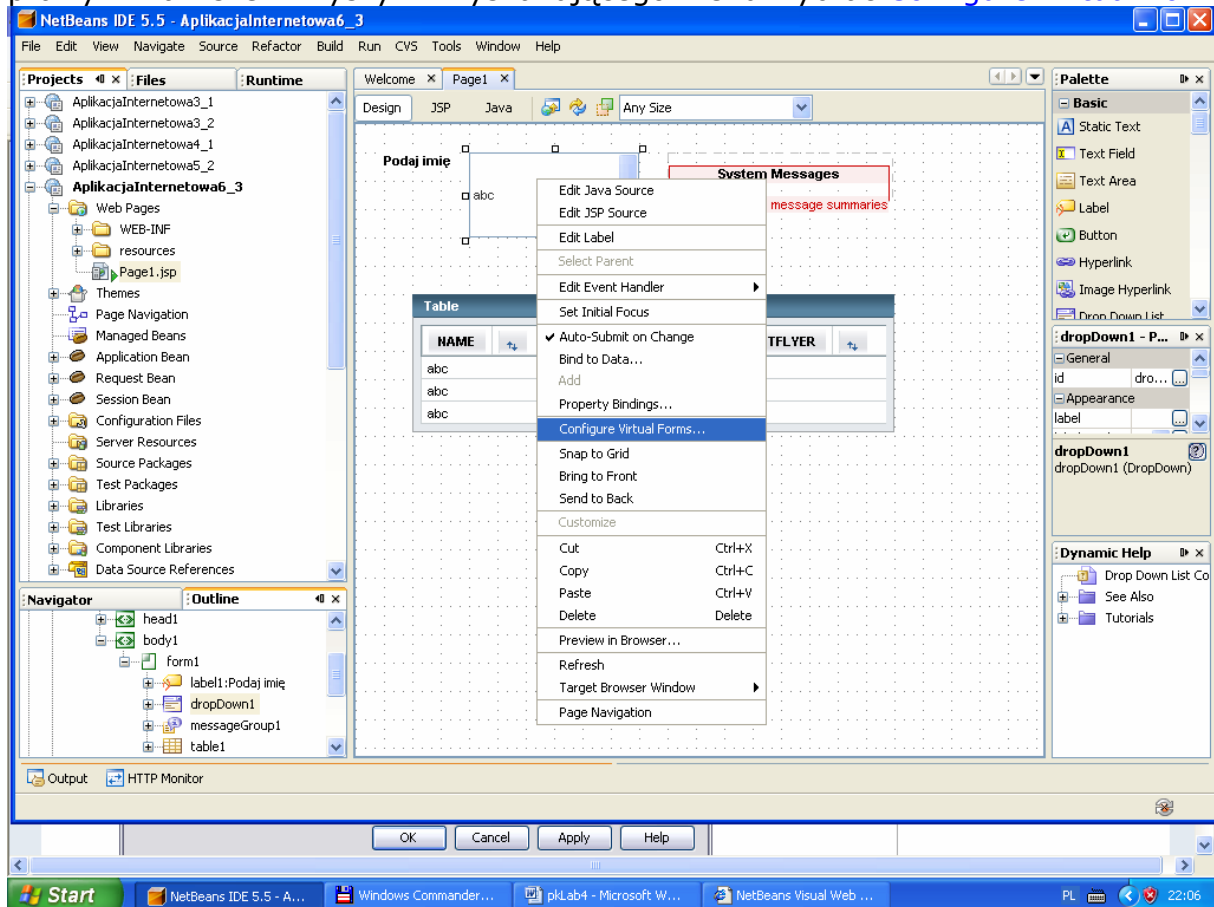


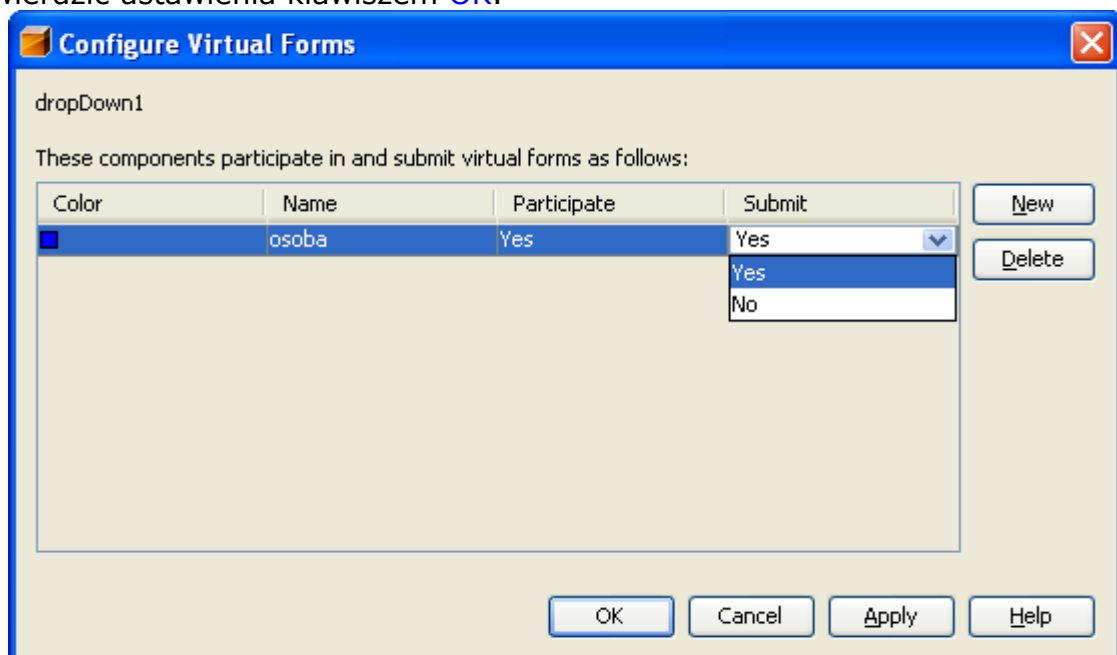
Lab4. – ciąg dalszy lab3.

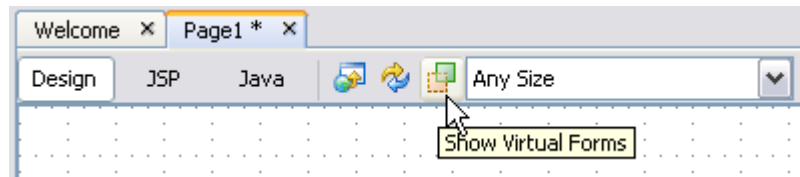
9. Konfiguracja wirualnego formularza ([Configure Virtual Forms](#))

9.1. W trybie **Design** strony **Page1** należy wybrać komponent **dropDown1** i kliknąć prawym klawiszem myszy. Z wyskakującego menu wybrać **Configure Virtual Forms**.

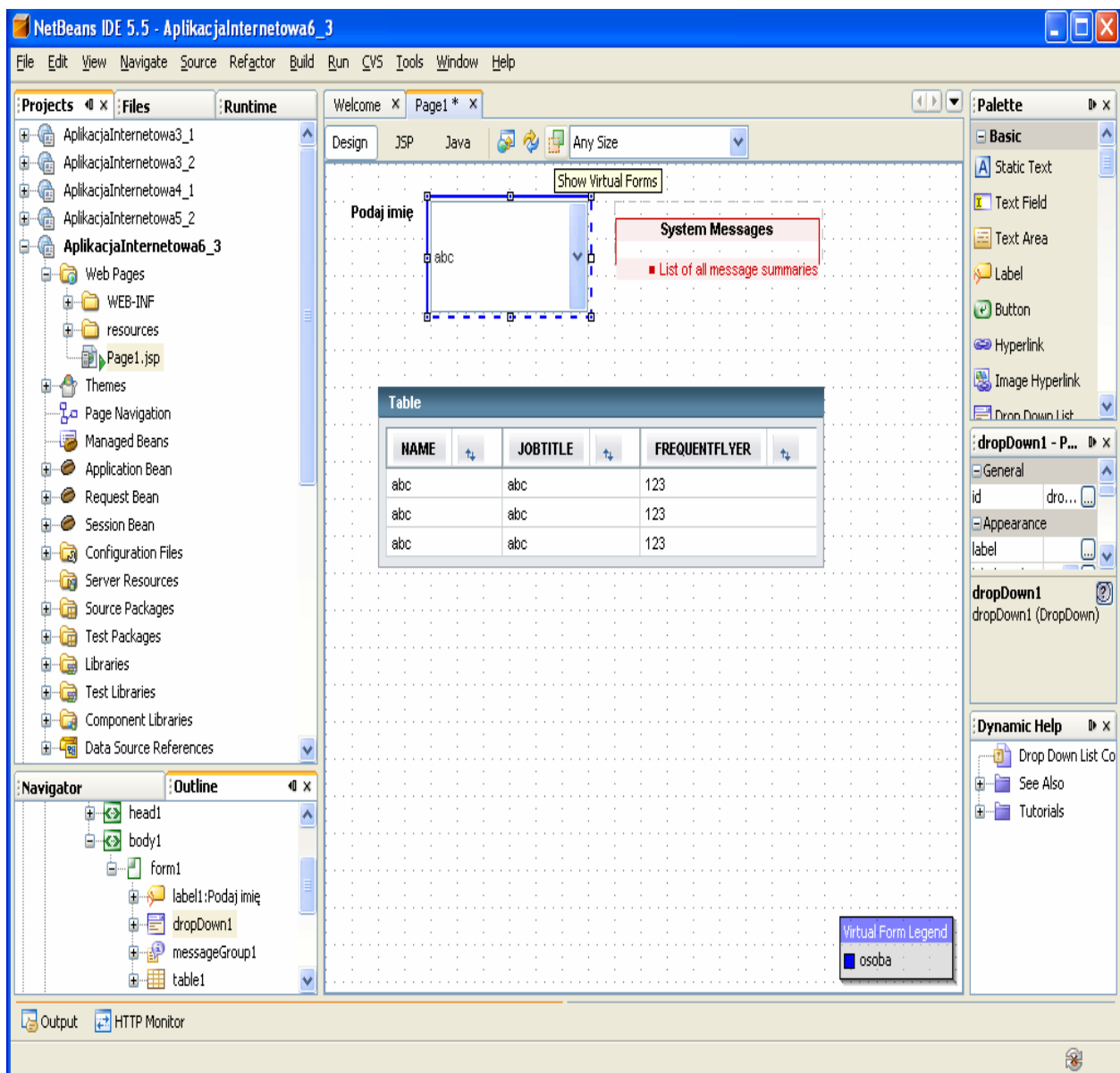


9.2. W formularzu opcji nacisnąć klawisz **New**. Nowy formularz nazwać np. **osoba** i oraz kliknąć dwa razy w kolumny **Participate** i **Submit** i w obu ustawić **Yes**. Zatwierdzić ustawienia klawiszem **OK**.



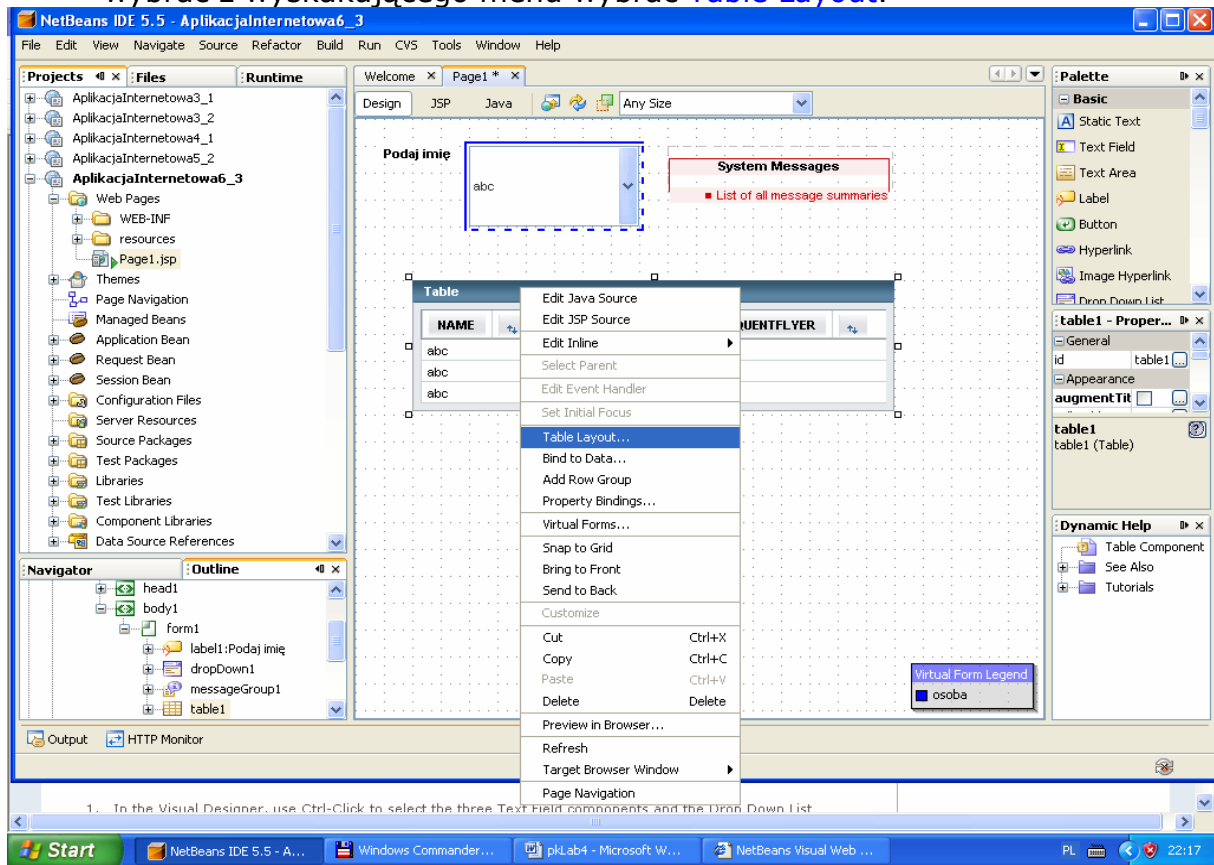


W wyniku ustawienia wirtualnego formularza nie są sprawdzane dane wyświetlane w komponencie `table1`. W trybie `Design` komponent należący do wirtualnego formularza jest podświetlony, pod warunkiem, że zostanie on pokazany po kliknięciu na ikonę pokazaną powyżej na belce narzędziowej.

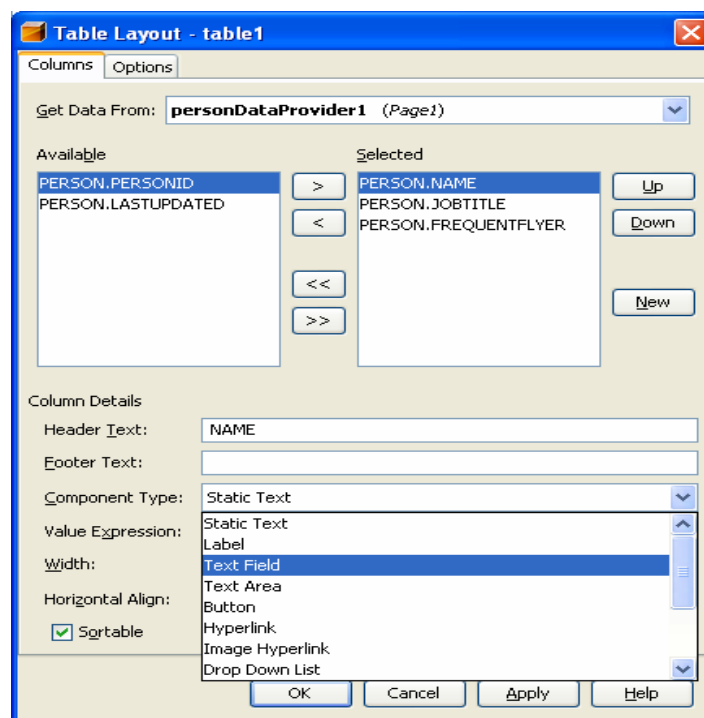


9.3. Wstawianie komponentów wejściowych do komórek komponentu [table1](#)

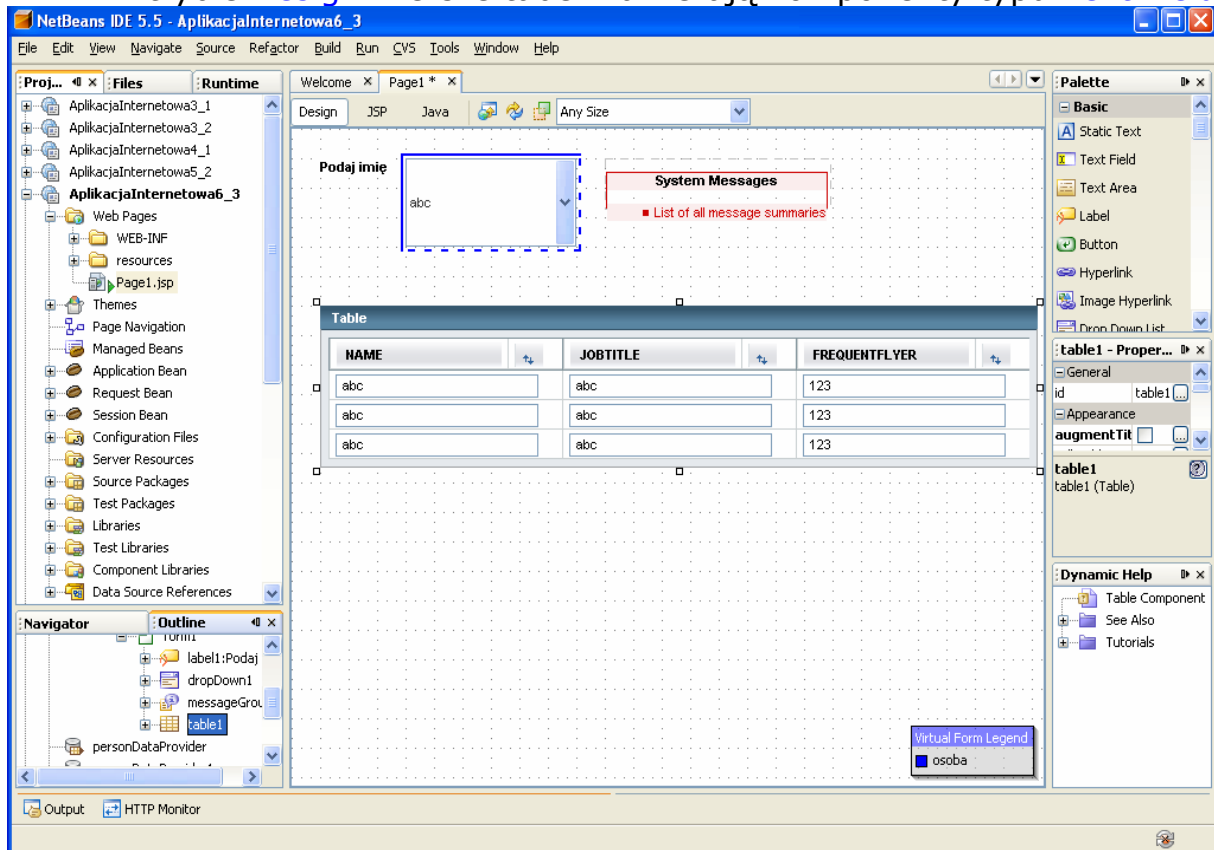
- Po zaznaczeniu komponentu [table1](#) należy kliknąć prawym klawiszem myszy i wybrać z wyskakującego menu wybrać [Table Layout](#).



- W formularzu wybrać kolejno kolumny tabeli w prawej części formularza i ustawić typ komponentu [Text Field](#) w liście Komponent [Type](#) i zatwierdzić kolejne ustawienia klawiszem [Apply](#). Zatwierdzić wszystko klawiszem [OK](#).

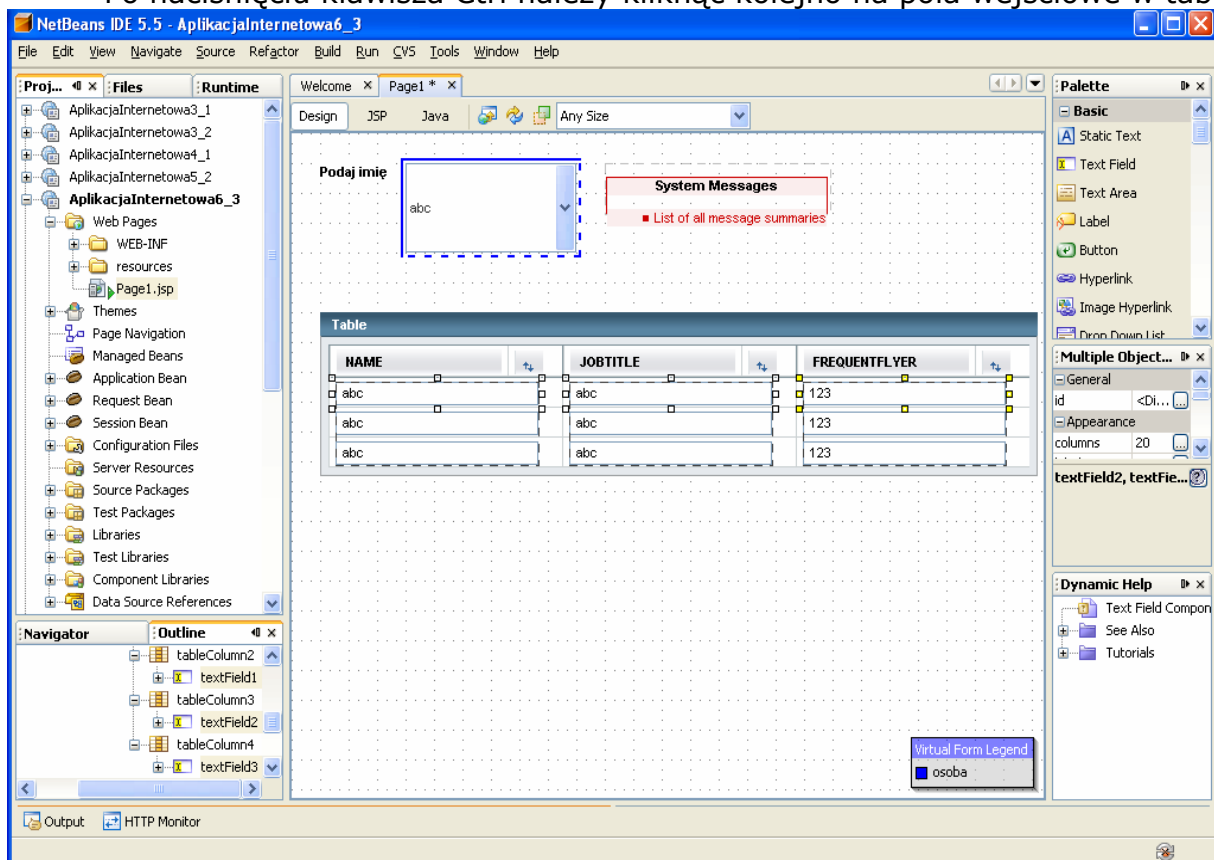


- W trybie Design wiersze tabeli zawierają komponenty typu Text Field

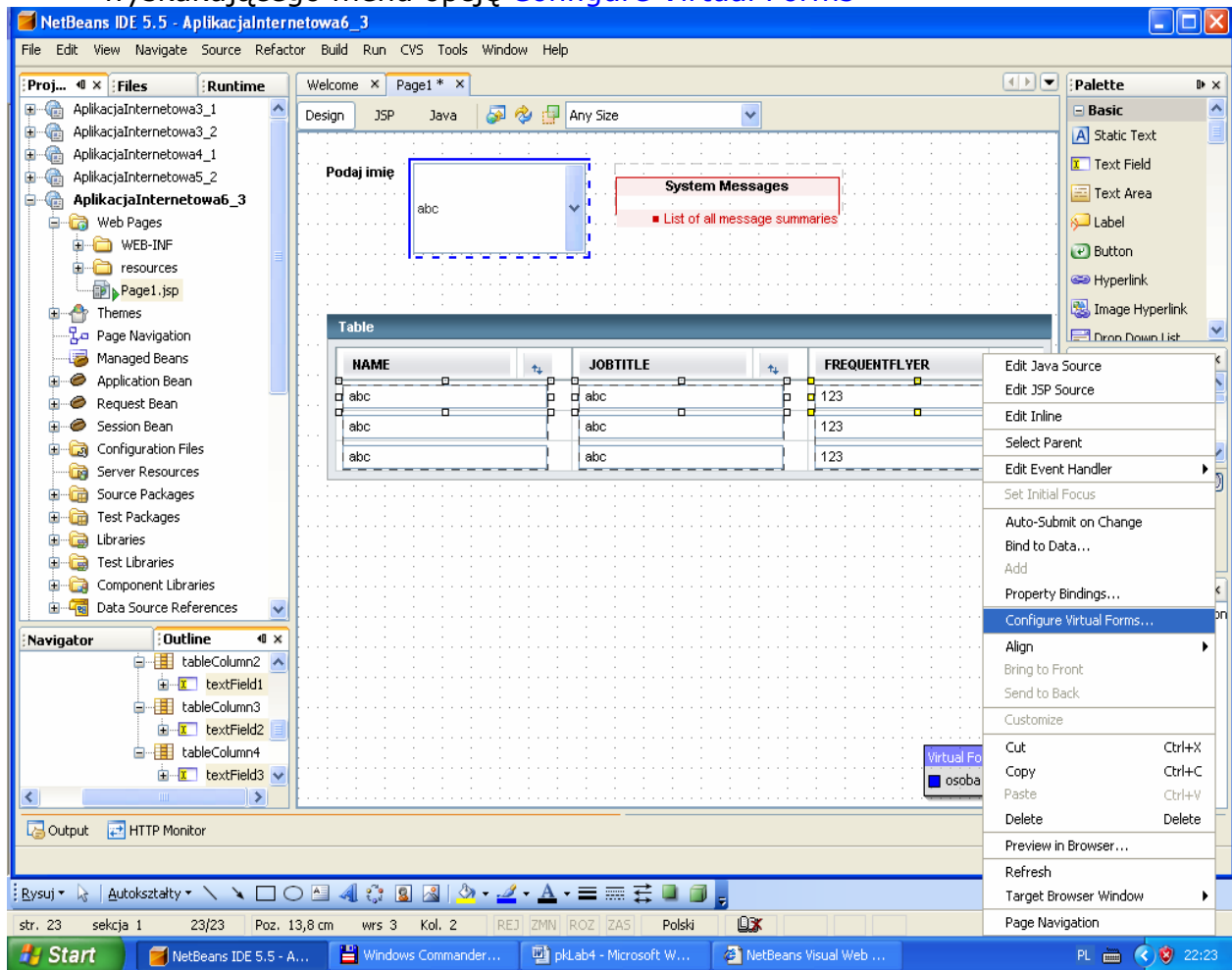


9.4. Utworzenie wirtualnego formularza dla pól wejściowych w komponencie table1

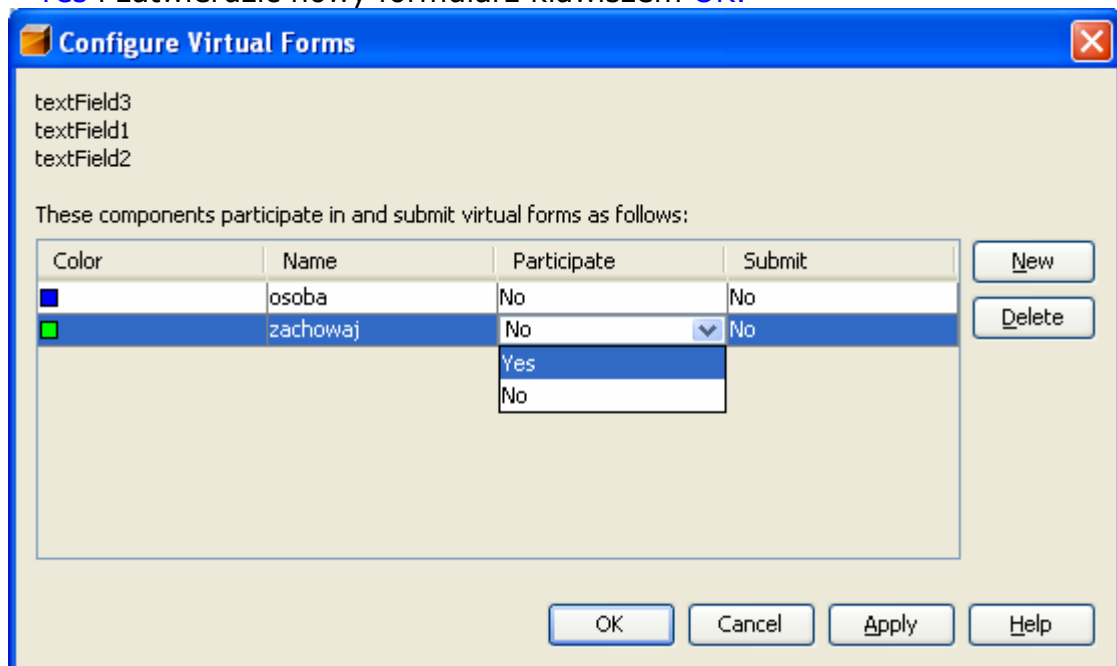
- Po naciśnięciu klawisza Ctrl należy kliknąć kolejno na pola wejściowe w table1



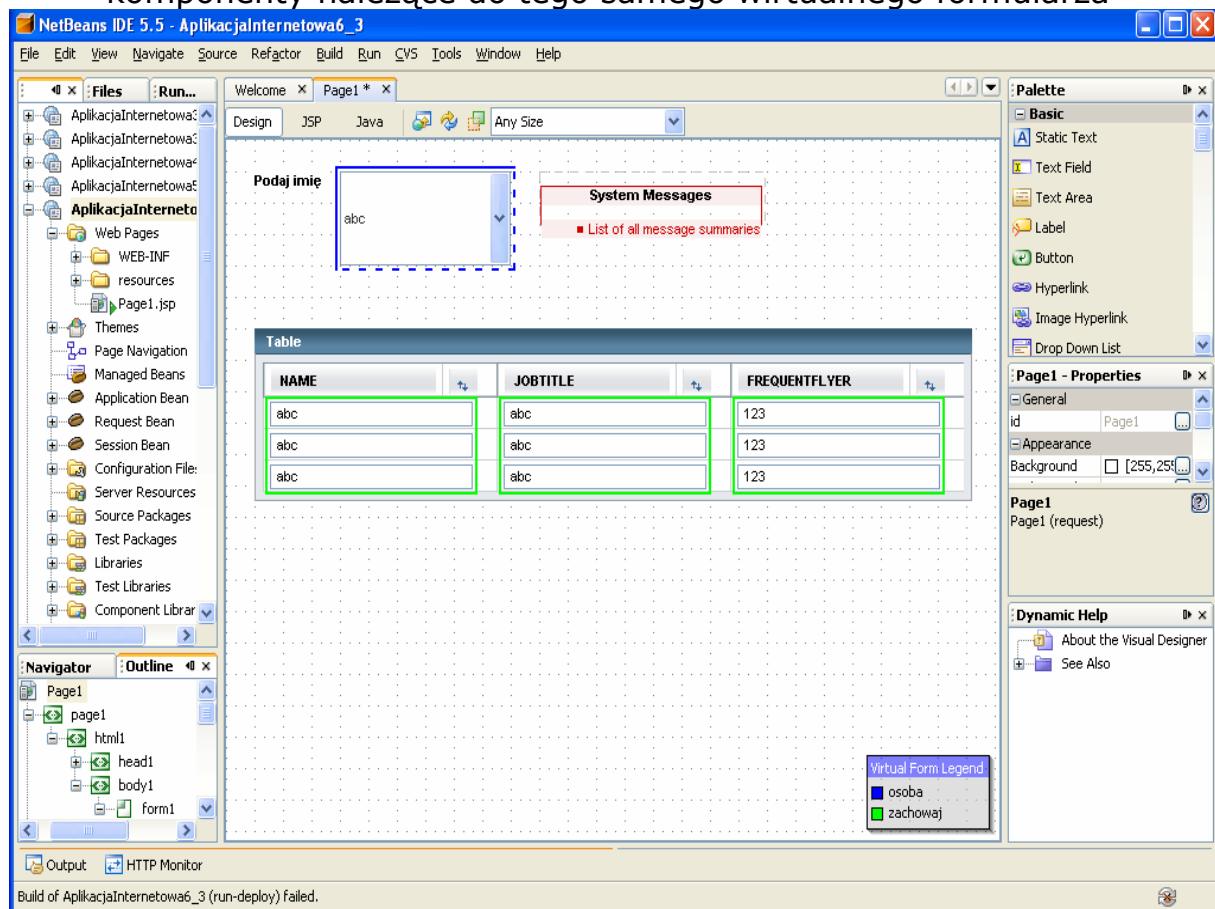
- Następnie po naciśnięciu prawego klawisza myszy należy wybrać z wyskakującego menu opcję **Configure Virtual Forms**



- Po naciśnięciu klawisza **New** należy wstawić nowy formularz np. o nazwie zachowaj. Należy podwójnie kliknąć dwukrotnie kolumnę **Participate** i ustawić **Yes** i zatwierdzić nowy formularz klawiszem **OK**.



- Na stronie **Page1** w trybie **Design** zaznaczone są odrębnymi kolorami komponenty należące do tego samego wirtualnego formularza



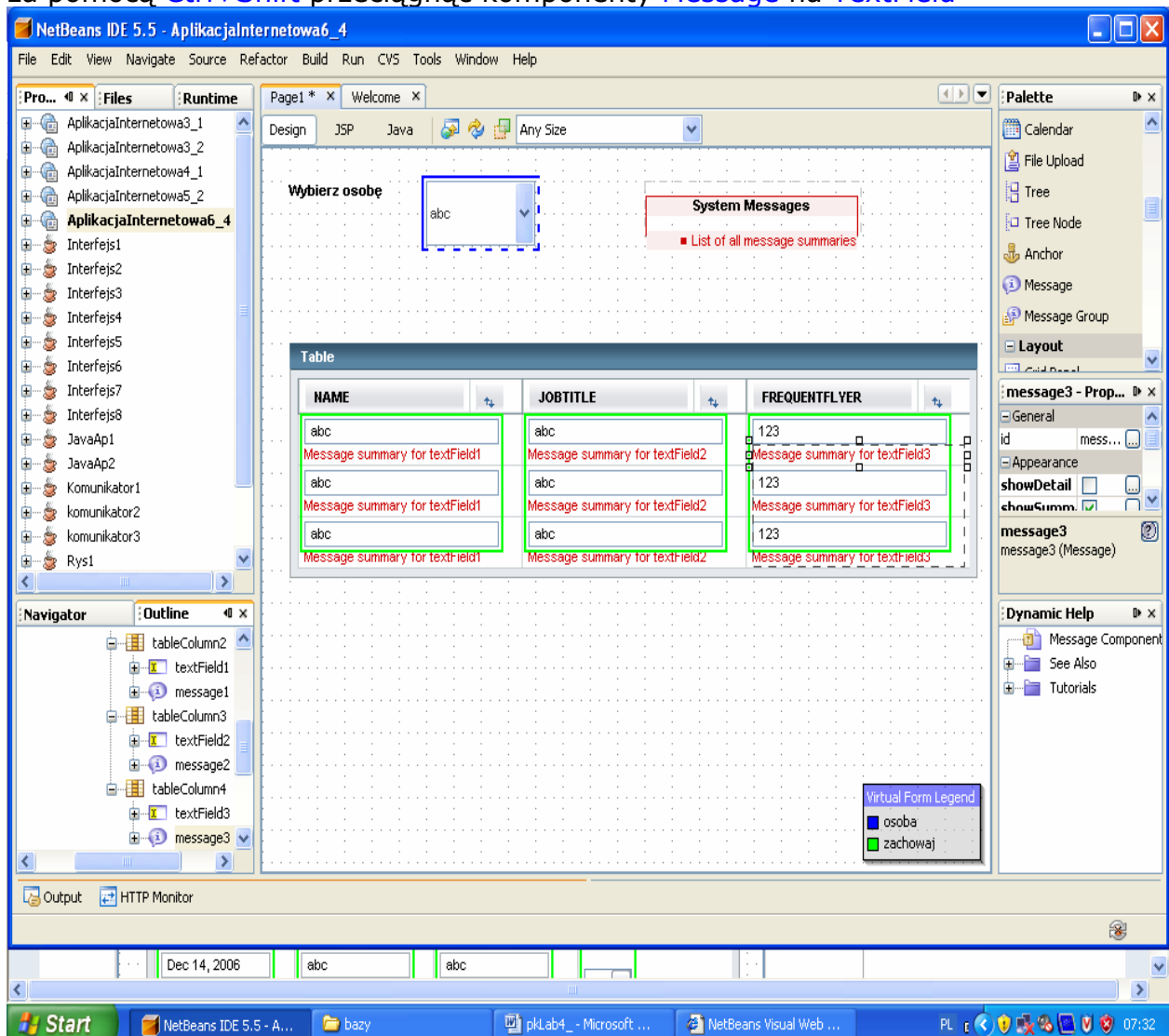
9.5. Uzupelnienie kodu źródłowego metody **dropDown1_processValueChange** dla zdarzenia typu **ProcessValueChange**
Plik **Page1.java**

```
public void dropDown1_processValueChange(ValueChangeEvent event)
{
    // TODO: Replace with your code
    try {
        getSessionBean1().getPersonRowSet1().setObject(
            1, dropDown1.getSelected());
        personDataProvider1.refresh();//odświeżenie zawartości komponentu table1
        form1.discardSubmittedValues("zachowaj");
        /*ta linia kodu powinna pojawić się po to, aby po wyborze nowej osoby ukazały się nowe dane
        w komponencie table1 */
        personDataProvider1.refresh();//odświeżenie zawartości komponentu dropDown1
    } catch (Exception e) {
        error("Cannot switch to person " + personDataProvider1.getValue(
            "PERSON.PERSONID"));
        log("Cannot switch to person " + personDataProvider1.getValue(
            "PERSON.PERSONID"), e);
    }
}
```

9.6. Należy uruchomić aplikację za pomocą **Build, Deploy i Run**. Po wyborze osoby z listy **dropDown1** w wierszu komponentu **table1** ukażą się pozostałe dane tej osoby.

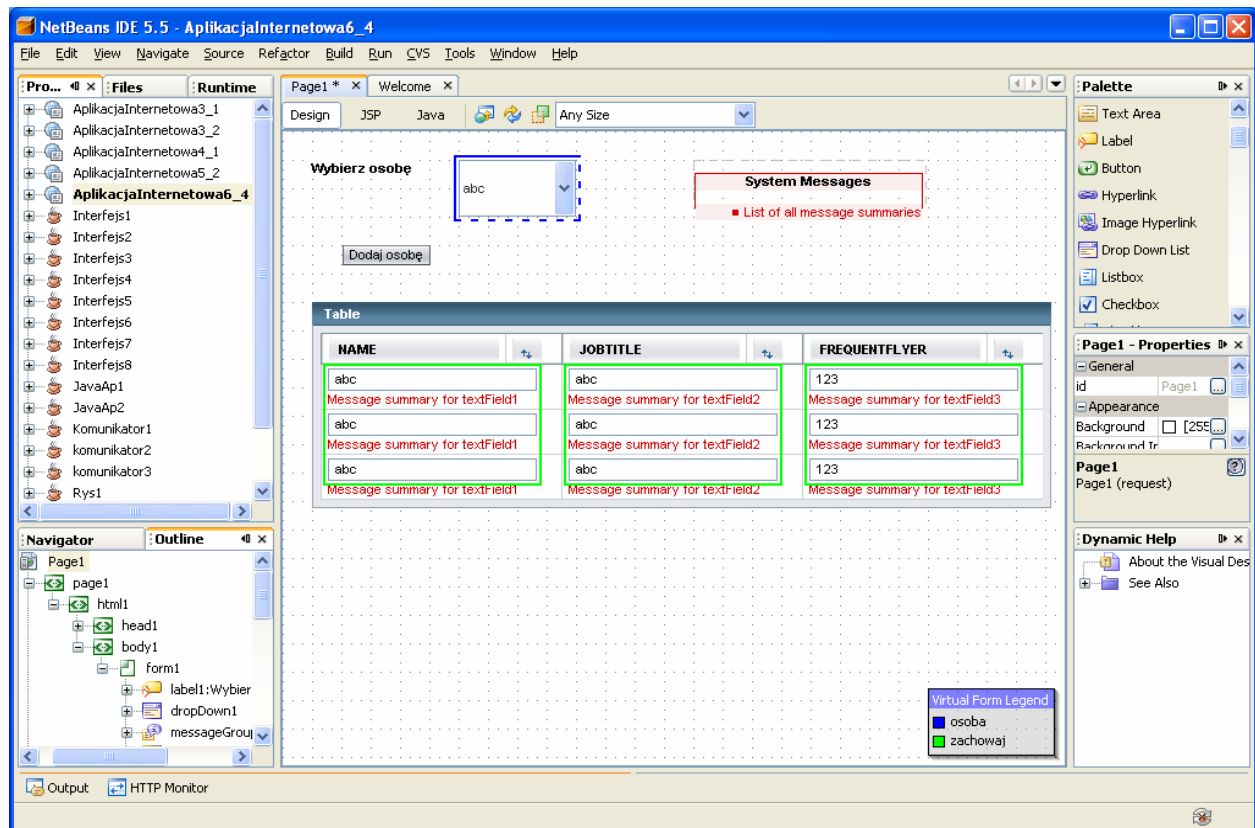


10. Należy przeciągnąć na każdy komponent typu **TextField** komponenty **Message** i za pomocą **Ctrl+Shift** przeciągnąć komponenty **Message** na **TextField**



11. Wstawienie przycisku **Dodaj osobę**

- Należy przeciągnąć komponent typu **Button** i napisać na nim **Dodaj osobę**.



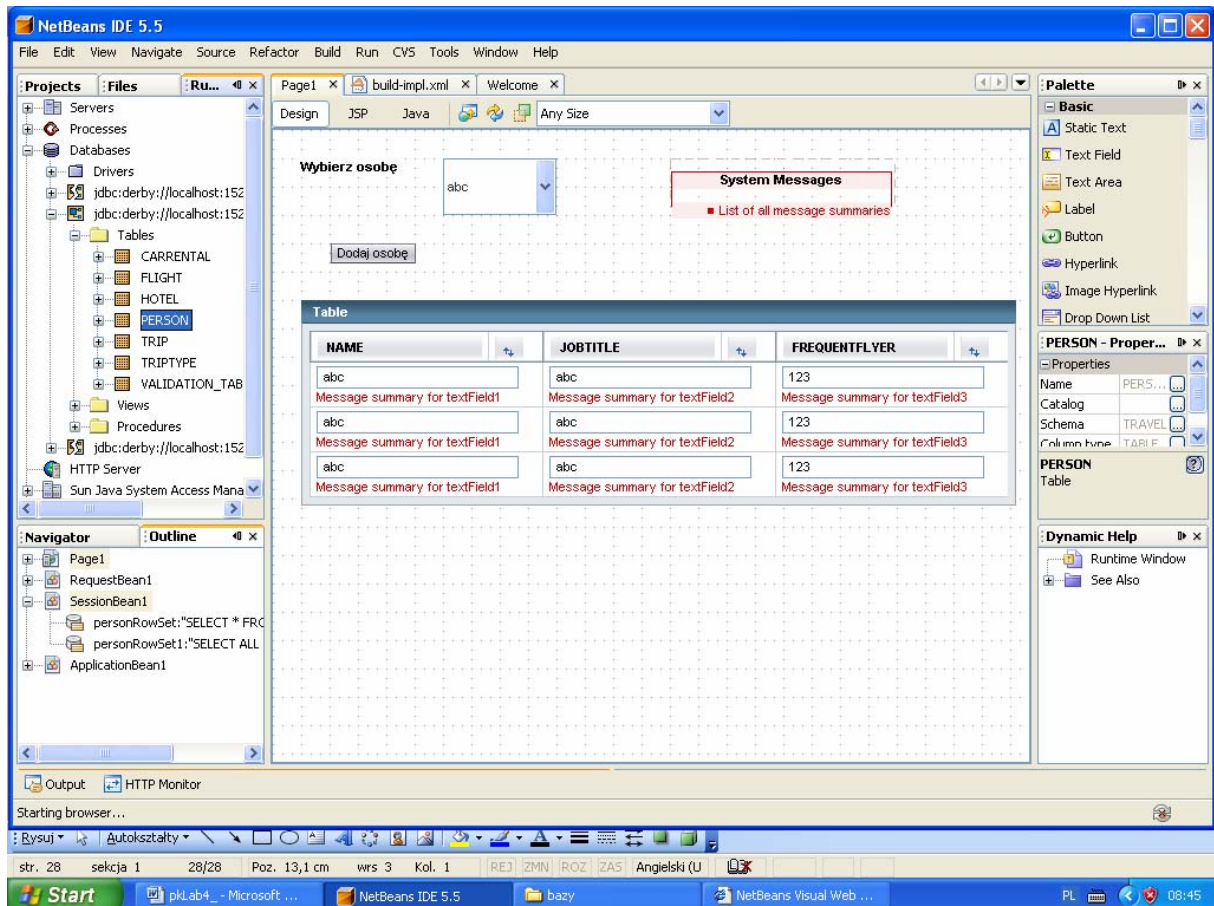
- Obsługa zdarzenia klawisza button1 **Dodaj osobę**. Po naciśnięciu klawisza **button1** pojawia się nowy rowek w komponencie **table1**.

Plik Page1.java

```
public String button1_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try {
        RowKey rk = personDataProvider1.appendRow();//dodanie nowego rowka
        personDataProvider1.setCursorRow(rk);//ustawienie kursora na nowym rowku
        personDataProvider1.setValue("PERSON.PERSONID", new Integer(0));
        //ustawienie zerowej wartości PERSONID w nowym rowku
    } catch (Exception ex) {
        log("Error Description", ex);
        error(ex.getMessage());
    }

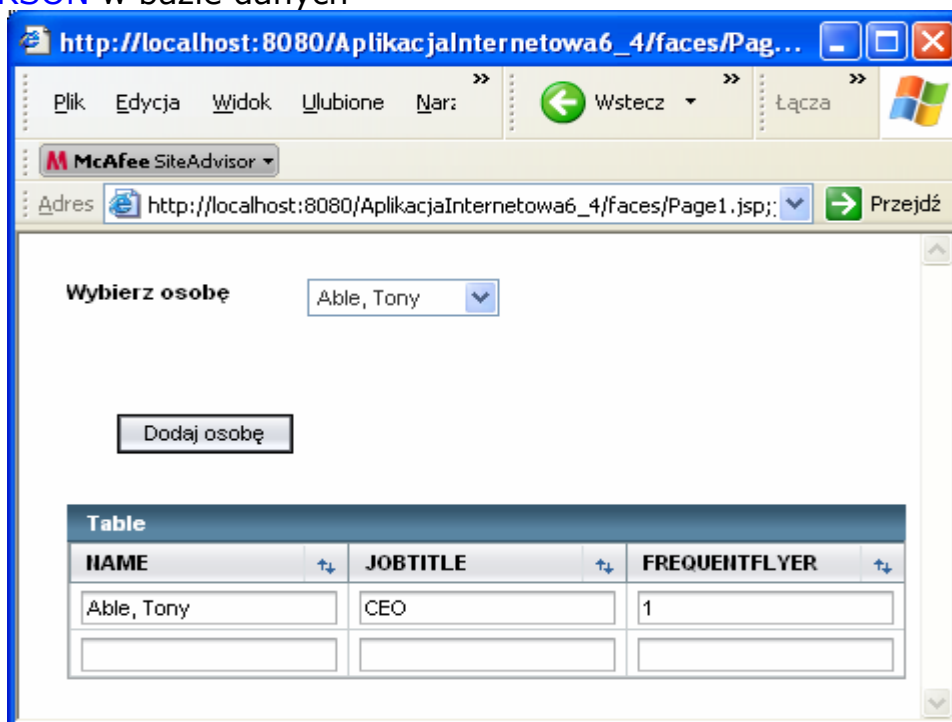
    return null;
}
```

Należy wykonać import `import com.sun.data.provider.RowKey;` za pomocą klawiszy **Ctrl+Alt+F**



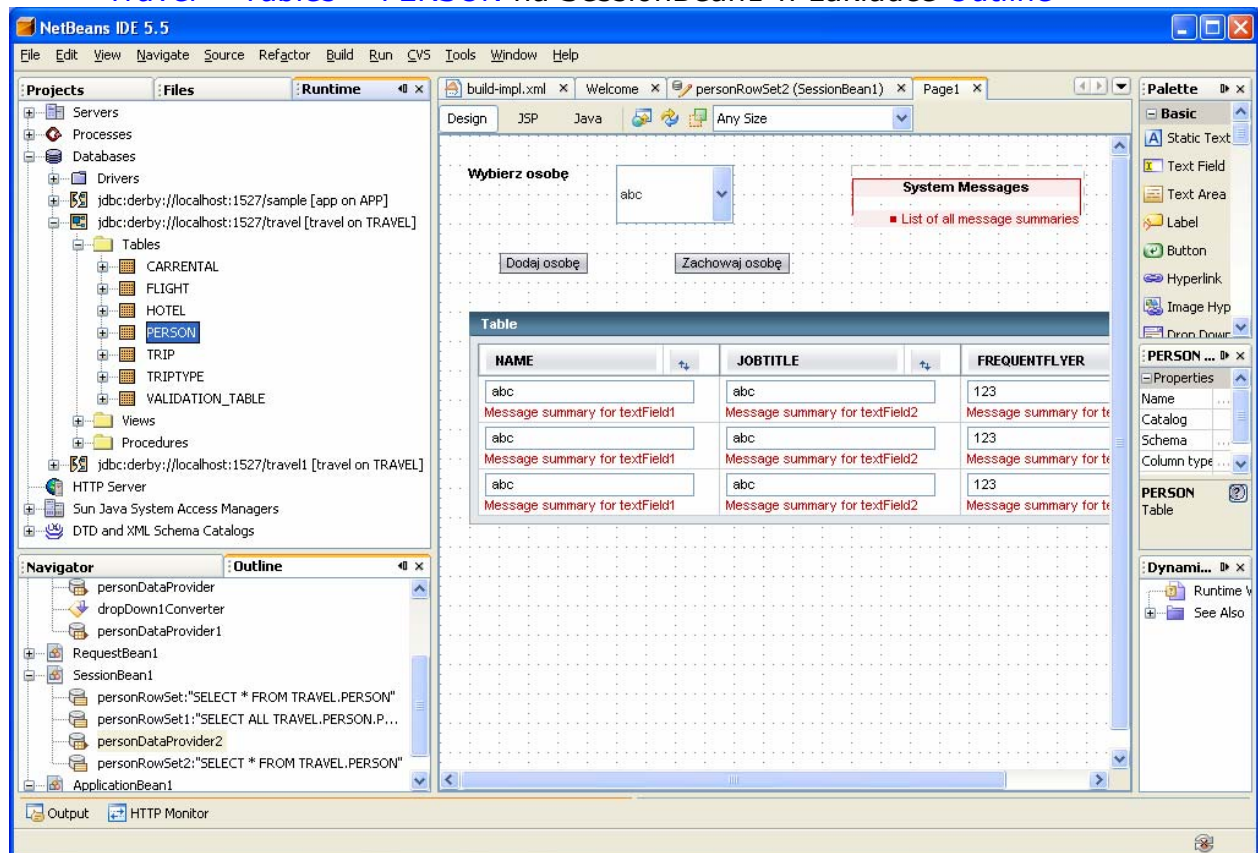
12. Należy uruchomić aplikację za pomocą **Build, Deploy i Run**. Po wyborze osoby z listy **dropDown1** w wierszu komponentu **table1** ukażą się pozostałe dane tej osoby.

- Po kliknięciu klawisza **Dodaj osobę** ukaże się pusty rowek w komponentcie **table1**. Dane wpisane tam nie są zapisywane do tabeli **PERSON** w bazie danych

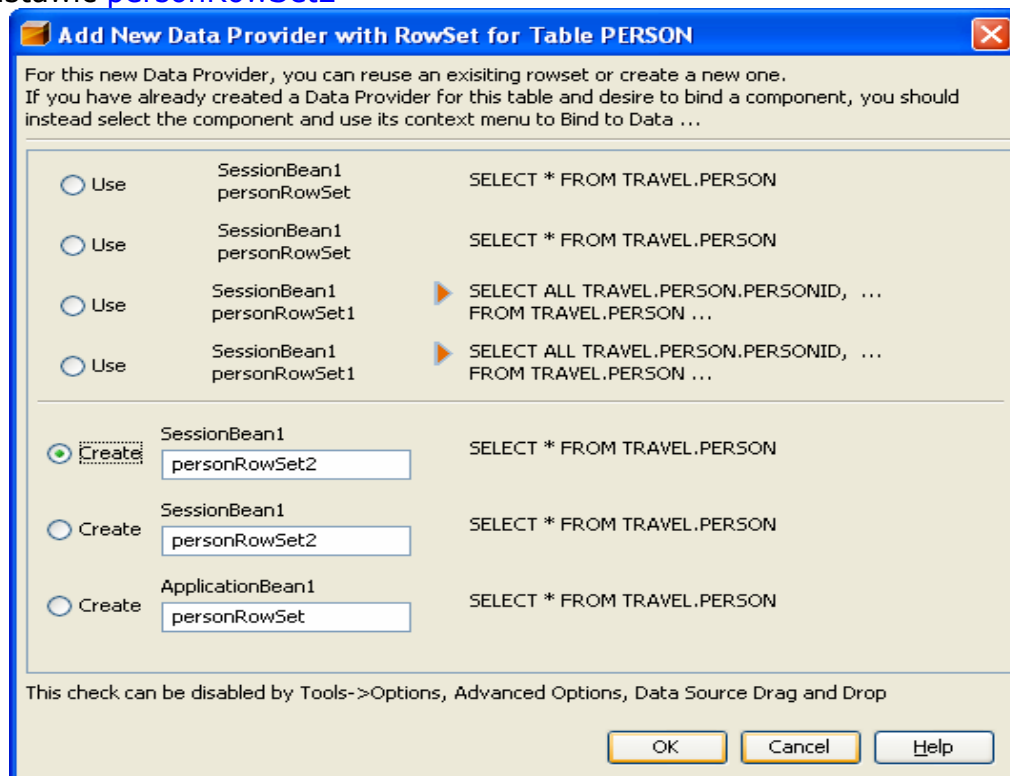


13. Zapisanie do tabeli nowej osoby

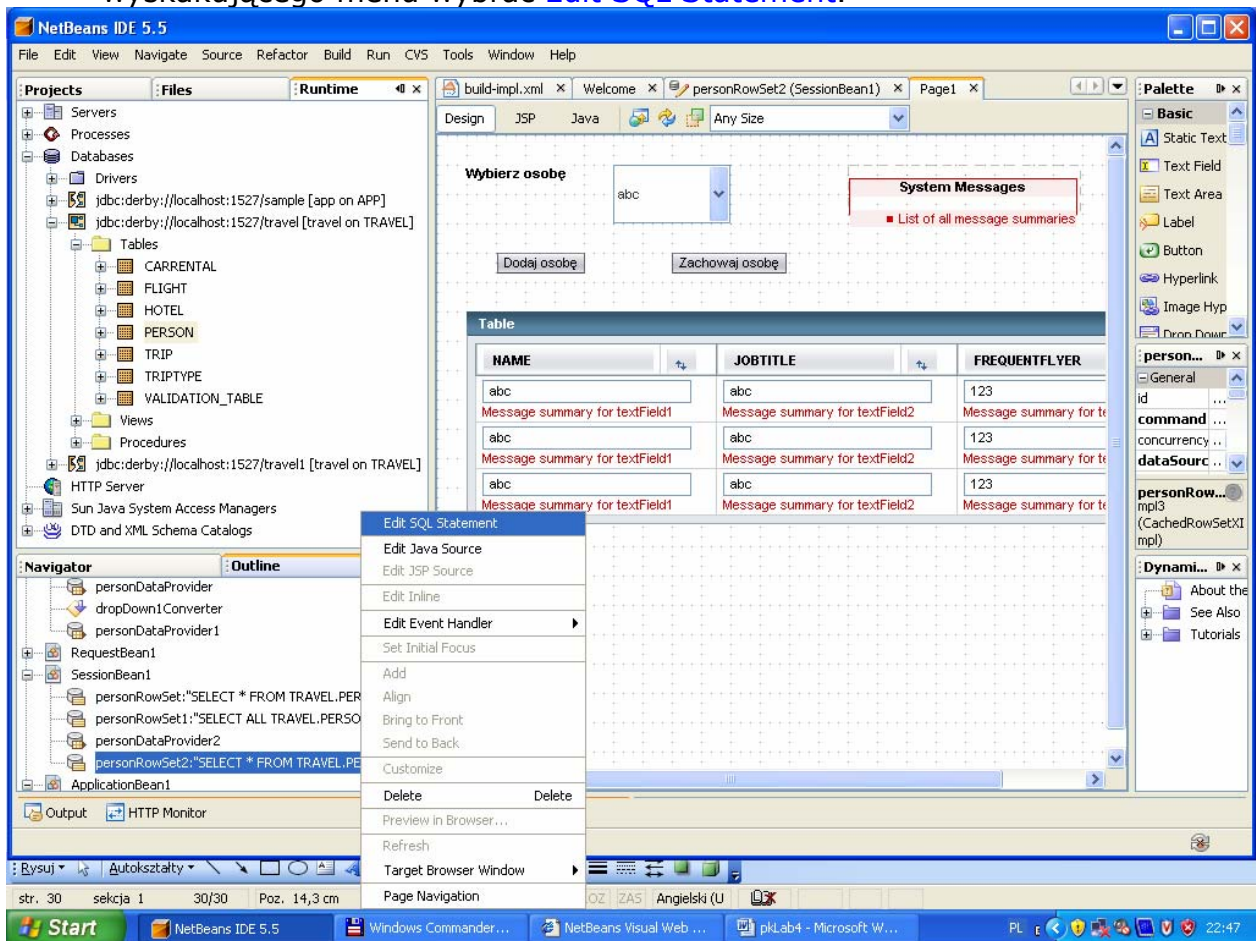
- Należy przeciągnąć klawisz **Button** i napisać na nim np. **Zachowaj osobę**
- Należy przeciągnąć tabelę **PERSON** z zakładki **Runtime** i opcji **Databases > Travel > Tables > PERSON** na **SessionBean1** w zakładce **Outline**



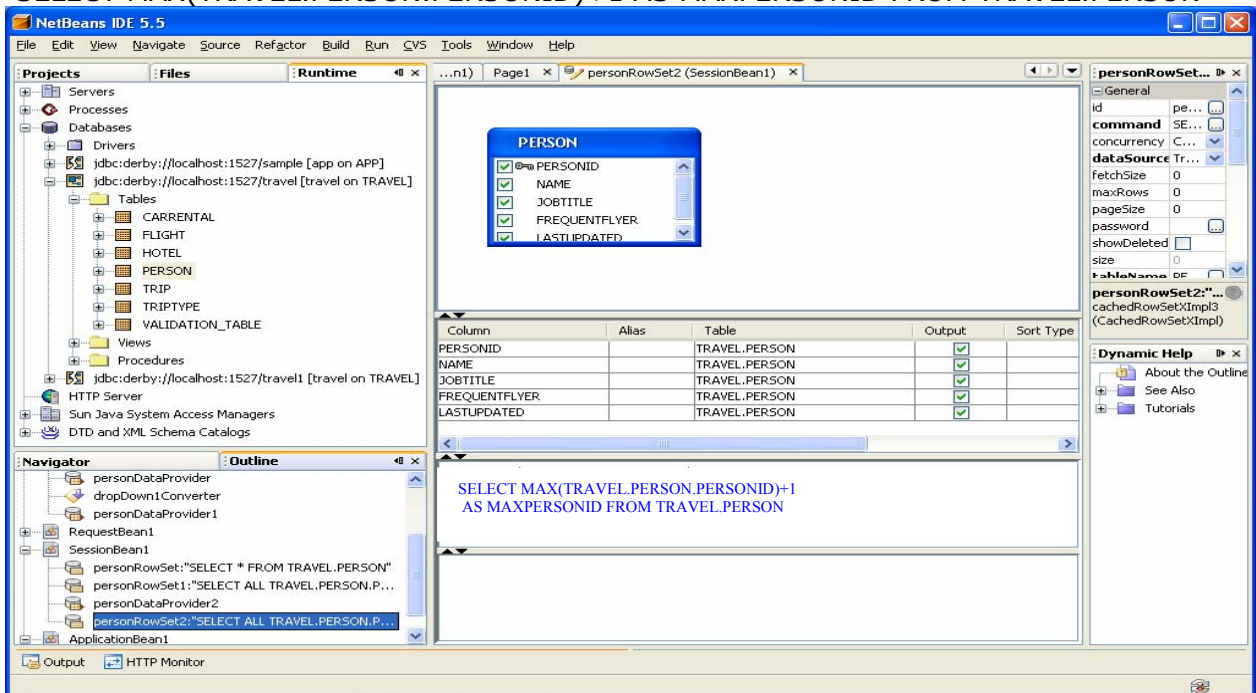
- W ukazanym formularzu wybrać za pomocą klawiszy typu **RadioButton** ustawić **personRowSet2**



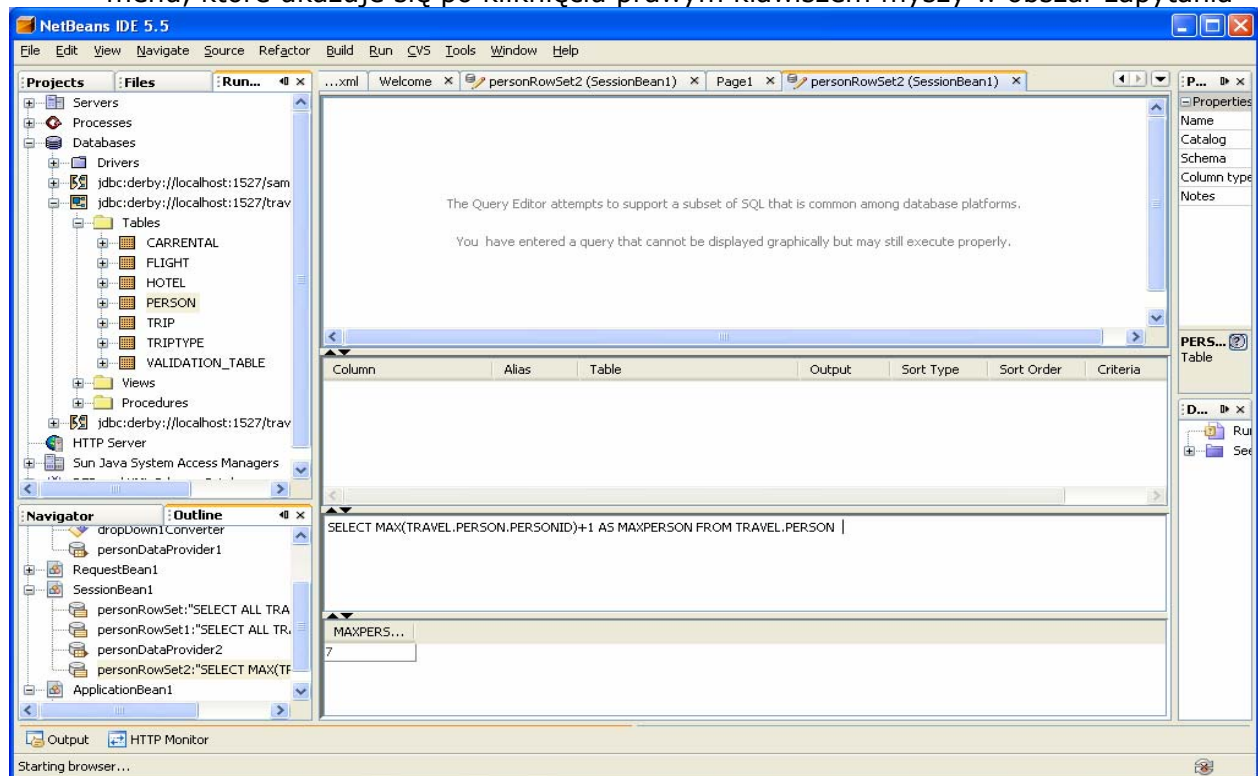
- Należy wykonać edycję zapytania SQL dla `personRowSet2`. W tym celu należy kliknąć prawym klawiszem myszy na `personRowSet2` w okienku **Outline** i z wyskakującego menu wybrać **Edit SQL Statement**.



- Wpisać tekst zapytania, które wyznacza PERSONID dla nowej osoby (większe o 1 od największego dotąd PERSONID w tabeli PERSON)
`SELECT MAX(TRAVEL.PERSON.PERSONID)+1 AS MAXPERSONID FROM TRAVEL.PERSON`



- Wykonanie zapytania za pomocą opcji **Run Query**, wybranej z wyskakującego menu, które ukazuje się po kliknięciu prawym klawiszem myszy w obszar zapytania



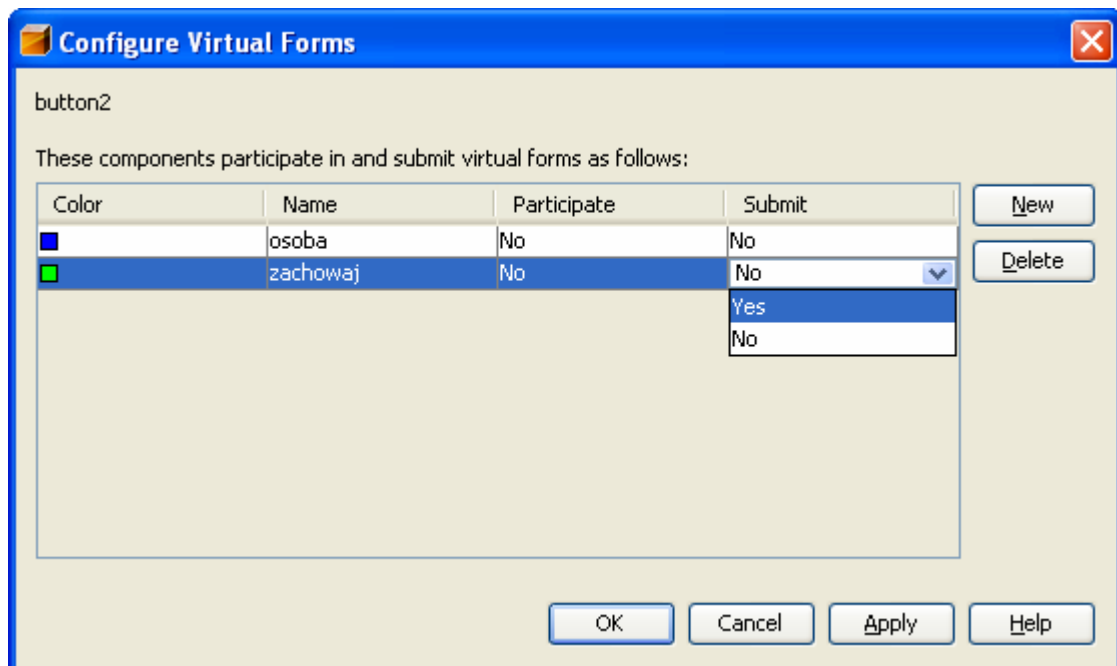
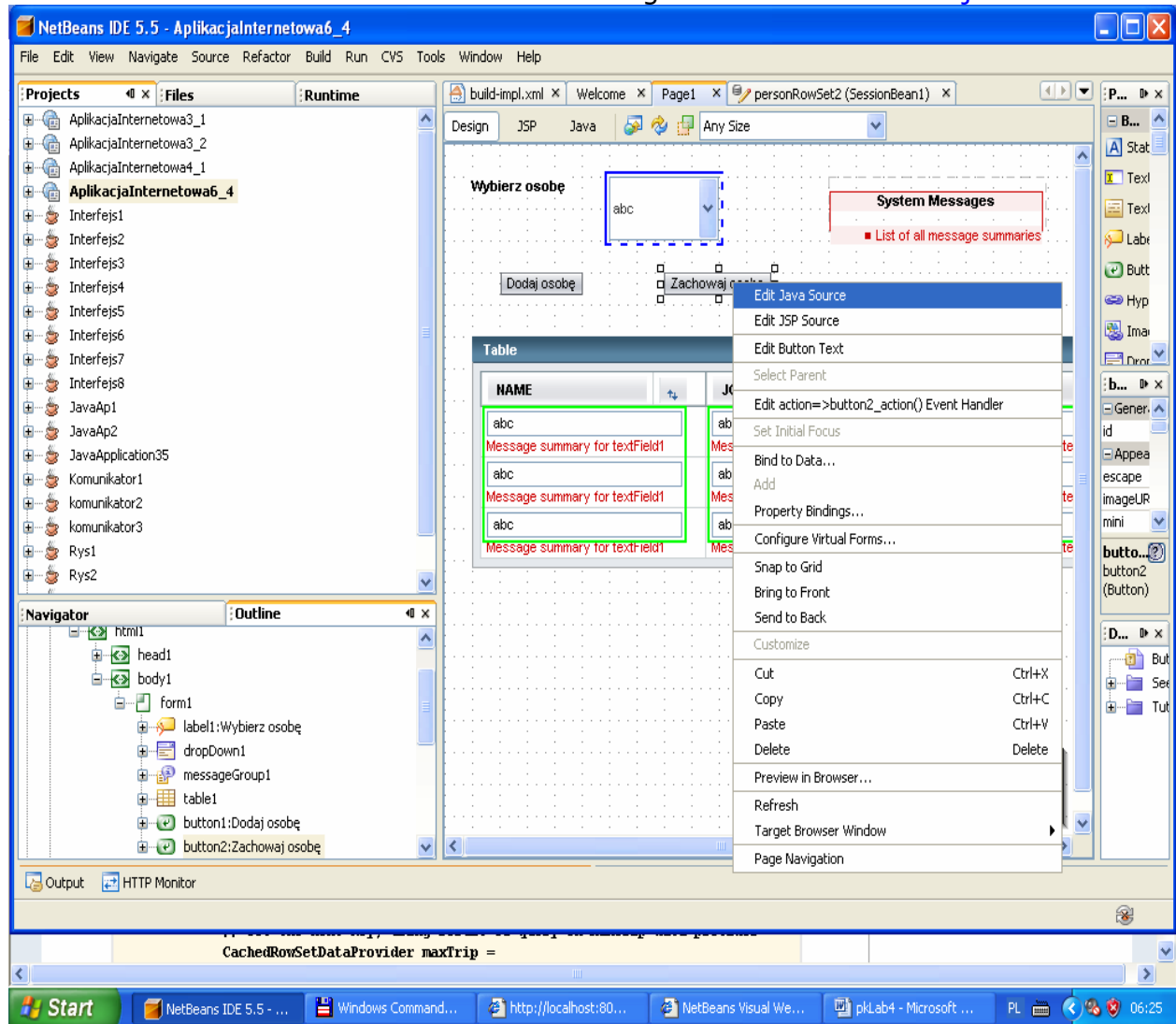
- Obsługa zdarzenia naciśnięcia klawisza **button2 Zachowaj osobę** w pliku Page1.java

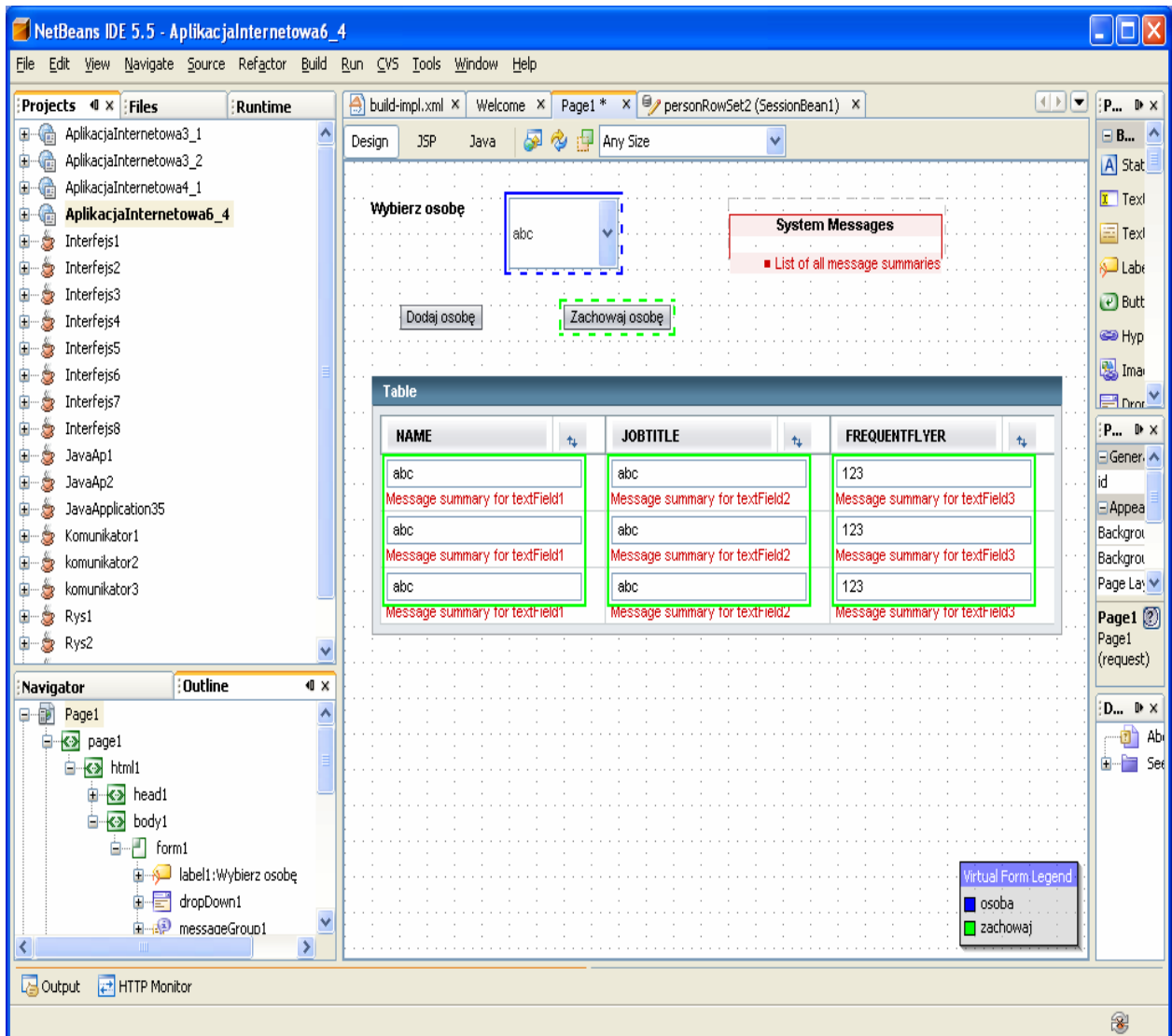
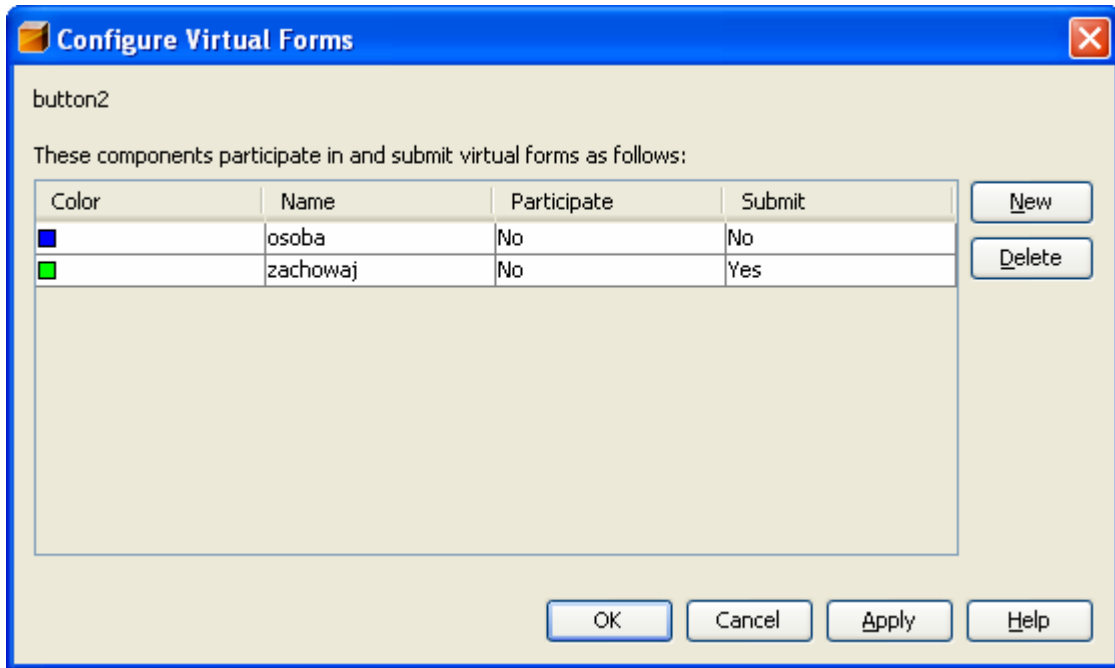
```

public String button2_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try { // Get the next key, using result of query on personDataProvider2()
        CachedRowSetDataProvider maxOsoba =
            getSessionBean1().getPersonDataProvider2();
        maxOsoba.refresh();
        maxOsoba.cursorFirst();
        int nowyOsobaId = ((Integer)maxOsoba.getValue("MAXPERSONID")).intValue();
        // Navigate through rows with data provider
        if (personDataProvider1.getRowCount() > 0)
        { personDataProvider1.cursorFirst();
            do {
                if ( personDataProvider1.getValue("PERSON.PERSONID").equals
                    (new Integer(0))) //po znalezieniu nowego rowka z PERSONID równym 0
                { personDataProvider1.setValue("PERSON.PERSONID",
                    new Integer(nowyOsobaId));
                    //wstawienie poprawnej wartości PERSONID obliczonej w zapytaniu wykonanym w obiekcie personRowSet2
                    nowyOsobaId++; }
                //wyznaczenie nowej wartości PERSONID dla kolejnego rowka, jeśli wprowadzono więcej niż jedną osobę
            } while (personDataProvider1.cursorNext());
        };
        personDataProvider1.commitChanges();//wprowadzenie do bazy wszystkich
            //zmian, nie tylko wynikających z nowo wprowadzonego rowka lub kilku nowych rowków
        personDataProvider.refresh(); //odświeżenie zawartości komponentu dropDown1
    } catch (Exception ex) {
        log("Error Description", ex);
        error("Error :"+ex.getMessage()); }
    return null; }

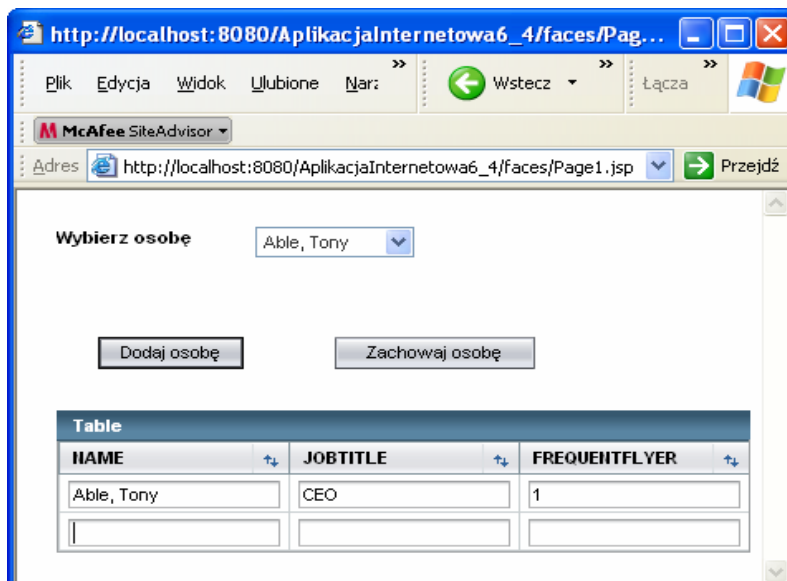
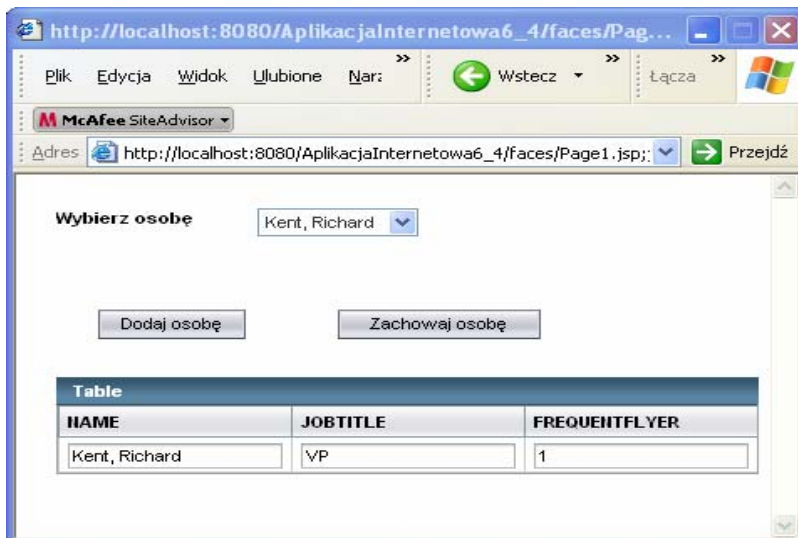
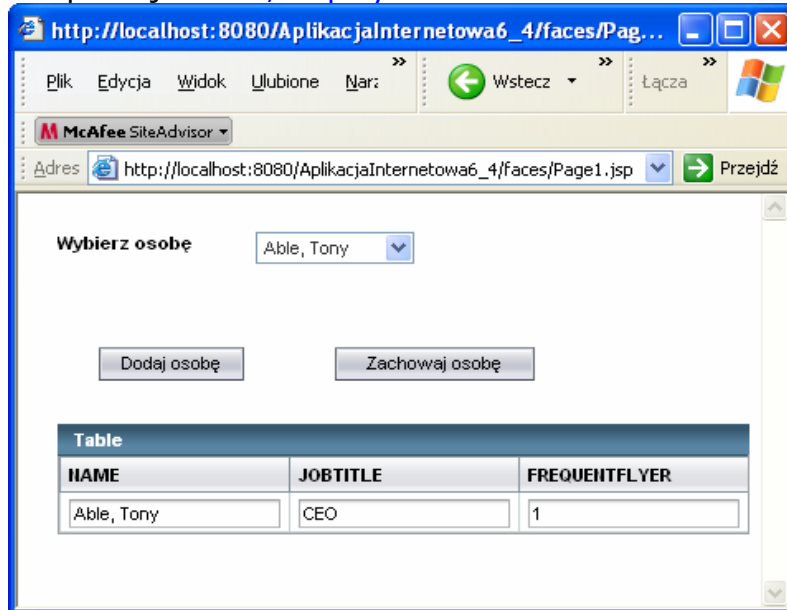
```

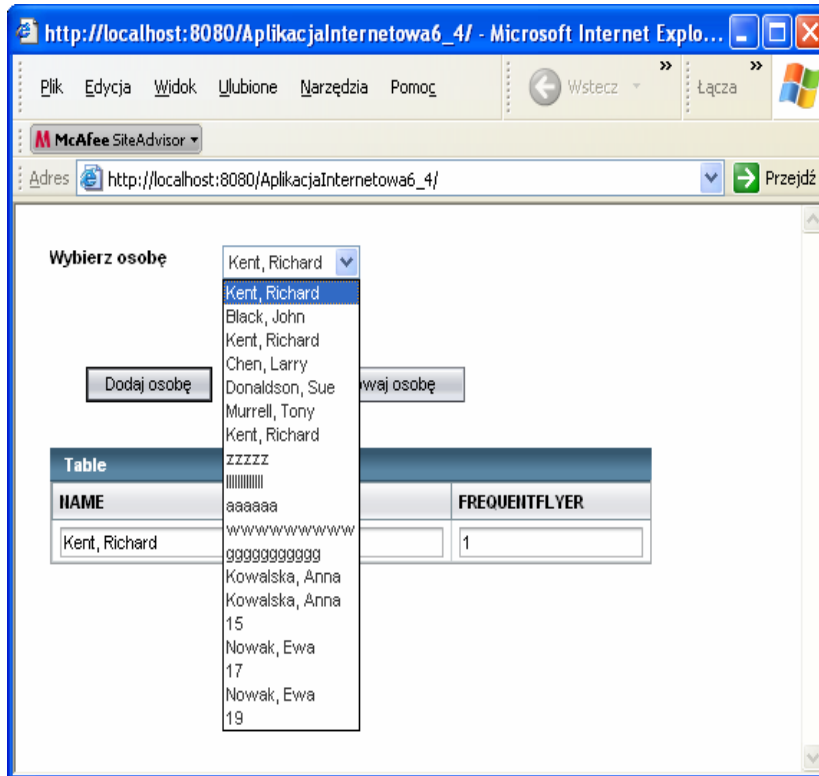
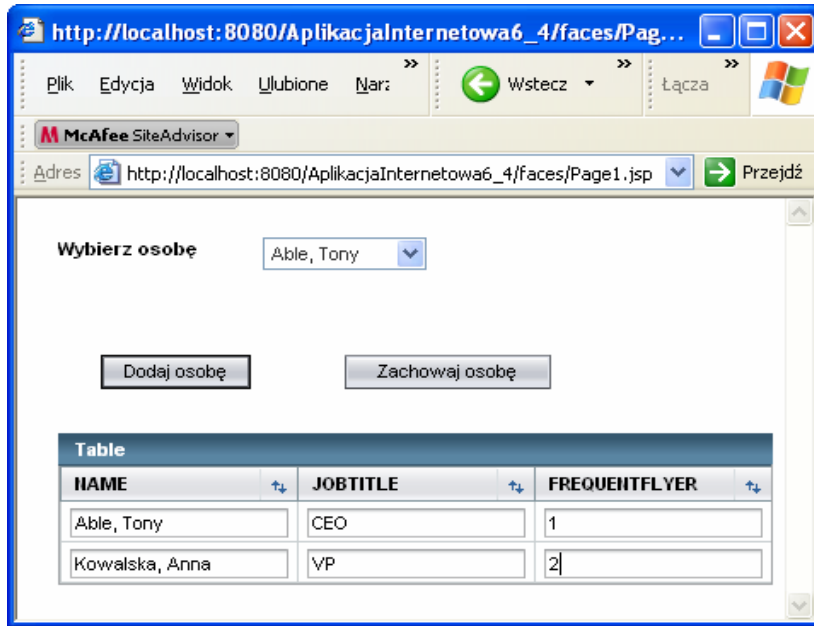
14. Wstawienie klawisza `button2` do wirtualnego formularza `zachowaj`



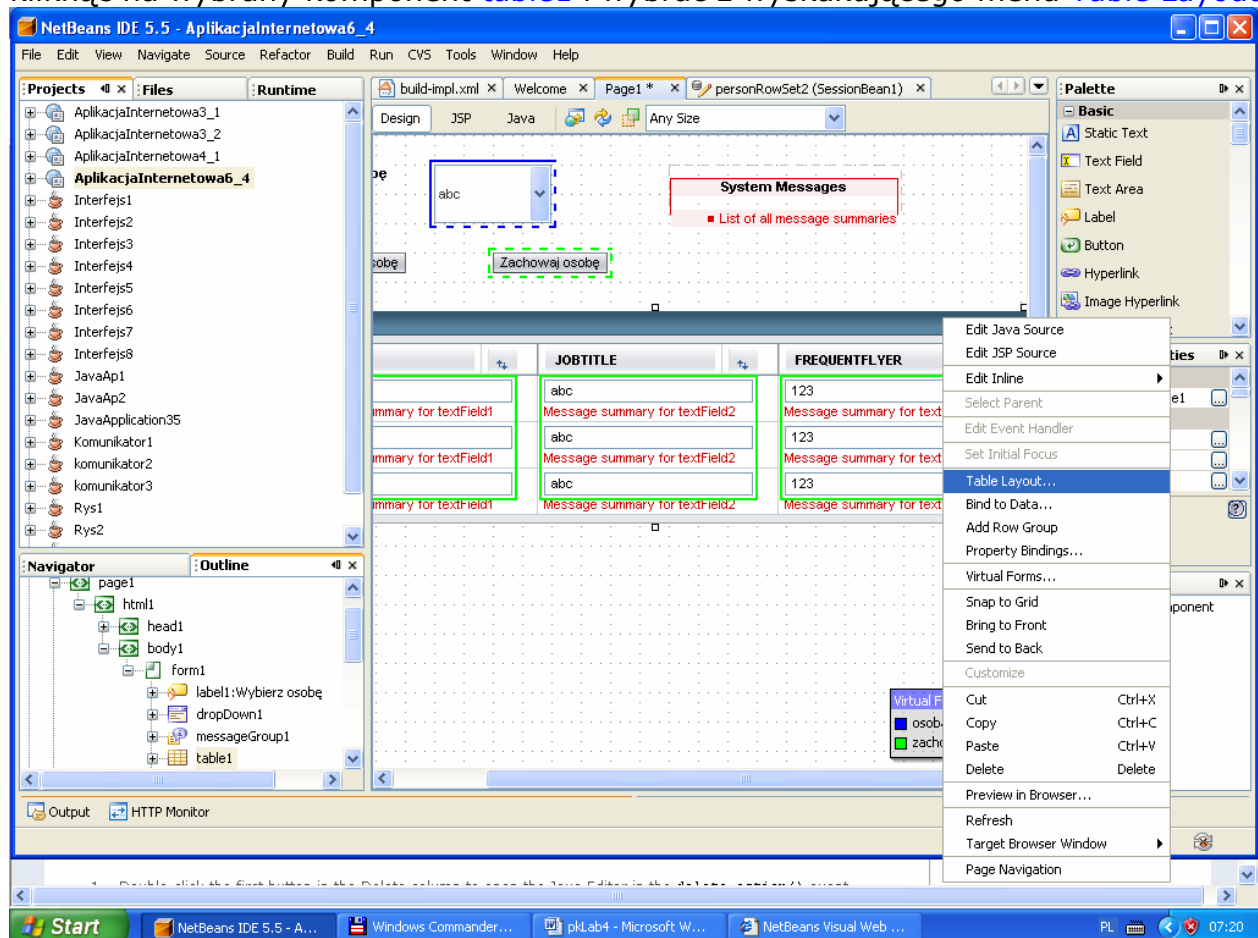


15. Uruchomienie aplikacji: Build, Deploy i Run

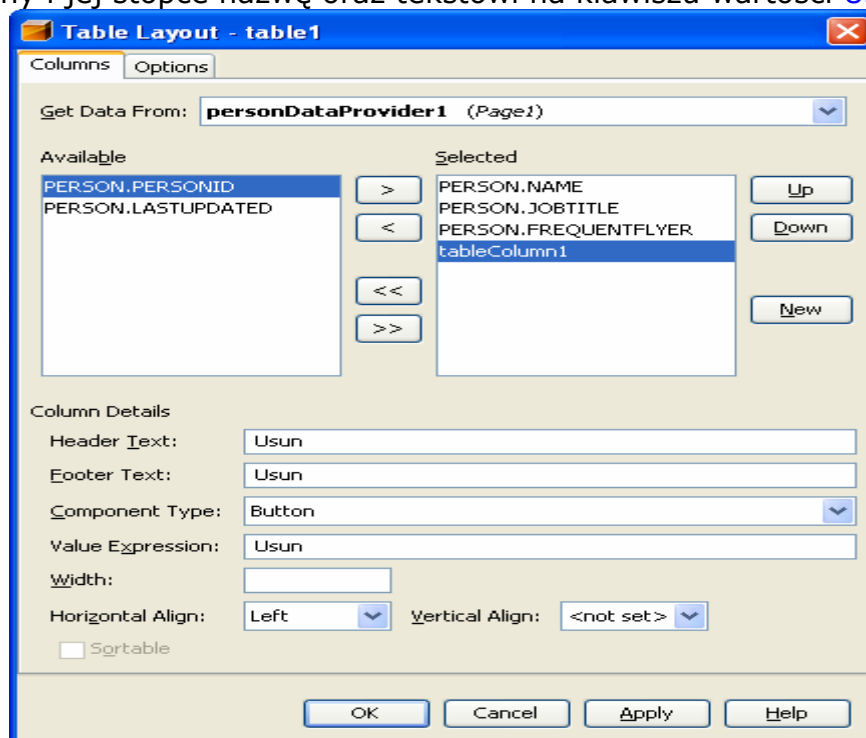




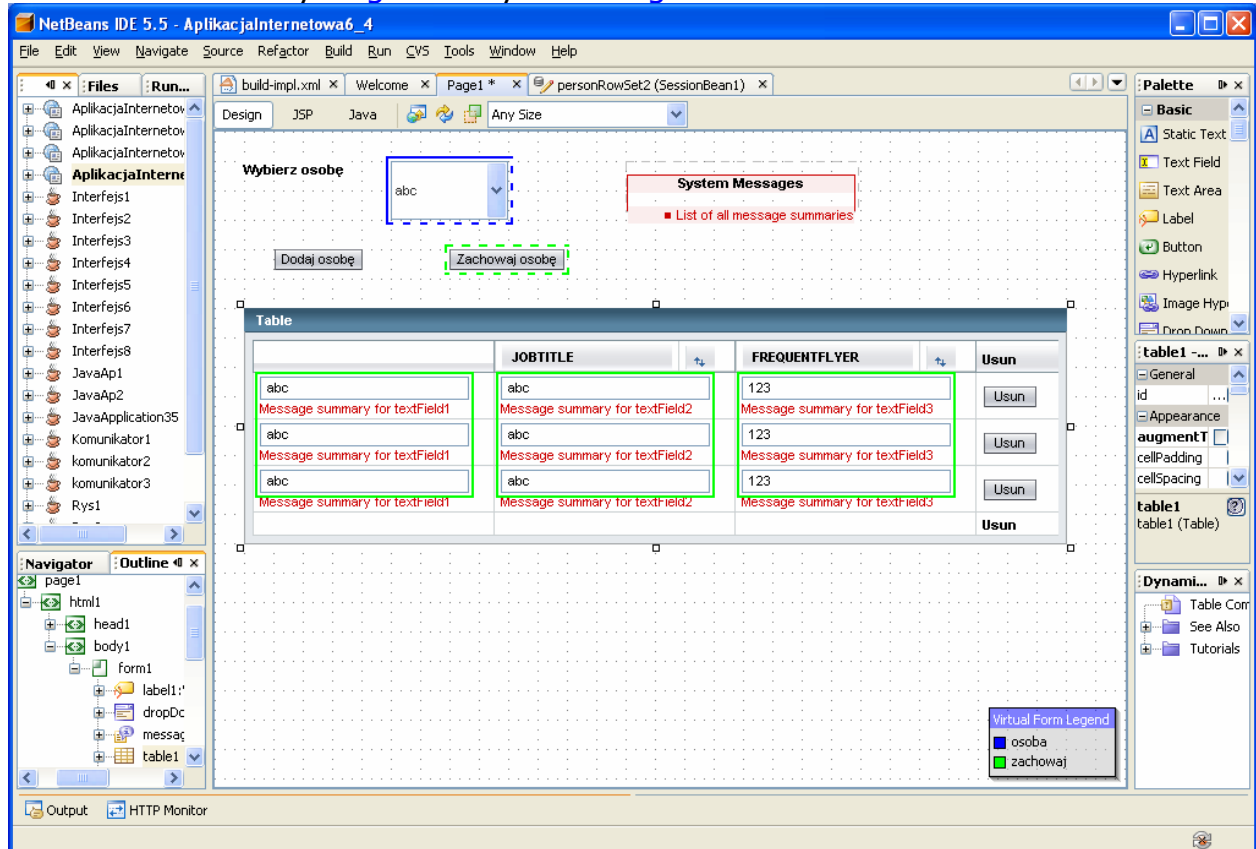
16. Wstawienie klawiszy usuwania danych – należy prawym klawiszem myszy kliknąć na wybrany komponent `table1` i wybrać z wyskakującego menu `Table Layout`



- Klawiszem `New` wstawić nową kolumnę do komponentu `table1`, wstaw do niej komponent typu `Button` w `Component Type`, nadaj nazwę nagłówkowi kolumny i jej stopce nazwę oraz tekstowi na klawiszu wartości `Usun`



- Widok strony Page1 w trybie Design

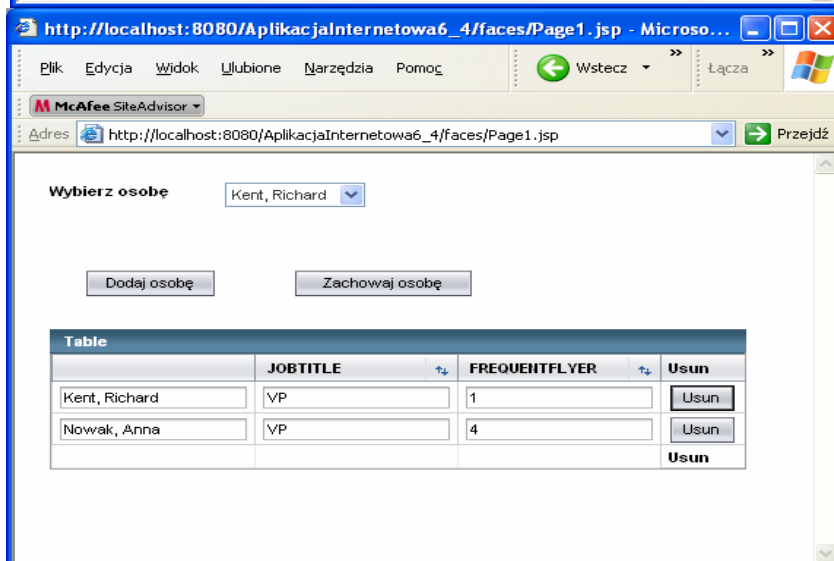
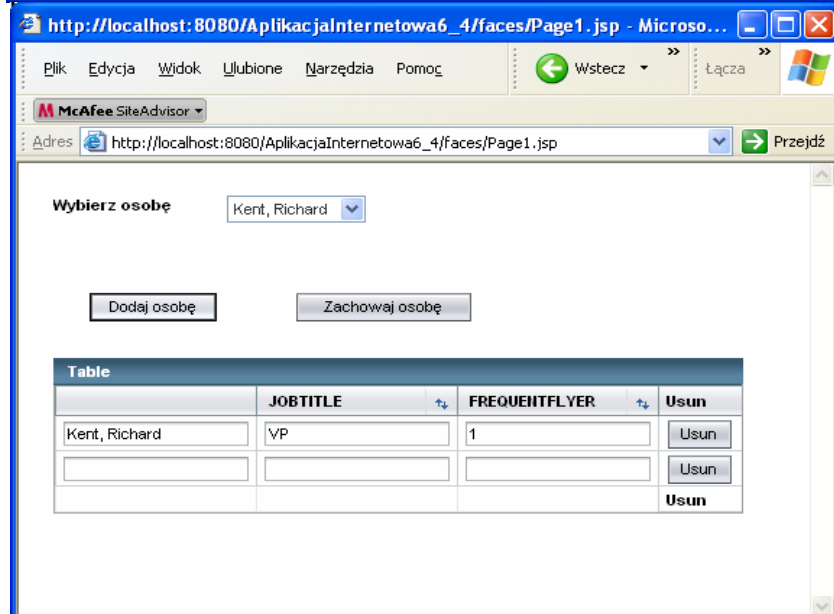
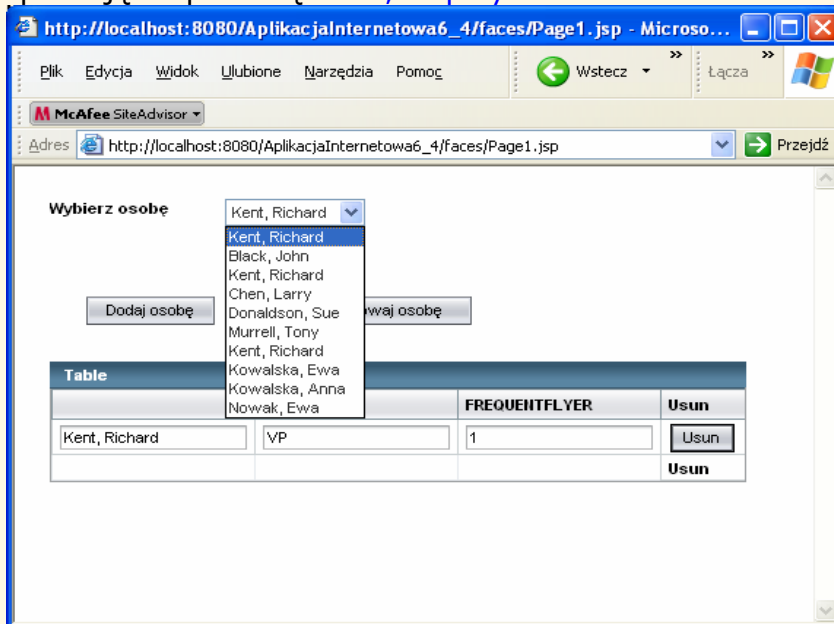


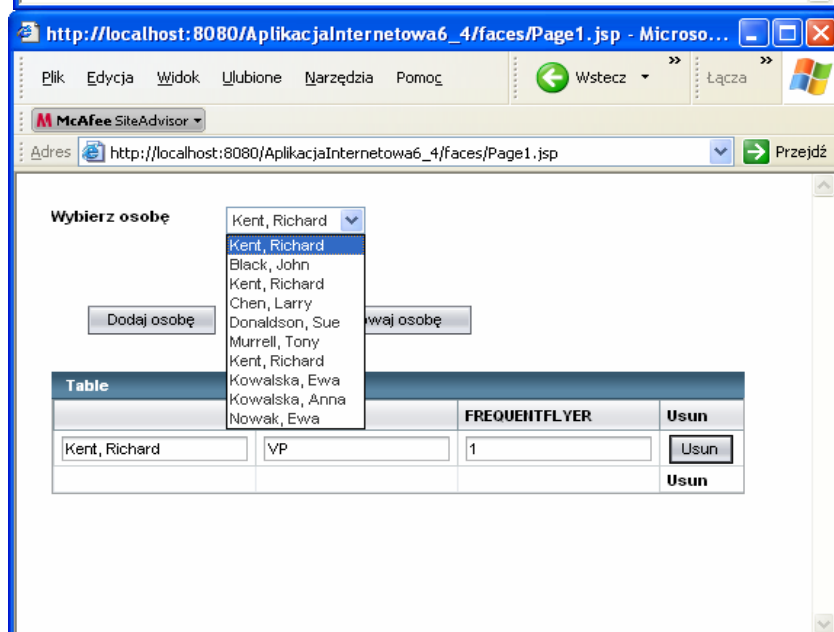
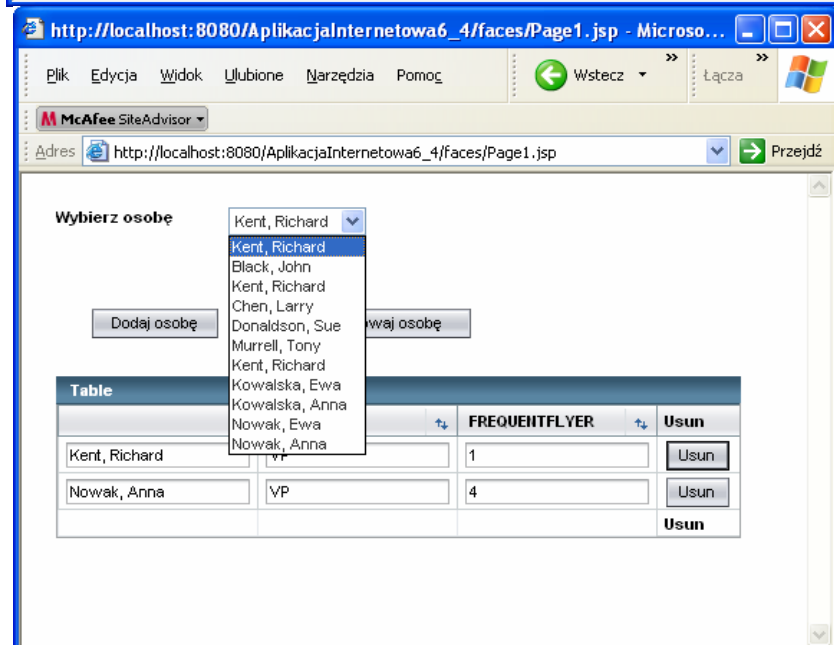
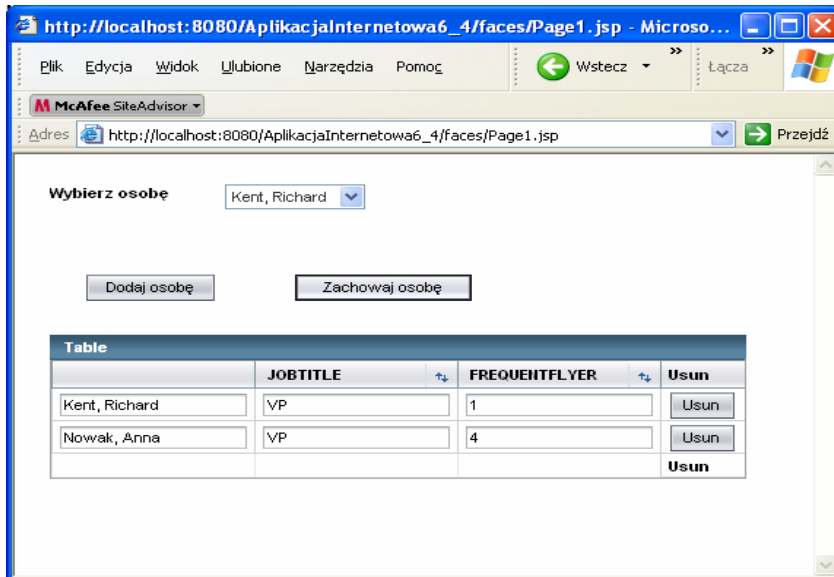
- Obsługa zdarzenia od naciśniętego klawisza button3 Usun

```
public String button3_action() {
```

```
    form1.discardSubmittedValues("zachowaj");
    try {
        RowKey rk = tableRowGroup1.getRowKey();
        if (rk != null) {
            personDataProvider1.removeRow(rk); // usunięcie wybranego rowka
            personDataProvider1.commitChanges();
            // zatwierdzenie wszystkich zmian w rowkach komponentu table1, nie tylko
            // usuwania, lecz również wstawionych rowków
            personDataProvider1.refresh();//odświeżenie zawartości komponentu table1
            personDataProvider.refresh(); //odświeżenie zawartości komponentu dropDown1
        }
    } catch (Exception ex) {
        log("ErrorDescription", ex);
        error(ex.getMessage());
    }
    return null;
}
```

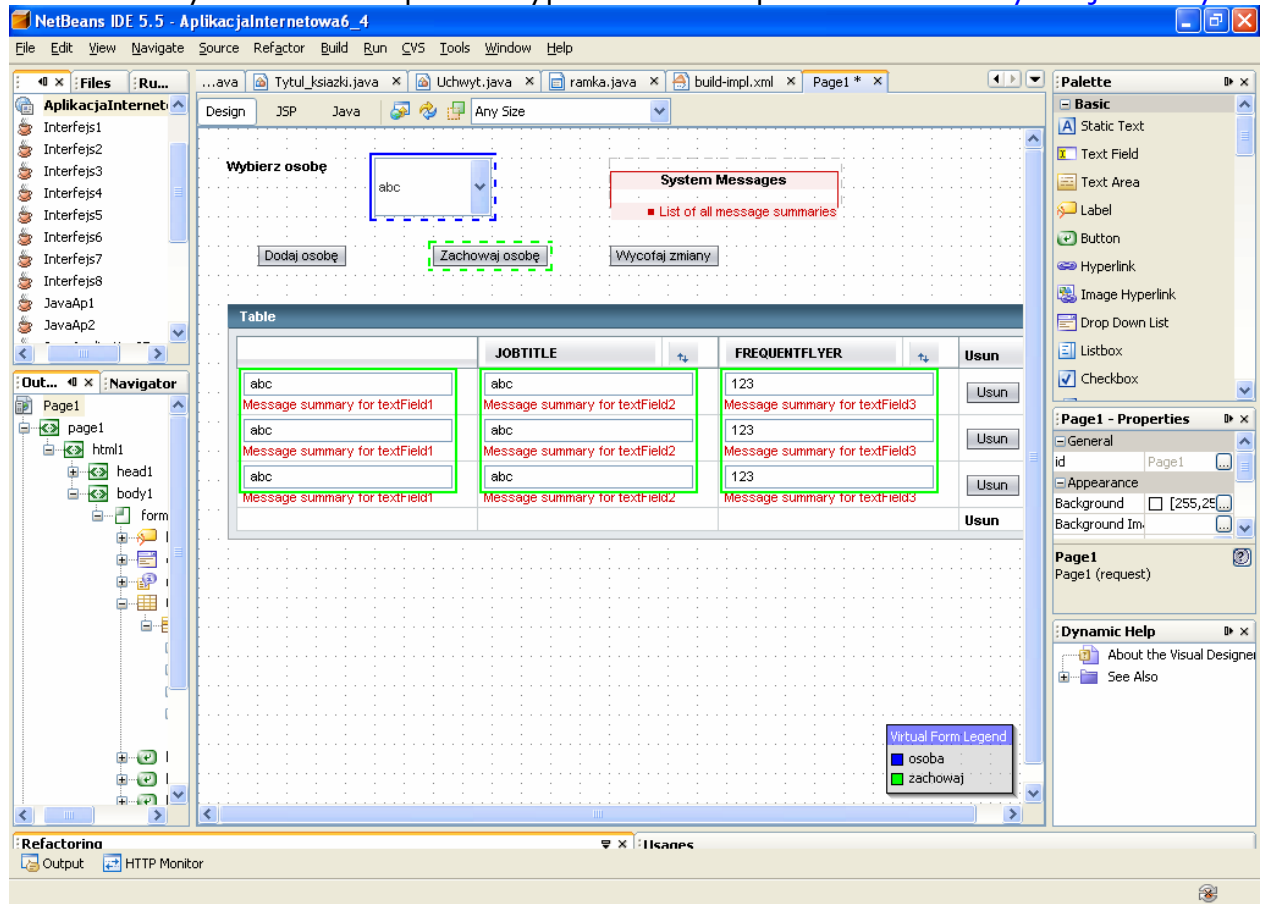
17. Uruchom aplikację za pomocą Build, Deploy i Run



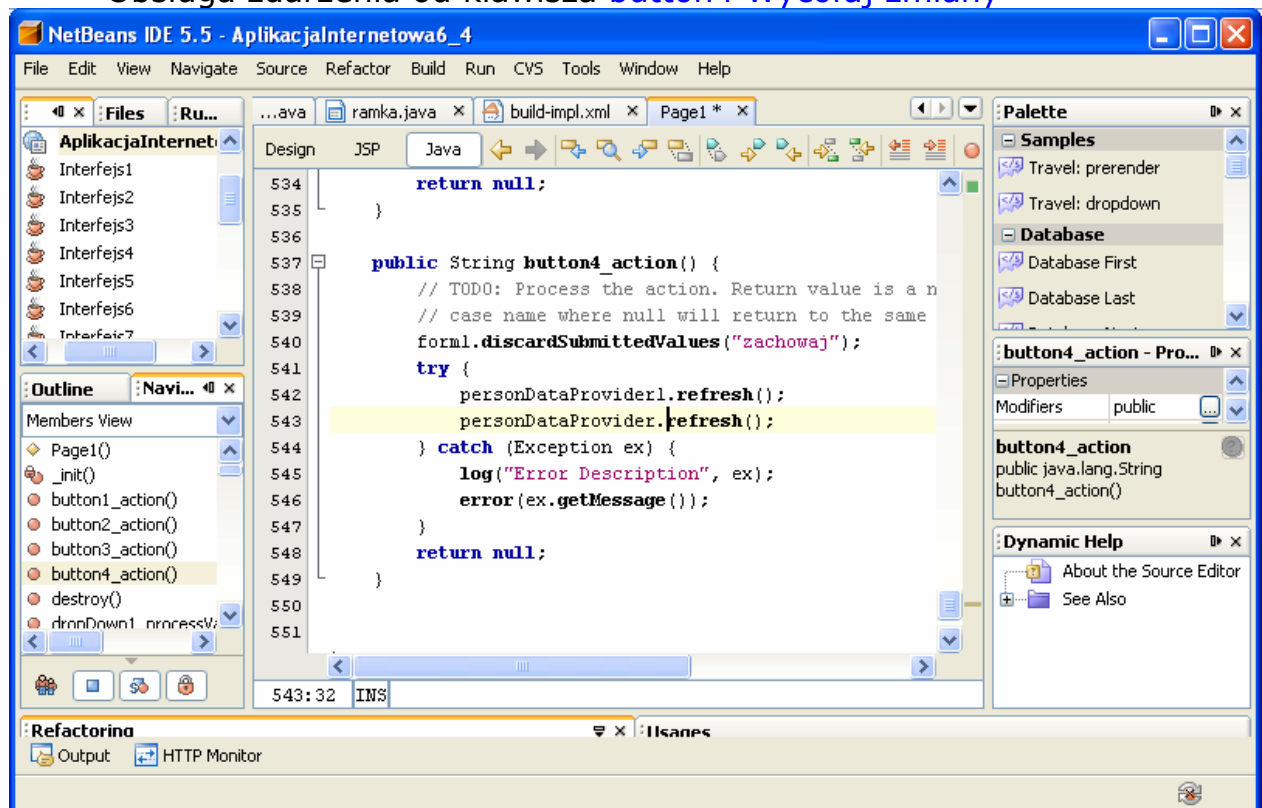


18. Wstawienie klawisza usuwania zmian wynikających z edycji

- Należy wstawić komponent typu **Button** i napsz naim tekst **Wycofaj zmiany**



- Obsługa zdarzenia od klawisza **button4 Wycofaj zmiany**

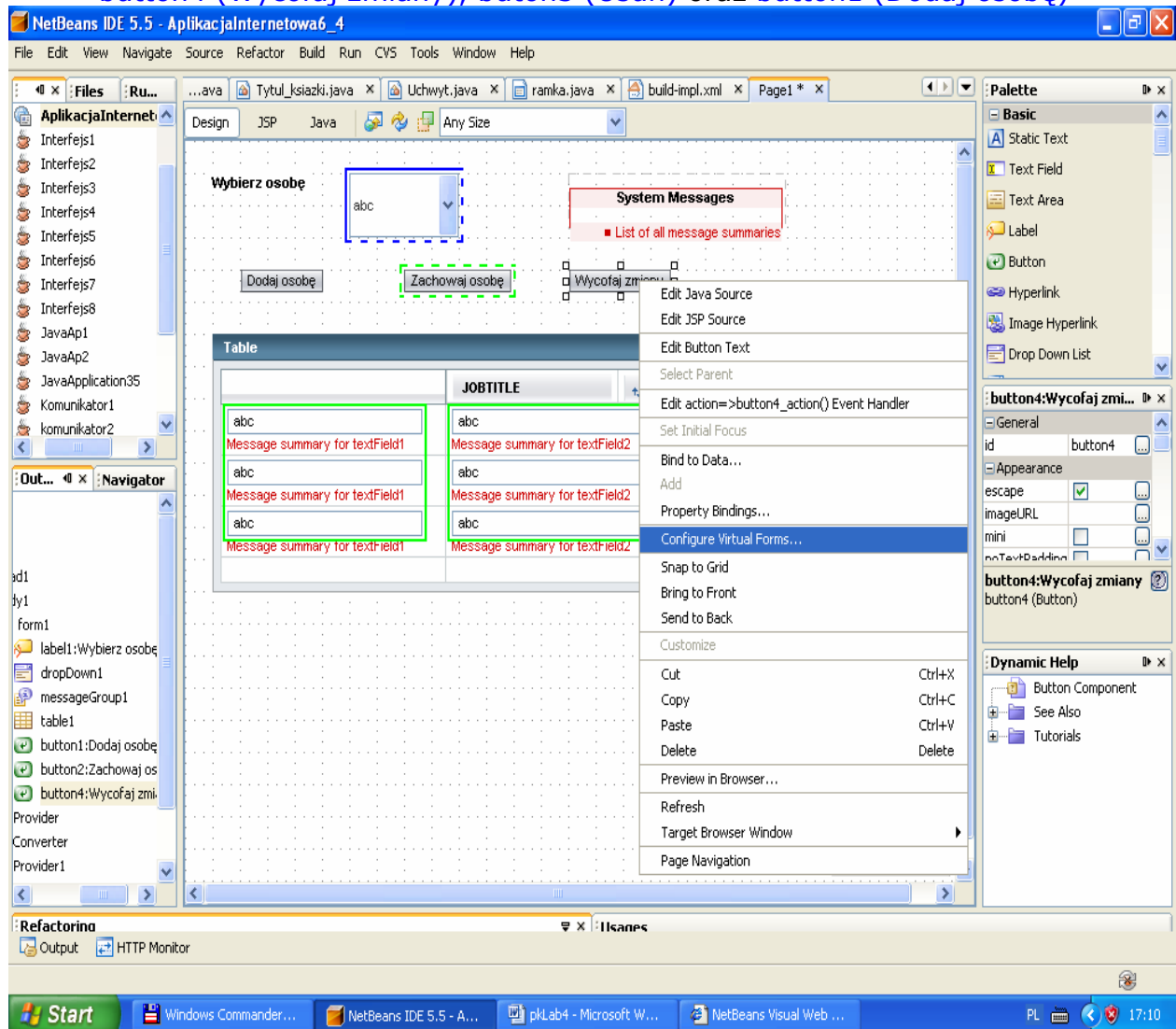


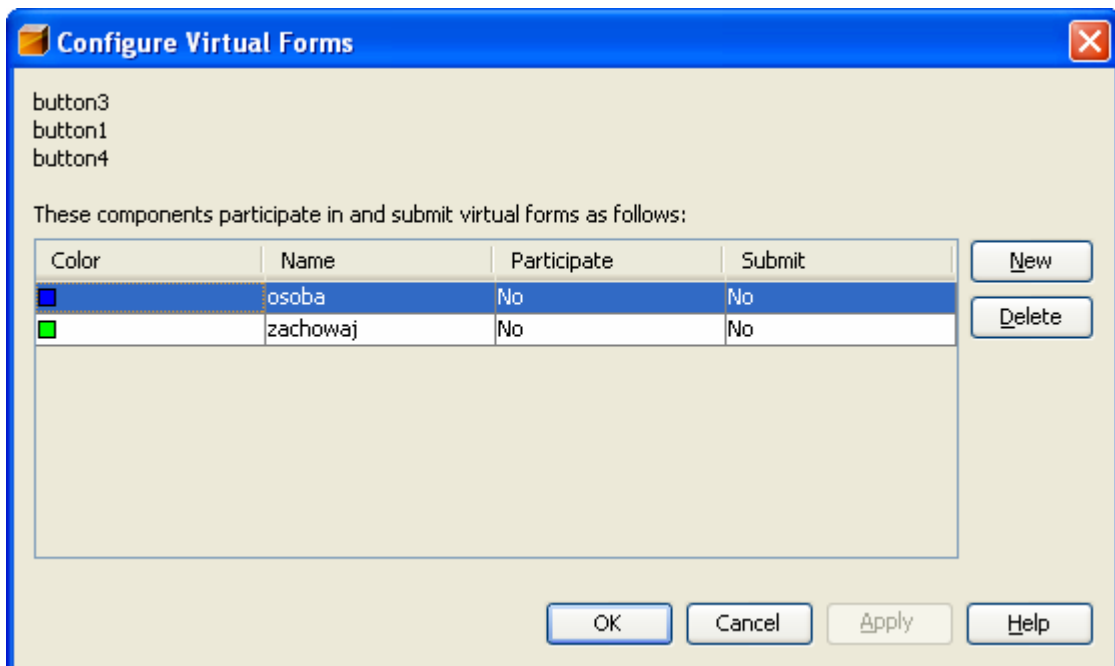
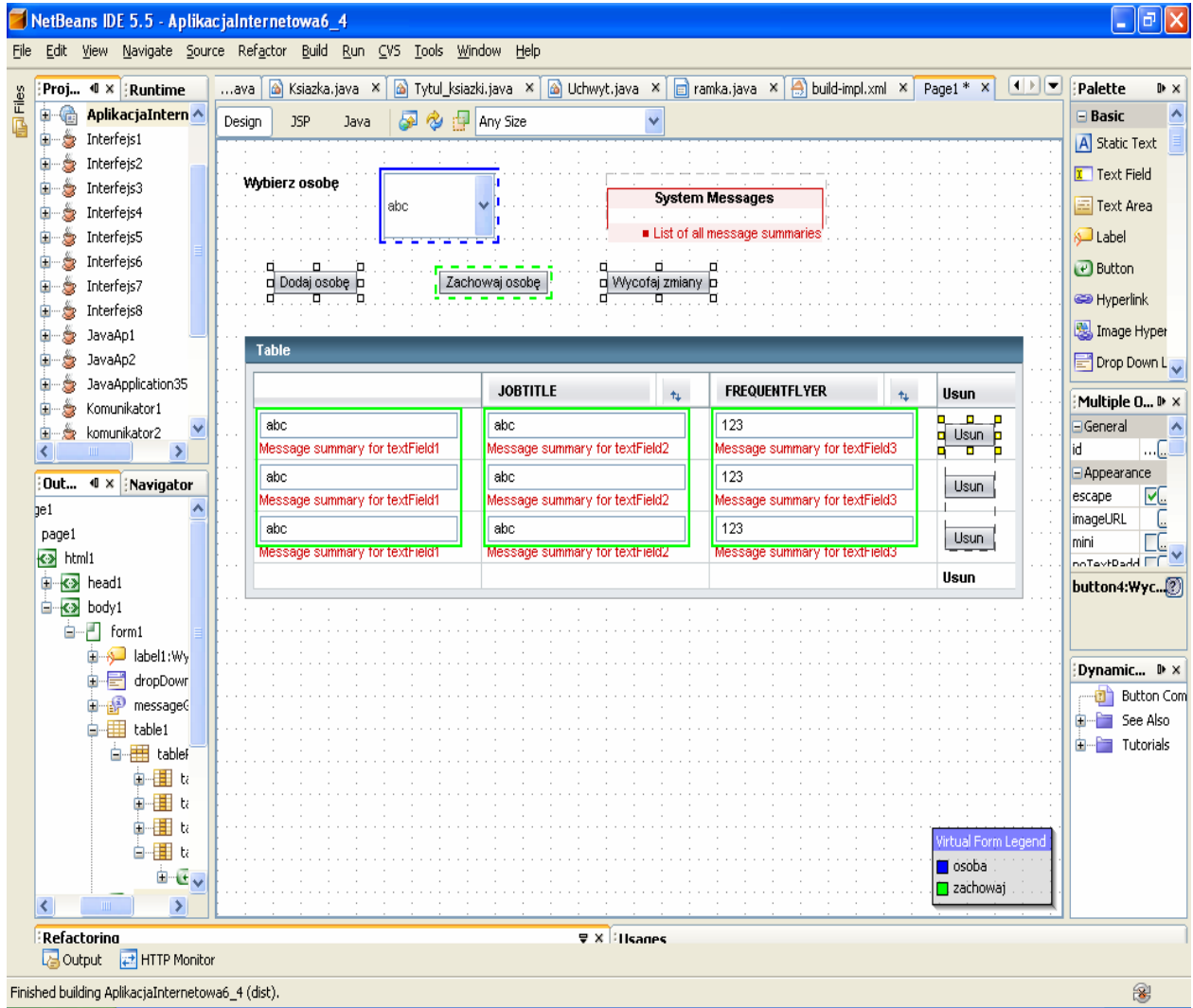
```

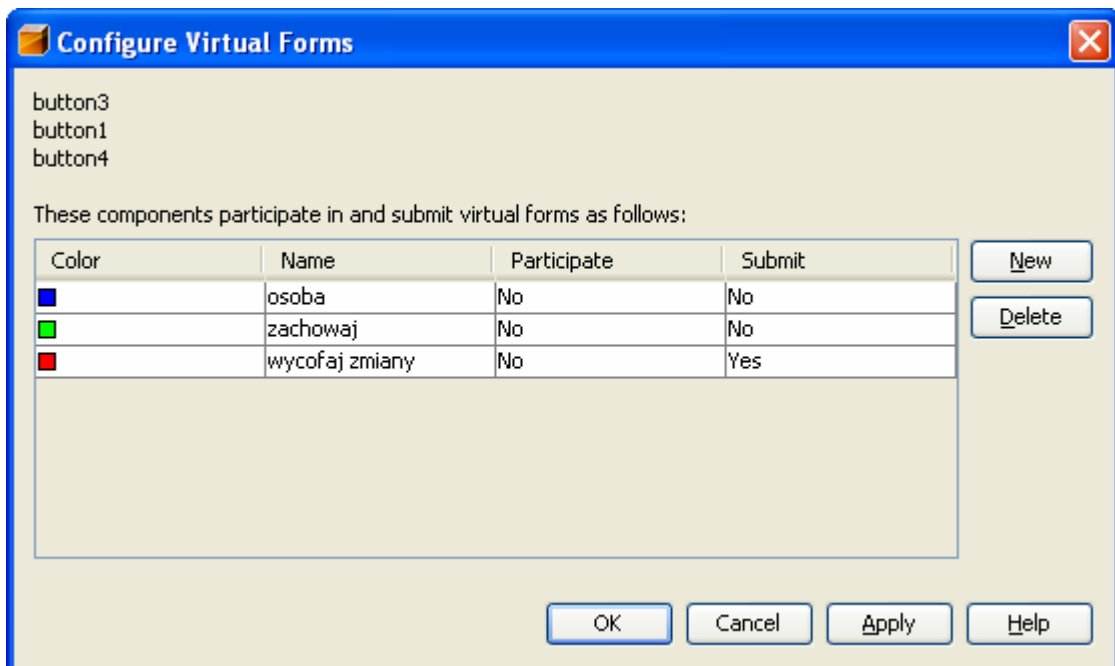
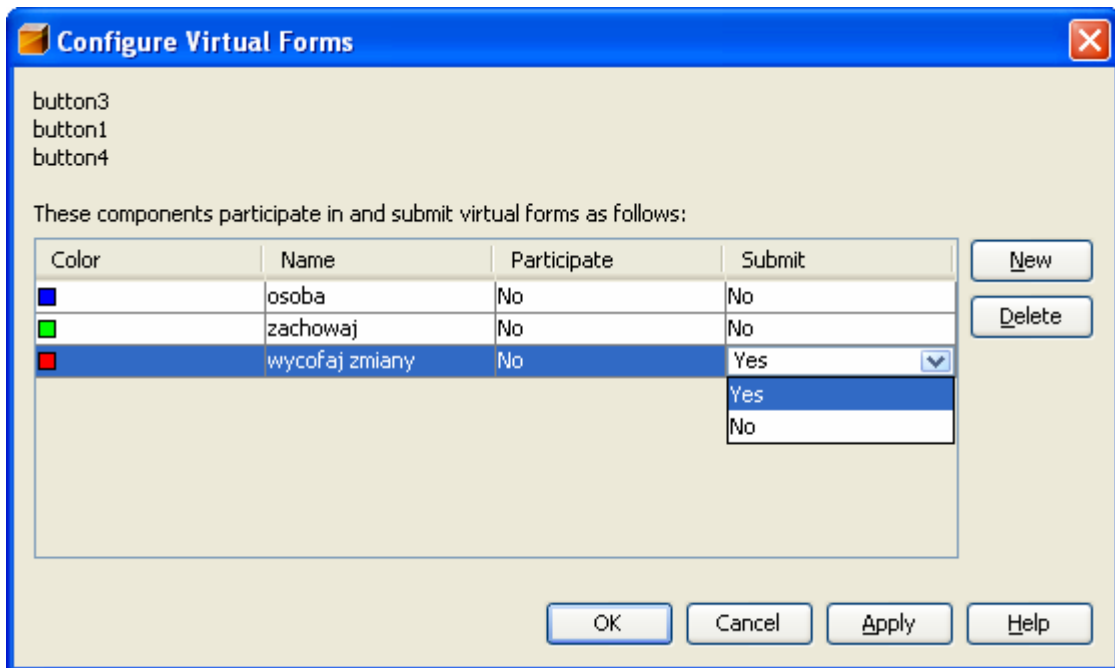
public String button4_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    form1.discardSubmittedValues("zachowaj");
    try {
        personDataProvider1.refresh();//odświeżenie zawartości komponentu droptable1
        personDataProvider.refresh();//odświeżenie zawartości komponentu dropDown1
    } catch (Exception ex) {
        log("Error Description", ex);
        error(ex.getMessage());
    }
    return null;
}
}

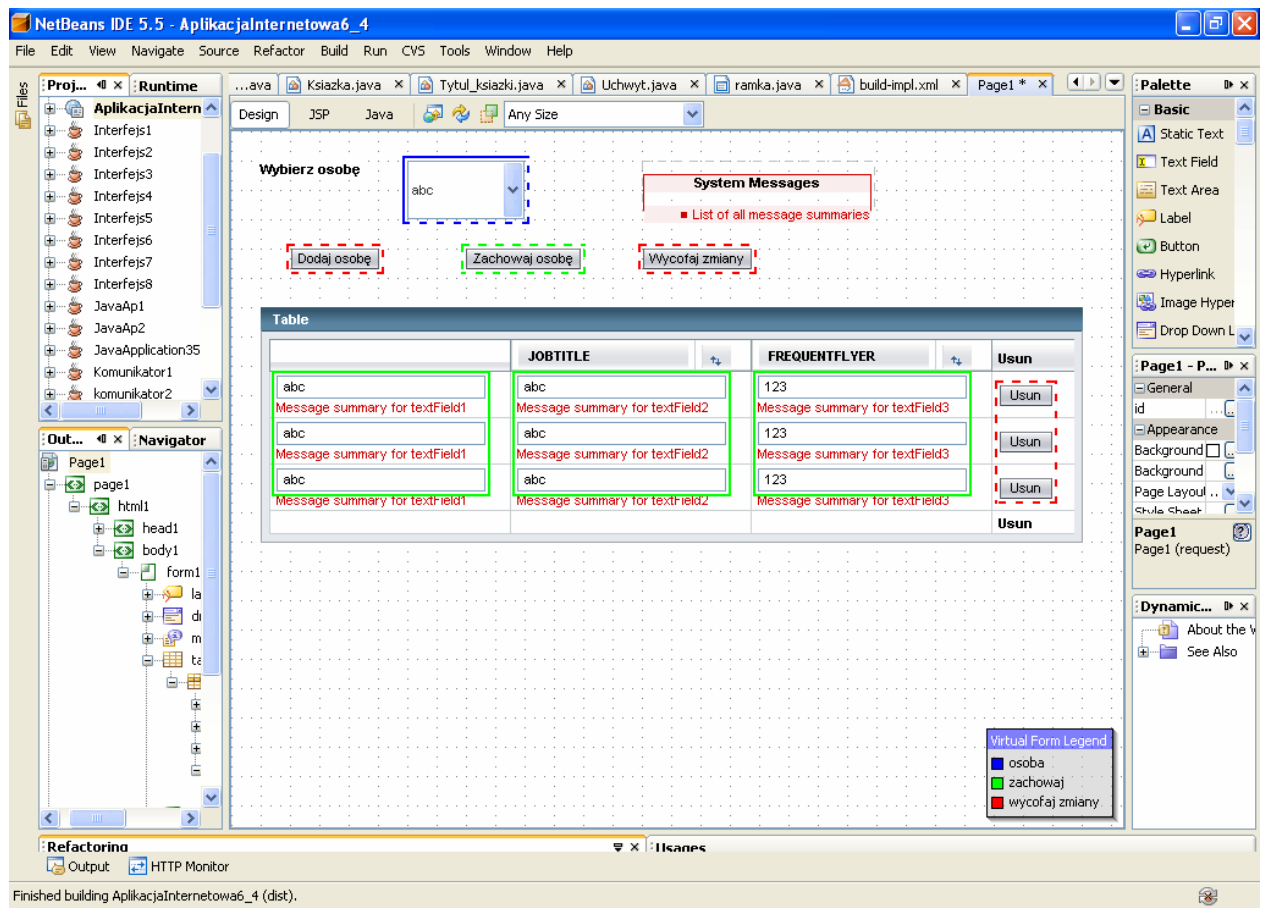
```

- Utworzenie kolejnego formularza wirtualnego **wycofaj zmiany** dla przycisków **button4 (Wycofaj zmiany)**, **buton3 (Usun)** oraz **button1 (Dodaj osobę)**

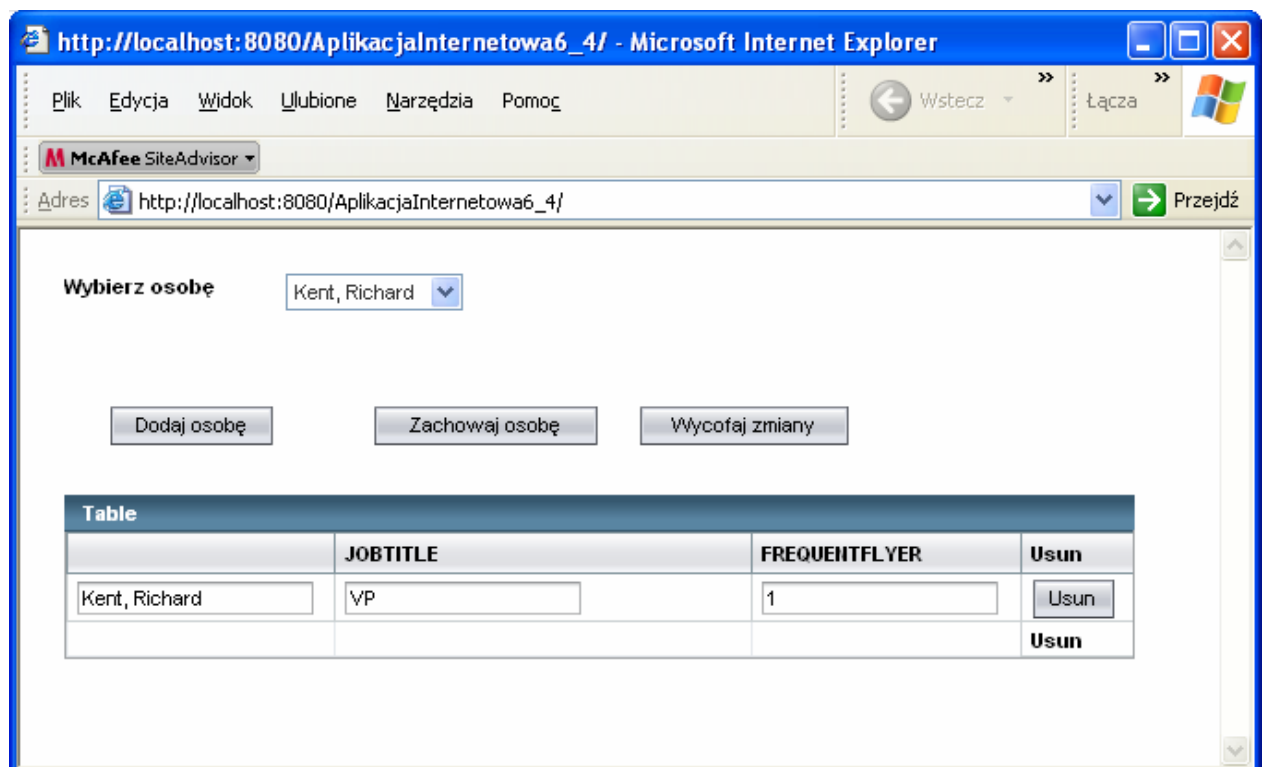








19. Uruchomienie aplikacji: Buid, Deploy, Run



- Po kliknięciu na inną osobę w **dropDown1** i wypełnieniu dodanego rowka przyciskiem **Dodaj osobę** danymi owej osoby

The screenshot shows a web browser window with the address `http://localhost:8080/AplikacjaInternetowa6_4/faces/Page1.jsp`. The page contains a form with a dropdown menu labeled "Wybierz osobę" that is open, showing a list of names: Black, John; Kent, Richard; Black, John; Kent, Richard; Chen, Larry; Donaldson, Sue; Murrell, Tony; and aaaaa. Below the dropdown are buttons for "Dodaj osobę", "Zachowaj osobę", and "Wycofaj zmiany". A table below the form contains the following data:

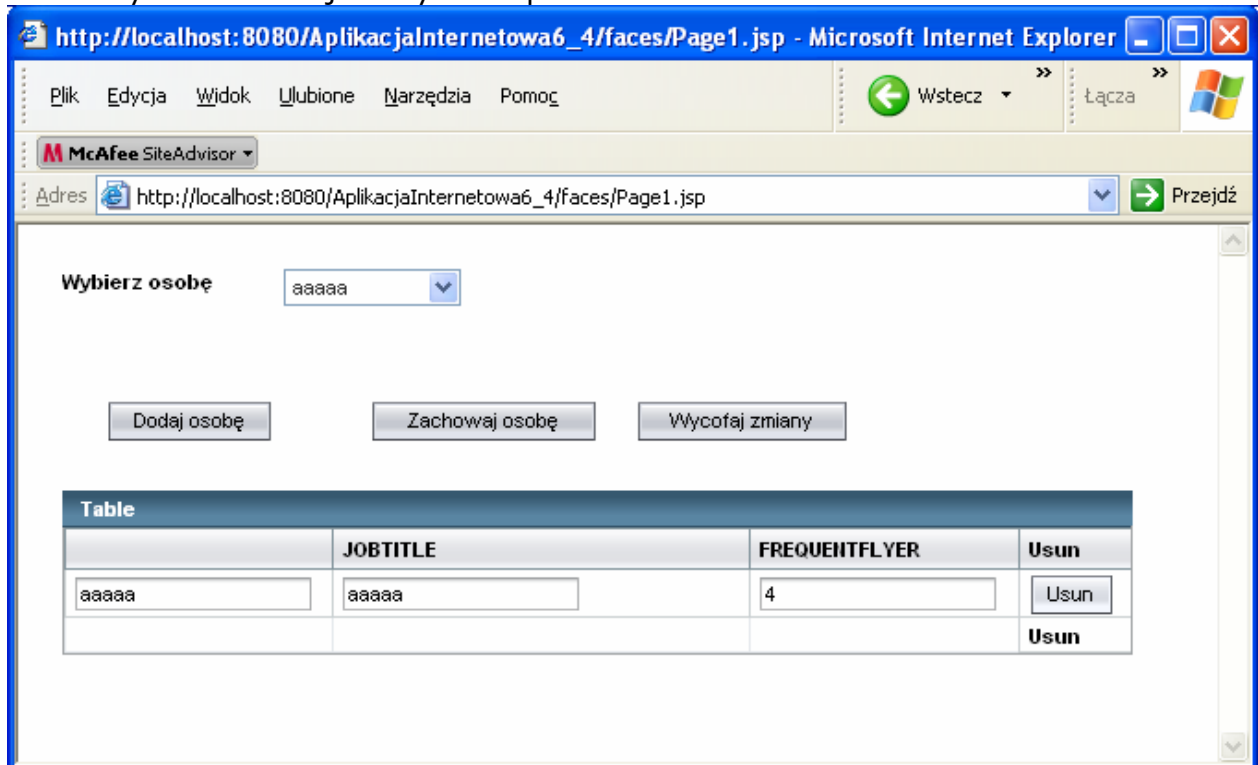
	JOBTITLE	FREQUENTFLYER	Usun
Black, John	CTO	2	Usun
Kowalska, Anna	VP	3	Usun
			Usun

- nowa osoba znajduje się w bazie danych po naciśnięciu klawisza **Zachowaj osobę**

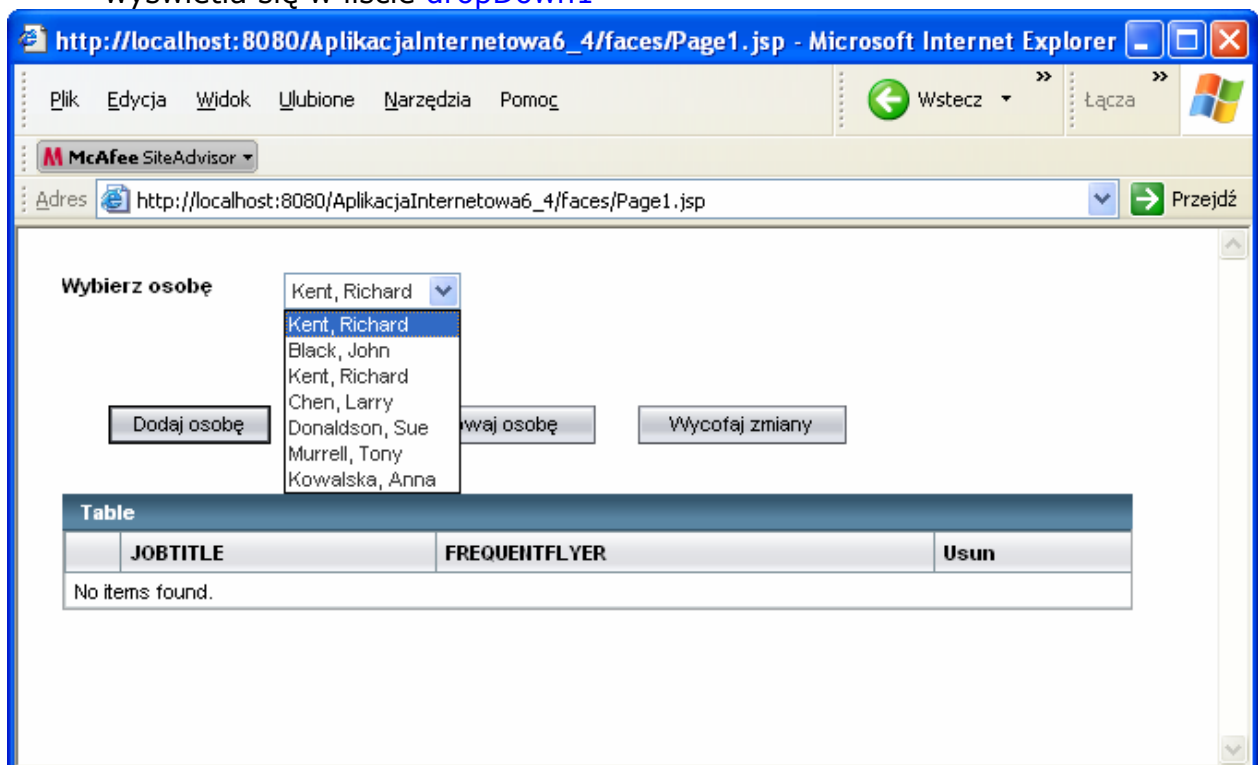
The screenshot shows the same web browser window. The dropdown menu is now closed, and the name "Kowalska, Anna" is visible in the input field. The table below the form contains the same data as in the previous screenshot:

	JOBTITLE	FREQUENTFLYER	Usun
Black, John	CTO	2	Usun
Kowalska, Anna	VP	3	Usun
			Usun

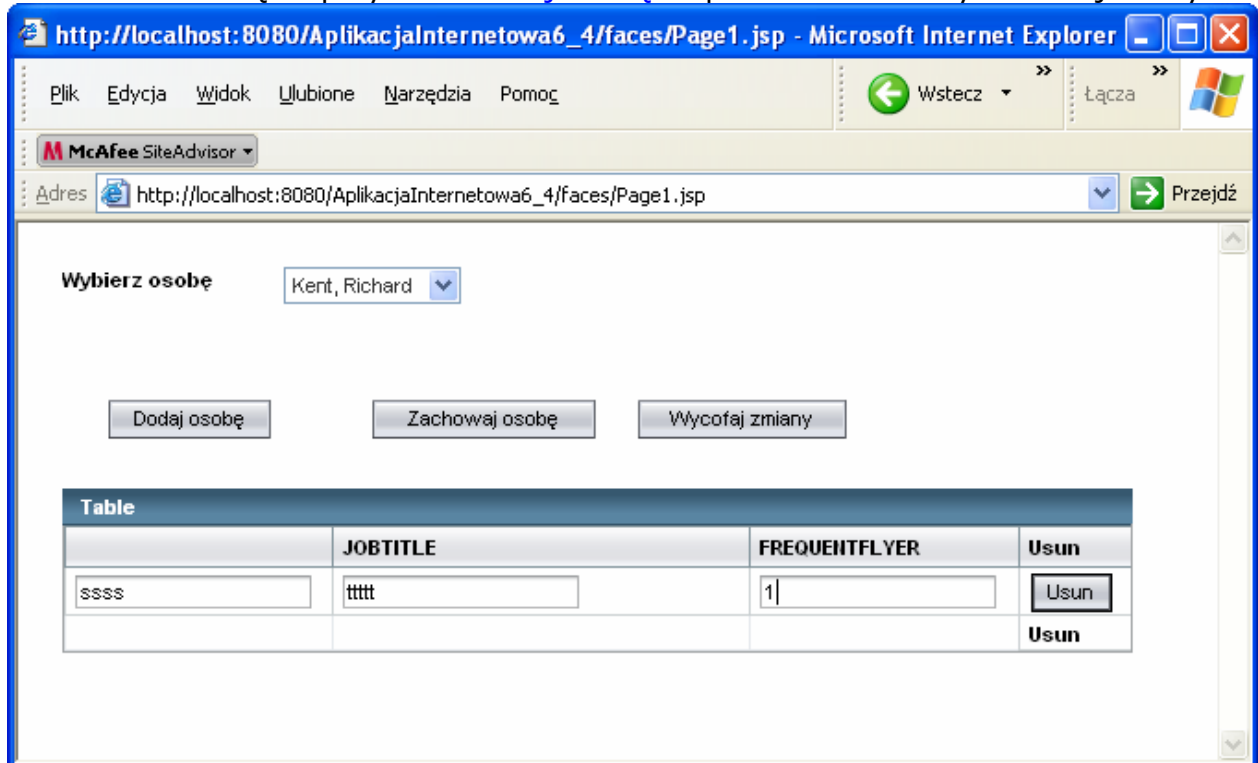
- wybranie nowej osoby w dropDown1



- Po kliknięciu na przycisk **Usun**, osoba została usunięta z bazy danych i już nie wyświetla się w liście **dropDown1**



- Po naciśnięciu przycisku **Dodaj osobę** i wprowadzeniu danych nowej osoby



- Po naciśnięciu przycisku **Wycofaj zmiany**

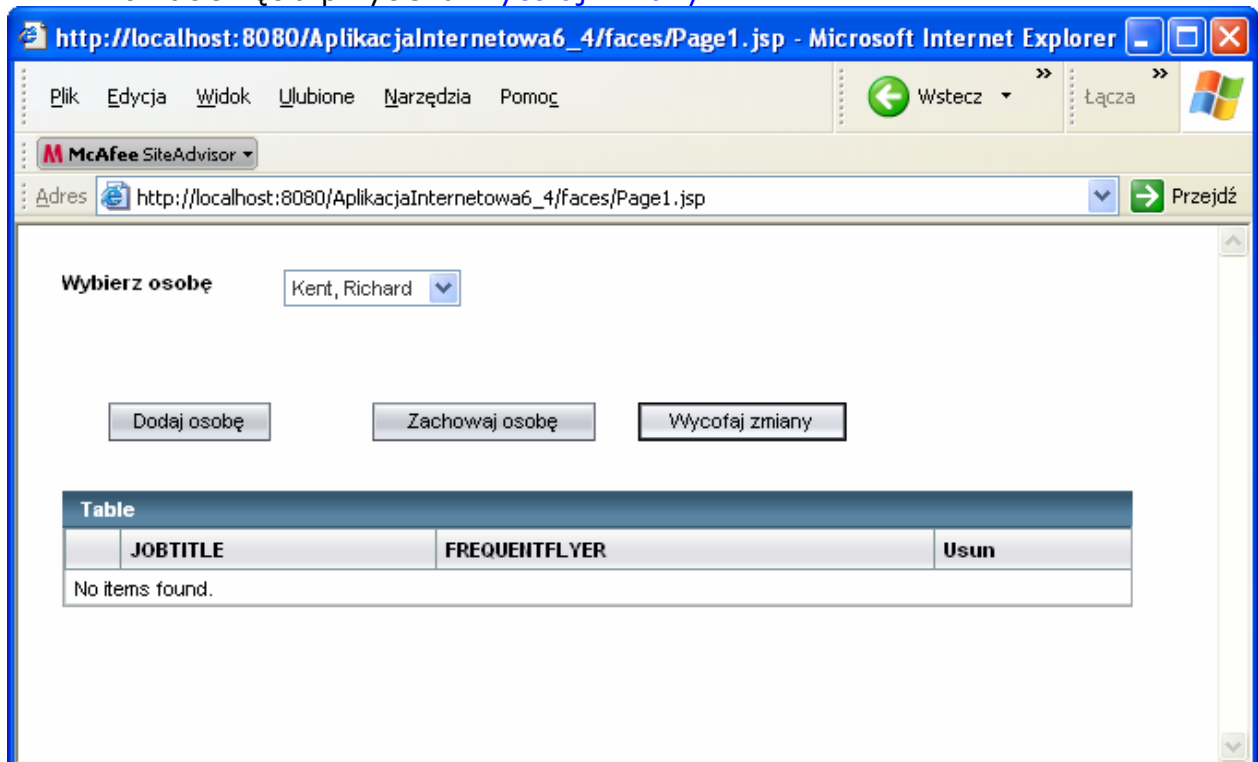


Table 1: Input Components

Component	Description	Palette Section
Text Field	An input field for a single line of text.	Basic
Text Area	An input field for multiple lines of text.	Basic
Password Field	An input field that echoes the input characters with a replacement character to mask the input.	Basic
Calendar	An input field with a pop-up calendar that allows the user to select a date.	Basic
Drop Down List	A drop-down menu, also referred to as a combo box.	Basic
Listbox	A list from which the user can select either one item or multiple items, depending on how the component is configured.	Basic
Checkbox	A single-character box that the user can either select (check) or clear.	Basic
Radio Button	A single radio button that the user can select (check).	Basic
Add Remove List	Two lists (one for available options, one for selected options) with buttons to move the options between the lists, and to order the selected options.	Composite
File Upload	A component with a text input field and a Browse button that displays a file chooser for specifying a file to upload. The application uploads the specified file when the user submits the page.	Basic

Table 2: Display Components

Component	Description	Palette Section
Static Text	Field for displaying text.	Basic
Label	Text field that can be associated with an input field and for which you can specify a weak, medium, or strong font style.	Basic
Image	Inline image.	Basic
Message	Text field that is linked to a specific component for displaying validation errors and other messages about that component.	Basic
Message Group	Text field for displaying runtime error messages, program generated error messages, and, optionally, validation errors and other messages about components that are on the page.	Basic
Alert	Displays an icon and informational text such as a warning, an error, or the successful completion of some event.	Composite
Page Alert	Similar to an Alert component, but is intended for displaying the icon and information on a separate page.	Layout

Table 3: Grouping Components

Component	Description	Palette Section
Checkbox Group	Displays two or more check boxes in a grid layout.	Basic
Radio Button Group	Displays two or more radio buttons in a grid layout and ensures that only one button can be selected at a time.	Basic
Table, Table Row Group, and Table Column	Displays data from a composite data type such as a database table or an array.	Basic
Tree and Tree Node	Renders an expandable list in a tree structure.	Basic
Tab Set and Tab	Displays different layouts on the same page. Also can be used as a navigational tool.	Layout
Grid Panel	Organizes the components within a layout of rows and columns.	Layout
Group Panel	Groups a set of components in flow layout mode.	Layout
Layout Panel	Use to group a set of components in flow layout mode or grid layout mode.	Layout
Property Sheet, Property Sheet Section, and Property	Lays out a single column of labeled components quickly, and divides the components into sections.	Layout
Breadcrumbs	Lays out a series of link components separated by right angle brackets (>).	Composite
Page Fragment Box	Groups components that you want to consistently display on two or more pages.	Layout

Table 4: Action Components

Component	Description	Palette Section
Button	Button that submits the associated form.	Basic
Hyperlink	Text field that submits a URL or submits a form.	Basic
Image Hyperlink	Image that submits a URL or submits a form.	Basic
Tree Node	Subcomponent of a Tree or Tree Node. A leaf tree node can optionally submit a URL or submit a form.	Basic
Tab	Subcomponent of a Tab Set or a Tab. A tab can optionally submit a URL or submit a form.	Composite