

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI  
FACULTATEA DE INFORMATICĂ



# Aplicație pentru partajarea utilizării bicicletelor

**Cocei Tiberiu**

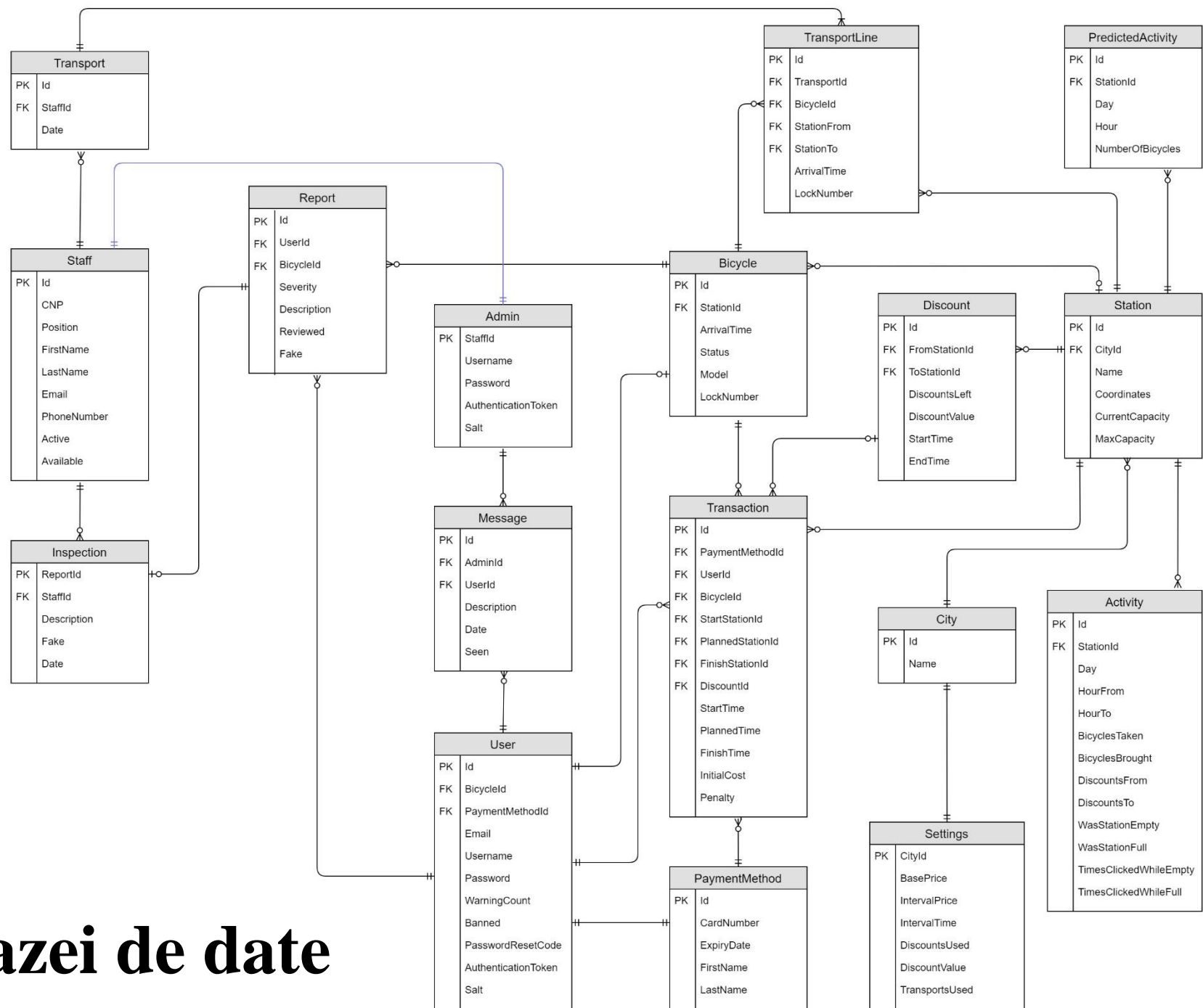
Coordonator științific  
**Lect. Dr. Cristian Frăsinaru**

# Cuprins

- Descrierea problemei
- Schema bazei de date
- Model C4
- Aplicațiile server și client
- Demo
- Concluzii
- Întrebări

# Descrierea problemei

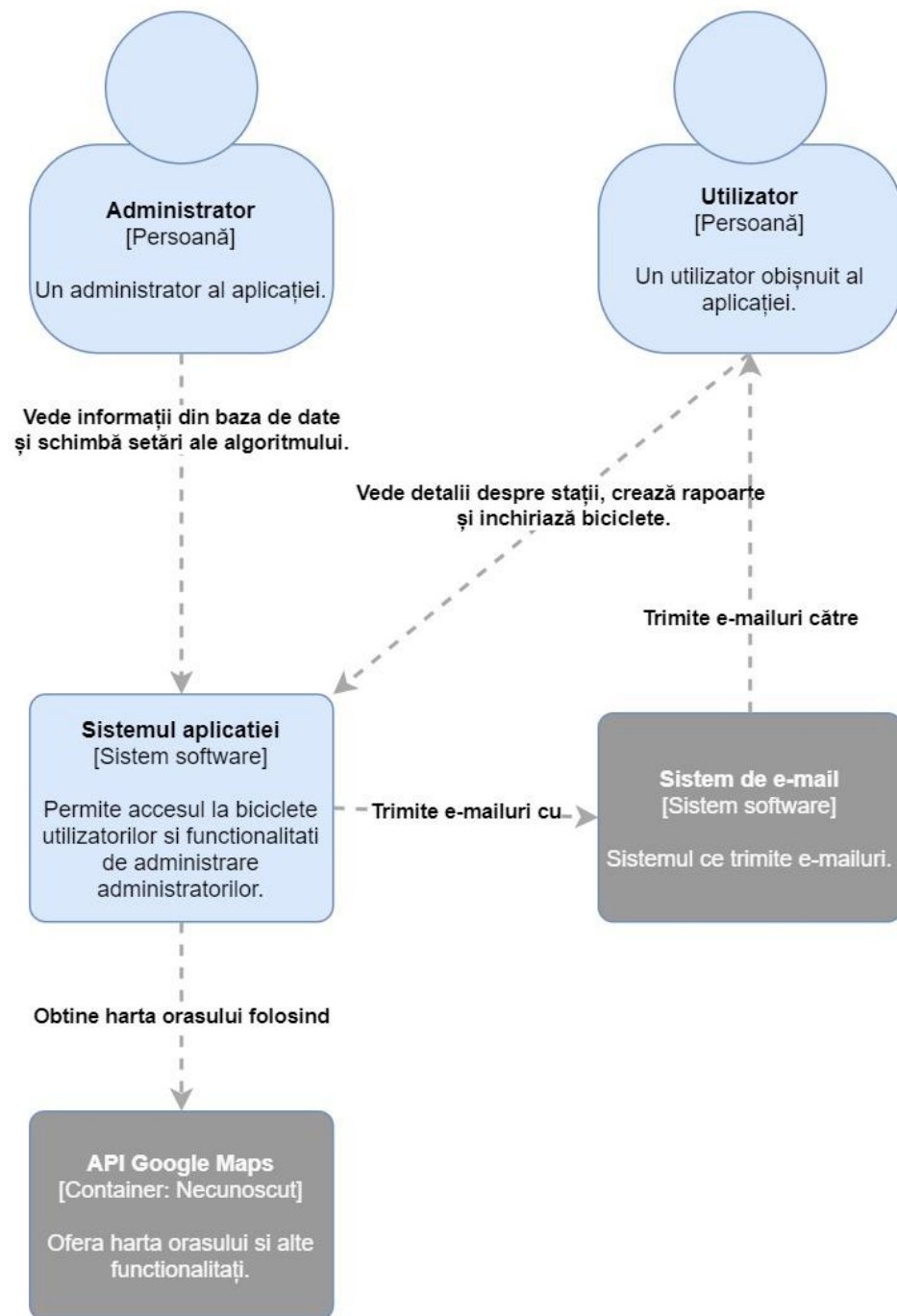




Schema bazei de date

# Model C4

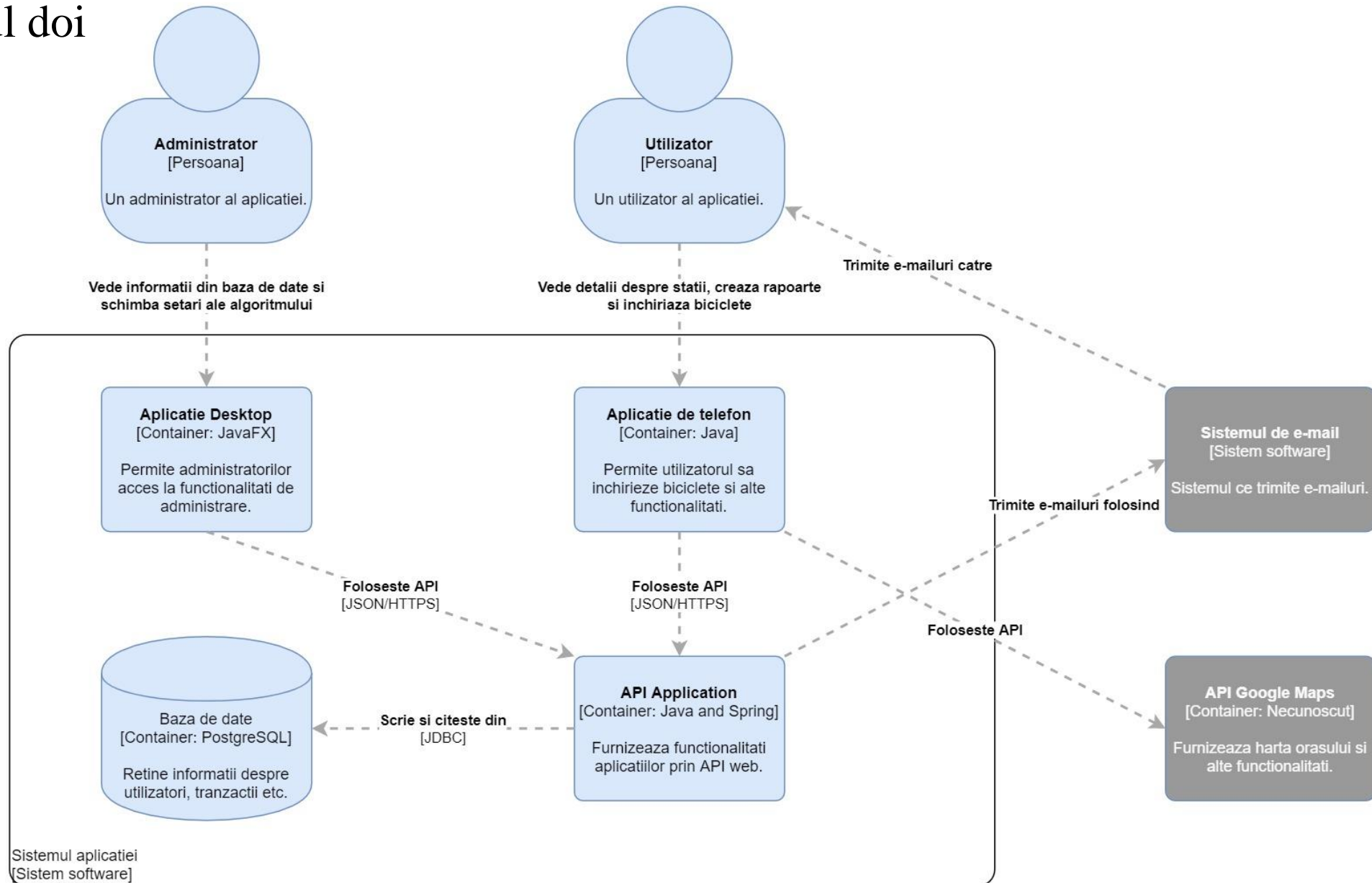
## Primul nivel





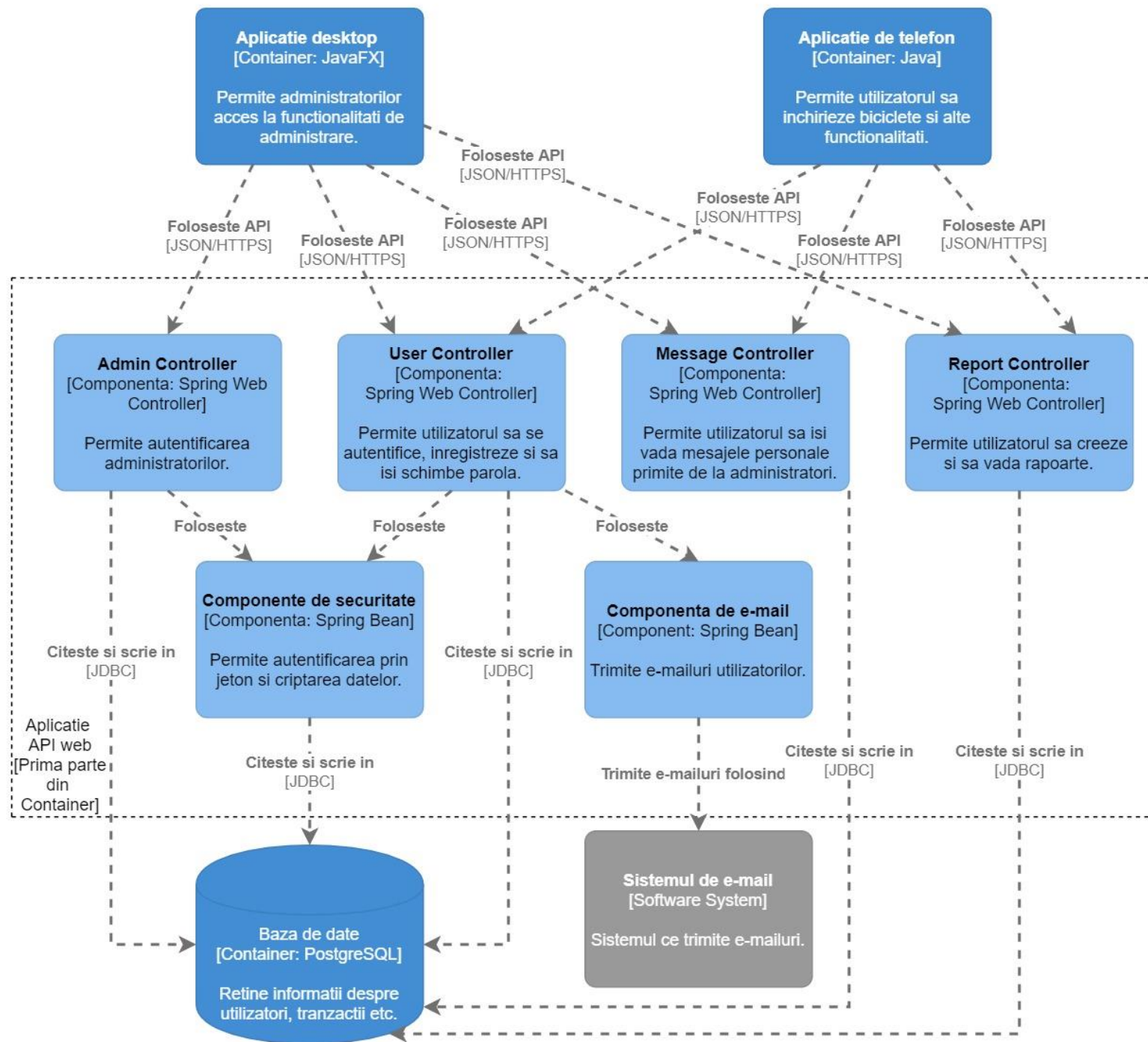
# Model C4

## Nivelul doi



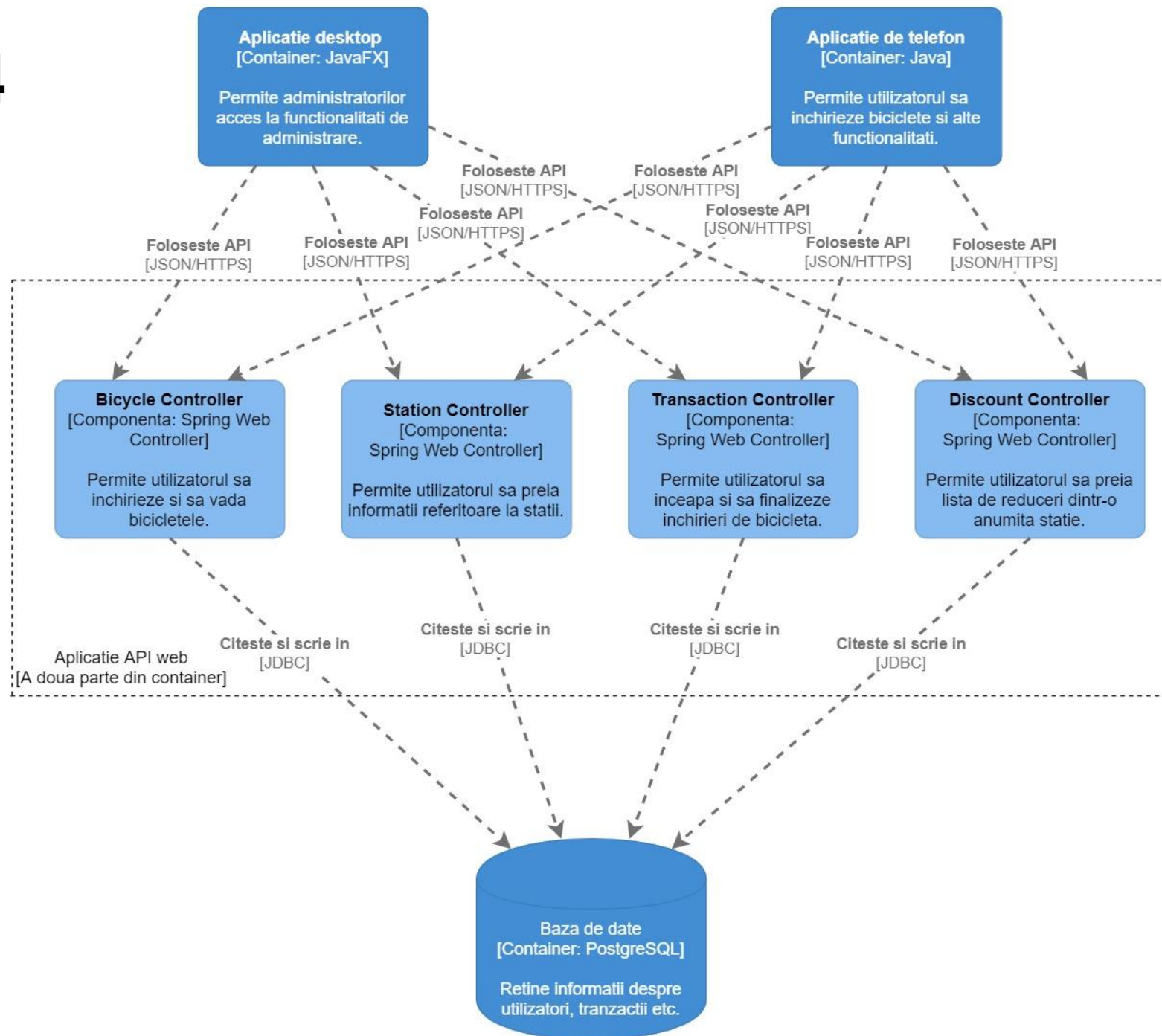
# Model C4

## Nivelul trei



# Model C4

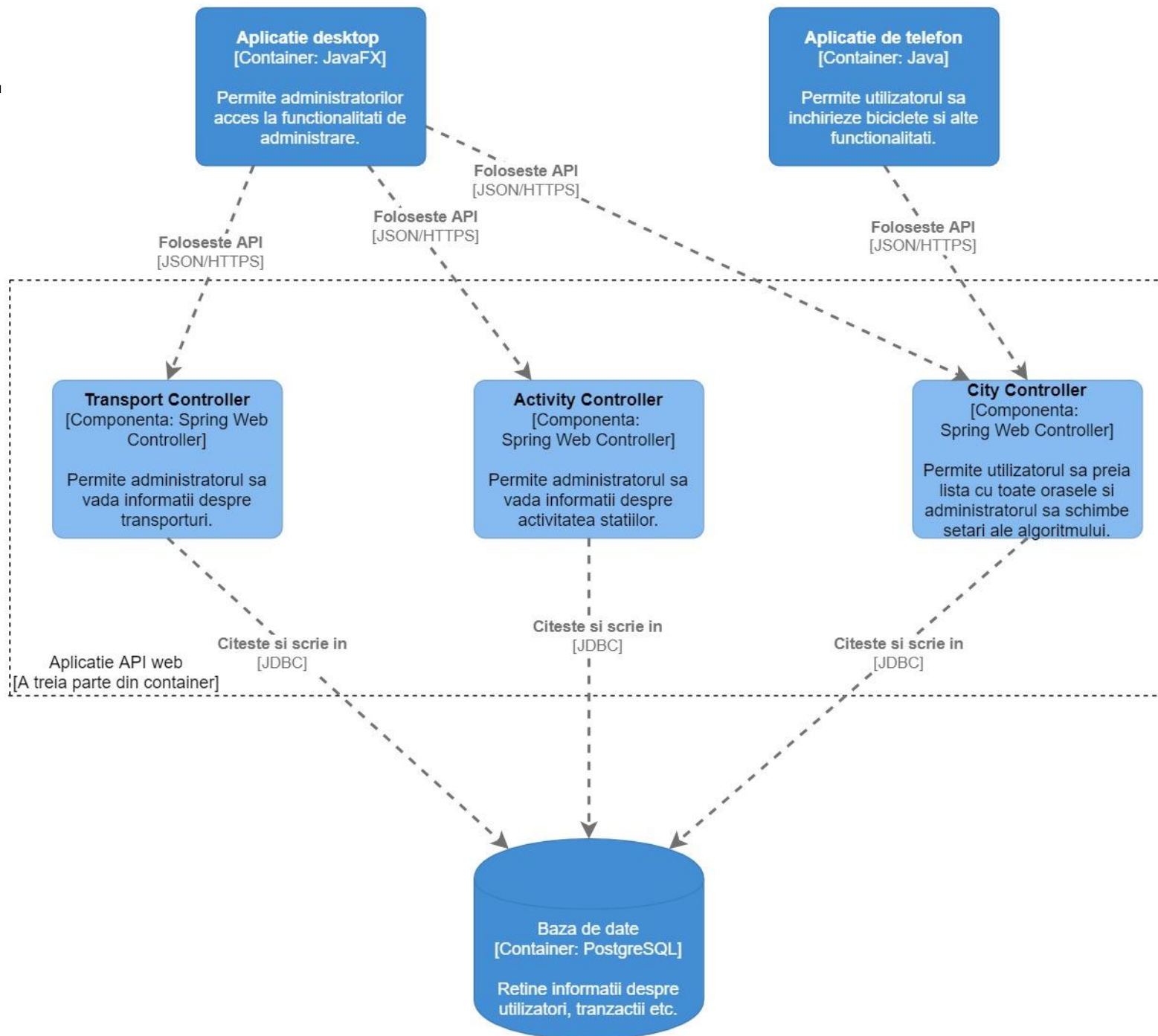
## Nivelul trei





# Model C4

## Nivelul trei



# Aplicația server - tehnologii utilizate



- Spring Email
- Spring Security
- Spring Boot JDBC și Data JPA
- Jasypt Library



```
spring.datasource.username=ENC(VEb/Dd0aAFH54s0NxHlurK/R3W/2FwYK)  
spring.datasource.password=ENC(JK9eYvoZGJ00LolgUbS/aECkoErS3kjB)
```

# Aplicația server - testare

- Mockito, AssertJ și JUnit

```
@Mock
MessageRepository messageRepository;

@InjectMocks
MessageServiceImpl messageService;
```

```
@Before
public void setup() {
    MockitoAnnotations.initMocks( testClass: this);

    userId = UUID.randomUUID();
    Message firstMessage = new Message(UUID.randomUUID(), userId, description: "First");
}
```

```
@Test
public void whenMarkMessageAsSeenIsCalled_WithExistingId_ThenReturnCorrectResponse() {
    //Arrange
    Mockito.when(messageRepository.getUnseenMessagesByUserId(message.getId())).thenReturn(messageList);

    //Act
    ResponseEntity<String> result = messageService.markMessagesAsSeen(userId);

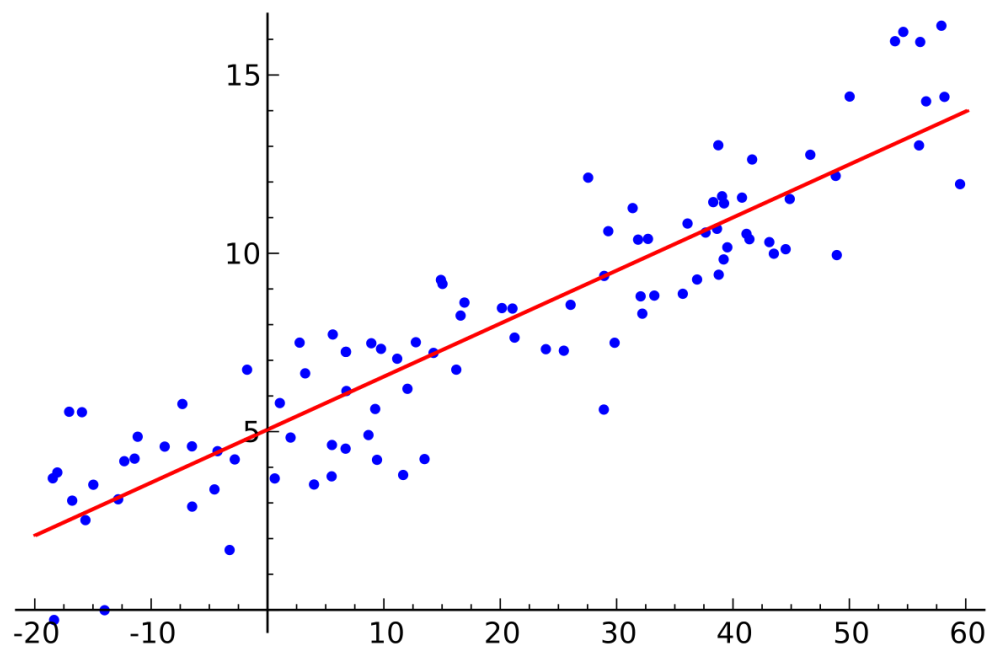
    //Assert
    Assertions.assertThat(result).isNotNull();
    Assertions.assertThat(result.getBody()).isEqualTo("Successfully marked messages as seen.");
    Assertions.assertThat(result.getStatusCode()).isEqualToComparingTo(HttpStatus.OK);
}
```

```
@After
public void tearDown() {
    userId = null;
    messageList = null;
    message = null;
}
```

# Aplicația server - algoritmi

- Algoritmul de predicție a numărului de biciclete

**Regresie liniara**  
Ecuația  $Y = a + bX$





# Aplicația server - algoritmi

- Algoritmul de predicție a numărului de biciclete

## Algoritmul utilizat

```
ArrayList<Integer> calculatedHourlyBicycleNr = new ArrayList<>();
for(Activity activity : pastActivities) {
    int calculatedBicycleNr = 0;
    calculatedBicycleNr += activity.getBicyclesTaken();
    calculatedBicycleNr -= activity.getBicyclesBrought() * 0.5;
    calculatedBicycleNr += activity.getDiscountsTo() * 0.5;
    calculatedBicycleNr -= activity.getDiscountsFrom() * 0.5;
    calculatedBicycleNr += activity.getTimesClickedWhileEmpty() * 0.0025;
    calculatedBicycleNr -= activity.getTimesClickedWhileFull() * 0.0025;

    if(calculatedBicycleNr < minVal) {
        calculatedBicycleNr = minVal;
    }
    else if(calculatedBicycleNr > maxVal) {
        calculatedBicycleNr = maxVal;
    }

    calculatedHourlyBicycleNr.add(calculatedBicycleNr);
}
```

```
double predictedBicycleNr = 0.0d;
double currentRate = 1;
double rateSum = 0;
for(int calculatedBicycleNr : calculatedHourlyBicycleNr) {
    if(currentRate <= 0) {
        break;
    }
    predictedBicycleNr += (float) calculatedBicycleNr * currentRate;
    rateSum += currentRate;
    currentRate *= 0.975;
}
```

```
if(rateSum > 0) {
    predictedBicycleNr /= rateSum;
    PredictedActivity predictedActivity = new PredictedActivity(
        stationId, day, hour, (int)Math.round(predictedBicycleNr));
    predictedActivityRepository.save(predictedActivity);
}
```

# Aplicația server - algoritmi

- Algoritmul de re poziționare a bicicletelor

```
private void createDiscountsAndTransports(HashMap<Station, Integer> positiveDifference,
                                          HashMap<Station, Integer> negativeDifference,
                                          List<Staff> staffList,
                                          Settings settings,
                                          boolean forceDiscounts) {
```

```
    for(Map.Entry<Station, Integer> positivePair : positiveDifference.entrySet()) {
        int maxValueForDiscount = (int) (positivePair.getKey().getMaxCapacity() * 0.2);
        if((positivePair.getValue() <= maxValueForDiscount || staffList.size() == 0 || forceDiscounts) && settings.areDiscountsUsed()) {
            for(Map.Entry<Station, Integer> negativePair : negativeDifference.entrySet()) {
```

```
                else if(settings.areTransportsUsed()) {
                    List<Bicycle> availableBicycles = bicycleRepository.getAvailableBicycles(positivePair.getKey().getId());
                    positivePair.setValue(Math.min(availableBicycles.size(), positivePair.getValue()));
                    for(Map.Entry<Station, Integer> negativePair : negativeDifference.entrySet()) {
```

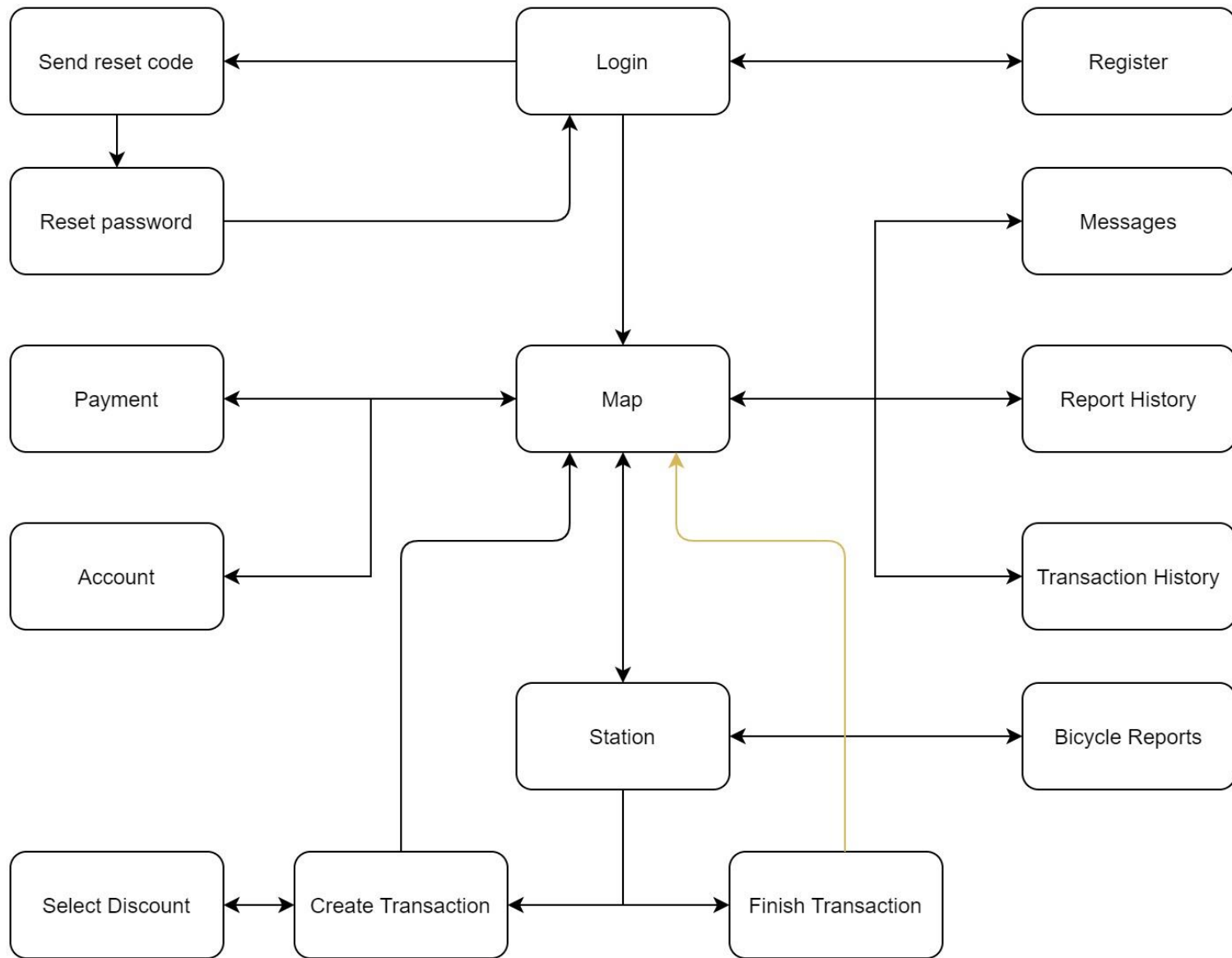
```
                        if(!forceDiscounts) {
                            createDiscountsAndTransports(positiveDifference, negativeDifference, staffList, settings, forceDiscounts: true);
                        }
                    }
                }
            }
        }
```

```
Attempting to bring station bicycle numbers to predicted values for the city Iasi...
Successfully created discount from the station Tatarasi to the station Poitiers. Number of discounts: 4
Successfully created discount from the station Palas to the station Poitiers. Number of discounts: 4
Successfully created discount from the station Copou to the station Poitiers. Number of discounts: 8
```

# Aplicația Android - tehnologii utilizate



# Aplicația Android - fluxul paginilor

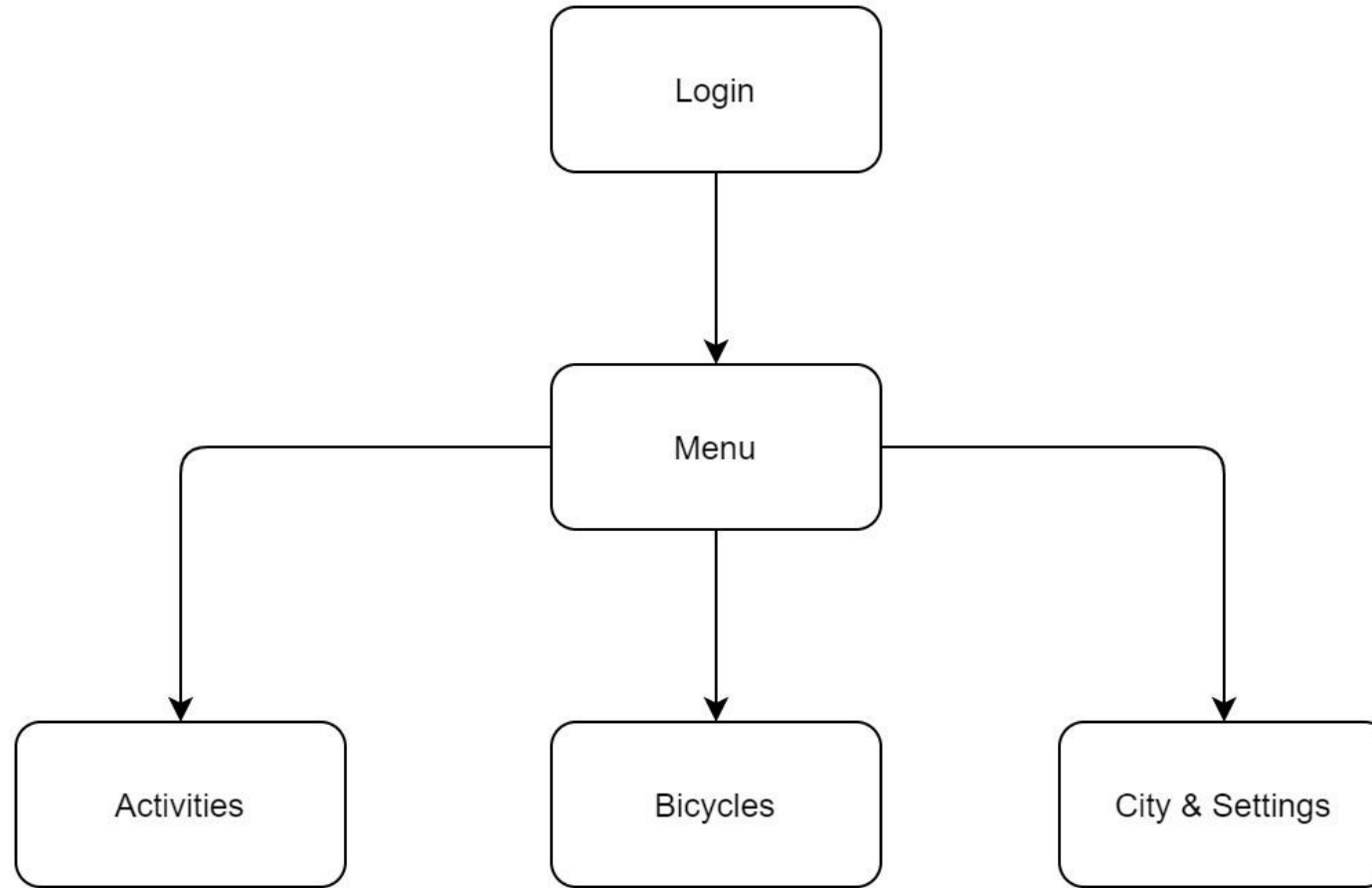




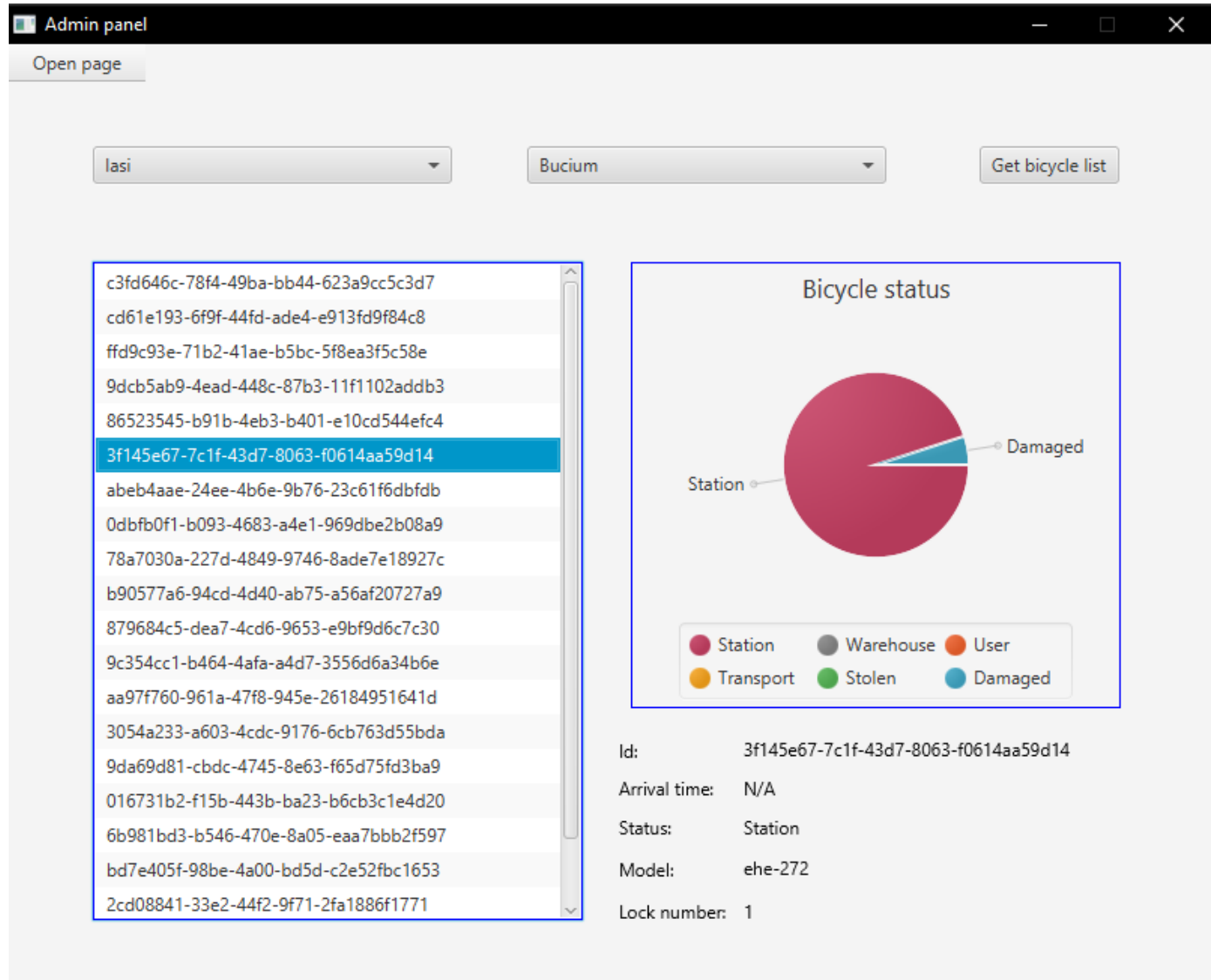
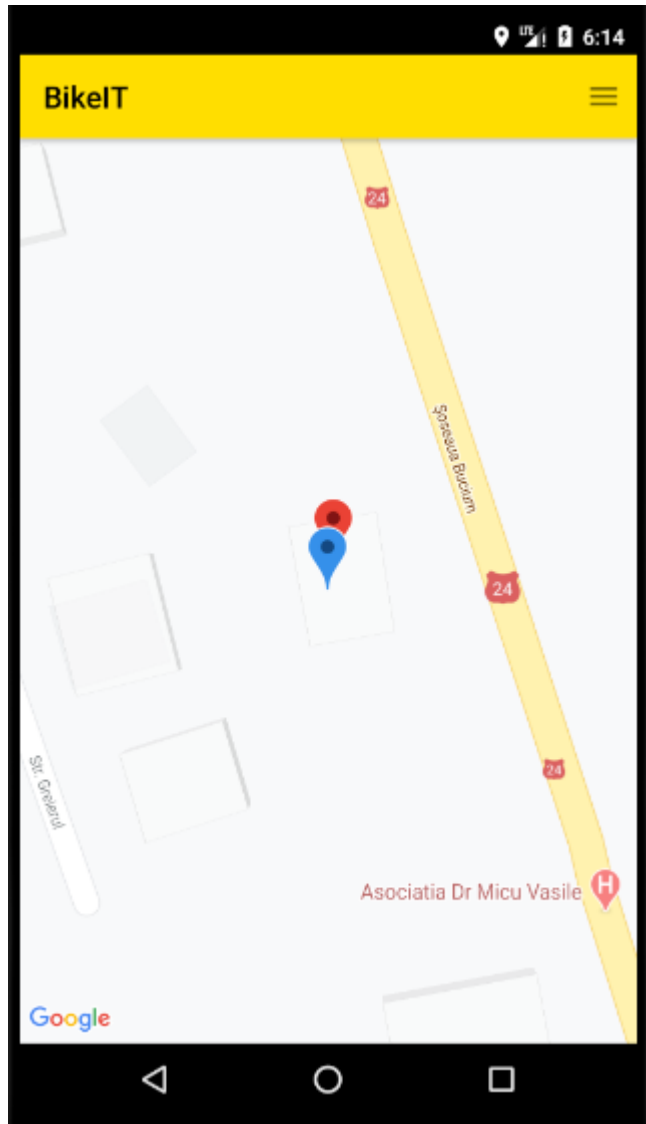
# Aplicația desktop - tehnologii utilizate



# Aplicația desktop - fluxul paginilor



# Demo



# Concluzii

- Aplicația server are toate endpoint-urile necesare
- Aplicația Android este completă și funcțională
- Aplicația desktop este parțial implementată
- Algoritmul de predicție și de re poziționare a bicicletelor sunt funcționale

# Direcții de viitor

- Îmbunătățirea algoritmului de predicție sau folosirea unei rețele neuronale
- Găsirea celui mai eficient drum posibil pentru transporturile de biciclete
- Implementarea în întregime a aplicației desktop
- Îmbunătățirea CSS-ului în aplicațiile client



Întrebări