**Division / Team:** GDP-Platform (Social) / One Client View (OneCV)

**Brief Description of your Div/ Team:** OneCV is a system enabler which pulls data from a range of agencies to provide frontline officers a comprehensive view of clients that approached them for aid. This enables officers to customise assistance and/or refer their clients to appropriate support avenue expeditiously.

**Assessment Title:** Coding Assignment: Financial Assistance Scheme Management System

**Overview/Background:** Develop a backend application that will be part of a system to allow administrators to manage financial assistance schemes and applications for schemes.

(Scroll to next page for assessment details)

# Coding Assignment: Financial Assistance Scheme Management System

## Introduction

As part of our internship hiring process, we require candidates to complete a coding assignment that demonstrates their technical capabilities. This assignment involves building a backend solution for managing financial assistance schemes for needy individuals and families. **The backend logic must be written in Go**.

## Objective

The goal of this assignment is to build a system that

* Allows the management of a fictitious set of financial assistance schemes.
* Save and update records of applicants who applied for schemes.
* The system should advise users of schemes that each applicant can apply for.
* The system should also save the outcome of granting of schemes to applicants.

## 

## Requirements

As this coding assignment is focused on management of financial assistance schemes and applications of schemes, system user login is omitted from assignment requirements.

### Backend

1. **Database Design**
   1. Design a relational database to store the information about financial assistance schemes and the beneficiaries (individuals and families).
   2. The database should include, but is not limited to the following tables:

|  |  |
| --- | --- |
| **Table Name** | **Purpose** |
| Applicants | Details of individuals who applied for financial assistance schemes. |
| Schemes | Information about different financial assistance schemes available. |
| Applications | Records and statuses of applications made by individuals for different schemes. |

1. **Database Design - Applicants and Schemes**Please take note of the following requirements when designing the **Applicants** and **Schemes** tables:
   1. Schemes available for an applicant to apply for depends on the following factors:
      1. Marital status: Single, Married, Widowed, Divorced
      2. Employment status: Employed, Unemployed
      3. Household family members: Family members staying in the same household with applicant and their relationship with applicant
   2. For example, assuming that the following financial assistance schemes (for illustration purpose only) are available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Scheme Name** | **Description** | **Eligibility Criteria** | **Benefits** |
| Retrenchment assistance scheme | Financial assistance for retrenched workers | * Citizens who are recently retrenched. * School meal vouchers are only for applicants with school-going children. | * Additional SkillsFuture credits * Additional CDC vouchers * Daily school meal vouchers for applicants (for applicants with children attending primary school only) |

And the following applicants are applying for financial assistance schemes:

|  |  |
| --- | --- |
| **Applicant** | **Background** |
| **James** | Single. Recently retrenched and still on job search. |
| **Mary** | Married with 2 kids aged 8 and 6. Recently retrenched and still on job search. |

* 1. Both James and Mary are eligible for the retrenchment assistance scheme as they are recently retrenched.
  2. Mary’s eldest child is eligible to receive daily school meal vouchers since the child is attending primary school.

1. **API Endpoints**
   1. Implement RESTful API endpoints that can support the CRUD operations for managing Beneficiaries, Schemes and Applications.
   2. Refer to the following table for required endpoints to build (you may add more endpoints where needed, but endpoints listed in the following table must be included in solution):

|  |  |  |
| --- | --- | --- |
| **Method** | **Path** | **Purpose** |
| GET | /api/applicants | Get all applicants. |
| POST | /api/applicants | Create a new applicant. |
| GET | /api/schemes | Get all schemes. |
| GET | /api/schemes/eligible?applicant={id} | Get all schemes that an applicant (represented by applicant query string parameter) is eligible to apply for. |
| GET | /api/applications | Get all applications. |
| POST | /api/applications | Create a new application. |

* 1. You may include the following optional endpoints if time permits:

|  |  |  |
| --- | --- | --- |
| **Method** | **Path** | **Purpose** |
| PUT | /api/applicants/{id} | Update an applicant’s details. |
| DELETE | /api/applicants/{id} | Delete an applicant. |
| POST | /api/schemes | Create a new scheme |
| PUT | /api/schemes/{id} | Update scheme details. |
| DELETE | /api/schemes/{id} | Delete a scheme. |
| PUT | /api/applications/{id} | Update application details. |
| DELETE | /api/applications/{id} | Delete an application. |

### User Stories

Your solution must fulfill the following user stories:

#### 1. As a system administrator, I want to view all available financial assistance schemes.

System administrators should be able to view all available financial assistance schemes.

Endpoints involved:

* GET /api/schemes

#### 2. As a system administrator, I want to add new scheme applicants.

Before applying for a financial assistance scheme, the applicant’s data should be saved into the system first.

Endpoints involved:

* POST /api/applicants

#### 3. As a system administrator, I want to view the schemes that an applicant is eligible to apply for.

Upon starting a new application for a financial assistance scheme applicant, the system administrator should be able to view the list of schemes that the applicant is eligible for based on their data.

Endpoints involved:

* GET /api/schemes/eligible?applicant={id}

#### 4. As a system administrator, I want to submit a new application on behalf of the applicant, based on schemes selected by them.

System administrators can choose schemes to grant to applicants and save the selected schemes as part of the submitted application.

Endpoints involved:

* POST /api/applications

## Sample Data

The following are some samples of applicants and financial assistance schemes data in a JSON format. You may use them for reference purposes when designing the database schema designs and API responses. You may come up with your own schema designs and response formats, as long as the assignment requirements are met.

|  |
| --- |
| // Beneficiaries  {  "applicants": [  {  "id": “01913b7a-4493-74b2-93f8-e684c4ca935c”,  "name": "James",  "employment\_status": "unemployed",  “sex”: “male”,  “date\_of\_birth”: ”1990-07-01”  “household”: []  },  {  "id": “01913b80-2c04-7f9d-86a4-497ef68cb3a0”,  "name": "Mary",  "employment\_status": "unemployed",  “sex”: “female”,  “date\_of\_birth”: ”1984-10-06”  “household”: [  {  “id”: “01913b88-1d4d-7152-a7ce-75796a2e8ecf”  "name": "Gwen",  "employment\_status": "unemployed",  “sex”: “female”,  “date\_of\_birth”: ”2016-02-01”  “relation”: “daughter”  },  {  “id”: “01913b88-65c6-7255-820f-9c4dd1e5ce79”  "name": "Jayden",  "employment\_status": "unemployed",  “sex”: “male”,  “date\_of\_birth”: ”2018-03-15”  “relation”: “son”  }  ]  }  ],  }  // Schemes  {  "schemes": [  {  "id": “01913b89-9a43-7163-8757-01cc254783f3”,  "name": "Retrenchment Assistance Scheme",  "criteria": {  “employment\_status”: “unemployed”  },  “benefits”: [  {  “id”: ”01913b8b-9b12-7d2c-a1fa-ea613b802ebc”  “name”: ”SkillsFuture Credits”,  “amount”: 500.00  }  ]  },  {  "id": “01913b89-befc-7ae3-bb37-3079aa7f1be0”,  "name": "Retrenchment Assistance Scheme (families)",  "criteria": {  “employment\_status”: “unemployed”,  “has\_children”: {  “school\_level”: “== primary”  }  }  }  ]  } |

## Deliverables

### Code Repository

* Provide a link to a public or private code repository (GitHub, GitLab, etc.) with your complete codebase.
* Ensure the repository contains a clear README.md with instructions on how to set up and run the application in local development environment.

### API Documentation

* Include documentation for the API endpoints using a tool like Swagger or Postman.
* Ensure the documentation is clear and easy to follow.

## Evaluation Criteria

* **Database Design**: Quality of the database schema design.
* **API Design**: Proper implementation of RESTful principles, clear and concise endpoints, proper endpoint authorisation checks in place and proper error handling logic in place.
* **Backend Logic**: Implementation of business logic, data validation, and security practices.
* **Code Quality**: Readability, maintainability, and use of best practices.
* **Documentation**: Clarity and completeness of the provided documentation.

## Submission Deadline

Please submit your completed assignment within 1 week’s time upon receiving this coding assignment document.

**Contact**

If you have any questions or need further clarification, please contact Huijie at [wu\_huijie@tech.gov.sg](mailto:wu_huijie@tech.gov.sg).