

# Storyboard - Tablice rejestracyjne

## 1 Omówienie poszczególnych widoków

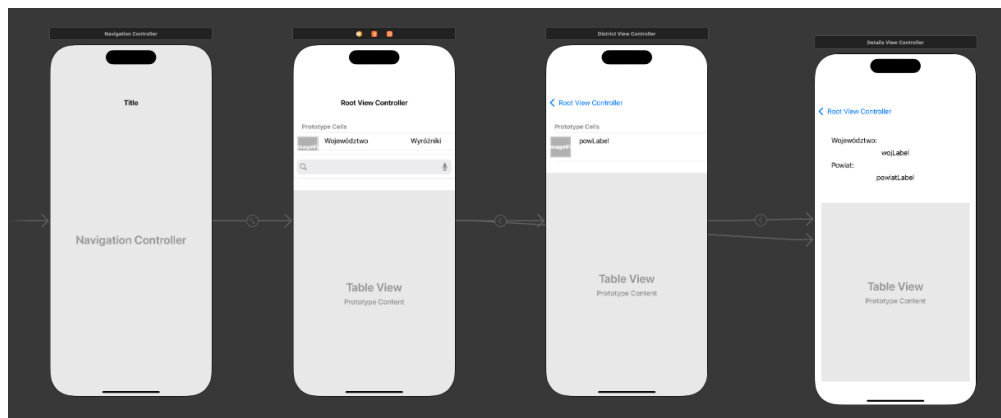
Do tej pory pracowaliśmy tylko wykorzystując `UIViewController`, czyli klasyczny pusty widok, gdzie sami musimy rozmieścić poszczególne elementy na stronie i je potem podłączyć. W tym projekcie wykorzystamy oprócz `UIViewController` także `UITableViewController`, który jest używany kiedy chcemy mieć widok "tabelaryczny" czyli tabela na całą stronę. Jest to łatwiejsze niż korzystanie z `UIViewController` i dodawaniem ręcznie `TableView` i podłączenie tego w kodzie klasy odpowiedzialnej za widok okna. W tym widoku możemy określić ilość komórek prototypowych - **Prototype cells**, wtedy możemy zaprojektować jak mają wyglądać poszczególne komórki tabeli. Jeżeli liczba prototypowych komórek będzie ustawiona na 0 to wtedy w kodzie możemy zaprogramować jak będą wyglądać lub użyć domyślnego wyglądu.

## 2 Wprowadzenie do projektu

W tym projekcie będziemy tworzyć "bazę" tablic rejestracyjnych. Schemat projektu poniżej:

Województwa -> Powiaty -> Szczegóły

Dane z tablicami są przechowywane w tablicy - plik `property list`



Wygląd storyboard'u

## 3 Pierwszy widok - województwa

W pierwszym widoku mamy dwie komórki prototypowe:

- Pierwsza ze zdjęciem, nazwą województwa i wyróżnikiem
- Druga - search bar do wyszukiwania

Ponieważ mamy dwie komórki to dodajemy dwie klasy: `VoivodshipTableViewCell` i `SearchTableViewCell`.

## VoivodshipTableViewCell

```
import UIKit

class VoivodshipTableViewCell: UITableViewCell {

    override func awakeFromNib() {
        super.awakeFromNib()
        // Initialization code
    }

    @IBOutlet weak var labelSelectors: UILabel!
    @IBOutlet weak var labelVoivodship: UILabel!
    @IBOutlet weak var voivodshipImg: UIImageView!

    override func setSelected(_ selected: Bool, animated: Bool) {
        super.setSelected(selected, animated: animated)

        // Configure the view for the selected state
    }
}
```

Na początku `awakeFromNib` jest używana do inicjalizacji po załadowaniu (można to nadpisać jeszcze), następnie mamy "gniazdka" do naszych obiektów ze storyboardu (tworzy się je tak: trzymając ctrl klikamy i przeciągamy do kodu i następnie wybieramy opcje z okienka), na koniec `setSelected` odpowiada za zmianę wyglądu komórki po zaznaczeniu - ale tutaj nie robi nic szczególnego ta funkcja. Te metody (`awakeFromNib` i `setSelected`) są już w szablonie, jak wybierzemy nowy plik według wzoru.

## SearchTableViewCell

```
import UIKit

class SearchTableViewCell: UITableViewCell {

    override func awakeFromNib() {
        super.awakeFromNib()
        // Initialization code
    }

    @IBOutlet weak var searchBar: UISearchBar!
    override func setSelected(_ selected: Bool, animated: Bool) {
        super.setSelected(selected, animated: animated)

        // Configure the view for the selected state
    }
}
```

Ta klasa odpowiada za `searchBar`. `awakeFromNib` odpowiada za inicjalizację, następnie mamy "gniazdka" do naszego `searchBar`'u. Na koniec `setSelected` odpowiada za zmianę wyglądu komórki po zaznaczeniu, ale w tym przypadku nie robi nic specjalnego.

## ViewController

Klasa `ViewController` to główna klasa odpowiedzialna za pierwszy widok oraz za przejścia do kolejnych widoków. Jest to na ten moment najdłuższa i najbardziej skomplikowana klasa, dlatego najpierw będą kawałki kodu z wyjaśnieniami a na koniec cały kod.

```
var dictionary:NSDictionary?
var voivodships:[String] = []
var selectedVoivodship = ""
var foundDistrict = ""
var foundDistrictPlates:NSArray?
var selectedSelectors: [String]?
```

Pierwsza zmienna przechowuje dane z pliku `.plist`, następnie mamy tablice z nazwami województw, a poniżej zmienne wykorzystywane przy przejściach.

---

```
override func viewDidLoad() {
    super.viewDidLoad()
```

Wywoływane przy ładowaniu widoku

---

```
let bundleURL = Bundle.main.url(forResource: "tablice", withExtension: "plist")!
dictionary = NSDictionary(contentsOf: bundleURL)
```

Wczytanie danych z pliku do zmiennej `dictionary`

---

```
if let v = dictionary?["Województwa"] as? NSDictionary {
    for k in v.allKeys {
        if let w = k as? String {
            voivodships.append(w)
        }
    }
    voivodships.sort { $0.compare($1, locale: NSLocale.current) == .orderedAscending }
}
```

Na początek:

-? po nazwie zmiennej (np. `dictionary?`) oznacza opcjonalne łańcuchowanie

-as? oznacza bezpieczne rzutowanie typu

Tutaj tworzymy listę województw odczytując klucze z pliku i na końcu sortujemy w porządku alfabetycznym.

---

```
self.title = "Wybierz województwo"
self.tableView.rowHeight = 70
```

Ustawiamy tytuł (nagłówek) i wysokość wiersza.

---

```
override func numberOfSections(in tableView: UITableView) -> Int {
    return 1
}
```

Tabela ma jedną sekcję (jedną kolumnę)

---

```
override func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section: Int) -> Int {
    return voivodships.count + 1
}
```

Liczba wierszy - liczba województw + dodatkowy wiersz na `searchBar`

---

```
if indexPath.row < voivodships.count {
    let cell = tableView.dequeueReusableCell(withIdentifier: "cellVoivodship", for: indexPath)
    as! VoivodshipTableViewCell
```

Tutaj tworzymy komórki korzystając z klasy `VoivodshipTableViewCell`, czyli tworzymy komórki niestandardowe. Dla ścisłości `indexPath` wskazuje na konkretną komórkę w tabeli

---

```
let voivodshipName = voivodships[indexPath.row]
cell.labelVoivodship.text = voivodshipName
```

Przypisujemy nazwę województwa to `label`'a w wierszu.

---

```
let selectors = getSelectors(for: voivodshipName)
cell.labelSelectors.text = arrayToString(strings: selectors)
```

Dla danego województwa pobieramy selector (lub dwa), w naszym przypadku `wyr` i konwertujemy go do stringa i przypisujemy jak nazwę województwa.

---

```
if let firstSelector = selectors.first {
    cell.voivodshipImg.image = UIImage(named: firstSelector)
} else {
    cell.voivodshipImg.image = nil
}
```

Pobieramy wyróżnik (ale tylko pierwszy, ponieważ niektóre województwa mają dwa wyróżniki) i dodajemy zdjęcie z herbem.

---

```
} else {
    let cell = tableView.dequeueReusableCell(withIdentifier: "cellSearch", for: indexPath) as! SearchTableViewCell
    cell.searchBar.delegate = self
    return cell
}
```

W ostatnim wierszu dodajemy komórkę niestandardową, czyli nasz wcześniej zadeklarowany `searchBar`

---

```
override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    if indexPath.row < voivodships.count {
        selectedVoivodship = voivodships[indexPath.row]
        selectedSelectors = getSelectors(for: voivodships[indexPath.row])
        performSegue(withIdentifier: "segueDistricts", sender: self)
    }
}
```

Tutaj informujemy co ma się wydarzyć po zaznaczeniu danego rzędu. W tym przypadku przypisujemy do zmiennych nazwę wybranego województwa i wyróżnik oraz przechodzimy do ekranu powiatów `segueDistrict`

---

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    switch segue.identifier {
```

Ta metoda jest wywoływana przed przejściem do kolejnego widoku, jej zadaniem jest przygotowanie danych dla kontrolera do którego przechodzisz.

```
case "segueDistricts":
    if let destination = segue.destination as? DistrictViewController {
```

Czyli: jeśli przechodzimy do ekranu z powiatami województwa, to:

```
    if let vships = dictionary?["Województwa"] as? NSDictionary {
        if let district = vships[selectedVoivodship] as? NSDictionary {
```

Pobieramy województwa z pliku, a następnie pobieramy listę powiatów dla wybranego województwa

```
        destination.districts = district
        destination.title = selectedVoivodship
        destination.selectors = selectedSelectors!
        destination.selectedVoivodship = selectedVoivodship
```

Dane, do których ma dostęp nowy ekran

```
    if let destination = segue.destination as? DetailsViewController {
```

Obsługa tego co się stanie kiedy wyszukamy konkretną rejestrację

```
        destination.plates = foundDistrictPlates ?? []
        destination.selectors = selectedSelectors ?? []
        destination.title = foundDistrict
        destination.selectedVoivodship = selectedVoivodship
        destination.selectedDistrict = foundDistrict
```

Przekazujemy dane znalezione podczas wyszukiwania:

plates → lista skrótów z pliku .plist (np. ["DW"])

selectors → wyr z województwa

title, selectedVoivodship, selectedDistrict → nazwy do wyświetlenia

---

```
func getSelectors(for voivodship:String) -> [String] {
    if let root = dictionary?["Województwa"] as? NSDictionary {
        if let v = root[voivodship] as? NSDictionary {
            if let selectors = v["wyr"] as? [String] {
                var sel:[String] = []
                for s in selectors {
                    sel.append(s)
                }
                return sel
            }
        }
    }
    return []
}
```

Ta funkcja pobiera z pliku wyróżniki dla danego województwa i zwraca listę z wyróżnikami

---

```

func arrayToString(strings: [String]) -> String {
    guard strings.count > 0 else { return "" }
    var res = ""
    for s in strings {
        res.append(s)
        res.append(",")
    }
    let end = res.count-1
    return res[0..

```

Ta metoda zamienia listę na string np. [D, X] na D, X

---

### Obsługa searchBar - po kliknięciu wyszukaj

```

if let term = searchBar.text {

```

Pobiera text z searchBar

```

    outerLoop: for k in v.allKeys {

```

Przegląda każde województwo

```

        for w in wyr {
            for a in d.allKeys {
                if let p = d[a] as? [String] {
                    for s in p {
                        if "\(w)\(s)" == term {

```

Dla wybranego województwa sprawdzamy każdą literę w wyróżnikach, następnie sprawdzamy wyróżniki dla powiatu i jeżeli one połączone z wyróżnikiem województwa są takie same jak wpisana wartość to:

```

            selectedVoivodship = k as! String
            selectedSelectors = wyr
            foundDistrict = a as! String
            foundDistrictPlates = d[a] as? NSArray
            performSegue(withIdentifier: "segueSearchResults", sender: self)

```

Szykujemy dane które są potrzebne do przejścia do następnego widoku

---

Cały kod klasy ViewController:

```

import UIKit

class ViewController: UITableViewController, UISearchBarDelegate {

    var dictionary:NSDictionary?
    var voivodships:[String] = []
    var selectedVoivodship = ""
    var foundDistrict = ""
    var foundDistrictPlates:NSArray?
    var selectedSelectors: [String]?

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.

```

```

let bundleURL = Bundle.main.url(forResource: "tablice", withExtension: "plist")!
dictionary = NSDictionary(contentsOf: bundleURL)
if let v = dictionary?["Województwa"] as? NSDictionary {
    for k in v.allKeys {
        if let w = k as? String {
            voivodships.append(w)
        }
    }
    voivodships.sort { $0.compare($1, locale: NSLocale.current) == .orderedAscending }
}
self.title = "Wybierz województwo"
self.tableView.rowHeight = 70
}

override func numberOfSections(in tableView: UITableView) -> Int {
    return 1
}

override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int)
-> Int {
    return voivodships.count + 1
}

override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath)
-> UITableViewCell {
    if indexPath.row < voivodships.count {
        let cell = tableView.dequeueReusableCell(withIdentifier: "cellVoivodship", for:
indexPath) as! VoivodshipTableViewCell
        let voivodshipName = voivodships[indexPath.row] // <-- Fix this

        cell.labelVoivodship.text = voivodshipName

        let selectors = getSelectors(for: voivodshipName)
        cell.labelSelectors.text = arrayToString(strings: selectors)

        if let firstSelector = selectors.first {
            cell.voivodshipImg.image = UIImage(named: firstSelector)
        } else {
            cell.voivodshipImg.image = nil
        }

        return cell
    } else {
        let cell = tableView.dequeueReusableCell(withIdentifier: "cellSearch", for:
indexPath) as! SearchTableViewCell
        cell.searchBar.delegate = self
        return cell
    }
}

override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    if indexPath.row < voivodships.count {
        selectedVoivodship = voivodships[indexPath.row]
        selectedSelectors = getSelectors(for: voivodships[indexPath.row])
    }
}

```

```

        performSegue(withIdentifier: "segueDistricts", sender: self)
    }
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    switch segue.identifier {
    case "segueDistricts":
        if let destination = segue.destination as? DistrictViewController {
            if let vships = dictionary?["Województwa"] as? NSDictionary {
                if let district = vships[selectedVoivodship] as? NSDictionary {
                    {
                        destination.districts = district
                        destination.title = selectedVoivodship
                        destination.selectors = selectedSelectors!
                        destination.selectedVoivodship = selectedVoivodship
                    }
                }
            }
        }
    case "segueSearchResults":
        if let destination = segue.destination as? DetailsViewController {
            destination.plates = foundDistrictPlates ?? []
            destination.selectors = selectedSelectors ?? []
            destination.title = foundDistrict
            destination.selectedVoivodship = selectedVoivodship
            destination.selectedDistrict = foundDistrict
        }
    default:
        return
    }
}

func getSelectors(for voivodship:String) -> [String] {
    if let root = dictionary?["Województwa"] as? NSDictionary {
        if let v = root[voivodship] as? NSDictionary {
            if let selectors = v["wyr"] as? [String] {
                var sel:[String] = []
                for s in selectors {
                    sel.append(s)
                }
                return sel
            }
        }
    }
    return []
}

func arrayToString(strings: [String]) -> String {
    guard strings.count > 0 else { return "" }
    var res = ""
    for s in strings {
        res.append(s)
        res.append(",")
    }
    let end = res.count-1

```



```

        return res[0..

```

## 4 Widok drugi - powiaty

W tym widoku (również UITableViewController) mamy jedną komórkę prototypową, która zawiera od lewej strony zdjęcie i nazwę powiatu. Dlatego trzeba stworzyć klasę odpowiedzialną za te komórki niestandardowe - DistrictTableViewCell.

```

import UIKit

class DistrictTableViewCell: UITableViewCell {

    override func awakeFromNib() {
        super.awakeFromNib()
        // Initialization code
    }

    @IBOutlet weak var districtImg: UIImageView!

    @IBOutlet weak var powLabel: UILabel!
    override func setSelected(_ selected: Bool, animated: Bool) {
        super.setSelected(selected, animated: animated)
    }
}

```

```

        // Configure the view for the selected state
    }

}

```

W tej klasie mamy tak samo jak poprzednio nadpisane dwie metody (automatycznie są dodawane) i "gniazdka" od zdjęcia i nazwy powiatu.

Teraz przejdziemy do umówienia klasy odpowiedzialnej za widok:

```

private var districtArray:[String] = []
private var selectedDistrict = ""
var selectors:[String] = []
var selectedVoivodship: String?

```

Na początku deklarujemy zmienne - tablice powiatów, wybrany powiat, oraz przekazane z poprzedniego widoku wyróżniki i wybrane województwo.

```

var districts:NSDictionary? {
    didSet {
        for k in districts!.allKeys {
            if let d = k as? String {
                if d != "wyr" {
                    districtArray.append(d)
                }
            }
        }
        districtArray.sort { $0.compare($1, locale: NSLocale.current) == .orderedAscending }
    }
}

```

Tutaj z przekazanej z poprzedniego widoku zmiennej pobieramy nazwy powiatów do tablicy (bez wyróżników)

---

```

override func viewDidLoad() {
    super.viewDidLoad()
    tableView.rowHeight = 70
    // Do any additional setup after loading the view.
}

```

Nadpisanie funkcji viewDidLoad i ustalenie wysokości wiersza na 70

---

```

override func numberOfSections(in tableView: UITableView) -> Int {
    return 1
}

override func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section: Int) -> Int {
    return districtArray.count
}

```

W tej części kodu, ustalamy, że tablema ma jedną kolumnę i ilość wierszy - tyle wierszy ile powiatów.

---

```

override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath)
-> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "cellDistrict", for: indexPath)

```

```

as! DistrictTableViewCell

let districtName = districtArray[indexPath.row]
cell.powLabel.text = districtName

if let plates = districts?[districtName] as? [String],
    let firstPlate = plates.first,
    let prefix = selectors.first {

    let fullPlate = "\\(prefix)\\(firstPlate)" // np. P + GN = PGN
    if let image = UIImage(named: fullPlate) {
        cell.districtImg.image = image
    } else {
        print("Brak obrazka dla: \\(fullPlate)")
        cell.districtImg.image = UIImage(named: "placeholder")
    }
} else {
    cell.districtImg.image = UIImage(named: "placeholder")
}

return cell
}

```

Na początku wybieramy, że komórka jest według wzoru `DistrictTableViewCell`. Następnie wybieramy nazwy powiatów i przypisujemy je do `Label`'a. Aby pobrać zdjęcie musimy mieć pełną nazwę rejestracji np. PO. Dlatego najpierw pobieramy pierwszy wyróżnik dla powiatu a następnie pierwszy wyróżnik dla województwa i łączymy to w całość i dodajemy zdjęcie. Na koniec zwracamy gotową komórkę.

---

```

override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    selectedDistrict = districtArray[indexPath.row]
    performSegue(withIdentifier: "segueDetails", sender: self)
}

```

Ta funkcja odpowiada za przygotowanie danych i określenie co się stanie po zaznaczeniu określonego rzędu.

---

```

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if let destination = segue.destination as? DetailsViewController {
        if let plates = districts?[selectedDistrict] as? NSArray {
            destination.plates = plates
            destination.selectors = selectors
            destination.title = selectedDistrict
            destination.selectedVoivodship = selectedVoivodship
            destination.selectedDistrict = selectedDistrict
        }
    }
}

```

Ta metoda jest używana tuż przed przejściem do następnego widoku. Przechodząc do `DetailsViewController` przekazujemy, kod tablicy rejestracyjnej, wyróżnik województwa, nazwę województwa i powiatu.

---

Cały kod klasy `DistrictViewController`:

```

import UIKit

```

```

class DistrictViewController: UITableViewController {

    private var districtArray:[String] = []
    private var selectedDistrict = ""
    var selectors:[String] = []
    var selectedVoivodship: String?

    var districts:NSDictionary? {
        didSet {
            for k in districts!.allKeys {
                if let d = k as? String {
                    if d != "wyr" {
                        districtArray.append(d)
                    }
                }
            }
            districtArray.sort { $0.compare($1, locale: NSLocale.current) == .orderedAscending }
        }
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        tableView.rowHeight = 70
        // Do any additional setup after loading the view.
    }

    override func numberOfSections(in tableView: UITableView) -> Int {
        return 1
    }

    override func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section: Int) -> Int {
        return districtArray.count
    }

    override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
    UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "cellDistrict", for: indexPath)
        as! DistrictTableViewCell

        let districtName = districtArray[indexPath.row]
        cell.powLabel.text = districtName

        if let plates = districts?[districtName] as? [String],
            let firstPlate = plates.first,
            let prefix = selectors.first {

            let fullPlate = "\\(prefix)\\(firstPlate)" // np. P + GN = PGN
            if let image = UIImage(named: fullPlate) {
                cell.districtImg.image = image
            } else {
                print("Brak obrazka dla: \\(fullPlate)")
                cell.districtImg.image = UIImage(named: "placeholder")
            }
        } else {

```

```

        cell.districtImg.image = UIImage(named: "placeholder")
    }

    return cell
}

override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    selectedDistrict = districtArray[indexPath.row]
    performSegue(withIdentifier: "segueDetails", sender: self)
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if let destination = segue.destination as? DetailsViewController {
        if let plates = districts?[selectedDistrict] as? NSArray {
            destination.plates = plates
            destination.selectors = selectors
            destination.title = selectedDistrict
            destination.selectedVoivodship = selectedVoivodship
            destination.selectedDistrict = selectedDistrict
        }
    }
}
}
}
}

```

## 5 Widok trzeci - szczegóły

W tej sekcji wyświetlane są informacje o nazwie powiatu i województwa oraz tabela z tablicami rejestracyjnymi. W odróżnieniu od poprzednich widoków, ten widok jest zwykłym `UIViewController` z dodaną tabelą. Daje nam to możliwość dodania elementów takij jak `Label`. W tym przypadku nie projektujemy komórek prototypowych ponieważ wyświetlamy tylko nazwy tablic rejestracyjnych.

```

var plates: NSArray = []
var selectors: [String] = []
var selectedVoivodship: String?
var selectedDistrict: String?

```

Zmienne, które są przekazane z poprzedniego widoku.

---

```

private var combinedPlates: [String] = [] // Docelowe tablice

// MARK: - Outlety
@IBOutlet weak var wojLabel: UILabel!
@IBOutlet weak var powiatLabel: UILabel!
@IBOutlet weak var table: UITableView!

```

Tabela docelowych tablic wyświetlana w `tableView` oraz "gniazdka" od nazwy województwa i powiatu.

---

```

override func viewDidLoad() {
    super.viewDidLoad()

    table.dataSource = self
    table.register(UITableViewCell.self, forCellReuseIdentifier: "plateCell")
}

```

```

        // Ustawienie etykiet
        wojLabel.text = selectedVoivodship ?? "Brak danych"
        powiatLabel.text = selectedDistrict ?? "Brak danych"
        self.title = selectedDistrict

        // Przetwarzanie danych
        loadPlates()
    }

```

W tej metodzie określamy ustawienia tabeli i etykiet oraz wywołujemy metodę `loadPlates()`. Ustawienia tabeli - `table.dataSource = self` oznacza że ta klasa steruje tabelą, a `table.register(UITableViewCell.self, forCellReuseIdentifier: "plateCell")` informuje że w tabeli będą komórki `UITableViewCell` o identyfikatorze "plateCell".

---

```

func loadPlates() {
    guard plates.count > 0 else {
        print("Brak plates")
        return
    }

    for selector in selectors {
        for i in 0..

```

Metoda widoczna powyżej ładuje tablice rejestracyjne. Jeżeli nie ma tablic to informuje nas o tym, a jeżeli są to mamy pętlę w pętli najpierw przechodzimy przez wyróżniki województwa a potem przez powiatów i sklejamy je i dodajemy to tablicy `combinedPlates`.

---

```

// MARK: - UITableViewDataSource

func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return combinedPlates.count
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "plateCell", for: indexPath)
    cell.textLabel?.text = combinedPlates[indexPath.row]
    return cell
}

```

Tutaj ustawiamy liczę wierszy w tabeli, a następnie tworzymy komórkę o identyfikatorze "plateCell" i dodajemy tablice z wcześniej stworzonej tabeli (`combinedPlates`).

---

Cały kod klasy `DetailsViewController`

```

import UIKit

class DetailsViewController: UIViewController, UITableViewDataSource {

    var plates: NSArray = []
    var selectors: [String] = []
    var selectedVoivodship: String?
    var selectedDistrict: String?

    private var combinedPlates: [String] = [] // Docelowe tablice

    // MARK: - Outleety
    @IBOutlet weak var wojLabel: UILabel!
    @IBOutlet weak var powiatLabel: UILabel!
    @IBOutlet weak var table: UITableView!

    override func viewDidLoad() {
        super.viewDidLoad()

        table.dataSource = self
        table.register(UITableViewCell.self, forCellReuseIdentifier: "plateCell")

        // Ustawienie etykiet
        wojLabel.text = selectedVoivodship ?? "Brak danych"
        powiatLabel.text = selectedDistrict ?? "Brak danych"
        self.title = selectedDistrict

        // Przetwarzanie danych
        loadPlates()
    }

    func loadPlates() {
        guard plates.count > 0 else {
            print("Brak plates")
            return
        }

        for selector in selectors {
            for i in 0..// MARK: - UITableViewDataSource

```

```

func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return combinedPlates.count
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "plateCell", for: indexPath)
    cell.textLabel?.text = combinedPlates[indexPath.row]
    return cell
}
}

```