T-Notes

DBMS Project

By: Brainstrome

Project Members

Jessica Byrd

Nghia Dang

Luis Gonzalez

Bryce Harrington

Phil Huddleston

Bhusan Paudel

Chandlor Ratcliffe

Kenneth Roberts

Hamza Syed

1. INTRODUCTION

As a college student, it is not uncommon to have a mess of notes scattered throughout your computer in multiple directories and sub-directories. T Notes is designed to help anyone organize and secure their notes in one place. The application allows for users to keep track of notes through multiple classes and semesters without the hassle of traditional note taking practices.

2. SYSTEM STUDY

2.1. Proposed system

T Notes provides the user with all the familiar functionality they would expect from such an application. Right off the bat, your notes are secured behind individual logins protected with user passwords. This allows for multiple users to share the application on one system or have an added layer of security. The application allows for the user to organize their notes by courses, which are defined by the course name, subject, semester, year, and the professor. T Notes provides the user with the ability to delete unwanted notes in order to clear clutter or modify existing notes to add information they might have missed the first time. Knowing the all too familiar structure of classes, T Notes furthermore allows for the user to define chapters and sections, so you can be sure to follow the professor with added ease. No more flipping through a traditional notebook, trying to keep up with that speedy professor! To add to the ease of going back and searching your notes, But wait, there's more! The body of the notes is used to automatically generate as many keywords as necessary to facilitate quick easy searches for the user. T Notes allows you to write a short summary in relation to each note you take, this way you can preview the notes before opening them and checking them out.

- **Ability to search notes by keyboards**
- **Records are efficiently maintained by DBMS**
- **A single machine allows access for multiple users**
- **Concrete dashboard gives quick access to all functions**
- **Hashing is added to the passwords for an extra added layer of security**

2.2. Introduction about the front end

This proposed software was developed using the technology from Microsoft called Visual Studio 2017 CE. It is the software that was used in order to sync, build and connect branches from all team members with GitHub as well as to design the user interface.

- Visual Studio was used for its highly interactive Graphical User Interface. It also partially generated code as it was built.
- Graphics were used from stock photography & photoshop to edit the stock photography
- Inkscape was used to create the T Notes logo from scratch
- The orange hue throughout the project RGB was used from the Sam Houston State University website as well as the SHSU logo

2.3. Introduction about the back end

Back end extensively utilized C# for its object-oriented design aiding in fast and streamlined development. C# was created by Microsoft to make software engineering for its windows environment developer friendly.

- C# provides essential feature rich libraries for concrete and rapid development
- Its object-oriented design allows for efficient and easy coding
- C# gives developers ample functionality for database interaction using the MySQL library

3. SYSTEM SPECIFICATION

3.1. Hardware Requirements:

- 512 MB RAM or above
- 5 GB Hard Disk or above
- Extended peripherals: Mouse, Keyboard, Monitor

3.2. Software Requirements:

- Operating system : Windows 7, 8, 10
- Front end : Visual Studio 2017
- Platform : MYSQL instance
- IDE : Visual Studio 2017 CE
- Database : MYSQL Workbench 8.0 CE
- Back end : C#

4. DATABASE DESIGN

4.1. Conceptual design

MySQL Workbench 8.0 CE was used to maximize the database performance and scalability. It provided the data modeling and SQL development for the server configuration on the backend for a single integrated environment.

- ***SQL Development***: Enables you to create and manage connections to database servers. As well as enabling you to configure connection parameters. MySQL Workbench provides the capability to execute SQL queries on the database.

- ***Data Modeling***: Enables you to create models of your database schema graphically, reverse and forward engineer between a schema and a live database, and edit all aspects of your database using the comprehensive Table Editor. The Table Editor provides easy-to-use facilities for editing Tables, Columns, Indexes, Triggers, Partitioning, Options, Inserts and Privileges, Routines and Views.

- ***Server Administration***: Enables you to create and administer server instances.

- ***Data Migration***: Allows you to migrate from Microsoft Visual Studio 2017

a.) **Requirement Analysis:**

List of Entities:

- User
- Note
- Course
- Keyword
- is_taking
- contains

List of attributes:

User:

1. user_id
2. username
3. password
4. first_name
5. last_name

Note:

1. note_id
2. note_title
3. chapter
4. section
5. summary
6. date
7. notes

Course:

1. course_id
2. course_name
3. subject
4. prof
5. semester
6. year

Keyword:

1. keyword_id
2. keyword

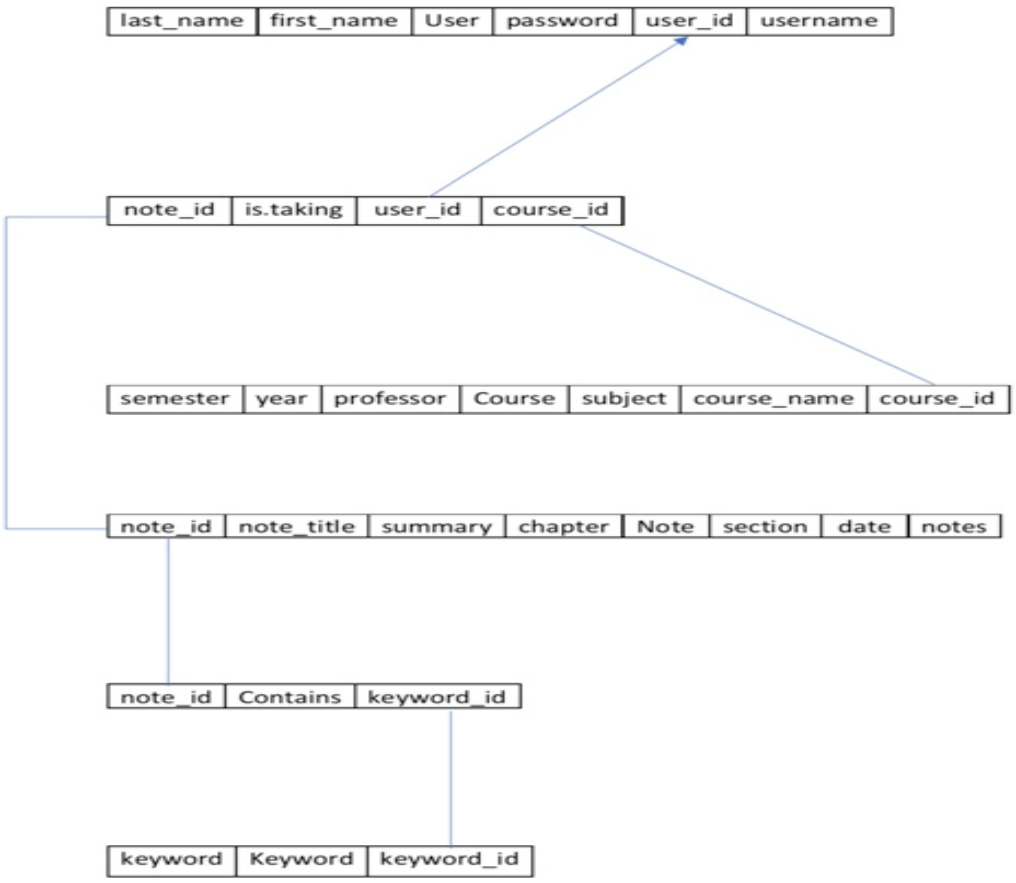
is_taking:

1. course_id
2. user_id
3. note_id

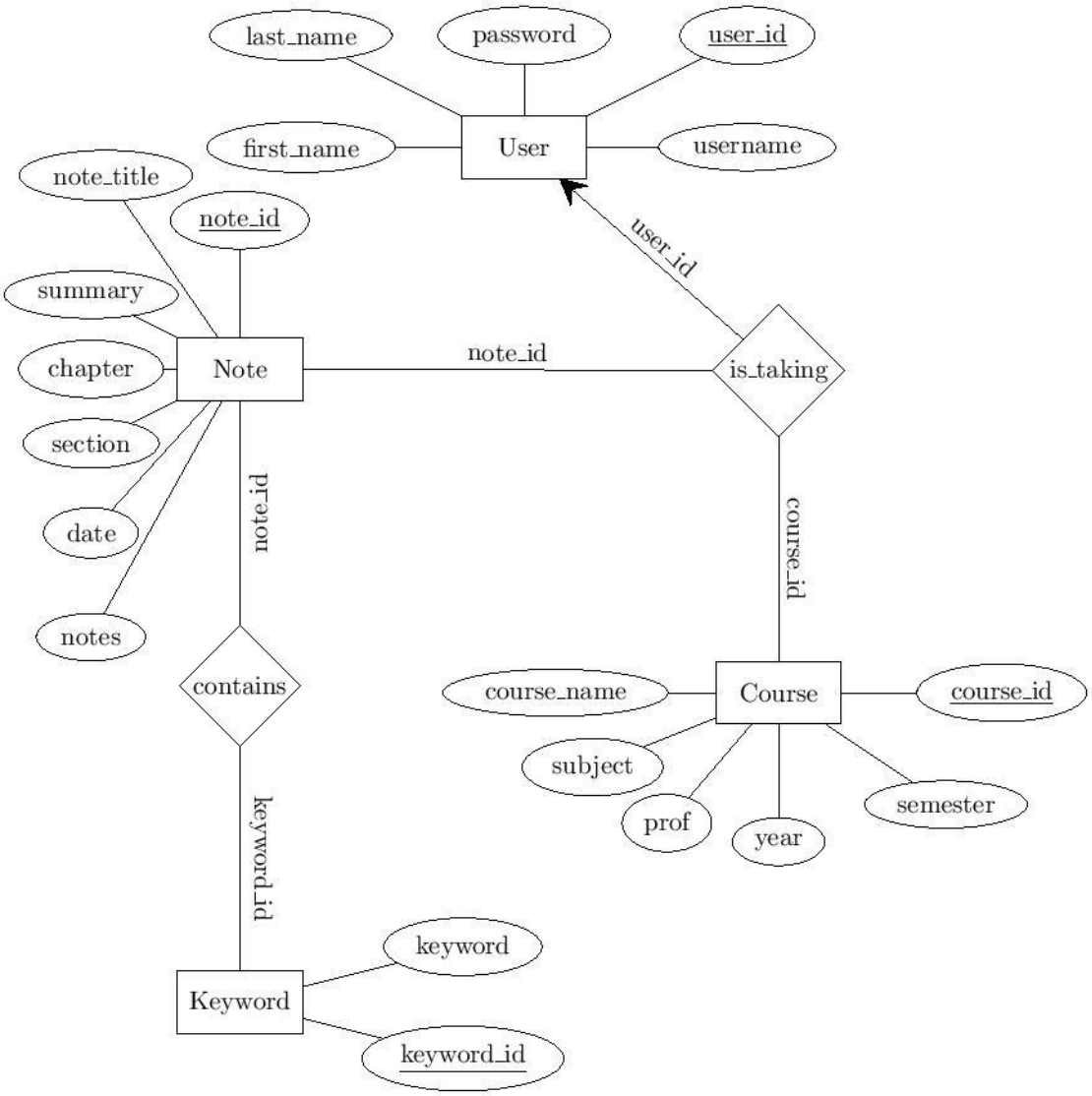
contains:

1. note_id
2. keyword_id

Schema:


last_name
first_name
User
password
user_id
username
note_id
is.taking
user_id
course_id
semester
year
professor
Course
subject
course_name
course_id
note_id
note_title
summary
chapter
Note
section
date
notes
note_id
Contains
keyword_id
keyword
Keyword
keyword_id

b.) ER-MODEL:


last_name
password
user_id
first_name
User
username
note_title
note_id
user_id
summary
chapter
Note
note_id
is_taking
section
note_id
course_id
date
notes
contains
course_name
Course
course_id
subject
prof
year
semester
keyword_id
keyword
Keyword
keyword_id

4.2. Logical design:

1. Note: Keeps record of notes.

```
CREATE TABLE `note` (
  `note_id` int(11) NOT NULL,
  `note_title` varchar(45) DEFAULT NULL,
  `chapter` int(11) DEFAULT NULL,
  `Section` int(11) DEFAULT NULL,
  `Summary` varchar(1000) DEFAULT NULL,
  `Date` date NOT NULL,
  `notes` longtext NOT NULL,
  PRIMARY KEY (`note_id`),
  UNIQUE KEY `note_id` (`note_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

.S.no	Field name	Data type	Description	Constraints
1	note_id	int(11)	Utilized for the Primary Key	Primary key
2	note_title	varchar(45)	Title of the note	
3	chapter	int(11)	Signifies the note chapter	
4	section	int(11)	Signifies the chapter section	
5	summary	varchar(1000)	Summary of the note	
6	date	date	Date the note was taken	
7	notes	longtext	Stores the note with a data limit of 4GB	

2. User: Keeps record of individual user information.

```
CREATE TABLE `user` (
  `user_id` int(11) NOT NULL,
  `username` varchar(45) NOT NULL,
  `password` varchar(45) NOT NULL,
  `first_name` varchar(45) DEFAULT NULL,
  `last_name` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`user_id`),
  UNIQUE KEY `username_UNIQUE` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

S.no	Field name	Data type	Description	Constraints
1	user_id	int(11)	Utilized for the Primary Key	Primary key
2	user_name	varchar(45)	Unique user name to identify the user	username_UNIQUE
3	password	varchar(45)	Password for each user account	
4	first_name	varchar(45)	Define users first name	
5	last_name	varchar(45)	Define users last name	

3. Course: Keeps track of user defined courses.

```
_CREATE TABLE `course` (
  `course_id` int(11) NOT NULL,
  `course_name` varchar(45) DEFAULT NULL,
  `subject` varchar(45) DEFAULT NULL,
  `prof` varchar(45) DEFAULT NULL,
  `semester` varchar(45) DEFAULT NULL,
  `year` int(11) DEFAULT NULL,
  PRIMARY KEY (`course_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

S.no	Field name	Data type	Description	Constraints
1	course_id	int(11)	Utilized for the Primary Key	Primary key
2	course_name	varchar(45)	User defined course name to identify course	
3	subject	varchar(45)	User defined subject to categorize the courses	
4	prof	varchar(45)	User defined professor to categorize the courses	
5	semester	varchar(45)	User defined semester to categorize the courses	
6	year	int(11)	User defined year to categorize the courses	

4. Keyword: Auto generated keywords for the notes.

```
CREATE TABLE `keyword` (
  `keyword_id` int(11) NOT NULL,
  `keyword` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`keyword_id`),
  UNIQUE KEY `keyword_id` (`keyword_id`),
  UNIQUE KEY `keyword` (`keyword`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

S.no	Field name	Data type	Description	Constraints
1	keyword_id	int(11)	Utilized for the Primary Key	Primary key
2	keyword	varchar(45)	Backend generated keywords from the notes	keyword

5. is_taking: Defines the relationship between the user, notes and courses.

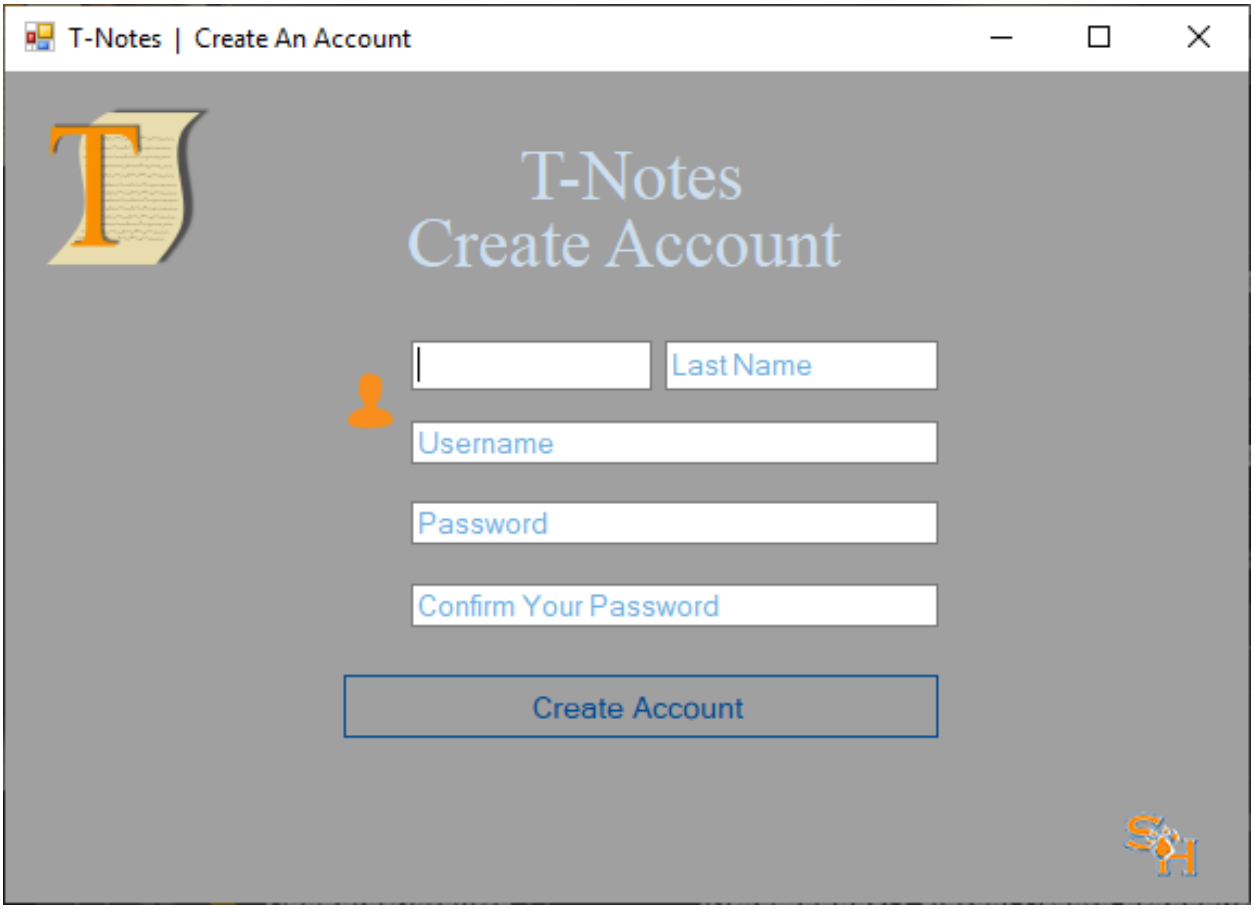
```
CREATE TABLE `is_taking` (
  `course_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `note_id` int(11) NOT NULL,
  PRIMARY KEY (`course_id`,`user_id`,`note_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

S.no	Field name	Data type	Description	Constraints
1	course_id	int(11)	Primary key to form relationships	Primary key
2	user_id	int(11)	Primary key to form relationships	Primary key
3	note_id	int(11)	Primary key to form relationships	Primary key

5. IMPLEMENTATION

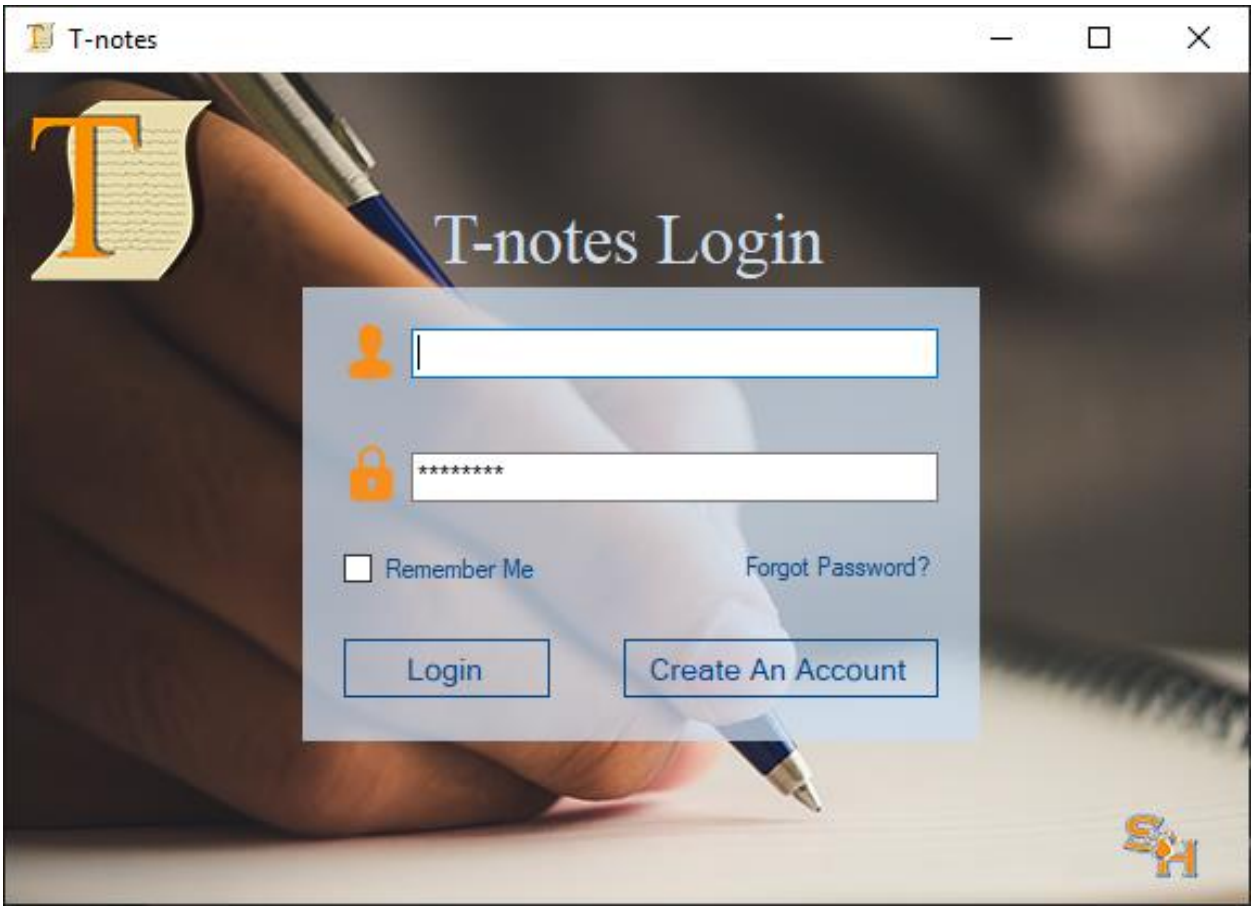
Briefly introduce each function of your program with screen shot to demonstrate (instruction manual of your program)

Create an account allows for the user to create a T-Notes account



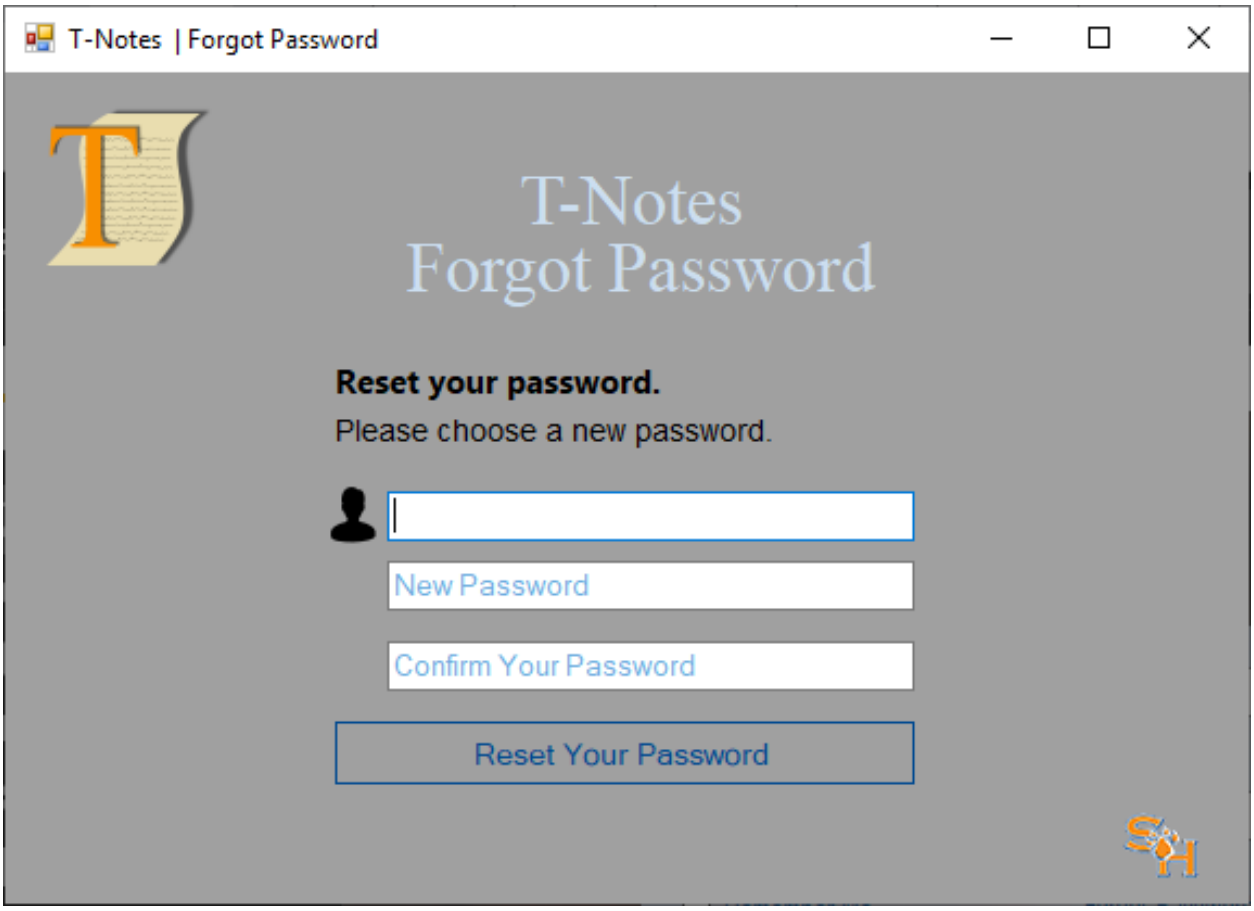
Login

Allows the user to Login to T-Notes

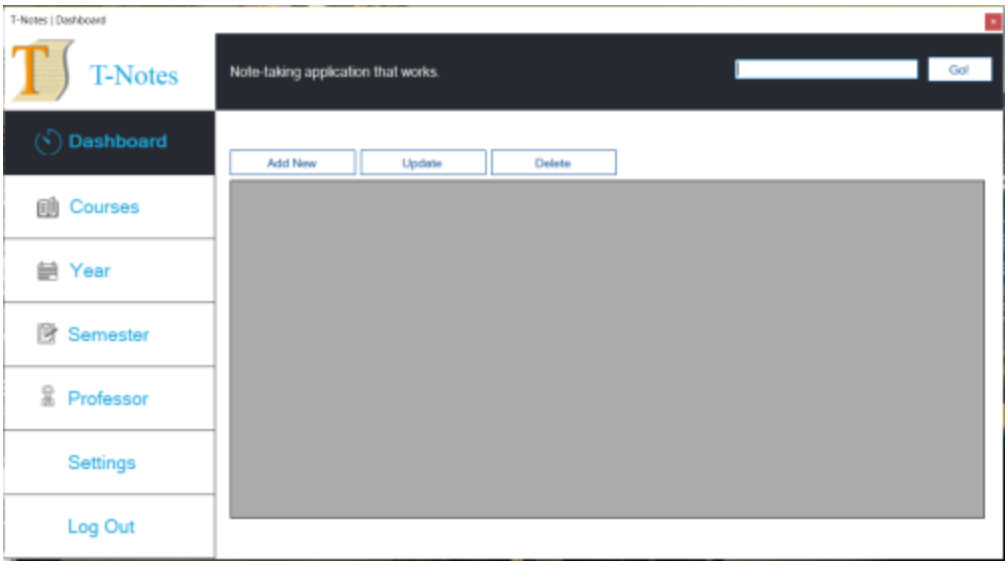


Forgot Password

Enables a user to reset their password



Dashboard – after login it takes the user to the dashboard



User Settings – Accessed via Dashboard

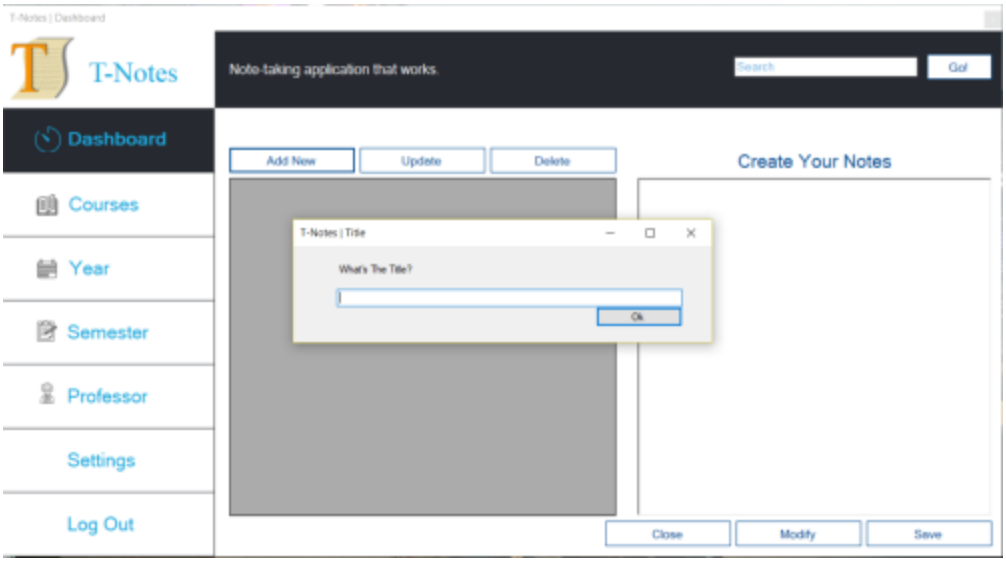
Ability to change their username and password after login

User Settings – Accessed via Dashboard

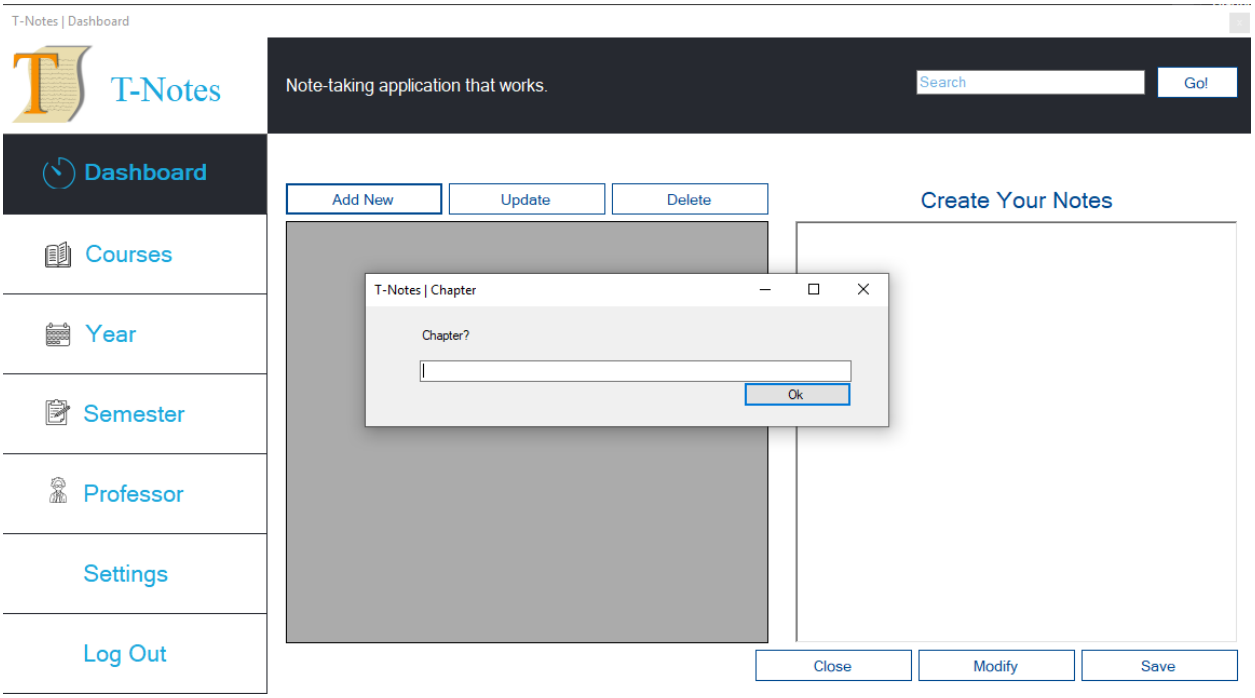
Ability to delete a user

ADD Function

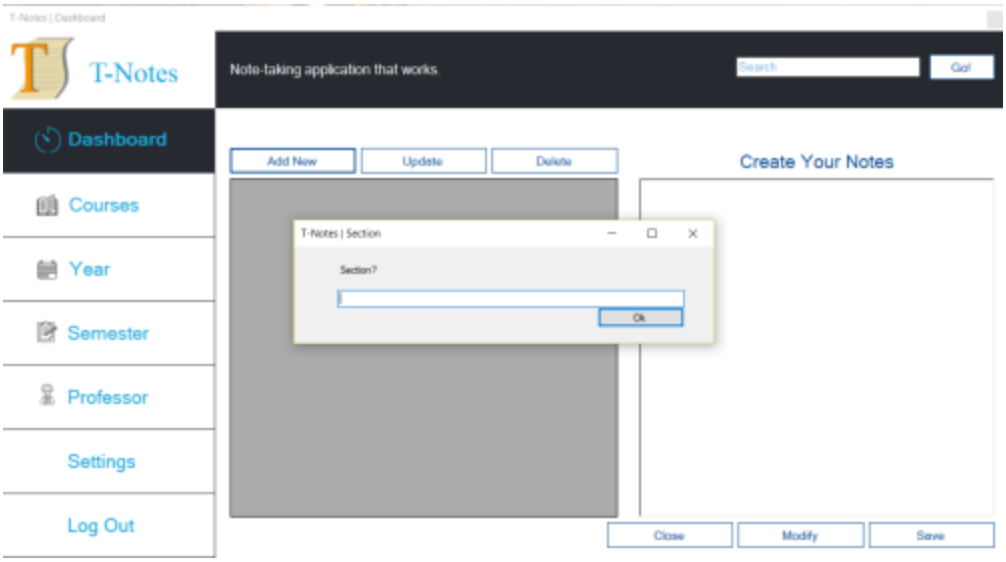
Ability to Add a note starting with title.



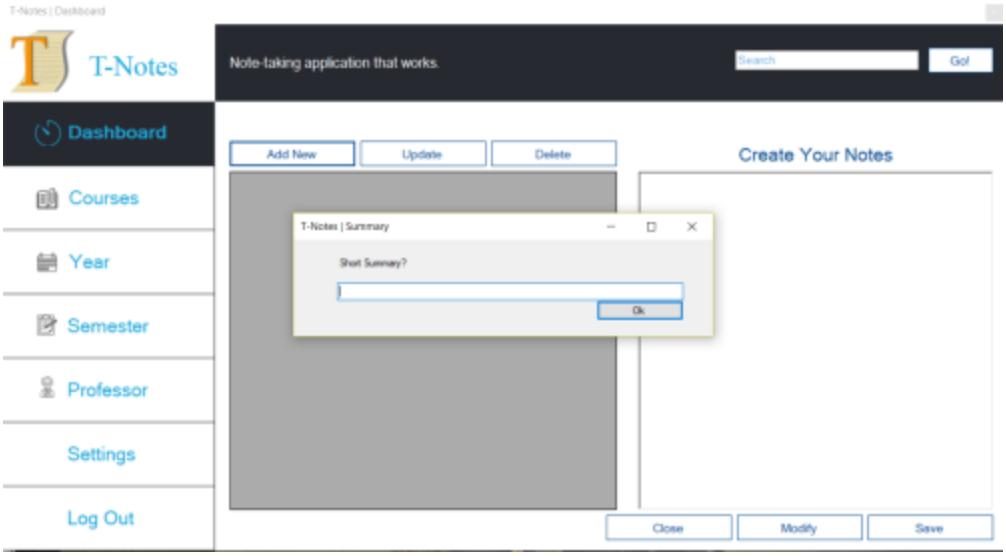
Next define the Chapter



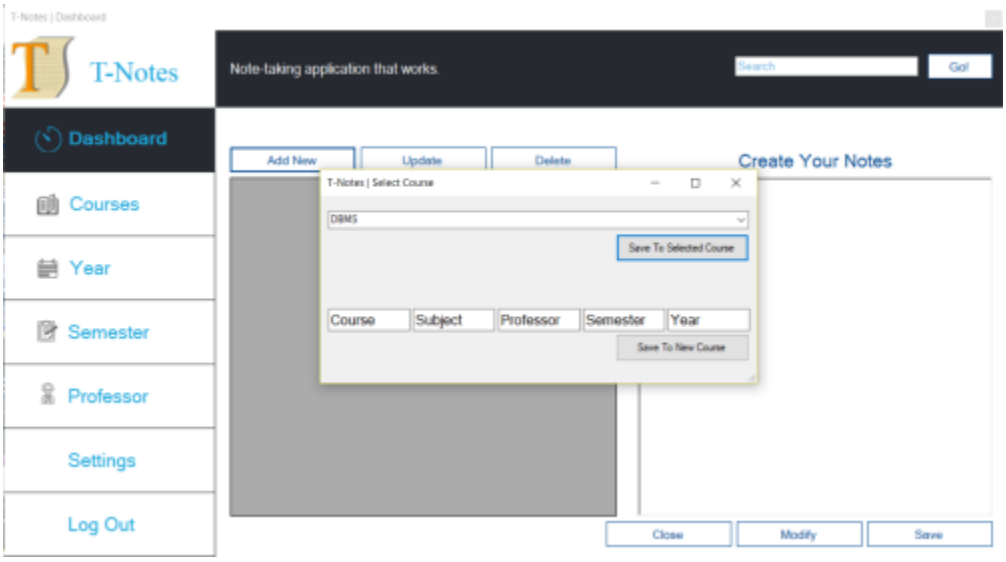
Define the section



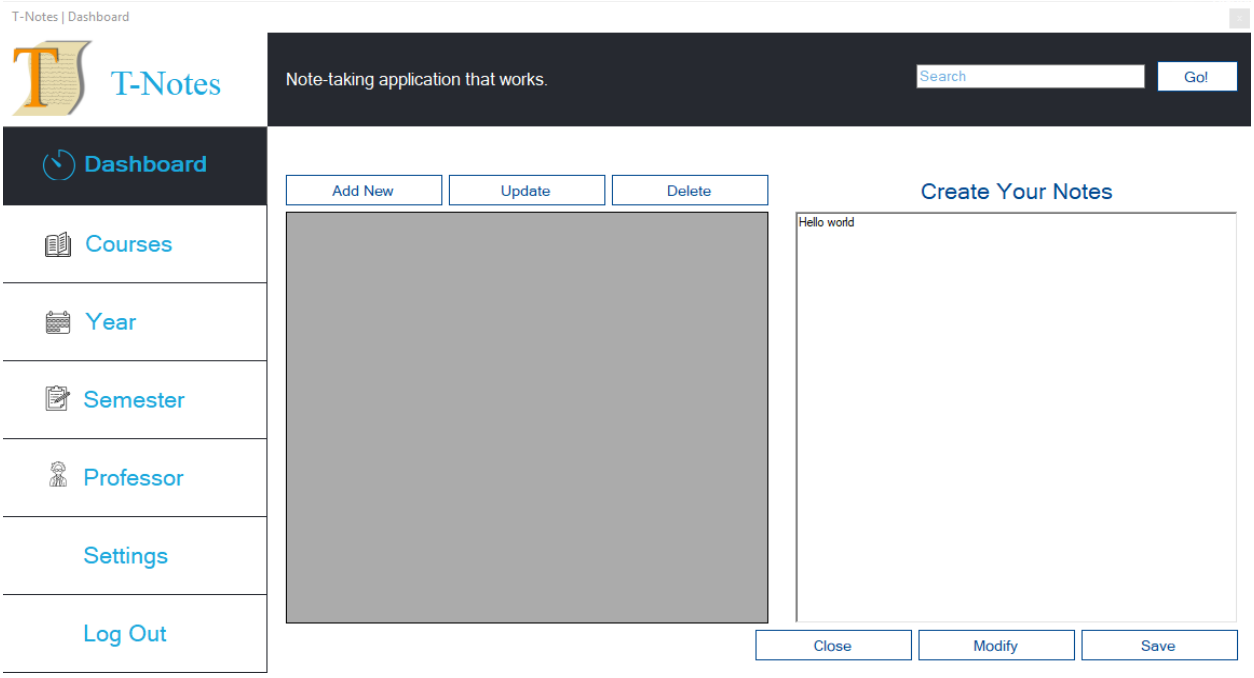
Define the summary of the note



Select the course you would like to save it to

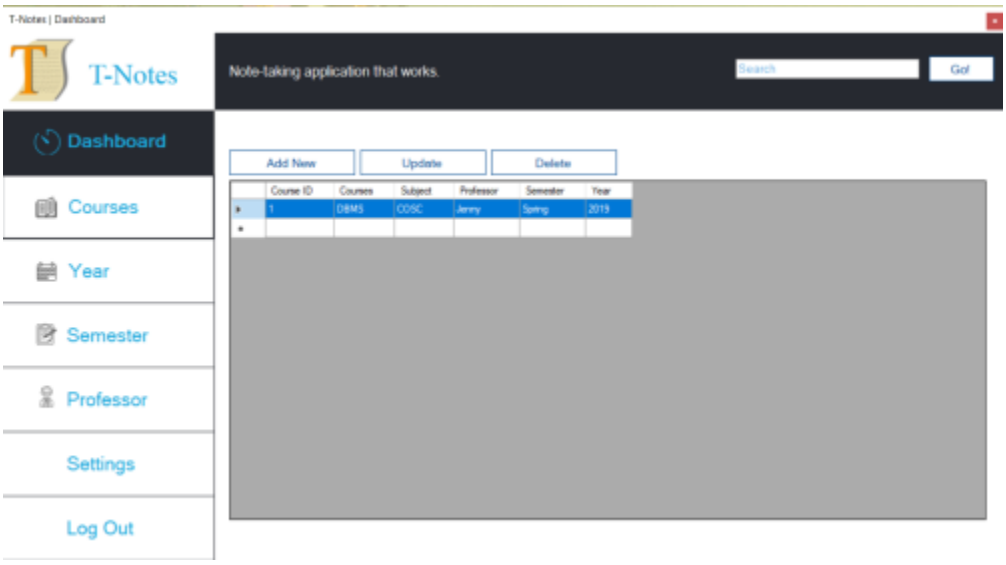


Now you can type out your note and save it

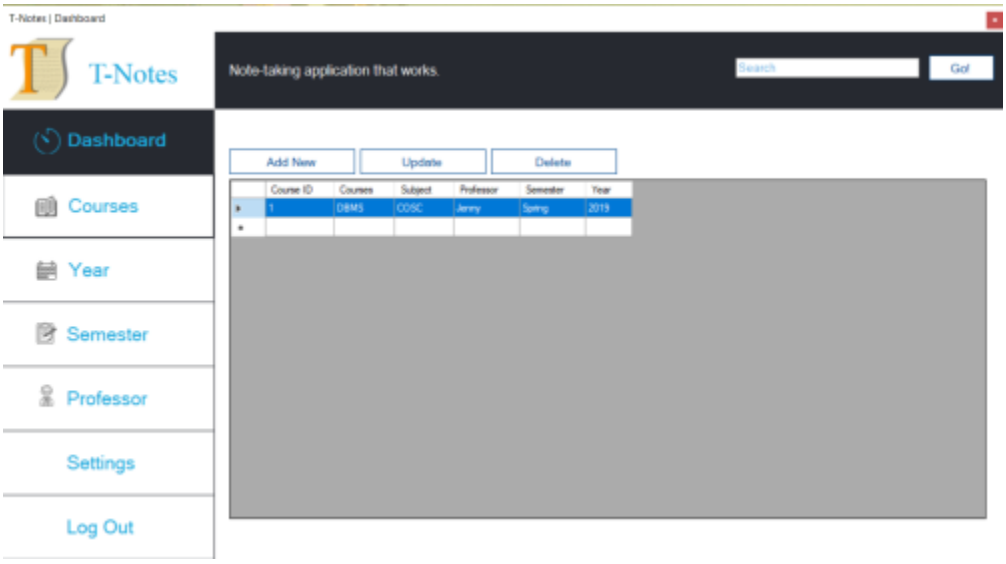


Function via Display all courses

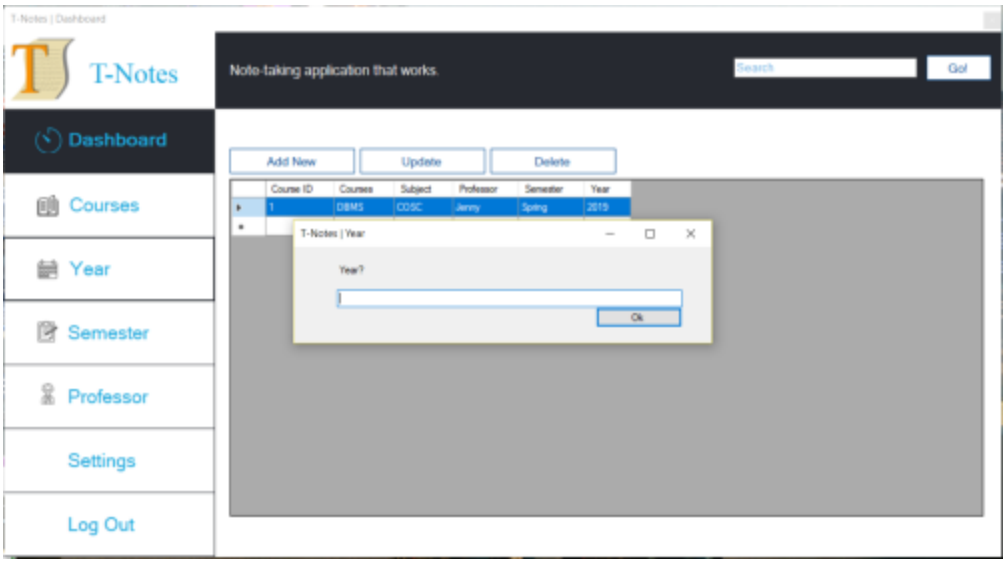
Ability to Add a note, Update a note and Delete a note



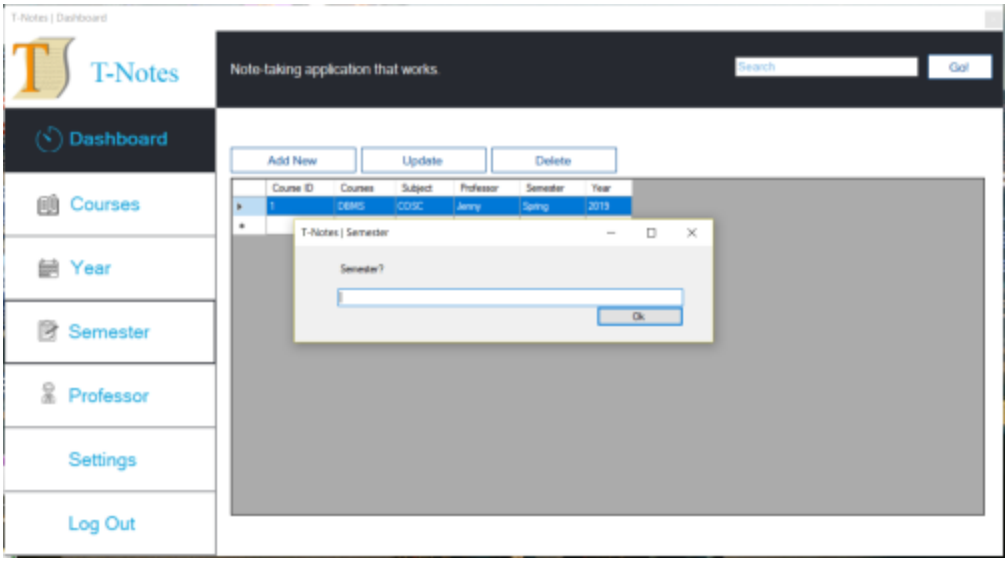
Display all available courses



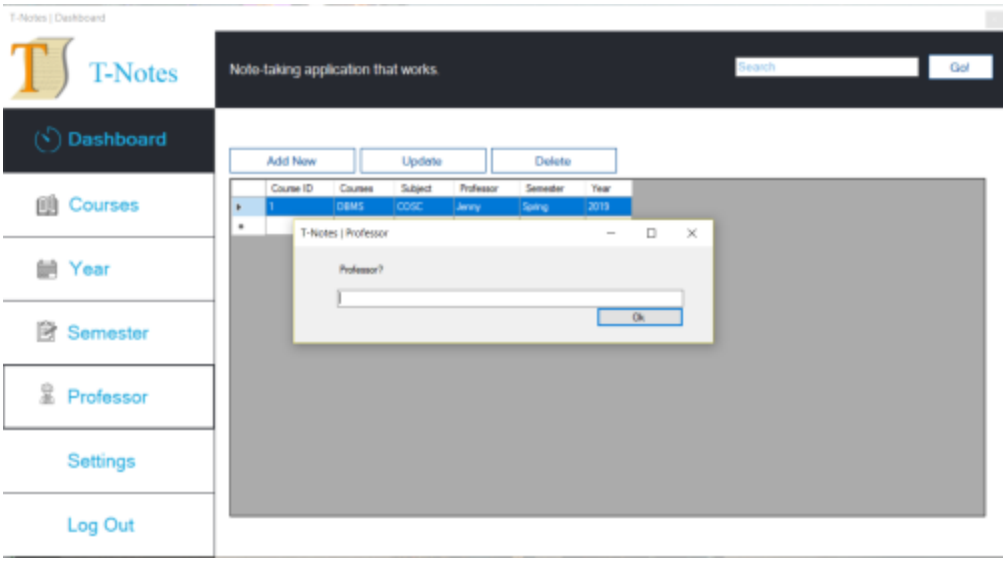
Display all courses by year



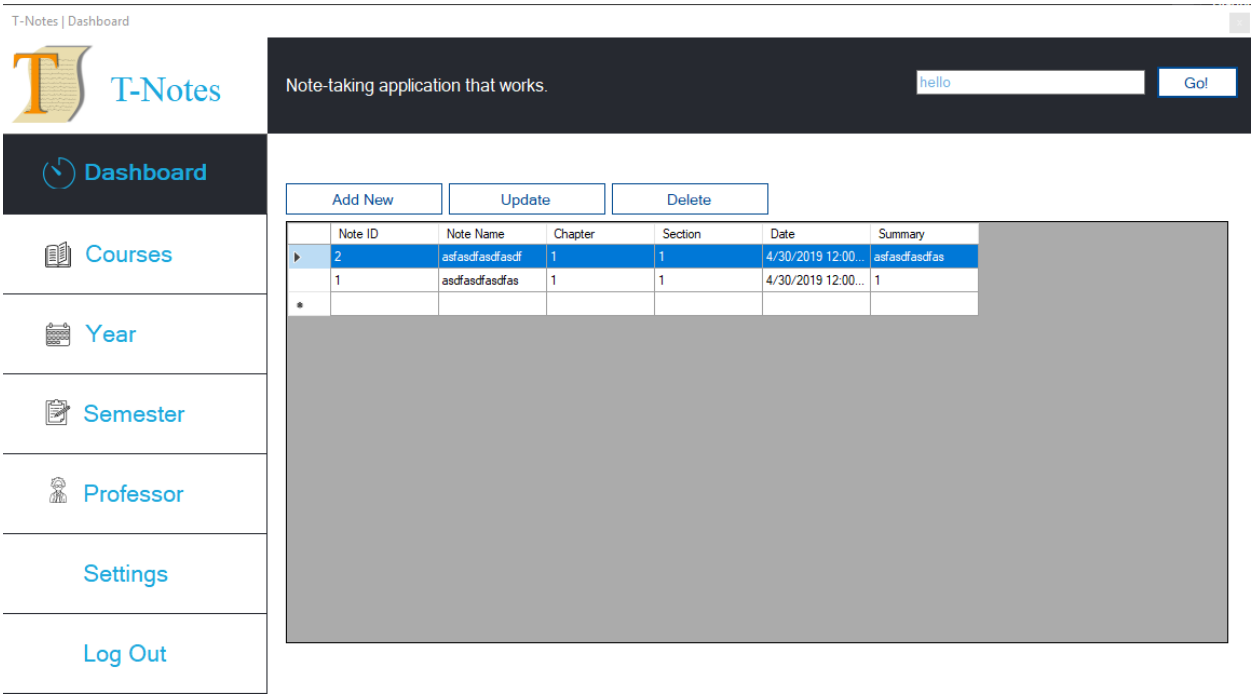
Display all courses by semester



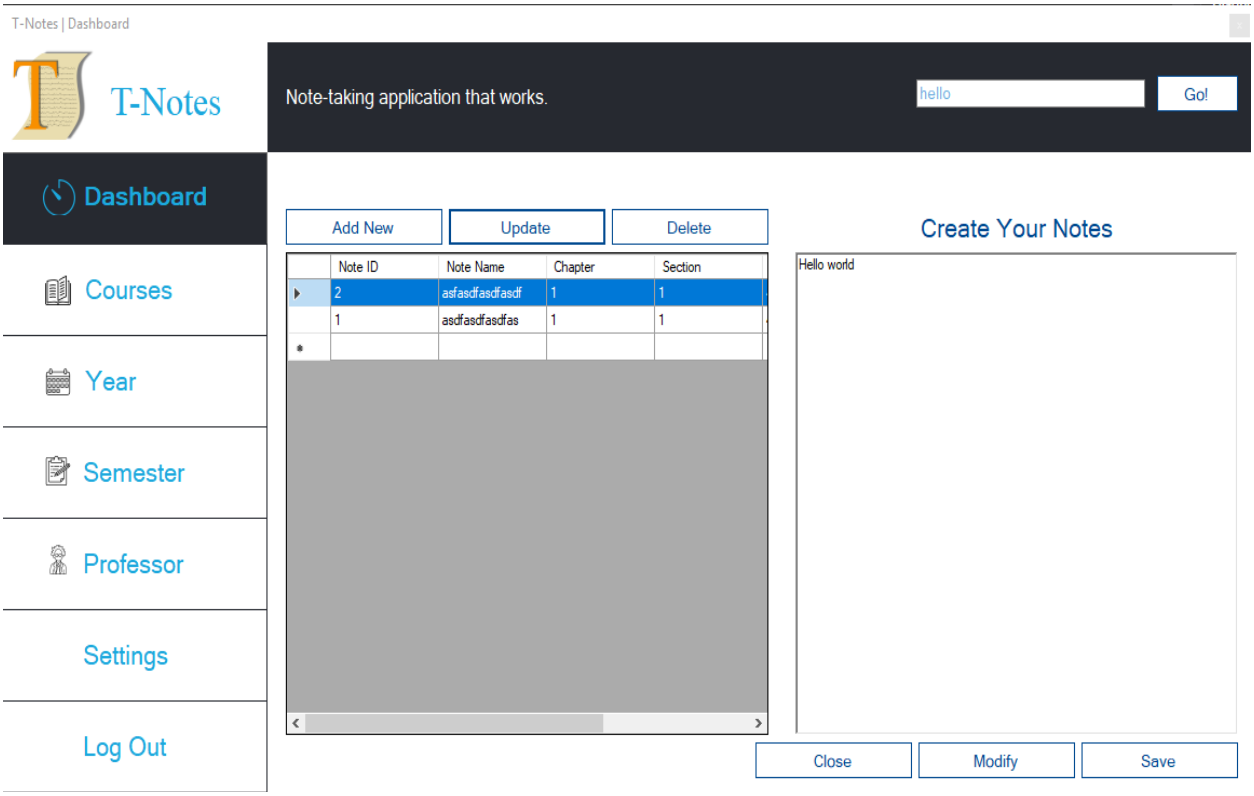
Display all courses by professor



Search a Keyword



Open notes from the Database



6. ADDITIONAL COMMENTS (optional)

Work more on different features to develop T-notes into a windows universal application.

Implement a web-based interface to allow access from anywhere.

Utilize the keyword entity to create flashcards for the user.

MD – 5 hashing on passwords

7. SOURCECODE

Attach you source code – Included in a Zip file in this email.