Daily Notebook & Social Networking Updater

**MCA 6th Semester**

**MCSP-060(Project Report)**

**Anirban Nandy**

**Enrolment Number: 105057536**

# Index

Contents

[**MCA 6th Semester** 1](#_Toc351840184)

[**MCSP-060(Project Report)** 1](#_Toc351840185)

[**Anirban Nandy** 1](#_Toc351840186)

[**Enrollment Number : 105057536** 1](#_Toc351840187)

[Index 2](#_Toc351840188)

[Introduction: 5](#_Toc351840189)

[**Figure:** Overview of Daily notebook & Social Networking Updater 6](#_Toc351840190)

[Objective: 7](#_Toc351840191)

[System Analysis 7](#_Toc351840192)

[Identification of Need: 7](#_Toc351840193)

[Preliminary Investigation: 7](#_Toc351840194)

[Feasibility Study: 8](#_Toc351840195)

[Project Planning & Scheduling: 8](#_Toc351840196)

[Gantt chart 8](#_Toc351840197)

[Tracking Gantt 8](#_Toc351840198)

[Pert Chart 9](#_Toc351840199)

[Software requirement specifications (SRS): 10](#_Toc351840200)

[Functional Requirement 11](#_Toc351840201)

[Non-functional Requirements 13](#_Toc351840202)

[Software Engineering Paradigm applied 14](#_Toc351840203)

[Data models 14](#_Toc351840204)

[Context Diagram 14](#_Toc351840205)

[0-Level DFD 15](#_Toc351840206)

[1-Level DFD 16](#_Toc351840207)

[2-Level DFD 17](#_Toc351840208)

[Sequence diagrams 17](#_Toc351840209)

[Entity Relationship Model, 20](#_Toc351840210)

[Class Diagrams 21](#_Toc351840211)

[Activity Diagrams 22](#_Toc351840212)

[System Design 25](#_Toc351840213)

[Modularisation details 25](#_Toc351840214)

[**DNBSN** Engine: 25](#_Toc351840215)

[**DNBSN** GUI: 25](#_Toc351840216)

[**DNBSN** Storage: 25](#_Toc351840217)

[Google Calendar: 25](#_Toc351840218)

[Facebook/Twitter/LinkedIn API: 26](#_Toc351840219)

[Data integrity and constraints 26](#_Toc351840220)

[Entity integrity 26](#_Toc351840221)

[Referential Integrity 26](#_Toc351840222)

[Domain Integrity 26](#_Toc351840223)

[User Defined Integrity 26](#_Toc351840224)

[Database design 26](#_Toc351840225)

[User Interface Design 28](#_Toc351840226)

[Test Cases (Unit Test Cases and System Test Cases) 32](#_Toc351840227)

[Unit Test Cases 32](#_Toc351840228)

[System Test Cases 42](#_Toc351840229)

[Coding 51](#_Toc351840230)

[Complete Project Coding 51](#_Toc351840231)

[Comments and Description of Coding segments 51](#_Toc351840232)

[Comments and API Documentation 51](#_Toc351840233)

[Description of coding 51](#_Toc351840234)

[Standardization of the coding 52](#_Toc351840235)

[Code Efficiency 52](#_Toc351840236)

[Error handling 52](#_Toc351840237)

[Parameters calling/passing 52](#_Toc351840238)

[Validation checks 52](#_Toc351840239)

[Testing 52](#_Toc351840240)

[Testing techniques and testing strategies used 52](#_Toc351840241)

[Database & Data Integrity Testing 52](#_Toc351840242)

[Functional Testing: 53](#_Toc351840243)

[Regression Testing: 53](#_Toc351840244)

[User Interface Testing: 54](#_Toc351840245)

[Performance Profiling: 54](#_Toc351840246)

[Load Testing: 55](#_Toc351840247)

[Stress Testing: 55](#_Toc351840248)

[Volume Testing: 56](#_Toc351840249)

[Security & Access Control Testing: 56](#_Toc351840250)

[Failover & Recovery Testing: 56](#_Toc351840251)

[Configuration Testing: 56](#_Toc351840252)

[Installation/Deploy & Back out Testing: 57](#_Toc351840253)

[Post Production Testing: 57](#_Toc351840254)

[Unit Testing: 57](#_Toc351840255)

[Smoke Testing: 57](#_Toc351840256)

[Data Migration Testing: 57](#_Toc351840257)

[Testing Plan used 58](#_Toc351840258)

[Test reports for Unit Test Cases and System Test Cases 59](#_Toc351840259)

[Debugging and Code improvement: 65](#_Toc351840260)

[System Security measures: 65](#_Toc351840261)

[Database/data security: 65](#_Toc351840262)

[Creation of User profiles and access rights 65](#_Toc351840263)

[Cost Estimation of the Project along with Cost Estimation Model 65](#_Toc351840264)

[Estimation of development effort 66](#_Toc351840265)

[Estimation of development time 66](#_Toc351840266)

[Reports (sample layouts should be placed) 67](#_Toc351840267)

[Future scope and further enhancement of the Project 67](#_Toc351840268)

[Bibliography 68](#_Toc351840269)

[Website 68](#_Toc351840270)

[Books 69](#_Toc351840271)

[Appendices 69](#_Toc351840272)

[IDE Used: 69](#_Toc351840273)

[Visual Studio 2010 69](#_Toc351840274)

[Front End - WPF (Windows Presentation Framework) 70](#_Toc351840275)

[Extensible applicationMarkup Language (XaML) 70](#_Toc351840276)

[Programming Framework (.NET 4) 71](#_Toc351840277)

[Database/backend - MySQL 71](#_Toc351840278)

[ide for Database –MySQL workbench 71](#_Toc351840279)

[Programming Language (C#) 72](#_Toc351840280)

[Dia for Diagram Drawing &Modeling 72](#_Toc351840281)

[Google Spreadsheet Interface: 73](#_Toc351840282)

[Windows Mobile 6 Professional SDK 73](#_Toc351840283)

[Cacoo:: online drawing tool 74](#_Toc351840284)

[Creating Diagrams 74](#_Toc351840285)

[Collaboration 74](#_Toc351840286)

[Sharing Diagrams 75](#_Toc351840287)

[Managing Diagrams 75](#_Toc351840288)

[Languages and Time Zones 75](#_Toc351840289)

[Security 75](#_Toc351840290)

[API 75](#_Toc351840291)

[Version Control System : GitHub 76](#_Toc351840292)

[Description 76](#_Toc351840293)

[Limitations and constraints 76](#_Toc351840294)

[Glossary. 76](#_Toc351840295)

# Introduction:

Nowadays social networking is the main trend among us. We like to share whatever is happening with us or around us and follow about others. We want to keep track of every possible thing in our life. Earlier day’s people used to maintain Notebook or Daily Diary to write down about their experiences. Now also we do same things but in different ways. With advent of new technologies we use twitter, facebook instead of maintaining a diary. As all of them are online services we cannot access them without an internet connection and sometimes it takes lots of time to load or query our own posts from the server.

Daily notebook & Social Networking Updater will provide a way to organize our daily notes and status updates for social networking sites. It will allow users to add notes and update it to most popular social networking sites. User can add notes anytime and mark it for update, and then whenever user comes online the pending updates will be uploaded to the destination sites. Users will be able to download the RSS feed of the websites and the Daily notebook & Social Networking Updater will automatically download the feeds and make it available for offline view also. The overview of this software is displayed below in the diagram.



**Figure:** Overview of Daily notebook & Social Networking Updater

Main features of the Daily notebook & Social Networking Updater are:

* Maintaining daily notes
* Updating the status in Social Networking Sites and downloading the RSS feeds

|  |
| --- |
| * Facebook integration * Twitter integration * Google Calendar integration * Linkedin integration * Google plus integration * Google Maps integration |

* Maintaining To Do/Task list
* Reminder/ alarm for scheduled task
* Printing of Notes, To Do & Tasks
* Archiving old data
* Searching Updates

# Objective:

* Single solution for updating any social networking sites
* Keeping track of daily activities
* Timely update whenever internet is available
* To simulate similar experience of Daily notebook.
* To be involved with full-fledged software development
* To know about new technologies like .NET, WPF, Facebook, Twitter.

# System Analysis

## Identification of Need:

I used to write diary regularly ten years ago when I was in school. Today, we use social networking sites as our notebook or diary. Though we can share all our thoughts and memories to the social networking sites, yet there are some drawbacks as well. The main problem is the availability of internet; we cannot share our thoughts while we are at such a place. So we might forget our thoughts. I faced this problem many times. Also, we might not want share all our feelings with others, we could store some of them for ourselves. In that case social networking sites are not as good as a diary or notebook. Sharing same data to different social networking site is a nuisance as we need to do same task repeatedly. I have been facing these problems for many years so I felt that we need a solution that could minimizethe drawbacks of social networking sites.

## Preliminary Investigation:

I spoke with many of my friends who use social networking sites regularly and most of them face similar kind of problem. I thought a desktop cum web application could be developed to minimize theses shortcomings of social networking sites. I then started gathering opinion of my friends and seniors among whom some are IT professionals. I gathered all the important points including my own opinion and decided to develop Daily Notebook.

## Feasibility Study:

It is an admitted fact that people are becoming more and more addicted to social networking sites day by day. People would love an application that would make their social networking experience more interesting and flawless. I have decided to provide a password manager that to keep track of the id and passwords created in various websites on the internet. So, undoubtedly it is going to be a popular web cum desktop application.

## Project Planning & Scheduling:

### Gantt chart



### Tracking Gantt



### Pert Chart



## Software requirement specifications (SRS):

### Functional Requirement

#### Register User

**Introduction:**

Register a new User.

**Input:**

Relevant User data like name, user id, password,hints etc.

**Processing:**

The **DNBSN**willcreate a new user entry.

**Output:**

The **DNBSN**will generate a user to use the application.

#### Login User

**Introduction:**

Logging in as an existingUser.

**Input:**

Userwill provideuser id, password.

**Processing:**

The **DNBSN**willcheck the authorization of the particular user.

**Output:**

The **DNBSN**will allow accessing feature to the user if the given data match with the internal information, otherwise denying user.

#### Update Note or Event

**Introduction:**

User can add note or event.

**Input:**

User will compose note or eventfor update to note book or social site. And select the option where he wishes to update the event.

**Processing:**

The **DNBSN**willcreate a new event for note book and share the event to the selected social site.

**Output:**

The **DNBSN**will save event in note book and share it in social site.

#### Load Google Map

**Introduction:**

User can load Google map through **DNBSN**.

**Input:**

Current location.

**Processing:**

The **DNBSN**willget location from google map.

**Output:**

The **DNBSN**will display the particular location.

#### Share event in Google calendar

**Introduction:**

User can share event in google calendar through **DNBSN**.

**Input:**

The user selects the event from note book to upload in google calendar.

**Processing:**

The **DNBSN**will upload event in calendar.

**Output:**

The google calendar will display the particular event.

#### Search Event

**Introduction:**

User can searchevent.

**Input:**

He will enter data like key word or date etc.

**Processing:**

The **DNBSN**will search for the requirement.

**Output:**

The **DNBSN**will display the search result.

#### Add Contact

**Introduction:**

The **DNBSN**will Store a new Contact.

**Input:**

Relevant contact data like name, e-mail id, mobile no, fax no. address, blood group etc.

**Processing:**

Admin will enter the data in the **DNBSN**and create a new Donor entry.

**Output:**

The **DNBSN**will generate a contact detail for future reference.

#### Add To Do

**Introduction:**

The **DNBSN**will Store a new task in To Do.

**Input:**

Relevant task data like task detail, priority etc.

**Processing:**

Admin will enter the data in the **DNBSN**and create a new task entry.

**Output:**

The **DNBSN**will generate a task detail.

### Non-functional Requirements

* The application will be **self-dependent** and no dependency on other parties required.
* There will be a digital **backup** and restore system.
* There will be more **opportunity** to extend the application in various type of device in future.
* The response time will be low and the system will **response** fast.
* It will be very **user friendly** and **usable** by any person with minimal computer knowledge.
* In terms of **security** unauthorized access will be denied and register user will be able to change as necessary.
* It will be **efficient** as it reduces manual labor and searching.
* **DNBSN** will have user manual and help **documents**.
* It is designed such a way that it can be **maintained** with minimal effort.

## Software Engineering Paradigm applied

## Data models

### Context Diagram



### 0-Level DFD



### 1-Level DFD



### 2-Level DFD



## Sequence diagrams

**Interaction Event**

**: Login**

**User**

**: Register**

**User**

**Controller**

**: Update**

**Event**

**: Events**

**Report**

**: View**

**Update Event**

**Report**

**register**

**register**

**modifyEvent**

**addEvent**

**deleteEvent**

**showError**

**showError**

**ShowError**

**updateEvent**

**updateEventReport**

**updateEventsReport**

**showError**

**showError**

**showError**

**viewReport**

**viewEventReport**

**displayEventReport**

**displayReport**

**Share Event in Social Site**

**register**

**register**

**updateEvent**

**syncNotComplete**

**syncComplete**

**syncEvent**

**syncEvent**

**displayEvent**

**syncComplete**

**: Login**

**User**

**: Register**

**User**

**Controller**

**: Update**

**Desktop**

**Event**

**: Update**

**Web**

**Application**

**:Update**

**Desktop**

**Application**

## Entity Relationship Model,

We will design a RDBMS for Daily notebook & Social Networking Updater. The entities and their attributes are listed below. Attributes in Bold letter is the unique key.

|  |  |
| --- | --- |
| **Entities** | **Attributes** |
| Note | **Note Id**, Content, Time, Size, Web Service id, user |
| RSS Feed | **RSS Feed Id** , Web Service Id, Content, time, size |
| Daily notebook & Social Networking Updater | **Sw Id**, Web Services Supported, Users, Size |
| User | **User Id**, Name, Social Network Data, Preferences. |
| Web Service | **Web Service Id,** Authentication Data, Feed Data, Preferences. |

**Relationship between Entities:**

* Daily notebook & Social Networking Updater has User 🡪 1 : N
* Users post Notes 🡪 1 : N
* Web Service generates Feeds🡪 M : N



E-R Diagram of Daily notebook & Social Networking Updater

## Class Diagrams



## Activity Diagrams



# System Design

## Modularisation details

### DNBSN Engine:

This module handles all the logical parts of **DNBSN**. It takes data from user through **DNBSN** GUI module and stores them to database using **DNBSN** Storage module. It sends the data to the user’s social networking account using corresponding site’s API. The data stored is taken from the **DNBSN** Storage and sent to the API. It saves the events details in the Google calendar.

### DNBSN GUI:

This part is the place through which user interacts. This module contains all the designs which are visible and intractable by users. User provides input through it and gets the output through it. It is generally created by various tools like buttons and listviews.

### DNBSN Storage:

In this module all the data are stored. **DNBSN** Engine stores data in this module and fetches them for output through this module.

### Google Calendar:

This place gets input from the **DNBSN** engine. All the relevant data sent by user to Google calendar is stored here which could be accessed by user globally.

### Facebook/Twitter/LinkedIn API:

These modules provide news feed to user using the **DNBSN** engine through **DNBSN** GUI. It further gets the input, i.e. status update to the user’s account.

## Data integrity and constraints

We have used Integrity constraints in **DNBSN** to ensure accuracy and consistency of data in a relational database. Data integrity is handled in a relational database through the concept of referential integrity. There are many types of integrity constraints in **DNBSN** that play a role in referential integrity.

Codd initially defined two sets of constraints but, in his second version of the relational model, he came up with four integrity constraints:

### Entity integrity

In **DNBSN** we used various type of primary key and consciously we set the primary key property as not null. The entity integrity constraint states that no primary key value can be null. This is because the primary key value is used to identify individual tuples in a relation. Having null value for the primary key implies that we cannot identify some tuples.This also specifies that there may not be any duplicate entries in primary key column key row.

### Referential Integrity

The referential integrity constraint is specified between two relations and is used to maintain the consistency among tuples in the two relations. Informally, the referential integrity constraint states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation. It is a rule that maintains consistency among the rows of the two relations.

### Domain Integrity

**DNBSN** has various type of data field with set by default value of Null because if the value is not provided by the user, the vale will be set as null. The domain integrity states that every element from a relation should respect the type and restrictions of its corresponding attribute. A type can have a variable length which needs to be respected. Restrictions could be the range of values that the element can have, the default value if none is provided, and if the element can be NULL.

### User Defined Integrity

A business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behaviour of the business.

## Database design

The database used for this software is called **Dnbdb**. Database tables and corresponding keys are shown in tabular form. It shows the tables and its columns. A key in **Bold** is the primary key.

Screenshots of table structures:

Table: user



Table: contact



Table: note



Table: password



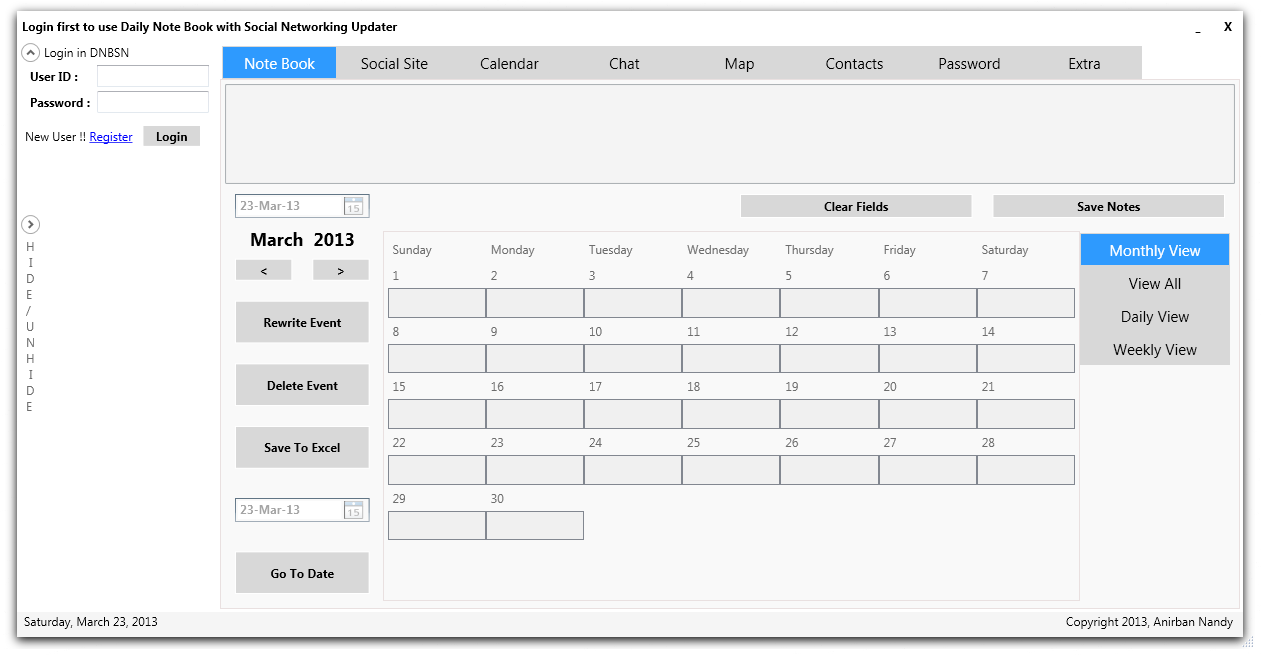
Table: tasks



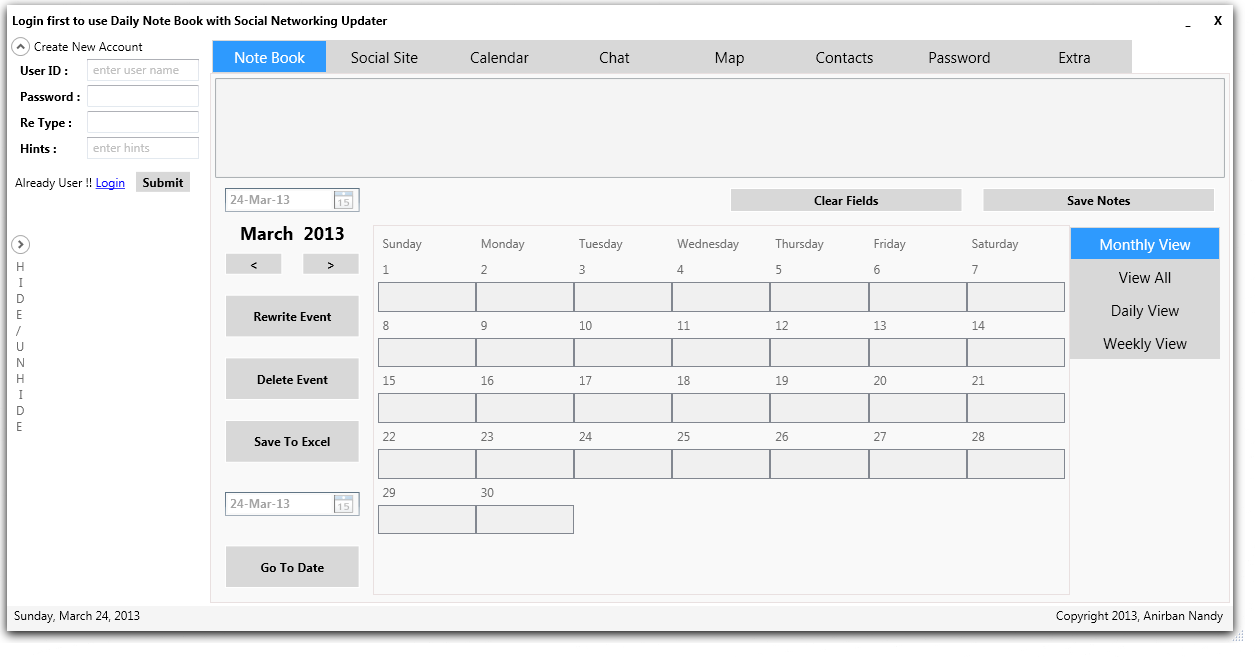
|  |  |
| --- | --- |
| **Tables** | **Keys** |
| Note | **Note Id**, Content, Time, Size, Web Service id, user |
| RSS Feed | **RSS Feed Id** , Web Service Id, Content, time, size |
| Daily notebook & Social Networking Updater | **Sw Id**, Web Services Supported, Users, Size |
| User | **User Id**, Name, Social Network Data, Preferences. |
| Web Service | **Web Service Id,** Authentication Data, Feed Data, Preferences. |

## User Interface Design

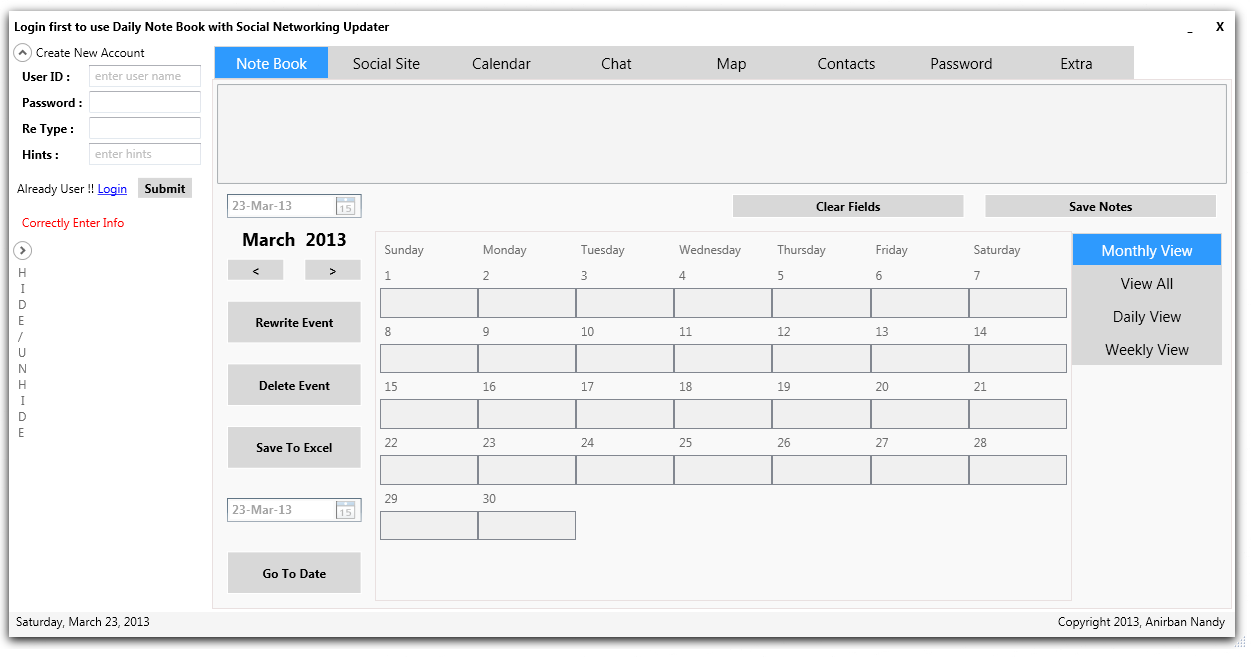
### Main Window without Login



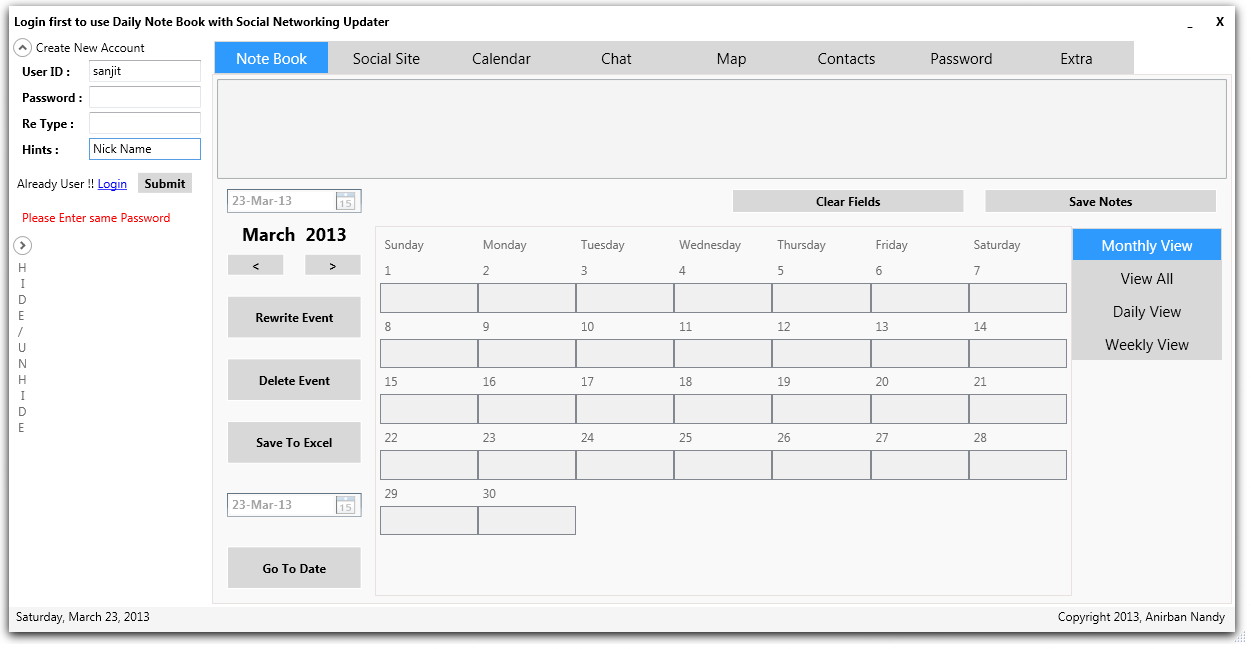
### Create New User Account



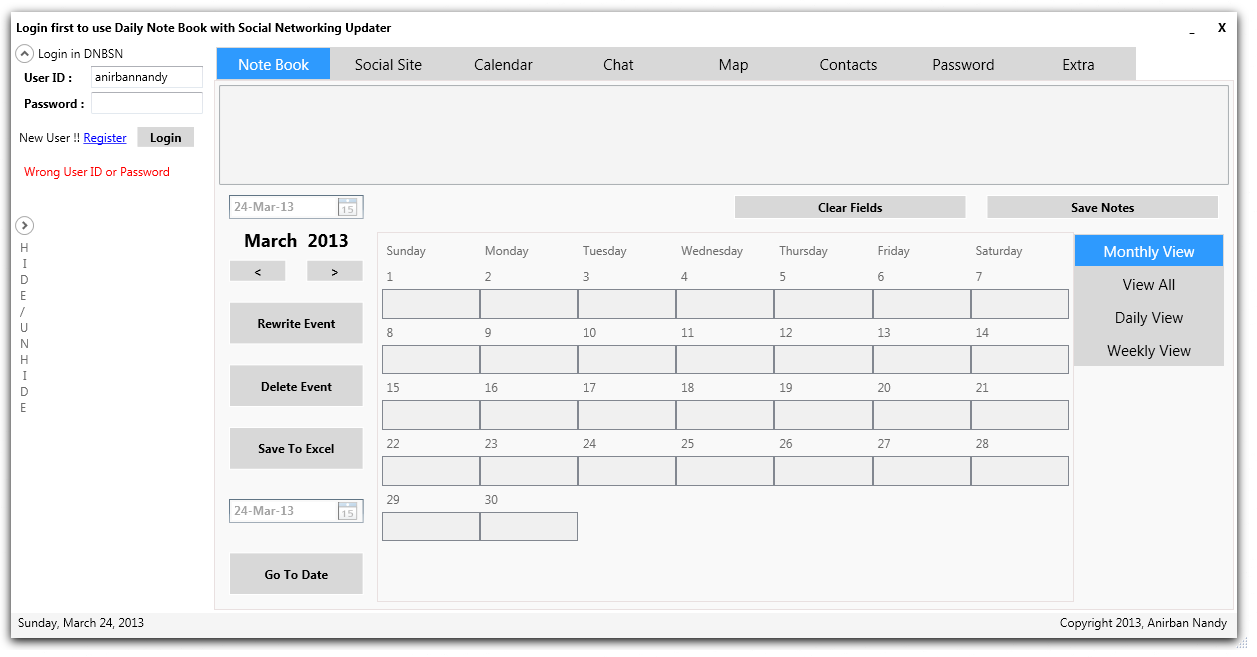
### Fields not filled properly: error message



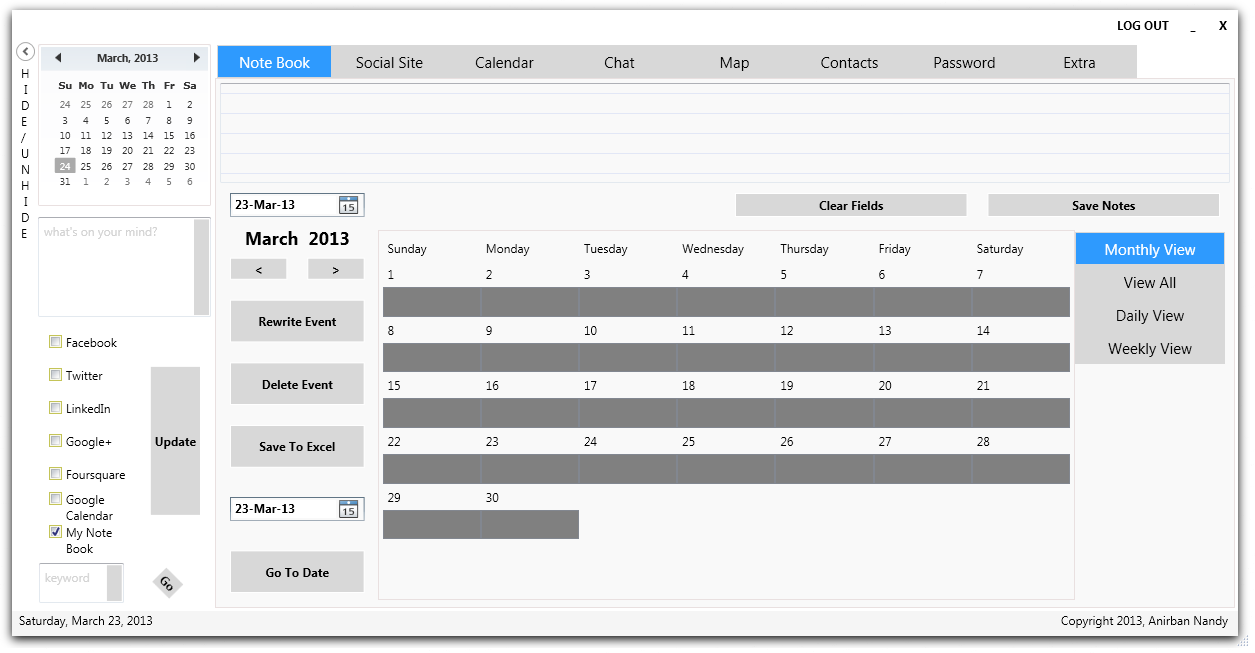
### Miss match Password



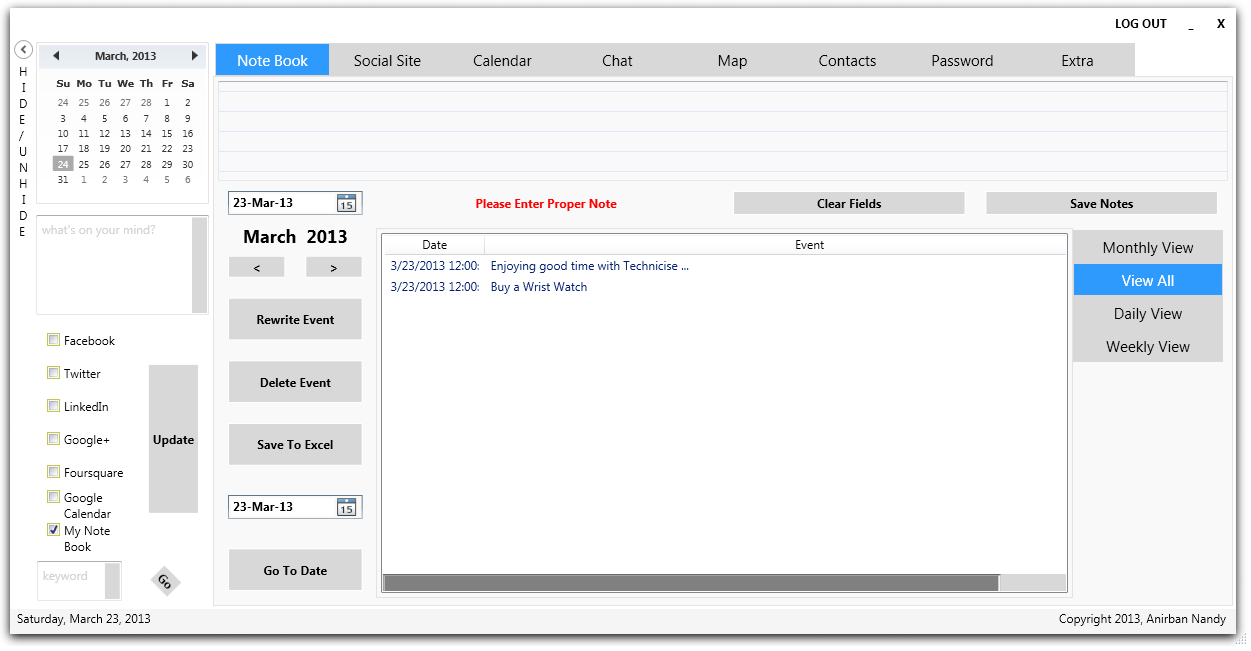
### Incorrect user Id or Password



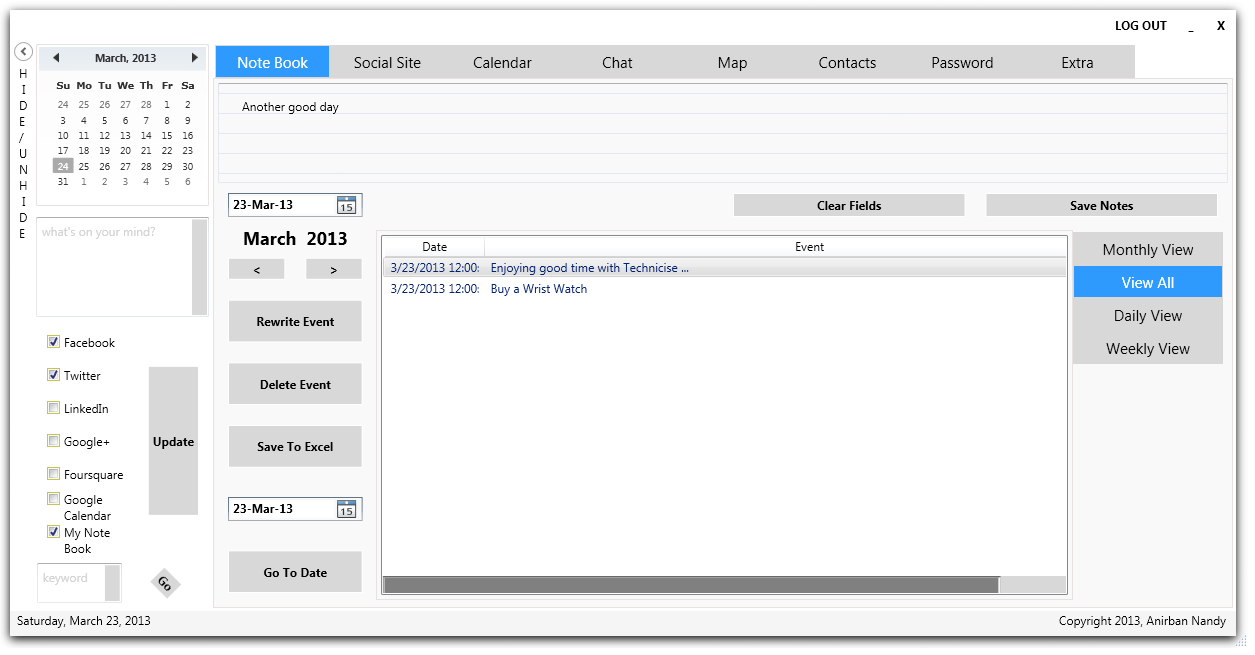
### After successfully login



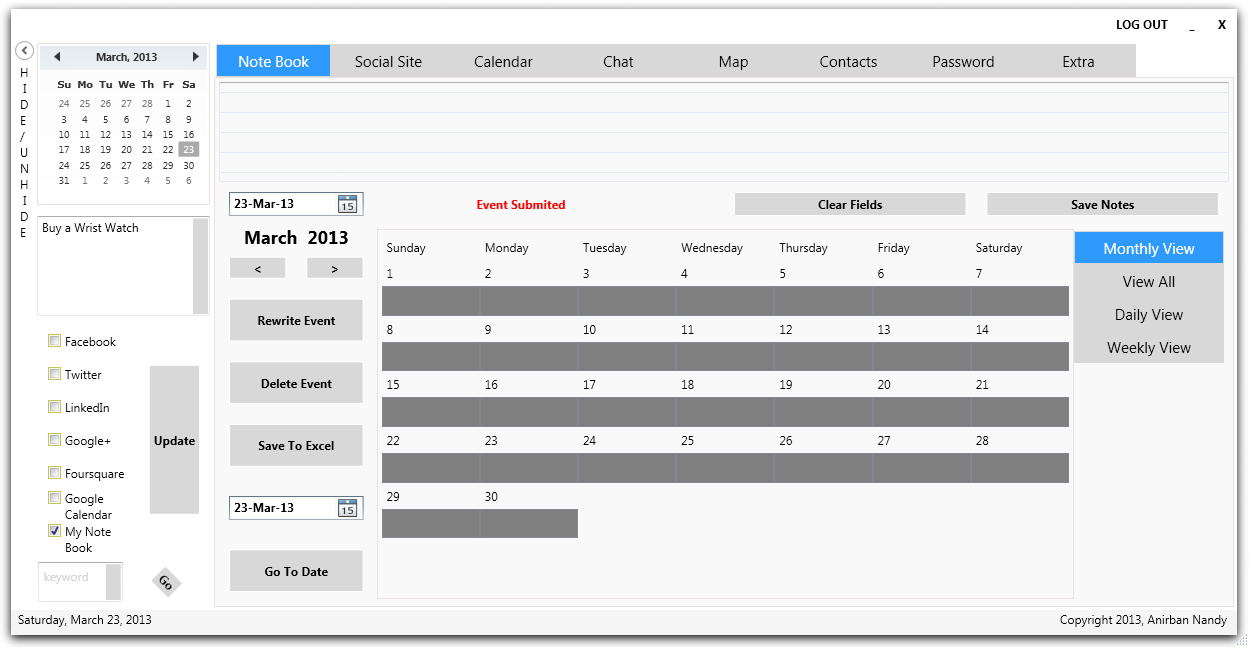
### When the Event Field is Empty



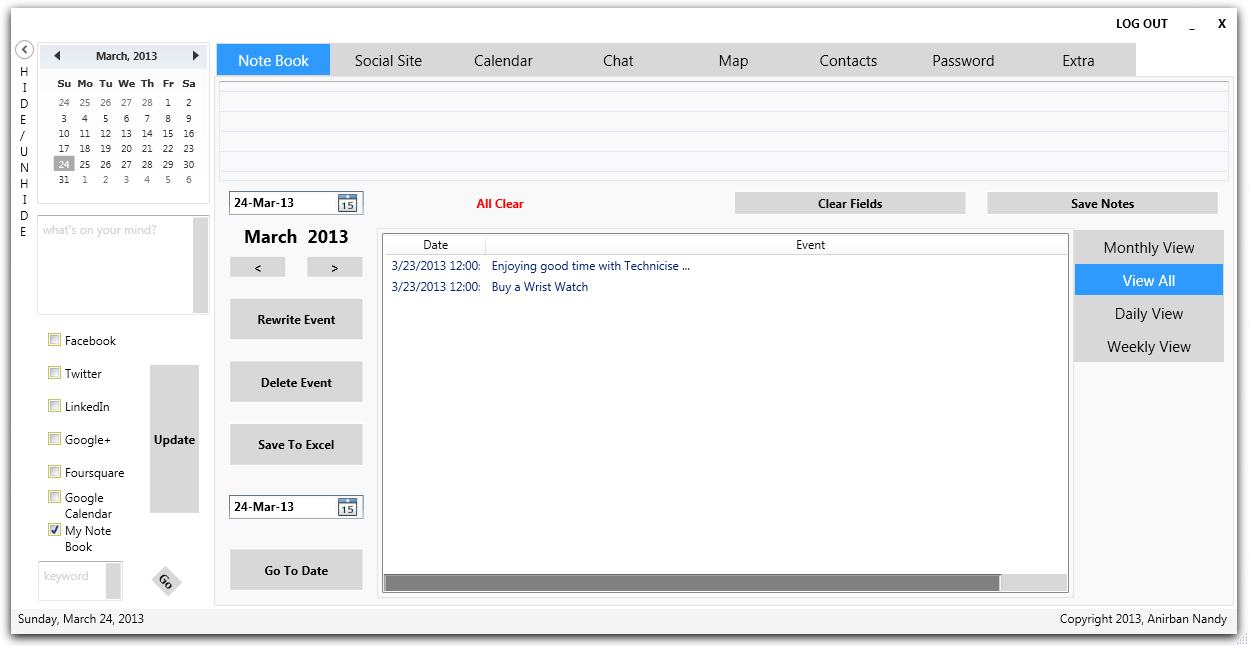
### Write text in text field



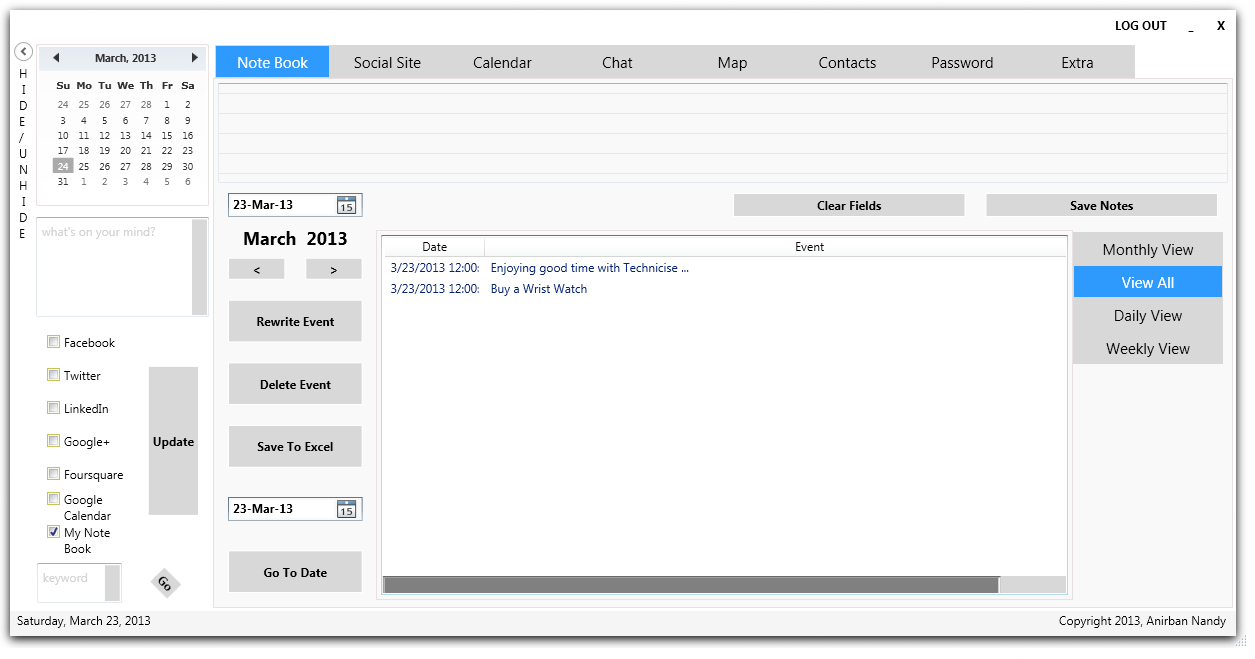
### After Event Submitted



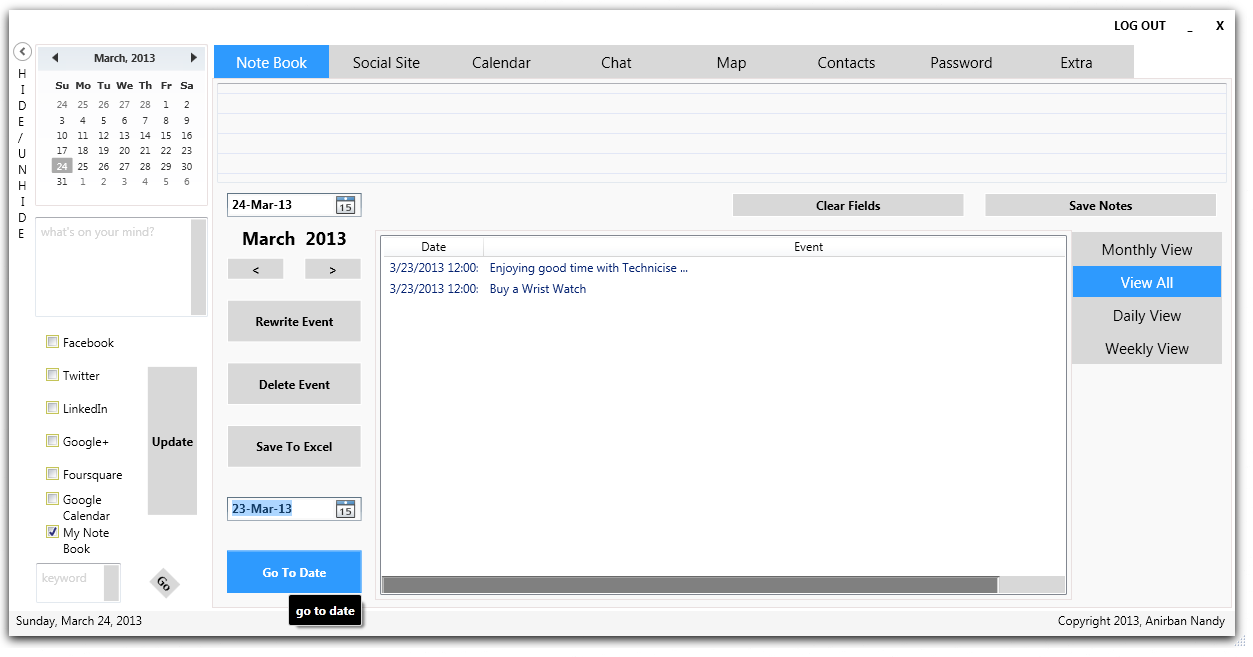
### After Event clear



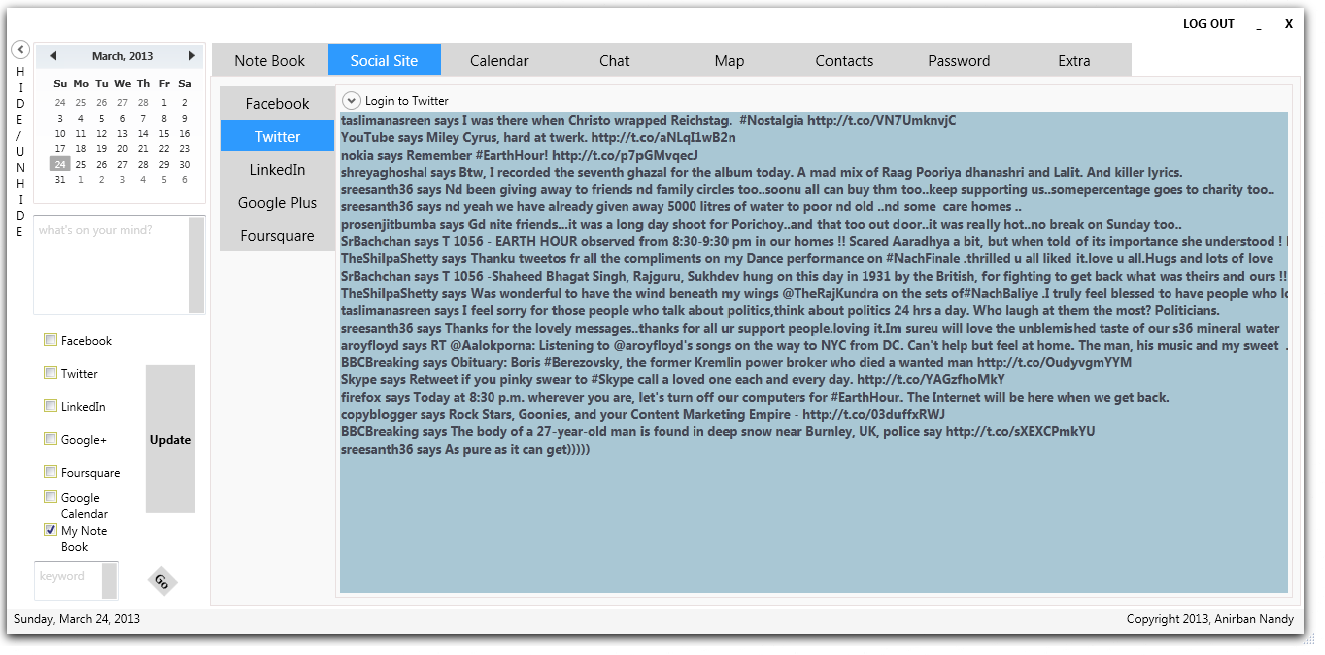
### View All Event



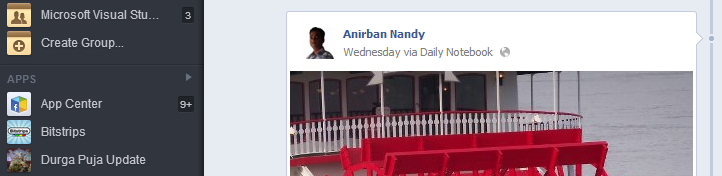
### Search By date



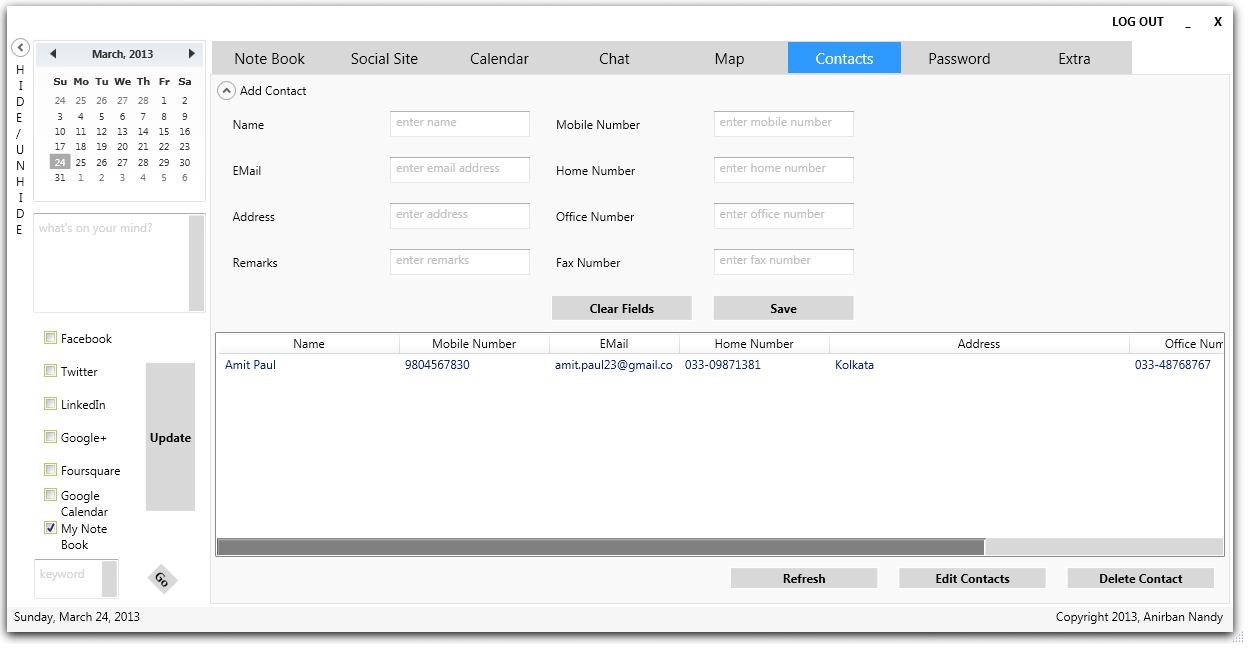
### Tweet from Twitter



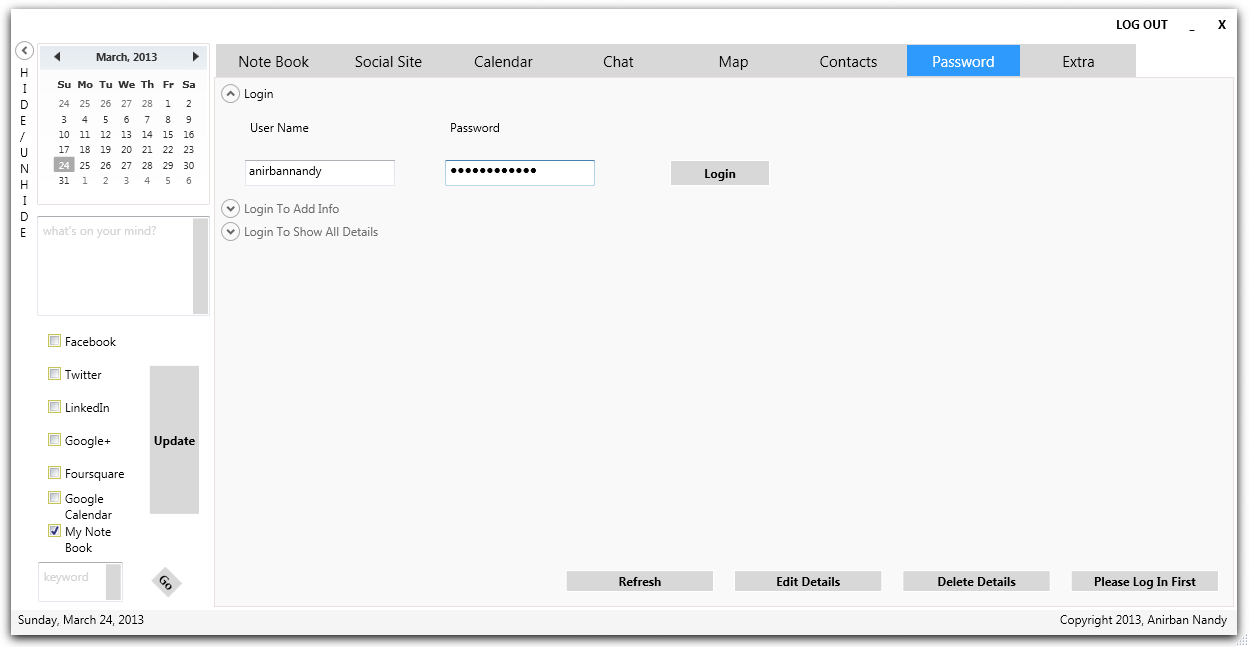
### After Facebook Sharing



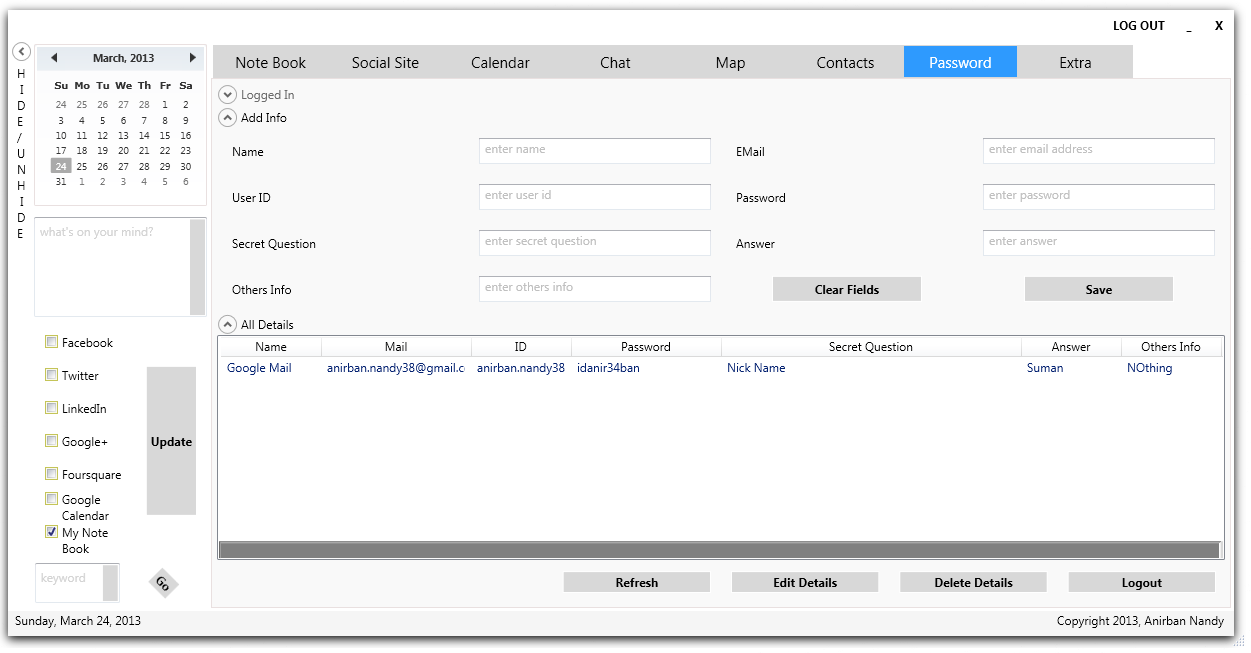
### All contacts



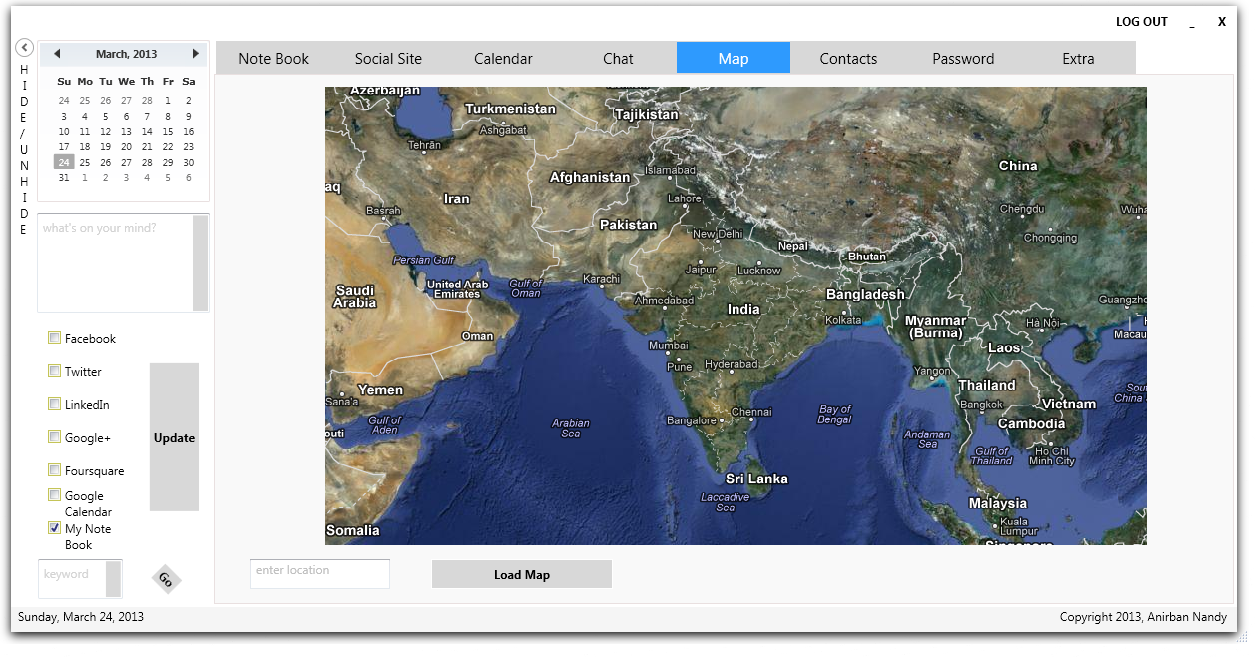
### Password Login



### Password Info



### Load location in map



## Test Cases (Unit Test Cases and System Test Cases)

### Unit Test Cases

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case Id | Type | Github ID | Subject | Test Name | Test Description | Step Name | Description | Expected Result |
| **DNBSN**-001 | Manual | f3563be0a9c431104f52839039e86043cf640cf1 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check Successful Login for **DNBSN** | The purpose of this test is to verify that the User Id and Password of user is valid. | Step1 | Insert wrong User Id and Password. And Click on Login Button. | **DNBSN** will display error message. And Failed to Login. |
| **DNBSN**-002 |  |  |  |  |  | Step2 | Insert Wrong User Id and valid Password. And Click on Login Button. | **DNBSN** will display error message. And Failed to Login. |
| **DNBSN**-003 |  |  |  |  |  | Step3 | Insert Valid User Id and Wrong Password. And Click on Login Button. | **DNBSN** will display error message. And Failed to Login. |
| **DNBSN**-004 |  |  |  |  |  | Step4 | Insert Nothing in User Id and Password fields. And Click on Login Button. | **DNBSN** will display error message. And Failed to Login. |
| **DNBSN**-005 |  |  |  |  |  | Step5 | Insert Nothing in User Id and insert Valid Password fields. And Click on Login Button. | **DNBSN** will display error message. And Failed to Login. |
| **DNBSN**-006 |  |  |  |  |  | Step6 | Insert Nothing in Password and insert Valid User Id fields. And Click on Login Button. | **DNBSN** will display error message. And Failed to Login. |
| **DNBSN**-007 |  |  |  |  |  | Step7 | Insert Nothing in User Id and insert invalid Password fields. And Click on Login Button. | **DNBSN** will display error message. And Failed to Login. |
| **DNBSN**-008 |  |  |  |  |  | Step8 | Insert Nothing in Password and insert invalid User Id fields. And Click on Login Button. | **DNBSN** will display error message. And Failed to Login. |
| **DNBSN**-009 |  |  |  |  |  | Step9 | Insert valid User Id and Password. And Click on Login Button. | Successfully login. |
| **DNBSN**-010 | Manual | d01197ee4cd3bee9245874b5937ba740019fd131 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check Successful Registration for New **DNBSN** User. | The purpose of this test is to verify that the all new connection could be creating new Account By Registration. | Step1 | Click on Registration link. | New Account creation area is opened. |
| **DNBSN**-011 |  |  |  |  |  | Step2 | Enter existing new User Id, Password, Retype Same password and Hints. And Click on Login Button. | **DNBSN** will display error message. And Failed to Registration. |
| **DNBSN**-012 |  |  |  |  |  | Step3 | Enter existing new User Id, Password, Retype Same password and Hints. And Click on Login Button. | **DNBSN** will display error message. And Failed to Registration. |
| **DNBSN**-013 |  |  |  |  |  | Step4 | Enter new User Id, Password, Retype Different password and Hints. And Click on Login Button. | **DNBSN** will display error message. And Failed to Registration. |
| **DNBSN**-014 |  |  |  |  |  | Step5 | Enter new User Id, Password, Retype Same password and enter nothing in Hints fields. And Click on Login Button. | **DNBSN** will display error message. And Failed to Registration. |
| **DNBSN**-015 |  |  |  |  |  | Step6 | Enter new User Id, Password, Hints and nothing in Retype password field. And Click on Login Button. | **DNBSN** will display error message. And Failed to Registration. |
| **DNBSN**-016 |  |  |  |  |  | Step7 | Enter nothing new User Id, Password, Retype password and Hints Fields. And Click on Login Button. | **DNBSN** will display error message. And Failed to Registration. |
| **DNBSN**-017 |  |  |  |  |  | Step8 | Enter Proper new User Id, Password, Retype Same password and Hints Fields. And Click on Login Button. | Successful Registration is done and this area is closed and come to login area. |
| **DNBSN**-018 | Manual | f0657bbdf47e26ec481fa172b0fa76f9becb2681 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check Successful Event Creator in **DNBSN** text field. | The purpose of this test is to check event insertion process. | Step1 | Insert Nothing in Event field, select on site where want to share. And click on Save note Button. | Insertion failed. Display error message. |
| **DNBSN**-019 |  |  |  |  |  | Step2 | Insert Note in Event field, if require change the current date, select on site where want to share. And click on Save note Button. | Successful Insertion. And Share with selected social site. Display success message. |
| **DNBSN**-020 |  |  |  |  |  | Step3 | Create note and Click the Clear Field button. | Successful to clear Note fields. |
| **DNBSN**-021 |  |  |  |  |  | Step4 | Click on Delete Event button without selecting event. | Display error message. |
| **DNBSN**-022 |  |  |  |  |  | Step5 | Select Delete Event and Click on Delete button. | Deleted successfully. |
| **DNBSN**-023 |  |  |  |  |  | Step6 | Click on Rewrite Event button without selecting event. | Display error message. |
| **DNBSN**-024 |  |  |  |  |  | Step7 | Select an information and Click on Rewrite Event button. | Asking to edit and click on Save Note to successfully update data. |
| **DNBSN**-025 | Manual | e22bd0e470f145f3db336ed9e28d474d8f4637d7 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check Successful Event Display in **DNBSN** Various view. | The purpose of this test is to verify that the Inserted event data can display in various View. | Step1 | Click the Monthly view under Note Book tab. | Display Event in monthly view successfully. |
| **DNBSN**-026 |  |  |  |  |  | Step2 | Click the View All under Note Book tab. | Display All Event in View All successfully. |
| **DNBSN**-027 |  |  |  |  |  | Step3 | Click the Daily View under Note Book tab. | Display All Event in Daily View successfully. |
| **DNBSN**-028 |  |  |  |  |  | Step4 | Click the Weekly View under Note Book tab. | Display All Event in Weekly View successfully. |
| **DNBSN**-029 | Manual | 90330b92328d862892fc77436539081cc2b7f70d | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check Sorting for Event Display in **DNBSN** Various view. | The purpose of this test is to verify that the data in various field is sorted. | Step1 | Click the Monthly view under Note Book tab. | All Data are sorted. |
| **DNBSN**-030 |  |  |  |  |  | Step2 | Click the View All under Note Book tab. | All Data are sorted. |
| **DNBSN**-031 |  |  |  |  |  | Step3 | Click the Daily View under Note Book tab. | All Data are sorted. |
| **DNBSN**-032 |  |  |  |  |  | Step4 | Click the Weekly View under Note Book tab. | All Data are sorted. |
| **DNBSN**-033 | Manual | 47fb570f63ffec837a49e235f629c49cc55a70f0 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check social site status. | It is to check that we can get and send post or tweet to a particular social site. | Step1 | Click Facebook under Social Site tab. And enter wrong Login id And Password. | Failed to connect with Facebook. And display error message. |
| **DNBSN**-034 |  |  |  |  |  | Step2 | Click Facebook under Social Site tab. And enter correct Login id And Password. | Connected successfully with the Facebook. |
| **DNBSN**-035 |  |  |  |  |  | Step3 | Enter Event in Post field. And Click on post button. | Successfully posted in Facebook. |
| **DNBSN**-036 |  |  |  |  |  | Step4 | Click Twitter under Social Site tab. And enter wrong Login id And Password. | Failed to connect with Twitter. And display error message. |
| **DNBSN**-037 |  |  |  |  |  | Step5 | Click Twitter under Social Site tab. And enter correct Login id And Password. | Connected successfully with the Twitter. |
| **DNBSN**-038 |  |  |  |  |  | Step6 | Create Tweet in Post field. And Click on Submit button. | Successfully Tweeted in Twitter. |
| **DNBSN**-039 | Manual | b8a65899fa408a4d99a8ee8cabbe2ac0b54226b0 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check social Google map. | It is to check that we can get the location from google map site. | Step1 | Without any location in location field and click on Load Map button. | Load the entire google map. |
| **DNBSN**-040 |  |  |  |  |  | Step2 | Enter location and click on Load Map button. | Successfully find the location. |
| **DNBSN**-041 | Manual | 6c53de9297011864bd6c07bdcce83415a76e8bd4 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check everything about contact information. | It is to check that contact works properly. | Step1 | Click on Contact tab. | Contact is display all saved data. |
| **DNBSN**-042 |  |  |  |  |  | Step2 | Click on Add Contact Expender and enter incorrect data or empty the mandatory field. | Failed to store contact information in database and display the error message. |
| **DNBSN**-043 |  |  |  |  |  | Step3 | Click on Add Contact Expender and enter correct data in the mandatory field. | Successfully Added contact Information. And display in View area. |
| **DNBSN**-044 |  |  |  |  |  | Step4 | Click on delete button without selecting details. | Display error message. |
| **DNBSN**-045 |  |  |  |  |  | Step5 | Select an information and Click on Delete button. | Deleted successfully. |
| **DNBSN**-046 |  |  |  |  |  | Step6 | Click on Edit button without selecting details. | Display error message. |
| **DNBSN**-047 |  |  |  |  |  | Step7 | Select an information and Click on Edit button. | Expender Opened and asking to edit and click on save to successfully update data. |
| **DNBSN**-048 | Manual | 90330b92328d862892fc77436539081cc2b7f70d | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check everything about Password Manager information. | It is to check that Password Manager works properly. | Step1 | Click on Password tab. | Password is asking for login password to access all information. |
| **DNBSN**-049 |  |  |  |  |  | Step2 | Insert wrong User Name and Password. And Click on Login Button. | **Password Manager** of **DNBSN** will display error message. And Failed to Login. |
| **DNBSN**-050 |  |  |  |  |  | Step3 | Insert correct User Name and Password. And Click on Login Button. | **Password Manager** login successfully. And display all Password related information. |
| **DNBSN**-051 |  |  |  |  |  | Step4 | Click on Add Info Expender and enter incorrect data or empty the mandatory field. | Failed to store Info information in database and display the error message. |
| **DNBSN**-052 |  |  |  |  |  | Step5 | Click on Add Info Expender and enter correct data in the mandatory field. | Successfully Added Info Information. And display in View area. |
| **DNBSN**-053 |  |  |  |  |  | Step6 | Click on delete button without selecting details. | Display error message. |
| **DNBSN**-054 |  |  |  |  |  | Step7 | Select an information and Click on Delete button. | Deleted successfully. |
| **DNBSN**-055 |  |  |  |  |  | Step8 | Click on Edit button without selecting details. | Display error message. |
| **DNBSN**-056 |  |  |  |  |  | Step9 | Select an information and Click on Edit button. | Expender Opened and asking to edit and click on Save to successfully update data. |
| **DNBSN**-057 |  |  |  |  |  | Step10 | Click on logout button to logout from password section. | Successfully Logout from Password. |
| **DNBSN**-058 | Manual | 90330b92328d862892fc77436539081cc2b7f70d | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check everything about Task information. | It is to check that Task works properly. | Step1 | Click on Task tab under Extra Tab. | Display all Task related information. |
| **DNBSN**-059 |  |  |  |  |  | Step2 | Click on Add Task Expender and enter incorrect data or empty the mandatory field. | Failed to store Task Info information in database and display the error message. |
| **DNBSN**-060 |  |  |  |  |  | Step3 | Click on Add Task Expender and enter correct data in the mandatory field. | Successfully Added Task Info Information. And display in View area. |
| **DNBSN**-061 |  |  |  |  |  | Step4 | Click on delete button without selecting details. | Display error message. |
| **DNBSN**-062 |  |  |  |  |  | Step5 | Select an information and Click on Delete button. | Deleted successfully. |
| **DNBSN**-063 |  |  |  |  |  | Step6 | Click on Edit button without selecting details. | Display error message. |
| **DNBSN**-064 |  |  |  |  |  | Step7 | Select an information and Click on Edit button. | Expender Opened and asking to edit and click on Save to successfully update data. |
| **DNBSN**-065 | Manual | e22bd0e470f145f3db336ed9e28d474d8f4637d7 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Search event by Date in **DNBSN**. | The purpose of this test is to verify that the Search option is working. | Step1 | Select a date and Click on Go To Date Button. | Display all events of the particular date . |
| **DNBSN**-066 | Manual | f3563be0a9c431104f52839039e86043cf640cf1 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check Logout. | It is to check that Logout works properly. | Step1 | Click on logout button to logout from **DNBSN** section. | Successfully Logout from **DNBSN**. |

### System Test Cases

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case Id | Type | Github ID | Subject | Test Name | Test Description | Step Name | Description | Expected Result |
| **DNBSN**-067 | Manual | f3563be0a9c431104f52839039e86043cf640cf1 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check Log in. | It is to check that Login works properly. | Step1 | Click on Login button after inserting invalid User id and password from **DNBSN**. | Login failed to **DNBSN**. And can’t able to use the feature. |
| **DNBSN**-068 |  |  |  |  |  | Step2 | Click on Login button after inserting valid User id and password from **DNBSN**. | Successfully Login to **DNBSN**. And can able to use the feature. |
| **DNBSN**-069 | Manual | d01197ee4cd3bee9245874b5937ba740019fd131 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check Successful Registration for New **DNBSN** User. | The purpose of this test is to verify that the all new connection could be creating new Account By Registration. | Step1 | Click on Registration link. | New Account creation area is opened. |
| **DNBSN**-070 |  |  |  |  |  | Step2 | Click on Registration button after inserting invalid information from **DNBSN**. | Registration failed to **DNBSN**. And can’t able to use the feature. |
| **DNBSN**-071 |  |  |  |  |  | Step3 | Click on Registration button after inserting valid information from **DNBSN**. | Registration Successfully done to **DNBSN**. |
| **DNBSN**-072 |  |  |  |  |  | Step4 | Click on Login button after inserting newly created valid User id and password from **DNBSN**. | Successfully Login to by new User Id And password **DNBSN**. And can able to use the feature. |
| **DNBSN**-073 | Manual | f0657bbdf47e26ec481fa172b0fa76f9becb2681 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check Successful Event Manage in **DNBSN**. | The purpose of this test is to check event insertion process. | Step1 | Login to **DNBSN** with valid User Id And Password. And compose event select the social site. Click on Save Note button. | Successfully Inserted event. Display the event in shared social site. |
| **DNBSN**-074 |  |  |  |  |  | Step2 | Check various view like monthly view, view all, weekly view, daily view to Show saved event. | Successfully Display events in monthly view, view all, daily view, weekly view. |
| **DNBSN**-075 |  |  |  |  |  | Step3 | Select an event from monthly view or view all or daily view or weekly view .And click on Delete Event. | Successfully Deleted events from monthly view, view all, daily view, weekly view. |
| **DNBSN**-076 |  |  |  |  |  | Step4 | Select an event from monthly view or view all or daily view or weekly view .And click on Rewrite Event. | Event ready in Edit field and after editing click on save note button to update the event. And Successfully Display edited event in monthly view, view all, daily view, weekly view. |
| **DNBSN**-077 | Manual | 47fb570f63ffec837a49e235f629c49cc55a70f0 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check social site status and update event. | It is to check that we can get and send post or tweet to a particular social site. | Step1 | Click Facebook under Social Site tab. And enter valid Login id And Password to connect and create event and click on post button. | Successfully connected to Facebook and can post in Facebook wall. |
| **DNBSN**-078 |  |  |  |  |  | Step2 | Select a post and click on delete button to delete post from Facebook wall. | Post Successfully deleted. |
| **DNBSN**-079 |  |  |  |  |  | Step3 | Click Twitter under Social Site tab. And enter valid Login id And Password to connect and create event and click on Tweet button. | Successfully connected to Twitter and can Tweet in Twitter wall. |
| **DNBSN**-080 |  |  |  |  |  | Step4 | Select a Tweet and click on delete button to delete post from Twitter wall. | Tweet Successfully deleted. |
| **DNBSN**-081 | Manual | b8a65899fa408a4d99a8ee8cabbe2ac0b54226b0 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check social Google map. | It is to check that we can get the location from Google map site. | Step1 | Without any location in location field and click on Load Map button. | Load the entire Google map. |
| **DNBSN**-082 |  |  |  |  |  | Step2 | Enter location and click on Load Map button. | Successfully find the particular location. |
| **DNBSN**-083 | Manual | 6c53de9297011864bd6c07bdcce83415a76e8bd4 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check everything about contact information. | It is to check that contact works properly. | Step1 | Click on Contact tab. Compose contact information and click on Save Button. | Contact display all saved data. And created contact successfully. |
| **DNBSN**-084 |  |  |  |  |  | Step2 | Select Contact from list view area .And click on Delete Event. | Successfully Deleted Contact from list view area. |
| **DNBSN**-085 |  |  |  |  |  | Step3 | Select Contact from list view area. And click on Edit button. | Contact ready in Edit field and after editing click on Save button to update the Contact. And Successfully Display edited Contact in list view area. |
| **DNBSN**-086 | Manual | 90330b92328d862892fc77436539081cc2b7f70d | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check everything about Password Manager information. | It is to check that Password Manager works properly. | Step1 | Click on Password tab. Compose Password information and click on Save Button. | Password display all saved data. And created Password successfully. |
| **DNBSN**-087 |  |  |  |  |  | Step2 | Select Password from list view area .And click on Delete Password. | Successfully Deleted Password from list view area. |
| **DNBSN**-089 |  |  |  |  |  | Step3 | Select Password from list view area. And click on Edit button. | Password ready in Edit field and after editing click on Save button to update the Password. And Successfully Display edited Password in list view area. |
| **DNBSN**-090 | Manual | 90330b92328d862892fc77436539081cc2b7f70d | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check everything about Task information. | It is to check that Task works properly. | Step1 | Click on Task tab Under Extra Tab. Compose Task information and click on Save Button. | Task display all saved data. And created Task successfully. |
| **DNBSN**-091 |  |  |  |  |  | Step2 | Select Task from list view area .And click on Delete Task. | Successfully Deleted Task from list view area. |
| **DNBSN**-092 |  |  |  |  |  | Step3 | Select Task from list view area. And click on Edit button. | Password ready in Edit field with status option and after editing click on Save button to update the Task. And Successfully Display edited Task with status in list view area. |
| **DNBSN**-093 | Manual | f3563be0a9c431104f52839039e86043cf640cf1 | E:\DEVELOPERS\_ZONE\GitHub\DailyNoteBook\code | Check logout. | It is to check that Logout works properly. | Step1 | Click on logout button to logout from **DNBSN** section. | Successfully Logout from **DNBSN**. And can’t able to use the feature any more. |

# Coding

## Complete Project Coding

### DNBSN GUI Design Coding: DNBSNGUI

|  |
| --- |
| MainWindow.xaml |
| <Window x:Class="DNBSNGUI.MainWindow"  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  xmlns:sys="clr-namespace:System;assembly=mscorlib"  xmlns:ViewEvent="clr-namespace:DNBSNGUI"  DataContext="{Binding RelativeSource={RelativeSource Self}}"  Title="DNBSN"  MinHeight="650" MinWidth="1250" AllowsTransparency="True" ResizeMode="CanResizeWithGrip" Background="Transparent"  WindowStyle="None" WindowStartupLocation="CenterScreen" Icon="/DNBSNGUI;component/Images/monotone\_pen\_write.png" Loaded="Window\_Loaded" >  <Window.Resources>  <ResourceDictionary>  <ResourceDictionary.MergedDictionaries>  <ResourceDictionary Source="/DNBSNStyles;component/ControlStyle.xaml" />  </ResourceDictionary.MergedDictionaries>  </ResourceDictionary>  </Window.Resources>  <Border Margin="10">  <Border.Effect>  <DropShadowEffect Color="Black"  Direction="270"  BlurRadius="15"  ShadowDepth="3" />  </Border.Effect>  <DockPanel LastChildFill="True" Margin="2" Background="White">  <DockPanel DockPanel.Dock="Top" LastChildFill="True">  <UniformGrid DockPanel.Dock="Right" HorizontalAlignment="Right" Rows="1">  <Button Width="30" Height="30" Name="minimizeBtn" ToolTip="minimize" Style="{StaticResource TitlebarBTN}" Click="minimizeBtn\_Click">\_</Button>  <Button Width="30" Height="30" Name="closeBtn" ToolTip="close" Style="{StaticResource TitlebarBTN}" Click="closeBtn\_Click">X</Button>  </UniformGrid>  <UniformGrid DockPanel.Dock="Right" HorizontalAlignment="Right">  <Button Margin="0" Width="70" Height="30" Name="logoutDnbsnBtn" IsEnabled="False" Style="{StaticResource TitlebarBTN}" Click="logoutDnbsnBtn\_Click"></Button>  </UniformGrid>  <Label DockPanel.Dock="Left" Name="hntLginLbl" Style="{StaticResource LblStyle}"> Login first to use Daily Note Book with Social Networking Updater </Label>  </DockPanel>  <UniformGrid DockPanel.Dock="Bottom" Rows="1" Background="WhiteSmoke">  <TextBlock Height="25" Padding="7,1,0,0" VerticalAlignment="Center" Text="{Binding Source={x:Static sys:DateTime.Today}, StringFormat='{}{0:dddd, MMMM dd, yyyy}'}" Foreground="Black" ></TextBlock>  <TextBlock Height="25" Padding="7,1,10,0" HorizontalAlignment="Right">Copyright 2013, Anirban Nandy</TextBlock>  </UniformGrid>  <DockPanel DockPanel.Dock="Top" LastChildFill="True">  <DockPanel >  <Expander Name="loginExpndr" DockPanel.Dock="Top" IsExpanded="True" Header="Login in DNBSN" Padding="2,0,0,0">  <UniformGrid Columns="1" >  <DockPanel Height="149" Width="186">  <DockPanel LastChildFill="True" DockPanel.Dock="Top">  <Label DockPanel.Dock="left" Width="70" Style="{StaticResource LblStyle}">User ID :</Label>  <TextBox x:Name="dNBSNUserIDTB" DockPanel.Dock="Right" Margin="2" ToolTip="enter user id" Tag="enter user name" Style="{StaticResource commonTBtyle}"/>  </DockPanel>  <DockPanel LastChildFill="True" DockPanel.Dock="Top">  <Label DockPanel.Dock="left" Width="70" Style="{StaticResource LblStyle}">Password :</Label>  <PasswordBox Name="dNBSNpassPB" Margin="2"></PasswordBox>  </DockPanel>  <DockPanel LastChildFill="True" DockPanel.Dock="Top">  <TextBlock Text="New User !! " DockPanel.Dock="left" VerticalAlignment="Center" >  <Hyperlink Click="createAcHp\_Click"> Register</Hyperlink>  </TextBlock>  <Button Name="dNBSNLoginBtn" ToolTip="login" Style="{StaticResource PlainBtnStyle}" Click="dNBSNLoginBtn\_Click">Login</Button>  </DockPanel>  <DockPanel DockPanel.Dock="Bottom">  <UniformGrid Rows="2">  <Label Name="errorMsgLbl" Foreground="Red"></Label>  <Label Name="hintsMsgLbl" Foreground="Red"></Label>  </UniformGrid>  </DockPanel>  </DockPanel>  </UniformGrid>  </Expander>  <Expander Name="newAcExpndr" DockPanel.Dock="Top" IsExpanded="False" Visibility="Collapsed" Header="Create New Account" Padding="2,0,0,0">  <UniformGrid Columns="1" >  <DockPanel Height="175" Width="186">  <DockPanel LastChildFill="True" DockPanel.Dock="Top">  <Label DockPanel.Dock="left" Width="70" Style="{StaticResource LblStyle}">User ID :</Label>  <TextBox x:Name="dNBSNNewUserIDTB" DockPanel.Dock="Right" Margin="2" ToolTip="enter user id" Tag="enter user name" Style="{StaticResource commonTBtyle}"/>  </DockPanel>  <DockPanel LastChildFill="True" DockPanel.Dock="Top">  <Label DockPanel.Dock="left" Width="70" Style="{StaticResource LblStyle}">Password :</Label>  <PasswordBox Name="dNBSNNewPassPB" Margin="2"></PasswordBox>  </DockPanel>  <DockPanel LastChildFill="True" DockPanel.Dock="Top">  <Label DockPanel.Dock="left" Width="70" Style="{StaticResource LblStyle}">Re Type :</Label>  <PasswordBox Name="dNBSNNewRepassPB" Margin="2"></PasswordBox>  </DockPanel>  <DockPanel LastChildFill="True" DockPanel.Dock="Top"></DockPanel>  <DockPanel LastChildFill="True" DockPanel.Dock="Top">  <Label DockPanel.Dock="left" Width="70" Style="{StaticResource LblStyle}">Hints :</Label>  <TextBox x:Name="dNBSNHintsTB" Margin="2" ToolTip="enter hints" Tag="enter hints" Style="{StaticResource commonTBtyle}"></TextBox>  </DockPanel>  <DockPanel DockPanel.Dock="Top">  <TextBlock Text="Already User !! " DockPanel.Dock="left" VerticalAlignment="Center" >  <Hyperlink Click="goLogin\_Click">Login</Hyperlink>  </TextBlock>  <Button Name="dNBSSubmitBtn" ToolTip="submit" Style="{StaticResource PlainBtnStyle}" Click="dNBSSubmitBtn\_Click">Submit</Button>  </DockPanel>  <DockPanel DockPanel.Dock="Bottom">  <Label Name="createErrorMsgLbl" Foreground="Red"></Label>  </DockPanel>  </DockPanel>  </UniformGrid>  </Expander>  <Expander Name="mainLeftExpndr" IsEnabled="False" IsExpanded="False" Width="200" DockPanel.Dock="Top" Padding="2,0,0,0" ExpandDirection="Right" Header="H&#x0a; I&#x0a;D&#x0a;E&#x0a;/&#x0a;U&#x0a;N&#x0a;H&#x0a; I&#x0a;D&#x0a;E">  <UniformGrid Columns="1">  <DockPanel>  <Calendar DockPanel.Dock="Top" BorderBrush="#FFE9E0E0">    </Calendar>  <TextBox Height="100" TextWrapping="Wrap" AcceptsReturn="True" VerticalScrollBarVisibility="Visible" SpellCheck.IsEnabled="True">  <TextBox.Style>  <Style TargetType="TextBox" xmlns:sys="clr-namespace:System;assembly=mscorlib">  <Style.Resources>  <VisualBrush x:Key="CueBannerBrush" AlignmentX="Left" AlignmentY="Top" Stretch="None">  <VisualBrush.Visual>  <Label Content="what's on your mind?" Foreground="LightGray" />  </VisualBrush.Visual>  </VisualBrush>  </Style.Resources>  <Style.Triggers>  <Trigger Property="Text" Value="{x:Static sys:String.Empty}">  <Setter Property="Background" Value="{StaticResource CueBannerBrush}" />  </Trigger>  <Trigger Property="Text" Value="{x:Null}">  <Setter Property="Background" Value="{StaticResource CueBannerBrush}" />  </Trigger>  <Trigger Property="IsKeyboardFocused" Value="True">  <Setter Property="Background" Value="White" />  </Trigger>  </Style.Triggers>  </Style>  </TextBox.Style>  </TextBox>  </DockPanel>  <DockPanel LastChildFill="True">  <UniformGrid DockPanel.Dock="Bottom" Rows="1">  <TextBox DockPanel.Dock="Bottom" Height="40" Width="85" TextWrapping="Wrap" AcceptsReturn="True" VerticalScrollBarVisibility="Visible" SpellCheck.IsEnabled="True">  <TextBox.Style>  <Style TargetType="TextBox" xmlns:sys="clr-namespace:System;assembly=mscorlib">  <Style.Resources>  <VisualBrush x:Key="CueBannerBrush" AlignmentX="Left" AlignmentY="Top" Stretch="None">  <VisualBrush.Visual>  <Label Content="keyword" Foreground="LightGray" />  </VisualBrush.Visual>  </VisualBrush>  </Style.Resources>  <Style.Triggers>  <Trigger Property="Text" Value="{x:Static sys:String.Empty}">  <Setter Property="Background" Value="{StaticResource CueBannerBrush}" />  </Trigger>  <Trigger Property="Text" Value="{x:Null}">  <Setter Property="Background" Value="{StaticResource CueBannerBrush}" />  </Trigger>  <Trigger Property="IsKeyboardFocused" Value="True">  <Setter Property="Background" Value="White" />  </Trigger>  </Style.Triggers>  </Style>  </TextBox.Style>  </TextBox>  <Button Name="goBtn" ToolTip="go" Content="Go" Style="{StaticResource PlainBtnStyle}">  <Button.LayoutTransform>  <RotateTransform Angle="45" />  </Button.LayoutTransform>  </Button>  </UniformGrid>  <Button Name="updateBtn" DockPanel.Dock="Right" ToolTip="update" Height="150" Style="{StaticResource PlainBtnStyle}">Update</Button>  <UniformGrid Columns="1">  <CheckBox Name="facebookCheckBox" VerticalAlignment="Center" Style="{StaticResource CheckBtyle}">Facebook</CheckBox>  <CheckBox Name="twitterCheckBox" VerticalAlignment="Center" Style="{StaticResource CheckBtyle}">Twitter</CheckBox>  <CheckBox Name="linkedInCheckBox" VerticalAlignment="Center" Style="{StaticResource CheckBtyle}">LinkedIn</CheckBox>  <CheckBox Name="googlePlusCheckBox" VerticalAlignment="Center" Style="{StaticResource CheckBtyle}">Google+</CheckBox>  <CheckBox Name="foursquareCheckBox" VerticalAlignment="Center" Style="{StaticResource CheckBtyle}">Foursquare</CheckBox>  <CheckBox Name="googlecalendarCheckBox" VerticalAlignment="Center" Style="{StaticResource CheckBtyle}">  <TextBlock TextWrapping="Wrap" Text="Google Calendar"/>  </CheckBox>  <CheckBox Name="myNoteBookCheckBox" VerticalAlignment="Center" Style="{StaticResource CheckBtyle}" IsChecked="True">  <TextBlock TextWrapping="Wrap" Text="My Note Book"/>  </CheckBox>  </UniformGrid>  </DockPanel>  </UniformGrid>  </Expander>  </DockPanel>  <TabControl Name="mainTabControl" IsEnabled="False" Margin="3" BorderBrush="#FFE9E0E0">  <TabItem Name="notesTabItem" Style="{StaticResource TItemStyle}" Header="Note Book" Background="red">    <DockPanel LastChildFill="True">  <DockPanel DockPanel.Dock="Top" >    <DockPanel DockPanel.Dock="Bottom" LastChildFill="True">    <UniformGrid Rows="1" >  <DatePicker ToolTip="today" Margin="10" Name="noteDateDP" SelectedDate="{x:Static sys:DateTime.Now}" Style="{StaticResource DatepkrStyle}"></DatePicker>  <Label Name="noteSuccessMsgLvl" Foreground="Red" Style="{StaticResource LblStyle}"></Label>  <Button Name="clearFieldsBtn" Content="Clear Fields" Style="{StaticResource PlainBtnStyle}" Click="clearFieldsBtn\_Click" />  <Button Name="saveNotesBtn" Content="Save Notes" Style="{StaticResource PlainBtnStyle}" Click="saveNotesBtn\_Click" />  </UniformGrid>    </DockPanel>  <TextBox Name="notesTB" DockPanel.Dock="Top" TextBlock.LineHeight="20" TextBlock.LineStackingStrategy="BlockLineHeight" Padding="20,10,20,20" Height="100">    <TextBox.Background>  <DrawingBrush TileMode="Tile" Stretch="None" Viewport="0,0,20,20" ViewportUnits="Absolute" Opacity="1">  <DrawingBrush.Drawing>  <GeometryDrawing>  <GeometryDrawing.Pen>  <Pen Brush="RoyalBlue" Thickness=".1" />  </GeometryDrawing.Pen>  <GeometryDrawing.Geometry>  <LineGeometry StartPoint="0,0" EndPoint="20,0"/>  </GeometryDrawing.Geometry>  </GeometryDrawing>  </DrawingBrush.Drawing>  </DrawingBrush>  </TextBox.Background>  <TextBox.Style>  <Style TargetType="TextBox" xmlns:sys="clr-namespace:System;assembly=mscorlib">  <Style.Resources>  <VisualBrush x:Key="CueBannerBrush" AlignmentX="Left" AlignmentY="Top" Stretch="None">  <VisualBrush.Visual>  <Label Content="what's on your mind?" Foreground="LightGray" />  </VisualBrush.Visual>  </VisualBrush>  </Style.Resources>  <Style.Triggers>  <Trigger Property="Text" Value="{x:Static sys:String.Empty}">  <Setter Property="Background" Value="{StaticResource CueBannerBrush}" />  </Trigger>  <Trigger Property="Text" Value="{x:Null}">  <Setter Property="Background" Value="{StaticResource CueBannerBrush}" />  </Trigger>  <Trigger Property="IsKeyboardFocused" Value="True">  <Setter Property="Background" Value="White" />  </Trigger>  </Style.Triggers>  </Style>  </TextBox.Style>  <!--Start writing from here...-->  </TextBox>  </DockPanel>    <DockPanel DockPanel.Dock="Left">  <UniformGrid DockPanel.Dock="Left" Columns="1">  <DockPanel LastChildFill="True">  <UniformGrid Rows="1" DockPanel.Dock="Bottom">  <Button Content="&lt;" Style="{StaticResource PlainBtnStyle}" Name="previousDateBtn" Click="previousDateBtn\_Click"></Button>  <Button Content="&gt;" Style="{StaticResource PlainBtnStyle}"></Button>  </UniformGrid>  <TextBlock Style="{StaticResource TxtblkStyle}" DockPanel.Dock="Top" Text="{Binding Source={x:Static sys:DateTime.Today}, StringFormat='{}{0:MMMM yyyy}'}" FontSize="18"></TextBlock>  </DockPanel>  <Button Name="rewriBtnBtn" ToolTip="save to cxcel" Style="{StaticResource PlainBtnStyle}">Rewrite Event</Button>  <Button Name="dltevntBtnBtn" ToolTip="save to cxcel" Style="{StaticResource PlainBtnStyle}">Delete Event</Button>  <Button Name="saveToExcelBtn" ToolTip="save to cxcel" Style="{StaticResource PlainBtnStyle}">Save To Excel</Button>  <DatePicker ToolTip="select a date" Margin="10" Name="goToDateDP" SelectedDate="{x:Static sys:DateTime.Now}" Style="{StaticResource DatepkrStyle}" ></DatePicker>  <Button Name="goToDateBtn" ToolTip="go to date" Style="{StaticResource PlainBtnStyle}" Click="goToDateBtn\_Click">Go To Date</Button>  </UniformGrid>  </DockPanel>    <DockPanel>  <TabControl Name="noteControlsTab" Margin="3" BorderBrush="#FFE9E0E0" TabStripPlacement="Right" HorizontalContentAlignment="Left" >  <TabItem Name="monthlyViewTC" Style="{StaticResource TItemStyle}" Header="Monthly View" Width="150" MouseRightButtonDown="monthlyViewTC\_click">  <DockPanel LastChildFill="True">    <UniformGrid Rows="1" Columns="7" DockPanel.Dock="Top">  <Label>Sunday</Label>  <Label>Monday</Label>  <Label>Tuesday</Label>  <Label>Wednesday</Label>  <Label>Thursday</Label>  <Label>Friday</Label>  <Label>Saturday</Label>  </UniformGrid>  <UniformGrid Name="dateUG" Rows="6" Columns="7" DockPanel.Dock="Bottom">    </UniformGrid>  </DockPanel>  </TabItem>  <TabItem Name="bItem" Style="{StaticResource TItemStyle}" Header="View All" Width="150">  <DockPanel LastChildFill="True">  <ListView Name="allnoteView" HorizontalAlignment="Stretch" ItemsSource="{Binding allnoteCollection}" >  <ListView.View>  <GridView>  <GridViewColumn Width="100" Header="Date" DisplayMemberBinding="{Binding noteDate}"/>  <GridViewColumn Width="650" Header="Event" DisplayMemberBinding="{Binding note}" />  </GridView>  </ListView.View>  </ListView>  </DockPanel>  </TabItem>  <TabItem Name="linktem" Style="{StaticResource TItemStyle}" Header="Daily View" Width="150">  <DockPanel LastChildFill="True">    <ListView></ListView>  </DockPanel>  </TabItem>  <TabItem Name="googlItem" Style="{StaticResource TItemStyle}" Header="Weekly View" Width="150">  <DockPanel LastChildFill="True">    <ListView></ListView>  </DockPanel>  </TabItem>    </TabControl>  </DockPanel>  </DockPanel>  </TabItem>  <TabItem Name="mainSocialSiteTabItem" Style="{StaticResource TItemStyle}" Header="Social Site" >  <TabControl Name="socialSiteTabControl" Margin="3" BorderBrush="#FFE9E0E0" TabStripPlacement="Left" HorizontalContentAlignment="Left" >  <TabItem Name="fbTabItem" Style="{StaticResource TItemStyle}" Header="Facebook" ToolTip="facebook">  <DockPanel LastChildFill="True">  <StackPanel DockPanel.Dock="Top" Orientation="Horizontal">  <Expander Name="loginToFBExpndr" DockPanel.Dock="Top" ExpandDirection="Down" IsExpanded="False" Header="Login to Facebook" VerticalAlignment="Center">  <UniformGrid Columns="1">  <DockPanel Width="600" >  <UniformGrid Columns="3" DockPanel.Dock="Left">  <Label Margin="10" Width="150">Facebook User Name</Label>  <Label Margin="10" Width="150">Password</Label>  <Label></Label>  <TextBox Margin="10" Width="150" Tag="enter user name" Style="{StaticResource commonTBtyle}" />  <PasswordBox Name="fbPB" Tag="••••••" Style="{StaticResource pBstyle}" Width="150" Margin="10"></PasswordBox>  <Button Name="fbLoginBtn" Width="100" ToolTip="login" Style="{StaticResource PlainBtnStyle}">Login</Button>  <Label></Label>  <CheckBox Name="fbStayLoginCheckBox" Margin="10" Width="150">Stay logged in</CheckBox>  <Label></Label>  </UniformGrid>  </DockPanel>  </UniformGrid>  </Expander>  <Button Content="Refresh Facebook Status" Style="{StaticResource PlainBtnStyle}" Name="fbRefreshButton" />  </StackPanel>  <ListView></ListView>  </DockPanel>  </TabItem>  <TabItem Name="twitterTabItem" Style="{StaticResource TItemStyle}" Header="Twitter" ToolTip="twitter">  <DockPanel LastChildFill="True">  <StackPanel DockPanel.Dock="Top" Orientation="Horizontal">  <Expander Name="logintotwitterExpndr" DockPanel.Dock="Top" ExpandDirection="Down" IsExpanded="False" Header="Login to Twitter" VerticalAlignment="Center">  <UniformGrid Columns="1">  <DockPanel Width="600" >  <UniformGrid Rows="1">  <Button Content="Launch Twitter App" Name="launchTwitterAppButton" Style="{StaticResource PlainBtnStyle}" Click="launchTwitterAppButton\_Click" />  <TextBox Name="verificationCodeTxtBox" Tag="enter apps code" Margin="20,8,0,8" Style="{StaticResource commonTBtyle}"></TextBox>  <Button Name="verifyTwitterApp" Style="{StaticResource PlainBtnStyle}" Click="verifyTwitterApp\_Click">Verify Twitter Apps Code</Button>  </UniformGrid>    </DockPanel>  </UniformGrid>  </Expander>  </StackPanel>  <TextBlock Name="tweetsTextBlock" Foreground="#FF8E6A6A" Background="#FFD06D6D" OpacityMask="#FF2F545F" />  </DockPanel>  </TabItem>  <TabItem Name="linkedInTabItem" Style="{StaticResource TItemStyle}" Header="LinkedIn" ToolTip="linkedIn" >  <DockPanel LastChildFill="True">  <StackPanel DockPanel.Dock="Top" Orientation="Horizontal">  <Expander DockPanel.Dock="Top" ExpandDirection="Down" IsExpanded="False" Header="Login" VerticalAlignment="Center">  <UniformGrid Columns="1">  <DockPanel Width="600" >  <UniformGrid Columns="3" DockPanel.Dock="Left">  <Label Margin="10" Width="150">LinkedIn User Name</Label>  <Label Margin="10" Width="150">Password</Label>  <Label></Label>  <TextBox Margin="10" Width="150" Tag="enter user name" Style="{StaticResource commonTBtyle}" />  <PasswordBox Width="150" Margin="10"></PasswordBox>  <Button Width="100" ToolTip="login" Style="{StaticResource PlainBtnStyle}">Login</Button>  <Label></Label>  <CheckBox Margin="10" Width="150">Stay logged in</CheckBox>  <Label></Label>  </UniformGrid>  </DockPanel>  </UniformGrid>  </Expander>  <Button Content="Refresh LinkedIn Status" ToolTip="refresh" Style="{StaticResource PlainBtnStyle}"/>  </StackPanel>  <ListView></ListView>  </DockPanel>  </TabItem>  <TabItem Name="googlePlusTabItem" Style="{StaticResource TItemStyle}" Header="Google Plus" ToolTip="google+">  <DockPanel LastChildFill="True">  <StackPanel DockPanel.Dock="Top" Orientation="Horizontal">  <Expander DockPanel.Dock="Top" ExpandDirection="Down" IsExpanded="False" Header="Login" VerticalAlignment="Center">  <UniformGrid Columns="1">  <DockPanel Width="600" >  <UniformGrid Columns="3" DockPanel.Dock="Left">  <Label Margin="10" Width="150">Google+ User Name</Label>  <Label Margin="10" Width="150">Password</Label>  <Label></Label>  <TextBox Margin="10" Width="150" Tag="enter user name" Style="{StaticResource commonTBtyle}" />  <PasswordBox Width="150" Margin="10"></PasswordBox>  <Button Width="100" ToolTip="login" Style="{StaticResource PlainBtnStyle}">Login</Button>  <Label></Label>  <CheckBox Margin="10" Width="150">Stay logged in</CheckBox>  <Label></Label>  </UniformGrid>  </DockPanel>  </UniformGrid>  </Expander>  <Button Content="Refresh Google+ Status" ToolTip="refresh" Style="{StaticResource PlainBtnStyle}"/>  </StackPanel>  <ListView></ListView>  </DockPanel>  </TabItem>  <TabItem Name="foursquareTabItem" Style="{StaticResource TItemStyle}" Header="Foursquare" ToolTip="foursquare">  <DockPanel LastChildFill="True">  <StackPanel DockPanel.Dock="Top" Orientation="Horizontal">  <Expander DockPanel.Dock="Top" ExpandDirection="Down" IsExpanded="False" Header="Login to Foursqare" VerticalAlignment="Center">  <UniformGrid Columns="1">  <DockPanel Width="600" >  <UniformGrid Columns="3" DockPanel.Dock="Left">  <Label Margin="10" Width="150">LinkedIn User Name</Label>  <Label Margin="10" Width="150">Password</Label>  <Label></Label>  <TextBox Margin="10" Width="150" Tag="enter user name" Style="{StaticResource commonTBtyle}" />  <PasswordBox Width="150" Margin="10"></PasswordBox>  <Button Width="100" ToolTip="login" Style="{StaticResource PlainBtnStyle}" >Login</Button>  <Label></Label>  <CheckBox Margin="10" Width="150">Stay logged in</CheckBox>  <Label></Label>  </UniformGrid>  </DockPanel>  </UniformGrid>  </Expander>  <Button Content="Refresh Foursqare" ToolTip="refresh" Style="{StaticResource PlainBtnStyle}"/>  </StackPanel>  <ListView></ListView>  </DockPanel>  </TabItem>  </TabControl>  </TabItem>  <TabItem Name="googleCalendarTabItem" Style="{StaticResource TItemStyle}" Header="Calendar">  <DockPanel LastChildFill="True">  <UniformGrid Rows="1" DockPanel.Dock="Bottom" Height="50">  <Button Style="{StaticResource PlainBtnStyle}" Name="gcalenderBtn" Click="gcalenderBtn\_Click">Load Calendar</Button>  <Label></Label>  <Label></Label>  <Label></Label>  <Label></Label>  </UniformGrid>  <WebBrowser Name="gCalenderWb"></WebBrowser>  </DockPanel>  </TabItem>  <TabItem Name="liveChatTabItem" Style="{StaticResource TItemStyle}" Header="Chat">  <DockPanel LastChildFill="True">  <Expander DockPanel.Dock="Top" ExpandDirection="Down" IsExpanded="False" Header="Login">  <UniformGrid Columns="1">  <DockPanel Width="600" >  <UniformGrid Columns="4" DockPanel.Dock="Left">  <Label Margin="10" Width="100">Chat Via</Label>  <Label Margin="10" Width="150">User Name</Label>  <Label Margin="10" Width="150">Password</Label>  <Label></Label>  <ComboBox Name="linkediBtn" Margin="10" Width="100" SelectedIndex="0">  <ComboBoxItem>Google+</ComboBoxItem>  <ComboBoxItem>Facebook</ComboBoxItem>  <ComboBoxItem>Yahoo</ComboBoxItem>  </ComboBox>  <TextBox Margin="10" Width="150" Tag="enter user name" Style="{StaticResource commonTBtyle}" />  <PasswordBox Width="150" Margin="10"></PasswordBox>  <Button Width="100" Style="{StaticResource PlainBtnStyle}">Login</Button>  <Label></Label>  <Label></Label>  <CheckBox Margin="10" Width="150">Stay logged in</CheckBox>  <Label></Label>  </UniformGrid>  </DockPanel>  </UniformGrid>  </Expander>  <ListView></ListView>  </DockPanel>  </TabItem>  <TabItem Name="googleMapTabItem" Style="{StaticResource TItemStyle}" Header="Map" >  <DockPanel LastChildFill="True">  <UniformGrid Rows="1" DockPanel.Dock="Bottom" Height="50">    <TextBox Name="loadMaplocationTB" Margin="10" Width="140" Tag="enter location" Style="{StaticResource commonTBtyle}" />  <Button Name="kolkataBtn" Style="{StaticResource PlainBtnStyle}" Click="kolkataBtn\_Click">Load Map</Button>  <Label></Label>  <Label></Label>  <Label></Label>  </UniformGrid>  <WebBrowser Name="mapWb"></WebBrowser>  </DockPanel>  </TabItem>  <TabItem Name="contactBookTabItem" Style="{StaticResource TItemStyle}" Header="Contacts" >  <DockPanel LastChildFill="True">  <StackPanel DockPanel.Dock="Top" Orientation="Horizontal">  <Expander Name="contacXpndr" DockPanel.Dock="Top" ExpandDirection="Down" IsExpanded="False" Header="Add Contact" VerticalAlignment="Center" >  <UniformGrid Columns="1">  <DockPanel Width="650" >  <UniformGrid Columns="4" DockPanel.Dock="Left" Width="647">  <Label Margin="10" Width="150">Name</Label>  <TextBox Name="contactnameTB" Margin="10" Width="140" Tag="enter name" Style="{StaticResource commonTBtyle}" />  <Label Margin="10" Width="150">Mobile Number</Label>  <TextBox Name="mobnoTB" Margin="10" Width="140" Tag="enter mobile number" Style="{StaticResource commonTBtyle}" />  <Label Margin="10" Width="150">EMail</Label>  <TextBox Name="emailTb" Margin="10" Width="140" Tag="enter email address" Style="{StaticResource commonTBtyle}" />  <Label Margin="10" Width="150">Home Number</Label>  <TextBox Name="homnoTB" Margin="10" Width="140" Tag="enter home number" Style="{StaticResource commonTBtyle}" />  <Label Margin="10" Width="150">Address</Label>  <TextBox Name="addressTB" Margin="10" Width="140" Tag="enter address" Style="{StaticResource commonTBtyle}" />  <Label Margin="10" Width="150">Office Number</Label>  <TextBox Name="ofcNoTb" Margin="10" Width="140" Tag="enter office number" Style="{StaticResource commonTBtyle}" />  <Label Margin="10" Width="150">Remarks</Label>  <TextBox Name="remrkTB" Margin="10" Width="140" Tag="enter remarks" Style="{StaticResource commonTBtyle}" />  <Label Margin="10" Width="150">Fax Number</Label>  <TextBox Name="faxNoTb" Margin="10" Width="140" Tag="enter fax number" Style="{StaticResource commonTBtyle}" />  <Label></Label>  <Label></Label>  <Button Content="Clear Fields" ToolTip="clear fields" Style="{StaticResource PlainBtnStyle}" />  <Button Content="Save" ToolTip="save info" Style="{StaticResource PlainBtnStyle}" Name="saveContactBtn" Click="saveContactBtn\_Click" />  </UniformGrid>  </DockPanel>  </UniformGrid>  </Expander>  </StackPanel>  <DockPanel DockPanel.Dock="Bottom">  <UniformGrid Rows="1">  <Label></Label>  <Label></Label>  <Label></Label>  <Button Name="contactrfrshBtn" Content="Refresh" ToolTip="Refresh details" Style="{StaticResource PlainBtnStyle}" Click="contactrfrshBtn\_Click" />  <Button Content="Edit Contacts" ToolTip="edit selected contact details" Style="{StaticResource PlainBtnStyle}" />  <Button Content="Delete Contact" ToolTip="delete selected contact details" Style="{StaticResource PlainBtnStyle}" />  </UniformGrid>  </DockPanel>  <ListView Name="contactView" HorizontalAlignment="Stretch" ItemsSource="{Binding contactCollection}">  <ListView.View>  <GridView>  <GridViewColumn Width="180" Header="Name" DisplayMemberBinding="{Binding name}" />  <GridViewColumn Width="150" Header="Mobile Number" DisplayMemberBinding="{Binding mobileno}" />  <GridViewColumn Width="130" Header="EMail" DisplayMemberBinding="{Binding email}" />  <GridViewColumn Width="150" Header="Home Number" DisplayMemberBinding="{Binding homeno}" />  <GridViewColumn Width="300" Header="Address" DisplayMemberBinding="{Binding address}" />  <GridViewColumn Width="150" Header="Office Number" DisplayMemberBinding="{Binding oficeno}" />  <GridViewColumn Width="150" Header="Fax Number" DisplayMemberBinding="{Binding faxno}" />  <GridViewColumn Width="100" Header="Remarks" DisplayMemberBinding="{Binding remark}" />  </GridView>  </ListView.View>  </ListView>  </DockPanel>  </TabItem>  <TabItem Name="passTabItem" Style="{StaticResource TItemStyle}" Header="Password">    <DockPanel LastChildFill="True">  <StackPanel DockPanel.Dock="Top" Orientation="Horizontal">  <Expander Name="loginpassPassExpndr" DockPanel.Dock="Top" ExpandDirection="Down" IsExpanded="True" Header="Login" VerticalAlignment="Center" >  <UniformGrid Columns="1">  <DockPanel Width="600" >  <UniformGrid Columns="3" DockPanel.Dock="Left">  <Label Margin="10" Width="150">User Name</Label>  <Label Margin="10" Width="150">Password</Label>  <Label></Label>  <TextBox Name="passwordMUserIdTB" Margin="10" Width="150" Tag="enter user name" Style="{StaticResource commonTBtyle}" />  <PasswordBox Name="passwordMPassBtnPB" Width="150" Margin="10"></PasswordBox>  <Button Name="passwordMLoginBtn" ToolTip="login" Width="100" Style="{StaticResource PlainBtnStyle}" Click="passwordMLoginBtn\_Click">Login</Button>  </UniformGrid>  </DockPanel>  </UniformGrid>  </Expander>  </StackPanel>  <DockPanel Name="mainPassDP" IsEnabled="False" LastChildFill="True">  <Expander Header="Login To Add Info" Name="addpassInfoExpndr" DockPanel.Dock="Top">  <UniformGrid Columns="4">  <Label Margin="10">Name</Label>  <TextBox Name="passNameTB" Margin="10" Tag="enter name" Style="{StaticResource commonTBtyle}" />  <Label Margin="10">EMail</Label>  <TextBox Name="emailforpassTB" Margin="10" Tag="enter email address" Style="{StaticResource commonTBtyle}" />  <Label Margin="10">User ID</Label>  <TextBox Name="userIdTB" Margin="10" Tag="enter user id" Style="{StaticResource commonTBtyle}" />  <Label Margin="10">Password</Label>  <TextBox Name="pawsrdTB" Margin="10" Tag="enter password" Style="{StaticResource commonTBtyle}" />  <Label Margin="10">Secret Question</Label>  <TextBox Name="scrtQstnTB" Margin="10" Tag="enter secret question" Style="{StaticResource commonTBtyle}" />  <Label Margin="10">Answer</Label>  <TextBox Name="secrtAnsTB" Margin="10" Tag="enter answer" Style="{StaticResource commonTBtyle}" />  <Label Margin="10">Others Info</Label>  <TextBox Name="othersInfoTB" Margin="10" Tag="enter others info" Style="{StaticResource commonTBtyle}" />  <Button Name="clearPassInfoFieldBtn" ToolTip="clear all fields" Width="150" Style="{StaticResource PlainBtnStyle}" Click="clearPassInfoFieldBtn\_Click">Clear Fields</Button>  <Button Name="paswrdSaveBtn" ToolTip="save info" Width="150" Style="{StaticResource PlainBtnStyle}" Click="paswrdSave\_Click">Save</Button>  </UniformGrid>  </Expander>  <UniformGrid DockPanel.Dock="Bottom" Columns="6" Name="logoutDP">  <Label></Label>  <Label></Label>  <Button Content="Refresh" Name="refrshpsdwBtn" ToolTip="refresh details" Style="{StaticResource PlainBtnStyle}" Click="refrshpsdwBtn\_Click" />  <Button Content="Edit Details" ToolTip="edit selected details" Style="{StaticResource PlainBtnStyle}" />  <Button Content="Delete Details" ToolTip="delete selected details" Style="{StaticResource PlainBtnStyle}" />  <Button Content="Please Log In First" Name="logoutPassBtn" IsEnabled="False" ToolTip="logout" Style="{StaticResource PlainBtnStyle}" Click="logoutPassBtn\_Click" />  </UniformGrid>  <UniformGrid DockPanel.Dock="Top">  <Expander Header="Login To Show All Details" Name="allDetailsExpndr">  <ListView Name="passView" HorizontalAlignment="Stretch" ItemsSource="{Binding passwordCollection}">  <ListView.View>  <GridView>  <GridViewColumn Width="100" Header="Name" DisplayMemberBinding="{Binding name}" />  <GridViewColumn Width="150" Header="Mail" DisplayMemberBinding="{Binding email}" />  <GridViewColumn Width="100" Header="ID" DisplayMemberBinding="{Binding userId}" />  <GridViewColumn Width="150" Header="Password" DisplayMemberBinding="{Binding password}" />  <GridViewColumn Width="300" Header="Secret Question" DisplayMemberBinding="{Binding scrtqstn}" />  <GridViewColumn Width="100" Header="Answer" DisplayMemberBinding="{Binding scrtans}"/>  <GridViewColumn Width="100" Header="Others Info" DisplayMemberBinding="{Binding otherInfo}" />  </GridView>  </ListView.View>  </ListView>  </Expander>  </UniformGrid>  </DockPanel>  </DockPanel>  </TabItem>  <TabItem Name="extraTabItem" Style="{StaticResource TItemStyle}" Header="Extra">    <TabControl Name="expnceControl" Margin="3" BorderBrush="#FFE9E0E0" TabStripPlacement="Left" HorizontalContentAlignment="Left" >  <TabItem Name="taskTabItem" Style="{StaticResource TItemStyle}" Header="Task" ToolTip="task">  <DockPanel LastChildFill="True">  <StackPanel DockPanel.Dock="Top" Orientation="Horizontal">  <Expander Name="tskxpndr" DockPanel.Dock="Top" ExpandDirection="Down" IsExpanded="False" Header="Add Task" VerticalAlignment="Center">  <UniformGrid Columns="1">  <DockPanel Width="650" Height="197">  <UniformGrid Columns="4" DockPanel.Dock="Left" Width="647" Height="198">  <Label Margin="10" Width="150" VerticalAlignment="Center">Task Value</Label>  <ComboBox Name="taskValueCB" Margin="10" Width="142" Height="30" ToolTip="task value" VerticalContentAlignment="Center" HorizontalContentAlignment="Center" SelectedIndex="0">  <ComboBoxItem>High</ComboBoxItem>  <ComboBoxItem>Medium</ComboBoxItem>  <ComboBoxItem>Low</ComboBoxItem>  </ComboBox>  <Label Margin="10" Width="150" VerticalAlignment="Center">About Task</Label>  <TextBox Name="taskDetailsTB" Height="100" TextWrapping="Wrap" AcceptsReturn="True" VerticalScrollBarVisibility="Visible" SpellCheck.IsEnabled="True">  <TextBox.Style>  <Style TargetType="TextBox" xmlns:sys="clr-namespace:System;assembly=mscorlib">  <Style.Resources>  <VisualBrush x:Key="CueBannerBrush" AlignmentX="Left" AlignmentY="Top" Stretch="None">  <VisualBrush.Visual>  <Label Content="what to do?" Foreground="LightGray" />  </VisualBrush.Visual>  </VisualBrush>  </Style.Resources>  <Style.Triggers>  <Trigger Property="Text" Value="{x:Static sys:String.Empty}">  <Setter Property="Background" Value="{StaticResource CueBannerBrush}" />  </Trigger>  <Trigger Property="Text" Value="{x:Null}">  <Setter Property="Background" Value="{StaticResource CueBannerBrush}" />  </Trigger>  <Trigger Property="IsKeyboardFocused" Value="True">  <Setter Property="Background" Value="White" />  </Trigger>  </Style.Triggers>  </Style>  </TextBox.Style>  </TextBox>  <Label></Label>  <Label></Label>  <Button Content="Clear" ToolTip="clear field" Style="{StaticResource PlainBtnStyle}" Height="25" Width="130" />  <Button Name="addTaskBtn" Content="Save" ToolTip="save task" Style="{StaticResource PlainBtnStyle}" Height="25" Width="130" Click="addTaskBtn\_Click" />  </UniformGrid>  </DockPanel>  </UniformGrid>  </Expander>  </StackPanel>  <DockPanel DockPanel.Dock="Bottom">  <UniformGrid Rows="1">  <Label></Label>  <Label></Label>  <Label></Label>  <Button Content="Refresh" Name="refrshTskBtn" ToolTip="refresh task" Style="{StaticResource PlainBtnStyle}" Click="contactrfrshBtn\_Click" />  <Button Content="Edit Task" ToolTip="edit selected task" Style="{StaticResource PlainBtnStyle}" />  <Button Content="Delete Task" ToolTip="delete selected task" Style="{StaticResource PlainBtnStyle}" />  </UniformGrid>  </DockPanel>  <ListView Name="taskView" HorizontalAlignment="Stretch" ItemsSource="{Binding taskCollection}">  <ListView.View>  <GridView>  <GridViewColumn Width="100" Header="Value" DisplayMemberBinding="{Binding value}"/>  <GridViewColumn Width="650" Header="Task" DisplayMemberBinding="{Binding taskDetails}" />  </GridView>  </ListView.View>  </ListView>  </DockPanel>  </TabItem>  <TabItem Name="expenseTabItem" Style="{StaticResource TItemStyle}" Header="Expense" ToolTip="expense">  </TabItem>  </TabControl>  </TabItem>  </TabControl>  </DockPanel>  </DockPanel>  </Border>  </Window> |

|  |
| --- |
| ShowEvent.xaml |
| <UserControl x:Class="DNBSNGUI.ShowEvent"  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  mc:Ignorable="d"  d:DesignHeight="300" d:DesignWidth="300">  <DockPanel LastChildFill="True">  <UniformGrid DockPanel.Dock="Top">  <Label Name="sequenceLabel"></Label>  </UniformGrid>  <UniformGrid DockPanel.Dock="Top">  <ListView Background="Gray"></ListView>  </UniformGrid>  </DockPanel>  </UserControl> |

### GUI Style : DNBSNStyles

|  |
| --- |
| ControlStyle.xaml |
| <ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">    <Style x:Key="PlainBtnStyle" TargetType="Button">  <Setter Property="Background" Value="#D8D8D8" />  <Setter Property="Foreground" Value="Black" />  <Setter Property="FontWeight" Value="Bold" />  <Setter Property="BorderBrush" Value="#FFC4C458" />  <Setter Property="Margin" Value="10"/>  <Setter Property="OpacityMask" Value="White" />  <Setter Property="Padding" Value="7" />  <Setter Property="Template">  <Setter.Value>  <ControlTemplate TargetType="Button">  <Border Name="border"  BorderThickness="1"  Padding="4,2"  BorderBrush="White"  CornerRadius="0"  Background="{TemplateBinding Background}">  <ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center" />  </Border>  <ControlTemplate.Triggers>  <Trigger Property="IsMouseOver" Value="True">  <Setter TargetName="border" Property="BorderBrush" Value="#2E9AFE" />  <Setter Property="Button.Background" Value="#2E9AFE" />  <Setter Property="Button.Foreground" Value="White" />  </Trigger>  <Trigger Property="IsPressed" Value="True">  <Setter TargetName="border" Property="BorderBrush" Value="White" />  <Setter Property="Button.Background" Value="#D8D8D8" />  </Trigger>    </ControlTemplate.Triggers>    </ControlTemplate>  </Setter.Value>  </Setter>  </Style>  <Style x:Key="TItemStyle" TargetType="TabItem">  <Setter Property="Width" Value="115" />  <Setter Property="Height" Value="33" />  <Setter Property="Background" Value="#D8D8D8" />  <Setter Property="BorderBrush" Value="white" />  <Setter Property="Template">  <Setter.Value>  <ControlTemplate TargetType="{x:Type TabItem}">  <Grid>  <Border Name="Border" Margin="0,0,0,0" Background="#D8D8D8"  BorderBrush="#D8D8D8" BorderThickness="1,1,1,1" CornerRadius="0">  <ContentPresenter x:Name="ContentSite" VerticalAlignment="Center"  HorizontalAlignment="Center"  ContentSource="Header" Margin="12,2,12,2"  RecognizesAccessKey="True">  <ContentPresenter.LayoutTransform>  <RotateTransform Angle="0" />  </ContentPresenter.LayoutTransform>  </ContentPresenter>  </Border>  </Grid>  <ControlTemplate.Triggers>  <Trigger Property="IsSelected" Value="True">  <Setter TargetName="Border" Property="Background" Value="#2E9AFE" />  <Setter Property="Button.Foreground" Value="White" />  </Trigger>  <Trigger Property="IsMouseOver" Value="True">  <Setter TargetName="Border" Property="BorderBrush" Value="#2E9AFE" />  <Setter TargetName="Border" Property="Background" Value="#2E9AFE" />  <Setter Property="Button.Foreground" Value="White" />  </Trigger>  </ControlTemplate.Triggers>  </ControlTemplate>  </Setter.Value>  </Setter>  <Setter Property="HeaderTemplate">  <Setter.Value>  <DataTemplate>  <Border x:Name="grid" >  <ContentPresenter>  <ContentPresenter.Content>  <TextBlock Margin="4" FontSize="15" Text="{TemplateBinding Content}"/>  </ContentPresenter.Content>  </ContentPresenter>  </Border>  </DataTemplate>  </Setter.Value>  </Setter>  </Style>  <Style x:Key="CheckBtyle" TargetType="CheckBox">  <Setter Property="Width" Value="80" />  <Setter Property="Background" Value="#FFF9F9EC" />  <Setter Property="BorderBrush" Value="#FFC4C458" />  <Setter Property="Foreground" Value="#000000" />  </Style>  <Style x:Key="pBstyle" TargetType="PasswordBox">  <Setter Property="Control.Template" >  <Setter.Value>  <ControlTemplate TargetType="{x:Type PasswordBox}">  <Border x:Name="border" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" SnapsToDevicePixels="True">  <Grid>  <ScrollViewer x:Name="PART\_ContentHost" Focusable="False" HorizontalScrollBarVisibility="Hidden" VerticalScrollBarVisibility="Hidden"/>  <TextBlock x:Name="InternalWatermarkLabel"  Text="{TemplateBinding Tag}"  Visibility="Collapsed" Focusable="False"  VerticalAlignment="Top" Margin=" 5 1 0 0"  Foreground="Silver"  Background="Transparent"/>  </Grid>  </Border>  <ControlTemplate.Triggers>  <MultiTrigger>  <MultiTrigger.Conditions>  <Condition Property="IsFocused" Value="False" />  </MultiTrigger.Conditions>  <MultiTrigger.Setters>  <Setter Property="Visibility" TargetName="InternalWatermarkLabel"  Value="Visible" />  </MultiTrigger.Setters>  </MultiTrigger>  <Trigger Property="IsEnabled" Value="False">  <Setter Property="Opacity" TargetName="border" Value="0.56"/>  </Trigger>  <Trigger Property="IsMouseOver" Value="True">  <Setter Property="BorderBrush" TargetName="border" Value="#FF7EB4EA"/>  </Trigger>  <Trigger Property="IsKeyboardFocused" Value="True">  <Setter Property="BorderBrush" TargetName="border" Value="#FF569DE5"/>  </Trigger>  </ControlTemplate.Triggers>  </ControlTemplate>  </Setter.Value>  </Setter>  </Style>  <Style x:Key="TitlebarBTN" TargetType="Button">  <Setter Property="OverridesDefaultStyle" Value="True"/>  <Setter Property="Margin" Value="0"/>  <Setter Property="FontWeight" Value="Bold" />  <Setter Property="Template">  <Setter.Value>  <ControlTemplate TargetType="Button">  <Border Name="border"  BorderThickness="1"  Padding="4,2"  BorderBrush="White"  CornerRadius="0"  Background="{TemplateBinding Background}">  <ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center" />  </Border>  <ControlTemplate.Triggers>  <Trigger Property="IsMouseOver" Value="True">  <Setter TargetName="border" Property="BorderBrush" Value="White" />  <Setter Property="Button.Background" Value="#D8D8D8" />  </Trigger>  <Trigger Property="IsPressed" Value="True">  <Setter TargetName="border" Property="BorderBrush" Value="White" />  <Setter Property="Button.Background" Value="#2E9AFE" />  </Trigger>  </ControlTemplate.Triggers>  </ControlTemplate>  </Setter.Value>  </Setter>  </Style>  <Style x:Key="{x:Type ToolTip}" TargetType="{x:Type ToolTip}">  <Setter Property="Background" Value="Black"/>  <Setter Property="BorderBrush" Value="#D8D8D8"/>  <Setter Property="Foreground" Value="White" />  <Setter Property="Padding" Value="7" />  <Setter Property="FontWeight" Value="Bold" />    </Style>  <Style x:Key="commonTBtyle" TargetType="TextBox">    <Setter Property="Control.Template" >  <Setter.Value>  <ControlTemplate TargetType="{x:Type TextBox}">  <Border x:Name="border" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" SnapsToDevicePixels="True">  <Grid>  <ScrollViewer x:Name="PART\_ContentHost" Focusable="False" HorizontalScrollBarVisibility="Hidden" VerticalScrollBarVisibility="Hidden"/>  <TextBlock x:Name="InternalWatermarkLabel"  Text="{TemplateBinding Tag}"  Visibility="Collapsed" Focusable="False"  VerticalAlignment="Top" Margin=" 5 1 0 0"  Foreground="Silver"  Background="Transparent"/>  </Grid>  </Border>  <ControlTemplate.Triggers>  <MultiTrigger>  <MultiTrigger.Conditions>  <Condition Property="IsFocused" Value="False" />  <Condition Property="Text" Value="" />  </MultiTrigger.Conditions>  <MultiTrigger.Setters>  <Setter Property="Visibility" TargetName="InternalWatermarkLabel"  Value="Visible" />  </MultiTrigger.Setters>  </MultiTrigger>  <Trigger Property="IsEnabled" Value="False">  <Setter Property="Opacity" TargetName="border" Value="0.56"/>  </Trigger>  <Trigger Property="IsMouseOver" Value="True">  <Setter Property="BorderBrush" TargetName="border" Value="#FF7EB4EA"/>    </Trigger>  <Trigger Property="IsKeyboardFocused" Value="True">  <Setter Property="BorderBrush" TargetName="border" Value="#FF569DE5"/>  </Trigger>  </ControlTemplate.Triggers>  </ControlTemplate>  </Setter.Value>  </Setter>  </Style>  <Style TargetType="{x:Type ScrollBar}">  <Setter Property="Stylus.IsFlicksEnabled" Value="True" />  <Setter Property="Background" Value="#D8D8D8" />  <Setter Property="Width" Value="15"/>  <Setter Property="MinWidth" Value="8" />  <Setter Property="Template">    <Setter.Value>  <ControlTemplate TargetType="{x:Type ScrollBar}">  <Grid x:Name="GridRoot" Width="19" Background="{TemplateBinding Background}">  <Grid.RowDefinitions>  <RowDefinition Height="0.00001\*" />  </Grid.RowDefinitions>  <Track x:Name="PART\_Track" Grid.Row="0" IsDirectionReversed="true" Focusable="false">  <Track.Thumb>  <Thumb x:Name="Thumb" Background="Gray" Style="{DynamicResource ScrollThumbs}" />  </Track.Thumb>  <Track.IncreaseRepeatButton>  <RepeatButton x:Name="PageUp" Command="ScrollBar.PageDownCommand" Opacity="0" Focusable="false" />  </Track.IncreaseRepeatButton>  <Track.DecreaseRepeatButton>  <RepeatButton x:Name="PageDown" Command="ScrollBar.PageUpCommand" Opacity="0" Focusable="false" />  </Track.DecreaseRepeatButton>  </Track>  </Grid>  <ControlTemplate.Triggers>  <Trigger SourceName="Thumb" Property="IsMouseOver" Value="true">  <Setter Value="#2E9AFE" TargetName="Thumb" Property="Background" />  </Trigger>  <Trigger SourceName="Thumb" Property="IsDragging" Value="true">  <Setter Value="#D8D8D8" TargetName="Thumb" Property="Background" />  </Trigger>  <Trigger Property="IsEnabled" Value="false">  <Setter TargetName="Thumb" Property="Visibility" Value="Collapsed" />  </Trigger>  <Trigger Property="Orientation" Value="Horizontal">  <Setter TargetName="GridRoot" Property="LayoutTransform">  <Setter.Value>  <RotateTransform Angle="-90" />  </Setter.Value>  </Setter>  <Setter TargetName="PART\_Track" Property="LayoutTransform">  <Setter.Value>  <RotateTransform Angle="-90" />  </Setter.Value>  </Setter>  <Setter Property="Width" Value="Auto" />  <Setter Property="Height" Value="15" />  <Setter TargetName="Thumb" Property="Tag" Value="Horizontal" />  <Setter TargetName="PageDown" Property="Command" Value="ScrollBar.PageLeftCommand" />  <Setter TargetName="PageUp" Property="Command" Value="ScrollBar.PageRightCommand" />  </Trigger>  </ControlTemplate.Triggers>  </ControlTemplate>  </Setter.Value>  </Setter>  </Style>  <Style x:Key="LblStyle" TargetType="Label">  <Setter Property="VerticalContentAlignment" Value="Center" />  <Setter Property="HorizontalContentAlignment" Value="Left" />  <Setter Property="Foreground" Value="Black" />  <Setter Property="FontWeight" Value="Bold" />    </Style>  <Style x:Key="TxtblkStyle" TargetType="TextBlock">  <Setter Property="VerticalAlignment" Value="Center" />  <Setter Property="HorizontalAlignment" Value="Center" />  <Setter Property="Foreground" Value="Black" />  <Setter Property="FontWeight" Value="Bold" />  </Style>  <Style x:Key="DatepkrStyle" TargetType="DatePicker">  <Setter Property="VerticalAlignment" Value="Center" />  <Setter Property="HorizontalAlignment" Value="Left" />  <Setter Property="Foreground" Value="Black" />  <Setter Property="FontWeight" Value="Bold" />  </Style>  </ResourceDictionary> |

### C# coding

|  |
| --- |
| MainWindow.xaml.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Windows;  using System.Windows.Controls;  using System.Windows.Data;  using System.Windows.Documents;  using System.Windows.Input;  using System.Windows.Media;  using System.Windows.Media.Imaging;  using System.Windows.Navigation;  using System.Windows.Shapes;  using DNBSNData;  using DNBSNDb;  using System.Collections.ObjectModel;  using TweetSharp;  using System.Diagnostics;  namespace DNBSNGUI  {  /// <summary>  /// Interaction logic for MainWindow.xaml  /// </summary>  public partial class MainWindow : Window  {  public MainWindow()  {  InitializeComponent();  this.MouseLeftButtonDown += delegate { this.DragMove(); };  FocusManager.SetFocusedElement(this, dNBSNUserIDTB);  }  private void Window\_Loaded(object sender, RoutedEventArgs e)  {  fetchTaskData();  fetchContactData();  fetchPasswordData();  fetchNoteData();  for (int i = 1; i <= 30; i++)  dateUG.Children.Add(new ShowEvent(i));  }  private void kolkataBtn\_Click(object sender, RoutedEventArgs e)  {  mapWb.NavigateToString("<html><body><iframe src=\"https://maps.google.co.in/?ll=20.98352,82.752628&spn=43.617141,86.572266&t=h&z=4\" width=\"1100\" height=\"480\" frameborder=\"0\" marginheight=\"0\" marginwidth=\"0\">Loading...</iframe></body></html>");  }  private void gcalenderBtn\_Click(object sender, RoutedEventArgs e)  {  gCalenderWb.NavigateToString("<html><body><iframe src=\"https://www.facebook.com/connect/login\_success.html#access\_token=THE\_TOKEN&expires\_in=7180\" width=\"1100\" height=\"480\" frameborder=\"0\" marginheight=\"0\" marginwidth=\"0\">Loading...</iframe></body></html>");  // fb like url https://www.facebook.com/plugins/like.php?href=https://www.facebook.com/technicise\  }  private void Image\_ImageFailed(object sender, ExceptionRoutedEventArgs e)  {  }  private void closeBtn\_Click(object sender, RoutedEventArgs e)  {  this.Close();  }  private void virtualKeyboardDateBtn\_Click(object sender, RoutedEventArgs e)  {  System.Diagnostics.Process.Start("calc");  }  private void minimizeBtn\_Click(object sender, RoutedEventArgs e)  {  this.WindowState = WindowState.Minimized;  }  //private void maximizeBtn\_Click(object sender, RoutedEventArgs e)  //{  // this.WindowState = WindowState.Maximized;  //}  private void passwordMLoginBtn\_Click(object sender, RoutedEventArgs e)  {  if ((passwordMUserIdTB.Text.Equals("1")) && (passwordMPassBtnPB.Password.Equals("1")))  {  mainPassDP.IsEnabled = true;  passwordMPassBtnPB.Password = string.Empty;  loginpassPassExpndr.IsExpanded = false;  loginpassPassExpndr.IsEnabled = false;  loginpassPassExpndr.Header = "Logged In";  loginpassPassExpndr.ToolTip = "Logged In";  allDetailsExpndr.IsExpanded = true;  logoutPassBtn.Content = "Logout";  allDetailsExpndr.Header = "All Details";  addpassInfoExpndr.Header = "Add Info";  logoutPassBtn.IsEnabled = true;  }  else  {  //MessageBox.Show("Please Enter correct Password");  //DNBSNGUI.ErrorMessage ErrorMessageObj = new DNBSNGUI.ErrorMessage();  //ErrorMessageObj.ShowDialog();  passwordMPassBtnPB.Password = string.Empty;  }  }  private void logoutDnbsnBtn\_Click(object sender, RoutedEventArgs e)  {  dNBSNUserIDTB.Text = string.Empty;  dNBSNpassPB.Password = string.Empty;  mainLeftExpndr.IsExpanded = false;  mainLeftExpndr.IsEnabled = false;  mainTabControl.IsEnabled = false;  logoutDnbsnBtn.IsEnabled = false;  logoutDnbsnBtn.Content = "";  loginExpndr.IsExpanded = true;  loginExpndr.IsEnabled = true;  hntLginLbl.Content = "Login first to use Daily Note Book with Social Networking Updater";  loginExpndr.Visibility = Visibility.Visible;  loginExpndr.IsExpanded = true;  dNBSNUserIDTB.Clear();  dNBSNpassPB.Clear();  }  private void dNBSNLoginBtn\_Click(object sender, RoutedEventArgs e)  {  if (dNBSNUserIDTB.Text.Equals(DNBSNDb.DbInteraction.FetcheId()) && dNBSNpassPB.Password.Equals(DNBSNDb.DbInteraction.FetchePassword()))  {  mainLeftExpndr.IsEnabled = true;  mainLeftExpndr.IsExpanded = true;  dNBSNpassPB.Password = string.Empty;  mainTabControl.IsEnabled = true;  loginExpndr.IsEnabled = false;  //loginExpndr.Header = "Logged In";  //loginExpndr.IsExpanded = false;  loginExpndr.Visibility = Visibility.Collapsed;  logoutDnbsnBtn.IsEnabled = true;  logoutDnbsnBtn.Content = "LOG OUT";  logoutDnbsnBtn.ToolTip = "log out";  hntLginLbl.Content = "";  }  else  {  //MessageBox.Show("Please Enter correct Password");  //DNBSNGUI.ErrorMessage ErrorMessageObj = new DNBSNGUI.ErrorMessage();  //ErrorMessageObj.ShowDialog();  errorMsgLbl.Content = "Wrong User ID or Password";  dNBSNpassPB.Password = string.Empty;  }  }  private void logoutPassBtn\_Click(object sender, RoutedEventArgs e)  {  mainPassDP.IsEnabled = false;  passwordMPassBtnPB.Password = string.Empty;  passwordMUserIdTB.Text = string.Empty;  loginpassPassExpndr.IsExpanded = true;  loginpassPassExpndr.IsEnabled = true;  //loginpassPassExpndr.Header = "Logged In";  //loginpassPassExpndr.ToolTip = "Logged In";  allDetailsExpndr.IsExpanded = false;  logoutPassBtn.Content = "Please Log In First";  addpassInfoExpndr.Header = "Login To Add Info";  allDetailsExpndr.Header = "Login To Show All Details";  //allDetailsExpndr.Header = "All Details";  //addInfoExpndr.Header = "Add Info";  logoutPassBtn.IsEnabled = false;  //mainLeftExpndr.IsEnabled = false;  //mainLeftExpndr.IsExpanded = false;  //dNBSNpassPB.Password = string.Empty;  //mainTabControl.IsEnabled = false;  //loginExpndr.IsEnabled = true;  //loginExpndr.Header = "Logged In";  //loginExpndr.IsExpanded = true;  }  private void createAcHp\_Click(object sender, RoutedEventArgs e)  {  newAcExpndr.Visibility = Visibility.Visible;  newAcExpndr.IsExpanded = true;  loginExpndr.Visibility = Visibility.Collapsed;  dNBSNNewUserIDTB.Clear();  dNBSNNewPassPB.Clear();  dNBSNNewRepassPB.Clear();  dNBSNHintsTB.Clear();  createErrorMsgLbl.Content = string.Empty;  }  private void goLogin\_Click(object sender, RoutedEventArgs e)  {  newAcExpndr.Visibility = Visibility.Collapsed;  loginExpndr.Visibility = Visibility.Visible;  loginExpndr.IsExpanded = true;  dNBSNUserIDTB.Clear();  dNBSNpassPB.Clear();  errorMsgLbl.Content = string.Empty;  hintsMsgLbl.Content = string.Empty;  }  TwitterService service;  OAuthRequestToken requestToken;  private void launchTwitterAppButton\_Click(object sender, RoutedEventArgs e)  {  // Pass your credentials to the service  service = new TwitterService("rfeDMlcxMQSqDSdmXDT3A", "7ZUD4YtVqOfFJcCHwWz7q0qTFjcKyVGSNyN4ElCsZZE");  // Step 1 - Retrieve an OAuth Request Token  requestToken = service.GetRequestToken();  // Step 2 - Redirect to the OAuth Authorization URL  Uri uri = service.GetAuthorizationUri(requestToken);  Process.Start(uri.ToString());  }  private void verifyTwitterApp\_Click(object sender, RoutedEventArgs e)  {  // Step 3 - Exchange the Request Token for an Access Token  OAuthAccessToken access = service.GetAccessToken(requestToken, verificationCodeTxtBox.Text);  // Step 4 - User authenticates using the Access Token  service.AuthenticateWith(access.Token, access.TokenSecret);  var tweets = service.ListTweetsOnHomeTimeline(new ListTweetsOnHomeTimelineOptions());  foreach (var tweet in tweets)  {  tweetsTextBlock.Text += tweet.User.ScreenName + " says " + tweet.Text + "\n";  }  }  private void dNBSSubmitBtn\_Click(object sender, RoutedEventArgs e)  {  if (!dNBSNNewUserIDTB.Text.Equals(string.Empty) &&  !dNBSNNewPassPB.Password.Equals(string.Empty) &&  !dNBSNNewRepassPB.Password.Equals(string.Empty) &&  !dNBSNHintsTB.Text.Equals(string.Empty))  {  if (!dNBSNNewPassPB.Password.Equals(dNBSNNewRepassPB.Password))  {  dNBSNNewPassPB.Clear();  dNBSNNewRepassPB.Clear();  createErrorMsgLbl.Content = "Please Enter same Password";  }  else  {  DNBSNData.UserInfo newUser = new DNBSNData.UserInfo();  newUser.id = GenerateId();  newUser.userId = dNBSNNewUserIDTB.Text;  newUser.pass = dNBSNNewPassPB.Password;  newUser.hints = dNBSNHintsTB.Text;  DNBSNDb.DbInteraction.DoRegisterNewUser(newUser);  newAcExpndr.Visibility = Visibility.Collapsed;  loginExpndr.Visibility = Visibility.Visible;  loginExpndr.IsExpanded = true;  dNBSNUserIDTB.Clear();  dNBSNpassPB.Clear();  }  }  else  {  createErrorMsgLbl.Content = "Correctly Enter Info ";  }  }  private void previousDateBtn\_Click(object sender, RoutedEventArgs e)  {  }  private string GenerateId()  {  return DateTime.Now.ToOADate().ToString();  }  #region Note  private void saveNotesBtn\_Click(object sender, RoutedEventArgs e)  {  if (!notesTB.Text.Equals(string.Empty) && !notesTB.Text.Equals("Start writing from here..."))  {  DNBSNData.NoteInfo newNote = new DNBSNData.NoteInfo();  newNote.id = GenerateId();  newNote.gotoDate = noteDateDP.SelectedDate.Value;  newNote.note = notesTB.Text;  DNBSNDb.DbInteraction.DoEnterNewNote(newNote);  notesTB.Clear();  noteSuccessMsgLvl.Content = "Event Submited";  fetchNoteData();  }  else  {  noteSuccessMsgLvl.Content = "Please Enter Proper Note";  }  }  private void clearFieldsBtn\_Click(object sender, RoutedEventArgs e)  {  noteSuccessMsgLvl.Content = "";  notesTB.Clear();  noteSuccessMsgLvl.Content = "All Clear";  }  ObservableCollection<NoteInfo> \_allnoteCollection = new ObservableCollection<NoteInfo>();  public ObservableCollection<NoteInfo> allnoteCollection  {  get  {  return \_allnoteCollection;  }  }  private void fetchNoteData()  {  List<NoteInfo> notes = DbInteraction.GetAllNoteList();  \_allnoteCollection.Clear();  foreach (NoteInfo note in notes)  {  \_allnoteCollection.Add(note);  }  }  #endregion  #region Password  private void paswrdSave\_Click(object sender, RoutedEventArgs e)  {  DNBSNData.PasswordInfo newPassword = new DNBSNData.PasswordInfo();  newPassword.id = GenerateId();  newPassword.name = passNameTB.Text;  newPassword.email = emailforpassTB.Text;  newPassword.userId = userIdTB.Text;  newPassword.password = pawsrdTB.Text;  newPassword.scrtqstn = scrtQstnTB.Text;  newPassword.scrtans = secrtAnsTB.Text;  newPassword.otherInfo = othersInfoTB.Text;  DNBSNDb.DbInteraction.DoRegisterNewPassword(newPassword);  passNameTB.Text = emailforpassTB.Text = userIdTB.Text = pawsrdTB.Text = scrtQstnTB.Text = secrtAnsTB.Text = othersInfoTB.Text = "";  addpassInfoExpndr.IsExpanded = false;  fetchPasswordData();  }  private void clearPassInfoFieldBtn\_Click(object sender, RoutedEventArgs e)  {  passNameTB.Text = emailforpassTB.Text = userIdTB.Text = pawsrdTB.Text = scrtQstnTB.Text = secrtAnsTB.Text = othersInfoTB.Text = "";  }  ObservableCollection<PasswordInfo> \_passwordCollection = new ObservableCollection<PasswordInfo>();  public ObservableCollection<PasswordInfo> passwordCollection  {  get  {  return \_passwordCollection;  }  }  private void fetchPasswordData()  {  List<PasswordInfo> passwords = DbInteraction.GetAllPasswordsList();  \_passwordCollection.Clear();  foreach (PasswordInfo password in passwords)  {  \_passwordCollection.Add(password);  }  }  private void refrshpsdwBtn\_Click(object sender, RoutedEventArgs e)  {  fetchPasswordData();  }  #endregion  #region Task  private void addTaskBtn\_Click(object sender, RoutedEventArgs e)  {  DNBSNData.TaskInfo newTask = new DNBSNData.TaskInfo();  newTask.id = GenerateId();  newTask.value = taskValueCB.Text;  newTask.taskDetails = taskDetailsTB.Text;  DNBSNDb.DbInteraction.DoRegisterNewTask(newTask);  taskValueCB.Text = taskDetailsTB.Text = "";  fetchTaskData();  tskxpndr.IsExpanded = false;  }  ObservableCollection<TaskInfo> \_taskCollection = new ObservableCollection<TaskInfo>();  public ObservableCollection<TaskInfo> taskCollection  {  get  {  return \_taskCollection;  }  }  private void fetchTaskData()  {  List<TaskInfo> tasks = DbInteraction.GetAllTaskList();  \_taskCollection.Clear();  foreach (TaskInfo task in tasks)  {  \_taskCollection.Add(task);  }  }  private void refrshTskBtn\_Click(object sender, RoutedEventArgs e)  {  fetchContactData();  }  #endregion  #region Contact  private void saveContactBtn\_Click(object sender, RoutedEventArgs e)  {  DNBSNData.ContactInfo newContact = new DNBSNData.ContactInfo();  newContact.id = GenerateId();  newContact.name = contactnameTB.Text;  newContact.mobileno = mobnoTB.Text;  newContact.homeno = homnoTB.Text;  newContact.oficeno = ofcNoTb.Text;  newContact.faxno = faxNoTb.Text;  newContact.address = addressTB.Text;  newContact.remark = remrkTB.Text;  newContact.email = emailTb.Text;  DNBSNDb.DbInteraction.DoRegisterNewContact(newContact);  contactnameTB.Text = mobnoTB.Text = homnoTB.Text = ofcNoTb.Text = faxNoTb.Text = addressTB.Text = remrkTB.Text = emailTb.Text = "";  fetchContactData();  contacXpndr.IsExpanded = true;  }  ObservableCollection<ContactInfo> \_contactCollection = new ObservableCollection<ContactInfo>();  public ObservableCollection<ContactInfo> contactCollection  {  get  {  return \_contactCollection;  }  }  private void fetchContactData()  {  List<ContactInfo> contacts = DbInteraction.GetAllContactList();  \_contactCollection.Clear();  foreach (ContactInfo Contacts in contacts)  {  \_contactCollection.Add(Contacts);  }  }  private void contactrfrshBtn\_Click(object sender, RoutedEventArgs e)  {  fetchContactData();  }  #endregion  #region Go to Date  private void goToDateBtn\_Click(object sender, RoutedEventArgs e)  {  noteControlsTab.SelectedIndex = 1;  NoteInfo noteinfo = new NoteInfo();  noteinfo.gotoDate = goToDateDP.SelectedDate.Value;  List<NoteInfo> notes = DbInteraction.searchNoteList(noteinfo);  \_allnoteCollection.Clear();  foreach (NoteInfo note in notes)  {  \_allnoteCollection.Add(note);  }  }  #endregion  private void monthlyViewTC\_click(object sender, RoutedEventArgs e)  {  List<NoteInfo> months = DbInteraction.searchMnthlyNoteList( goToDateDP.SelectedDate.Value);  \_allnoteCollection.Clear();  foreach (NoteInfo note in months)  {  \_allnoteCollection.Add(note);  }  }  }  } |

|  |
| --- |
| ShowEvent.xaml.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Windows;  using System.Windows.Controls;  using System.Windows.Data;  using System.Windows.Documents;  using System.Windows.Input;  using System.Windows.Media;  using System.Windows.Media.Imaging;  using System.Windows.Navigation;  using System.Windows.Shapes;  namespace DNBSNGUI  {  /// <summary>  /// Interaction logic for ShowEvent.xaml  /// </summary>  public partial class ShowEvent : UserControl  {  public ShowEvent(int no)  {  InitializeComponent();  sequenceLabel.Content = no;  }  }  } |

### Datbase Classes : DNBSNData

|  |
| --- |
| ContactInfo.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace DNBSNData  {  public class ContactInfo  {  public string id { get; set; }  public string name { get; set; }    public string mobileno { get; set; }  public string homeno { get; set; }  public string oficeno { get; set; }  public string faxno { get; set; }  public string address { get; set; }  public string remark { get; set; }  public string email { get; set; }  }  } |

|  |
| --- |
| NoteInfo.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace DNBSNData  {  public class NoteInfo  {  public string id { get; set; }  public DateTime noteDate { get; set; }  public string note { get; set; }  public DateTime gotoDate { get; set; }    }  } |

|  |
| --- |
| PasswordInfo.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace DNBSNData  {  public class PasswordInfo  {  public string id { get; set; }  public string name { get; set; }  public string email { get; set; }  public string userId { get; set; }  public string password { get; set; }  public string scrtqstn { get; set; }  public string scrtans { get; set; }  public string otherInfo { get; set; }  }  } |

|  |
| --- |
| TaskInfo.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace DNBSNData  {  public class TaskInfo  {  public string id { get; set; }  public string value { get; set; }  public string taskDetails { get; set; }  }  } |

|  |
| --- |
| UserInfo.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace DNBSNData  {  public class UserInfo  {  public string id { get; set; }  public string userId { get; set; }  public string pass { get; set; }  public string hints { get; set; }  }  } |

### Database connector: DNBSNDb

|  |
| --- |
| DbInteraction.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using DNBSNData;  namespace DNBSNDb  {  public class DbInteraction  {  static string passwordCurrent = "technicise";  static string dbmsCurrent = "dnbsndb";  private static MySql.Data.MySqlClient.MySqlConnection OpenDbConnection()  {  MySql.Data.MySqlClient.MySqlConnection msqlConnection = null;  msqlConnection = new MySql.Data.MySqlClient.MySqlConnection("server=localhost;user id=root;Password=" + passwordCurrent + ";database=" + dbmsCurrent + ";persist security info=False");  //open the connection  if (msqlConnection.State != System.Data.ConnectionState.Open)  msqlConnection.Open();  return msqlConnection;  }  #region User  public static int DoRegisterNewUser(UserInfo NewUser)  {  return DoRegisterNewuserindb(NewUser);  }  private static int DoRegisterNewuserindb(UserInfo NewUser)  {  int returnVal = 0;  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  {  //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  //define the connection used by the command object  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "INSERT INTO user(id,userid,passwrd,hints) " + "VALUES(@id,@userid,@passwrd,@hints)";  msqlCommand.Parameters.AddWithValue("@id", NewUser.id);  msqlCommand.Parameters.AddWithValue("@userid", NewUser.userId);  msqlCommand.Parameters.AddWithValue("@passwrd", NewUser.pass);  msqlCommand.Parameters.AddWithValue("@hints", NewUser.hints);  msqlCommand.ExecuteNonQuery();  returnVal = 1;  }  catch (Exception er)  {  returnVal = 0;  }  finally  {  //always close the connection  msqlConnection.Close();  }  return returnVal;  }  #endregion  #region Note  public static int DoEnterNewNote(NoteInfo NewNote)  {  return DoEnterNewNoteindb(NewNote);  }  private static int DoEnterNewNoteindb(NoteInfo NewNote)  {  int returnVal = 0;  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  {  //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  //define the connection used by the command object  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "INSERT INTO note(id,date,note) " + "VALUES(@id,@date,@note)";  msqlCommand.Parameters.AddWithValue("@id", NewNote.id);  msqlCommand.Parameters.AddWithValue("@date", NewNote.gotoDate);  msqlCommand.Parameters.AddWithValue("@note", NewNote.note);    msqlCommand.ExecuteNonQuery();  returnVal = 1;  }  catch (Exception er)  {  returnVal = 0;  }  finally  {  //always close the connection  msqlConnection.Close();  }  return returnVal;  }  public static List<NoteInfo> GetAllNoteList()  {  return QueryAllNoteList();  }  private static List<NoteInfo> QueryAllNoteList()  {  List<NoteInfo> NoteList = new List<NoteInfo>();  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  { //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "Select \* From note ;";  MySql.Data.MySqlClient.MySqlDataReader msqlReader = msqlCommand.ExecuteReader();  while (msqlReader.Read())  {  NoteInfo Note = new NoteInfo();  //Note.id = msqlReader.GetString("id");  Note.noteDate = msqlReader.GetDateTime("date");  Note.note = msqlReader.GetString("note");  NoteList.Add(Note);  }  }  catch (Exception er)  {  }  finally  {  //always close the connection  msqlConnection.Close();  }  return NoteList;  }    #endregion  #region ID password  public static string FetcheId()  {  string idStr = string.Empty;  int returnVal = 0;  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  {  //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  //define the connection used by the command object  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "Select userid from user;";  MySql.Data.MySqlClient.MySqlDataReader msqlReader = msqlCommand.ExecuteReader();  msqlReader.Read();  idStr = msqlReader.GetString("userid");  }  catch (Exception er)  {  //Assert//.Show(er.Message);  }  finally  {  //always close the connection  msqlConnection.Close();  }  return idStr;  }  public static string FetchePassword()  {  string passwordStr = string.Empty;  int returnVal = 0;  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  {  //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  //define the connection used by the command object  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "Select passwrd from user;";  MySql.Data.MySqlClient.MySqlDataReader msqlReader = msqlCommand.ExecuteReader();  msqlReader.Read();  passwordStr = msqlReader.GetString("passwrd");  }  catch (Exception er)  {  //Assert//.Show(er.Message);  }  finally  {  //always close the connection  msqlConnection.Close();  }  return passwordStr;  }  #endregion  #region Contact  public static int DoRegisterNewContact(ContactInfo contactDetails)  {  int returnVal = 0;  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  {  //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  //define the connection used by the command object  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "INSERT INTO contact(contactId,name,mobile,homePhone,officePhone,email,address,faxNumber,remark) "  + "VALUES(@contactId,@name,@mobile,@homePhone,@officePhone,@email,@address,@faxNumber,@remark)";  msqlCommand.Parameters.AddWithValue("@contactId", contactDetails.id);  msqlCommand.Parameters.AddWithValue("@name", contactDetails.name);  msqlCommand.Parameters.AddWithValue("@mobile", contactDetails.mobileno);  msqlCommand.Parameters.AddWithValue("@homePhone", contactDetails.homeno);  msqlCommand.Parameters.AddWithValue("@officePhone", contactDetails.oficeno);  msqlCommand.Parameters.AddWithValue("@email", contactDetails.email);  msqlCommand.Parameters.AddWithValue("@address", contactDetails.address);  msqlCommand.Parameters.AddWithValue("@faxNumber", contactDetails.faxno);  msqlCommand.Parameters.AddWithValue("@remark", contactDetails.remark);  msqlCommand.ExecuteNonQuery();  returnVal = 1;  }  catch (Exception er)  {  returnVal = 0;  }  finally  {  //always close the connection  msqlConnection.Close();  }  return returnVal;  }  public static List<ContactInfo> GetAllContactList()  {  return QueryAllContactList();  }  private static List<ContactInfo> QueryAllContactList()  {  List<ContactInfo> ContactList = new List<ContactInfo>();  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  { //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "Select \* From contact ;";  MySql.Data.MySqlClient.MySqlDataReader msqlReader = msqlCommand.ExecuteReader();  while (msqlReader.Read())  {  ContactInfo Contact = new ContactInfo();  Contact.id = msqlReader.GetString("contactId");  Contact.name = msqlReader.GetString("name");  Contact.mobileno = msqlReader.GetString("mobile");  Contact.homeno = msqlReader.GetString("homePhone");  Contact.oficeno = msqlReader.GetString("officePhone");  Contact.email = msqlReader.GetString("email");  Contact.address = msqlReader.GetString("address");  Contact.faxno = msqlReader.GetString("faxNumber");  Contact.remark = msqlReader.GetString("remark");    ContactList.Add(Contact);  }  }  catch (Exception er)  {  }  finally  {  //always close the connection  msqlConnection.Close();  }  return ContactList;  }  #endregion  #region Task  public static int DoRegisterNewTask(TaskInfo TaskDetails)  {  int returnVal = 0;  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  {  //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  //define the connection used by the command object  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "INSERT INTO task(taskId,priority,details) "  + "VALUES(@taskId,@priority,@details)";  msqlCommand.Parameters.AddWithValue("@taskId", TaskDetails.id);  msqlCommand.Parameters.AddWithValue("@priority", TaskDetails.value);  msqlCommand.Parameters.AddWithValue("@details", TaskDetails.taskDetails);  msqlCommand.ExecuteNonQuery();  returnVal = 1;  }  catch (Exception er)  {  returnVal = 0;  }  finally  {  //always close the connection  msqlConnection.Close();  }  return returnVal;  }  public static List<TaskInfo> GetAllTaskList()  {  return QueryAllTaskList();  }  private static List<TaskInfo> QueryAllTaskList()  {  List<TaskInfo> TaskList = new List<TaskInfo>();  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  { //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "Select \* From task;";  MySql.Data.MySqlClient.MySqlDataReader msqlReader = msqlCommand.ExecuteReader();  while (msqlReader.Read())  {  TaskInfo Task = new TaskInfo();  Task.id = msqlReader.GetString("taskId");  Task.value = msqlReader.GetString("priority");  Task.taskDetails = msqlReader.GetString("details");    TaskList.Add(Task);  }  }  catch (Exception er)  {  }  finally  {  //always close the connection  msqlConnection.Close();  }  return TaskList;  }    #endregion  #region Password Manager  public static int DoRegisterNewPassword(PasswordInfo passwordDetails)  {  int returnVal = 0;  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  {  //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  //define the connection used by the command object  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "INSERT INTO password(passwordId,name,email,userId,password,secretQuestion,secretAnswer,otherInfo) "  + "VALUES(@passwordId,@name,@email,@userId,@password,@secretQuestion,@secretAnswer,@otherInfo)";  msqlCommand.Parameters.AddWithValue("@passwordId", passwordDetails.id);  msqlCommand.Parameters.AddWithValue("@name", passwordDetails.name);  msqlCommand.Parameters.AddWithValue("@email", passwordDetails.email);  msqlCommand.Parameters.AddWithValue("@userId", passwordDetails.userId);  msqlCommand.Parameters.AddWithValue("@password", passwordDetails.password);  msqlCommand.Parameters.AddWithValue("@secretQuestion", passwordDetails.scrtqstn);  msqlCommand.Parameters.AddWithValue("@secretAnswer", passwordDetails.scrtans);  msqlCommand.Parameters.AddWithValue("@otherInfo", passwordDetails.otherInfo);  msqlCommand.ExecuteNonQuery();  returnVal = 1;  }  catch (Exception er)  {  returnVal = 0;  }  finally  {  //always close the connection  msqlConnection.Close();  }  return returnVal;  }  public static List<PasswordInfo> GetAllPasswordsList()  {  return QueryAllPasswordList();  }  private static List<PasswordInfo> QueryAllPasswordList()  {  List<PasswordInfo> PasswordList = new List<PasswordInfo>();  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  { //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "Select \* From password ;";  MySql.Data.MySqlClient.MySqlDataReader msqlReader = msqlCommand.ExecuteReader();  while (msqlReader.Read())  {  PasswordInfo Password = new PasswordInfo();  Password.id = msqlReader.GetString("passwordId");  Password.name = msqlReader.GetString("name");  Password.email = msqlReader.GetString("email");  Password.userId = msqlReader.GetString("userId");  Password.password = msqlReader.GetString("password");  Password.scrtqstn = msqlReader.GetString("secretQuestion");  Password.scrtans = msqlReader.GetString("secretAnswer");  Password.otherInfo = msqlReader.GetString("otherInfo");    PasswordList.Add(Password);  }  }  catch (Exception er)  {  }  finally  {  //always close the connection  msqlConnection.Close();  }  return PasswordList;  }  #endregion  #region Go To Date / Search  public static List<NoteInfo> searchNoteList(NoteInfo noteinfo)  {  return searchAllNoteList(noteinfo);  }  private static List<NoteInfo> searchAllNoteList(NoteInfo noteinfo)  {  List<NoteInfo> NoteList = new List<NoteInfo>();  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  { //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "Select \* From note where date = @date ; ";  msqlCommand.Parameters.AddWithValue("@date", noteinfo.gotoDate);  MySql.Data.MySqlClient.MySqlDataReader msqlReader = msqlCommand.ExecuteReader();  while (msqlReader.Read())  {  NoteInfo Note = new NoteInfo();  Note.noteDate = msqlReader.GetDateTime("date");    NoteList.Add(Note);  }  }  catch (Exception er)  {  }  finally  {  //always close the connection  msqlConnection.Close();  }  return NoteList;  }  #endregion  #region Monthly View  public static List<NoteInfo> searchMnthlyNoteList(DateTime date)  {  DateTime day1st = new DateTime(date.Year, date.Month, 1);  DateTime dayLast = new DateTime(date.Year, date.Month + 1, 1);  dayLast = dayLast.Subtract(new TimeSpan(1, 0, 0, 0));  List<NoteInfo> NoteList = new List<NoteInfo>();  MySql.Data.MySqlClient.MySqlConnection msqlConnection = OpenDbConnection();  try  { //define the command reference  MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();  msqlCommand.Connection = msqlConnection;  msqlCommand.CommandText = "SELECT \* FROM note where date(note.date) >= DATE\_SUB( @dayLast, INTERVAL @diff DAY) group by date;";  msqlCommand.Parameters.AddWithValue("@dayLast", dayLast);  msqlCommand.Parameters.AddWithValue("@diff", dayLast.Subtract(day1st));  MySql.Data.MySqlClient.MySqlDataReader msqlReader = msqlCommand.ExecuteReader();  while (msqlReader.Read())  {  NoteInfo Note = new NoteInfo();  Note.noteDate = msqlReader.GetDateTime("date");  Note.note = msqlReader.GetString("note");  NoteList.Add(Note);  }  }  catch (Exception er)  {  }  finally  {  //always close the connection  msqlConnection.Close();  }  return NoteList;  }  #endregion    }  } |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

## Comments and Description of Coding segments

### Comments and API Documentation

Sayan working on.

### Description of coding

#### Desktop Application Coding

#### Facebook App

#### Twitter App

#### Google Map API

## Standardization of the coding

12

Sayan working on.

## Code Efficiency

We started working on the project keeping in mind that we must develop it in a way that it not only provides a very easy to use GUI but also provide a fast and flexible service to the users. We know that a particular work can be done in more than one ways. We have tried all the options and then chose the one which provides the fastest and most secure performance. First of all, we have used the latest technologies of Microsoft like visual studio 2010 as IDE and WPF as GUI to keep our application’s performance few steps ahead. We have studies all the rules of software development life cycle and applied them to keep our application flexible. We have given special attention to the storage related codes. We have avoided all the unnecessary database codes and kept them as short as possible without harming our purpose so that insertion, updating, deletion and fetching of data take place flexibly. You can see the result as a user; our application does all the works very smoothly.

## Error handling

## Parameters calling/passing

## Validation checks

# Testing

## Testing techniques and testing strategies used

**DNBSN** application will be tested using following strategies to ensure that the application succeeds to complete all the functional and non functional requirements:

### Database & Data Integrity Testing

The databases and the database processes should be tested as a subsystem within the **DNBSN** Application.These subsystems should be tested with the target-of-test’s User Interface as the interface to the database.

|  |  |
| --- | --- |
| Test Objective: | Ensure that data is stored correctly, audits can be performed, access is controlled |
| Technique: | * SQL queries will be executed in the DB to verify the data content and correctness. |
| Completion Criteria: | * All planned tests have been executed. * All defects that have been identified have been resolved * All resolutions have been implemented. |

### Functional Testing:

Function testing focuses on any requirements for test that can be traced directly to use cases or business functions and business rules. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box techniques; that are verifying the application and its internal processes by interacting with the application via the Graphical User Interface (GUI) and analyzing the output or results. Identified below is an outline of the function testing recommended for **DNBSN**:

|  |  |
| --- | --- |
| Test Objective: | Ensure proper target-of-test functionality, including business process validation. |
| Technique: | Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following:   * The expected results occur when valid data is used. * The appropriate error or warning messages are displayed when invalid data is used. * Business rules are properly applied. * Black Box end to end testing of configured processes. Manual validation of required and optional fields. |
| Completion Criteria: | * All planned tests have been executed. * All defects that have been identified have been resolved * All resolutions have been implemented. |

### Regression Testing:

Regression testing focuses on software functionality that may have been previously working however through subsequent changes may have been inadvertently impacted. The goals of these tests are to verify that the broader impact of changes has been verified. Identified below is an outline of the regression testing recommended for each application(s)/module(s) of **DNBSN**.

|  |  |
| --- | --- |
| Test Objective: | Ensure that previously passed test cases continue to pass as the new system development is deployed and that surrounding systems that may be impacted by a change are still functioning as expected. |
| Technique: | * Execute previous passed testing suites to ensure the following: * The expected results occur when valid data is used. * The appropriate error or warning messages are displayed when invalid data is used. * Each business rule is properly applied. |
| Completion Criteria: | • All planned regression tests have been executed.  • All identified defects have been resolved. |

### User Interface Testing:

User Interface (UI) testing verifies a user’s interaction with the software. The goal of UI testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the target-of-test. In addition, UI testing ensures that the objects within the UI function as expected and conform to corporate or industry standards. Most of this testing will have been done during functional testing. The areas of focus will be on design, layout and navigation of the screens.

|  |  |
| --- | --- |
| Test Objective: | UI testing will verify the screens and the layouts and navigation |
| Technique: | * Verify the design and layout of the screen. * Identify the integration links. * Test the functioning of the links – that the proper page is displayed and correct messages, pop-ups are shown when they need to be displayed etc * Validation of general navigation |
| Completion Criteria: | * All navigation test cases have been executed. * All screens have been verified as per design and layouts * All defects that have been identified have been resolved. |

### Performance Profiling:

Performance profiling is a performance test in which response times, transaction rates, and other time-sensitive requirements are measured and evaluated. The goal of Performance Profiling is to verify performance requirements have been achieved. Performance profiling is implemented and executed to profile and tune performance behaviours as a function of conditions such as workload or hardware configurations

|  |  |
| --- | --- |
| Test Objective: | The purpose of performance profiling is to ensure the performance of the **DNBSN** application is up to the desired level. |
| Technique: | * Use a subset of Test Procedures developed for Function and Business Cycle Testing. * Modify data files to increase the number of transactions or the scripts to increase the number of iterations each transaction occurs. * This will be done by using Load Runner or Quick Test Professional (QTP). |
| Completion Criteria: | * Single Transaction or single user: Successful completion of the test scripts without any failures and within the expected or required time allocation per transaction. * Results are recorded and a performance baseline is created for the major logical functions within the scenarios listed above. * All performance defects are reviewed and triaged to an acceptable resolution. |

.

### Load Testing:

Load testing is a performance test which subjects the target-of-test to varying workloads to measure and evaluate the performance behaviours and ability of the target-of-test to continue to function properly under these different workloads. The goal of load testing is to determine and ensure that the system functions properly at the expected maximum workload. Additionally, load testing evaluates the performance characteristics, such as response times, transaction rates, and other time sensitive issues.

|  |  |
| --- | --- |
| Test Objective: | The purpose of load testing is to verify performance behaviour time for designated transactions or business cases under varying workload conditions. |
| Technique: | * Use a subset of Test Procedures developed for Function and Business Cycle Testing. * Scripts will be executed to simulate the peak load for 1 hour and report will be generated and analysed. * This will be done using Load Runner. |
| Completion Criteria: | * Multiple transactions or multiple users: Successful completion of the test scripts without any failures and within acceptable time allocation. * Results are recorded and a performance baseline is created for the major business functions within the scenarios listed above. * All load testing defects are reviewed and triaged to an acceptable resolution. |

### Stress Testing:

Stress testing is a type of performance test implemented and executed to find errors due to low resources or competition for resources. Low memory or disk space may reveal defects in the target-of-test that aren't apparent under normal conditions. Other defects might result from competition for shared resources like database locks or network bandwidth. Stress testing can also be used to identify the peak workload the target-of-test can handle, which is often beyond the production workload.

### Volume Testing:

Volume Testing subjects the target-of-test to large amounts of data to determine if limits are reached that cause the software to fail. Volume Testing also identifies the continuous maximum load or volume the target-of-test can handle for a given period. For example, if the target-of-test is processing a set of database records to generate a report, a Volume Test would use a large test database and check that the software behaved normally and produced the correct report.

### Security & Access Control Testing:

Security and Access Control Testing focus on following key areas of security:

* Application-level security, including access to the Data or Business Functions

Application-level security ensures the authentication and authorization of a user. Authentication ensures that the user is a valid user of the system and authorization ensures that the user has the proper privileges to perform specific tasks on desired resources of the system. Testing will be conducted to validate the rules by taking into considerations the various roles applicable for the system.

### Failover & Recovery Testing:

Failover and Recovery Testing ensures that the target-of-test can successfully failover and recover from a variety of hardware, software or network malfunctions with undue loss of data or data integrity.

Failover testing ensures that, for those systems that must be kept running, when a failover condition occurs, the alternate or backup systems properly “take over” for the failed system without loss of data or transactions.

Recovery testing is an antagonistic test process in which the application or system is exposed to extreme conditions, or simulated conditions, to cause a failure, such as device Input/Output (I/O) failures or invalid database pointers and keys. Recovery processes are invoked and the application or system is monitored and inspected to verify proper application, or system, and data recovery has been achieved.

### Configuration Testing:

Configuration testing verifies the operation of the target-of-test on different software and hardware configurations. In most production environments, the particular hardware specifications for the client workstations, network connections and database servers vary. Client workstations may have different software loaded⎯for example, applications, drivers, and so on⎯and at any one time, many different combinations may be active using different resources.

### Installation/Deploy & Back out Testing:

Installation testing has two purposes. The first is to ensure that the software can be installed under different conditions⎯such as a new installation, an upgrade and a complete or custom installation⎯under normal and abnormal conditions. Abnormal conditions include insufficient disk space, lack of privilege to create directories, and so on. The second purpose is to verify that, once installed; the software operates correctly and can be backed out successfully. This usually means running a number of the tests that were developed for Function testing before and after the back out.

### Post Production Testing:

The purpose of Post production testing is to verify that, once deployed, the software operates correctly. This usually means running a number of the tests that were developed for Function Testing ensuring that no data is changed/modified in production. Typically, the business SME’s assist with Post production testing.

### Unit Testing:

Unit testing will take place within the construction phase of the project. After application module has been built to meet design specifications, each component (screen, view, interface, conversion program, etc.) will be tested individually to help confirm that it functions properly as an individual unit. Basic performance testing will also be completed during unit test to resolve obvious issues with performance prior to the System Testing.

The resource responsible for development will conduct testing of their module using the unit test conditions defined by the developer based on detailed design documents. The final step of unit test will be a review by the team lead to obtain their signoff on the component test checklist.

### Smoke Testing:

|  |  |
| --- | --- |
| Test Objective: | Verifies the major functionality at high level in order to determine if further testing is possible. |
| Technique: | * After initial deployment to the test environment validate all critical components of the application prior to proceeding with testing. |
| Completion Criteria: | * Navigation through the application at high level is possible, testing can continue. |

### Data Migration Testing:

This is the process of testing to verify whether or not the data migration (or conversion) has been successfully completed. The testing process will be carried out by running SQL scripts on both the source and destination databases.

The fields which are present in the newdata Model in the Destination DB(s) will be migrated from the existing systemssource DB(s) to Destination DB(s).

|  |  |
| --- | --- |
| Test Objective: | The objective of this test is to verify that data migration is successful from source DB(s) to destination DB(s). |
| Technique: | * The Team is notified before the data migration. * Team runs queries on the source DB and fetches the data. * Data Migration is done. * Mapped data needs to be determined. * Team runs the queries on the Destination DB and fetches the data. * Cross verification of the data is done to see that data fetched from the old database is same as the data fetched from the new database. * Verification of the table structure. * Verification of record counts. * Verification of the data formatting. |
| Completion Criteria: | * Data fetched from the Source DB(s) and Destination DB(s) matches. * The record count in the Source and the Destination databases should be equal. * No data are truncated. * Data formatting is proper (if required at any instance). * All defects that have been identified have been resolved. |

## Testing Plan used

## Test reports for Unit Test Cases and System Test Cases

### Test reports for Unit Test Cases

|  |  |  |
| --- | --- | --- |
| Test Case Id | Comment | Status |
| **DNBSN**-001 | NA | PASS |
| **DNBSN**-002 | NA | PASS |
| **DNBSN**-003 | NA | PASS |
| **DNBSN**-004 | NA | PASS |
| **DNBSN**-005 | NA | PASS |
| **DNBSN**-006 | NA | PASS |
| **DNBSN**-007 | NA | PASS |
| **DNBSN**-008 | NA | PASS |
| **DNBSN**-009 | NA | PASS |
| **DNBSN**-010 | NA | PASS |
| **DNBSN**-011 | NA | PASS |
| **DNBSN**-012 | NA | PASS |
| **DNBSN**-013 | NA | PASS |
| **DNBSN**-014 | NA | PASS |
| **DNBSN**-015 | NA | PASS |
| **DNBSN**-016 | NA | PASS |
| **DNBSN**-017 | NA | PASS |
| **DNBSN**-018 | NA | PASS |
| **DNBSN**-019 | NA | PASS |
| **DNBSN**-020 | NA | PASS |
| **DNBSN**-021 | NA | PASS |
| **DNBSN**-022 | NA | PASS |
| **DNBSN**-023 | NA | PASS |
| **DNBSN**-024 | NA | PASS |
| **DNBSN**-025 | NA | PASS |
| **DNBSN**-026 | NA | PASS |
| **DNBSN**-027 | NA | PASS |
| **DNBSN**-028 | NA | PASS |
| **DNBSN**-029 | NA | PASS |
| **DNBSN**-030 | NA | PASS |
| **DNBSN**-031 | NA | PASS |
| **DNBSN**-032 | NA | PASS |
| **DNBSN**-033 | NA | PASS |
| **DNBSN**-034 | NA | PASS |
| **DNBSN**-035 | NA | PASS |
| **DNBSN**-036 | NA | PASS |
| **DNBSN**-037 | NA | PASS |
| **DNBSN**-038 | NA | PASS |
| **DNBSN**-039 | NA | PASS |
| **DNBSN**-040 | NA | PASS |
| **DNBSN**-041 | NA | PASS |
| **DNBSN**-042 | NA | PASS |
| **DNBSN**-043 | NA | PASS |
| **DNBSN**-044 | NA | PASS |
| **DNBSN**-045 | NA | PASS |
| **DNBSN**-046 | NA | PASS |
| **DNBSN**-047 | NA | PASS |
|  |  |  |
| **DNBSN**-048 | NA | PASS |
| **DNBSN**-049 | NA | PASS |
| **DNBSN**-050 | NA | PASS |
| **DNBSN**-051 | NA | PASS |
| **DNBSN**-052 | NA | PASS |
| **DNBSN**-053 | NA | PASS |
| **DNBSN**-054 | NA | PASS |
| **DNBSN**-055 | NA | PASS |
| **DNBSN**-056 | NA | PASS |
| **DNBSN**-057 | NA | PASS |
| **DNBSN**-058 | NA | PASS |
| **DNBSN**-059 | NA | PASS |
| **DNBSN**-060 | NA | PASS |
| **DNBSN**-061 | NA | PASS |
| **DNBSN**-062 | NA | PASS |
| **DNBSN**-063 | NA | PASS |
| **DNBSN**-064 | NA | PASS |
| **DNBSN**-065 | NA | PASS |
| **DNBSN**-066 | NA | PASS |

### Test reports for System Test Cases

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Test Case Id | Comment | Status |
| **DNBSN**-067 | NA | PASS |
| **DNBSN**-068 | NA | PASS |
| **DNBSN**-069 | NA | PASS |
| **DNBSN**-070 | NA | PASS |
| **DNBSN**-071 | NA | PASS |
| **DNBSN**-072 | NA | PASS |
| **DNBSN**-073 | NA | PASS |
| **DNBSN**-074 | NA | PASS |
| **DNBSN**-075 | NA | PASS |
| **DNBSN**-076 | NA | PASS |
| **DNBSN**-077 | NA | PASS |
| **DNBSN**-078 | NA | PASS |
| **DNBSN**-079 | NA | PASS |
| **DNBSN**-080 | NA | PASS |
| **DNBSN**-081 | NA | PASS |
| **DNBSN**-082 | NA | PASS |
| **DNBSN**-083 | NA | PASS |
| **DNBSN**-084 | NA | PASS |
| **DNBSN**-085 | NA | PASS |
| **DNBSN**-086 | NA | PASS |
| **DNBSN**-087 | NA | PASS |
| **DNBSN**-089 | NA | PASS |
| **DNBSN**-090 | NA | PASS |
| **DNBSN**-091 | NA | PASS |
| **DNBSN**-092 | NA | PASS |
| **DNBSN**-093 | NA | PASS |

## Debugging and Code improvement:

Sayan working on

# System Security measures:

## Database/data security:

It encrypts the data stored in the database so that even if someone succeeds to hack the database still not much harm could be done.

The application will use Google open-id authentication for web interface.

## Creation of User profiles and access rights

The software requires a predefined username and password to login.

It allows a guest login as well which lets a guest user this application withvery limited access to the user data.

# Cost Estimation of the Project along with Cost Estimation Model

We used the basic COCOMO model, which gives an approximate estimate of our **DNBSN** project parameters. The basic COCOMO estimation model is given by the following expressions:

Effort = a1 \* (KLOC)a2 PM

Tdev = b1 \* (Effort)b2months

Where

KLOC is the estimated size of the software product expressed in Kilo Lines of Code a1, a2, b1, b2 are constants for each category of software products.

Tdev is the estimated time to develop the software, expressed in months.

Effort is the total effort required to develop the software product, expressed in person-month (PM).

Our project is semidetached type, because the development team consists of a mixture of experienced and inexperienced staff like my guide and me. Team members may have limited experience on related system but may be unfamiliar with aspects of the system being developed.

## Estimation of development effort

For our Semi-detached class software product **DNBSN**, the formula for estimating the effort based on the code size is shown below:

Semi-detached **DNBSN**: Tdev = 3.0\*(KLOC)1.12 PM

## Estimation of development time

For our Semi-detached class software product **DNBSN**, the formula for estimating the development time based on the effort is given below:

Semi-detached **DNBSN**: Tdev = 2.5\*(Effort)0.35 months

Assume that the size of a Semi-detached **DNBSN**product has been estimated to be 3,200 lines of source code. Assume that the average salary of software engineer(me) is Rs. 20,000 per month.

Assume that the size of our

The basic COCOMO estimation formula for **DNBSN** semidetached software:

Our Effort =3.0\*(3.2)1.12PM

= 11 PM

Normal Development time = 2.5\*(11)0.35months

=6 months

Cost required to develop the product = Rs. 6 \* 20000

= Rs. 120,000

# Reports (sample layouts should be placed)

* List of Facebook updates could be generated.
* List of twitter update could be generated.
* A list of events could be generated.
* List of LinkedIn update could be generated.
* List of google plus update could be generated.

# Future scope and further enhancement of the Project

* Now it will display the text based RSS feeds and link of the multimedia contents. We will display the Multimedia contents like Video, Audio & Image in future.
* To support UNIX / Linux Based Operating systems.
* To Support Mobile Operating systems for Symbian, Meego& Android.

# Bibliography

## Website

* <http://en.wikipedia.org>
* http://en.wikipedia.org/wiki/Windows\_Presentation\_Foundation
* <http://msdn.microsoft.com/en-us/>
* http://www.c-sharpcorner.com/beginners/
* <http://www.microsoft.com/en-us/default.aspx>
* <http://www.codeplex.com/>
* <http://stackoverflow.com/>
* <http://www.codeguru.com/>
* http://www.csharpcourse.com/
* <http://www.w3schools.com>
* <http://blogs.technicise.com/>
* http://connect.technicise.com/
* <http://www.mysql.com/>
* http://www.mysql.com/support/
* <http://dev.mysql.com/>
* http://dev.mysql.com/support/
* <http://www.homeandlearn.co.uk/csharp/csharp.html>
* <http://www.wpftutorial.net/Home.html>
* https://www.facebook.com/
* <https://developers.facebook.com/>
* <https://developers.facebook.com/apps>
* <https://www.facebook.com/creatormyapp>
* https://www.facebook.com/help/
* https://twitter.com/#
* <https://dev.twitter.com/>
* <https://dev.twitter.com/discussions/3477>
* https://support.twitter.com/
* <https://developers.google.com/>
* <https://maps.google.co.in/>
* http://support.google.com/maps/?hl=en&rd=1
* <http://www.youtube.com/?gl=IN>
* <http://blendinsider.com/>
* https://github.com/
* <https://github.com/anirban-nandy>
* [https://github.com/anirban-nandy/DailyNoteBook](https://github.com/%20anirban-nandy%20/DailyNoteBook)
* <http://learn.github.com/p/intro.html>
* <http://www.vogella.com/articles/Git/article.html>
* <http://try.github.com/levels/1/challenges/1>
* <https://enterprise.github.com/support>
* <https://support.enterprise.github.com/home>

## Books

* Fundamentals of software engineering by Rajib Mall
* Pro C# 2010 and the .NET 4.0 Platform by Andrew Troselen
* C# Programming by Rob Miles

# Appendices

## IDE Used:

### Visual Studio 2010

visual_studio_logo

Microsoft Visual Studio is a powerful IDE that ensures quality code throughout the entire application lifecycle, from design to deployment. Whether we are developing applications for SharePoint, the web, Windows, Windows Phone, and beyond, Visual Studio is the ultimate all-in-one solution. Visual Studio includes a [code editor](http://en.wikipedia.org/wiki/Code_editor) supporting [IntelliSense](http://en.wikipedia.org/wiki/IntelliSense) as well as [code refactoring](http://en.wikipedia.org/wiki/Code_refactoring). The integrated [debugger](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building [GUI](http://en.wikipedia.org/wiki/GUI) applications, web designer, [class](http://en.wikipedia.org/wiki/Class_(computing)) designer, and [database schema](http://en.wikipedia.org/wiki/Database_schema) designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for [source-control](http://en.wikipedia.org/wiki/Source_control) systems (like [Subversion](http://en.wikipedia.org/wiki/Subversion_(software)) and [Visual SourceSafe](http://en.wikipedia.org/wiki/Visual_SourceSafe)) and adding new toolsets like editors and visual designers for [domain-specific languages](http://en.wikipedia.org/wiki/Domain-specific_language) or toolsets for other aspects of the [software development lifecycle](http://en.wikipedia.org/wiki/Software_development_lifecycle) (like the [Team Foundation Server](http://en.wikipedia.org/wiki/Team_Foundation_Server) client: Team Explorer).

#### Standout Features

* User interface built on Windows Presentation Foundation (WPF)
* Improved Start page
* Improved code editor
* Improved IntelliSense
* Call Hierarchy Viewer

#### What problems does it solve?

The newly designed user experience is refreshing for an application showing its age. The user interface is built on WPF and no longer relies on the limited MDI interface in previous versions; this allows for better multi-monitor support with fly-out windows. The first thing you might notice when opening Visual Studio 2010 is the new Start page. As an xaml file, this page is completely customizable and includes the ability to remove and pin project files in the Recent Projects section.

The code editor has a number of enhancements. You can scale the font by holding down [Ctrl] while scrolling the mouse wheel. In previous versions of Visual Studio, users had to set the font size through a dialog and exit to see if the changes were correct.

In Visual Studio 2010, Box Selection is enhanced to allow for zero-length boxes and improved pasting.

The feature that will see the most use (by accident if not design) is Highlight References. By selecting any symbol, such as a variable or a property, all references to the symbol are highlighted. The symbols can then be navigated by holding down [Ctrl][Shift] and pressing the up/down keys.

IntelliSense has been improved to allow for acronyms based on Pascal casing. For example, typing String.INOE and then a non-alphanumeric character will convert the call toString.IsNullOrEmpty. This still doesn’t prevent IntelliSense from interfering when you’re writing code that doesn’t exist, as you would with a unit test.

The Suggestion Completion mode allows you to type freely without IntelliSense changing the text you typed. You can toggle between Standard and Suggestion Completion modes by pressing [Ctrl][Alt]space.

IntelliSense for JavaScript has seen the most improvement, as it is now able to determine the correct structure of a variable even after the structure is changed.

In the past, I would use .NET Reflector or another tool to analyze a user’s call hierarchy; now that functionality is built-in. Right-click the user and choose View Call Hierarchy, and calls to and from the user will be available for browsing.

## Front End - WPF (Windows Presentation Framework)



Windows Presentation Foundation (WPF) is a next-generation presentation system for building Windows client applications with visually stunning user experiences. With WPF, you can create a wide range of both standalone and browser-hosted applications.

Windows Presentation Foundation (WPF) provides developers with a unified programming model for building rich Windows smart client user experiences that incorporate UI, media, and documents. Windows Presentation Foundation (WPF) is a next-generation presentation system for building Windows client applications with visually stunning user experiences. With WPF, you can create a wide range of both standalone and browser-hosted applications. The core of WPF is a resolution-independent and vector-based rendering engine that is built to take advantage of modern graphics hardware. WPF extends the core with a comprehensive set of application-development features that include Extensible Application Markup Language (XAML), controls, data binding, layout, 2-D and 3-D graphics, animation, styles, templates, documents, media, text, and typography. WPF is included in the Microsoft .NET Framework, so you can build applications that incorporate other elements of the .NET Framework class library.

The core of WPF is a resolution-independent and vector-based rendering engine that is built to take advantage of modern graphics hardware. WPF extends the core with a comprehensive set of application-development features that include Extensible Application Markup Language (XAML), controls, data binding, layout, 2-D and 3-D graphics, animation, styles, templates, documents, media, text, and typography. WPF is included in the Microsoft .NET Framework, so you can build applications that incorporate other elements of the .NET Framework class library.

#### Programming with wpf

WPF exists as a subset of .NET Framework types that are for the most part located in the [System.Windows](http://msdn.microsoft.com/en-IN/library/system.windows.aspx) namespace. If you have previously built applications with .NET Framework using managed technologies like ASP.NET and Windows Forms, the fundamental WPF programming experience should be familiar; you instantiate classes, set properties, call methods, and handle events, all using your favorite .NET Framework programming language, such as C# or Visual Basic.

#### Markup & code-behind

WPF offers additional programming enhancements for Windows client application development. One obvious enhancement is the ability to develop an application using both *markup* and *code-behind*, an experience that ASP.NET developers should be familiar with. You generally use Extensible Application Markup Language (XAML) markup to implement the appearance of an application while using managed programming languages (code-behind) to implement its behavior.

#### security

Because XBAPs are hosted in a browser, security is important. In particular, a partial-trust security sandbox is used by XBAPs to enforce restrictions that are less than or equal to the restrictions imposed on HTML-based applications. Furthermore, each HTML feature that is safe to run from XBAPs in partial trust has been tested using a comprehensive security process.

#### controls

The user experiences that are delivered by the application model are constructed controls. In WPF, "control" is an umbrella term that applies to a category of WPF classes that are hosted in either a window or a page, have a user interface (UI), and implement some behavior.

#### Wpf controls by function

The built-in WPF controls are listed here.

* **Buttons**: [Button](http://msdn.microsoft.com/en-IN/library/system.windows.controls.button.aspx) and [RepeatButton](http://msdn.microsoft.com/en-IN/library/system.windows.controls.primitives.repeatbutton.aspx).
* **Data Display**: [DataGrid](http://msdn.microsoft.com/en-IN/library/system.windows.controls.datagrid.aspx), [ListView](http://msdn.microsoft.com/en-IN/library/system.windows.controls.listview.aspx),and [TreeView](http://msdn.microsoft.com/en-IN/library/system.windows.controls.treeview.aspx).
* **Date Display and Selection**: [Calendar](http://msdn.microsoft.com/en-IN/library/system.windows.controls.calendar.aspx) and [DatePicker](http://msdn.microsoft.com/en-IN/library/system.windows.controls.datepicker.aspx).
* **Dialog Boxes**: [OpenFileDialog](http://msdn.microsoft.com/en-IN/library/microsoft.win32.openfiledialog.aspx), [PrintDialog](http://msdn.microsoft.com/en-IN/library/system.windows.controls.printdialog.aspx), and [SaveFileDialog](http://msdn.microsoft.com/en-IN/library/microsoft.win32.savefiledialog.aspx).
* **Digital Ink**: [InkCanvas](http://msdn.microsoft.com/en-IN/library/system.windows.controls.inkcanvas.aspx) and [InkPresenter](http://msdn.microsoft.com/en-IN/library/system.windows.controls.inkpresenter.aspx).
* **Documents**: [DocumentViewer](http://msdn.microsoft.com/en-IN/library/system.windows.controls.documentviewer.aspx), [FlowDocumentPageViewer](http://msdn.microsoft.com/en-IN/library/system.windows.controls.flowdocumentpageviewer.aspx), [FlowDocumentReader](http://msdn.microsoft.com/en-IN/library/system.windows.controls.flowdocumentreader.aspx), [FlowDocumentScrollViewer](http://msdn.microsoft.com/en-IN/library/system.windows.controls.flowdocumentscrollviewer.aspx), and[StickyNoteControl](http://msdn.microsoft.com/en-IN/library/system.windows.controls.stickynotecontrol.aspx).
* **Input**: [TextBox](http://msdn.microsoft.com/en-IN/library/system.windows.controls.textbox.aspx), [RichTextBox](http://msdn.microsoft.com/en-IN/library/system.windows.controls.richtextbox.aspx), and [PasswordBox](http://msdn.microsoft.com/en-IN/library/system.windows.controls.passwordbox.aspx).
* **Layout**: [Border](http://msdn.microsoft.com/en-IN/library/system.windows.controls.border.aspx), [BulletDecorator](http://msdn.microsoft.com/en-IN/library/system.windows.controls.primitives.bulletdecorator.aspx), [Canvas](http://msdn.microsoft.com/en-IN/library/system.windows.controls.canvas.aspx), [DockPanel](http://msdn.microsoft.com/en-IN/library/system.windows.controls.dockpanel.aspx), [Expander](http://msdn.microsoft.com/en-IN/library/system.windows.controls.expander.aspx), [Grid](http://msdn.microsoft.com/en-IN/library/system.windows.controls.grid.aspx), [GridView](http://msdn.microsoft.com/en-IN/library/system.windows.controls.gridview.aspx), [GridSplitter](http://msdn.microsoft.com/en-IN/library/system.windows.controls.gridsplitter.aspx), [GroupBox](http://msdn.microsoft.com/en-IN/library/system.windows.controls.groupbox.aspx), [Panel](http://msdn.microsoft.com/en-IN/library/system.windows.controls.panel.aspx),[ResizeGrip](http://msdn.microsoft.com/en-IN/library/system.windows.controls.primitives.resizegrip.aspx), [Separator](http://msdn.microsoft.com/en-IN/library/system.windows.controls.separator.aspx), [ScrollBar](http://msdn.microsoft.com/en-IN/library/system.windows.controls.primitives.scrollbar.aspx), [ScrollViewer](http://msdn.microsoft.com/en-IN/library/system.windows.controls.scrollviewer.aspx), [StackPanel](http://msdn.microsoft.com/en-IN/library/system.windows.controls.stackpanel.aspx), [Thumb](http://msdn.microsoft.com/en-IN/library/system.windows.controls.primitives.thumb.aspx), [Viewbox](http://msdn.microsoft.com/en-IN/library/system.windows.controls.viewbox.aspx), [VirtualizingStackPanel](http://msdn.microsoft.com/en-IN/library/system.windows.controls.virtualizingstackpanel.aspx), [Window](http://msdn.microsoft.com/en-IN/library/system.windows.window.aspx), and[WrapPanel](http://msdn.microsoft.com/en-IN/library/system.windows.controls.wrappanel.aspx).
* **Media**: [Image](http://msdn.microsoft.com/en-IN/library/system.windows.controls.image.aspx), [MediaElement](http://msdn.microsoft.com/en-IN/library/system.windows.controls.mediaelement.aspx), and [SoundPlayerAction](http://msdn.microsoft.com/en-IN/library/system.windows.controls.soundplayeraction.aspx).
* **Menus**: [ContextMenu](http://msdn.microsoft.com/en-IN/library/system.windows.controls.contextmenu.aspx), [Menu](http://msdn.microsoft.com/en-IN/library/system.windows.controls.menu.aspx), and [ToolBar](http://msdn.microsoft.com/en-IN/library/system.windows.controls.toolbar.aspx).
* **Navigation**: [Frame](http://msdn.microsoft.com/en-IN/library/system.windows.controls.frame.aspx), [Hyperlink](http://msdn.microsoft.com/en-IN/library/system.windows.documents.hyperlink.aspx), [Page](http://msdn.microsoft.com/en-IN/library/system.windows.controls.page.aspx), [NavigationWindow](http://msdn.microsoft.com/en-IN/library/system.windows.navigation.navigationwindow.aspx), and [TabControl](http://msdn.microsoft.com/en-IN/library/system.windows.controls.tabcontrol.aspx).
* **Selection**: [CheckBox](http://msdn.microsoft.com/en-IN/library/system.windows.controls.checkbox.aspx), [ComboBox](http://msdn.microsoft.com/en-IN/library/system.windows.controls.combobox.aspx), [ListBox](http://msdn.microsoft.com/en-IN/library/system.windows.controls.listbox.aspx), [RadioButton](http://msdn.microsoft.com/en-IN/library/system.windows.controls.radiobutton.aspx), and [Slider](http://msdn.microsoft.com/en-IN/library/system.windows.controls.slider.aspx).
* **User Information**: [AccessText](http://msdn.microsoft.com/en-IN/library/system.windows.controls.accesstext.aspx), [Label](http://msdn.microsoft.com/en-IN/library/system.windows.controls.label.aspx), [Popup](http://msdn.microsoft.com/en-IN/library/system.windows.controls.primitives.popup.aspx), [ProgressBar](http://msdn.microsoft.com/en-IN/library/system.windows.controls.progressbar.aspx), [StatusBar](http://msdn.microsoft.com/en-IN/library/system.windows.controls.primitives.statusbar.aspx), [TextBlock](http://msdn.microsoft.com/en-IN/library/system.windows.controls.textblock.aspx), and [ToolTip](http://msdn.microsoft.com/en-IN/library/system.windows.controls.tooltip.aspx).

#### layout

When you create a UI, you arrange your controls by location and size to form a layout. A key requirement of any layout is to adapt to changes in window size and display settings. Rather than forcing you to write the code to adapt a layout in these circumstances, WPF provides a first-class, extensible layout system for you.

The cornerstone of the layout system is relative positioning, which increases the ability to adapt to changing window and display conditions. In addition, the layout system manages the negotiation between controls to determine the layout. The negotiation is a two-step process: first, a control tells its parent what location and size it requires; second, the parent tells the control what space it can have.

The layout system is exposed to child controls through base WPF classes. For common layouts such as grids, stacking, and docking, WPF includes several layout controls:

* [Canvas](http://msdn.microsoft.com/en-IN/library/system.windows.controls.canvas.aspx) : Child controls provide their own layout.
* [DockPanel](http://msdn.microsoft.com/en-IN/library/system.windows.controls.dockpanel.aspx) : Child controls are aligned to the edges of the panel.
* [Grid](http://msdn.microsoft.com/en-IN/library/system.windows.controls.grid.aspx) : Child controls are positioned by rows and columns.
* [StackPanel](http://msdn.microsoft.com/en-IN/library/system.windows.controls.stackpanel.aspx) : Child controls are stacked either vertically or horizontally.
* [VirtualizingStackPanel](http://msdn.microsoft.com/en-IN/library/system.windows.controls.virtualizingstackpanel.aspx) : Child controls are virtualized and arranged on a single line that is either horizontally or vertically oriented.
* [WrapPanel](http://msdn.microsoft.com/en-IN/library/system.windows.controls.wrappanel.aspx) : Child controls are positioned in left-to-right order and wrapped to the next line when there are more controls on the current line than space allows.

#### graphics

WPF introduces an extensive, scalable, and flexible set of graphics features that have the following benefits:

* **Resolution-independent and device-independent graphics**. The basic unit of measurement in the WPF graphics system is the device independent pixel, which is 1/96th of an inch, regardless of actual screen resolution, and provides the foundation for resolution-independent and device-independent rendering. Each device-independent pixel automatically scales to match the dots-per-inch (dpi) setting of the system it renders on.
* **Improved precision**. The WPF coordinate system is measured with double-precision floating-point numbers rather than single-precision. Transformations and opacity values are also expressed as double-precision. WPF also supports a wide color gamut (scRGB) and provides integrated support for managing inputs from different color spaces.
* **Advanced graphics and animation support**. WPF simplifies graphics programming by managing animation scenes for you; there is no need to worry about scene processing, rendering loops, and bilinear interpolation. Additionally, WPF provides hit-testing support and full alpha-compositing support.
* **Hardware acceleration**. The WPF graphics system takes advantage of graphics hardware to minimize CPU usage.

## Extensible application Markup Language (XAML)



XAML stands for Extensible Application Markup Language. Its a simple language based on XML to create and initialize .NET objects with hierarchical relations. Although it was originally invented for WPF it can by used to create any kind of object trees.

Today XAML is used to create user interfaces in WPF, Silverlight, declare workflows in WF and for electronic paper in the XPS standard.

All classes in WPF have parameter less constructors and make excessive usage of properties. That is done to make it perfectly fit for XML languages like XAML.

All you can do in XAML can also be done in code. XAML ist just another way to create and initialize objects. You can use WPF without using XAML. It's up to you if you want to declare it in XAML or write it in code. Declare your UI in XAML has some advantages:

* XAML code is short and clear to read
* Separation of designer code and logic
* Graphical design tools like Expression Blend require XAML as source.
* The separation of XAML and UI logic allows it to clearly separate the roles of designer and developer.

## Programming Framework

### .NET 4.5



The .NET Framework is a development platform for building apps for Windows, Windows Phone, Windows Server, and Windows Azure. It consists of the common language runtime (CLR) and the .NET Framework class library, which includes classes, interfaces, and value types that support an extensive range of technologies. The .NET Framework provides a managed execution environment, simplified development and deployment, and integration with a variety of programming languages, including Visual Basic and Visual C#.

#### .net framework class libraries

The .NET Framework class library is a library of classes, interfaces, and value types that provide access to system functionality. It is the foundation on which .NET Framework applications, components, and controls are built. The namespaces and namespace categories in the class library are listed in the following table and documented in detail in this reference. The namespaces and categories are listed by usage, with the most frequently used namespaces appearing first.

|  |  |
| --- | --- |
| **Namespace** | **Description** |
| [System](http://msdn.microsoft.com/en-us/library/system.aspx) | The [System](http://msdn.microsoft.com/en-us/library/system.aspx) namespace contains fundamental classes and base classes that define commonly-used value and reference data types, events and event handlers, interfaces, attributes, and processing exceptions. |
| [System.Activities](http://msdn.microsoft.com/en-us/library/gg145022.aspx) | The System.Activities namespaces contain all the classes necessary to create and work with activities in Window Workflow Foundation. |
| [System.AddIn](http://msdn.microsoft.com/en-us/library/gg145020.aspx) | The System.AddIn namespaces contain types used to identify, register, activate, and control add-ins, and to allow add-ins to communicate with a host application. |
| [System.CodeDom](http://msdn.microsoft.com/en-us/library/gg145034.aspx) | The System.CodeDom namespaces contain classes that represent the elements of a source code document and that support the generation and compilation of source code in supported programming languages. |
| [System.Collections](http://msdn.microsoft.com/en-us/library/gg145035.aspx) | The System.Collections namespaces contain types that define various standard, specialized, and generic collection objects. |
| [System.ComponentModel](http://msdn.microsoft.com/en-us/library/gg145042.aspx) | The System.ComponentModel namespaces contain types that implement the run-time and design-time behavior of components and controls. Child namespaces support the Managed Extensibility Framework (MEF), provide attribute classes that define metadata for ASP.NET Dynamic Data controls, and contain types that let you define the design-time behavior of components and their user interfaces. |
| [System.Configuration](http://msdn.microsoft.com/en-us/library/gg145027.aspx) | The System.Configuration namespaces contain types for handling configuration data, such as data in machine or application configuration files. Child namespaces contain types that are used to configure an assembly, to write custom installers for components, and to support a pluggable model for adding functionality to, or removing functionality from, both client and server applications. |
| [System.Data](http://msdn.microsoft.com/en-us/library/gg145028.aspx) | The System.Data namespaces contain classes for accessing and managing data from diverse sources. The top-level namespace and a number of the child namespaces together form the ADO.NET architecture and ADO.NET data providers. For example, providers are available for SQL Server, Oracle, ODBC, and OleDB. Other child namespaces contain classes used by the ADO.NET Entity Data Model (EDM) and by WCF Data Services. |
| [System.Deployment](http://msdn.microsoft.com/en-us/library/gg145029.aspx) | The System.Deployment namespaces contain types that support deployment of ClickOnce applications. |
| [System.Device.Location](http://msdn.microsoft.com/en-us/library/system.device.location.aspx) | The [System.Device.Location](http://msdn.microsoft.com/en-us/library/system.device.location.aspx) namespace allows application developers to easily access the computer's location by using a single API. Location information may come from multiple providers, such as GPS, Wi-Fi triangulation, and cell phone tower triangulation. The [System.Device.Location](http://msdn.microsoft.com/en-us/library/system.device.location.aspx) classes provide a single API to encapsulate the multiple location providers on a computer and support seamless prioritization and transitioning between them. As a result, application developers who use this API do not need to tailor applications to specific hardware configurations. |
| [System.Diagnostics](http://msdn.microsoft.com/en-us/library/gg145030.aspx) | The System.Diagnostics namespaces contain types that enable you to interact with system processes, event logs, and performance counters. Child namespaces contain types to interact with code analysis tools, to support contracts, to extend design-time support for application monitoring and instrumentation, to log event data using the Event Tracing for Windows (ETW) tracing subsystem, to read to and write from event logs and collect performance data, and to read and write debug symbol information. |
| [System.DirectoryServices](http://msdn.microsoft.com/en-us/library/gg145037.aspx) | The System.DirectoryServices namespaces contain types that provide access to Active Directory from managed code. |
| [System.Drawing](http://msdn.microsoft.com/en-us/library/gg145023.aspx) | The System.Drawing parent namespace contains types that support basic GDI+ graphics functionality. Child namespaces support advanced two-dimensional and vector graphics functionality, advanced imaging functionality, and print-related and typographical services. A child namespace also contains types that extend design-time user-interface logic and drawing. |
| [System.Dynamic](http://msdn.microsoft.com/en-us/library/system.dynamic.aspx) | The [System.Dynamic](http://msdn.microsoft.com/en-us/library/system.dynamic.aspx) namespace provides classes and interfaces that support Dynamic Language Runtime. |
| [System.EnterpriseServices](http://msdn.microsoft.com/en-us/library/gg145047.aspx) | The System.EnterpriseServices namespaces contain types that define the COM+ services architecture, which provides an infrastructure for enterprise applications. A child namespace supports Compensating Resource Manager (CRM), a COM+ service that enables non-transactional objects to be included in Microsoft Distributed Transaction Coordinator (DTC) transactions. Child namespaces are described briefly in the following table and documented in detail in this reference. |
| [System.Globalization](http://msdn.microsoft.com/en-us/library/system.globalization.aspx) | The [System.Globalization](http://msdn.microsoft.com/en-us/library/system.globalization.aspx) namespace contains classes that define culture-related information, including language, country/region, calendars in use, format patterns for dates, currency, and numbers, and sort order for strings. These classes are useful for writing globalized (internationalized) applications. Classes such as [StringInfo](http://msdn.microsoft.com/en-us/library/system.globalization.stringinfo.aspx) and[TextInfo](http://msdn.microsoft.com/en-us/library/system.globalization.textinfo.aspx) provide advanced globalization functionalities, including surrogate support and text element processing. |
| [System.IdentityModel](http://msdn.microsoft.com/en-us/library/gg145031.aspx) | The System.IdentityModel namespaces contain types that are used to provide authentication and authorization for .NET applications. |
| [System.IO](http://msdn.microsoft.com/en-us/library/gg145019.aspx) | The System.IO namespaces contain types that support input and output, including the ability to read and write data to streams either synchronously or asynchronously, to compress data in streams, to create and use isolated stores, to map files to an application's logical address space, to store multiple data objects in a single container, to communicate using anonymous or named pipes, to implement custom logging, and to handle the flow of data to and from serial ports. |
| [System.Linq](http://msdn.microsoft.com/en-us/library/gg145016.aspx) | The System.Linq namespaces contain types that support queries that use Language-Integrated Query (LINQ). This includes types that represent queries as objects in expression trees. |
| [System.Management](http://msdn.microsoft.com/en-us/library/gg145024.aspx) | The System.Management namespaces contain types that provide access to management information and management events about the system, devices, and applications instrumented to the Windows Management Instrumentation (WMI) infrastructure. These namespaces also contain types necessary for instrumenting applications so that they expose their management information and events through WMI to potential customers. |
| [System.Media](http://msdn.microsoft.com/en-us/library/system.media.aspx) | The [System.Media](http://msdn.microsoft.com/en-us/library/system.media.aspx) namespace contains classes for playing sound files and accessing sounds provided by the system. |
| [System.Messaging](http://msdn.microsoft.com/en-us/library/gg145046.aspx) | The System.Messaging namespaces contain types that enable you to connect to, monitor, and administer message queues on the network and to send, receive, or peek messages. A child namespace contains classes that can be used to extend design-time support for messaging classes. |
| [System.Net](http://msdn.microsoft.com/en-us/library/gg145039.aspx) | The System.Net namespaces contain classes that provide a simple programming interface for a number of network protocols, programmatically access and update configuration settings for the System.Net namespaces, define cache policies for web resources, compose and send e-mail, represent Multipurpose Internet Mail Exchange (MIME) headers, access network traffic data and network address information, and access peer-to-peer networking functionality. Additional child namespaces provide a managed implementation of the Windows Sockets (Winsock) interface and provide access to network streams for secure communications between hosts. |
| [System.Numerics](http://msdn.microsoft.com/en-us/library/system.numerics.aspx) | The [System.Numerics](http://msdn.microsoft.com/en-us/library/system.numerics.aspx) namespace contains numeric types that complement the numeric primitives, such as [Byte](http://msdn.microsoft.com/en-us/library/system.byte.aspx), [Double](http://msdn.microsoft.com/en-us/library/system.double.aspx), and [Int32](http://msdn.microsoft.com/en-us/library/system.int32.aspx), that are defined by the .NET Framework. |
| [System.Printing](http://msdn.microsoft.com/en-us/library/gg145044.aspx) | The System.Printing namespaces contain types that support printing, that provide access to the properties of print system objects and enable rapid copying of their property settings to another object of the same type, and that support the interconversion of managed System.PrintTicket objects and unmanaged GDI DEVMODE structures. |
| [System.Reflection](http://msdn.microsoft.com/en-us/library/gg145033.aspx) | The System.Reflection namespaces contain types that provide a managed view of loaded types, methods, and fields, and that can dynamically create and invoke types. A child namespace contains types that enable a compiler or other tool to emit metadata and Microsoft intermediate language (MSIL). |
| [System.Resources](http://msdn.microsoft.com/en-us/library/gg145043.aspx) | The System.Resources namespaces contain types that enable developers to create, store, and manage an application's culture-specific resources. |
| [System.Runtime](http://msdn.microsoft.com/en-us/library/gg145017.aspx) | The System.Runtime namespaces contain types that support an application's interaction with the common language runtime, and types that enable features such as application data caching, advanced exception handling, application activation within application domains, COM interop, distributed applications, serialization and deserialization, and versioning. Additional namespaces enable compiler writers to specify attributes that affect the run-time behavior of the common language runtime, define a contract for reliability between a set of code and other code that takes a dependency on it, and implement a persistence provider for Windows Communication Foundation (WCF). |
| [System.Security](http://msdn.microsoft.com/en-us/library/gg145025.aspx) | The System.Security namespaces contain classes that represent the .NET Framework security system and permissions. Child namespaces provide types that control access to and audit securable objects, allow authentication, provide crytographic services, control access to operations and resources based on policy, and support rights management of application-created content. |
| [System.ServiceModel](http://msdn.microsoft.com/en-us/library/gg145010.aspx) | The System.ServiceModel namespaces contain the types necessary to build Windows Communication Foundation (WCF) service and client applications. |
| [System.ServiceProcess](http://msdn.microsoft.com/en-us/library/gg145038.aspx) | The System.ServiceProcess namespaces contain types that enable you to implement, install, and control Windows service applications and extend design-time support for Windows service applications. |
| [System.Speech](http://msdn.microsoft.com/en-us/library/gg145021.aspx) | The System.Speech namespaces contain types that support speech recognition. |
| [System.Text](http://msdn.microsoft.com/en-us/library/gg145012.aspx) | The System.Text namespaces contain types for character encoding and string manipulation. A child namespace enables you to process text using regular expressions. |
| [System.Threading](http://msdn.microsoft.com/en-us/library/gg145014.aspx) | The System.Threading namespaces contain types that enable multithreaded programming. A child namespace provides types that simplify the work of writing concurrent and asynchronous code. |
| [System.Timers](http://msdn.microsoft.com/en-us/library/system.timers.aspx) | The [System.Timers](http://msdn.microsoft.com/en-us/library/system.timers.aspx) namespace provides the [Timer](http://msdn.microsoft.com/en-us/library/system.timers.timer.aspx) component, which allows you to raise an event on a specified interval. |
| [System.Transactions](http://msdn.microsoft.com/en-us/library/gg145032.aspx) | The System.Transactions namespaces contain types that support transactions with multiple, distributed participants, multiple phase notifications, and durable enlistments. A child namespace contains types that describe the configuration options used by the System.Transactions types. |
| [System.Web](http://msdn.microsoft.com/en-us/library/gg145018.aspx) | The System.Web namespaces contain types that enable browser/server communication. Child namespaces include types that support ASP.NET forms authentication, application services, data caching on the server, ASP.NET application configuration, dynamic data, HTTP handlers, JSON serialization, incorporating AJAX functionality into ASP.NET, ASP.NET security, and web services. |
| [System.Windows](http://msdn.microsoft.com/en-us/library/gg145013.aspx) | The System.Windows namespaces contain types used in Windows Presentation Foundation (WPF) applications, including animation clients, user interface controls, data binding, and type conversion. System.Windows.Forms and its child namespaces are used for developing Windows Forms applications. |
| [System.Workflow](http://msdn.microsoft.com/en-us/library/gg145026.aspx) | The System.Workflow namespaces contain types used to develop applications that use Windows Workflow Foundation. These types provide design time and run-time support for rules and activities, to configure, control, host, and debug the workflow runtime engine. |
| [System.Xaml](http://msdn.microsoft.com/en-us/library/gg145048.aspx) | The System.Xaml namespaces contain types that support parsing and processing the Extensible Application Markup Language (XAML). |
| [System.Xml](http://msdn.microsoft.com/en-us/library/gg145036.aspx) | The System.Xml namespaces contain types for processing XML. Child namespaces support serialization of XML documents or streams, XSD schemas, XQuery 1.0 and XPath 2.0, and LINQ to XML, which is an in-memory XML programming interface that enables easy modification of XML documents. |
| [Accessibility](http://msdn.microsoft.com/en-us/library/accessibility.aspx) | The [Accessibility](http://msdn.microsoft.com/en-us/library/accessibility.aspx) and all of its exposed members are part of a managed wrapper for the Component Object Model (COM) accessibility interface. |
| [Microsoft.Activities](http://msdn.microsoft.com/en-us/library/hh135392.aspx) | The Microsoft.Activities namespaces contain types that support MSBuild and debugger extensions for Windows Workflow Foundation applications. |
| [Microsoft.Aspnet.Snapin](http://msdn.microsoft.com/en-us/library/microsoft.aspnet.snapin.aspx) | The [Microsoft.Aspnet.Snapin](http://msdn.microsoft.com/en-us/library/microsoft.aspnet.snapin.aspx) namespace defines the types necessary for the ASP.NET management console application to interact with Microsoft Management Console (MMC). For more information, see "MMC Programmer's Guide" in the [MSDN Library](http://go.microsoft.com/fwlink/?linkid=37118). |
| [Microsoft.Build](http://msdn.microsoft.com/en-us/library/gg145008.aspx) | The Microsoft.Build namespaces contain types that provide programmatic access to, and control of, the MSBuild engine. |
| [Microsoft.CSharp](http://msdn.microsoft.com/en-us/library/gg145015.aspx) | The Microsoft.CSharp namespaces contain types that support compilation and code generation of source code written in the C# language, and types that support interoperation betwen the dynamic language runtime (DLR) and C#. |
| [Microsoft.Data.Entity.Build.Tasks](http://msdn.microsoft.com/en-us/library/microsoft.data.entity.build.tasks.aspx) | The [Microsoft.Data.Entity.Build.Tasks](http://msdn.microsoft.com/en-us/library/microsoft.data.entity.build.tasks.aspx) namespace contains two MSBuild tasks that are used by the ADO.NET Entity Data Model Designer (Entity Designer). |
| [Microsoft.JScript](http://msdn.microsoft.com/en-us/library/gg145041.aspx) | The Microsoft.JScript namespaces contain classes that support compilation and code generation using the JScript language. |
| [Microsoft.SqlServer.Server](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.server.aspx) | The [Microsoft.SqlServer.Server](http://msdn.microsoft.com/en-us/library/microsoft.sqlserver.server.aspx) namespace contains classes, interfaces, and enumerations that are specific to the integration of the Microsoft .NET Framework common language runtime (CLR) into Microsoft SQL Server, and the SQL Server database engine process execution environment. |
| [Microsoft.VisualBasic](http://msdn.microsoft.com/en-us/library/gg145009.aspx) | The Microsoft.VisualBasic namespaces contain classes that support compilation and code generation using the Visual Basic language. Child namespaces contain types that provide services to the Visual Basic compiler and types that include support for the Visual Basic application model, the My namespace, lambda expressions, and code conversion. |
| [Microsoft.VisualC](http://msdn.microsoft.com/en-us/library/gg145040.aspx) | The Microsoft.VisualC namespaces contain types that support the Visual C++ compiler and types that implement the STL/CLR Library and the generic interface to the STL/CLR Library. |
| [Microsoft.Win32](http://msdn.microsoft.com/en-us/library/gg145011.aspx) | The Microsoft.Win32 namespaces provide types that handle events raised by the operating system, that manipulate the system registry, and that represent file and operating system handles. |
| [Microsoft.Windows](http://msdn.microsoft.com/en-us/library/hh135393.aspx) | The Microsoft.Windows namespaces contain types that support themes and preview in Windows Presentation Framework (WPF) applications. |
| [UIAutomationClientsideProviders](http://msdn.microsoft.com/en-us/library/uiautomationclientsideproviders.aspx) | Contains a single type that maps client automation providers. |
| [XamlGeneratedNamespace](http://msdn.microsoft.com/en-us/library/xamlgeneratednamespace.aspx) | Contains compiler-generated types that are not intended to be used directly from your code. |

## Database/backend:

### MySQL



MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout its history.

The MySQL Community Edition includes:

* Pluggable Storage Engine Architecture
* Multiple Storage Engines: InnoDB , MyISAM, NDB (MySQL Cluster),Memory ,Merge , Archive, CSV
* MySQL Replication to improve application performance and scalability
* MySQL Partitioning to improve performance and management of large database applications
* Stored Procedures to improve developer productivity

#### Detailed features of mysql

The following list shows the most important properties of MySQL. This section is directed to the reader who already has some knowledge of relational databases. We will use some terminology from the relational database world without defining our terms exactly. On the other hand, the explanations should make it possible for database novices to understand to some extent what we are talking about.

**Relational Database System:** Like almost all other database systems on the market, MySQL is a relational database system.

**Client/Server Architecture:** MySQL is a client/server system. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc. The clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).

|  |  |
| --- | --- |
|  | |
|  |  |
|  | |

Almost all of the familiar large database systems (Oracle, Microsoft SQL Server, etc.) are client/server systems. These are in contrast to the file-server systems, which include Microsoft Access, dBase and FoxPro. The decisive drawback to file-server systems is that when run over a network, they become extremely inefficient as the number of users grows.

**SQL compatibility:** MySQL supports as its database language -- as its name suggests – SQL (Structured Query Language). SQL is a standardized language for querying and updating data and for the administration of a database. There are several SQL dialects (about as many as there are database systems). MySQL adheres to the current SQL standard (at the moment SQL:2003), although with significant restrictions and a large number of extensions.

Through the configuration setting sql-mode you can make the MySQL server behave for the most part compatibly with various database systems. Among these are IBM DB/2 and Oracle. (The setting sql-mode changes some of the syntax conventions, and performs no miracles.

**SubSELECTs:** Since version 4.1, MySQL is capable of processing a query in the form SELECT \* FROM table1 WHERE x IN (SELECT y FROM table2) (There are also numerous syntax variants for subSELECTs.)

**Views:** Put simply, views relate to an SQL query that is viewed as a distinct database object and makes possible a particular view of the database. MySQL has supported views since version 5.0.

**Stored procedures:** Here we are dealing with SQL code that is stored in the database system.

Stored procedures (SPs for short) are generally used to simplify certain steps, such as inserting or deleting a data record. For client programmers this has the advantage that they do not have to process the tables directly, but can rely on SPs. Like views, SPs help in the administration of large database projects. SPs can also increase efficiency. MySQL has supported SPs since version 5.0.

**Triggers:** Triggers are SQL commands that are automatically executed by the server in certain database operations (INSERT, UPDATE, and DELETE). MySQL has supported triggers in a limited form from version 5.0, and additional functionality is promised for version 5.1.

**Unicode:** MySQL has supported all conceivable character sets since version 4.1, including Latin-1, Latin-2, and Unicode (either in the variant UTF8 or UCS2).

**User interface:** There are a number of convenient user interfaces for administering a MySQL server.

**Full-text search:** Full-text search simplifies and accelerates the search for words that are located within a text field. If you employ MySQL for storing text (such as in an Internet discussion group), you can use full-text search to implement simply an efficient search function.

**Replication:** Replication allows the contents of a database to be copied (replicated) onto a number of computers. In practice, this is done for two reasons: to increase protection against system failure (so that if one computer goes down, another can be put into service) and to improve the speed of database queries.

**Transactions:** In the context of a database system, a transaction means the execution of several database operations as a block. The database system ensures that either all of the operations are correctly executed or none of them. This holds even if in the middle of a transaction there is a power failure, the computer crashes, or some other disaster occurs. Thus, for example, it cannot occur that a sum of money is withdrawn from account A but fails to be deposited in account B due to some type of system error.

Transactions also give programmers the possibility of interrupting a series of already executed commands (a sort of revocation). In many situations this leads to a considerable simplification of the programming process. In spite of popular opinion, MySQL has supported transactions for a long time. One should note here that MySQL can store tables in a variety of formats. The default table format is called MyISAM, and this format does not support transactions. But there are a number of additional formats that do support transactions. The most popular of these is InnoDB, which will be described extensively in this book.

**Foreign key constraints:** These are rules that ensure that there are no cross references in linked tables that lead to nowhere. MySQL supports foreign key constraints for InnoDB tables.

**GIS functions:** Since version 4.1, MySQL has supported the storing and processing of two-dimensional geographical data. Thus MySQL is well suited for GIS (geographic information systems) applications.

**Programming languages:** There are quite a number of APIs (application programming interfaces) and libraries for the development of MySQL applications. For client programming you can use, among others, the languages C, C++, Java, Perl, PHP, Python, and Tcl.

**ODBC:** MySQL supports the ODBC interface [Connector/ODBC](http://searchenterpriselinux.techtarget.com/definition/MySQL-Connector-ODBC). This allows MySQL to be addressed by all the usual programming languages that run under Microsoft Windows (Delphi, Visual Basic, etc.). The ODBC interface can also be implemented under Unix, though that is seldom necessary.

Windows programmers who have migrated to Microsoft's new .NET platform can, if they wish, use the ODBC provider or the .NET interface Connector/NET.

**Platform independence:** It is not only client applications that run under a variety of operating systems; MySQL itself (that is, the server) can be executed under a number of operating systems. The most important are Apple Macintosh OS X, Linux, Microsoft Windows, and the countless Unix variants, such as AIX, BSDI, FreeBSD, HP-UX, OpenBSD, Net BSD, SGI Iris, and Sun Solaris.

**Speed:** MySQL is considered a very fast database program. This speed has been backed up by a large number of benchmark.

## ide for Database

### MySQL workbench

MySQL Workbench is a visual database design tool that integrates SQL evelopment,administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system. It is the successor to DBDesigner 4 from fabFORCE.net, and replaces the previous package of software,MySQL GUI Tools Bundle. [MySQL Workbench](http://www.mysql.com/products/workbench/) enables a DBA, developer, or data architect to visually design, generate, and manage all types of databases including Web, OLTP, and data warehouse databases. It includes everything a data modeler needs for creating complex ER models, and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort. MySQL Workbench is available on Windows, Linux and Mac OS.

#### benefits

* Simplifies database design and maintenance
* Automates time-consuming and error-prone tasks
* Enables data architects to visualize requirements, communicate with stakeholders, and resolve design issues before a major investment of time and resources is made
* Enables model-driven database design—the most efficient methodology for creating valid and well-performing databases—while providing the flexibility to respond to evolving business requirements
* Provides capabilities to forward-engineer physical database designs and reverse-engineer existing databases
* Allows you to import SQL scripts to build models and export models to DDL scripts that can be run at a later time
* Enables you to compare two live databases or a model and a live database, visually see the differences, and perform a synchronization between a model and a live database or vice versa
* Simplifies the documentation of database designs, providing a point-and-click process that delivers documentation in HTML or plain-text format

#### tools

The three main tools of MySQL Workbench are:

* SQL Development
* Data Modelling
* Server Administration

## Programming Language

### C# - C sharp



C# is a type-safe, object-oriented language that is simple yet powerful, allowing programmers to build a breadth of applications. C# is a [multi-paradigm programming language](http://en.wikipedia.org/wiki/Multi-paradigm_programming_language) encompassing [imperative](http://en.wikipedia.org/wiki/Imperative_programming), [declarative](http://en.wikipedia.org/wiki/Declarative_programming), [functional](http://en.wikipedia.org/wiki/Functional_programming), [generic](http://en.wikipedia.org/wiki/Generic_programming), [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming)([class-based](http://en.wikipedia.org/wiki/Class_(computer_science))), and [component-oriented](http://en.wikipedia.org/wiki/Component-based_software_engineering) programming disciplines. It was developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft) within the [.NET](http://en.wikipedia.org/wiki/.NET_Framework) initiative and later approved as a standard by [Ecma](http://en.wikipedia.org/wiki/Ecma_International) (ECMA-334) and [ISO](http://en.wikipedia.org/wiki/International_Organization_for_Standardization) (ISO/IEC 23270). C# is one of the programming languages designed for the [Common Language Infrastructure](http://en.wikipedia.org/wiki/Common_Language_Infrastructure).

C# is intended to be a simple, modern, general-purpose, object-oriented programming language.

#### Characteristics of C#:

C# was developed to bring rapid development to C++ without sacrificing the power and control of C and C++. C# provides various characteristics, which are:  
Simple:  
C# eliminates the use of tedious operators such as -->, and pointers. C# treats inter and Boolean as two different data types, which enable the compiler   
to recognize the use of = in place of = = with if statement.  
  
**Consistent:-**  
C# supports only one integer type and there is no limitation of range.  
**Modern:-**  
C# contains various features necessary to develop web applications. Following are the features of C#:  
It provides automatic garbage collection.  
It provides robust security model.  
It provides decimal data type for financial application.  
It provides modern approach for debugging.  
It provides a rich intrinsic model for error handling.  
**Object Oriented:-**  
C# supports all the features of object oriented language such as encapsulation, inheritance and polymorphism. It treats everything as an object and there are no global   
functions, variables and constants in C#.  
**Type Safe:-**  
C# provides various type safe measures, which are:   
Dynamically allocated objects and arrays are initialized to zero.  
Products an error message while using an uninitialized variable.  
Checks the range of an array and warns when the access goes out of bound.  
Unsafe casts are not allowed.  
Enforces overflow checking in arithmetic operations.  
**Versionable:-**  
C# supports versioning that enables the existing applications to run on different versions with the help of new and override command.  
Compatible:  
C# contains the .NET specifications and therefore, allows inter operation with other .NET languages.  
**Flexible:-**  
C# does not support pointers but you may use pointers to manipulate the data of certain classes and methods by declaring them unsafe.  
Inter-operability:  
C# enables a program to call out any native API. It also allows the use of COM objects written in different languages.

## Dia for Diagram Drawing &Modeling

Dia is free and open source general-purpose diagramming software, developed as part of the GNOME project's office suite and was originally created by Alexander Larsson. Dia uses a controlled single document interface (CSDI) similar to GIMP and Sodipodi.

Dia has a modular design with several shape packages available for different needs: flowchart, network diagrams, circuit diagrams, and more. It does not restrict symbols and connectors from various categories from being placed together.

Dia is a gtk+ based diagram creation program released under the GPL license.

Dia is inspired by the commercial Windows program 'Visio', though more geared towards informal diagrams for casual use. It can be used to draw many different kinds of diagrams. It currently has special objects to help draw entity relationship diagrams, UML diagrams, flowcharts, network diagrams, and many other diagrams. It is also possible to add support for new shapes by writing simple XML files, using a subset of SVG to draw the shape.

It can load and save diagrams to a custom XML format (gzipped by default, to save space), can export diagrams to a number of formats, including EPS, SVG, XFIG, WMF and PNG, and can print diagrams (including ones that span multiple pages).

## Google Spreadsheet Interface:

*With Google Spreadsheets, we can easily create, share, and edit spreadsheets online. Here are a few specific things we can do:*

* *Import and export these file types: .xls, .csv, .txt and .ods. We can also export data to a PDF or an HTML file.*
* *Format cells and edit formulas so we can calculate results and make data look the way we want it.*
* *Chat in real time with others who are editing our spreadsheet.*
* *Embed a spreadsheet, or a section of a spreadsheet, in our blog or website.*

## Cacoo:: online drawing tool

 Cacoo is a diagram creation tool that runs in your web browser.Multiple people can work together on the same diagram in real time.Diagrams can be published directly to websites, wikis, and blogs.

### Creating Diagrams

* Elements can be dragged and drop to easily create diagrams.
* Elements can be linked together with connectors.
* Connectors automatically move when elements are repositioned.
* You can use a text box and put text anywhere you like.
* You can upload images from your PC and include them in Diagrams.
* You can take screenshots of your computer from within Cacoo.
* Smart styles can easily be applied to stencils.
* You can have multiple sheets in a diagram and use them as backgrounds or layers.
* When you move the objects on your canvas, they will be snapped at the objects or grids nearby and align automatically.
* Copying, pasting and other functionality of basic drawing software is also built in to Cacoo.
* All actions are stored so there are unlimited levels of undo.
* You can import an image from the other websites by indicating the URL.
* The imported image can be easily trimmed only using your mouse.
* According to your editing status, tips will be shown on the right bottom corner of the canvas.

### Collaboration

* You can invite collaborators to work with you in Cacoo.
* Multiple people can edit a diagram in real time.
* There is a chat function in the editor so people can communicate while creating diagrams.
* People can leave comments about the diagrams.
* Each user can set their own user icon.
* When editing with multiple people, users icons appear on selected objects.
* Sharing diagrams become much smoother. Diagrams in the shared folders can be accessible and editable by people who you have shared the folder with.

### Sharing Diagrams

* If you keep the diagram private then other users can't see it.
* If you make the diagram URL public, then anyone who knows the URL can see it.
* Publishing a diagram to a blog can be useful in various ways.
* You can place code into blogs to create a slideshow
* Published images always display the most recent version.
* Diagrams can be exported to SVG format (Plus Plan users only) and PNG format. (More formats will be available in the future.)
* Diagrams can be posted to Twitter/Facebook/GoogleBuzz
* Diagrams can be displayed in SVG format for printing. (Plus Plan users only. A few browsers are not supported.)

### Managing Diagrams

* Diagrams can be placed into folders.
* Diagrams can be copied.
* Diagrams can be displayed as thumbnails or as a list.

### Languages and Time Zones

* All pages and notification e-mails support English and Japanese
* Users can enter text from almost all languages.
* Dates are displayed relative to your local time zone.

### Security

* Private diagrams can only be seen by users you select.
* URLs which you do not share can not be found by other users or search engines.
* All editing and management is protected by SSL.
* In order to access information about diagrams a Cacoo ID and password are requited.
* User passwords are encrypted on Cacoo's server.

### API

* You can access Cacoo using the API.
* The Cacoo API supports OAuth and an API Key.

By using the Cacoo API you are able to interact with Cacoo from other services and applications.

Authorization Methods

There are two ways to access the Cacoo API.

#### 1. API Key

The API key allows you make requests to the Cacoo API. You can make an API key here.

#### API Key

Append your API key to requests to the API to return data from your account.(Parameter name "apiKey")

Example: https://cacoo.com/api/v1/diagrams.json?apiKey=abcdefghijklmn

#### 2. OAuth

OAuth 1.0a is supported as an authorization method for Cacoo. You can register applications here.

You can get your Access Token from the following links.

#### applications

Access Token:https://cacoo.com/oauth/access\_token

Authorize:https://cacoo.com/oauth/authorize

Request Token:https://cacoo.com/oauth/request\_token

## http://t1.gstatic.com/images?q=tbn:ANd9GcS-CmbHGLD4MH83JH1oNIr_acREqblVhrcFuvQfYZR8HFi1Upaqlg Version Control System : GitHub

GitHub is a web-based hosting service for software development projects that use the Git revision control system. GitHub offers both paid plans for private repositories, and free accounts for open source projects. As of May 2011, GitHub was the most popular open source code repository site.GitHub Inc. was founded in 2008 and is based in San Francisco, California.

### Description

The site provides social networking functionality such as feeds, followers and the network graph to display how developers work on their versions of a repository.

GitHub also operates other services: a pastebin-style site called Gist that provides wikis for individual repositories and web pages that can be edited through a Git repository, a slide hosting service called Speaker Deck, and a web analytics platform called Gauges.

As of January 2010, GitHub is operated under the name GitHub, Inc.

The software that runs GitHub was written using Ruby on Rails and Erlang by GitHub, Inc. (previously known as Logical Awesome) developers Chris Wanstrath, PJ Hyett, and Tom Preston-Werner.

### Limitations and constraints

According to the terms of service,if an account's bandwidth usage significantly exceeds the average of other GitHub customers, the account's file hosting service may be immediately disabled or throttled until bandwidth consumption is reduced. In addition, while there is no hard limit, the guideline for the maximum size of a repository is one gigabyte.

# Glossary.