Categis Software Pvt. Ltd. www.softwaretogo.de

softwaretogo | a beta systems company

OTTOPro

Technical documentation

Change Log:

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Author** | **Changes** | **Date** |
| 1 | Narendar Reddy | Initial Draft | 20-Apr-20 |

Contents

[1 Overview 4](#_Toc38287696)

[1.1 Current System 4](#_Toc38287697)

[1.2 Existing system(IGEL) 5](#_Toc38287698)

[2 Technologies 6](#_Toc38287699)

[3 Third party tools 6](#_Toc38287700)

[4 IDE tools 6](#_Toc38287701)

[5 Software requirements 6](#_Toc38287702)

[6 Hardware Requirements 7](#_Toc38287703)

[7 Solution Architecture 7](#_Toc38287704)

[7.1 OTTOPro Layer 7](#_Toc38287705)

[7.2 BL (Business Layer) 7](#_Toc38287706)

[7.3 EL (Entity Layer) 8](#_Toc38287707)

[7.4 DAL (Data Access /Layer) 8](#_Toc38287708)

[7.4.1 SQL Conn Class: 8](#_Toc38287709)

[7.5 GKSRVUtility Project 9](#_Toc38287710)

[8 Validations and Messages 9](#_Toc38287711)

[9 Internationalization, Localization and Translations 9](#_Toc38287712)

[10 Database Design and Implementations 10](#_Toc38287713)

[11 Security and Privacy Considerations 10](#_Toc38287714)

[12 Third Party Considerations 11](#_Toc38287715)

[12.1 DevExpress windows Forms controls 11](#_Toc38287716)

[12.1.1 Layout control 11](#_Toc38287717)

[12.1.2 Grid Control 12](#_Toc38287718)

[12.1.3 Tree list: 16](#_Toc38287719)

[12.1.4 Lookup Edit 17](#_Toc38287720)

[12.1.5 Xtra Report 17](#_Toc38287721)

[12.2 GAEB Converter 18](#_Toc38287722)

[12.3 Trinity DATANORM import 18](#_Toc38287723)

[13 Error and Exception Handling 19](#_Toc38287724)

[14 Error and Exceptions Logging 20](#_Toc38287725)

[15 Code structure and Coding standards 20](#_Toc38287726)

[16 Version Controlling on production 21](#_Toc38287727)

[17 Build 22](#_Toc38287728)

[17.1 Installer 22](#_Toc38287729)

[17.2 Database deployment 22](#_Toc38287730)

[17.2.1 Table 22](#_Toc38287731)

[17.2.2 Stored procedures, Triggers, Views, Keys, User defined functions and user defined types 22](#_Toc38287732)

[18 Test Objective 22](#_Toc38287733)

[19 Testing Techniques 23](#_Toc38287734)

[19.1 Functional testing 23](#_Toc38287735)

[19.1.1 Functionality testing 23](#_Toc38287736)

[19.1.2 Integration testing 24](#_Toc38287737)

[19.1.3 System testing 24](#_Toc38287738)

[19.1.4 Platform/ compatibility testing 24](#_Toc38287739)

[19.1.5 Smoke and sanity testing 24](#_Toc38287740)

[19.2 Nonfunctional testing 24](#_Toc38287741)

[19.2.1 Performance testing 24](#_Toc38287742)

[19.3 Maintenance testing 24](#_Toc38287743)

[19.3.1 Regression testing 24](#_Toc38287744)

[20 Testing Deliverables 25](#_Toc38287745)

[20.1 Requirement clarifications 25](#_Toc38287746)

[20.2 Test cases 25](#_Toc38287747)

[20.3 Release notes 25](#_Toc38287748)

[21 Testing Entry and Exit Criteria 26](#_Toc38287749)

[21.1 Entry criteria 26](#_Toc38287750)

[21.2 Exit criteria 27](#_Toc38287751)

[22 Defect Management 27](#_Toc38287752)

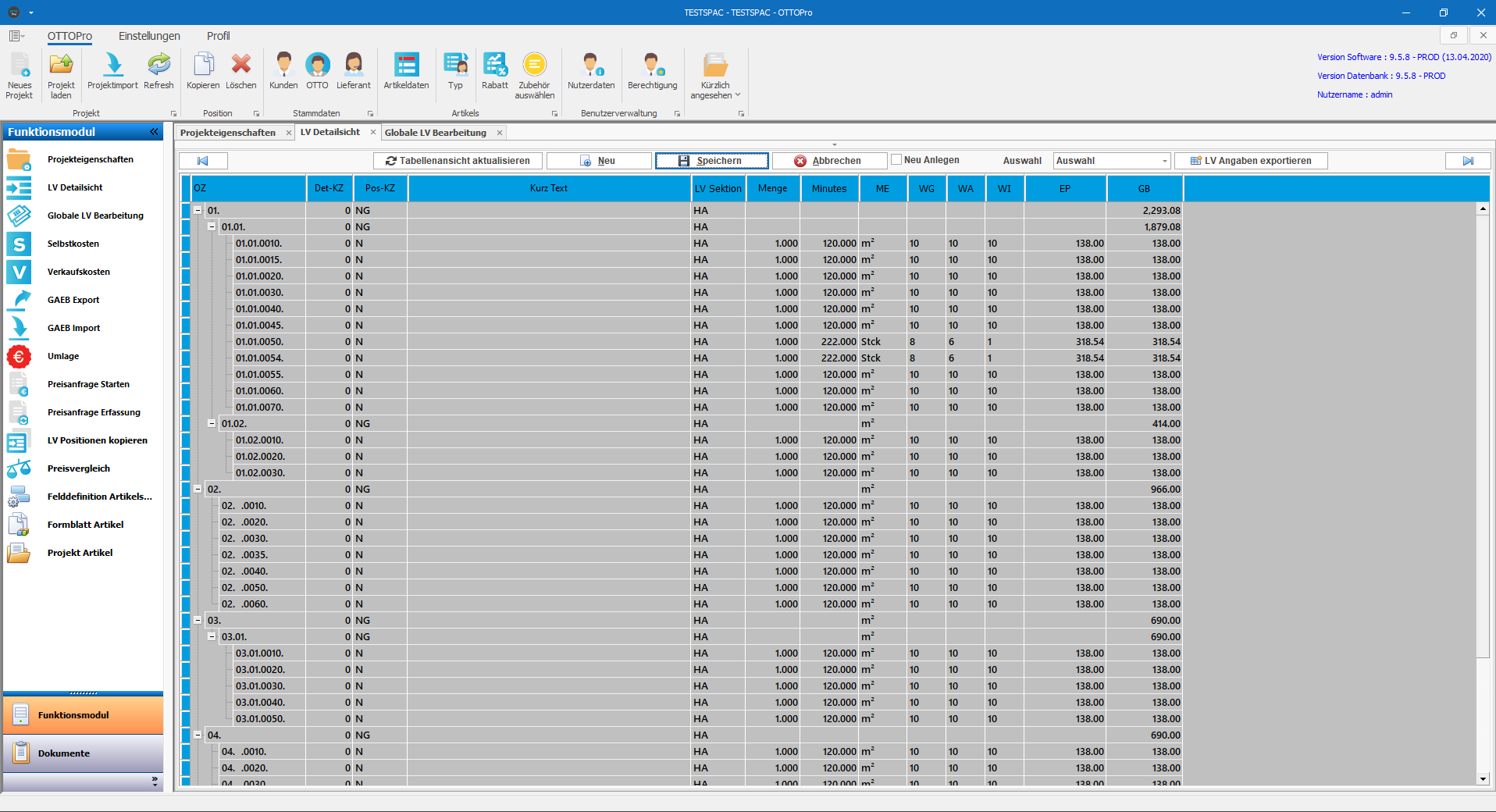
# Overview

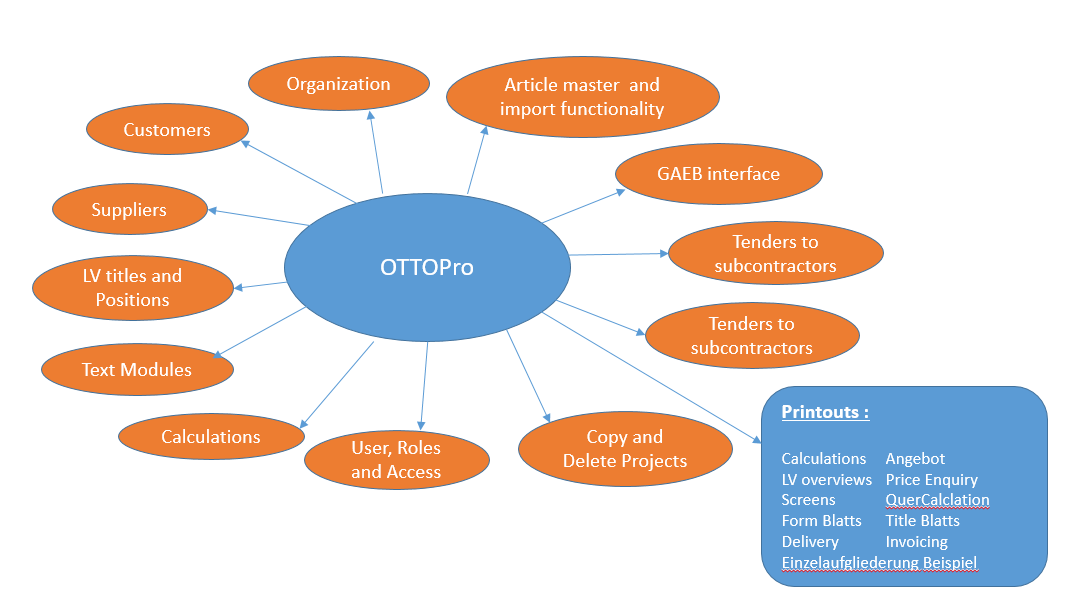
## Current System

OTTOPro is a calculation tool for construction industries. It also includes information of organization, customer, supplier, articles along with dimensions along with prices.

Project will have many sub modules like Project meta data, LV Positions, Bulk or Batch processing and few additional cost. We can also send proposals to customers and send price enquiry to suppliers.

There other modules like GAEB import and Export, DATANORM import, Submitting prices from suppliers to positions, delivery and invoicing.



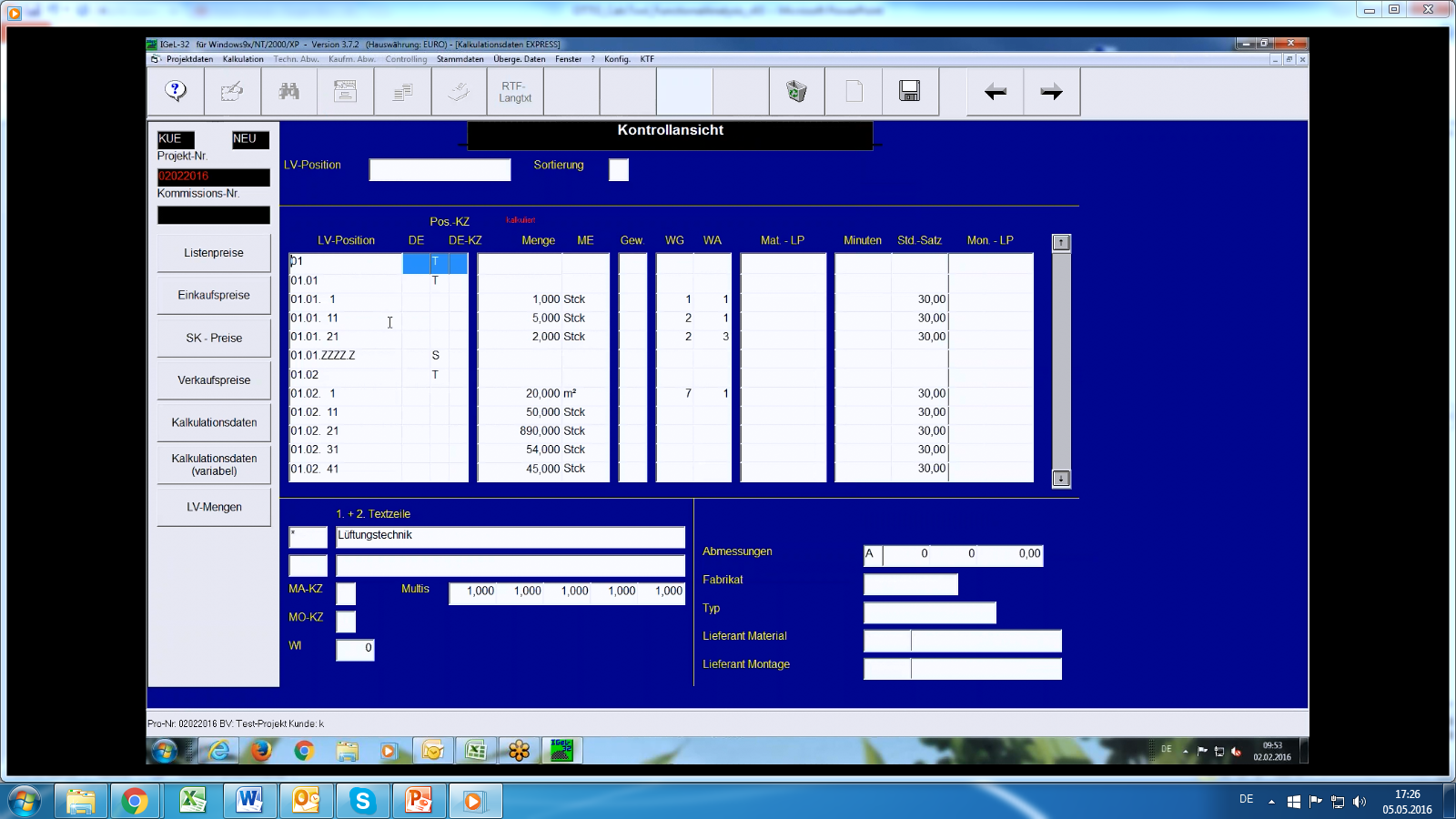


## Existing system(IGEL)

IGEL was developed using VB 5.0 with basic features. It is not so user friendly with current usage of application.

Here few major features not exist in IGEL

* Import and Export of XML version of GAEB files.
* Lang text with Images, media and links.
* View of LV details, Cost Details and tabular view in single form.
* View Position hierarchy.
* Grid printing and exporting.



# Technologies

* C#
* VB.net
* Windows Forms
* Microsoft SQL Server 2008 R2
* MS word templates
* Microsoft add in projects

# Third party tools

* DevExpress Windows Forms Controls 18.2.10
* GAEB Converter 9 and GAEB dll.
* Data Cross Media API for Importing Data Norm Files.

# IDE tools

* Visual Studio Professional 2015
* SQL Management Studio 2017

# Software requirements

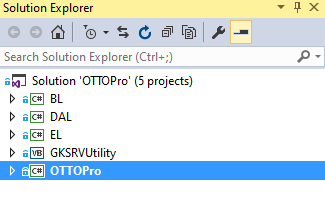
* Windows 7/8/10 (64-bit/32-bit)
* MS office 2013/2016/2019
* GAEB Converter 9

# Hardware Requirements

* Processor type: Pentium IV-compatible processor or faster
* Processor speed: Recommended: 2.0 GHz or faster
* Recommended RAM size: 2 GB or more

# Solution Architecture

We are using 3-tier architecture with centralized database. User can connect from multiple branches or locations. Below is the screenshot of solution explorer for reference.



## User Interface(OTTOPro)

When user requests on UI, this layer creates an object of respective class in entity layer and it binds data to properties, then it will send entity object to Business Layer.

Below is the list of objects in this layer.

* Windows forms.
* Reports (DevExpress Xtra Reports) in a folder.
* Datasets related to reports.

Each object will have it designer class and code behind class.

## EL (Entity Layer)

It contains entity class for each module and each class contains appropriate properties. These properties we will use for data transfer between layers. Naming convention will be like “ECustomer”,”ESupplier” and EProject etc…

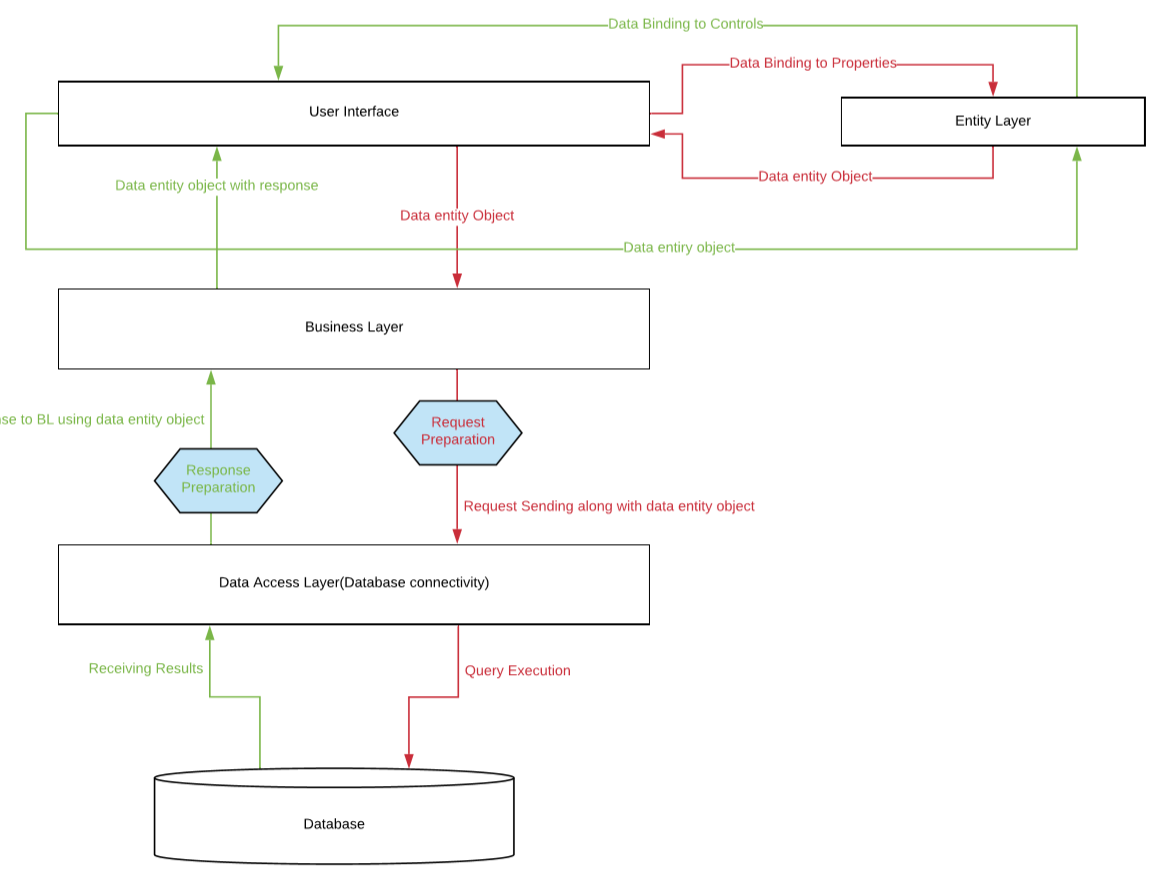
## BL (Business Layer)

It contains all the business classes separately for each module with proper naming convention. For example, if we have a customer module then business layer class name will be “BCustomer” in the same way few more examples “BSupplier” and “BProject” etc... As soon as data entity object comes from UI, it will prepare request and send to Data Access Layer.

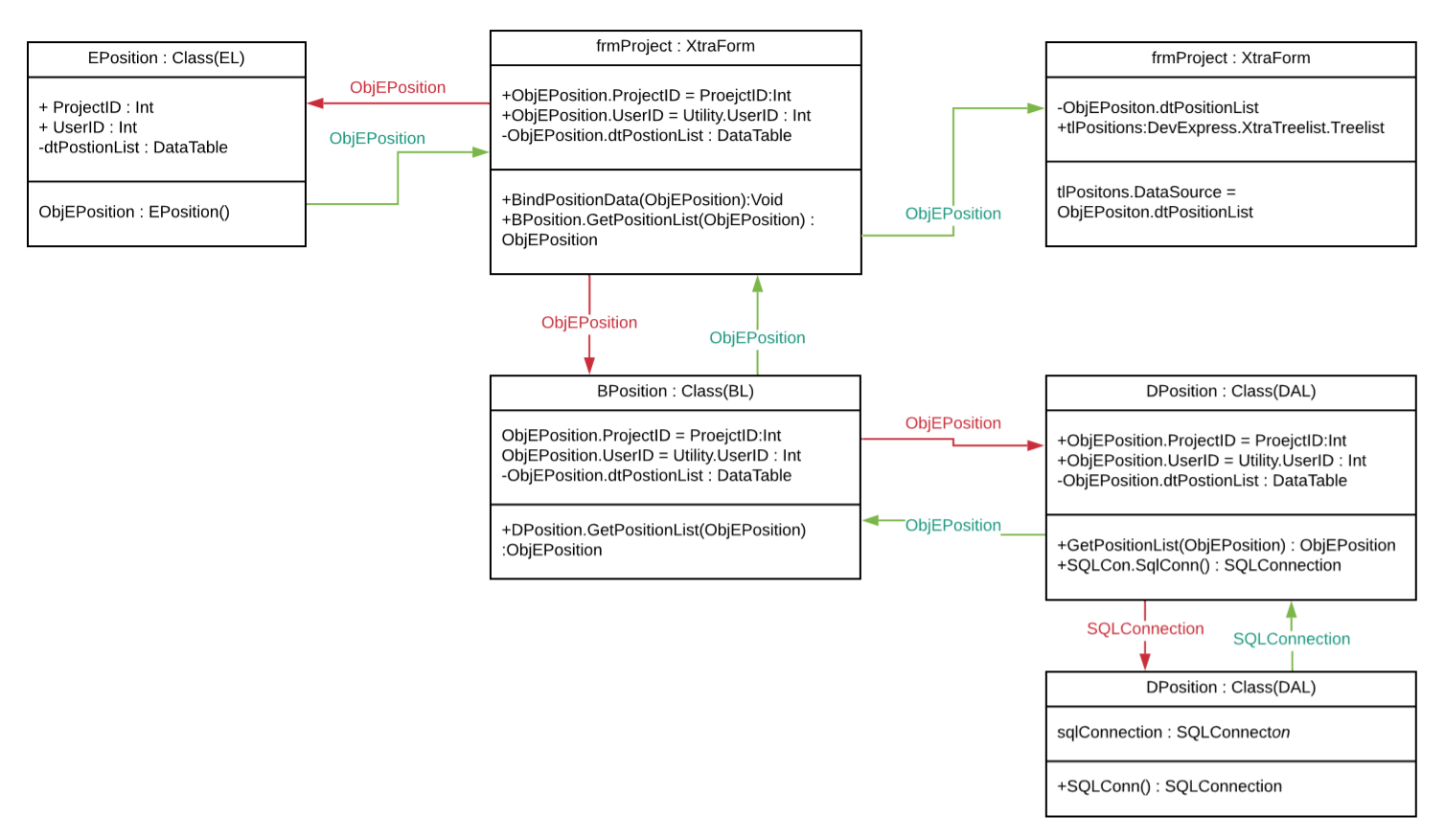
## DAL (Data Access /Layer)

It is a bridge between business layer and database. It will create a connection with database once request received from BL and executes the query by passing input parameters. There is a static call for SQL server connection.

Here is he graphical representation of solution.



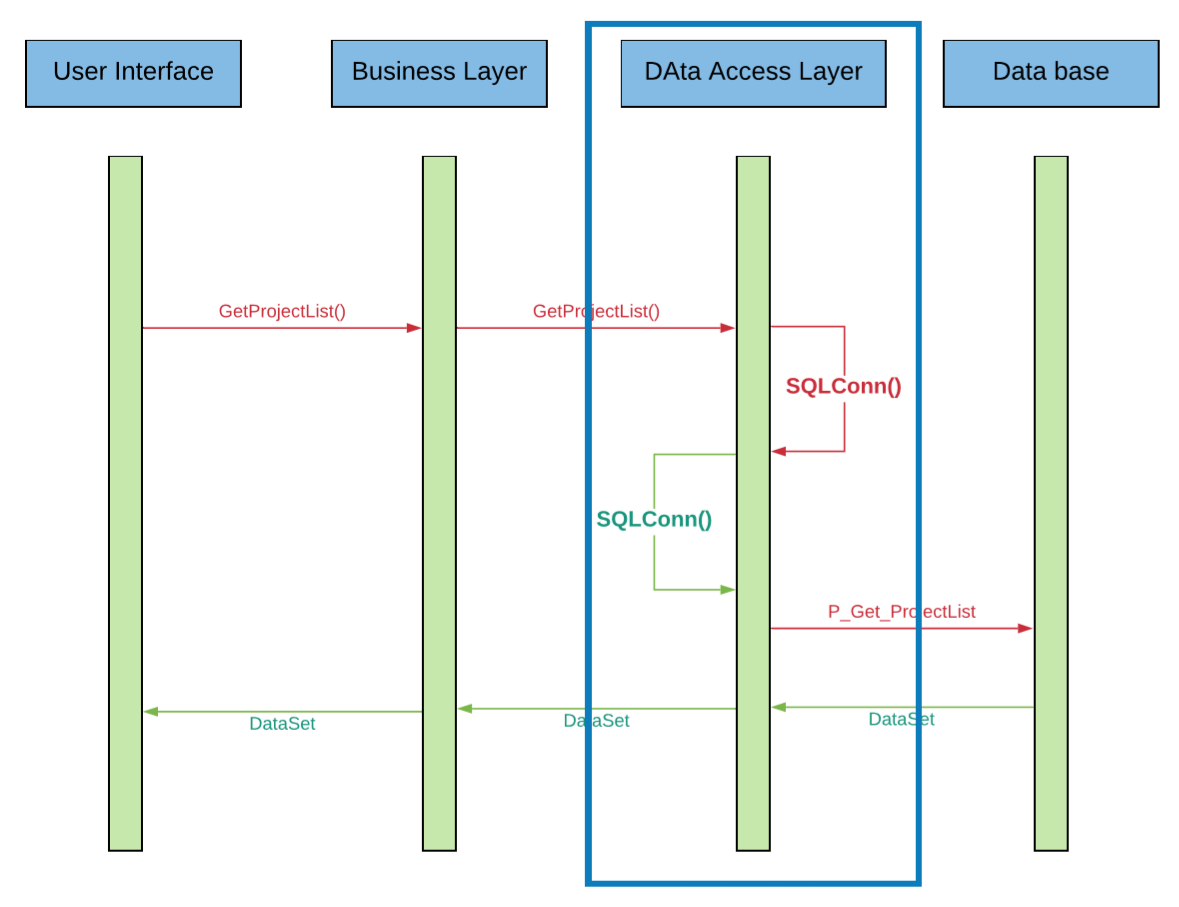
Here is a class diagram for one module.



### SQL Conn Class:

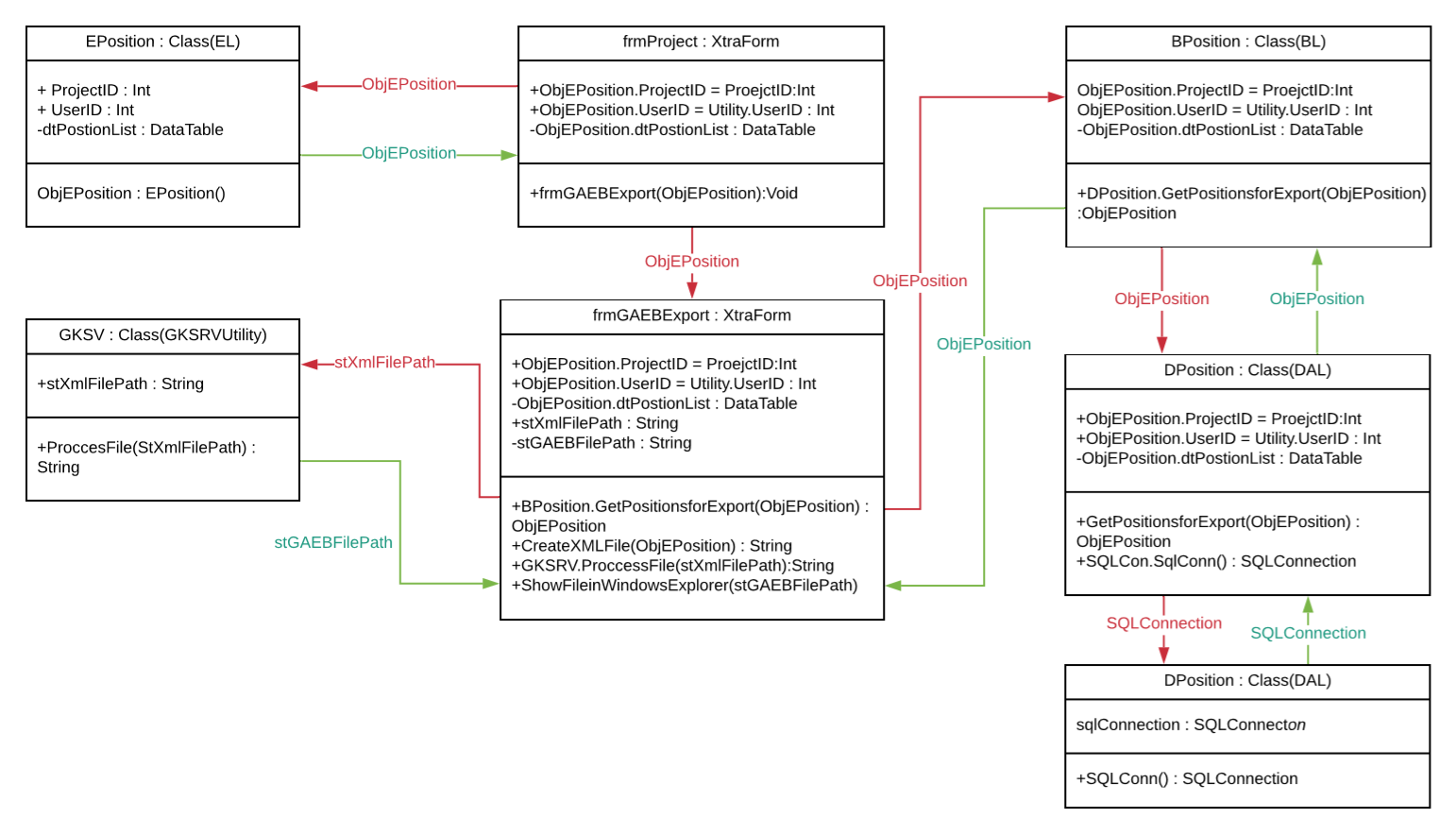
It contains two static SQL connections. One connection will be used for Login as it will run in one thread and other connection will be used for rest of transactions throughout application. Both connections will open till the application gets closed. Both connection states will be validated in each transaction with database in case of network failure or any other reason.





## GAEB Import and Export Utility(GKSRVUtility)

This project is specific to GAEB export and import functionality and it is implemented in vb.net because GAEB converter dll will support only vb.net.



# Validations and Messages

We are using DevExpress validations providers for validating entries as we have licensed DevExpress Controls.



# Internationalization, Localization and Translations

By default, DevExpress controls are having localization properties for all cultures and time zones. We are manually writing the translations for Labels, Message box descriptions and etc… As of now we are hardcoding these control captions manually with German language as OTTOPro will be used in only Germany region, but we will write translations dynamically based in culture and region once we go for the commercialization.

We can compare below screenshots of formatting decimal and date values which are captured in in English machine and German machine with same installer. As of now we are using only German language for labels but in future we will maintain resource files for each region.

**English machine screenshot:**



**German machine screenshot:**



# Database Design and Implementations

We are using SQL server 2012 Standard version for database development.

We have designed OTTOPro database with very good normalization and each table will have a primary key as default clustered index. Also we have created Foreign Keys, Unique Keys and etc... based on functionality requirements. SQL Transactions are used in procedures which are having dependent queries.

Below are objects used in Database implementation:

* Tables
* Stored Procedures
* Cursors
* Triggers
* Views
* User defined functions
* User defined table types
* Global and Local Temporary tables.

We did best performance optimization for major modules which are regularly used by users. Below are the methods used:

* Created Clustered and non-clustered indexes.
* Used normalization forms.
* Query restructuring.
* Comparing execution plans.

Few other modules pending for perform tuning those taken care in Phase-2 of OTTOPro.

# Security and Privacy Considerations

We have written a security class for encryption using Cryptography which is provided by .net Framework. And it is used to encrypt user’s passwords and other login details before saving into the database. In future we are going to implement same for database connection strings because we have placed connection string details in App.Config file. we can also use this class for any other sensitive information. There is no scope for SQL injection because we are using SQL stored procedures.

Here is the algorithm we are using for encryption and decryption using “CryptoStream” provided by .net frame work.



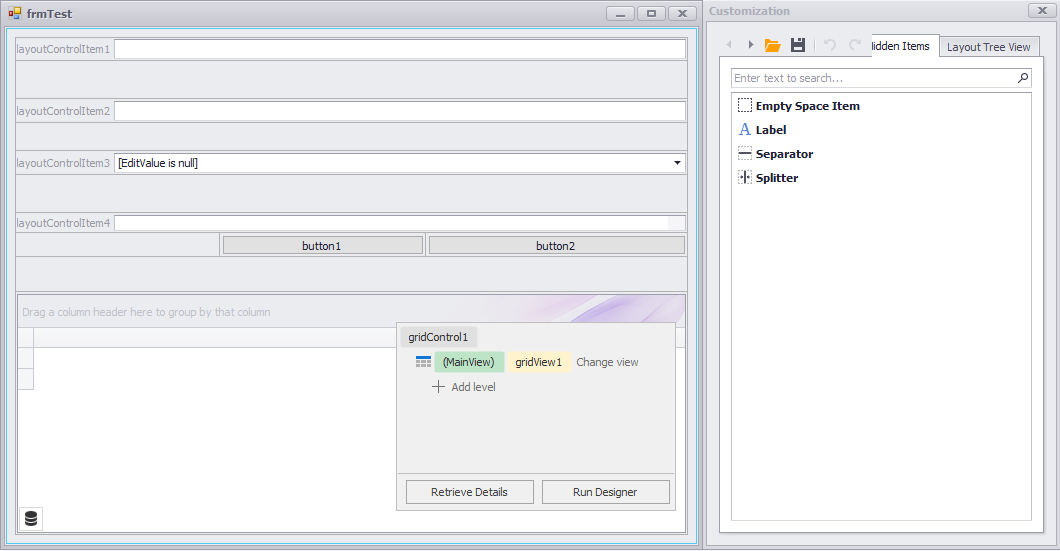
# Third Party Considerations

## DevExpress windows Forms controls

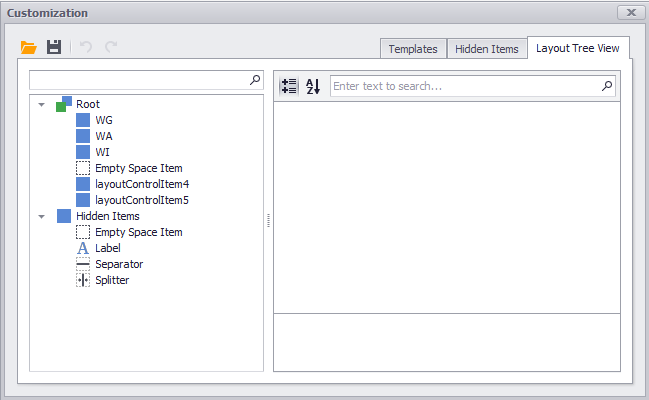
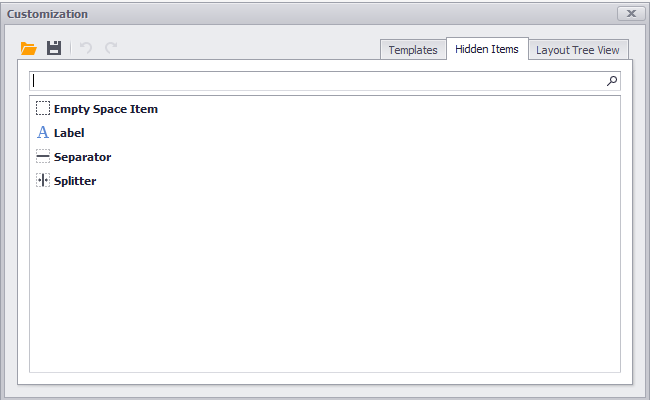
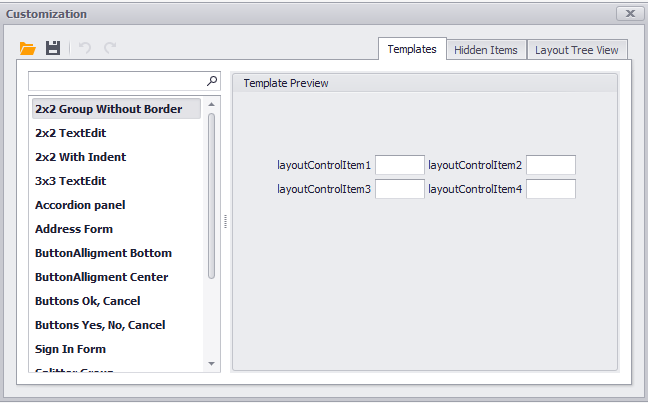
We used DevExpress windows Forms Controls Version 18.2 for better user interface and to provide more customization for end users. Below is the list of controls majorly used for OTTOPro.

### Layout control

Used for UI design. We can place all controls required in form and arranging them in proper manner. This will also provide us a customization form in design time for development and runtime for end user.



Here is the sample customization form.

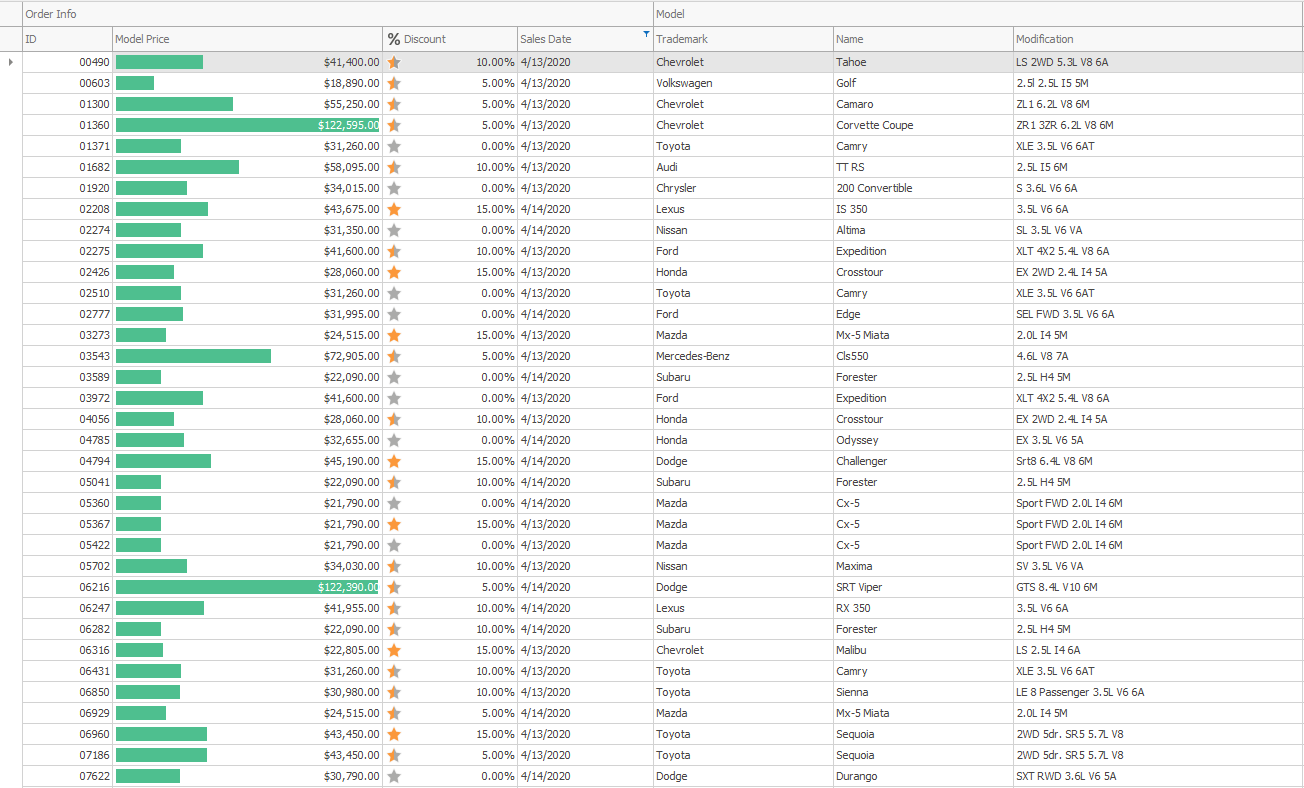


### Grid Control

Used for better data representation and it also has more customization options for end user like sorting with multi column, grouping on multiple columns, adding summary item from end user side etc. It also has inbuilt printing and reporting functionality.

You can choose between the following data presentation formats (Views) in the Data Grid

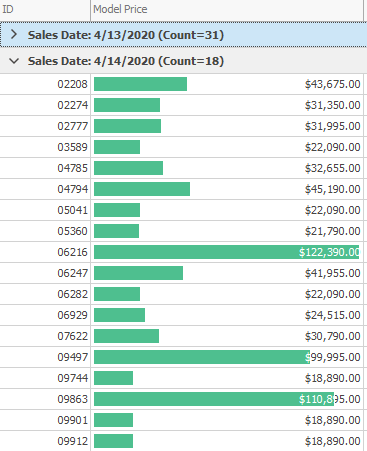
* [**Grid View**](https://documentation.devexpress.com/WindowsForms/3464/Controls-and-Libraries/Data-Grid/Views/Grid-View) - Displays data in a tabular form.
* [**Banded Grid View**](https://documentation.devexpress.com/WindowsForms/114637/Controls-and-Libraries/Data-Grid/Views/Banded-Grid-Views) - Displays data in a tabular form and allows grouping of columns into [bands](https://documentation.devexpress.com/WindowsForms/559/Controls-and-Libraries/Data-Grid/Visual-Elements/Banded-Grid-View-Elements/Band).
* [**Advanced Banded Grid View**](https://documentation.devexpress.com/WindowsForms/114637/Controls-and-Libraries/Data-Grid/Views/Banded-Grid-Views) - Displays data in a tabular form, allows grouping columns into [bands](https://documentation.devexpress.com/WindowsForms/559/Controls-and-Libraries/Data-Grid/Visual-Elements/Banded-Grid-View-Elements/Band) and supports complex data cell arrangements.
* [**Layout View**](https://documentation.devexpress.com/WindowsForms/5787/Controls-and-Libraries/Data-Grid/Views/Card-and-Layout-Views/Layout-View) - Presents records as cards, which can be displayed in one or multiple columns, one or multiple rows, in an ellipse (carousel mode) or a single card at a time. Supports complex card field layouts, built-in groups, tabbed groups and labels.
* [**Card View**](https://documentation.devexpress.com/WindowsForms/114638/Controls-and-Libraries/Data-Grid/Views/Card-and-Layout-Views) - Presents data records as cards, arranged down and then across. Card fields are always arranged in a single column.
* [**Win Explorer View**](https://documentation.devexpress.com/WindowsForms/114759/Controls-and-Libraries/Data-Grid/Views/WinExplorer-View) - Displays records using one of seven styles supported by MS Windows Explorer - Small, Medium, Large, Extra Large, List, Tiles and Content.
* [**Tile View**](https://documentation.devexpress.com/WindowsForms/114728/Controls-and-Libraries/Data-Grid/Views/Tile-View) - Displays records as read-only tiles, using one of the following layout modes: default (one or multiple columns/rows), list (without spaces between records) and [Kanban](https://documentation.devexpress.com/WindowsForms/117616/Controls-and-Libraries/Data-Grid/Views/Tile-View/Kanban-Board). This View provides the advanced field positioning feature, which helps you arrange fields relative to other fields, specify absolute or relative field display bounds, etc.



Here are the few operations we can perform on a grid control.

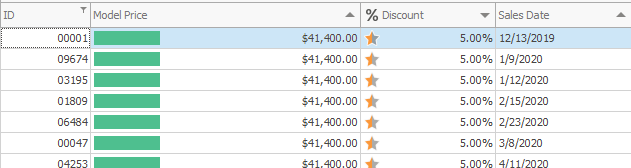
#### Grouping

* An end-user can group data at runtime by dragging a column to a dedicated Group Panel.
* Data group modes: by column values or display text.



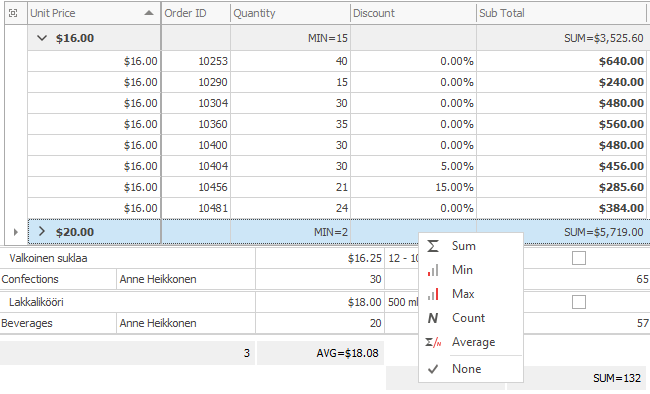
#### Sorting

* An end-user can sort data at runtime (in Grid Views, Banded Grid Views, Advanced Banded Grid Views, Layout Views, and Card Views).
* The Data Grid sorts its data itself and does not delegate this functionality to the underlying data source (except for server modes).
* Sort by values or display text.
* Implement custom sorting algorithms using an event.



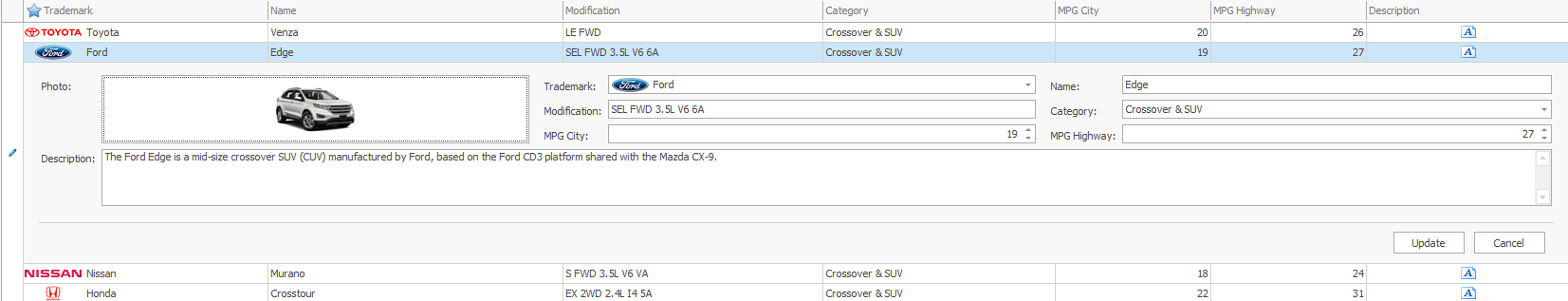
#### Summaries

* Total summaries (calculated against all rows)
* Group summaries (calculated against grouped data).
* Manual summary calculation



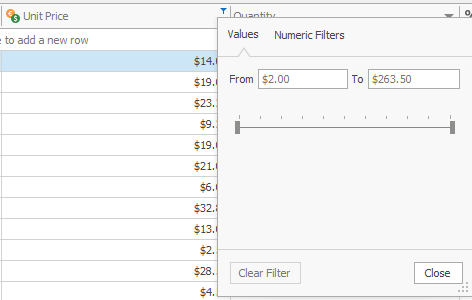
#### Edit Data

* Add and remove rows, in code and using a Data Navigator control
* Edit data with the help of 35+ in-place editors
* Edit data in a standalone (modal) or in-line edit form



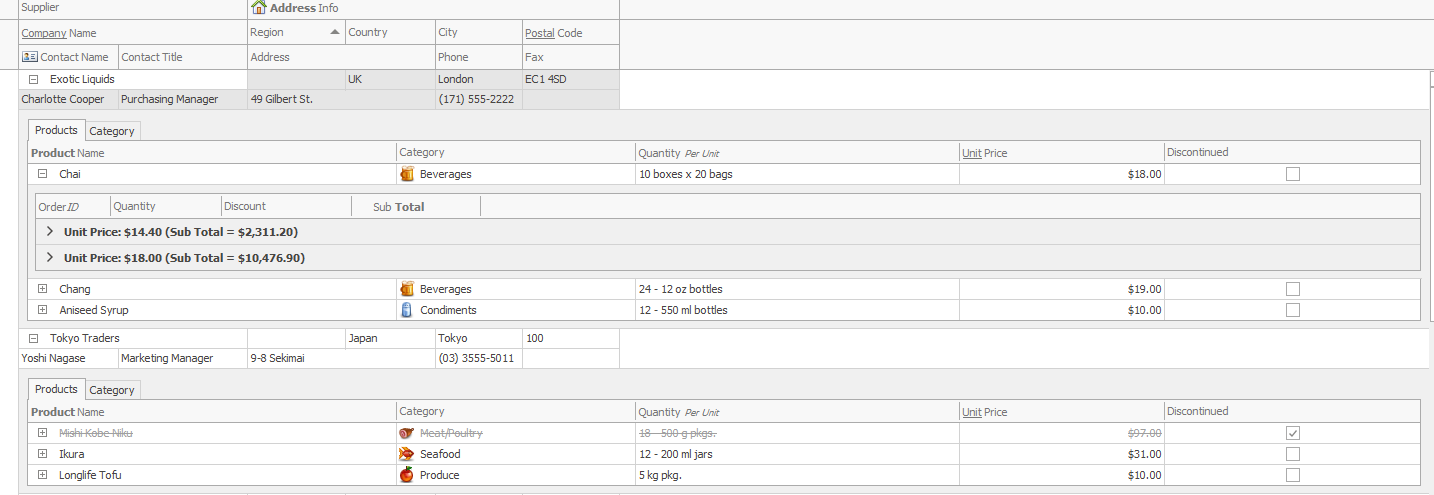
#### Filter and find rows

* Built-in Find Panel
* An end-user can filter data with column filter dropdowns
* Regular and MS Excel-inspired filter dropdowns



#### Master-Detail

The Data Grid supports hierarchical data sources, which consist of master and detail tables (lists) linked by one-to-many relationships. Here is sample screenshot of master detail view.

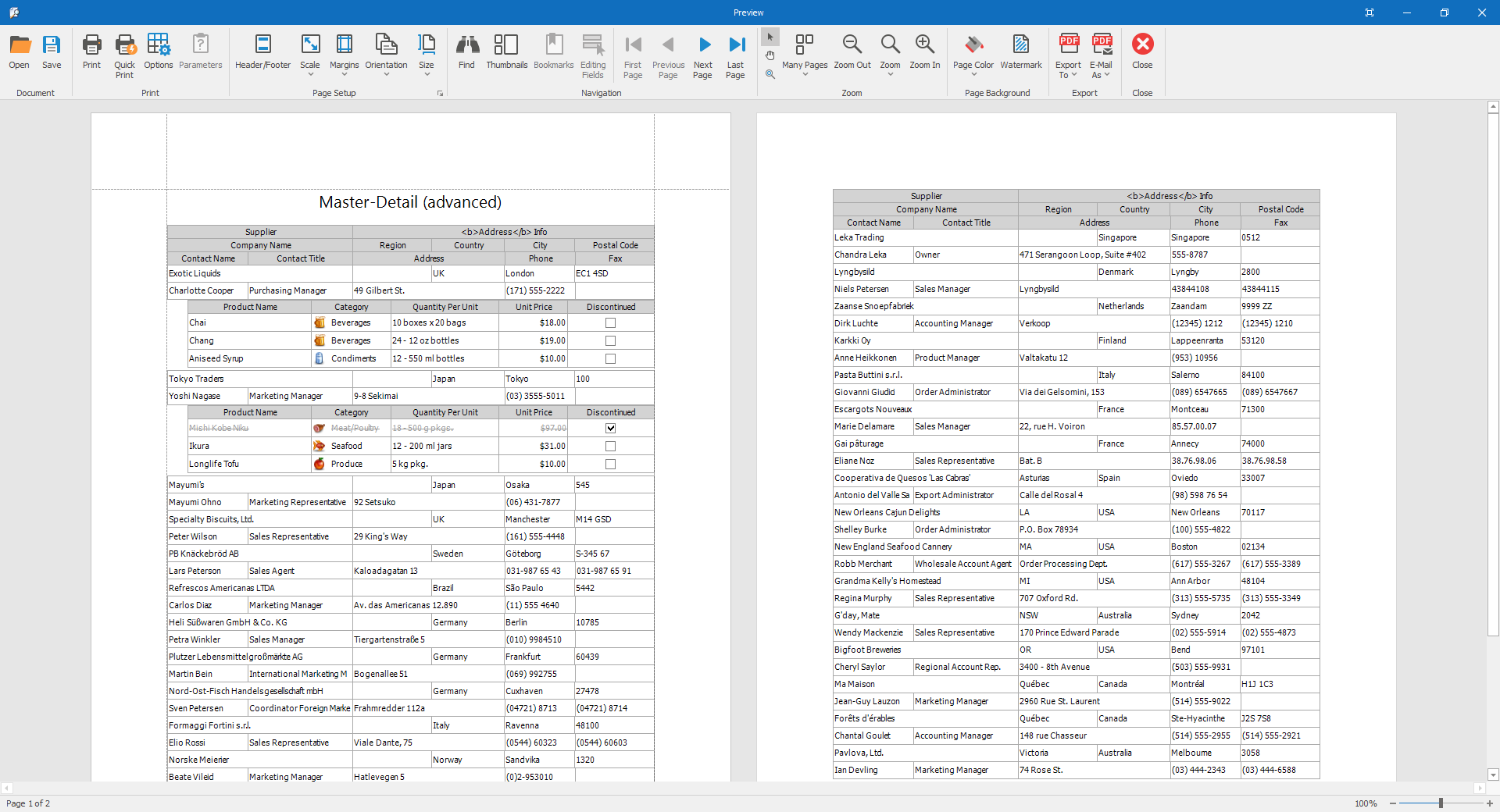


#### Grid Print

The Grid Control provides methods for printing displayed data, showing a preview window and invoking a print dialog. Grid Control provides direct methods for printing displayed data, these methods delegate the printing functionality to the [DevExpress Printing Library](https://documentation.devexpress.com/WindowsForms/2079/Controls-and-Libraries/Printing-Exporting).

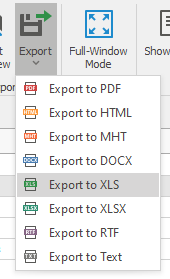
By default, a printed grid utilizes the same appearances as when it is displayed onscreen (these appearance settings can be customized for each View using [BaseView.Appearance](https://documentation.devexpress.com/WindowsForms/DevExpress.XtraGrid.Views.Base.BaseView.Appearance.property) property).However, each View provides print appearances used to paint the View's visual elements (buttons, headers, cells, etc.) in print output.

Below screenshot illustrates grid printing in all the cases.



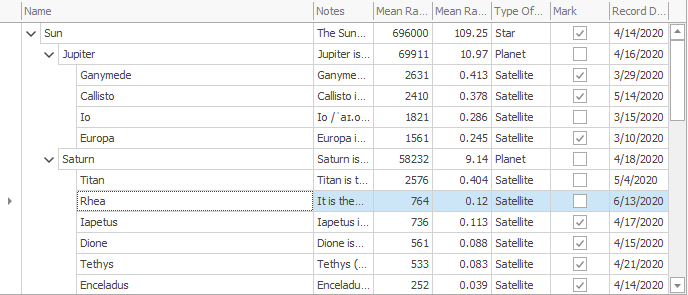
#### Grid Export

The Grid Control allows you to export its data to a file or stream in formats of pdf, excel, word and etc…



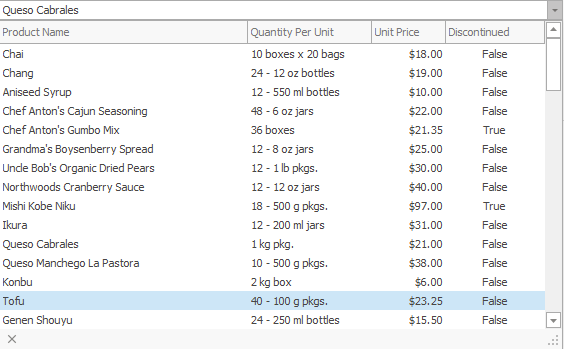
### Tree list:

Used for data representation in tree view with multiple columns. It will also support all functionality provided by grid control. Here is the sample screenshot of DevExpress Tree list.



### Lookup Edit

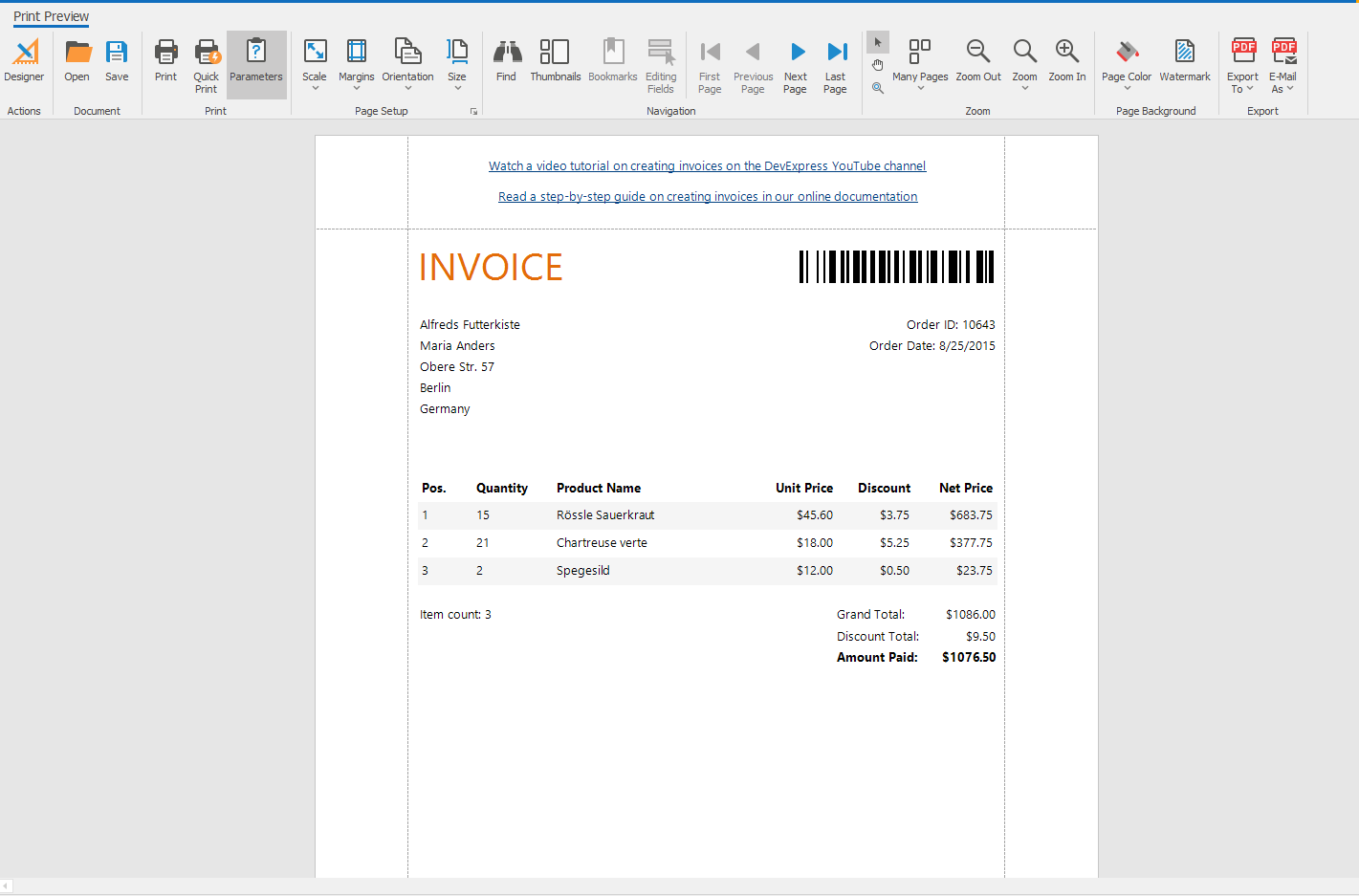
This is nothing but a combo box in windows forms controls. It will also have advanced features like combo box with multiple columns. It will also support functionality provided by grid control.



### Xtra Report

This is reporting suit provided by DevExpress windows forms controls. We can easily generate a report using this from a grid control or tree list. we can also design with expressions without any calculated fields.

Here is the sample screenshot of XtraReport and it will have the default print and export options.

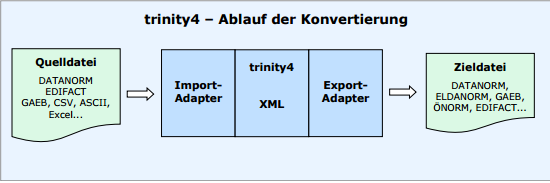


## GAEB Converter

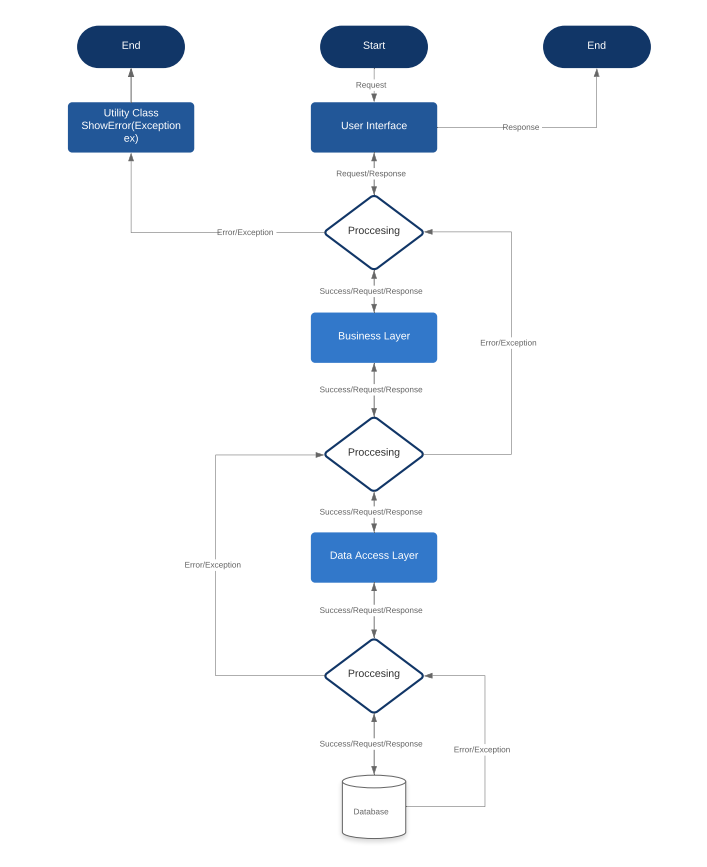
The GAEB-Converter is a software, that is able to create, edit and repair GAEB files. It also converts contract specifications between formats like GAEB 90, GAEB 2000, GAEB XML, Excel, Word, Access, PDF, RTF, dBase, Ö-Norm, Data-norm, UGL and REB. Licensed version of GAEB converter 9 we are using for import and export of GAEB files.

## Trinity DATANORM import

trinity is a software tool for converting standard formats to product data and project exchange formats and / or vice versa. Numerous adapters for standard formats (e.g. MS Excel) and data exchange formats (e.g. DATANORM, GAEB) from the construction industry are available for import and export. We integrate the trinity adapters into their application in a completely transparent manner for the user.



# Error and Exception Handling

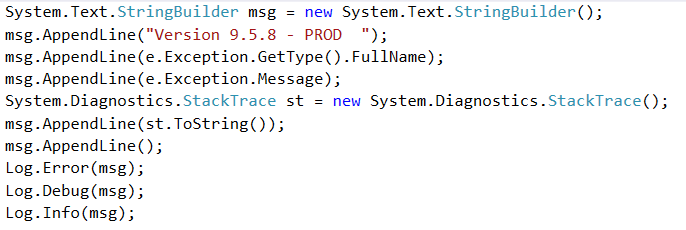
Error and exception work flow:

# Error and Exceptions Logging

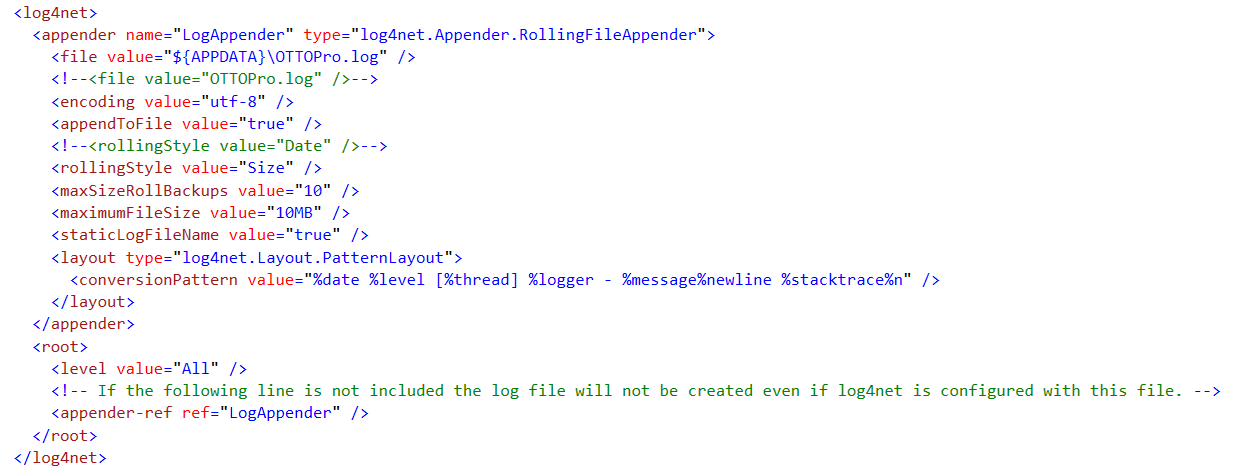
We are using log4net from nugget package to log debug information and Errors or Exceptions. All the properties related to log4net package are in App.Config file. We also have a functionality to send log file to production team which is hard coded, but in future we will be implementing configurable module for sending log file.

**Writing to log files:**

****

****

**Log4net configuration:**

****

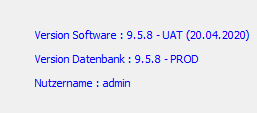
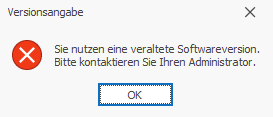
# Code structure and Coding standards

We have segregated the code across the solution i.e. each module will have separate classes in each layer. Each class will have multiple regions along with summary section for each work flow and this can be easily understandable by any new resource.



# Version Controlling on production

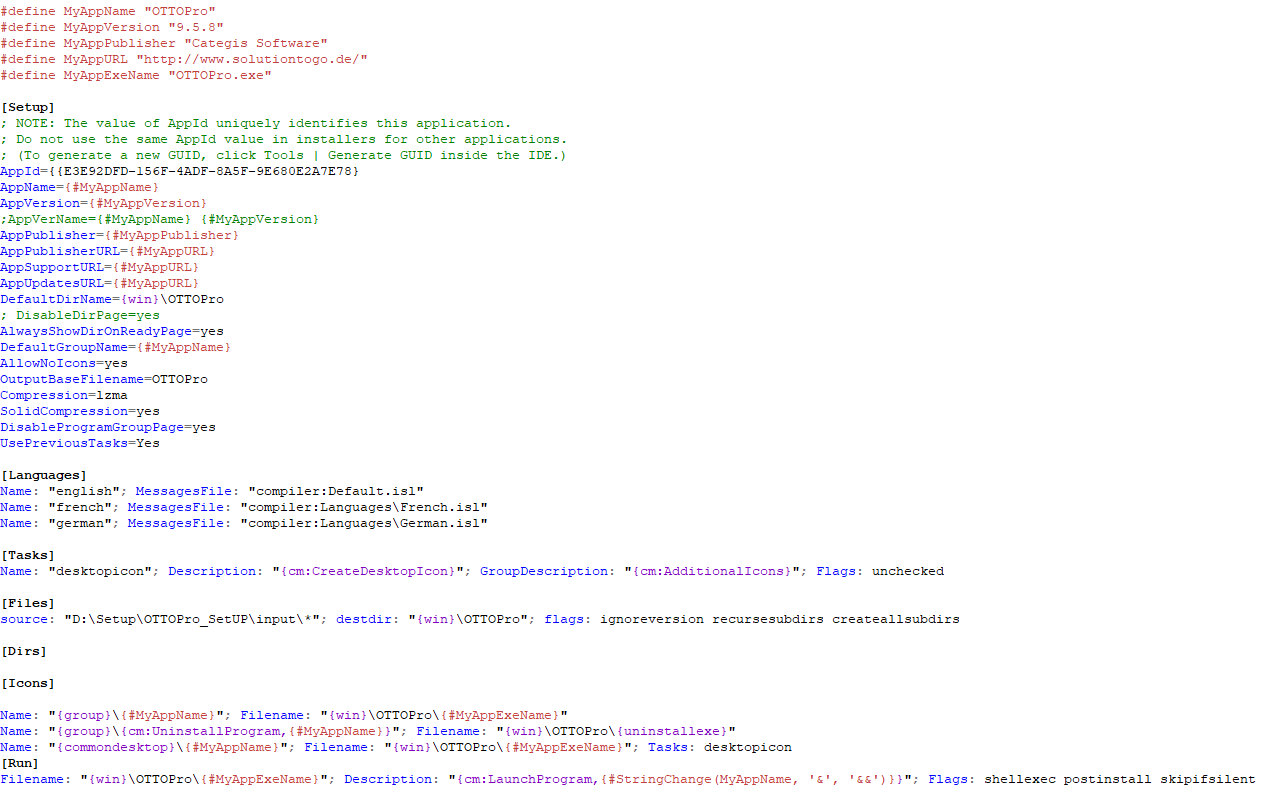
We using version matching system on production environment. We are saving version information in database. Whenever user will attempt login to OTTOPro, tool will check for version if the version info is matching then tool will get opened otherwise it will get closed by showing a message to user.

# Build

## Installer

We are using “Inno Setup” for creating installer which is a third party open source tool with a basic installation setting. We will also mention version info for installer in this file.



We can also customize installation using this tool like, directory selection, dll Registrations and Un Registrations, Registry reads and writes etc…

## Database deployment

While implementation of backlog items of OTTOPro, we will note down the changes in database and write scripts for the same in below way.

### Table

In case of new columns or changing datatype of columns, we will write alter scripts on specific table and also write scripts for existing data adjustments.

### Stored procedures, Triggers, Views, Keys, User defined functions and user defined types

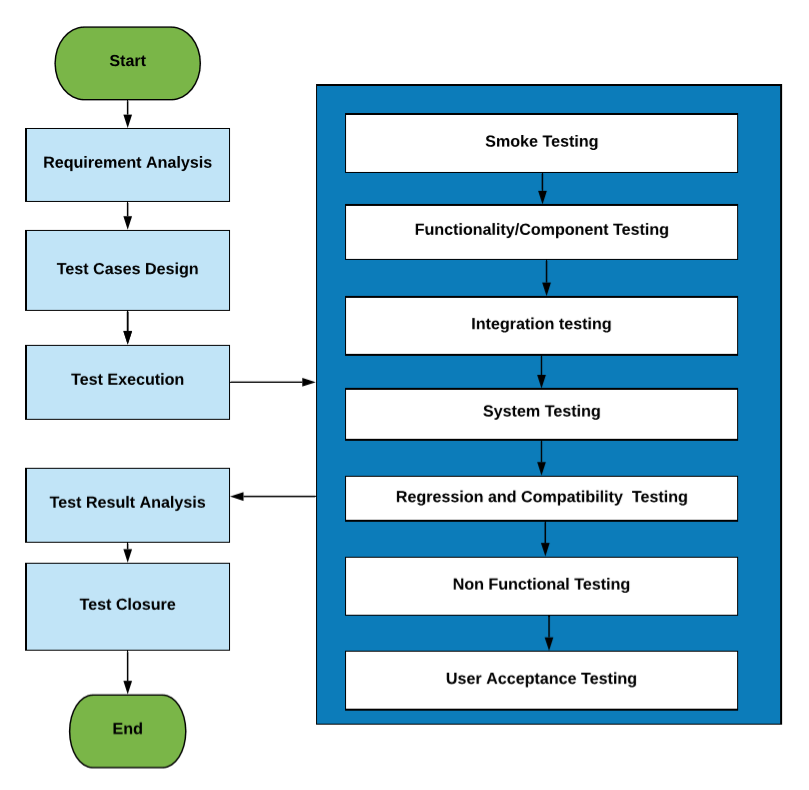
We will write drop and create with validations for all the objects which are changed as they will not have any physical data.

# Test Objective

The objective of this document is to define and specify the different kinds of the testing that is performed on the OTTO tool.

# Testing Techniques

Testing technique is the systematic process that is followed for testing an application. Each and every phase of the testing carried out has different kinds of the checks performed on the application.



## Functional testing

### Functionality testing

Testing the independent individual components of the form/module is functionality testing.

* Text boxes, List boxes, checkboxes and buttons.
* Complete form and its functionality.

Test cases will be derived based on the Boundary value analysis, Equivalence partition and error guessing.

### Integration testing

Testing the flow of the data between the two modules is the Integration testing.Integration testing will be carried out when a new module will be added to the application.The new module needs to be checked for the correct data flow from the old module to the newly added module and vice versa.

### System testing

Testing the end to end business logic of the application is the System testing. The complete business flow will be tested. This might involve not only one module but many modules.

### Platform/ compatibility testing

Testing the application on multiple different platforms i.e., different operating system, different Language settings different hardware and software components.

### Smoke and sanity testing

Smoke and sanity are the build verification testing that is performed at the different stages of the release cycle.

#### Smoke testing

This is carried out on the application as and when the application has been released from the Development to the testing. This is performed to check whether the basic and the critical features are working or not, and to check whether the build provided is testable or not.

#### Sanity testing

This is carried on the application to check the correctness of the new functionality added in the application. This is to test the all the scenarios of the functionality in deep.

## Nonfunctional testing

### Performance testing

This is carried out on the application to make sure that when multiple users are using the application at the same time, what is the correctness of the tool to give the appropriate results to the end users.

On the OTTO tool we have performed the concurrent testing by making 8-10 users to work on the application at the same time.

* All the users working on the same module at a time in different projects.
* All the users working on the same project in different modules.

## Maintenance testing

### Regression testing

When there will be a major change to any of the module/a new module has been added to the application/changes to the Database scheme in these cases the impact on the application will be wide.

Hence a complete test of the application should be performed to make sure that the latest changes have not affected any other modules of the application.

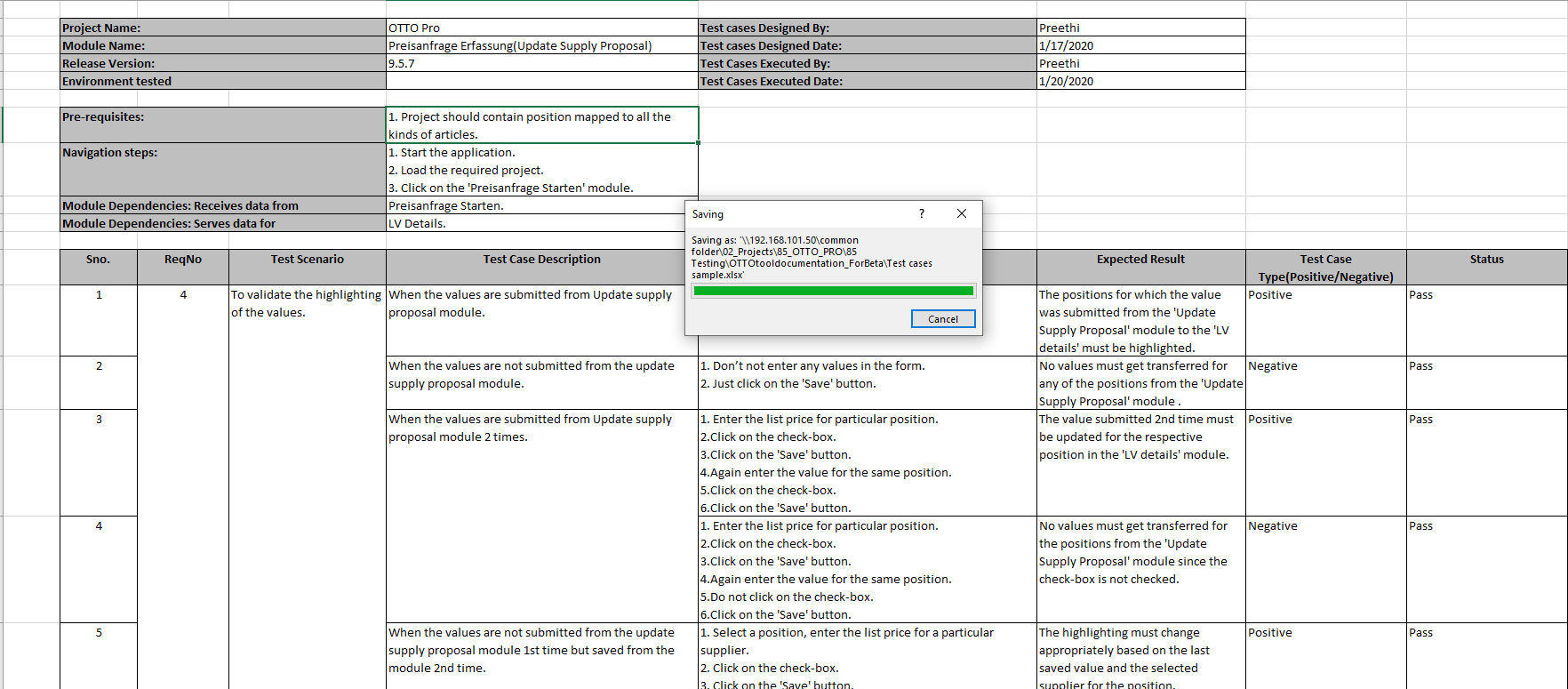
# Testing Deliverables

## Requirement clarifications

When any clarification required on the requirements this will be added in the document and sent across to the team requesting for the clarifications.

## Test cases

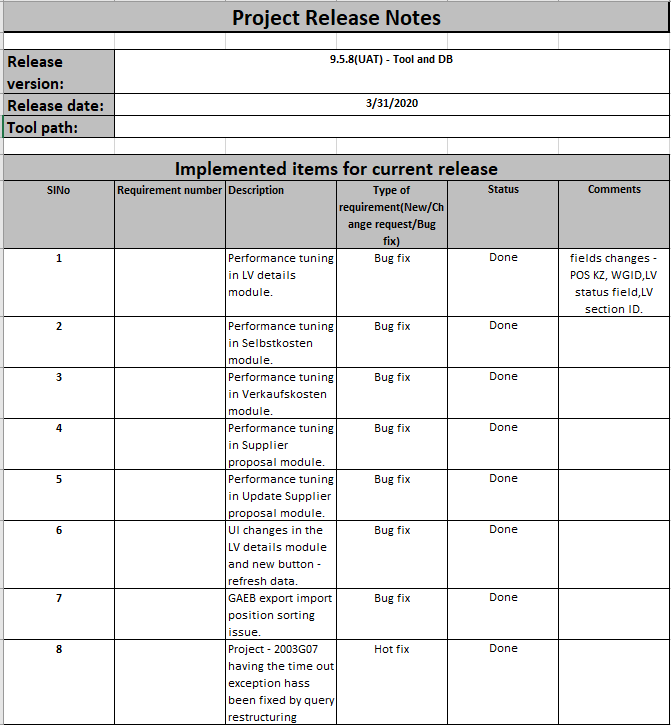
All the scenarios for testing the application will be documented in an Excel workbook segregated based on the modules.



## Release notes

This document will the proof of releasing the application to the client. The information contained in the release notes document is:

* Functionality added newly to the application
* Known issues
* Platform tested on and platforms supported by the application
* Scope of the next release.



# Testing Entry and Exit Criteria

## Entry criteria

* Testing platform must be setup prior to the test execution.
* Test cases of the functionality that is in scope of testing.
* Application must be available for testing.

## Exit criteria

* All the test cases documented must be executed.
* There must be no blocker or the critical bugs.

# Defect Management

If the application is not working as per the requirements or if there is an any deviation from the requirements, then it will be as a defect/bug.

As and when the defect is faced by the tester then the defect tracking/management needs to be followed. This is done using the defect tracking tool “Bugzilla”.

Different stages in the Defect life cycle are:

* **New:** When a new issue has been added to the tool, it will be in the new state.
* **Open:** When the issue is assigned to the respective developer it will be in the open state.
* **Fixed:** If the issue is accepted and fixed by the developer then the status of the issue will be changed to Fixed.
* **Deferred:** If the issue is not accepted by the developer then there are different reasons
* **Not an issue/Invalid:** This could be because of the misunderstanding of the requirements or change in the requirements which are not updated to the tester.
* **Duplicate:** The issue has been already raised earlier.
* **Not reproducible**: The issue is not able to be reproduced by the developer.
* **Next release**: The issue has been deferred for the next release because of the low severity and the priority of the issue.
* **Re-Testing:** Once the issue has been fixed, the tester will re-test the application to check whether the issue has fixed and working correctly or not.
* **Re-Opened**: If the issue is not yet fixed then change the status to re-open.
* **Verified and closed:** If the issue has been fixed and working correctly then the status will be changed to Verified and closed

