Task Management: API Design Frontend Integration

Queued Active Tasks and Completed Tasks

- **API:** GET “http://localhost:8080/api/v1/queues/stats” (Already Existing Endpoint)
- **Input:** None
- **Output:** JSON object representing **QueueStatsDTO** containing overall queue stats.
  eg: {

```
"timestamp": "2025-09-13 12:24:00",
"totalQueues": 3,
"totalPendingItems": 42,          // Total queued (unassigned) tasks
"totalProcessingItems": 8,        // Total active (processing) tasks
"totalCompletedItems": 120,
"totalFailedItems": 5,
"queueSizes": {
  "manual_processing_queue": 25,
  "success_queue": 10,
  "error_queue": 7
},
"priorityDistribution": {
  "PREMIUM (9)": 5,
  "HIGH (8)": 15,
  "NORMAL (5)": 25,
  "LOW (2)": 5
},
"statusDistribution": {
  "PENDING": 42,
  "PROCESSING": 8,
  "PROCESSED": 120,
  "FAILED": 5
},
"avgProcessingTimeMinutes": 37,
"systemStatus": "HEALTHY"
   }
```

1. Average SLA Transaction Queue Tab

- **API:** GET “http://localhost:8080/api/v1/transactions/all”
- **Input:** None
- **Output:**

JSON array of **Transactions**

Eg:{{

```
  id: 'trans-1',
  client_id: '1',
  transaction_reference: 'HP-INV-2024-001',
  task_type: 'single_document',
```

```
  priority: 'high',
  source: 'api',
  received_at: '2024-01-25T09:30:00Z',
  sla_deadline: '2024-01-26T09:30:00Z',
  is_frozen: false,
  created_at: '2024-01-25T09:30:00Z',
  updated_at: '2024-01-25T09:30:00Z',
  client: mockClients[0],
  assignment: mockAssignments[0]
 },
 {
  id: 'trans-2',
  client_id: '2',
  transaction_reference: 'GL-SHIP-2024-045',
  task_type: 'multi_document',
  priority: 'normal',
  source: 'ftp',
  received_at: '2024-01-25T10:45:00Z',
  sla_deadline: '2024-01-27T10:45:00Z',
  is_frozen: false,
  created_at: '2024-01-25T10:45:00Z',
  updated_at: '2024-01-25T10:45:00Z',
  client: mockClients[1],
  assignment: mockAssignments[1]
 },
 {
  id: 'trans-3',
  client_id: '3',
  transaction_reference: 'FS-KYC-2024-012',
  task_type: 'process_discovery',
  priority: 'urgent',
  source: 'api',
  received_at: '2024-01-25T11:15:00Z',
  sla_deadline: '2024-01-25T23:15:00Z',
  is_frozen: false,
  created_at: '2024-01-25T11:15:00Z',
  updated_at: '2024-01-25T11:15:00Z',
  client: mockClients[2],
  assignment: mockAssignments[2]
 },
 {
  id: 'trans-4',
  client_id: '1',
  transaction_reference: 'HP-CLAIM-2024-078',
  task_type: 'single_document',
  priority: 'normal',
  source: 'api',
  received_at: '2024-01-25T07:00:00Z',
```

```
    sla_deadline: '2024-01-26T07:00:00Z',
    is_frozen: false,
    created_at: '2024-01-25T07:00:00Z',
    updated_at: '2024-01-25T09:30:00Z',
    client: mockClients[0],
    assignment: mockAssignments[3]
  },
  {
    id: 'trans-5',
    client_id: '2',
    transaction_reference: 'GL-POD-2024-156',
    task_type: 'multi_document',
    priority: 'high',
    source: 'ftp',
    received_at: '2024-01-25T12:30:00Z',
    sla_deadline: '2024-01-27T12:30:00Z',
    is_frozen: false,
    created_at: '2024-01-25T12:30:00Z',
    updated_at: '2024-01-25T14:00:00Z',
    client: mockClients[1],
    assignment: mockAssignments[4]
  },
  {
    id: 'trans-6',
    client_id: '3',
    transaction_reference: 'FS-LOAN-2024-089',
    task_type: 'single_document',
    priority: 'critical',
    source: 'api',
    received_at: '2024-01-25T14:00:00Z',
    sla_deadline: '2024-01-26T02:00:00Z',
    is_frozen: false,
    created_at: '2024-01-25T14:00:00Z',
    updated_at: '2024-01-25T14:00:00Z',
    client: mockClients[2],
    assignment: {
      id: 'assign-6',
      transaction_id: 'trans-6',
      status: 'queued',
      time_spent_seconds: 0,
      created_at: '2024-01-25T14:00:00Z',
      updated_at: '2024-01-25T14:00:00Z'
    }
  }
}
```

Average SLA:

Include the transaction in the result if:

- Its SLA deadline has passed (deadline date is earlier than now), and
- The related assignment status is **not** 'completed'.

Fetching all the transactions and filtering can be done in the backend part itself as discussed in yesterdays meet.

2. User Management

Get All Users

- **API:** GET “http://localhost:9190/api/v1/users/all”
- **Input:** None
- **Output:** JSON response of details of all users.
- **Actually GET** “http://localhost:9190/api/v1/users/all” gets the users in hierarchial order (uses UserResponseDTO), but the response doesn’t contain the property is_logged_in. So, instead of creating new endpoint, we can add a property called is_logged_in in it’s response.

JSON Eg: {{
  id: 'user-1',
  username: 'john.doe',
  email: 'john.doe@company.com',
  full_name: 'John Doe',
  role: 'digitization_agent',
  is_active: true,
  is_logged_in: true,
  last_login: '2024-01-25T09:00:00Z',
  last_activity: '2024-01-25T14:30:00Z',
  created_at: '2024-01-01T00:00:00Z',
  updated_at: '2024-01-25T14:30:00Z'
 },
{
  id: 'user-1',
  username: 'john.doe',
  email: 'john.doe@company.com',
  full_name: 'John Doe',
  role: 'digitization_agent',
  is_active: true,
  is_logged_in: true,
  last_login: '2024-01-25T09:00:00Z',
  last_activity: '2024-01-25T14:30:00Z',
  created_at: '2024-01-01T00:00:00Z',

```
    updated_at: '2024-01-25T14:30:00Z'
  }
}
```

3. Performance

Tasks Completed Today, Team Productivity, Avg Time

- **API:** GET “http://localhost:8080/api/v1/transactions/all” :get all transactions
- **Input:** None
- **Output:** JSON array of transaction objects (TransactionDTO format)

Get the completed tasks count by filtering with **status==”Completed” and checking the date.**

Productivity =112% if completed tasks>10 else 98%, “above target” if >10 else “below target”

Avg Time Taken to be calculated by (Total time taken today/Tasks completed today). But Transaction DTO doesn’t mention anything about TaskAssignment

```
const totalTimeToday = completedToday.reduce((acc, t) => acc + (t.assignment?.time_spent_seconds || 0),
const avgTaskTime = completedToday.length > 0 ? totalTimeToday / completedToday.length : 0;
```

Quality Score

Get List of all Performance Metrics

- API: GET “http://localhost:8080/api/v1/users/performance-metrics"
- Input: None
- Output: List of JSON objects performance-metrics of all individual users

```
const avgQualityScore = performanceMetrics.length > 0
  ? performanceMetrics.reduce((acc, m) => acc + (m.quality_score || 0), 0) / performanceMetrics.length
  : 0;
```

avgQualityScore > 90 ? '+2% vs yesterday' : 'Needs improvement'

For getting performance metrics of Individual users

- **API:** GET “http://localhost:8080/api/v1/users/{userId}/performance-metrics" (Already existing endpoint)
- **Input:** userId(UUID)
- **Output:**

  avgQualityScore > 90 ? '+2% vs yesterday' : 'Needs improvement'

user_id property should be added to the output

JSON output eg: {


```
  "averageCompletionTime": 5400.75,
  "qualityScore": 88.50,
  "tasksCompleted": 45,
  "tasksInProgress": 5,
  "successRate": 95.56,
  "averageQualityScore": 88.50,
  "totalWorkingHours": 67.25,
  "mostRecentActivity": "2025-09-12T16:45:00",
  "experienceLevel": "INTERMEDIATE",
  "performanceRating": "GOOD"
}
```


Individual Performance Today

- Fetch all performance metrics
- Map User name, tasks completed, averaget time, quality score

Team Efficiency Metrix

- Avg Processing Time same as Avg Task Time above mentioned
- Active Users : Get all users, filter logged in users
- Total Throughput = (Total tasks completed today) Task/Day
- avgQualityScore > 90 ? 'Target Met' : 'Below Target'

4.Priority Management

To be updated. I will update as I work along integrating this tab.

5. SLA Monitoring

API call required:

On Time, At Risk, Breached

- Get all the transactions(refer Transaction Queue tab). Map through them and get the count of tasks completed OnTime, at Risk and Breached.

SLA Alerts & Critical Transactions

- Filter all transactions whose deadlines are breached or are at risk.

s