PinNote: Desktop Note Application

Seden Acar

Computer Programming, Girne Amerikan Üniversitesi

CP 210 - Term Project

Advisor

Arif Sarı

Girne - 2024

ABSTRACT

With the development of technology, traditional methods fell by the wayside. This development affected many sectors as well as people's note habits. Agendas were replaced by note applications. Although these note applications made people's work easier, the comprehensive ones were difficult to learn, and the interfaces of the simple ones were not neat enough. This situation is difficult for people who have not kept up with the speed of technology. My work PinNote, which I prepared as a solution to this problem, is a desktop note application that appeals to a wide audience thanks to its simple use and stylish interface. Thanks to the account creation feature in this note application, you can access your notes and reminders from different computers where the application is installed. You can organize your notes and reminders, share them with your friends, and view your friends' profiles. I discussed the methods, tools and resources I used in my study in the following sections.

Keywords: technology, development, note, application,

INTRODUCTION

Note apps are our digital assistants for an organized life. Digital note applications have replaced the annual agendas we used before digitalization. Although agendas are still widely used, they will be overshadowed by digitalization day by day. The biggest reason for this is that agendas are single copies and it is not possible to access the contents if your agenda is damaged or lost. On the other hand, with digital note applications, you can access your notes and reminders from every device you log in to your account. However, some people have not been able to keep up with the pace of technological development, and these people constitute a significant percentage of the working population. Therefore, the complexity of comprehensive note applications creates a challenging situation for these people. There are many note applications that serve many specific purposes. We will get to know these applications in the next section. PinNote is a desktop note application that aims for maximum efficiency with minimum effort with its user-friendly interface.

It provides maximum efficiency with minimum effort, thanks to the features of adding notes and reminders and sharing the notes and reminders you add with your friends. Thanks to the notes and reminders you share with your colleagues in your work life, you can avoid forgetting about meetings and important tasks, and you can follow the progress of the project through the notes you create with your work group. During your education life, you can create common notes and reminders with your classmates and keep track of your tasks such as exams, homework, and projects. In your private life, you can keep track of what you need to pick up on your way home, your child's school meeting, your doctor's appointment and similar situations

during the day, and organize your life, thanks to the reminder you set for work or after school hours.

In conclusion, PinNote is a very simple but effective note application that appeals to everyone. No matter how bad you are with technology, using this application is simple and understandable for everyone.

RELATED WORKS

There are many digital note applications. Among these applications, some are general note applications, while others serve very different specific areas. Let's examine some of these note apps. Google Keep[1] is a pretty simple notes app. For general use. You can create notes and reminders and share them with your contacts, but the interface cannot be said to be successful. Obsidion[2] is a more complex notes application. The good thing is your notes are stored on your device. This gives you offline access and privacy. You can also create link links between your notes and visualize the relationship between them graphically. In addition, it also offers services such as simultaneous teamwork and synchronizing files across devices. Although these services are great for people who are good with technology, the middle-aged and elderly working population needs to make a separate effort to use this note application and it is a great application for daily use. What distinguishes PinNote from these applications is its easily understandable interface and simple working principle.

[1] https://keep.google.com/

[2] obsidian.md

IMPORTANCE OF WORK

Sometimes people who share common areas also share activities and important notes. As a result of verbal notification of this important information and events, people may forget, or as the number of people to be notified increases, some people may not be able to access the information and the consequences may not be pleasant. The purpose of the PinNote application is to share notes and reminders interpersonally. People who share common areas have common activities and common notes. Sharing these with people through an application ensures that important information reaches the right person at the right time. This transfer of information has different importance at very different times and places in our lives. An application that will remind you of important meetings in business life, important exams and assignments in education life, appointments and invoices in private life, and send reminders to people who will attend joint events with you will help you get your life in order. With these features, PinNote is a great secretary for you. You and your co-participants will not miss your meetings, exams and events with PinNote.

PROPOSED OF METHODOLOGY

In this section, we will get to know the technologies and methods used in the production of PinNote, a desktop note application.

Tools used:

Visual Studio 2022

SQL Server Management Systems (SSMS)

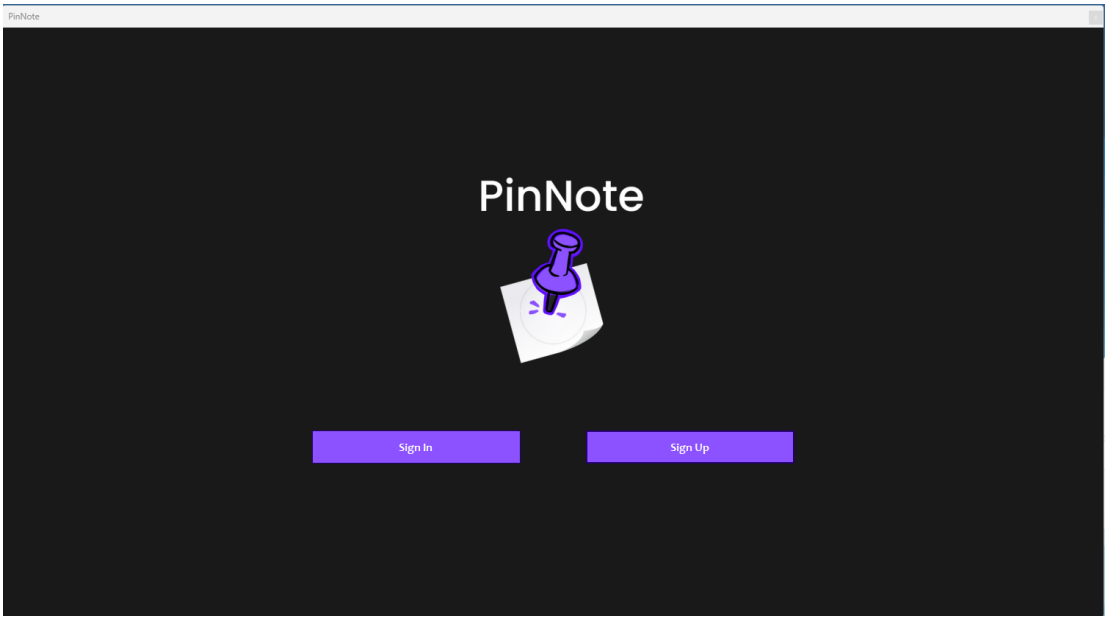
Windows Forms Application (C#.Net Framework)

1.1 Introduction of the application interface

I created a simple and stylish design for my desktop notes app. When we open the application, we are greeted by the Login page. There are two buttons on this page to log in and register to our account.

Figure 1.1

Landing Page

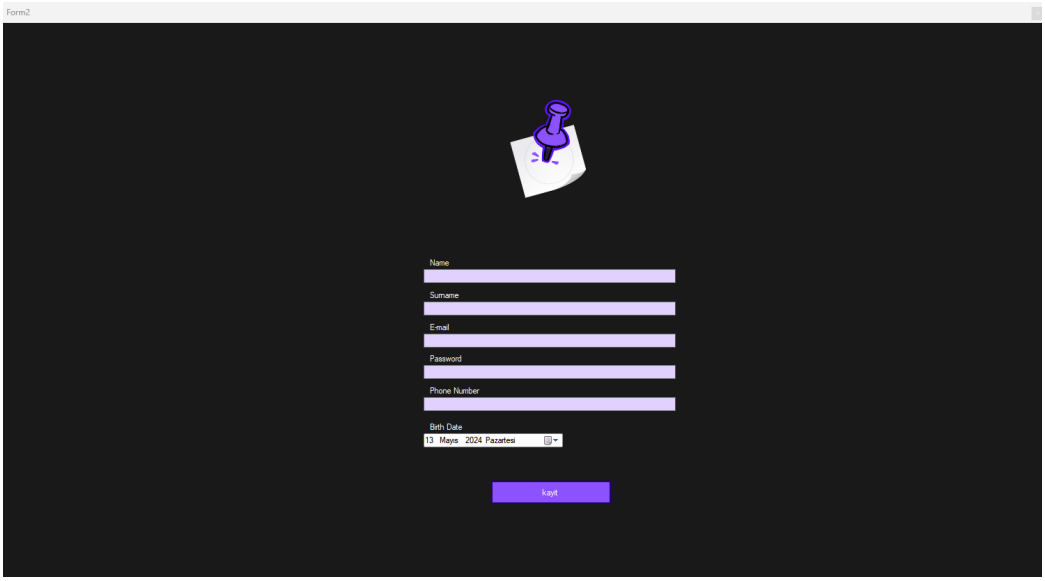


Note The page that welcomes the user when the program first runs.

You will be directed to different pages depending on the selection you make on this page.

Figure 1.2

Registration Page

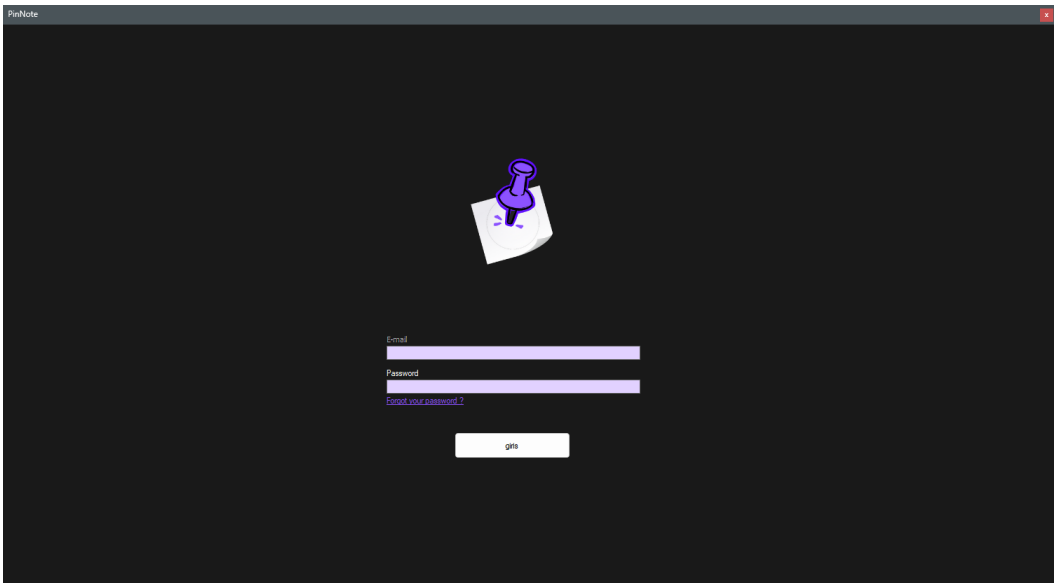


Note The area where the user will register to the system

The code at the back of this page checks whether the information is complete and meets the criteria, and if so, creates an account and directs you to the login page. If the information is not appropriate, it gives an error message and tells the user to correct the information.

Figure 1.3

Login to Account

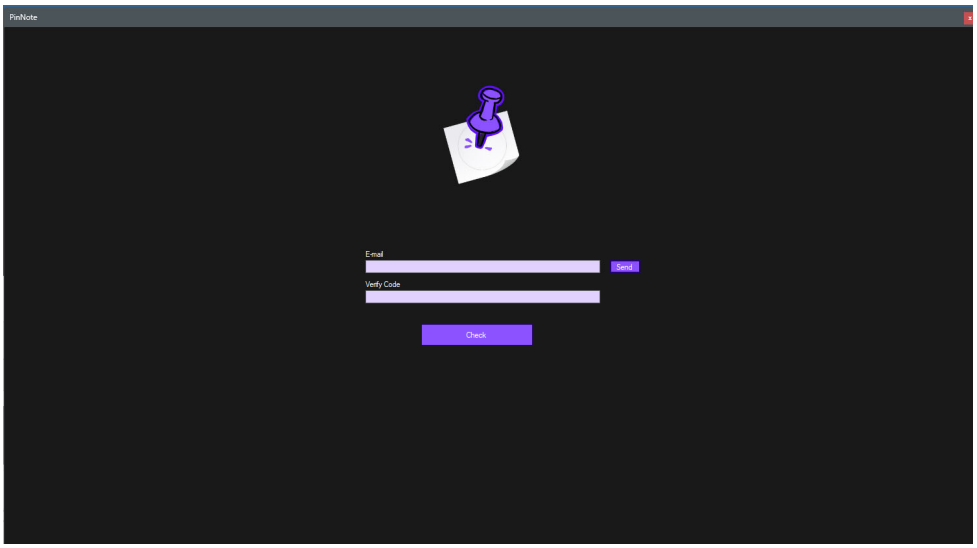


Note The page where the user logs into his account with their information

After registering, the user can log in to his account by entering his information, or if he has forgotten his password, he will be directed to the password reset screen by clicking on the relevant section. On the password reset screen, the user clicks the "Send" button after entering his e-mail information into the box. If the e-mail entered by the user matches the database data, the program sends the verification code to the user by e-mail. If the e-mail does not match the database data, it displays an error message to the user.

Figure 1.4

Verification Code Page

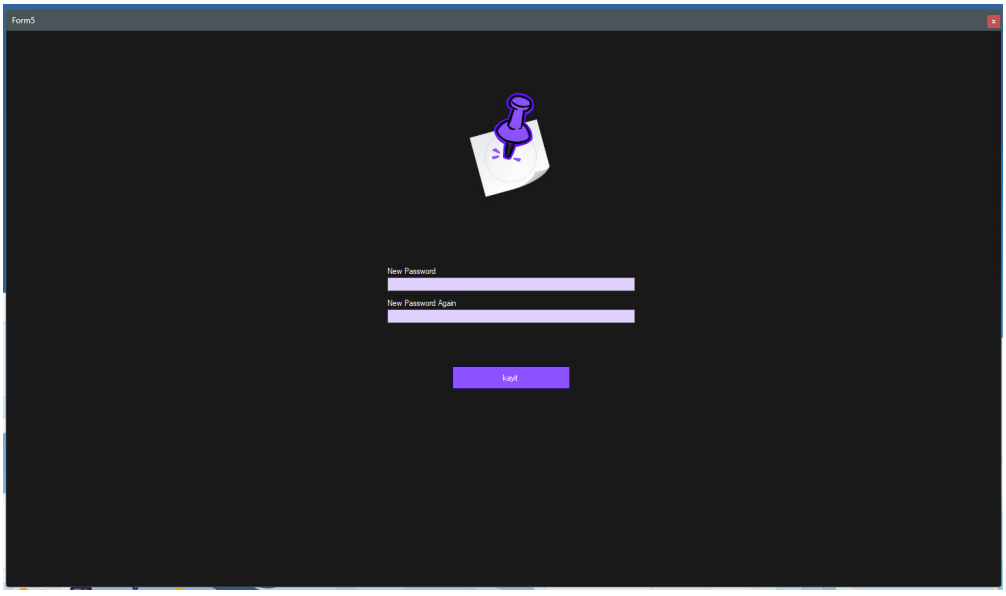


Note The user must verify via e-mail before resetting the password.

If the e-mail is OK, the verification code is checked. If the sent code matches the code entered by the user, the user is directed to the password creation screen.

Figure 1.5

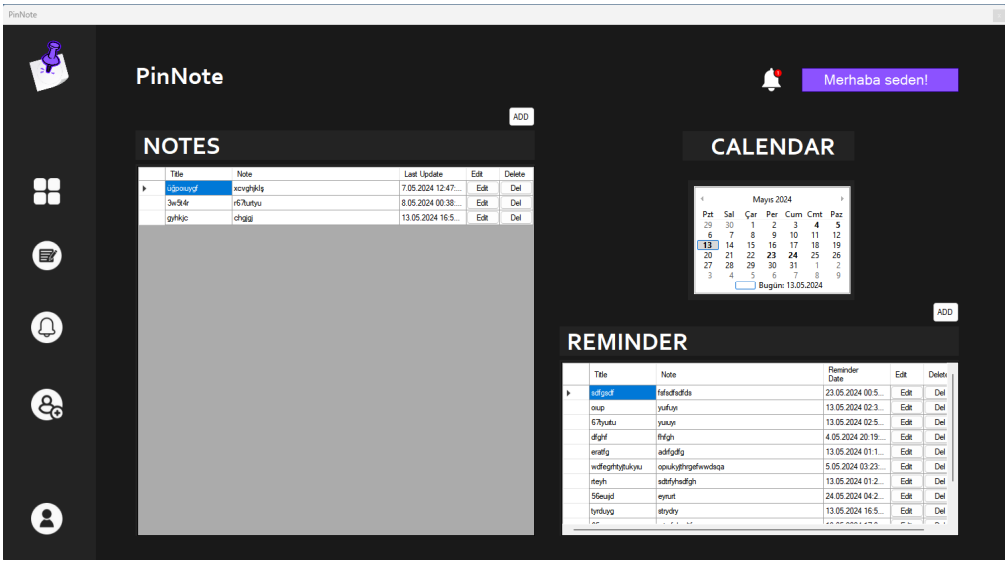
Password Reset Screen



If the passwords entered by the user match each other and meet the criteria, they are saved in the database and the user is directed to the main screen.

Figure 1.6

Homepage Screen

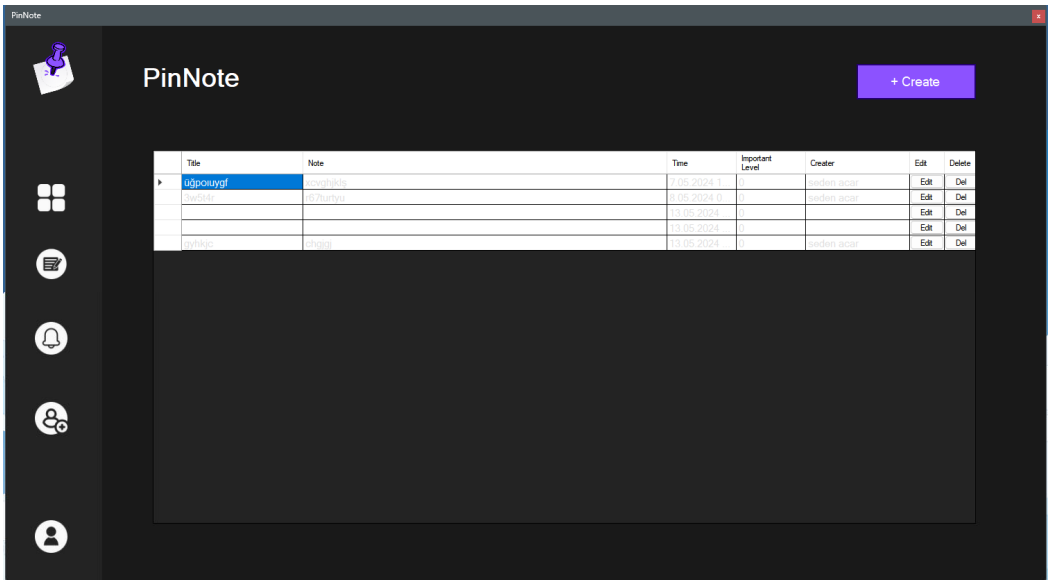


Note Home page of the application. The first screen that appears when you log in to your user account.

Notes and reminders are listed on the main screen, you can be directed to the editing screen or delete notes and reminders through the list. Days for which reminders have been created on the calendar are shown in bold. From the side menu, you can go to Homepage-Notes-Reminders-Friends-Profile pages respectively. Additionally, you can quickly switch to the creation screen from the small boxes just above the list of notes and reminders.

Figure 1.7

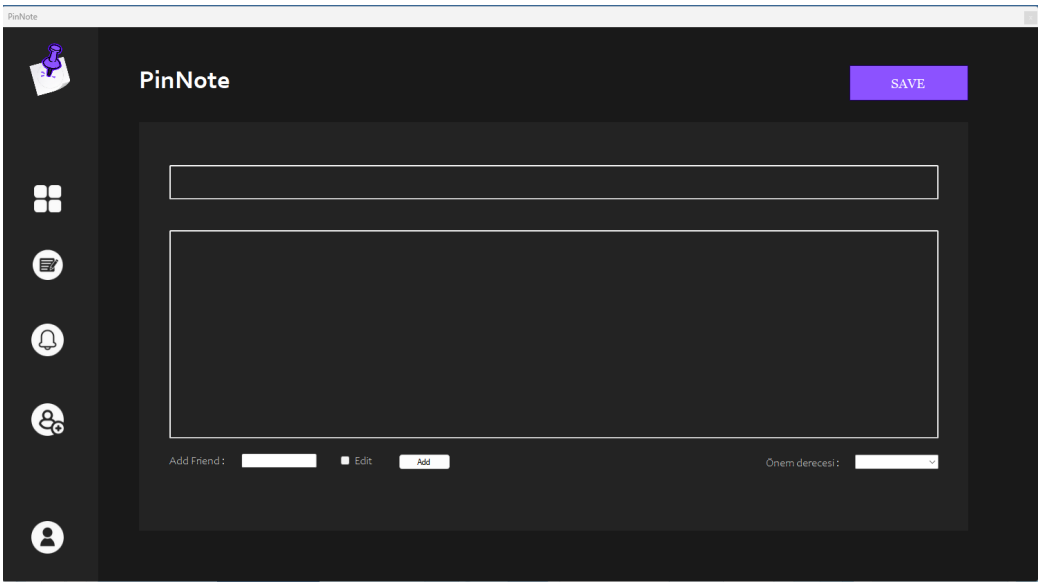
Notes Screen



Your grades are listed on the Grades page. You can create new notes with the “+Create” button or edit or delete existing notes with the buttons on the list.

Figure 1.8

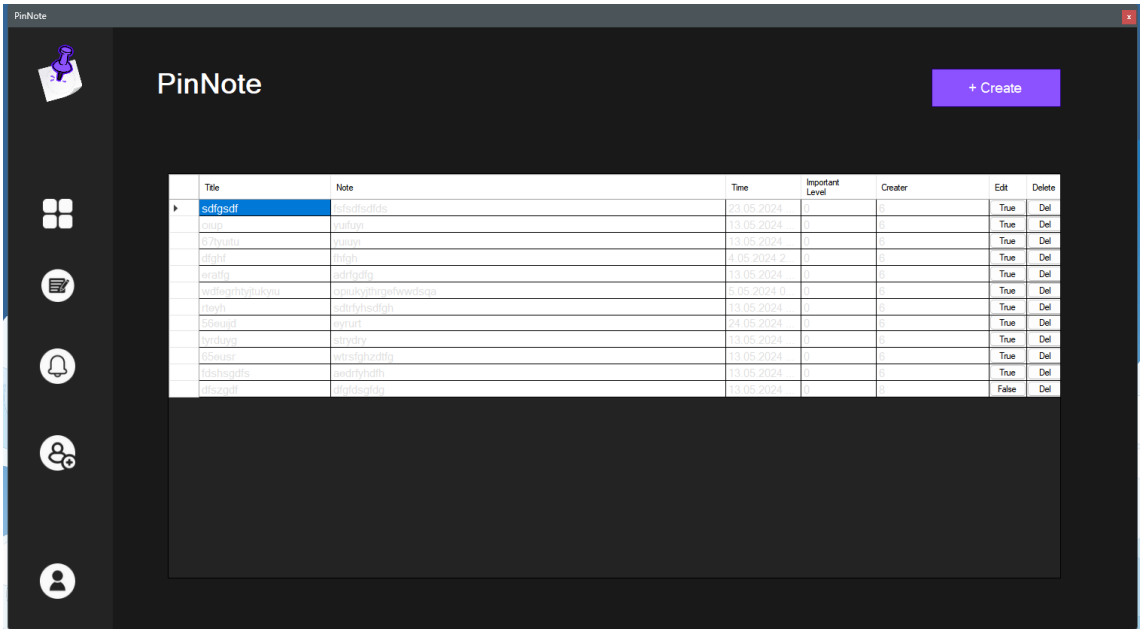
Note Editing Screen



On the note editing screen, the note title and content are written in the relevant boxes. Friends with whom the note will be shared are added with the Friend ID. The information is added to the database with the “SAVE” button.

Figure 1.9

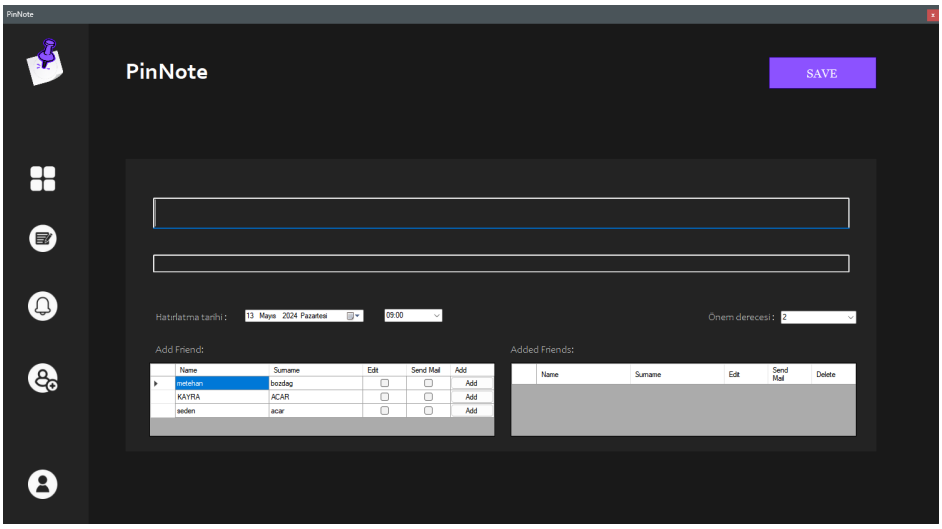
Reminder Screen



Reminders are listed on the reminder screen, and the selected reminder can be edited or deleted with the relevant buttons in the list. With the “+Create” button, you will be directed to the new reminder creation page.

Figure 1.10

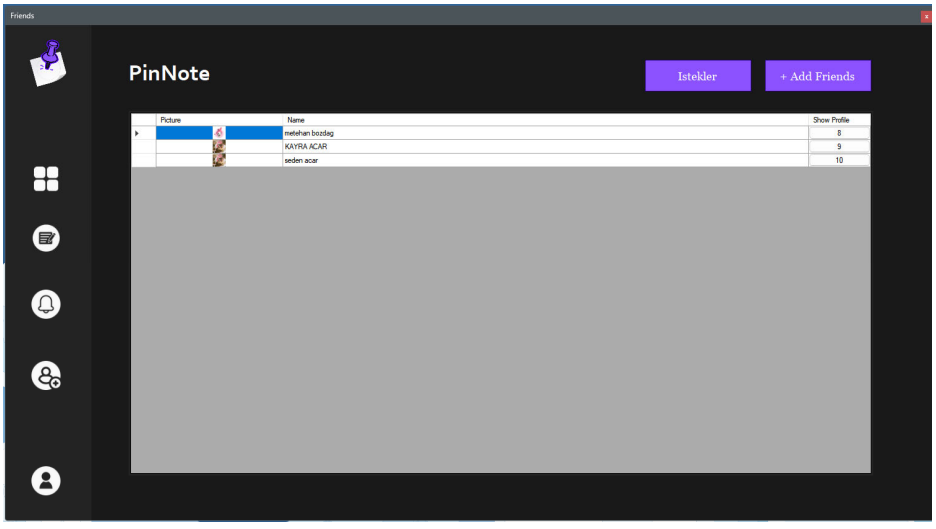
Reminder Creation Screen



On the reminder creation screen, the reminder title and reminder detail are written in the relevant boxes. The date and time of the reminder are selected from the section below the boxes. The importance level of the reminder is selected on the right of the date and time section. Finally, the friend to share with is selected from the friend list at the bottom and the reminder is saved in the database with the "SAVE" button.

Figure 1.11

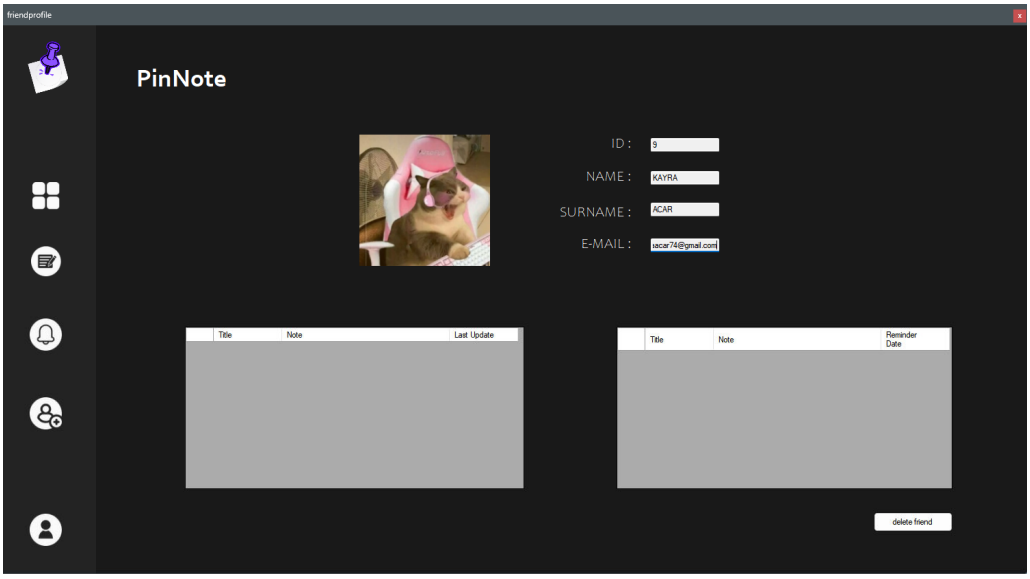
Friends Page



On the Friends page, the user's added friends are listed. You will be directed to the friend's profile from the button in the list.

Figure 1.12

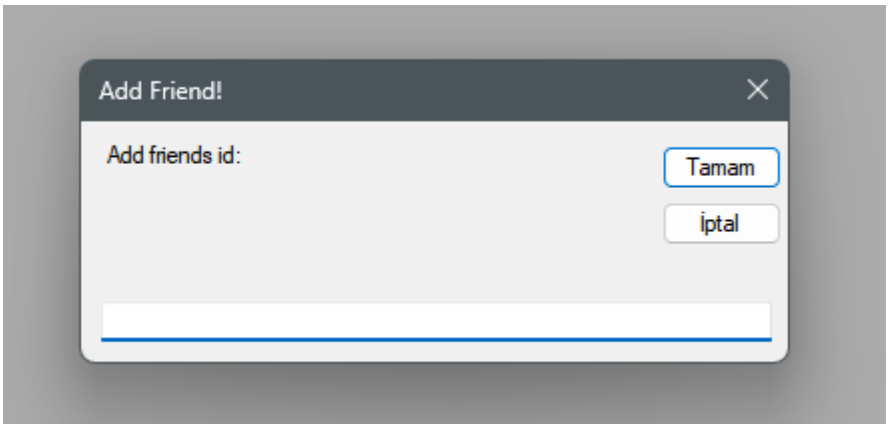
Friend's Profile



Information is accessed from the friend's profile.

Figure 1.13

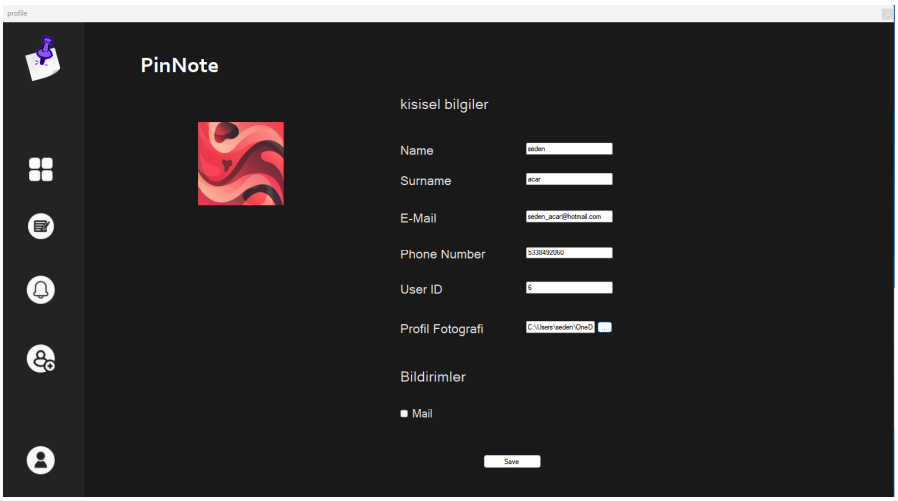
Add Friend Screen



By clicking the "+Add Friends" button on the Friends page, the friend adding section opens and a friend request is sent by entering the ID of the person to be added.

Figure 1.14

Profile Page



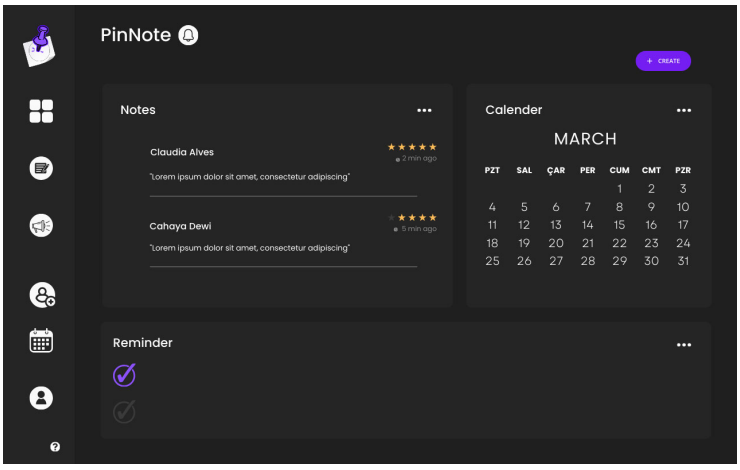
The user's personal information appears on the profile screen. User can edit profile photo.

1.2 Application Construction Process

Until this section, we have examined the project from the user's perspective. Next is the preparation part of the project. When starting the project, I first visually designed what the interface of the project should be from the "Canva" site.

Figure 2.1

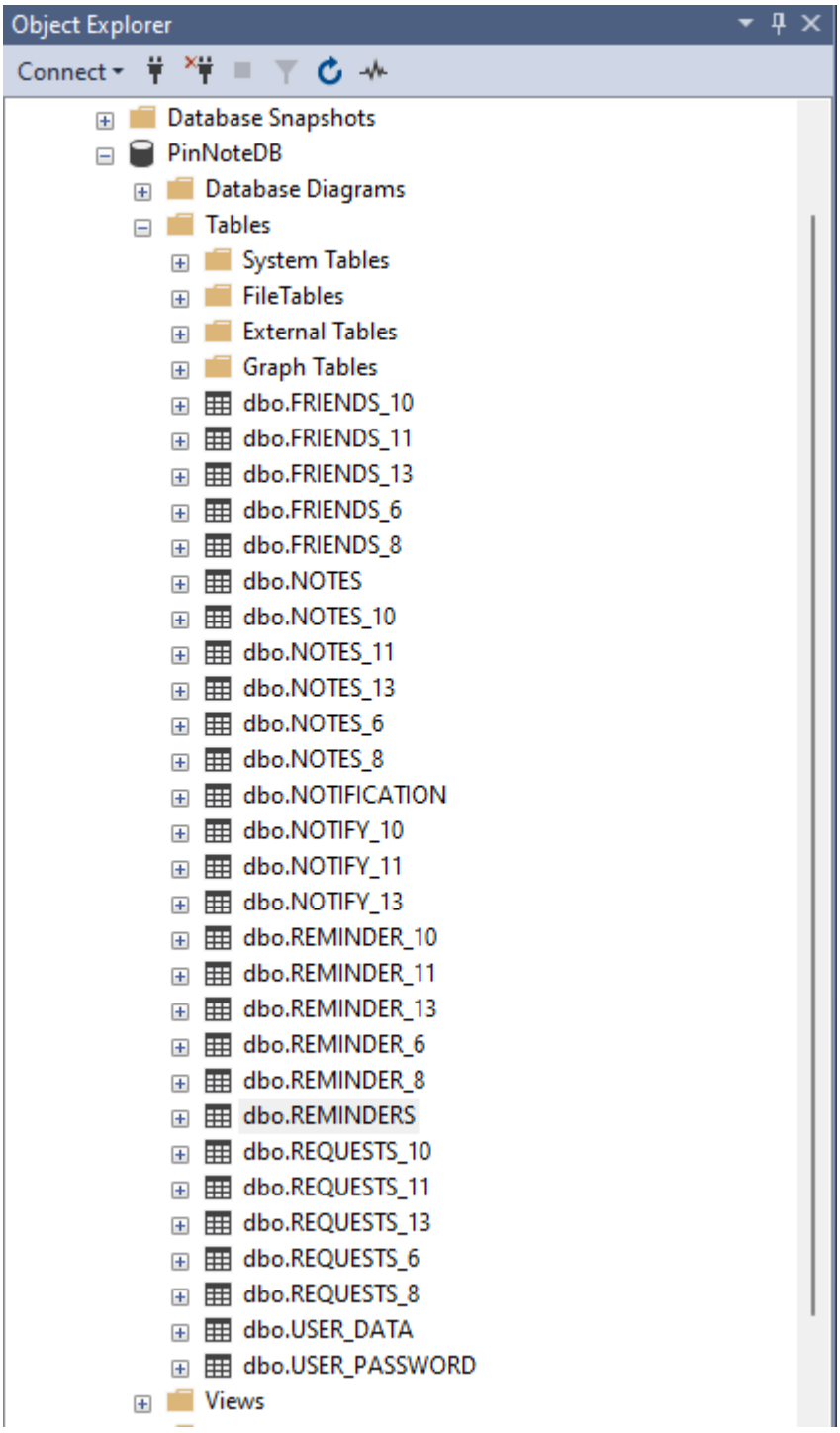
Homepage Image Planned During Project Design



During this process, I also determined the features that the project would have. After the project was formed as an idea, I did research to choose the platform, database and language I would use. During the research, I determined that Xamarin.Forms, Flutter and Windows Forms were suitable for my job as platforms. Since I knew the C# language, I eliminated the Flutter option because I would have to learn the Dart programming language for Flutter and this would require extra time. Among Xamarin and Windows.Forms, I chose Windows Forms (C# .Net Framework) because it seems simpler. After choosing the platform, I researched the database application suitable for my platform. I chose SSMS among MongoDB, MySQL and SSMS (SQL Server Management Studio 19). After completing the general design of the project, I created the UI design in Windows Forms (C# .Net Framework). After completing the UI design, I moved on to database design. I created the information that should be kept in the database and the relationships between the tables. Although I may need to make additions and edits later in the project, I have basically completed the design at this stage.

Figure 2.2

Database Tables



FRIENDS_userid, NOTES_userid, NOTIFY_userid, REMINDER_userid, REQUESTS_userid tables are created for each registered user. Default notes are added to note and reminder tables.

After creating the database, I entered the coding part, starting from the Login page in the middle of the 3rd week. For the login section, I coded the orientation with two buttons. One of these buttons directs the user to the login page, while the other directs the user to register.

Figure 2.3

Landing Page Codes

```
1 başvuru
private void button1_Click(object sender, EventArgs e)
{
    kayit_ekran_2 formm2 = new kayit_ekran_2();
    formm2.Show();
    this.Hide();
}
1 başvuru
private void button2_Click(object sender, EventArgs e)
{
    hesaba_giris_ekranı_3 formm3 = new hesaba_giris_ekranı_3();
    formm3.Show();
    this.Hide();
}
```

After the login page, coding of the account login screen started. A series of coding processes were carried out, simply checking whether the e-mail data received from the user was in the USER_DATA table and checking whether the data, if present, matched the user's password. The form redirect code was also written for the password reset link. The random verification code generation process has been defined on the password reset screen. This verification code had to be sent to the user by e-mail. Since I don't know how to do this. The code on the page (Figure 2.4) was adapted to my own code.

Figure 2.4

C# E-mail Send Code

C# Kopyala

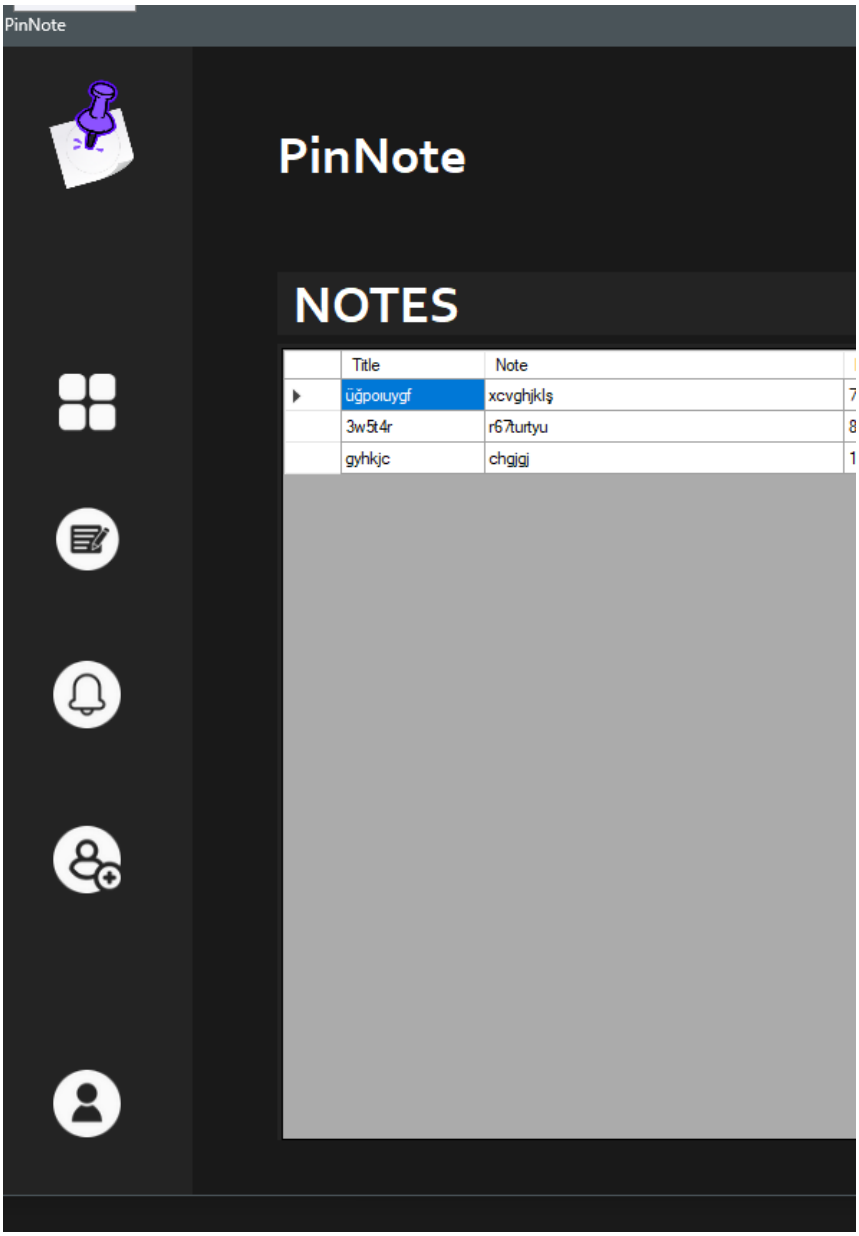
```
private void ThisAddIn_Startup(object sender, System.EventArgs e)
{
    SendEmailtoContacts();
}

private void SendEmailtoContacts()
{
    string subjectEmail = "Meeting has been rescheduled.";
    string bodyEmail = "Meeting is one hour later.";
    Outlook.MAPIFolder sentContacts = (Outlook.MAPIFolder)
        this.Application.ActiveExplorer().Session.GetDefaultFolder
        (Outlook.OlDefaultFolders.olFolderContacts);
    foreach (Outlook.ContactItem contact in sentContacts.Items)
    {
        if (contact.Email1Address.Contains("example.com"))
        {
            this.CreateEmailItem(subjectEmail, contact
                .Email1Address, bodyEmail);
        }
    }
}

private void CreateEmailItem(string subjectEmail,
       string toEmail, string bodyEmail)
{
    Outlook.MailItem eMail = (Outlook.MailItem)
        this.Application.CreateItem(Outlook.OlItemType.olMailItem);
    eMail.Subject = subjectEmail;
    eMail.To = toEmail;
    eMail.Body = bodyEmail;
    eMail.Importance = Outlook.OlImportance.olImportanceLow;
    ((Outlook._MailItem)eMail).Send();
}
```

If the verification code sent to the user by e-mail and the verification code generated by the program match, the user is directed to the password reset screen. On this screen, the user is asked to set a password. To prevent spelling errors in the specified password, you are asked to enter the same password in two different fields. If the created password meets the criteria and is the same, the password is saved in the database and the user is directed to the login page. After the login process is completed, the registration process begins. For the registration screen, a form is created to obtain the necessary personal information from the user. The suitability of this information is checked and if there is no problem during the checks, the information is

recorded in the database and the necessary tables are created and default notes are added, thus completing the registration process. Once the login and registration procedures are completed, coding of the main important points of the project begins. First of all, the side menu for switching between forms was coded in all forms.

Figure 2.5

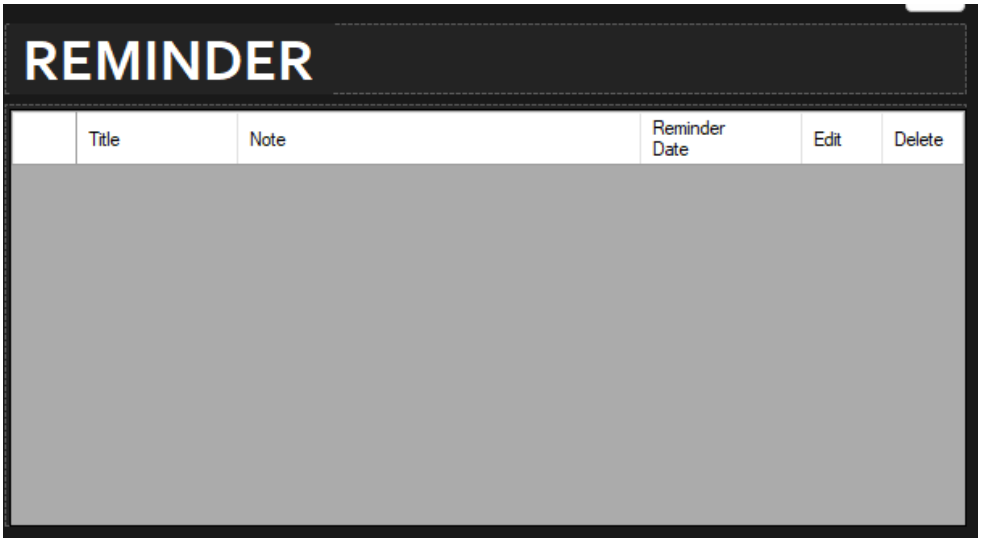
Side Menu



I researched the use of datagridview to list notes and reminders after the side menu. Datagridview was used on the home page, listing reminders and listing notes.

Figure 2.6

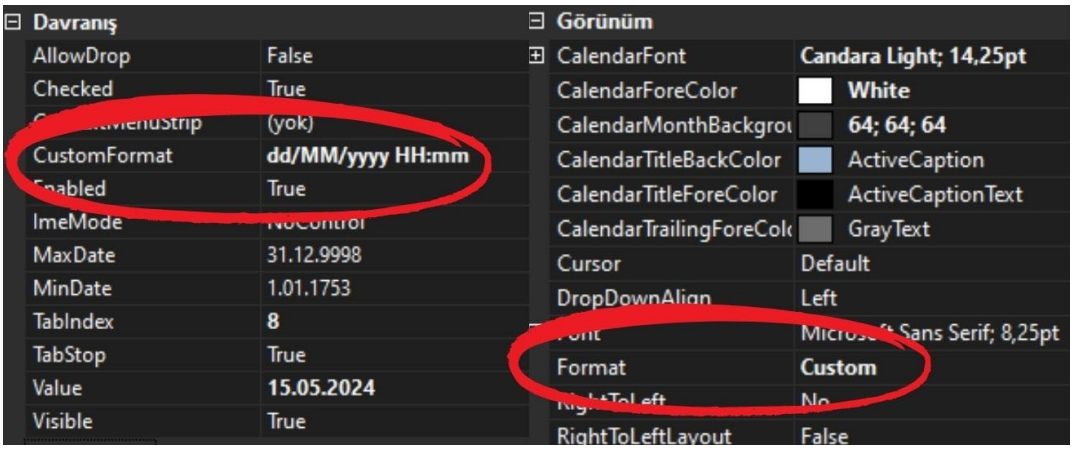
DataGridView



A calendar has been added to display the days with reminders on the homepage. Two textboxes were added to the note creation page to hold the note title and content. Added textbox and checkbox to share the note with friends. Also added combobox to save the importance of the note. Finally, a save note button was added. The save note button checks the entered data and records the note into the database according to its suitability and adds the note id to the user's notes table. Similarly, two text boxes were added for the reminder title and detail section on the reminder creation screen. Used datetimepicker to get the reminder date. By default, Datetimepicker only allows the user to select a date. In this project, we need to select both date and time. Therefore, the settings of datetimepicker should be changed in the properties tab.

Figure 2.7

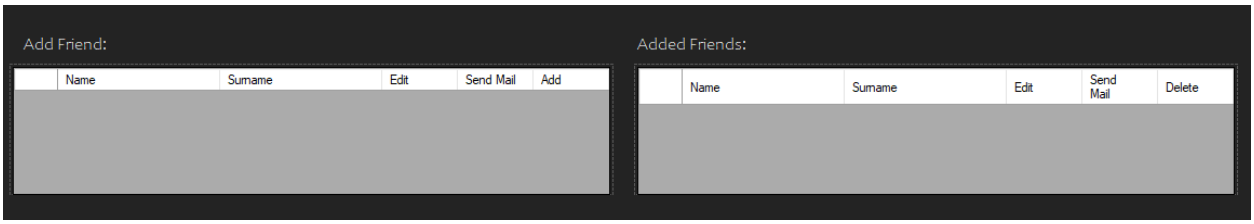
DateTimePicker



After setting Format to Custom on the Appearance tab in the properties, CustomFormat should be set to "dd/MM/yyyy HH:mm" on the Behavior tab. In this form, unlike creating notes, I followed a different method for adding friends.

Figure 2.8

Reminder Creation Screen - DataGridView



Two different datagridviews are created. One lists friends. An “Add” button is added to the datagridview that lists friends. Friends to share with are selected with this button. The selected friends are shown on the second datagridview on the right. Reminders are shared with friends added to the list on the right during registration. For the Friends form, data retrieved from the database is displayed again with datagridview. There are two buttons in the upper right corner. The one on the left is used to view friend requests.

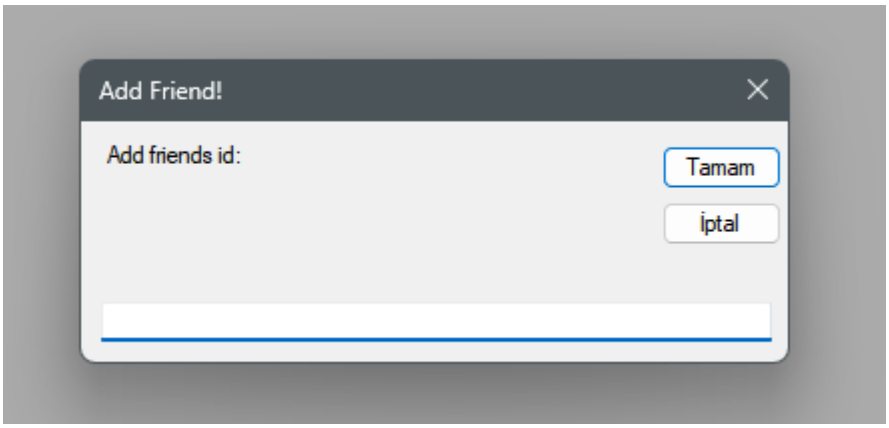
Figure 2.9

Friend Requests

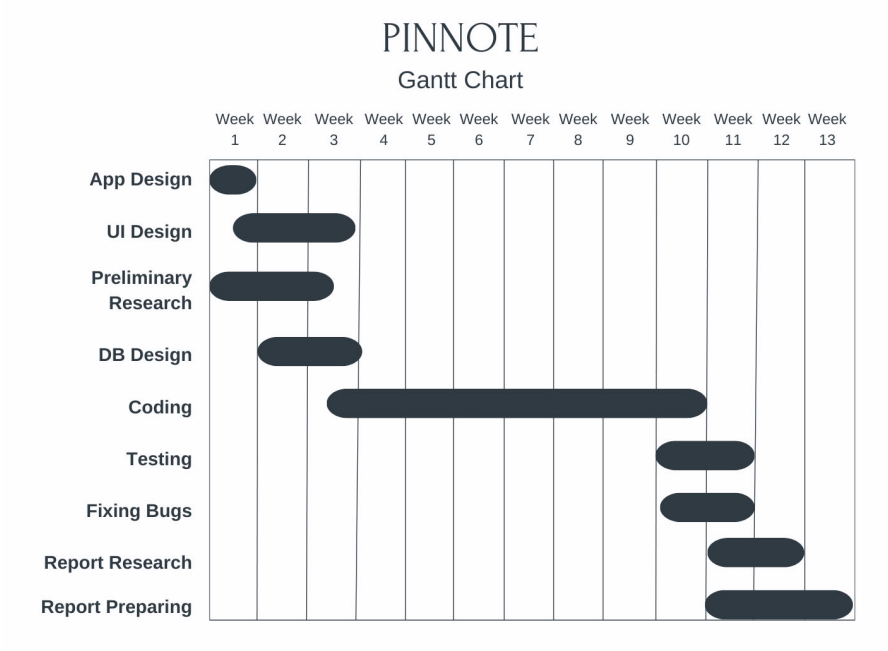
It was created as a widget form. Sorts the friend requests received by the user. The button on the right is for adding friends. With Interaction.InputBox, the ID information of the person to be added is entered in the pop-up that appears on the screen.

Figure 2.10

Add Friend



If the entered id information is available in the database, it is added to the requests table of the id owner. The datagridview button is used to view the friend profile. The friend's profile photo and personal information are displayed on the friend profile. With the icon at the bottom of the side menu, the user can also view his own profile and edit their profile photo. Photos are kept in the database.



CONCLUSION

Creating this project was a good experience for me. It allowed me to consolidate and learn a new platform by using the C# language I had just learned. The platform I learned is Windows Forms (C# .Net Framework). It is used to create Windows desktop application. I had a little difficulty in the UI design part, but the coding and algorithm part was easier. I learned database connection and queries. Until I started doing this project, I didn't think I could do a project at this level. I thought I needed to learn more to create a working application, but I created a largely successful project. There are also errors in my project that I cannot solve and features that are not available. Even though I had problems with forms and database, I improved myself. There are some features that can be added to my project, some of them are grouping by color for notes and reminders, grouping friends such as work, school, home. Sending reminders via SMS and e-mail. I will try to do these to improve myself, but for now, I tried to create the best project I could in three months.

REFERANCES

https://www.easybib.com/guides/citation-guides/apa-format/(APA Format)

https://www.youtube.com/@MurattYucedag (sending e-mail and adding photos to the database with C#)

learn.microsoft.com (How to use windows forms components)

evernote.com (literature review)

obsidian.md (literature review)

www.w3schools.com (SQL queries)

https://mustafabukulmez.com/2018/04/22/c-sharp-timer-kullanimi/

https://mustafabukulmez.com/2018/01/22/c-formlar-arasi-veri-nasil-gonderilir/

https://www.btkakademi.gov.tr/portal/course/c-programlama-26083 (C# course)

https://www.hikmetokumus.com/makale/41/csharp-ile-inputbox-kullanimi

https://www.mustafakarsli.com/makale/72/C-Sharp/C-Net-Mesaj-Penceresi-MessageBox-Islemleri.html

https://learn.microsoft.com/tr-tr/visualstudio/vsto/how-to-programmatically-send-e-mail-programmatically?view=vs-2022