

בעז"ה

מיני פרויקט במערכת חלונות

תרגיל 3 א' – WPF

תוכן העניינים

| | |
|--------|---|
| 2..... | מטרת התרגיל |
| 2..... | הנחיות להגשת התרגיל |
| 2..... | המטרה הסופית |
| 3..... | תחילת העבודה |
| 3..... | צעד 1 : יצירת פרויקט חדש מסוג WPF. (שם הפרויקט לפי ההנחיות) |
| 4..... | צעד 2 : הגדרת מחלקות רגילות לשימושנו בהמשך התרגיל |
| 5..... | צעד 3 : חלוקת ה Grid ל 4 שורות |
| 6..... | צעד 4 : יצירת Grid פנימי בתוך השורה העליונה של ה Grid הראשי |
| 7..... | צעד 5 : בחלון הראשי - יצירת ComboBox ו TextBox בתוך עמודות ה Grid הפנימי (העליון) |
| 8..... | צעד 6 : הוספת רכיב תצוגה עבור רשימת התחנות בקו |
| 8..... | צעד 7 : יצירת מאגר קווי אוטובוס וקישורם ל ComboBox |
| 9..... | צעד 8 : הצגת מסלול הקו בבחירת קו אוטובוס דרך ה ComboBox |
| 9..... | צעד 9 : לסיום. נגיעות אישיות. |

מטרת התרגיל

בתרגיל זה נבנה תוכנית המציגה מידע על קווי אוטובוס ותחנות במסלול הקו. לצורך זה נשתמש במערכת WPF : מערכת המאפשרת יצירת ממשק גרפי עבור המשתמש. באופן כללי מערכת WPF מהווה 'שכבה' אחת (UI/PL) של מערכת שלמה שבה הפעילות והמידע מופרדים, ונרשמים ב'שכבות' אחרות כפי שילמד בהמשך הקורס. עבור תרגיל זה, מתוך רצון להתמקד ברכיב הגרפי, נחרוג מהמקובל ונרשום את הרכיבים הגרפיים, הפעילות והמידע ללא הפרדה.

הנחיות להגשת התרגיל

- ✓ העבודה תתבצע באותם הזוגות כמו בתרגילים הקודמים
- ✓ **חובה** להשתמש במערכת ניהול גרסאות Git ומאגר מרוחק משותף של GitHub – חובה להשתמש באותו המאגר כמו התרגילים הקודמים
- ✓ פרויקט התרגיל ייפתח באותו ה-Solution יחד עם התרגילים הקודמים
- ✓ יש להגיש במודל קישור על פי ההנחיות בקובץ "הגשת מטלות בקורס מיני פרויקט במערכות חלונות 153007!"
- ✓ מספר תרגיל זה הינו 03A
- ✓ נא להקפיד על פורמט זה על מנת למנוע מצב של אי קבלת ציון על תרגיל מסוים

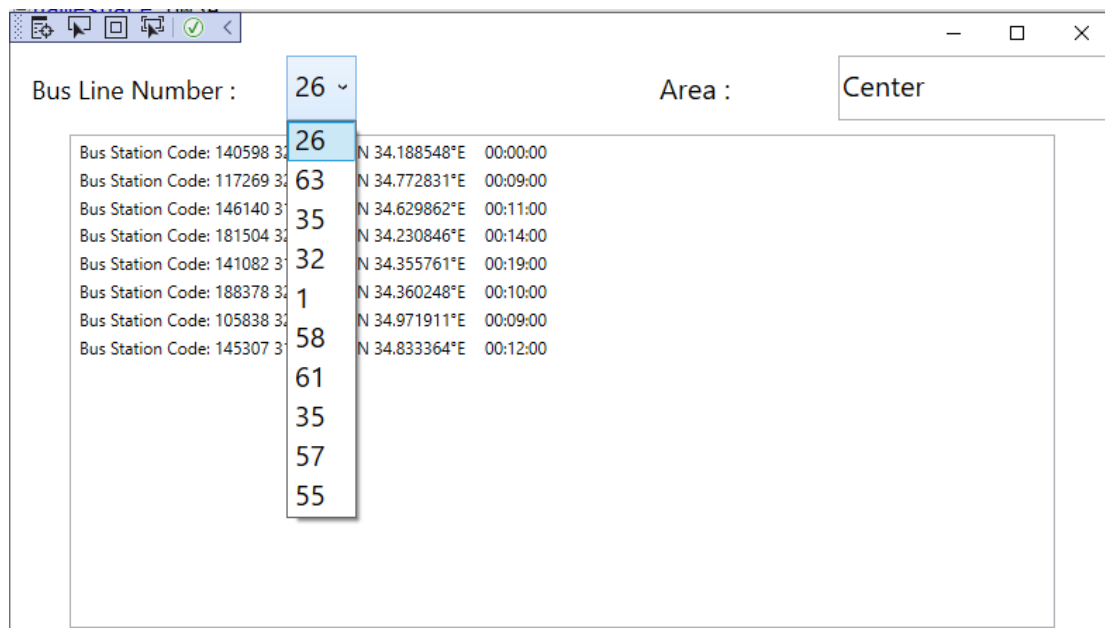
הערה חשובה:

לאחר שתבצעו את התרגיל במלואו, ותבינו את הנעשה – הנכם מתבקשים לבצע בו שינויים גרפיים שיהיו יחודיים לביצוע שלכם. כגון: גודל וסוג גופן, בחירת פריטי מידע שיוצגו וכדומה.

המטרה הסופית

ניתן לראות במודל דוגמת הרצה של האפליקציה המבוקשת.

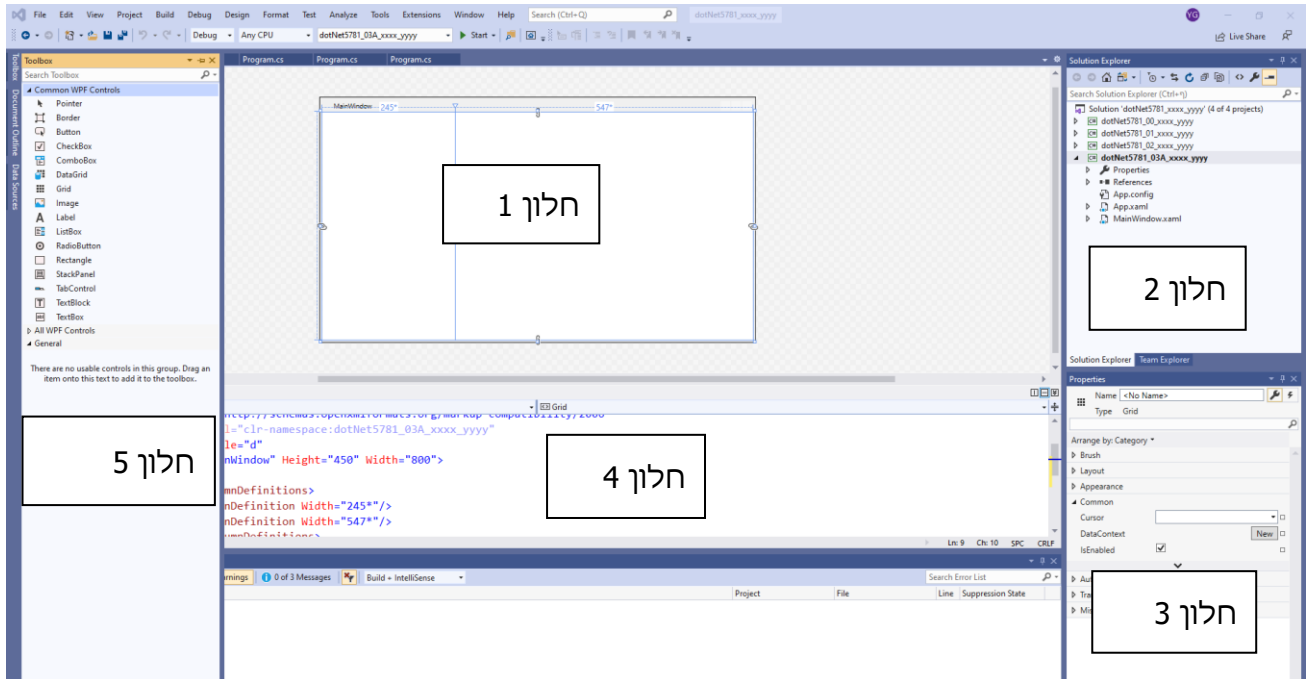
בסופו של תהליך, אנו מעוניינים הגיע לחלון הנראה בערך כך:



תחילת העבודה

צעד 1 : יצירת פרויקט חדש מסוג WPF. (שם הפרויקט לפי ההנחיות)

1. צרו פרויקט חדש מסוג WPF App.
את הפרויקט יש ליצור בתוך אותו solution של התרגילים הקודמים



איור 1: התצוגה על המסך לאחר יצירת הפרויקט

- חלון 1:** חלון ראשי; תצוגת קבצי הקוד והגרפיקה (כך ייראה החלון בזמן ריצה).
חלון 2: סייר לקבצי הפרויקט.
חלון 3: Properties - מאפיין הרכיב הנבחר.
חלון 4: קוד XAML של רכיבי התצוגה.
חלון 5: Toolbox - רשימת הרכיבים הגרפיים שאפשר להציב על החלון הראשי.

במידה ואחד החלונות אינו מופיע ניתן להציגו ע"י בחירה בתפריט view, או לחיצה על שמו הכתוב במאונך בצד שמאל.
ניתן לשלב חלון כך שלא יסתיר חלונות אחרים ע"י לחיצה על ה"נעץ" שבראש החלון.

קוד התוכנית מורכב משני קבצים: קוד ה XAML שאנו רואים בחלון 4, וקוד CS המכונה Code Behind, שניתן לראותו בלחיצה על הקובץ בחלון 2. שני קבצים אלו יחד יוצרים את הפרויקט. על פי רוב, בקוד ה XAML יוגדרו הרכיבים, ובקוד ה CS יוגדרו הפעולות שיש לבצע בעקבות לחיצה על כפתור וכד'. בתרגיל שלנו נשנה מעט בכדי לתרגל טכניקות שונות.

כשאנו יוצרים חלון חדש אנו מקבלים מ'עליו' כברירת המחדל מסוג Panel מסוג Grid. פאנל הוא פקד שיכול להכיל מספר פקדים אחרים.

ישנם מספר סוגי Panel, הנבדלים זה מזה בצורת הסידור הפנימית של הרכיבים:

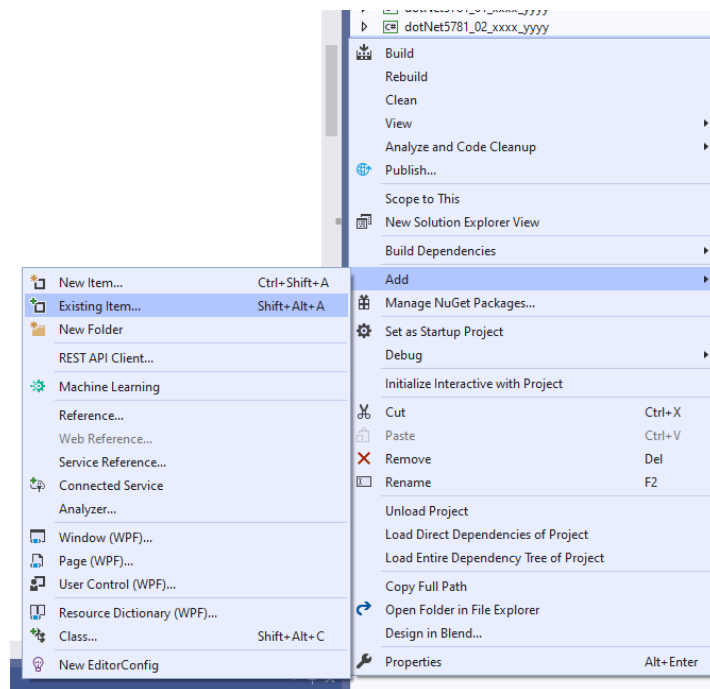
- StackPanel – ערימה. הרכיבים מונחים אחד אחרי השני.
- WarpPanel – ערימה, אך בה - כאשר יש גלישה, הרכיבים יועברו לשורה הבאה.
- Canvas – הצבה של כל רכיב ע"י ציון מיקום הנקודה השמאלית העליונה שלו.
- Grid - ב Panel זה ניתן לארגן את הצבת הרכיבים בצורה טבלאית (חלוקה לשורות ועמודות), כך שהיחס ביניהם קבוע (ואזי - גם אם יישתנה גודל החלון הרכיבים ייוצגו באותה תבנית).

בקוד ה XAML שלפנינו, יש הגדרת החלון, ופרטים כלליים להצגה. ו'בתוך' החלון מוגדר ה Grid שנוצר.

צעד 2 : הגדרת מחלקות רגילות לשימושנו בהמשך התרגיל

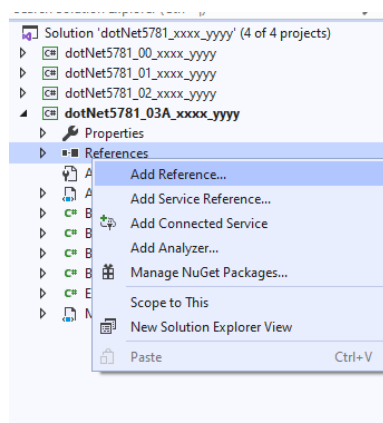
לפני שנתחיל בעיצוב, נכין את מבני הנתונים שנהיה מעוניינים לשלב אותם בהמשך יחד עם הרכיבים הגרפיים.
תזכורת : כאמור לעיל, בפרויקט הבנוי נכון, תהיה הפרדה בין המידע לבין הרכיבים הגרפיים. בהמשך – בפרויקט הסיכום של הקורס, דרישות המחלקות תהיינה שונות.

בתרגיל 2 יצרתם מחלקות המייצגות קו אוטובוס ותחנה במסלול הקו.
בתרגיל זה באפשרותכם ליצור את המחלקות הנצרכות מחדש או לצרף את המחלקות עי"י הוספת פריט קיים ובחירת המחלקות שכבר יצרתם.

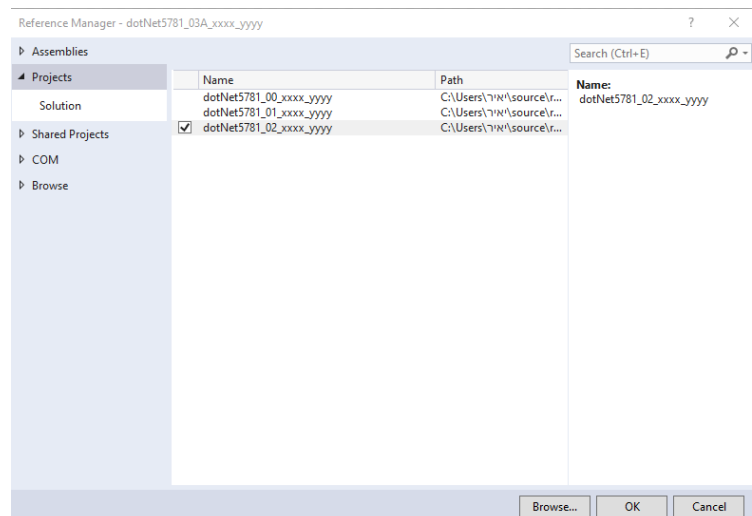


אפשרות אחרת היא להוסיף הפנייה (Add reference) לפרויקטים הקודמים.

1)



2)



צעד 3 : חלוקת ה Grid ל 4 שורות

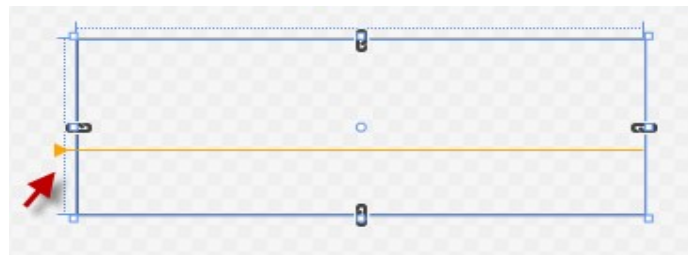
ראינו שהחלון נוצר עם רכיב Grid בתוכו.

1. דרך ה XAML שנו את גובהו של החלון בתוך התג <Window> ל "Height=640".
2. שנו את שם ה Grid ל <Grid Name="MainGrid">
3. חלקו את ה Grid ל 2 שורות ביחס של 50*:400*. תוכלו לעשות זאת ב 2 דרכים (בחרו את הדרך המועדפת עליכם):

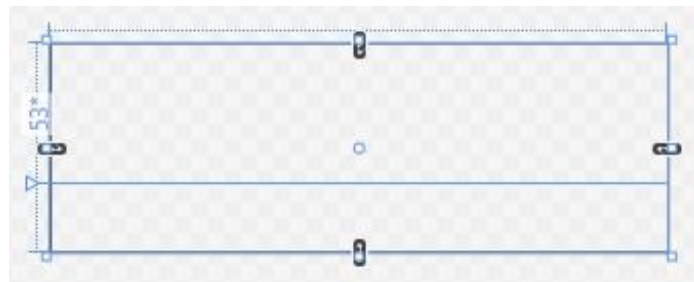
○ דרך 1: כתבו את השינויים ידנית ל-XAML (חלון 5) בתוך התג <Grid> :

```
<Grid Name="MainGrid">
  <Grid.RowDefinitions>
    <RowDefinition Height="50*"/>
    <RowDefinition Height="400*"/>
  </Grid.RowDefinitions>
</Grid>
```

○ דרך 2: בחלון הגרפי (חלון 2) עברו עם העכבר בשוליו של ה grid, ראו קו כתום שמופיע באופן זמני וזו בהתאם עם העכבר. כל עוד העכבר נמצא במסגרת של ה grid.



במידה ונלחץ מקש שמאלי על העכבר, קו זה יישאר ויהווה חלוקה של ה-grid:

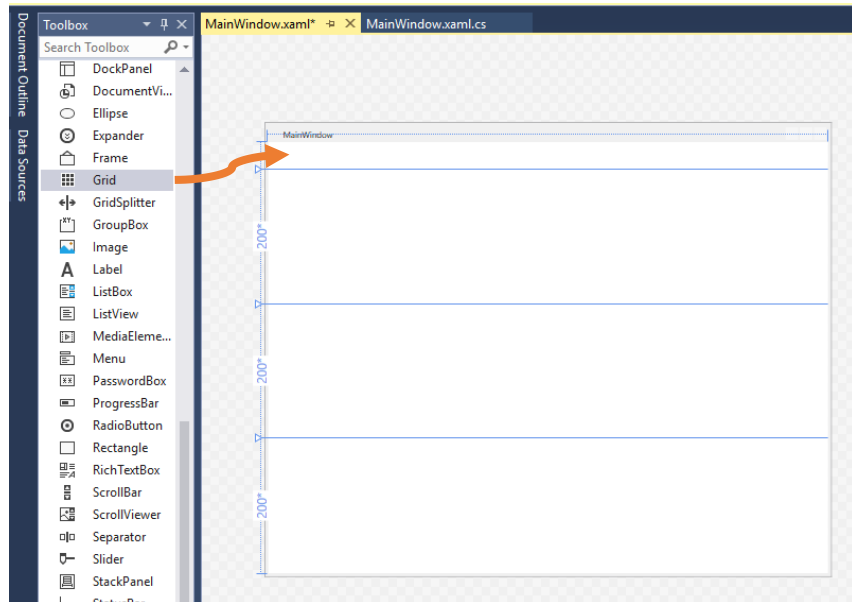


עיינו בקוד שהתקבל בחלון 4, חלון ה XAML:

- קבלתם שתי שורות של RowDefinition מה מסמל הערך שנתנו לגובהן?
- מספרים בלבד - הערכים יישארו קבועים בשינוי גודל החלון.
 - מספרים עם כוכבית - יישמרו היחסים בשינוי גודל החלון.
 - כוכבית בלבד - יישמרו יחסים זהים בשינוי גודל החלון.

צעד 4 : יצירת Grid פנימי בתוך השורה העליונה של ה Grid הראשי

1. צרו Grid פנימי שיופיע בתוך השורה העליונה של ה Grid הראשי, באופן הבא :
גררו עם העכבר מחלון 5 (Toolbox) את הרכיב הגרפי Grid. גררו אותו לשורה העליונה של ה grid הראשי.



2. כתוצאה מהגרירה, נוצר xml tag חדש בשם <Grid> בתוך תג ה <Grid> החיצוני.
החליפו אותו בהגדרה הבאה :

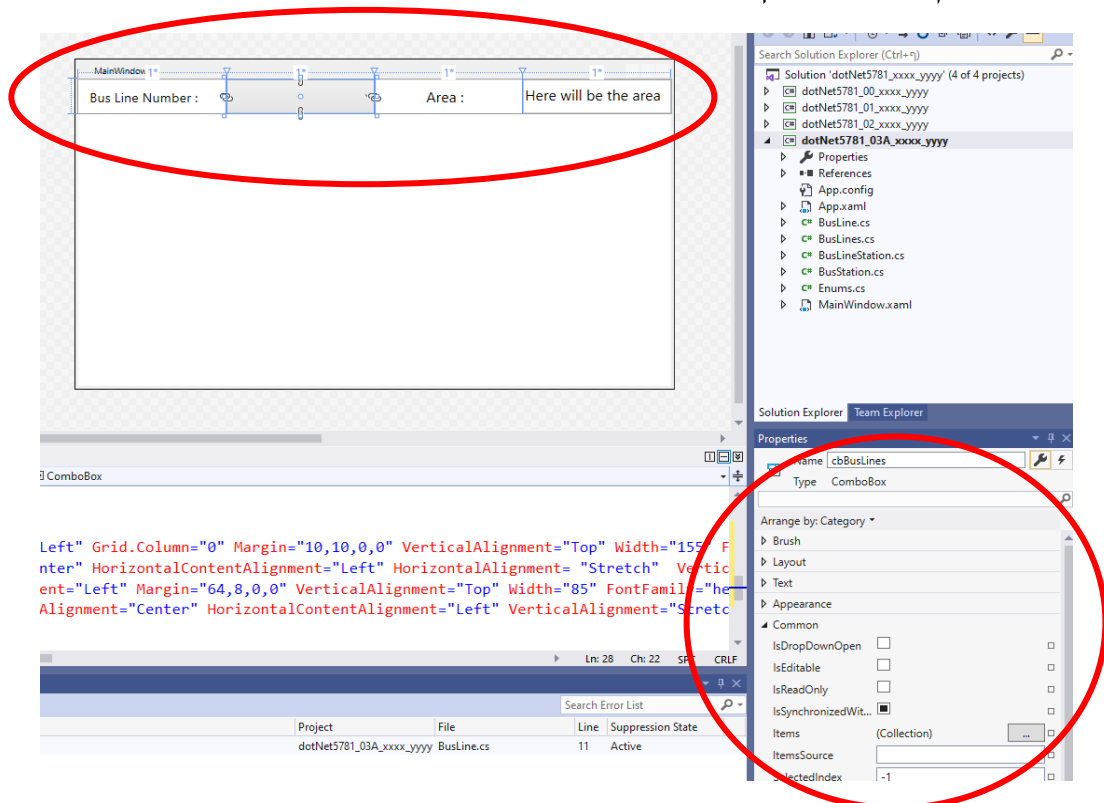
```
<Grid Name="UpGrid" HorizontalAlignment="Stretch" Height="auto" Grid.Row="0"
VerticalAlignment="Stretch" Width="auto">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="*" />
  </Grid.ColumnDefinitions>
</Grid>
```

הסבר :

ה-grid הפנימי החדש ממוקם בשורה 0 של ה-grid שבו הוא מוכל (הספירה מתחילה ב-0), הוא מתוח לרוחב ולגובה ומקבל את הגובה והרוחב שלו באופן אוטומטי. הוא מחולק ל 4 עמודות שוות רוחב (*).

צעד 5 : בחלון הראשי - יצירת TextBox ו ComboBox בתוך עמודות ה Grid הפנימי (העליון)

נרצה שהחלון הראשי יראה כך :



1. גררו עם העכבר מחלון 5 (Toolbox), לתוך העמודה הראשונה של ה grid הפנימי, את הרכיבים הגרפיים הנחוצים כדי להגיע לתוצאה שבאיור לעיל (כעת יש לגרור אותם בלבד, כאשר מיקום מדויק יקבע בהמשך).
2. כתוצאה מהגרירה, ייווצרו בתוך תג ה <Grid> הפנימי, xml tags חדשים בשם <ComboBox> ו <TextBox> ושני <Label>
3. מכיוון שבהמשך נרצה להתייחס לרכיבים האלו בקוד cs אזי כעת שנו לכל אחד מהרכיבים את שמו ועוד כמה מאפיינים נצרכים, באופן הבא :

| סוג הפקד | מאפיינים שיש לשנות | הסבר חשוב |
|----------|---|---|
| ComboBox | Name ="cbBusLines" HorizontalAlignment = "Stretch" VerticalAlignment ="Stretch" Grid.Column ="1" | ניצור ComboBox שיופיע בתוך העמודה השנייה של ה Grid הפנימי. ה ComboBox הוא רכיב גרפי להצגת רשימה, בהמשך נאפשר בחירה של קו אוטובוס מתוך הרשימה שתופיע ב ComboBox. |
| TextBox | Name ="tbArea" HorizontalAlignment = "Stretch" VerticalAlignment ="Stretch" Grid.Column ="3" Text ="{Binding Path=Area}" | ניצור תיבת טקסט, TextBox, שתופיע בתוך העמודה הרביעית של ה Grid הפנימי. עשינו Binding מכיוון שבהמשך, נרצה לקשר תיבת טקסט זו באופן אוטומטי לאזור בו הקו שנבחר ב ComboBox פועל |

4. בחלון הגרפי (חלון 2) עמדו עם העכבר על תיבת הטקסט שיצרתם ודרך חלון 4 (properties) שנו לתיבת הטקסט את סוג הפונט ואת גודלו **כרצונכם**. ראו הדגמה באיור שלעיל.
5. לאחר מכן, התבוננו בקוד ה XAML וראו דרכו את תוצאות השינויים שערכתם. **הריצו את הפרויקט ובדקו את התוצאות המתקבלות.**

צעד 6 : הוספת רכיב תצוגה עבור רשימת התחנות בקו

1. גררו עם העכבר מחלון 5 (Toolbox), לתוך Grid החיצוני רכיב תצוגה מסוג ListBox, התאימו את הרכיב לחלון לפי אחת השיטות בצעד 3.
2. מכיוון שבהמשך נרצה להתייחס לרכיב זה נקבע את שמו : `Name="lbBusLineStations"`
3. אנו רוצים שהמידע יתקבל מתוך אוסף קיים ולכן נקשר את מקור הנתונים, בהמשך נקבע את המידע המקושר : `ItemsSource="{Binding}"`

צעד 7 : יצירת מאגר קווי אוטובוס וקישורם ל ComboBox

1. **צרו** במחלקה `MainWindow` (קוד CS של החלון שנוצר) אוסף של קווי אוטובוס, אתם יכולים להשתמש במחלקה שיצרת עבור האוסף בתרגיל 2.
 2. בתוך הבנאי של החלון הראשי, עליכם להגריל בצורה רנדומלית 10 קווי אוטובוס שונים ותחנות לאותם קווים, יש לוודא שעבור קו אוטובוס קיים מידע על מסלול הקו, כגון משך הנסיעה בין התחנות.
 3. **הוסיפו** לבנאי של מחלקת `MainWindow`, לאחר אתחול הרכיבים (`InitializeComponent()`), **טעינה וקישור** של האוסף שבנינו ל `ComboBox` כדלהלן :
- ```
cbHostList.ItemsSource = busLines;
cbHostList.DisplayMemberPath = " BusLineNum ";
cbHostList.SelectedIndex = 0;
ShowBusLine(.....)
```
- הסבר :  
מקור הנתונים של ה `ComboBox` הוא אובייקט מסוג האוסף, התכונה שתוצג היא `BusLineNum` בלבד, האיבר הנבחר בהתחלה הוא האיבר הראשון שברשימה  
המתודה `ShowBusLine` צריכה להציג את המידע על קו האוטובוס, רשימת התחנות ועוד.



## צעד 8 : הצגת מסלול הקו בבחירת קו אוטובוס דרך ה ComboBox

1. **הוסיפו** למחלקת `MainWindow` שדה פרטי שישמור את הקו הנוכחי שמוצג.

```
private BusLine currentDisplayBusLine;
```

2. **הגדירו** את האירוע שיתרחש בעקבות בחירה של מארח דרך ה `ComboBox` :

- בחרו עם העכבר את ה `ComboBox` בחלון הגרפי ונלחץ עליו לחיצה כפולה.
- באופן אוטומטי קיבלתם מעבר לקוד לפונקציה שתבצע בעקבות בחירה של פריט ממנו בזמן הריצה.
- תוכן הפעולה יכול להיות כזה :

```
private void cbBusLines_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
 ShowBusLine((cbBusLines.SelectedValue as BusLine).BusLineNum);
}
```

הסבר : כאשר מתרחש אירוע של שינוי הבחירה ב `ComboBox` יש קריאה למתודה שמציגה את הפרטים של הקו שנבחר. מבצעים down-casting לערך הנבחר ושולפים את מספר הקו שישלח למתודה שמציגה את הפרטים.

```
private void ShowBusLine(int index)
{
 currentDisplayBusLine = busLines[index];
 UpGrid.DataContext = currentDisplayBusLine;

 lbBusLineStations.DataContext = currentDisplayBusLine.Stations;
}
```

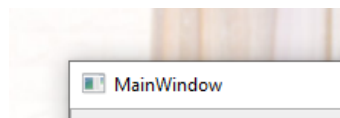
הסבר הפעולה :

הקו שנבחר להיות מוצג מתקבל מתוך האוסף שהגדרנו ע"י `indexern`.  
הערה : אם `indexern` שמימשתם מחזיר רשימה של קווי אוטובוס אתם צריכים להציג רק את הראשון : `busLines[index].First()`  
מעדכנים את רכיב `DataContext` ב `Grid` העליון וב `ListBox`.  
העדכון ב `Grid` ישפיע על רכיב `TextBox` שנמצא בתוכו.

3. **בצעו rebuild לפרוייקט והריצו אותו**. ראו את התוצאות שהתקבלו עד כה. כמעט סיימנו.

## צעד 9 : לסיום. נגיעות אישיות.

1. כעת חזרו שוב על הדברים – תוך שאתם מוסיפים להם "מגע" משלכם.
2. צבעים, סוגי פונטים, גדלים ועוד..
3. מה המידע שאתם מציגים עבור מסלול הקו?
4. שנו את שם ערך הכותרת שתופיע בראש החלון בפניה השמאלית. במקום `MainWindow` שתופיע כותרת משלכם למערכת.
5. שנו את האייקון (התמונה הקטנה) שתופיע לצד אותה כותרת לתמונה משלכם.
6. שימו לב שכשמריצים את היישום (נמצא בתיקיית `bin -> debug`) זהו האייקון.



😊 בהצלחה