Business Service Implementation Task

Database Schema

You'll need to work with the following tables:

`business`

Column	Type	Description
business_id	SERIAL	Primary key
user_id	INTEGER	Foreign key to users table
business_name	VARCHAR(255)	Name of the business
business_type	VARCHAR(100)	Type of business
registration_number	VARCHAR(100)	Business registration number
gst_id	VARCHAR(100)	GST identification
business_address	TEXT	Physical address
credit_balance	DECIMAL(10, 2)	Available credit balance

`pickup_request`

Column	Type	Description
request_id	SERIAL	Primary key
business_id	INTEGER	Foreign key to business
waste_type	VARCHAR(100)	Type of waste for pickup
estimated_volume	FLOAT	Estimated waste volume in kg

preferred_pickup_date	DATE	Requested pickup date
preferred_time_slot	VARCHAR(50)	Morning/Afternoon/Evening
pickup_address	TEXT	Address for pickup
status	VARCHAR(50)	Current request status
created_at	TIMESTAMP	Creation timestamp

business_payment

Column	Type	Description
payment_id	SERIAL	Primary key
business_id	INTEGER	Foreign key to business
amount	DECIMAL(10, 2)	Payment amount
payment_method	VARCHAR(50)	Payment method used
transaction_id	VARCHAR(100)	External transaction ID
status	VARCHAR(50)	Payment status
created_at	TIMESTAMP	When payment was created

business_invoice

Column	Type	Description
invoice_id	SERIAL	Primary key
business_id	INTEGER	Foreign key to business
service_id	INTEGER	Related service ID
amount	DECIMAL(10, 2)	Invoice amount

status	VARCHAR(50)	Invoice status
due_date	DATE	Payment due date
issue_date	TIMESTAMP	When invoice was issued

Required API Endpoints

Profile Management

1. `GET /profile` - Get business profile details
2. `PUT /profile` - Update business profile

Pickup Requests

3. `POST /pickup-requests` - Create a new pickup request
4. `GET /pickup-requests` - List all pickup requests with filtering
5. `GET /pickup-requests/{id}` - Get specific pickup request details
6. `PUT /pickup-requests/{id}/cancel` - Cancel a pickup request

Payment and Billing

7. `GET /invoices` - List all invoices with filtering
8. `GET /invoices/{id}` - Get specific invoice details
9. `POST /payments` - Process a new payment
10. `GET /payments` - View payment history

Analytics and Reporting

11. `GET /dashboard` - Get business dashboard metrics
12. `GET /reports/waste` - Generate waste collection report
13. `GET /reports/sustainability` - Generate sustainability metrics

Payment Integration

Implement payment processing using our payment gateway integration:

1. Credit card processing (Stripe integration)
2. Wallet balance management
3. Invoice generation and tracking

4. Payment receipt generation

Kafka Events

Your implementation should produce these events:

- `PickupRequestCreated` - When a business creates a new pickup request
- `PickupRequestCancelled` - When a business cancels a pickup request
- `PaymentProcessed` - When a business makes a payment
- `InvoiceGenerated` - When an invoice is generated for a business

And consume these events:

- `PickupRequestAssigned` - When a collector is assigned to a request
- `TripScheduled` - When a trip is scheduled for a pickup
- `TripCompleted` - When a waste collection trip is completed
- `PaymentConfirmed` - When payment processing is confirmed

Code Structure

Follow this structure for your implementation:

```
internal/business/
├── handler.go      # HTTP API handlers
├── service.go      # Business logic
├── repository.go   # Database operations
├── events.go       # Kafka event producer/consumer
├── payment.go      # Payment processing integration
├── reports.go      # Report generation functions
└── models.go       # Data structures
```

Additional Guidelines

- Validate all incoming request data
- Include pagination for list endpoints
- Implement proper error handling with appropriate HTTP status codes
- Use prepared statements for all database queries
- Add logging for important operations
- Ensure payment processing follows security best practices
- Write unit tests for critical functions, especially payment processing

Integration Points

Your implementation will need to interact with:

1. User service (for authentication)
2. Collector service (for assignment information)
3. Payment service (for payment processing)
4. Notification service (for sending confirmations)
5. Analytics service (for report generation)

Contact the project lead if you need specific details about these integration points.