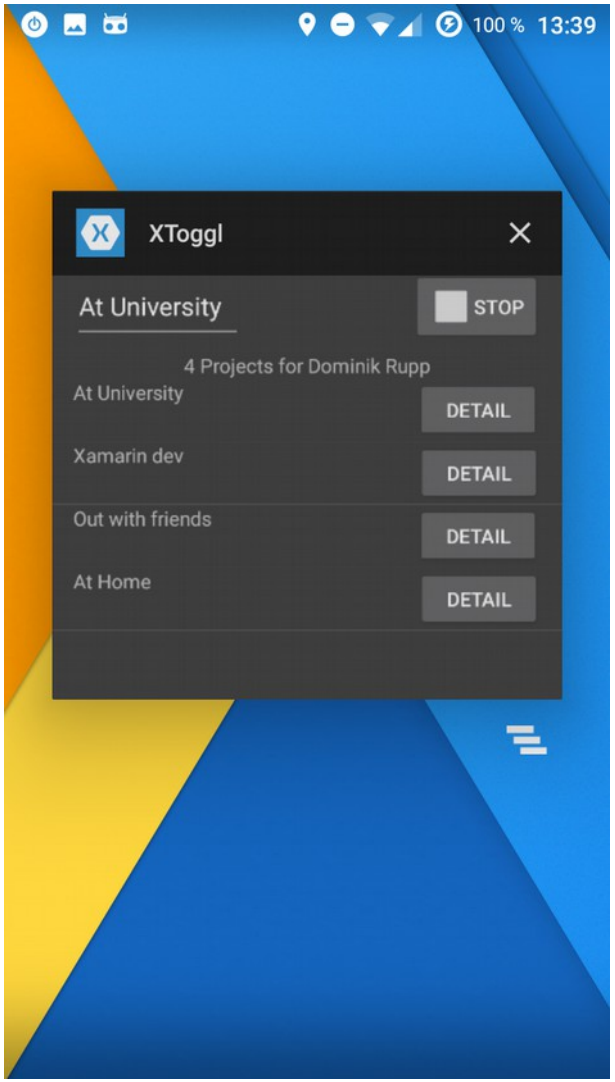


TECHNISCHE UNIVERSITÄT DRESDEN

Screenshots III

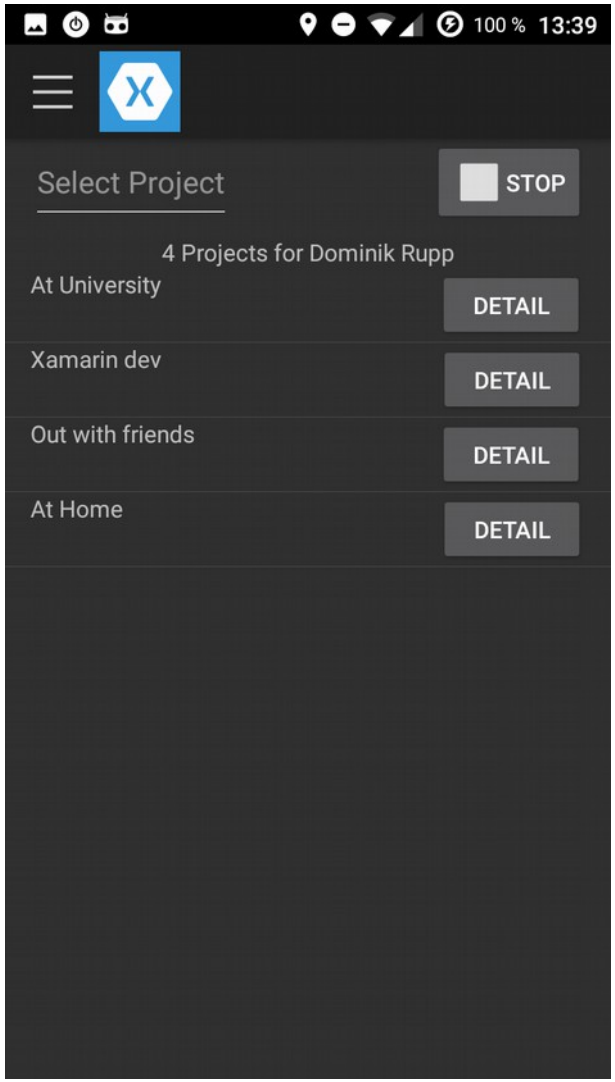


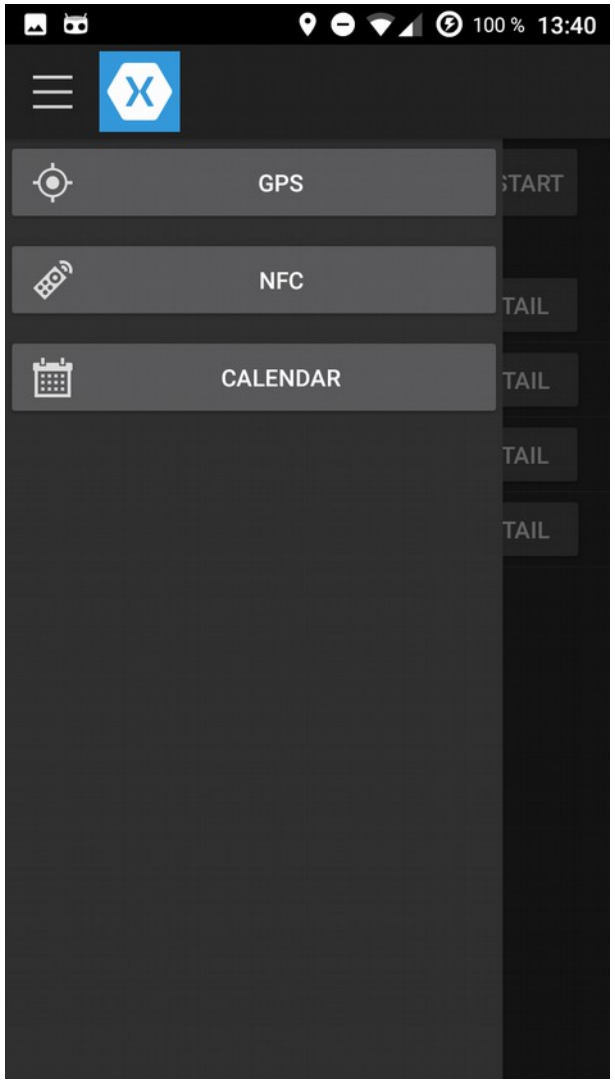


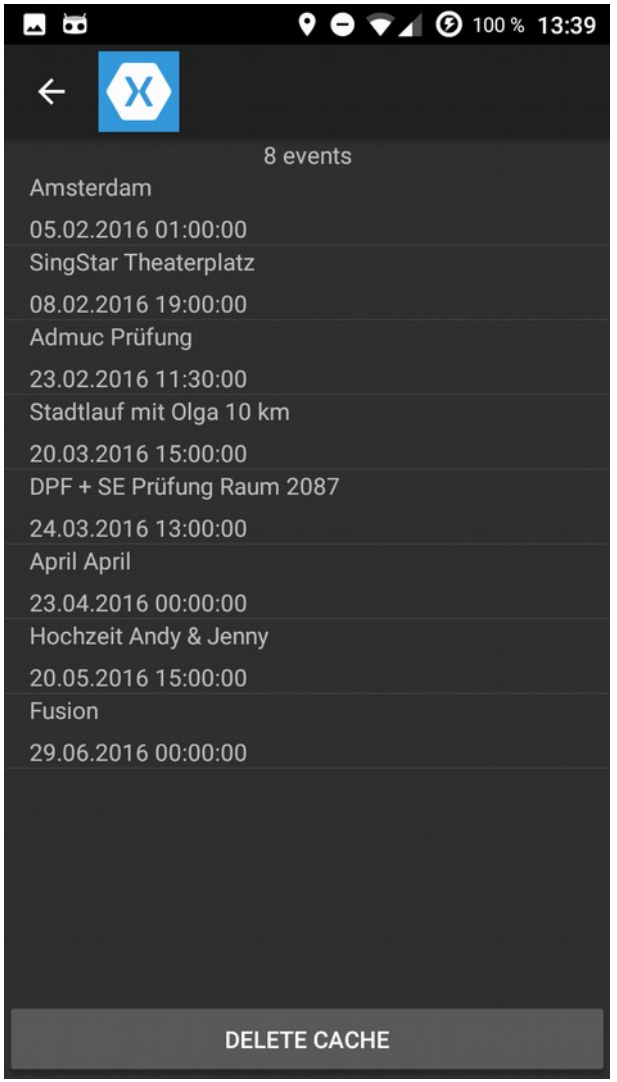


TECHNISCHE UNIVERSITÄT DRESDEN

Screenshots IV

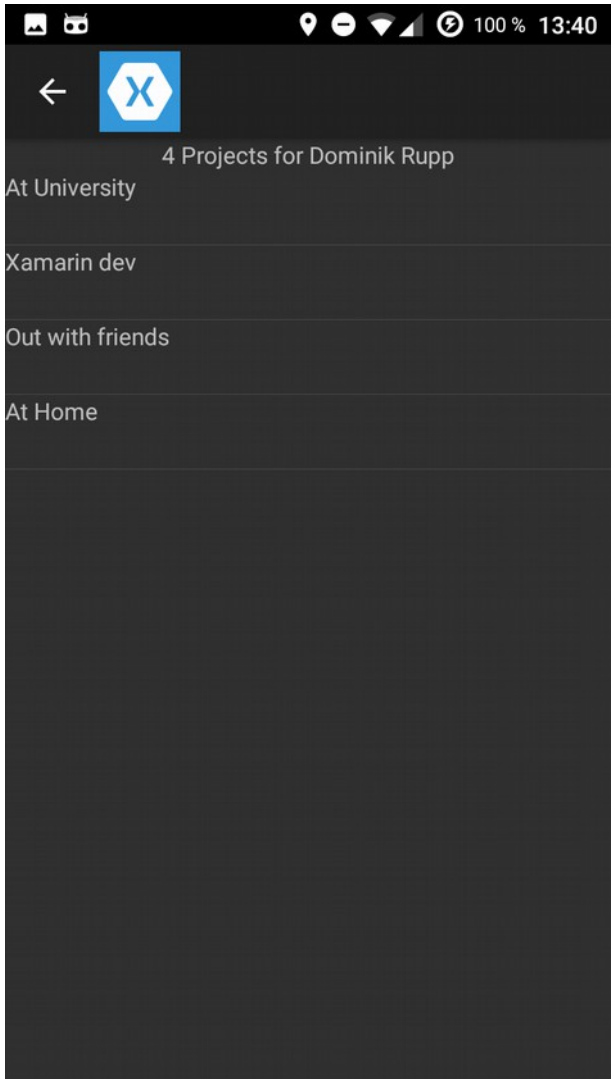


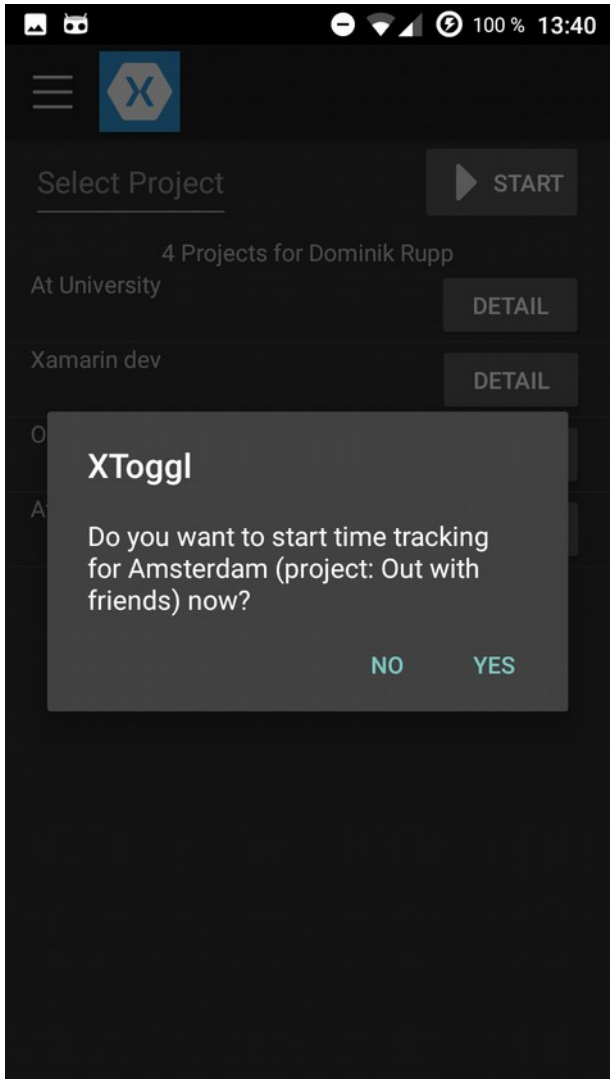


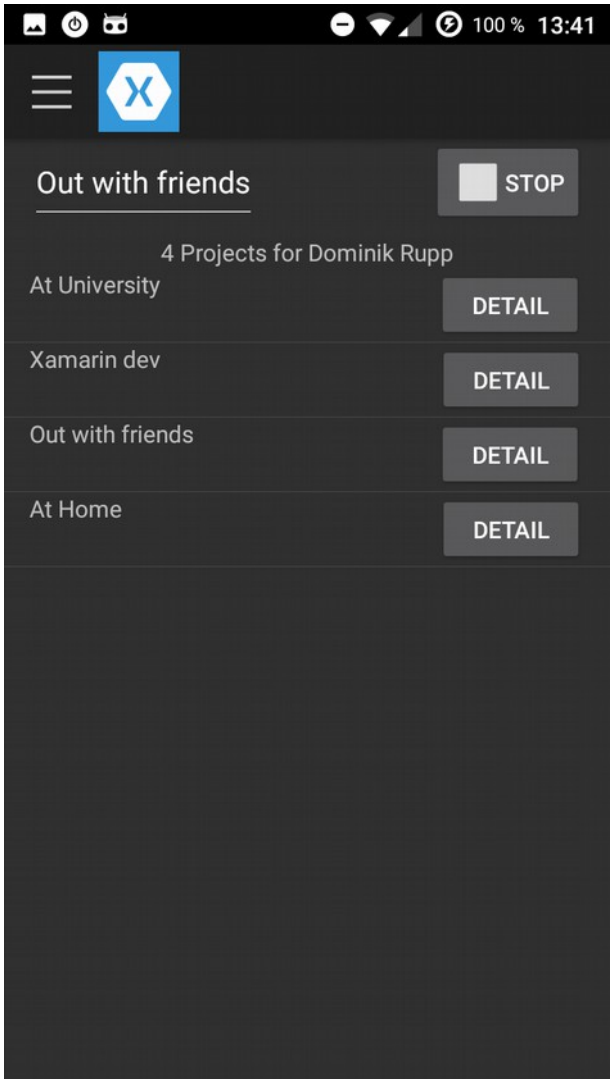


TECHNISCHE UNIVERSITÄT DRESDEN

Screenshots V







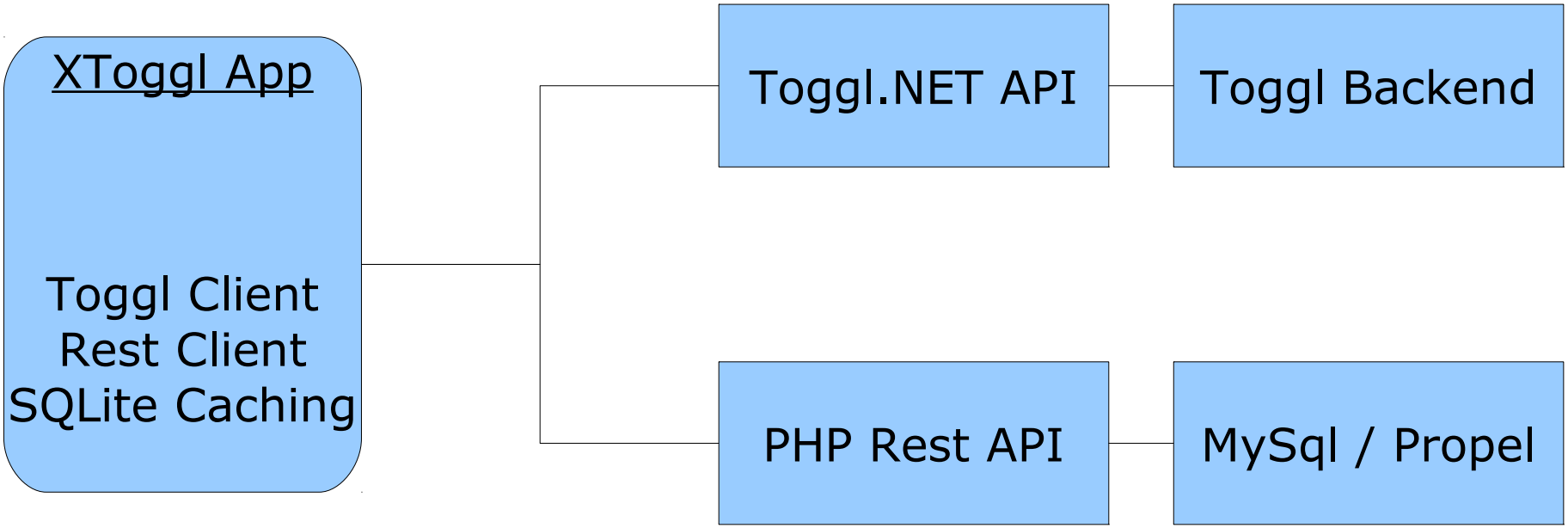
Technologies

- Xamarin Forms
- XLabs
    - Caching (SQLiteSimpleCache) + SQLite.NET
    - IoC (DependencyService)
    - Platform (Geolocator)
    - Serialization (StreamSerializer)
    - Web (JsonRestClient)

- Toast Plugin
- MySql through RestAPI (PHP)
- Propel ORM
- Toggl.NET API

TECHNISCHE UNIVERSITÄT DRESDEN

Architecture



Adaptation & Context

- Linking current GPS position to Toggl projects
  - Notification when entering area (GPS active)
- Linking upcoming calendar events to Toggl projects
  - Notification through countdown until next event
- Quicker loading through caching
  - Does not sync offline data though

TECHNISCHE UNIVERSITÄT DRESDEN

Implementation Details (Android)

```
EventProvider.ContentResolver = ContentResolver;
DependencyService.Register<EventProvider> ();
DependencyService.Register<EventNotification> ();
DependencyService.Register<Geolocator> ();

if (!Resolver.IsSet) {
    var documents = System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);
    var pathToDatabase = Path.Combine(documents, "xforms.db");

    var resolverContainer = new SimpleContainer();
    var serializer = new SystemJsonSerializer();

    resolverContainer
        .Register<IJsonSerializer> (t => serializer)
        .Register<IRestClient>(new JsonRestClient(serializer))
        .Register<ISimpleCache> (
        t => new SQLiteSimpleCache (new SQLite.Net.Platform.XamarinAndroid.SQLitePlatformAndroid (),
            new SQLite.Net.SQLiteConnectionString (pathToDatabase, true), t.Resolve<IJsonSerializer> ()));

    Resolver.SetResolver(resolverContainer.GetResolver());
}
```

MainActivity.cs (Android)

Implementation Details (Cross)

```
private void InitializeEventProvider()
{
    var eventProvider = DependencyService.Get<IEventProvider> ();
    eventProvider.Init ();
}
```

```
var eventNotification = DependencyService.Get<IEventNotification> ();
eventNotification.RegisterEvent (_upcomingEvent.StartDate, PrintMsg);
```

```
ISimpleCache cache = Resolver.Resolve<ISimpleCache>();
var keyValue = cache.Get<List<Toggl.Project>>(ProjectsKey);
_projects = keyValue != null ? new ObservableCollection<Toggl.Project> (keyValue) :
    new ObservableCollection<Toggl.Project> ();
```

Usage in cross platform project

Implementation Details (Rest Client)

```
public class RestAPI
{
    private static IRestClient _client = Resolver.Resolve<IRestClient> ();
    private static string _baseUrl = "http://ns377884.ip-5-196-89.eu/Dominik/x/";

    #region Events

    public async static Task<List<Event>> GetEventsWithProjects()
    {
        if (!App.User.Id.HasValue)
            throw new InvalidOperationException ("unauthorized user");

        string url = _baseUrl + "events/" + App.User.Id;
        var response = await _client.GetAsync<EventsResult>(url);
        return response.Events;
    }

    public async static void SetEventsWithProject(Event e)
    {
        if (!App.User.Id.HasValue)
            throw new InvalidOperationException ("unauthorized user");

        string url = _baseUrl + "event";
        await _client.PostAsync<Event> (url, e);
    }

    #endregion
```

TECHNISCHE UNIVERSITÄT DRESDEN

Implementation Details (Rest server)

```php
/**
 * Gets events and connected projectIds
 *
 * @url GET /events/$id
 */
public function getEvents($id = null) {
    if ($id) {
      $events = EventQuery::create()->filterByUserId($id)->find();
      return $events->toJson();
    } else return null;
}
/**
 * Create Event with name, startDate and projectId /$name/$startDate/$projectId/
 *
 * @url POST /event
 */
public function setEvent($data) {
    $event = new Event();
    $event->setName($data->Name);
    $event->setStartDate($data->StartDate);
    $event->setProjectId($data->ProjectId);
    $event->setUserId($data->UserId);
    $event->save();
    return $event->toJson();
}
```

Based on open source project: https://github.com/jacwright/RestServer

Implementation Details (Propel)

```xml
<database name="default" defaultIdMethod="native"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="http://xsd.propelorm.org/1.6/database.xsd"
          namespace="">

    <table name="event" phpName="Event">
        <column name="id" type="integer" required="true" primaryKey="true" autoIncrement="true"/>
        <column name="name" type="varchar" size="255" required="true"/>
        <column name="start_date" type="date" required="true" />
        <column name="project_id" type="integer" required="true"/>
        <column name="user_id" type="integer" required="true"/>
    </table>

    <table name="gps_position" phpName="GpsPosition">
        <column name="id" type="integer" required="true" primaryKey="true" autoIncrement="true"/>
        <column name="latitude" type="double" required="true" />
        <column name="longitude" type="double" required="true" />
        <column name="project_id" type="integer" required="true"/>
        <column name="user_id" type="integer" required="true"/>
    </table>

</database>
```

```
mysql> Show tables;
+------------------+
| Tables_in_Xamarin |
+------------------+
| event            |
| gps_position     |
+------------------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM event;
+----+---------+------------+------------+---------+
| id | name    | start_date | project_id | user_id |
+----+---------+------------+------------+---------+
|  1 | Testiii | 2016-01-08 |   12108323 | 1944084 |
+----+---------+------------+------------+---------+
1 row in set (0.00 sec)
```

propel sql:build

propel model:build

Event.php	42.451 B	06.01.2016 22:00:59
EventQuery.php	22.323 B	06.01.2016 22:00:59
GpsPosition.php	41.564 B	10.01.2016 12:14:00
GpsPositionQuery.php	23.511 B	10.01.2016 12:14:00

Implementation Details (JSON2C#)

ns377884.ip-5-196-89.eu/Dominik/x/events/1944084


```
{"Events":[{"Id":1,"Name":"Testiii","StartDate":"2016-01-08T00:00:00+01:00","ProjectId":12108323,"UserId":1944084}]}
```


Paste here


```
{"Events":
[{"Id":1,"Name":"Testiii","StartDate":"2016-01-08T00:00:00+01:00","ProjectId":1210832
3,"UserId":1944084}]}
```


Generate

? ! @

x

```
public class Event
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string StartDate { get; set; }
    public int ProjectId { get; set; }
    public int UserId { get; set; }
}

public class RootObject
{
    public List<Event> Events { get; set; }
}
```

http://json2csharp.com/

Questions?

Thank you for your attention!