Daily Notebook & Social Networking Updater

**MCA 6th Semester**

**MCSP-060 (Project Report)**

**Anirban Nandy**

**Enrollment No-105057536**

# Index

Contents

[**MCA 6th Semester** 1](#_Toc351476508)

[**MCSP-060 (Project Report)** 1](#_Toc351476509)

[**Anirban Nandy** 1](#_Toc351476510)

[**Enrollment No-105057536** 1](#_Toc351476511)

[Index 2](#_Toc351476512)

[Introduction: 4](#_Toc351476513)

[**Figure:** Overview of Daily notebook & Social Networking Updater 5](#_Toc351476514)

[Objective: 5](#_Toc351476515)

[System Analysis 6](#_Toc351476516)

[Identification of Need: 6](#_Toc351476517)

[Preliminary Investigation: 6](#_Toc351476518)

[Feasibility Study: 6](#_Toc351476519)

[Project Planning & Scheduling: 6](#_Toc351476520)

[Gantt chart 6](#_Toc351476521)

[Tracking Gantt 7](#_Toc351476522)

[Pert Chart 7](#_Toc351476523)

[Software requirement specifications (SRS): 8](#_Toc351476524)

[Functional Requirement 9](#_Toc351476525)

[Non-functional Requirements 11](#_Toc351476526)

[Software Engineering Paradigm applied 12](#_Toc351476527)

[Data models 12](#_Toc351476528)

[Context Diagram 12](#_Toc351476529)

[0-Level DFD 13](#_Toc351476530)

[1-Level DFD 14](#_Toc351476531)

[2-Level DFD 15](#_Toc351476532)

[Sequence diagrams 15](#_Toc351476533)

[Entity Relationship Model, 18](#_Toc351476534)

[Class Diagrams 19](#_Toc351476535)

[Activity Diagrams 20](#_Toc351476536)

[System Design 23](#_Toc351476537)

[Modularisation details 23](#_Toc351476538)

[DNBSN Engine: 23](#_Toc351476539)

[DNBSN GUI: 23](#_Toc351476540)

[DNBSN Storage: 23](#_Toc351476541)

[Google Calendar: 23](#_Toc351476542)

[Facebook/Twitter/LinkedIn API: 24](#_Toc351476543)

[Data integrity and constraints 24](#_Toc351476544)

[Entity integrity 24](#_Toc351476545)

[Referential Integrity 24](#_Toc351476546)

[Domain Integrity 24](#_Toc351476547)

[User Defined Integrity 24](#_Toc351476548)

[Database design 24](#_Toc351476549)

[User Interface Design 28](#_Toc351476550)

[Test Cases (Unit Test Cases and System Test Cases) 28](#_Toc351476551)

[Coding 28](#_Toc351476552)

[Complete Project Coding 28](#_Toc351476553)

[Comments and Description of Coding segments 28](#_Toc351476554)

[Standardization of the coding 28](#_Toc351476555)

[Code Efficiency 28](#_Toc351476556)

[Error handling 29](#_Toc351476557)

[Parameters calling/passing 29](#_Toc351476558)

[Validation checks 29](#_Toc351476559)

[Testing 29](#_Toc351476560)

[Testing techniques and Testing strategies used 29](#_Toc351476561)

[Testing Plan used 29](#_Toc351476562)

[Test reports for Unit Test Cases and System Test Cases 29](#_Toc351476563)

[Debugging and Code improvement: 29](#_Toc351476564)

[System Security measures: 29](#_Toc351476565)

[Database/data security: 29](#_Toc351476566)

[Creation of User profiles and access rights 29](#_Toc351476567)

[Cost Estimation of the Project along with Cost Estimation Model 29](#_Toc351476568)

[Reports (sample layouts should be placed) 30](#_Toc351476569)

[Future scope and further enhancement of the Project 31](#_Toc351476570)

[Bibliography 31](#_Toc351476571)

[Appendices (if any) 31](#_Toc351476572)

[Glossary. 31](#_Toc351476573)

# 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 minimize the 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 existing User.

**Input:**

User will provide user 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 event for 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 search event.

**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

Screen shot

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

### Unit Test Cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Type | Github ID | Subject | Test Name | Test Description | Step Name | Description | Expected Result |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Check Successful Connection Rest Console with IS server | The purpose of this test is to verify that the Rest Console response log displays Imagenow version number while connecting with IS server. | Step1 | Open Rest Console in Google Chrome. | Rest Console should open successfully. |
|  |  |  |  |  | Step2 | Configure Rest Console in VM as following: 1.Target: **Request URI:** http://localhost:8080/integrationserver/connection Request Method : GET 2. Authorization Basic Authorization Username: Administrator Password: abcd1234!@#$ | Rest Console should configure successfully. |
|  |  |  |  |  | Step3 | Click the Send button. | Imagenow version should be displayed under Response Body tab in Rest Console. |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Manually Deletion of Connection session log from SQL sever | The purpose of this test is to verify that the all active connection should be closed from SQL server | Step1 | Open SQL server. | SQL server should open successfully. |
|  |  |  |  |  | Step2 | Execute the following query:   delete FROM [INOW6].[inuser].[IN\_LIC\_MON] where [INOW6].[inuser].[IN\_LIC\_MON].LIC\_TYPE = '93' | Query should be executed successfully. |
|  |  |  |  |  | Step3 | Check IN\_LIC\_MON table again(SELECT \*  FROM [INOW6].[inuser].[IN\_LIC\_MON]  where LIC\_TYPE='93') | The all opened connection session should be closed. |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Get drawer information from IS using Rest Console | The purpose of this test is to verify that the drawers of INOW should be displayed in Rest Console by IS | Step1 | Open Rest Console in Chrome | Rest Console should open successfully. |
|  |  |  |  |  | Step2 | Configure the Rest Console in a virtual machine as following: 1.Target: **Request URI:** http://localhost:8080/integrationserver/drawer Request Method : GET 2. Authorization Basic Authorization Username: Administrator Password: abcd1234!@#$ | Rest Console should configure successfully. |
|  |  |  |  |  | Step3 | Click the Send button. | Drawer information should be displayed under Response Body tab in Rest Console. |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Get extension and display of saved document of imagenow using EPIC solution | The purpose of this test is to verify that the saved document of imagenow should be displayed using EPIC solution(Get File Operation) | Step1 | Open Epic Test App. | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | 1.Configure Epic Test App  2.Select GetFile from the dropdown list and enter a valid document id in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and request imagenow file should display. |
|  |  |  |  |  | Step4 | Check IN\_LIC\_MON table again(SELECT \*  FROM [INOW6].[inuser].[IN\_LIC\_MON]  where LIC\_TYPE='93') | The all opened connection session should be closed.(IS\_Active should be 0) |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Get extension and display of saved document of imagenow using EPIC solution | The purpose of this test is to verify that the document saved in ImageNow should be displayed using EPIC solution(Get File Operation). | Step1 | Open Epic Test App. | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select GetFile from dropdown list. 3.Enter a valid document id in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and requested imagenow file should display. |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Error message should be displayed if file is not found in Imagenow and there is no active connection. | The purpose of this test is to verify that a proper error message is displayed if file is not found in Imagenow and no active connection exists in SQL server | Step1 | Open Epic Test App. | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | Configure Epic Test App  Select GetFile from dropdown and enter an invalid document id(not stored in Imagenow) in Document Key(s) text box | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Error message, like 'Document not found in ImageNow', should be displayed. |
|  |  |  |  |  | Step4 | Check IN\_LIC\_MON table again(SELECT \*  FROM [INOW6].[inuser].[IN\_LIC\_MON]  where LIC\_TYPE='93') | The all opened connection session should be closed.(IS\_Active should be 0) |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Get thumbnail image of a page using the EPIC solution. | The purpose of this test is to verify that the thumbnail image (png file) of a page is displayed using EPIC solution(Get thumbnail Operation). | Step1 | Open Epic Test App | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select Getthumbnail from the dropdown list and enter a valid document id in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and thumbnail png file of a page should display. |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | GetImagePage for allowing Epic to retrieve binary data for a particular page of an image/scan | The purpose of this test is to verify that the binary data for a particular page of an image/scan using EPIC Solution(GetImagePage operation) | Step1 | Open Epic Test App | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select GetImagePage from the dropdown list and enter a valid document id in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and binary data for a particular page of an image/scan should display. |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | MultiGetImageProperties for allowing Epic to get image properties of one or more images/scans | The purpose of this test is to verify that the image properties of one or more images/scans should be retrieved using EPIC Solution(MultiGetImageProperties operation) | Step1 | Open Epic Test App | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select MultiGetImageProperties from the dropdown list and enter a valid document id/ids in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and image properties of one or more images/scans should display |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | MultiFileQuery for allowing Epic to get file properties of one or more files | The purpose of this test is to verify that the image properties of one or more images/scans should be retrieved using EPIC Solution(MultiFileQuery operation) | Step1 | Open Epic Test App | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select MultiFileQuery from the dropdown list and enter a valid document id/ids in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and file properties of one or more files should display |

### System Test Cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Type | Github ID | Subject | Test Name | Test Description | Step Name | Description | Expected Result |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Check Successful Connection Rest Console with IS server | The purpose of this test is to verify that the Rest Console response log displays Imagenow version number while connecting with IS server. | Step1 | Open Rest Console in Google Chrome. | Rest Console should open successfully. |
|  |  |  |  |  | Step2 | Configure Rest Console in VM as following: 1.Target: **Request URI:** http://localhost:8080/integrationserver/connection Request Method : GET 2. Authorization Basic Authorization Username: Administrator Password: abcd1234!@#$ | Rest Console should configure successfully. |
|  |  |  |  |  | Step3 | Click the Send button. | Imagenow version should be displayed under Response Body tab in Rest Console. |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Manually Deletion of Connection session log from SQL sever | The purpose of this test is to verify that the all active connection should be closed from SQL server | Step1 | Open SQL server. | SQL server should open successfully. |
|  |  |  |  |  | Step2 | Execute the following query:   delete FROM [INOW6].[inuser].[IN\_LIC\_MON] where [INOW6].[inuser].[IN\_LIC\_MON].LIC\_TYPE = '93' | Query should be executed successfully. |
|  |  |  |  |  | Step3 | Check IN\_LIC\_MON table again(SELECT \*  FROM [INOW6].[inuser].[IN\_LIC\_MON]  where LIC\_TYPE='93') | The all opened connection session should be closed. |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Get drawer information from IS using Rest Console | The purpose of this test is to verify that the drawers of INOW should be displayed in Rest Console by IS | Step1 | Open Rest Console in Chrome | Rest Console should open successfully. |
|  |  |  |  |  | Step2 | Configure the Rest Console in a virtual machine as following: 1.Target: **Request URI:** http://localhost:8080/integrationserver/drawer Request Method : GET 2. Authorization Basic Authorization Username: Administrator Password: abcd1234!@#$ | Rest Console should configure successfully. |
|  |  |  |  |  | Step3 | Click the Send button. | Drawer information should be displayed under Response Body tab in Rest Console. |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Get extension and display of saved document of imagenow using EPIC solution | The purpose of this test is to verify that the saved document of imagenow should be displayed using EPIC solution(Get File Operation) | Step1 | Open Epic Test App. | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | 1.Configure Epic Test App  2.Select GetFile from the dropdown list and enter a valid document id in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and request imagenow file should display. |
|  |  |  |  |  | Step4 | Check IN\_LIC\_MON table again(SELECT \*  FROM [INOW6].[inuser].[IN\_LIC\_MON]  where LIC\_TYPE='93') | The all opened connection session should be closed.(IS\_Active should be 0) |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Get extension and display of saved document of imagenow using EPIC solution | The purpose of this test is to verify that the document saved in ImageNow should be displayed using EPIC solution(Get File Operation). | Step1 | Open Epic Test App. | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select GetFile from dropdown list. 3.Enter a valid document id in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and requested imagenow file should display. |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Error message should be displayed if file is not found in Imagenow and there is no active connection. | The purpose of this test is to verify that a proper error message is displayed if file is not found in Imagenow and no active connection exists in SQL server | Step1 | Open Epic Test App. | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | Configure Epic Test App  Select GetFile from dropdown and enter an invalid document id(not stored in Imagenow) in Document Key(s) text box | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Error message, like 'Document not found in ImageNow', should be displayed. |
|  |  |  |  |  | Step4 | Check IN\_LIC\_MON table again(SELECT \*  FROM [INOW6].[inuser].[IN\_LIC\_MON]  where LIC\_TYPE='93') | The all opened connection session should be closed.(IS\_Active should be 0) |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | Get thumbnail image of a page using the EPIC solution. | The purpose of this test is to verify that the thumbnail image (png file) of a page is displayed using EPIC solution(Get thumbnail Operation). | Step1 | Open Epic Test App | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select Getthumbnail from the dropdown list and enter a valid document id in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and thumbnail png file of a page should display. |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | GetImagePage for allowing Epic to retrieve binary data for a particular page of an image/scan | The purpose of this test is to verify that the binary data for a particular page of an image/scan using EPIC Solution(GetImagePage operation) | Step1 | Open Epic Test App | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select GetImagePage from the dropdown list and enter a valid document id in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and binary data for a particular page of an image/scan should display. |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | MultiGetImageProperties for allowing Epic to get image properties of one or more images/scans | The purpose of this test is to verify that the image properties of one or more images/scans should be retrieved using EPIC Solution(MultiGetImageProperties operation) | Step1 | Open Epic Test App | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select MultiGetImageProperties from the dropdown list and enter a valid document id/ids in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and image properties of one or more images/scans should display |
| Manual | US20251 | Subject\Solutions\Future\DNBSN\2013\Q1\Sprint5 | MultiFileQuery for allowing Epic to get file properties of one or more files | The purpose of this test is to verify that the image properties of one or more images/scans should be retrieved using EPIC Solution(MultiFileQuery operation) | Step1 | Open Epic Test App | Epic Test App should open successfully. |
|  |  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select MultiFileQuery from the dropdown list and enter a valid document id/ids in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and file properties of one or more files should display |

# Coding

## Complete Project Coding

Code

## 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 new data Model in the Destination DB(s) will be migrated from the existing systems source 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Type | Github ID | Test Name | Test Description | Step Name | Description | Expected Result |
| Manual | US20251 | Check Successful Connection Rest Console with IS server | The purpose of this test is to verify that the Rest Console response log displays Imagenow version number while connecting with IS server. | Step1 | Open Rest Console in Google Chrome. | Rest Console should open successfully. |
|  |  |  |  | Step2 | Configure Rest Console in VM as following: 1.Target: **Request URI:** http://localhost:8080/integrationserver/connection Request Method : GET 2. Authorization Basic Authorization Username: Administrator Password: abcd1234!@#$ | Rest Console should configure successfully. |
|  |  |  |  | Step3 | Click the Send button. | Imagenow version should be displayed under Response Body tab in Rest Console. |
| Manual | US20251 | Manually Deletion of Connection session log from SQL sever | The purpose of this test is to verify that the all active connection should be closed from SQL server | Step1 | Open SQL server. | SQL server should open successfully. |
|  |  |  |  | Step2 | Execute the following query:   delete FROM [INOW6].[inuser].[IN\_LIC\_MON] where [INOW6].[inuser].[IN\_LIC\_MON].LIC\_TYPE = '93' | Query should be executed successfully. |
|  |  |  |  | Step3 | Check IN\_LIC\_MON table again(SELECT \*  FROM [INOW6].[inuser].[IN\_LIC\_MON]  where LIC\_TYPE='93') | The all opened connection session should be closed. |
| Manual | US20251 | Get drawer information from IS using Rest Console | The purpose of this test is to verify that the drawers of INOW should be displayed in Rest Console by IS | Step1 | Open Rest Console in Chrome | Rest Console should open successfully. |
|  |  |  |  | Step2 | Configure the Rest Console in a virtual machine as following: 1.Target: **Request URI:** http://localhost:8080/integrationserver/drawer Request Method : GET 2. Authorization Basic Authorization Username: Administrator Password: abcd1234!@#$ | Rest Console should configure successfully. |
|  |  |  |  | Step3 | Click the Send button. | Drawer information should be displayed under Response Body tab in Rest Console. |
| Manual | US20251 | Get extension and display of saved document of imagenow using EPIC solution | The purpose of this test is to verify that the saved document of imagenow should be displayed using EPIC solution(Get File Operation) | Step1 | Open Epic Test App. | Epic Test App should open successfully. |
|  |  |  |  | Step2 | 1.Configure Epic Test App  2.Select GetFile from the dropdown list and enter a valid document id in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and request imagenow file should display. |
|  |  |  |  | Step4 | Check IN\_LIC\_MON table again(SELECT \*  FROM [INOW6].[inuser].[IN\_LIC\_MON]  where LIC\_TYPE='93') | The all opened connection session should be closed.(IS\_Active should be 0) |
| Manual | US20251 | Get extension and display of saved document of imagenow using EPIC solution | The purpose of this test is to verify that the document saved in ImageNow should be displayed using EPIC solution(Get File Operation). | Step1 | Open Epic Test App. | Epic Test App should open successfully. |
|  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select GetFile from dropdown list. 3.Enter a valid document id in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and requested imagenow file should display. |
| Manual | US20251 | Error message should be displayed if file is not found in Imagenow and there is no active connection. | The purpose of this test is to verify that a proper error message is displayed if file is not found in Imagenow and no active connection exists in SQL server | Step1 | Open Epic Test App. | Epic Test App should open successfully. |
|  |  |  |  | Step2 | Configure Epic Test App  Select GetFile from dropdown and enter an invalid document id(not stored in Imagenow) in Document Key(s) text box | Epic Test App should configure successfully. |
|  |  |  |  | Step3 | Click the Submit button. | Error message, like 'Document not found in ImageNow', should be displayed. |
|  |  |  |  | Step4 | Check IN\_LIC\_MON table again(SELECT \*  FROM [INOW6].[inuser].[IN\_LIC\_MON]  where LIC\_TYPE='93') | The all opened connection session should be closed.(IS\_Active should be 0) |
| Manual | US20251 | Get thumbnail image of a page using the EPIC solution. | The purpose of this test is to verify that the thumbnail image (png file) of a page is displayed using EPIC solution(Get thumbnail Operation). | Step1 | Open Epic Test App | Epic Test App should open successfully. |
|  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select Getthumbnail from the dropdown list and enter a valid document id in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and thumbnail png file of a page should display. |
| Manual | US20251 | GetImagePage for allowing Epic to retrieve binary data for a particular page of an image/scan | The purpose of this test is to verify that the binary data for a particular page of an image/scan using EPIC Solution(GetImagePage operation) | Step1 | Open Epic Test App | Epic Test App should open successfully. |
|  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select GetImagePage from the dropdown list and enter a valid document id in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and binary data for a particular page of an image/scan should display. |
| Manual | US20251 | MultiGetImageProperties for allowing Epic to get image properties of one or more images/scans | The purpose of this test is to verify that the image properties of one or more images/scans should be retrieved using EPIC Solution(MultiGetImageProperties operation) | Step1 | Open Epic Test App | Epic Test App should open successfully. |
|  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select MultiGetImageProperties from the dropdown list and enter a valid document id/ids in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and image properties of one or more images/scans should display |
| Manual | US20251 | MultiFileQuery for allowing Epic to get file properties of one or more files | The purpose of this test is to verify that the image properties of one or more images/scans should be retrieved using EPIC Solution(MultiFileQuery operation) | Step1 | Open Epic Test App | Epic Test App should open successfully. |
|  |  |  |  | Step2 | 1.Configure Epic Test App.  2.Select MultiFileQuery from the dropdown list and enter a valid document id/ids in Document Key(s) text box. | Epic Test App should configure successfully. |
|  |  |  |  | Step3 | Click the Submit button. | Response Time log with file extension and file properties of one or more files should display |

## 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 with very 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)b2 months

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)1.12 PM

= 11 PM

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

= 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

* <http://en.wikipedia.org>
* <http://msdn.microsoft.com/en-us/>
* <http://www.microsoft.com/en-us/default.aspx>
* <http://www.codeplex.com/>
* <http://stackoverflow.com/>
* <http://www.codeguru.com/>
* <http://www.w3schools.com>
* <http://blogs.technicise.com/>
* [www.mysql.org](http://www.mysql.org)

# Appendices

## IDE Used:

### Visual Studio 2010

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).

## Front End - WPF (Windows Presentation Framework)

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.

## 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. Altough 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)

The .NET 4 Framework is Microsoft's platform for building applications that have visually stunning user experiences, seamless and secure communication, and the ability to model a range of business processes. The .Net Framework consists of:

Common Language Runtime – provides an abstraction layer over the operating system

Base Class Libraries – pre-built code for common low-level programming tasks

Development frameworks and technologies – reusable, customizable solutions for larger programming tasks.

The framework's Base Class Library provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. The class library is used by programmers, who combine it with their own code to produce applications.

## 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

## ide for Database –MySQL workbench

MySQL Workbench is a visual database design tool that integrates SQL development,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.

## Programming Language (C#)

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.

## 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.*

## Windows Mobile 6 Professional SDK

Windows Mobile is a mobile operating system developed by Microsoft that was used in smartphones and mobile devices. This features a suite of basic applications developed with the Microsoft Windows API. It is designed to be somewhat similar to desktop versions of Windows, feature-wise and aesthetically. Additionally, third-party software development is available for Windows Mobile, and software applications can be purchased via the Windows Marketplace for Mobile. Windows Mobile 6.1 was announced April 1, 2008. It is a minor upgrade to the existing Windows Mobile 6 platform which brings with it various performance enhancements, a redesigned Home screen featuring horizontal tiles that expand on clicking to display more information, although this new home screen is featured only on Windows Mobile Standard edition. This feature was inexplicably left out of the Professional edition. Several other changes such as threaded SMS, full page zooming in Internet Explorer and 'Domain Enroll' have also been added, along with a "mobile" version of the Microsoft OneNote program and an interactive "Getting Started" wizard. Windows Mobile 6.1 also featured improved bandwidth efficiency in its push-email protocol "Activesync" of "up to 40%", this reduced data usage was the cause of considerably improved battery life in many devices.

# Glossary.