



medLOG

Concept and Implementation

© 2014 Symmetric GmbH, all rights reserved

# Table of Contents

I.	<b>Introduction</b> .....	<b>3</b>
	Development Tools	
	Programming Style	
	Naming	
	Project File Structure	
	Logo	
II.	<b>Technical Concept</b> .....	<b>6</b>
	Description	
	Overview (medLOG Standard)	
	Overview (medLOG Pro)	
	Details	
III.	<b>medLOG Standard</b> .....	<b>8</b>
	Start / Splash Screen	
	Startmenu Icon (Tile)	
	Basic navigation	
	New Entry	
	Entries (grid)	
	Analysis	
	Report	
IV.	<b>medLOG Pro</b> .....	<b>13</b>
V.	<b>Code</b> .....	<b>14</b>
	SQL	
	Multi Language	
VI.	<b>Appendix</b> .....	<b>22</b>
	Costs / Milestones	

## List of changes

Version	Date	Author	Comments
1.0	09.05.2014	Frank Uray	Initial Version

## References

Ref.	Document	Version	Date

## Questions / open points

1	How do we ensure a update path (app and database) ?	
2		
3		
4		
5		
6		

# Introduction

**medLOG** is a Windows Phone (and Windows 8) App for pattern recognition in case of medical complaints. It will be found over the Microsoft App Store.

In case of Windows 8 it should be also a Store App (not a Desktop App), I think it can be the same app like for Windows Phone.

Payment for Standard Version will proceed over Microsoft Store.

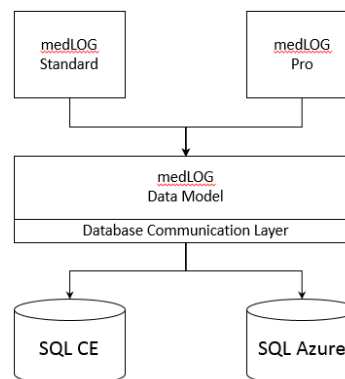
There will be two versions:

## **medLOG Standard:**

- Local Database (maybe SQL CE or SQLITE)
- One time payment

## **medLOG Pro:**

- Cloud Database (maybe Azure)
- Subscription payment



## Development Tools

medLOG has to be developed using Microsoft Visual Studio 2013 with Framework 4.0 or higher and in the Language C# / XAML.

Use Standard Framework Controls.

If possible, use Online TFS: [symmetric.visualstudio.com](http://symmetric.visualstudio.com)

## Programming Style

- Keep code as simple possible
- Use OO approach whenever possible
- Use Multitier architecture
- Write same code only once: Use shared classes approach to outsource code

## Naming

- Project / Solution: medlog
- Shared Classes: medlog.shared
- Data / Data Entities: medlog.data (if possible Entity Framework)
- WCF Proxy: medlog.proxy

## Project File Structure

- 10-Code
  - \_CodeBuild
    - PlugIn
    - \_References
    - \_Resources
  - Applications
    - \_Classes
    - \_PlugIns
  - ApplicationsMobile
    - \_Classes
    - \_PlugIns
  - ApplicationsStore
    - \_Classes
    - \_PlugIns
  - IntegrationServices
  - Interfaces
  - Services
    - \_Classes
    - \_PlugIns
  - Setups
  - SharedClasses
  - SQL
  - UnitTests
  - Webs
  - WebServices
- 15-Storyboard
- 20-Documents
- 30-Tasks
- 40-Files
- 45-Changes-Tasks
- 50-Databases
- 60-Errors
- 90-Versions
  - Current
    - PlugIns
    - Publish
- 95-History
- BuildProcessTemplates

**Logo**

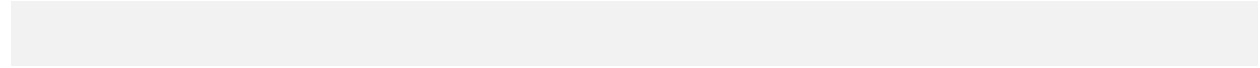
The logo files in png format will be provided.



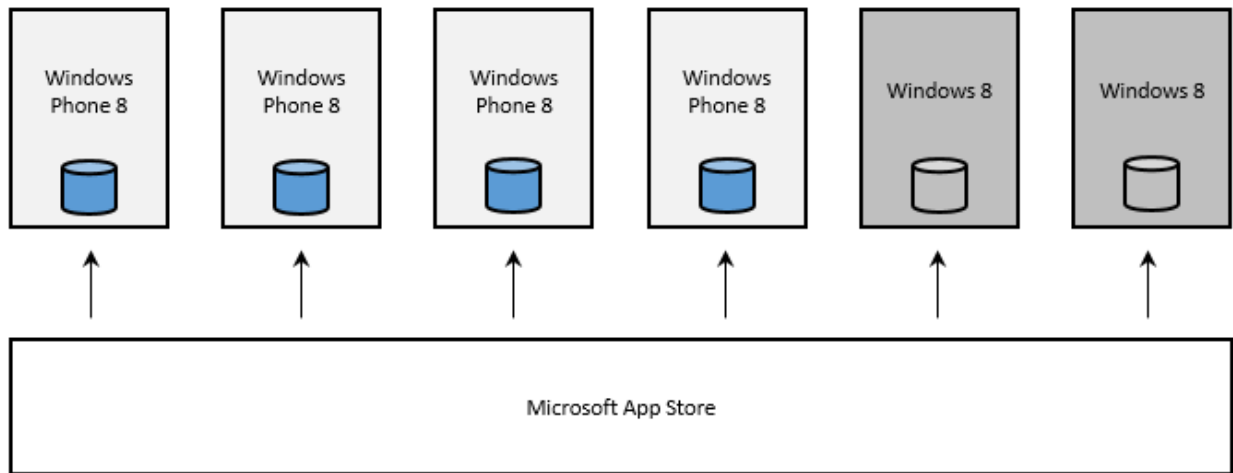
# Technical Concept

---

## Description



## Overview (medLOG Standard)



## Overview (medLOG Pro)



## Details

- As a local database, SQL CE or SQLITE can be used
- Local data will not be backed .



# medLOG Standard

---

The Mockups gives the developer an idea of what is expected.

The app does not have to look exactly the same, just close to that. There is room for improvement ☺

Use everywhere the same font (Verdana).

## Start / Splash Screen

When the application is starting, this screen is showing.



## Startmenu Icon (Tile)

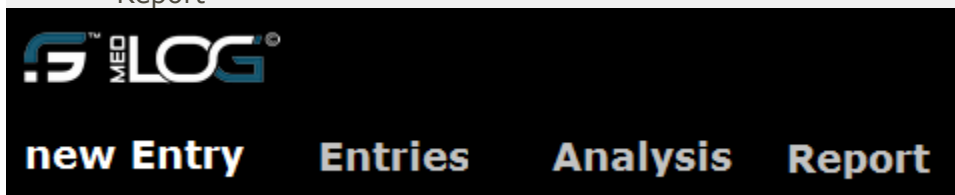
Tile back color has to be the accent color of windows phone or windows 8.



## Basic navigation

There will be 4 main navigation pages:

- New Entry
- Entries (grid)
- Analysis
- Report



## New Entry

Description of input page:

- Select Category: when selected a dropdown with all the categories from the database is shown. Must be filled.
- Select date (start): initially current date is shown. When selected a date selector is shown.
- Select time (start): initially current time is shown. When selected a time selector is shown.
- Select date (end): initially current date is shown. When selected a date selector is shown. (can be none)
- Select time (end): initially current time + 30 minutes is shown. When selected a time selector is shown. (can be none)
- Duration: calculated from the above start and end. Must be filled.
- Text: while typing a auto complete dropdown (like google) must be shown, if any matching.
- Enter details: a grid to enter some details
- Enter description: a textbox to enter some description
- Intensity: Stars left to right from the database. For example: when tapping the 6<sup>th</sup>. star, bgcolor (#CC0000) of the stars 1 to 6 are changed.
- Add new entry: button to add the new entry. After adding, the above selections and entries (category, text, details, description) are cleared to enter a new entry.

new entry

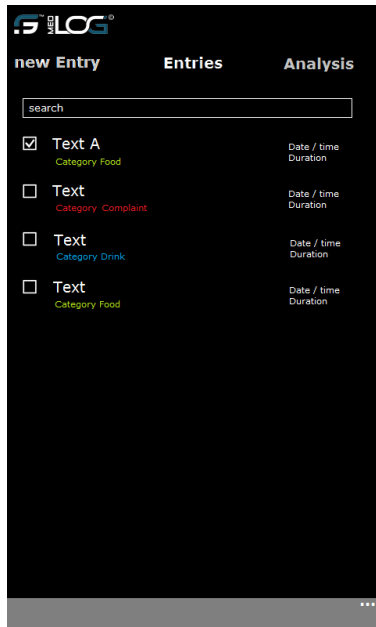
details

description

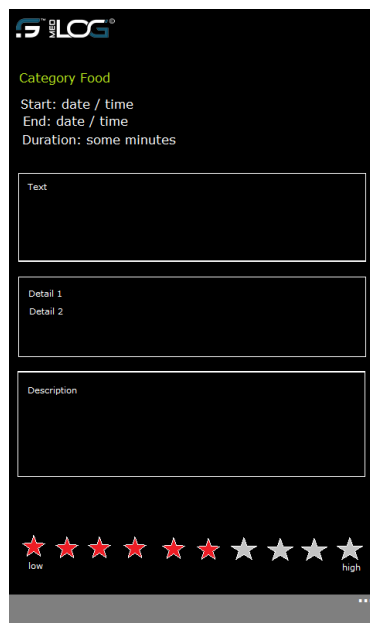
## Entries (grid)

Description of entries and show/edit entry:

- Text, category (in category color from database), start date/time and duration is shown.
- Ordered by Start date / time descending
- Search textbox for search over category / text / details / description. Grid will be changed while typing.
- Tap entry will open the entry show page.
- Selector (from menu bar) for delete multiple entries.
- Long hold on entry will show context menu: edit and delete (only current entry)
- Edit entry is basically the same like new entry, just already filled and some texts are different. See image below.



Entries



show



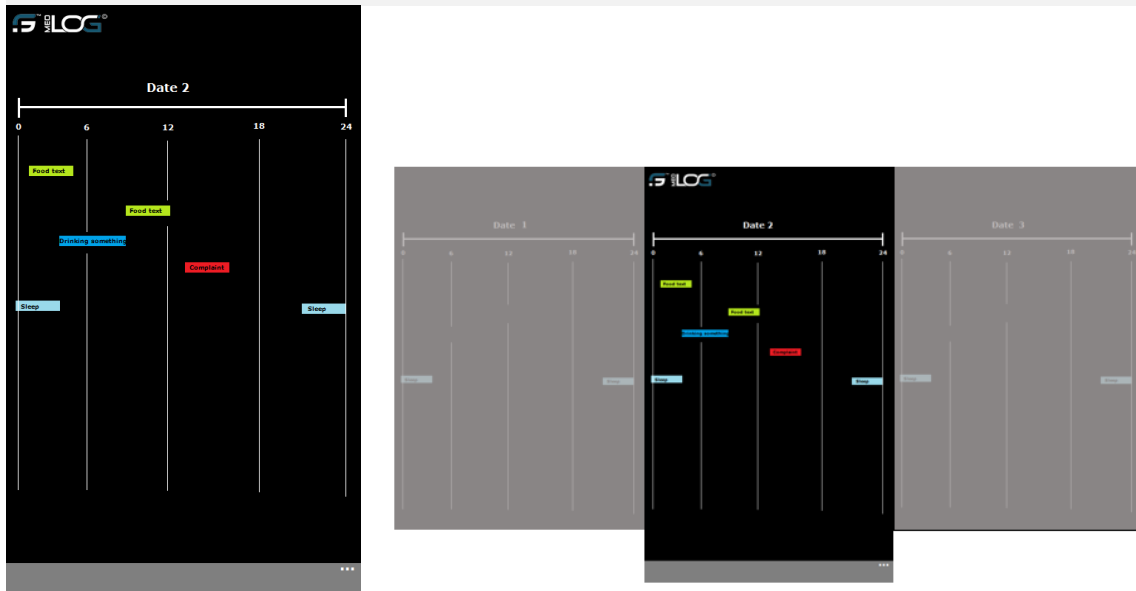
edit

# Analysis

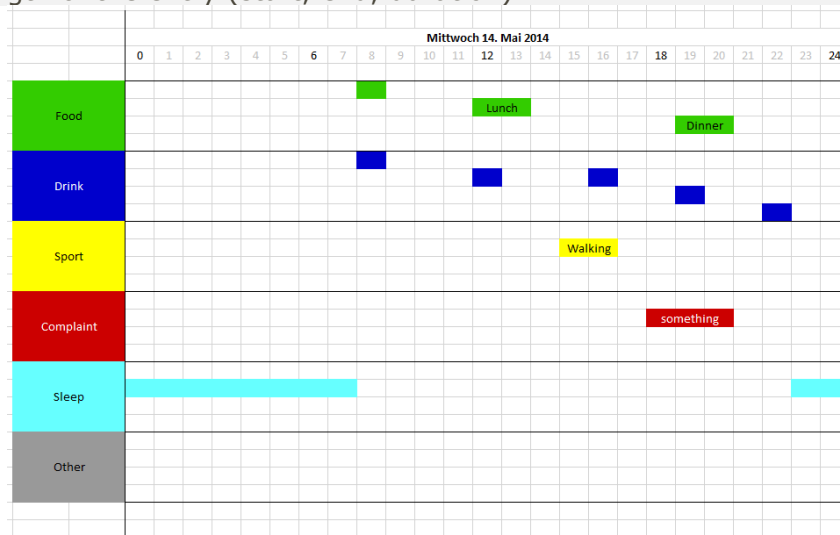
Description of Analysis:

- Scrollable days (left to right and right to left). Initially current day is in the middle.
- Entries are shown in correct time ranges
- Entries are in category color
- When tapping an entry, the show entry page is loaded. Hit back goes back to analysis.

I am not sure what kind of control will be used here.  
We can discuss about the design.



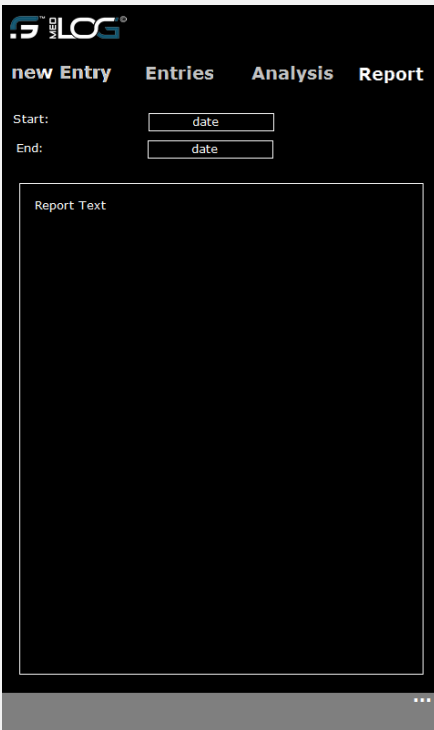
I would try (just as an idea) to build a grid as you see in the following picture. The categories on the y-axis are evenly distributed and ordered by the Order attribute from database. The entries on their y-axis are also evenly distributed on their section, the x-axis is the length of the entry (start, end, duration).



## Report

Description of Report:

- Basicly this will have from / to selector and a textbox.
- The logic will be implemented by Frank Uray



The screenshot displays the MEDLOG software interface. At the top left is the MEDLOG logo. Below it is a navigation menu with four items: "new Entry", "Entries", "Analysis", and "Report". The "Report" item is highlighted. Below the menu are two input fields: "Start:" followed by a text box containing "date", and "End:" followed by another text box containing "date". Below these fields is a large, empty text area labeled "Report Text". At the bottom right of the interface, there are three small dots indicating a menu or options.

# medLOG Pro

---

Will not be implemented yet.

# Code

---

## SQL

Initial database for medLOG Standard

```
USE [medLOG]
GO
ALTER TABLE [_medlog].[Entries] DROP CONSTRAINT [fk_medlog_Entries_Intensities]
GO
ALTER TABLE [_medlog].[Entries] DROP CONSTRAINT [fk_medlog_Entries_Categories]
GO
ALTER TABLE [_medlog].[Details] DROP CONSTRAINT [fk_medlog_Details_Entries]
GO
ALTER TABLE [_medlog].[Entries] DROP CONSTRAINT [df_medlog_Entries_DurationMinutes]
GO
ALTER TABLE [_medlog].[Entries] DROP CONSTRAINT [df_medlog_Entries_Start]
GO
/***** Object: Index [pk_meta_Categories]    Script Date: Mittwoch, 7, Mai, 2014
19:47:25 *****/
ALTER TABLE [_meta].[Categories] DROP CONSTRAINT [pk_meta_Categories]
GO
/***** Object: Table [_meta].[Intensities]    Script Date: Mittwoch, 7, Mai, 2014
19:47:25 *****/
DROP TABLE [_meta].[Intensities]
GO
/***** Object: Table [_meta].[Categories]    Script Date: Mittwoch, 7, Mai, 2014
19:47:25 *****/
DROP TABLE [_meta].[Categories]
GO
/***** Object: Table [_medlog].[Entries]    Script Date: Mittwoch, 7, Mai, 2014
19:47:25 *****/
DROP TABLE [_medlog].[Entries]
GO
/***** Object: Table [_medlog].[Details]    Script Date: Mittwoch, 7, Mai, 2014
19:47:25 *****/
DROP TABLE [_medlog].[Details]
GO
/***** Object: Schema [_meta]    Script Date: Mittwoch, 7, Mai, 2014 19:47:25 *****/
DROP SCHEMA [_meta]
GO
/***** Object: Schema [_medlog]    Script Date: Mittwoch, 7, Mai, 2014 19:47:25
*****/
DROP SCHEMA [_medlog]
GO
/***** Object: Schema [_medlog]    Script Date: Mittwoch, 7, Mai, 2014 19:47:25
*****/
CREATE SCHEMA [_medlog]
GO
/***** Object: Schema [_meta]    Script Date: Mittwoch, 7, Mai, 2014 19:47:25 *****/
CREATE SCHEMA [_meta]
GO
```

```

/***** Object: Table [_medlog].[Details]      Script Date: Mittwoch, 7, Mai, 2014
19:47:25 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [_medlog].[Details](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [FK_Entry] [int] NOT NULL,
    [Detail] [varchar](max) NOT NULL,
    [Amount] [varchar](50) NULL,
    [Description] [varchar](max) NULL,
    CONSTRAINT [pk_medlog_Details] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [DATA]
) ON [DATA] TEXTIMAGE_ON [DATA]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [_medlog].[Entries]      Script Date: Mittwoch, 7, Mai, 2014
19:47:25 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [_medlog].[Entries](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [FK_Category] [int] NOT NULL,
    [FK_Intensity] [int] NULL,
    [Start] [smalldatetime] NOT NULL,
    [End] [smalldatetime] NULL,
    [DurationMinutes] [int] NOT NULL,
    [Amount] [varchar](50) NULL,
    [Text] [varchar](max) NOT NULL,
    [Description] [varchar](max) NULL,
    CONSTRAINT [pk_medlog_Entries] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [DATA]
) ON [DATA] TEXTIMAGE_ON [DATA]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [_meta].[Categories]      Script Date: Mittwoch, 7, Mai, 2014
19:47:25 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [_meta].[Categories](

```



```

        [ID] [int] IDENTITY(1,1) NOT NULL,
        [Category] [varchar](255) NOT NULL,
        [Color] [varchar](10) NULL,
        [Description] [varchar](max) NULL,
        [Order] [int] NULL
    ) ON [DATA] TEXTIMAGE_ON [DATA]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [_meta].[Intensities]    Script Date: Mittwoch, 7, Mai, 2014
19:47:25 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [_meta].[Intensities](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Intensity] [varchar](50) NULL,
    [Description] [varchar](max) NULL,
    CONSTRAINT [pk_meta_Intensities] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [DATA]
) ON [DATA] TEXTIMAGE_ON [DATA]

GO
SET ANSI_PADDING OFF
GO
SET IDENTITY_INSERT [_medlog].[Details] ON

GO
INSERT [_medlog].[Details] ([ID], [FK_Entry], [Detail], [Amount], [Description]) VALUES
(1, 2, N'Bread', N'1 Piece', NULL)
GO
INSERT [_medlog].[Details] ([ID], [FK_Entry], [Detail], [Amount], [Description]) VALUES
(2, 2, N'Butter', NULL, NULL)
GO
INSERT [_medlog].[Details] ([ID], [FK_Entry], [Detail], [Amount], [Description]) VALUES
(3, 2, N'Honey', NULL, NULL)
GO
INSERT [_medlog].[Details] ([ID], [FK_Entry], [Detail], [Amount], [Description]) VALUES
(4, 6, N'Steak', NULL, NULL)
GO
INSERT [_medlog].[Details] ([ID], [FK_Entry], [Detail], [Amount], [Description]) VALUES
(5, 6, N'Pommes', NULL, NULL)
GO
INSERT [_medlog].[Details] ([ID], [FK_Entry], [Detail], [Amount], [Description]) VALUES
(6, 8, N'Corn Flakes', NULL, NULL)
GO
SET IDENTITY_INSERT [_medlog].[Details] OFF
GO
SET IDENTITY_INSERT [_medlog].[Entries] ON

GO
INSERT [_medlog].[Entries] ([ID], [FK_Category], [FK_Intensity], [Start], [End],
[DurationMinutes], [Amount], [Text], [Description]) VALUES (1, 5, 5,

```

```

CAST(0x0000A323017B0740 AS DateTime), CAST(0x0000A324007B98A0 AS DateTime), 510, NULL,
N'sleeping', NULL)
GO
INSERT [_medlog].[Entries] ([ID], [FK_Category], [FK_Intensity], [Start], [End],
[DurationMinutes], [Amount], [Text], [Description]) VALUES (2, 1, NULL,
CAST(0x0000A3240083D600 AS DateTime), NULL, 10, NULL, N'Breakfest', NULL)
GO
INSERT [_medlog].[Entries] ([ID], [FK_Category], [FK_Intensity], [Start], [End],
[DurationMinutes], [Amount], [Text], [Description]) VALUES (3, 2, NULL,
CAST(0x0000A3240083D600 AS DateTime), CAST(0x0000A3240087F4B0 AS DateTime), 15, N'3
dl', N'Tee', NULL)
GO
INSERT [_medlog].[Entries] ([ID], [FK_Category], [FK_Intensity], [Start], [End],
[DurationMinutes], [Amount], [Text], [Description]) VALUES (4, 3, 6,
CAST(0x0000A324009C8E20 AS DateTime), CAST(0x0000A32400AD08E0 AS DateTime), 60, N'20
km', N'Bike', NULL)
GO
INSERT [_medlog].[Entries] ([ID], [FK_Category], [FK_Intensity], [Start], [End],
[DurationMinutes], [Amount], [Text], [Description]) VALUES (5, 4, 8,
CAST(0x0000A32400A8EA30 AS DateTime), CAST(0x0000A32400B54640 AS DateTime), 45, NULL,
N'Bellyache', NULL)
GO
INSERT [_medlog].[Entries] ([ID], [FK_Category], [FK_Intensity], [Start], [End],
[DurationMinutes], [Amount], [Text], [Description]) VALUES (6, 1, NULL,
CAST(0x0000A32400DA5A70 AS DateTime), NULL, 30, NULL, N'Lunch', NULL)
GO
INSERT [_medlog].[Entries] ([ID], [FK_Category], [FK_Intensity], [Start], [End],
[DurationMinutes], [Amount], [Text], [Description]) VALUES (7, 5, 5,
CAST(0x0000A324018344A0 AS DateTime), CAST(0x0000A3250083D600 AS DateTime), 510, NULL,
N'sleeping', NULL)
GO
INSERT [_medlog].[Entries] ([ID], [FK_Category], [FK_Intensity], [Start], [End],
[DurationMinutes], [Amount], [Text], [Description]) VALUES (8, 1, NULL,
CAST(0x0000A3250087F4B0 AS DateTime), NULL, 10, NULL, N'Breakfest', NULL)
GO
INSERT [_medlog].[Entries] ([ID], [FK_Category], [FK_Intensity], [Start], [End],
[DurationMinutes], [Amount], [Text], [Description]) VALUES (9, 2, NULL,
CAST(0x0000A3250087F4B0 AS DateTime), CAST(0x0000A325008C1360 AS DateTime), 15, N'3
dl', N'Tee', NULL)
GO
SET IDENTITY_INSERT [_medlog].[Entries] OFF
GO
SET IDENTITY_INSERT [_meta].[Categories] ON

GO
INSERT [_meta].[Categories] ([ID], [Category], [Color], [Description]) VALUES (1,
N'Food', '#33CC00', NULL)
GO
INSERT [_meta].[Categories] ([ID], [Category], [Color], [Description]) VALUES (2,
N'Drink', '#0000CC', NULL)
GO
INSERT [_meta].[Categories] ([ID], [Category], [Color], [Description]) VALUES (3,
N'Sport', '#FFFF00', NULL)
GO
INSERT [_meta].[Categories] ([ID], [Category], [Color], [Description]) VALUES (4,
N'Complaint', '#CC0000', NULL)
GO
INSERT [_meta].[Categories] ([ID], [Category], [Color], [Description]) VALUES (5,
N'Sleep', '#66FFFF', NULL)
GO

```

```

INSERT [_meta].[Categories] ([ID], [Category], [Color], [Description]) VALUES (6,
N'Other', '#999999', NULL)
GO
SET IDENTITY_INSERT [_meta].[Categories] OFF
GO
SET IDENTITY_INSERT [_meta].[Intensities] ON

GO
INSERT [_meta].[Intensities] ([ID], [Intensity], [Description]) VALUES (1, N'1',
N'low')
GO
INSERT [_meta].[Intensities] ([ID], [Intensity], [Description]) VALUES (2, N'2', NULL)
GO
INSERT [_meta].[Intensities] ([ID], [Intensity], [Description]) VALUES (3, N'3', NULL)
GO
INSERT [_meta].[Intensities] ([ID], [Intensity], [Description]) VALUES (4, N'4', NULL)
GO
INSERT [_meta].[Intensities] ([ID], [Intensity], [Description]) VALUES (5, N'5',
N'medium')
GO
INSERT [_meta].[Intensities] ([ID], [Intensity], [Description]) VALUES (6, N'6',
N'medium')
GO
INSERT [_meta].[Intensities] ([ID], [Intensity], [Description]) VALUES (7, N'7', NULL)
GO
INSERT [_meta].[Intensities] ([ID], [Intensity], [Description]) VALUES (8, N'8', NULL)
GO
INSERT [_meta].[Intensities] ([ID], [Intensity], [Description]) VALUES (9, N'9', NULL)
GO
INSERT [_meta].[Intensities] ([ID], [Intensity], [Description]) VALUES (10, N'10',
N'high')
GO
SET IDENTITY_INSERT [_meta].[Intensities] OFF
GO
/***** Object: Index [pk_meta_Categories]    Script Date: Mittwoch, 7, Mai, 2014
19:47:25 *****/
ALTER TABLE [_meta].[Categories] ADD CONSTRAINT [pk_meta_Categories] PRIMARY KEY
NONCLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
IGNORE_DUP_KEY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[DATA]
GO
ALTER TABLE [_medlog].[Entries] ADD CONSTRAINT [df_medlog_Entries_Start] DEFAULT
(getdate()) FOR [Start]
GO
ALTER TABLE [_medlog].[Entries] ADD CONSTRAINT [df_medlog_Entries_DurationMinutes]
DEFAULT ((30)) FOR [DurationMinutes]
GO
ALTER TABLE [_medlog].[Details] WITH NOCHECK ADD CONSTRAINT
[fk_medlog_Details_Entries] FOREIGN KEY([FK_Entry])
REFERENCES [_medlog].[Entries] ([ID])
ON DELETE CASCADE
NOT FOR REPLICATION
GO
ALTER TABLE [_medlog].[Details] CHECK CONSTRAINT [fk_medlog_Details_Entries]
GO
ALTER TABLE [_medlog].[Entries] WITH NOCHECK ADD CONSTRAINT
[fk_medlog_Entries_Categories] FOREIGN KEY([FK_Category])
REFERENCES [_meta].[Categories] ([ID])

```

```
ON DELETE CASCADE
NOT FOR REPLICATION
GO
ALTER TABLE [_medlog].[Entries] CHECK CONSTRAINT [fk_medlog_Entries_Categories]
GO
ALTER TABLE [_medlog].[Entries] WITH NOCHECK ADD CONSTRAINT
[fk_medlog_Entries_Intensities] FOREIGN KEY([FK_Intensity])
REFERENCES [_meta].[Intensities] ([ID])
ON DELETE CASCADE
NOT FOR REPLICATION
GO
ALTER TABLE [_medlog].[Entries] CHECK CONSTRAINT [fk_medlog_Entries_Intensities]
GO
```

# Multi Language

Use my symmetric.languages framework.

## Code Sample:

```
symmetric.languages.foundation.clsLanguages _clsLanguages = new symmetric.languages.foundation.clsLanguages();
_clsLanguages.LoadLanguageFiles(System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().GetName().CodeBase).Replace("file:\\", "") + @"\");
string _SampleText = _clsLanguages.GetText("plugin.Name", "eXXam.exams");
```

## XML Sample

```
<?xml version="1.0" standalone="yes"?>
<language>
  <eXXam.exams>
    <CultureInfo>de-AT</CultureInfo>
    <Description>Deutsch (Österreich)</Description>
    <Key>plugin.Name</Key>
    <Text>Prüfungen</Text>
  </eXXam.exams>
  <eXXam.exams>
    <CultureInfo>de-CH</CultureInfo>
    <Description>Deutsch (Schweiz)</Description>
    <Key>plugin.Name</Key>
    <Text>Prüfungen</Text>
  </eXXam.exams>
  <eXXam.exams>
    <CultureInfo>de-DE</CultureInfo>
    <Description>Deutsch (Deutschland)</Description>
    <Key>plugin.Name</Key>
    <Text>Prüfungen</Text>
  </eXXam.exams>
  <eXXam.exams>
    <CultureInfo>de-LI</CultureInfo>
    <Description>Deutsch (Liechtenstein)</Description>
    <Key>plugin.Name</Key>
    <Text>Prüfungen</Text>
  </eXXam.exams>
  <eXXam.exams>
    <CultureInfo>default</CultureInfo>
    <Description>Default</Description>
    <Key>plugin.Name</Key>
    <Text>Exams</Text>
  </eXXam.exams>
  <eXXam.exams>
    <CultureInfo>en-GB</CultureInfo>
    <Description>English (England)</Description>
    <Key>plugin.Name</Key>
    <Text>Exams</Text>
  </eXXam.exams>
  <eXXam.exams>
    <CultureInfo>en-US</CultureInfo>
    <Description>English (US)</Description>
    <Key>plugin.Name</Key>
    <Text>Exams</Text>
  </eXXam.exams>
  <eXXam.exams>
    <CultureInfo>es-ES</CultureInfo>
    <Description>Spanisch (Spanien)</Description>
    <Key>plugin.Name</Key>
    <Text>Pruebas</Text>
  </eXXam.exams>
  <eXXam.exams>
    <CultureInfo>fr-CH</CultureInfo>
    <Description>Französisch (Schweiz)</Description>
    <Key>plugin.Name</Key>
    <Text>Tests</Text>
  </eXXam.exams>
  <eXXam.exams>
    <CultureInfo>fr-FR</CultureInfo>
    <Description>Französisch (Frankreich)</Description>
    <Key>plugin.Name</Key>
    <Text>Tests</Text>
  </eXXam.exams>
  <eXXam.exams>
    <CultureInfo>it-CH</CultureInfo>
    <Description>Italienisch (Schweiz)</Description>
    <Key>plugin.Name</Key>
    <Text>Test</Text>
  </eXXam.exams>
  <eXXam.exams>
    <CultureInfo>it-IT</CultureInfo>
    <Description>Italienisch (Italien)</Description>
    <Key>plugin.Name</Key>
    <Text>Test</Text>
  </eXXam.exams>
  <eXXam.exams>
    <CultureInfo>nl-NL</CultureInfo>
    <Description>Dutch (Netherlands)</Description>
    <Key>plugin.Name</Key>
    <Text>Tests</Text>
  </eXXam.exams>
</language>
```

## Database Communication Layer

Use my symmetric.data framework.

### Code Sample

```
symmetric.data.clsData _clsData = new symmetric.data.clsData(symmetric.data.enums.clsEnums.ConnectionTypes.SQLCE, @"V:\_ DEVELOPMENT\_PROJECTS\medLOG\50-Databases\medLOG.sdf", "medLOG", "", symmetric.foundation._Variables.static_GlobalPassword);  
System.Data.DataTable _DataTable = _clsData.GetDataTable("SELECT * FROM [medLOG].[_meta].[Categories]");
```

```
symmetric.data.clsData _clsData = new symmetric.data.clsData(symmetric.data.enums.clsEnums.ConnectionTypes.SQLEnterprise, @"V:\_ DEVELOPMENT\_PROJECTS\medLOG\50-Databases\medLOG.sdf", "medLOG", "Username", "Password");  
System.Data.DataTable _DataTable = _clsData.GetDataTable("SELECT * FROM [medLOG].[_meta].[Categories]");
```

# Appendix

## Costs / Milestones

<b>TOTAL COSTS</b>	