Technical design document

[eNeg] – Phase [phase 1.]

Content

[1. Document Contents 5](#_Toc220150272)

[1.1. Document Purpose 5](#_Toc220150273)

[1.2. Document Stockholders 5](#_Toc220150274)

[1.3. Document Organization 5](#_Toc220150275)

[2. System Overview 6](#_Toc220150276)

[2.1. System Overview 6](#_Toc220150277)

[2.2. System Scope 6](#_Toc220150278)

[2.3. Assumptions and Dependencies (Not Mandatory) 6](#_Toc220150279)

[2.4. General Constraints 6](#_Toc220150280)

[2.5. Goals and Guidelines (Not Mandatory) 7](#_Toc220150281)

[2.6. Used Technologies 7](#_Toc220150282)

[2.7. Used Frameworks 7](#_Toc220150283)

[2.8. Special Hardware Requirements (Not Mandatory) 7](#_Toc220150284)

[2.9. System Context (Not Mandatory) 7](#_Toc220150285)

[3. System Requirements 8](#_Toc220150286)

[3.1. System Stakeholders/Users 8](#_Toc220150287)

[3.2. Stakeholders/Users Requirements 8](#_Toc220150288)

[3.3. Use Cases 9](#_Toc220150289)

[4. System Architecture 10](#_Toc220150290)

[5. System Sub-System (Not Mandatory) 11](#_Toc220150291)

[6. Detailed System Modules/Working Packages 12](#_Toc220150292)

[6.1. Facebook Application 12](#_Toc220150293)

[6.1.1. Friends Module 12](#_Toc220150294)

[6.1.2. Album Module 12](#_Toc220150295)

[6.2. Data Access Tier 13](#_Toc220150296)

[6.2.1. DB Abstraction Classes 13](#_Toc220150297)

[7. System Objects 14](#_Toc220150298)

[7.1. User Object 14](#_Toc220150299)

[7.2. Photo Object 14](#_Toc220150300)

[8. Database Schema / ERD 16](#_Toc220150301)

[8.1. User Table 16](#_Toc220150302)

[8.2. Album Table 16](#_Toc220150303)

[8.3. Photo Table 16](#_Toc220150304)

[9. System Components Integration/Interaction 18](#_Toc220150305)

[9.1. Client / Web Application Communication 18](#_Toc220150306)

[9.2. Data Access Tier / Database Communication 18](#_Toc220150307)

[9.3. Web Application/Data Access Tier Communication 18](#_Toc220150308)

[10. System Objects Interaction/Usage 20](#_Toc220150309)

[11. System Sequence Diagram 21](#_Toc220150310)

[11.1. Login Service 21](#_Toc220150311)

[12. Time Plan 23](#_Toc220150312)

[13. Glossary 24](#_Toc220150313)

[14. References 25](#_Toc220150314)

# Document Contents

This document contains the Technical Design Specifications (TDS) for eNeg project.

This project considered as one of the most important services that handles all online negotiations between multiple users.

The stakeholders for this document are solution architect, project owner, developers, QA, QE, and technical consultants.

## Document Purpose

*This document is a technical design document for eNeg project. It illustrates eNeg scope, requirements, technical design. The users of this document are system stakeholders and developers.*

## Document Stakeholders

* ***Project owner:*** *Sander Van de Rijdt.*
* ***Solution Architect****: Khaled Mohammed Ramadan.*
* ***Technical Team Lead****: Khaled Mohammed Ramadan.*
* ***Technical Team****:*
  + *Ahmed Dergham.*

* ***QA & QE Team****:*
  + *Ali Ibrahim.*

* **Consultant to Product Owner Technical**: Thomas Ott.
* **Consultant to Product Owner SRS:** Ibrahim Imam.

## Document Organization

Document Organization sections shows the organization of the document. This section in most cases is written after finishing the whole document and determining the whole document section.

*e.g. This document is divided to 16 Section;*

*\* Section 2 shows system overview.*

*\* Section 3 shows system users and user requirements.*

*\* Section 4 shows system architecture...etc*

# System Overview

## System Overview

eNeg Negotiation Framework is a system that has two main targets:

1. Collect as much information as possible from different users‘communication channels via Add-ons (left side of the figure) and store it centralized in one place (middle of the figure).
2. Use this collected information as input for data analysis through Apps (right side of the figure) and allow these Apps to provide valuable support information to users.

The Framework contains different sub-systems to achieve these targets; in a first phase there will be:

1. A Desktop Add-on that can collect information.
2. A Mobile Add-on for mobile devices (developed by the Vienna University of Technology).
3. A Web Platform that shows all collected information centralized at one place.
4. An App that analyses the collected information in terms of preferences.

In future the system should be flexible and extendable to any communication technology and to any type of negotiation support via Apps.



## System Scope

Target users of the system will be all kind of users that negotiate online; e.g. via E-Mail or Instant Messenger etc. Currently users don’t have a possibility to collect and document all information at one place no matter which communication technology they are using. Currently it is also not possible to analyze these messages and give adequate negotiation support to users.

Online negotiators can be purchasers, suppliers, sales-men, lawyers, HR agents, business-men etc. that need to negotiate about goods and services that are purchased and sold, contract details, salary packages, co-operation agreements, etc. and already do this using electronic media such as E-mail.

Core features of the Negotiation FrameWork are:

* To allow users to track and collect information messages from different communication channels (e.g. from Outlook)
* Activate negotiation support when the users need it
* Select a suitable set of support Apps that fits to the current negotiation event or situation
* Adjust the settings of the support Apps to assure good support
* Receive negotiation support messages (provided by Apps) during ongoing negotiations
* View and manage negotiations (ongoing, past)
* See all details of a negotiation event

## Assumptions and Dependencies

*eNeg system assume that all users have Silverlight installed on their machines because most of applications use Silverlight..*

## General Constraints

Windows 7+

Silverlight 4+

Discourse Based Technology (used by Vienna University of Technology for Mobile Add-on)

SQL Server 2008+/ IIS 7.5+

VS2010.

Silverlight4 tools.

MEF.

MVVM Light.*.*

## Used Technologies

* Silverlight4.
* C#4.
* MEF.
* MVVM Light.
* RIA Service.
* ADO.Net Entity Framework.

## Used Frameworks

* .Net Visual Studio 2010
* Managed Extension Framework (MEF).

## System Context

The three parts of the Negotiation Framework must transfer data between each other. The user sends data via the Desktop Add-on interfaces to the Web Platform. This data is used by the Apps to give the user support. The results of the Apps are sending back to the Platform and the user can access the results at the App, via the Platform directly or via the interfaces in his Add-ons. The exact data, which must be transferred, is defined later on, but in general it is numbers, text and graphics.

As it is planned to allow any developers to also create Apps for the Framework there must be a clear and easy to understand documentation of the API to inform how to interact with Add-on and Web Platform.

# System Requirements

## System Stakeholders/Users

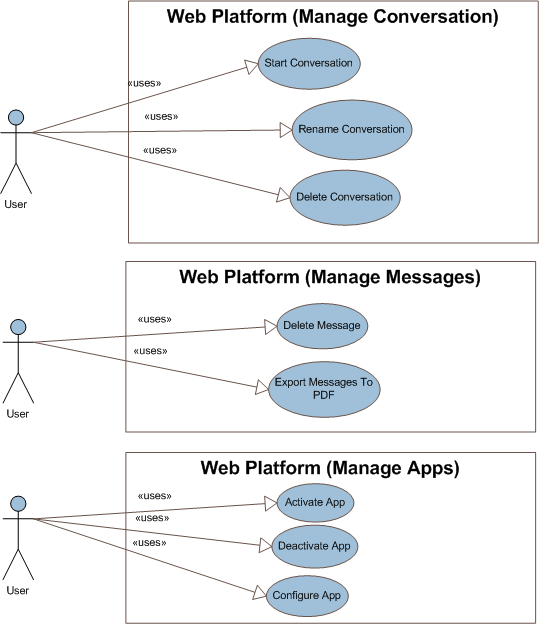
|  |  |
| --- | --- |
| ***Stakeholder Name*** | ***Roles*** |
| *Site Admin* | * *Has the role to administrate the Users* |
| *User* | * *Has the role to negotiate with other people* |

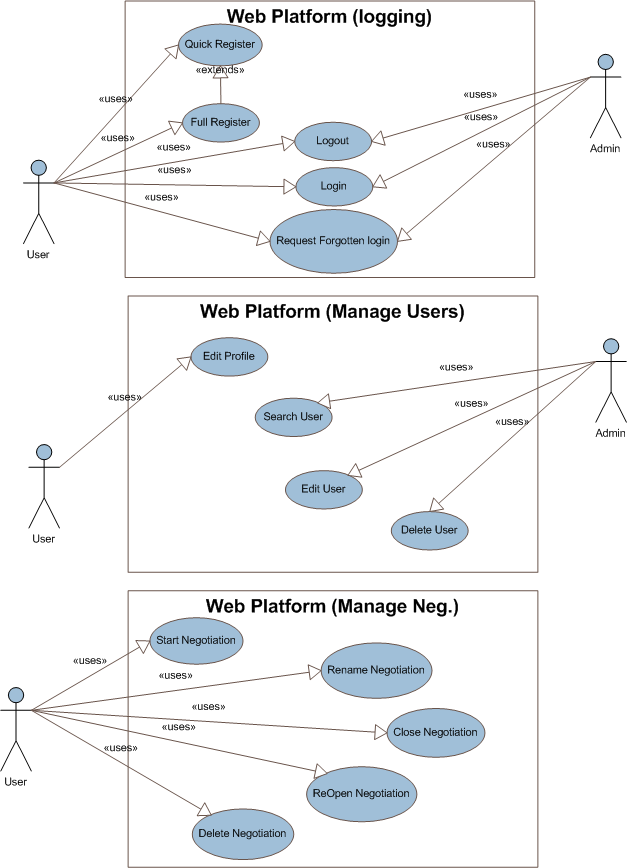
## Stakeholders/Users Requirements

|  |  |
| --- | --- |
| ***Stakeholder Name*** | ***Services*** |
| *Site Admin* | ***Web Platform***   * *Login* * *Request forgotten Login* * *Manage (see, edit, delete, search/filter) Users* * *Manage (see, edit, delete) „My profile“* * *Logout* |
| *User* | ***Add-on***   * *Login* * *Manage (see, start, close, re-open, rename, delete) Negotiations* * *Manage (see, activate/deactivate, configure) Apps* * *Add text inputs* * *Logout*   ***Web Platform***   * *Quick Register* * *Full Register* * *Login* * *Request forgotten Login* * *Manage (see, start, close, re-open, rename, delete) Negotiations* * *Manage (see, delete, export) Conversation details* * *Manage (see, activate/deactivate, configure) Apps* * *Manage (see, edit, delete) „My profile“* * *Logout* |

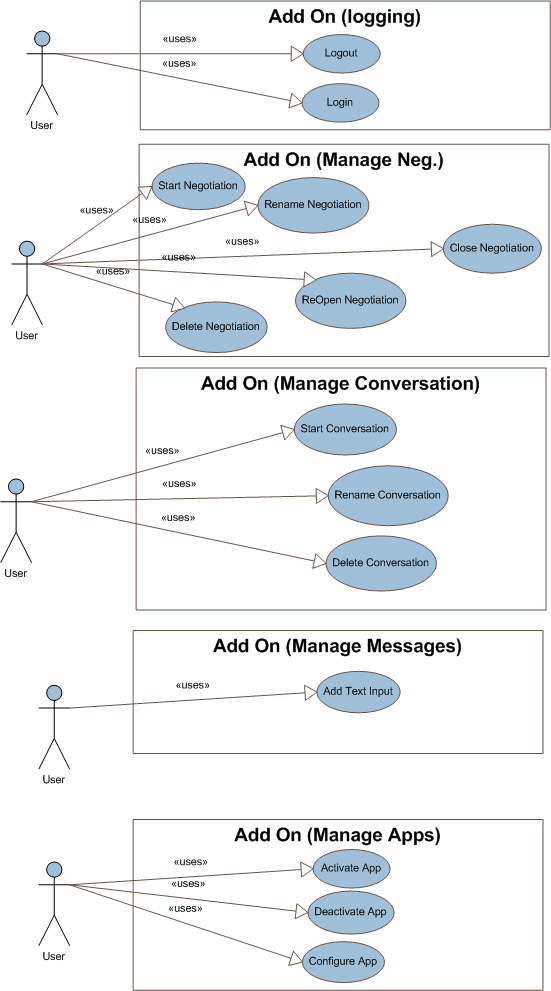
## Use Cases

### 3.3.1 Web Platform Use Cases



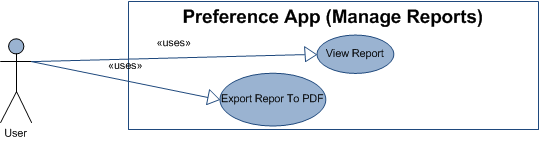


### 3.3.2 Addon Use Cases

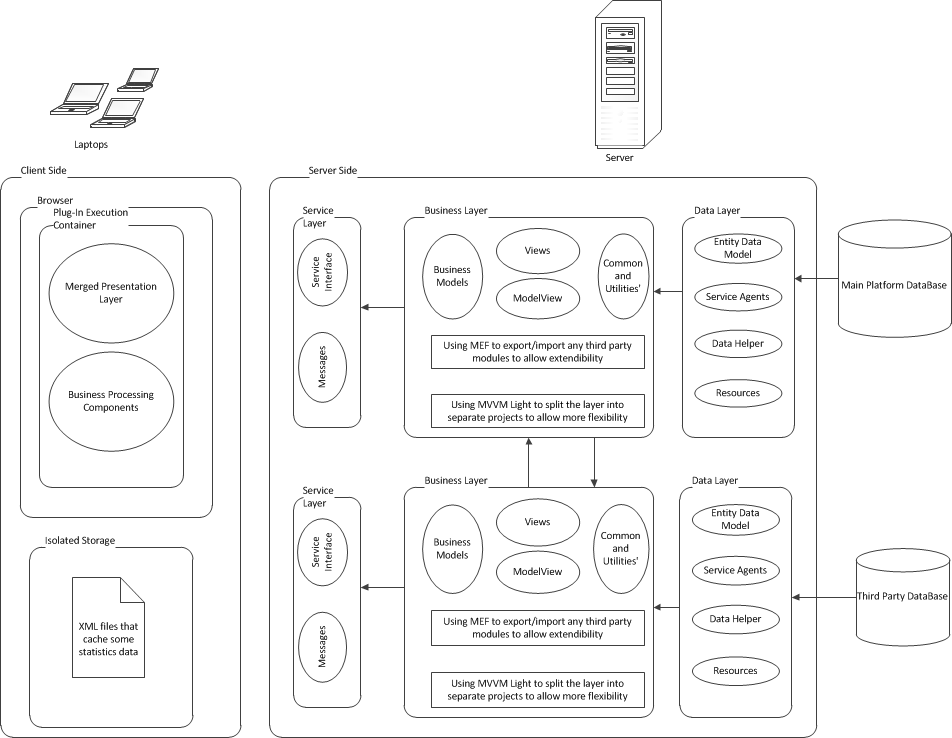


### 3.3.3 Preference Apps

### 



# System Architecture

The architecture for this Silverlight service will be built upon the concept of RIA [Rich Internet Application], because the target of this service is to support a huge amount of users with the corresponding requests in no time.  
So we need to make this project as fast as possible, and for that reason we need to load our modules on demand to save time consuming and also memory, and resource usage.

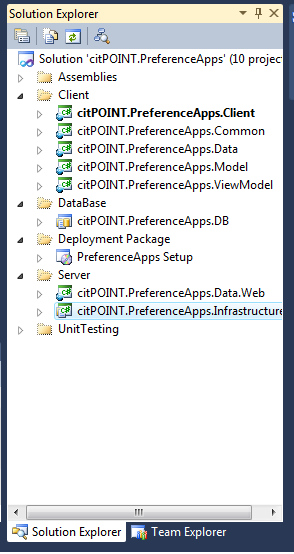
# Detailed System Modules/Working Packages

This project consists of 2 main parts as the following, of course it contains another part which is Add-On (Desktop Application) but this part will be exported from the eNeg Platform which means it will be done implicitly during the implementation of the main Platform.

## eNeg Platform



## Preference Apps



## eNeg Platform

*This section aims to list the modules inside eNeg Platform. Each module will provide list of services that should be provided by this module.*

### Negotiation Module

*This module is used to deal with Negotiation. This module provides the following services;*

***1. Start Negotiation:*** *This service enables the user to start new negotiation.*

***2. Close Negotiation:*** *This service enables the user to close a specific negotiation.*

***3. Reopen Negotiation:*** *This service enables the user to reopen the selected closed negotiation.*

***4. Delete Negotiation:*** *This service enables the user to delete a specific negotiation.*

***5. Rename Negotiation:*** *This service enables the user to rename a specific negotiation.*

### Conversation Module

*This module aims to deal with Conversations. This module should provide the following service;*

***1. Start Conversation:*** *This service enables the user to create a new conversation.*

***2. Delete Conversation:*** *This service enables the user to delete selected conversation.*

***3. Rename Conversation:*** *This service enables the user to rename existing conversation.*

### Message Module

*This module aims to deal with Messages that are used during conversation or negotiation. This module should provide the following service;*

***1. Delete Message:*** *This service enables the user to delete selected message.*

***3. Export Message:*** *This service enables the user to export selected message to PDF.*

### Apps Manipulation Module

*This module aims to deal with* the manipulation for the third party apps*. This module should provide the following service;*

***1. Activate App:*** *This service enables the user to activate selected app.*

***2. Deactivate App:*** *This service enables the user to deactivate selected app.*

***3. Configure App:*** *This service enables the user to configure existing app.*

### User Manipulation Module

*This module aims to deal with* the manipulation for the new or existing user*. This module should provide the following service;*

***1. Edit Profile:*** *This service enables the user to modify his profile.*

***2. Search User:*** *This service enables the site admin to search for specific user.*

***3. Edit User:*** *This service enables the site admin to edit data for a specific user.*

***4. Delete User:*** *This service enables the site admin to delete a specific user.*

### User Authentication Module

*This module aims to deal with* the authentication and authorization for users*. This module should provide the following service;*

***1. Quick Register:*** *This service enables the user to quickly register to the system.*

***2. Full Registration:*** *This service enables the user to register for the system.*

***3. Log In:*** *This service enables both of the site admin and user to log in to the system with the respective privilege.*

***4. Log Out:*** *This service enables both of the site admin and user to log off from the system.*

***5. Request New Credential:*** *This service enables both of the site admin and user to request new username and/or password.*

## Preference Apps

*This section aims to list the modules inside the Preference Apps . Each module will provide list of services that should be provided by this module.*

### User Authentication Module

*This module aims to handle the authentication and authorization for preference app users. This module will contains the following classes;*

1. ***Log In:*** *This service will check firstly if the user already logged in to the main system then it will logged in user directly to the preference app else it will request the credential from the user.*
2. ***Log Out:*** *This service enables user to log out from the preference apps.*

### Sets Management Module

*This module aims to manage different kind of sets for the preference app . This module will contains the following classes;*

1. ***Create Preference Set:*** *This service enables the user to create new preference set.*
2. ***Rename Preference Set:*** *This service enables the user to rename existing preference set.*
3. ***Delete Preference Set:*** *This service enables the user to delete existing preference set.*

### Issues/Values Management Module

*This module aims to handle manage issues for corresponding preference sets and values of those issues. This module will contains the following classes;*

* 1. ***Add Issue:*** *This service enables the user to add new issue.*
  2. ***Edit Issue:*** *This service enables the user to edit existing issue.*
  3. ***Delete Issue:*** *This service enables the user to delete existing issue.*
  4. ***Add Value:*** *This service enables the user to add new value.*
  5. ***Edit Value:*** *This service enables the user to edit existing value.*
  6. ***Delete Value:*** *This service enables the user to delete existing value.*

### Data Matching Module

*This module aims to handle manage issues for corresponding preference sets and values of those issues. This module will contains the following classes;*

* 1. ***Add Negotiation to Preference Set:*** *This service enables the user to assign existing negotiation to the existing preference set.*
  2. ***Add and Match Text Input:*** *This service enables the user to drag and drop text from any message to the app and matching it to the specified preference sets.*
  3. ***Add Undefined Issue Data:*** *This service enables the user to add undefined issue data.*
  4. ***Add Undefined Option:*** *This service enables the user to add undefined option.*
  5. ***Add Related Later Issue:*** *This service enables the user to issue of type related later.*

### Report Management Module

*This module aims to manage reports that are generated from the preference app. This module will contains the following classes;*

* 1. ***View Report:*** *This service enables the user to view the generated report.*
  2. ***Export Report:*** *This service enables the user to export the generated report as PDF.*

# System Objects

The aim of this section is to define all objects inside the system. The output of this section is all system objects listed in sections. Each section will show Object Name, Object Description, Object Attributes, and Object List of Services that it will provide. Finally figure with all system objects is important.

*e.g. This section shows the objects that will be used in Facebook application. Each object will be listed separately in one section. The section will contains Object Name, Object Description, Object Attributes, and Object List of Services that it will provide.*

## User Object

*This section aims to define User Object. This object is used to identify users of system. This object has the following properties;*

1. ***Name:*** *name of user.*
2. ***Mail:*** *mail of user.*
3. ***Password:*** *password of user.*

*This object provides the following functions;*

1. ***Login:*** *to login to system.*
2. ***Change Password:*** *to change user password.*
3. ***Update Profile:*** *to update his profile.*
4. ***Join Group:*** *to join group.*

## Photo Object

*This section aims to define Photo Object. This object used to identify photos that are used by inside the application. This object contains the following properties;*

1. ***Photo Title:*** *title of photo.*
2. ***Photo Width:*** *width of photo.*
3. ***Photo Height:*** *height of photo.*
4. ***Photo ID:*** *album that have this photo.*

*This object provides the following functions:*

1. ***Tag Person:*** *to tag person in the photo.*
2. ***Add Comment:*** *to add comment on photo…etc*

*Next figure shows object/class diagram*

******

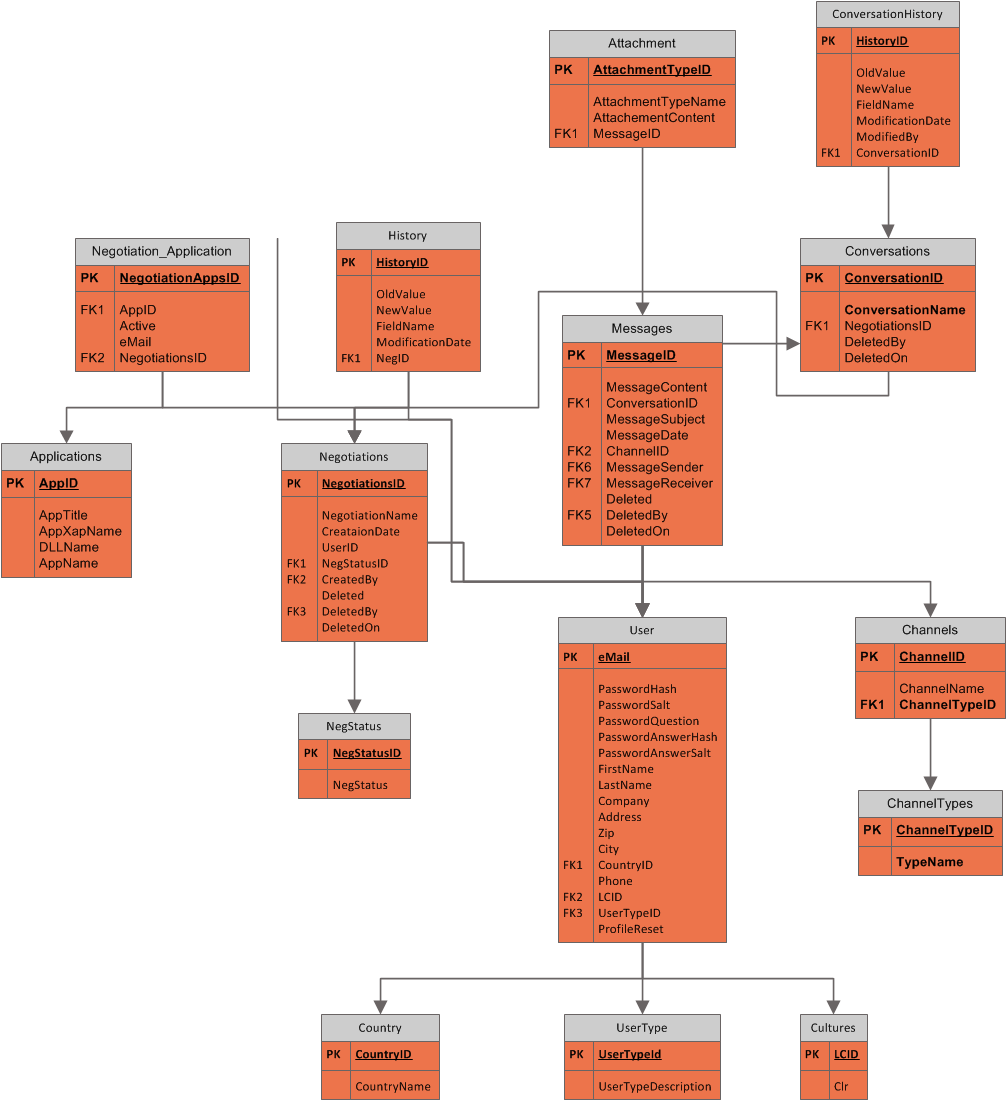
# Database Schema / ERD

## eNeg Web Platform

This is the schema for the main platform, and it consists of some tables that will be described in more details in next sections.

**Table List**

1. User
2. Country
3. Culture
4. Negotiation
5. NegStatus
6. Conversation
7. Application
8. Channel Type
9. NegHistory
10. Negotiation\_Application
11. Channel
12. Conversation History
13. Message
14. Attachment
15. UserType



## User Table

*This table is used to represent eNeg user. This table contains the user data that will be used in the application. This table has the following fields;*

1. *eMail,nVarchar (300)(PK): this field represents user name.*
2. *PasswordHash,nVarchar (50): this field represents hash for password.*
3. *PasswordSalt, nvarchar(50): this field represents salt for password.*
4. *PasswordQuestions,nvarchar(100): this field represents password question.*
5. *PasswordAnswerHash,nvarchar(100): this field represents hashing for password question answer.*
6. *PasswordAnswerSalt,nvarchar(100): this field represents password question answer salt.*
7. *First Name, nvarchar(50): this field represents user first name.*
8. *UserTypeID,int: represent the userd role type.*
9. *Last Name, nvarchar(50):this field represents user last name.*
10. *Company, nvarchar(100): this field represents company name.*
11. *Address, nvarchar(200): this field represents user address.*
12. *Zip,varchar(10): this field represents user zip code.*
13. *City, nvarchar(50):this field represents user city name.*
14. *CountryID(FK), Int: this field represents the ID for the country that user belongs to.*
15. *Phone, varchar(30): this field represents the user phone no.*
16. *LCID (FK), int: this field represents the id for the user favorite language.*
17. *.*

## CountryTable

*This table represents All Countries over the world. This table consists of the following fields;*

1. *CountryID(PK), Int: this field represent Country ID.*
2. *Country Name: nVarchar (100): this field represent country name.*

## Cultures Table

*This table represents All Cultures over the world . This table consists of the following fields;*

1. *LCID(PK), Int: this field represent Culture ID.*
2. *CodeBaseName: nVarchar (50): this field represent culture name.*

## Negotiation table

*This table represents All Negotiations that will be saved during the application. This table consists of the following fields;*

1. *NegotiationID(PK),GUID: this field represent NegotiationID.*
2. *NegotiationName: nVarchar (150): this field represent Negotiation name.*
3. *CreatedBy(FK),GUID: this field represent the user id that created this negotiation.*
4. *CreationDate,DateTime: this field represent the negotiation start date.*
5. *NegStatusID(FK), Int:this field represent the status for this negotiation.*
6. *Deleted, bit: this field used as a flag to mark this negotiation as deleted or no.*
7. *DeletedBy(FK), GUID: this field represent the user that deleted this negotiation.*
8. *DeletedOn,DateTime: this field represent the deletion date of the negotiation.*

## NegStatusTable

*This table represents All status for a specific negotiation . This table consists of the following fields;*

1. *NegStatusID(PK), Int: this field represent Status ID.*
2. *NegStatus: nVarchar (50): this field represent the status.*

## ConversationTable

*This table will used to save all conversations that will be happened during application. This table consists of the following fields;*

1. *ConversationID(PK), GUID: this field represent Conversation ID.*
2. *ConversationName: nVarchar (100): this field represent conversation name.*
3. *NegotiationID(FK),GUID:this field represent the id for the corresponding negotiation.*
4. *Deleted, bit:this field used as a flag to mark this record as deleted.*
5. *DeletedBy(FK),GUID: this field used to specify which user deleted this conversation.*
6. *DeletedOn, DateTime: this field used to spcify when this conversation deleted.*

## ApplicationTable

*This table represents All third party apps that will be used during application . This table consists of the following fields;*

1. *ApplicationID(PK), GUID: this field represent Application ID.*
2. *ApplicationName: nVarchar (100): this field represent user control name of the specified silverlight control.*
3. *ApplicationTitle, nvarchar(100): this field represent the title of the application.*
4. *ApplicationXapName, nvarchar(100): this field represent the name of the XAP file for this silverlight application.*
5. *DllName, nvarchar(100): this field represents the dll name of the application.*

## ChannelTable

*This table represents All Channel types. This table consists of the following fields;*

1. *ChanneTypeID(PK), Int: this field represent channel type ID.*
2. *TypeName: nVarchar (100): this field represent channel type name.*

## NegHistoryTable

*This table represents All history data for the negotiation table, it acts like log table . This table consists of the following fields;*

1. *HistoryID(PK), GUID: this field represent HistoryID.*
2. *OldValue: nVarchar (100): this field represent old value for specified field.*
3. *NewValue: nvarchar(100): this field repersent new value for specified field.*
4. *FieldName:varchar(50): this field represent the field name that has been modified.*
5. *ModificationDate:DateTime:this field represent the modification date.*
6. *ModifiedBy(FK):GUID: this field represent the persond who did the modification.*
7. *NegotiationID(FK): GUID: this field represent the id for the modified negotiation.*

## Negotiation\_Application Table

*This table is a conjunction table between application table and negotiation table. This table consists of the following fields;*

1. *AppID(PK,FK), GUID: this field represent application ID.*
2. *UserID(PK,FK),GUID: this field represent the current user.*
3. *NegotiationID(PK,FK),GUID: this field represent the current negotiation.*
4. *Active,Byte: this field represent the state of the current application for the corresponding*

*negotiation.*

## Channel Table

*This table represents All channels that may be used during application like chat channel, mail channel .... This table consists of the following fields;*

1. *ChannelID(PK), Int: this field represent ChannelID.*
2. *Channel Name: nVarchar (100): this field represent channel name.*
3. *ChannelTypeID(FK),int: this field represent the channel type.*

## Conversation History Table

*This table represents All modifications that occurred in conversation table during the application lifecycle. This table consists of the following fields;*

1. *HistoryID(PK),GUID: this field represent History ID.*
2. *OldValue: nVarchar (100): this field represent the old value for the modified field.*
3. *NewValue:nvarchar(100): this field represent the new value for the modified field.*
4. *FieldName:varchar(50): this field represent the name of the modified field.*
5. *ModifiedBy(FK):GUID: this field represent the user that did the modification.*
6. *ModificationDate:DateTime: this field represent the modification date.*
7. *ConversationID(FK):GUID: this field represent which conversation has been modified.*

## Message Table

*This table represents All messages that will be send and received during the application. This table consists of the following fields;*

1. *MessageID(PK), GUID: this field represent message ID.*
2. *ConversationID(FK),GUID: this field represent the corresponding conversation.*
3. *MessageContent,nText: this field represent the message body.*
4. *MessageSubject,nvarchar(200): this field represent the subject of the message.*
5. *MessageSender(FK),GUID: this field represent the sender user.*
6. *MessageReceiver(FK),GUID: this field represent the receiver user.*
7. *MessageDate,DateTime: this field represent the date of receiving message.*
8. *ChannelID(FK),Int: this field represent the corresponding channel.*
9. *Deleted,bit: this field act as a flage for deleting this record.*
10. *DeletedBy(FK),GUID: this field represent the user that deleted this message.*
11. *DeletedOn,DateTime: this field represent the date of deletion.*

## Attachment Table

*This table represents All attachement content in specific message. This table consists of the following fields;*

1. *AttachementID(PK), GUID: this field represent ID.*
2. *AttachementName: nVarchar (100): this field represent the name of the attached file.*
3. *AttachementContent:VarBinary(max): this field represent the content of the attachement.*
4. *MessageID(FK),GUID: this field represent the corresponding message.*

## UserType Table

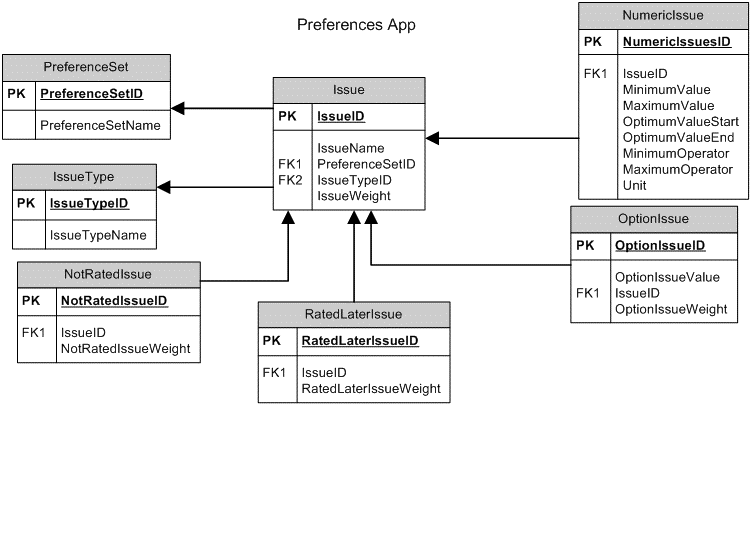
*This table represents All user types. This table consists of the following fields;*

1. *UserTypeID(PK), int: this field represent ID.*
2. *UserTypeDescription: nVarchar (100): this field represent the type description of the user.*

## Preference App

This is the schema for the Preference App, and it consists of some tables that will be described in more details in next sections:

**Table List**

1. PreferenceSet
2. Issue
3. IssueType
4. NumericIssue
5. OptionIssue
6. RatedLaterIssue
7. NotRatedIssue

## PreferenceSet Table.

*This table is used to represent a preference set . This table has the following fields;*

*1.        PreferenceSetID (PK), GUID: this field represents PreferenceSetID.*

*2.        PreferenceSetName, Varchar (300): this field represents user name.*

## Issue Table

*This table represents the issues of a preference set . This table consists of the following fields;*

1. *IssueID (PK), GUID: this field represent Issue ID.*
2. *IssueName,varchar(300):this field represents the name of an issue*
3. *PreferenceSetID(FK),GUID: this field refers to the prefernceSet that this issue belongs to.*
4. *IssueTypelD (FK), GUID: this field represent the type of issue.*
5. *IssueWeight,Decimal:this field represents the weight of an issue as a percentage*

## IssuType Table

*This table contains the possible type of an issue. It wil contain the following fields:*

1. *IssueTypelD(PK),GUID: the ID of the Issue type.*
2. *IssueTypeName,varchar(300):the name of the issue type.*

## NumericIssue

*This Table contains the values of issues that's type is numeric. The following fields are included in this table:*

1. *NumericlssueID(PK),GUID: the ID of a numeric issue.*
2. *IssuelD(FK),GUID: refers to the Issue that's type is numeric.*
3. *MinimumValue ,Decimal: the minimum limit value of a numeric issue .*
4. *MaximumValue,Decimal: the maximum limit value of a numeric issue .*
5. *OptimumValueStart,Decimal: the start limit of optimum value of a numeric issue the rates 100%*
6. *OptimumValueEnd,Decimal: the end limit of optimum value of a numeric issue the rates 100%*
7. *MinimumOperator,Byte: the operator that determines how rate curve is treated when optimum value equals the minimum value*
8. *MaximumOperator,Byte:the operator that determines how rate curve is treated when optimum value equals the maximum value*
9. *Unit,varchar(100):the unit that measures the numeric issue*

## OptionIssue Table

This table represents the values of issue of option type .*The following fields are included in this table:*

1. *Option IssueID(PK),GUID: the ID of issue that's type is Option.*
2. *OptionlssueValue,varchar(100): a value of an option issue .*
3. *IssuelD(FK),GUID: refers to the issue that's type is option.*
4. *OptionIssueWeight,Decimal: the weight that's used to rate this value.(in percentage)*

## RatedLaterIssue Table

This table represents the issues of type RatedLater.*The following fields are included in this table:*

1. *RatedLaterIssuelD(PK),GUID,: the ID of Rated Later Issue.*
2. *IssuelD(FK),GUID:refers to the Issue that's type is Rated Later.*
3. *RatedLaterIssueWeight,Decimal: the weight of this issue (in percentage)*

## NotRatedIssue Table

This table represents the issues of type NotRated.*The following fields are included in this table:*

1. *NotRatedIssuelD(PK),GUID,: the ID of Not Rated Issue.*
2. *IssuelD(FK),GUID:refers to the Issue that's type is Not Rated.*
3. *NotRatedIssueWeight,Decimal: the weight of this issue (in percentage)*

# System Components Integration/Interaction

The aim of this section is to show the whole interaction between all components in the system.

## Client / Web Application Communication

*The communication between client and web application is done via HTTP Request and the response is obtained via HTTP Reponses.*

## Data Access Tier / Database Communication

*The communication between Data Access Tier and Database is done using ADO.Net entity framework + RIA services.*

## Web Application/Data Access Tier Communication

*The communication between web application and Database is done using RIA services using REST protocol.*

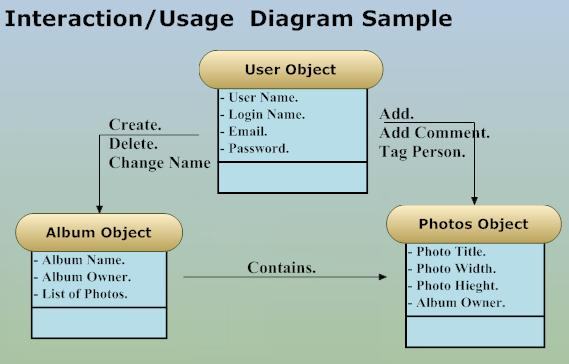
## Third Party Apps and Main Platform

*The communication between any third party apps and the main platform is being done using MEF [Managed Extension Framework] that supports plug in architecture for our platform.*

# System Objects Interaction/Usage

The aim of this section is to define all interactions/usage between objects and each others. The output of this section is a diagram shows the overview interaction between different objects in the system. This interaction helps a lot for creating Class Diagram.

*e.g. This section represents the interaction diagram between different objects in Facebook application. The following graph shows the interaction between different objet in the system.*



# System Sequence Diagram

This section aims to give the reader the sequence of manipulating application services. Each sequence diagram represents one request that system should provide. Sequence diagrams are listed in sections. Each section represents one sequence diagram for one service.

**Note**

Some times it will be preferable to use Activity Diagram instead of Sequence Diagram because of the existence of alternative paths. Any way, it is left to the writer to use Sequence or Activity Diagram.

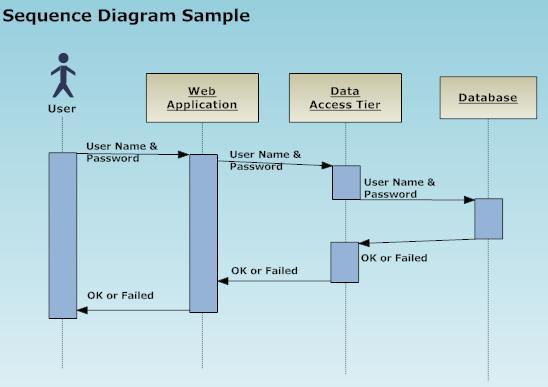
*e.g. This section shows the sequence diagrams for services that is provided via Facebook application. Each service will be listed in one section. The following sections show the sequence diagram for each service. Each section shows also the scenario*

## Login Service

*Next scenario and diagram shows login service sequence.*

1. *User enters his login name and password.*
2. *Web page sends his user name and hashed password via HTML response.*
3. *Web application receives user name and password and then sends them again to Data Access Tier via XML Request.*
4. *Data Access Tier sends call the function login that deals directly with database using ADO and returns result to Data Access Tier.*
5. *Data Access Tier returns result to Web Application via XML Response.*
6. *Web Application shows result to user (login or failed) via HTTP Response.*

*Next figure shows Sequence Diagram for Login Service.*



# Time Plan

The aim of this section is to create time plane system breakdowns. This plan will be created using Microsoft Project by Team Leader and reviewed internally by management then by customer. If any changes happened, the plan should be updated. The team should follow the plan.

# Glossary

***[A]***

***Add-ons:***

An Add-on is a client-side application that is installed on a Desktop PC or a Mobile Device and can be either a separate application (e.g. eNeg Desktop Add-on, eNeg Mobile Add-on) or integrated in existing applications (e.g. eNeg Outlook Add-on, eNeg Skype Add-on etc.).

***APP:***

An App is a kind of module or set of features that Users can activate or deactivate via eNeg Desktop Add-on or eNeg Web Platform. Apps can be provided by eNeg or by Third Party Software Developers.

***[C]***

***Conversation****:*

A Conversation is always linked to one Negotiation. A Conversation is a collection of messages between the User and 1 to n participants of a Conversation.

***[D]***

***Data Access Tier:*** *Middle tier used to make communication between web application and database….etc*

***[E]***

***eNeg****:*

*Electronic Negotiation, and in the same time is name for the current solution.*

***[M]***

***Message:***

A Message is like an e-Mail and thus contains Sender, Receiver(s), Message Subject, Message Content etc.

***[N]***

***Negotiation:***

A Negotiation consists of 1 to n Conversations and thus categorizes them. It is used to define a topic the User wants to negotiate about e.g. “Car Purchase Negotiation”, “Co-operation Contract Negotiation” etc. The differentiation between Negotiation and Conversation is needed to better execute 1 to n negotiations, e.g. during a “Car Purchase Negotiation” the User negotiates about the same topic with different suppliers.

***[P]***

***Preference App:***

A third party application that can be activated or deactivated upon customer request to do specific operations on the current negotiation.

***[S]***

***Support:***

Support is individual to User needs composed information provided by Apps to support the User during his Negotiation. Examples of Support are e.g. Analysis and rating of offers (e.g. this offer is 87%), etc.

References

## Service Architecture Pocket Guide.pdf

## RIA Architecture Pocket Guide.pdf

## App Arch Guide 2.0 .pdf

## [*http://www.devtoolshed.com/tutorial-configuring-silverlight-4-entity-framework-and-wcf-ria-services-separate-component-assembli*](http://www.devtoolshed.com/tutorial-configuring-silverlight-4-entity-framework-and-wcf-ria-services-separate-component-assembli)

## *[http://whitepapers.techrepublic.com.com/thankyou.aspx?&docid=1028633&view=1028633&load=1](http://www.devtoolshed.com/tutorial-configuring-silverlight-4-entity-framework-and-wcf-ria-services-separate-component-assembli)*

## *[http://www.davidezordan.net/blog/?p=1902](http://www.devtoolshed.com/tutorial-configuring-silverlight-4-entity-framework-and-wcf-ria-services-separate-component-assembli)*

## [*http://mef.codeplex.com/*](http://mef.codeplex.com/)

## [*http://blogs.msdn.com/b/gblock/archive/2008/10/02/mef-going-ms-pl-the-little-engine-that-could.aspx*](http://blogs.msdn.com/b/gblock/archive/2008/10/02/mef-going-ms-pl-the-little-engine-that-could.aspx)

## [*http://weblogs.asp.net/dwahlin/archive/2009/12/08/getting-started-with-the-mvvm-pattern-in-silverlight-applications.aspx*](http://weblogs.asp.net/dwahlin/archive/2009/12/08/getting-started-with-the-mvvm-pattern-in-silverlight-applications.aspx)

## [*http://blogs.msdn.com/b/erwinvandervalk/archive/2009/08/18/implementing-the-model-view-viewmodel-pattern.aspx*](http://blogs.msdn.com/b/erwinvandervalk/archive/2009/08/18/implementing-the-model-view-viewmodel-pattern.aspx)

## [*http://www.astaticstate.com/2010/04/silverlight-4-using-mvvm-patter-ria.html*](http://www.astaticstate.com/2010/04/silverlight-4-using-mvvm-patter-ria.html)

## [*http://blogs.msdn.com/b/erwinvandervalk/archive/2009/08/14/the-difference-between-model-view-viewmodel-and-other-separated-presentation-patterns.aspx*](http://blogs.msdn.com/b/erwinvandervalk/archive/2009/08/14/the-difference-between-model-view-viewmodel-and-other-separated-presentation-patterns.aspx)

## [*http://johnpapa.net/silverlight/5-minute-overview-of-mvvm-in-silverlight/*](http://johnpapa.net/silverlight/5-minute-overview-of-mvvm-in-silverlight/)

## [*http://en.wikipedia.org/wiki/Model\_View\_ViewModel#cite\_note-2*](http://en.wikipedia.org/wiki/Model_View_ViewModel%23cite_note-2)

## [*http://www.silverlight.net/learn/videos/silverlight-videos/build-data-driven-silverlight-application-using-the-mvvm-pattern/*](http://www.silverlight.net/learn/videos/silverlight-videos/build-data-driven-silverlight-application-using-the-mvvm-pattern/)

## [*http://www.codeproject.com/KB/silverlight/IssueVisionForSilverlight.aspx*](http://www.codeproject.com/KB/silverlight/IssueVisionForSilverlight.aspx)

## [*http://live.visitmix.com/MIX10/Sessions/EX15*](http://live.visitmix.com/MIX10/Sessions/EX15)

## [*http://community.infragistics.com/pixel8/media/p/91949.aspx*](http://community.infragistics.com/pixel8/media/p/91949.aspx)

## [*http://msdn.microsoft.com/en-us/library/cc907912.aspx*](http://msdn.microsoft.com/en-us/library/cc907912.aspx)

## [*http://wildermuth.com/2009/09/29/Choosing\_a\_Data\_Access\_Layer\_for\_Silverlight\_3*](http://wildermuth.com/2009/09/29/Choosing_a_Data_Access_Layer_for_Silverlight_3)

## [*http://msdn.microsoft.com/en-us/library/cc907912.aspx*](http://msdn.microsoft.com/en-us/library/cc907912.aspx)