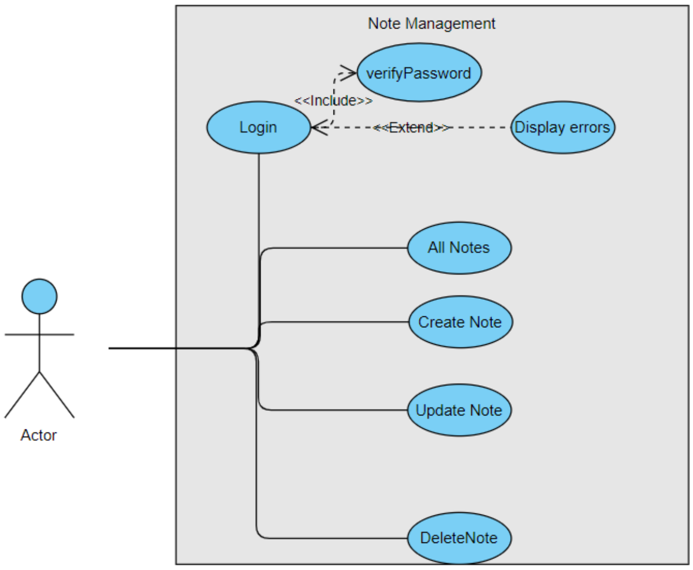
API Documentation

Name : Satender Kumar

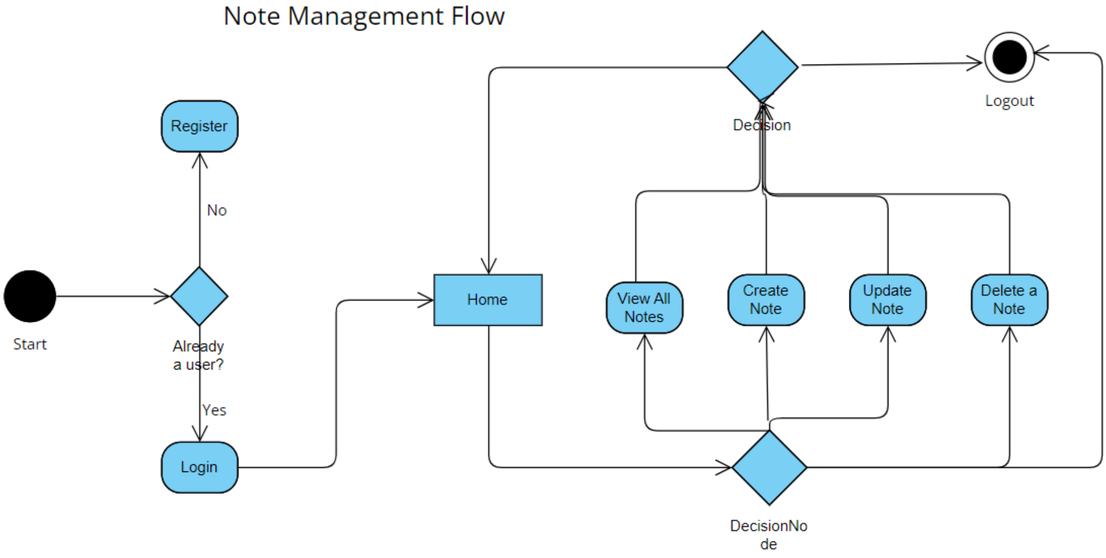
E-mail: satenderk8700@gmail.com

GitHub Backend : https://github.com/ProgrammingWithSid/NotesManagement-backend/

- Use Case Diagram



- Activity Diagram



Authentication

Here’s how authentication works

1. User’s Registration

- Endpoint : POST /user/register
- Description : Allows user to register their account
- Request Body:
  - first_name (string, required) : User’s first name
  - last_name (string, required) : User’s last name
  - email (string, required) : User’s E-mail
  - password (string, required) : User’s password

```
{
  "first_name":"satender",
  "last_name":"kumar",
  "email":"satender@gmail.com",
  "password":"12345678"
}
```

- Response:
- Status: 201 Created

```
{
  "success": true,
  "message": "User Created Successfully"
}
```

2. User’s Login

- Endpoint : POST /user/register
- Description : Allows user to Login their account
- Request Body:
  - email (string, required) : User’s E-mail
  - password (string, required) : User’s password

```
{
  "email":"sid@gmail.com",
  "password":"12345678"
}
```

- Response Body:
  - Status: 200 OK

```
{
  "success": true,
  "message": "Login Successful",
  "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NWFmOWIzYTJhNTlkMmE4YTRmMWMyNmUiLCJpYXQiOjE3MDYwMDg2MDAsImV4cCI6MTcwNjA5NTAwMH0.bVR4WFXDbtrd0OXwm7cbLafaeF1ig_1gN1cUiI7IZyE"
}
```

Notes Endpoint

1. View All Notes
   - Endpoint : GET /notes/getAllNotes
   - Description : Allows user to View All their notes
   - Response Body:
     - Status: 200 OK

```
{
  "success": true,
  "notes": [
    {
      "_id": "65af9b542a59d2a8a4f1c276",
      "user": "65af9b3a2a59d2a8a4f1c26e",
      "title": "Second User Note",
      "description": "This is description",
      "isRead": false,
      "__v": 0
    }
  ]
}
```

2. Endpoint : POST /notes/createNote
   - Description : Allows user to create a new note
   - Request Body:
     - title (string,required) : User’s title
     - description (string,required) : User’s description

```
{
  "title": "This is a test note",
  "description":"This is description"
}
```

   - Response Body :
   - Status: 201 Created

```
{
  "success": true,
  "message": "Note Created Successfully",
  "newNotes": {
    "user": "65af9b3a2a59d2a8a4f1c26e",
    "title": "This is a test note",
    "description": "This is description",
    "isRead": false,
    "_id": "65afa02a19c010d8db1698bd"
  }
}
```

3. Endpoint : PUT /notes/updateNote/:id

    - Description : Allows user to create a update a note
    - Request Body:

```
{
"title": "This is a test note",
"description":"This is description"
}
```

    - Response Body:
    - Status : 200 Ok

```
{
  "success": true,
  "message": "Note Updated Successfully",
  "updatedNote": {
    "_id": "65afa02a19c010d8db1698bd",
    "user": "65af9b3a2a59d2a8a4f1c26e",
    "title": "Update",
    "description": "Third note has changed",
    "isRead": false,
    "__v": 0
  }
}
```

4. Endpoint : DELETE /notes/deleteNote/:id

    - Description : Allows user to delete a note
    - Response Body:
    - Status: 200 OK

```
{
  "success": true,
  "message": "Note Deleted Successfully"
}
```

5. Endpoint : PUT /notes/toggleNote/:id

- Description : Allows user to change status of a note
    - Response Body :
    - Status: 200 OK

```
{
  "success": true,
  "message": "Updated Successfully",
  "updatedNote": {
    "_id": "65af95d05dbb529e87fde555",
    "user": "65aed78c5b807f14a97f7645",
    "title": "Second User Note",
    "description": "This is description",
    "isRead": true,
    "__v": 0
  }
}
```