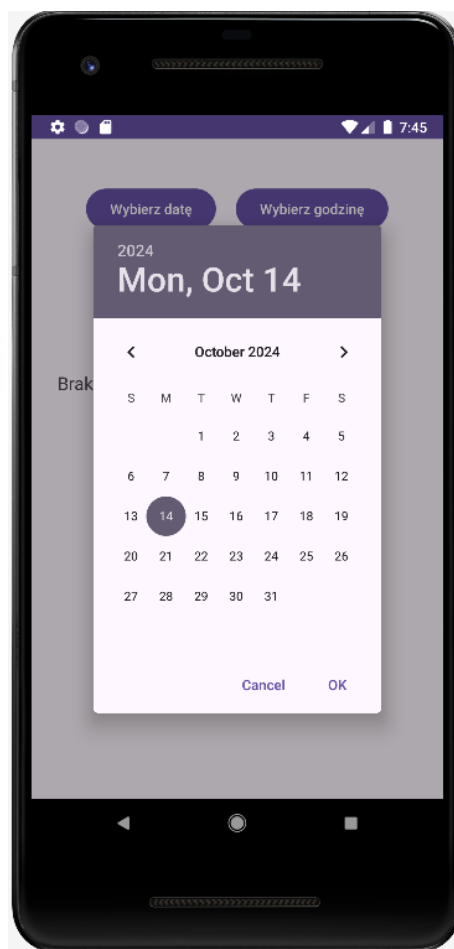
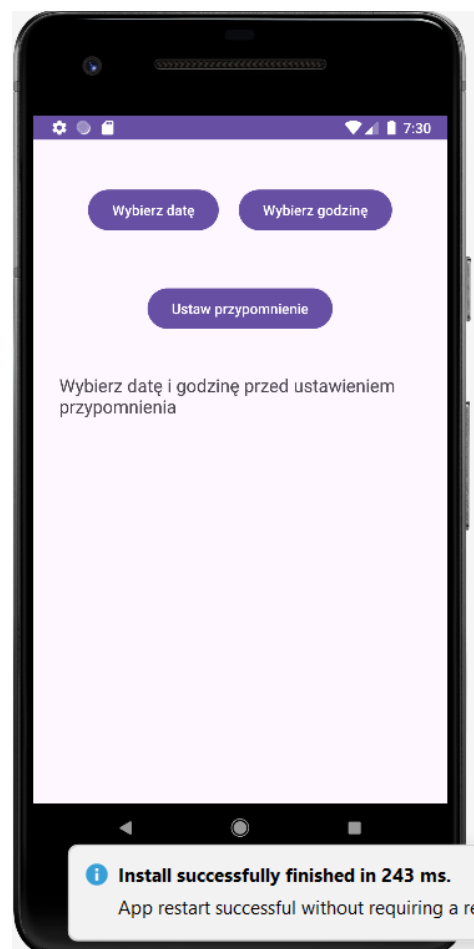


Funkcjonalny przykład użycia zarówno DatePicker, jak i TimePicker w aplikacji przypomnień.

- Użytkownik wybiera datę za pomocą przycisku "Wybierz datę".
- Użytkownik wybiera godzinę za pomocą przycisku "Wybierz godzinę".
- Po kliknięciu "Ustaw przypomnienie" aplikacja sprawdza, czy zarówno data, jak i godzina zostały wybrane. Jeśli tak, wyświetla przypomnienie z wybraną datą i godziną w TextView.
- Jeśli użytkownik nie wybrał daty lub godziny, wyświetlany jest komunikat z prośbą o wybór.





## 1. DatePicker

- **Typ komponentu:** DatePicker jest widżetem (komponentem UI), który można dodać bezpośrednio do layoutu XML.
- **Użycie:** Można go umieścić na ekranie na stałe jako część interfejsu. Użytkownik może bezpośrednio manipulować widżetem na ekranie (przesuwać miesiące, wybierać dni).
- **Kiedy używać:** DatePicker używany jest wtedy, gdy chcesz, aby wybór daty był widoczny cały czas na ekranie, np. w formularzach, gdzie użytkownik musi podać dokładną datę (np. urodzin, daty ważnego wydarzenia).

## 2. DatePickerDialog

- **Typ komponentu:** DatePickerDialog to modalne okno dialogowe, które wyświetla DatePicker, ale nie jest stałym elementem interfejsu. Jest to dialog, który pojawia się po akcji użytkownika (np. po kliknięciu przycisku).
- **Kiedy używać:** DatePickerDialog jest wygodny, gdy chcesz, aby wybór daty odbywał się tylko na żądanie, np. gdy użytkownik klika przycisk, aby wybrać datę, a po wyborze dialog znika.

```

btnDatePicker.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Calendar calendar = Calendar.getInstance();
        int year = calendar.get(Calendar.YEAR);
        int month = calendar.get(Calendar.MONTH);
        int day = calendar.get(Calendar.DAY_OF_MONTH);

        DatePickerDialog datePickerDialog = new DatePickerDialog(context: MainActivity.this, new DatePickerDialog.OnDateSe
            1 usage
            @Override
            public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
                Year = year;
                Month = month;
                Day = dayOfMonth;
                isDateSet = true;
            }
        }, year, month, day);
        datePickerDialog.show();
    }
});

```

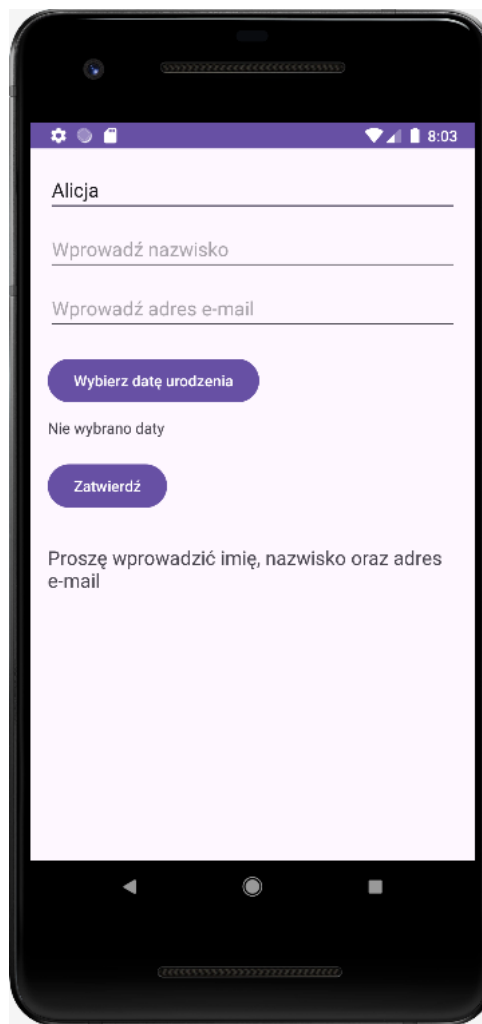
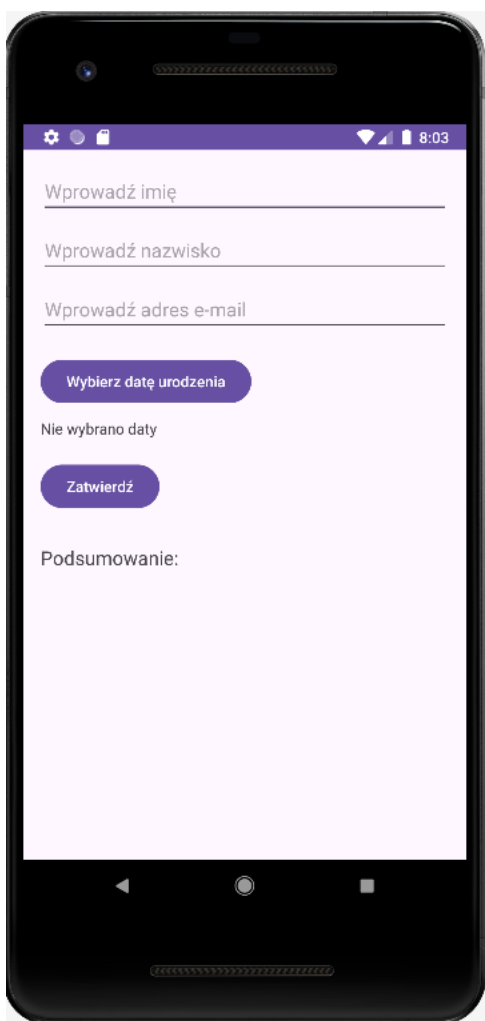
```

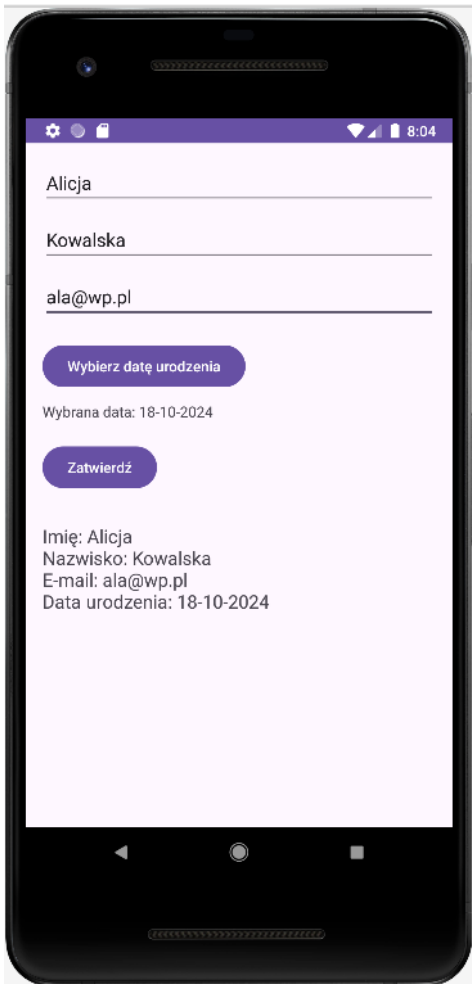
btnSetText.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isDateSet && isTimeSet) {
            String razem = String.format("Przypomnienie ustawione na %02d-%02d-%d, %02d:%02d",
                Day, Month + 1, Year, Hour, Minute);
            txt.setText(razem);
        } else {
            txt.setText("Wybierz datę i godzinę przed ustawieniem przypomnienia");
        }
    }
});

```

## Zadanie 2 Formularz z DatePickerDialog.

1. Użytkownik wprowadza swoje imię, nazwisko i adres e-mail.
2. Wybiera datę urodzenia za pomocą przycisku "Wybierz datę urodzenia".
3. Po kliknięciu przycisku "Zatwierdź" aplikacja sprawdza, czy wszystkie dane zostały wprowadzone, a następnie wyświetla podsumowanie w TextView.
4. Sprawdź czy adres e-mail posiada znak „@”





### Zadanie 3. Formularz rejestracji i logowania.

Aby zaprojektować aplikację z oddzielnymi instancjami (czyli oddzielnymi ekranami/aktywnościami) dla rejestracji i logowania, wykorzystamy dwie osobne aktywności: **RegisterActivity** do rejestracji użytkownika oraz **LoginActivity** do logowania. Będzie również **MainActivity**, która wyświetli ekran początkowy, pozwalający wybrać, czy chcemy przejść do logowania, czy do rejestracji.

#### Struktura aplikacji:

- **MainActivity:** Ekran startowy z dwoma przyciskami do przejścia do aktywności rejestracji lub logowania.
- **RegisterActivity:** Formularz rejestracyjny, który pozwala zarejestrować użytkownika i dodaje go do listy.
- **LoginActivity:** Formularz logowania, który sprawdza, czy podany użytkownik jest zarejestrowany.

