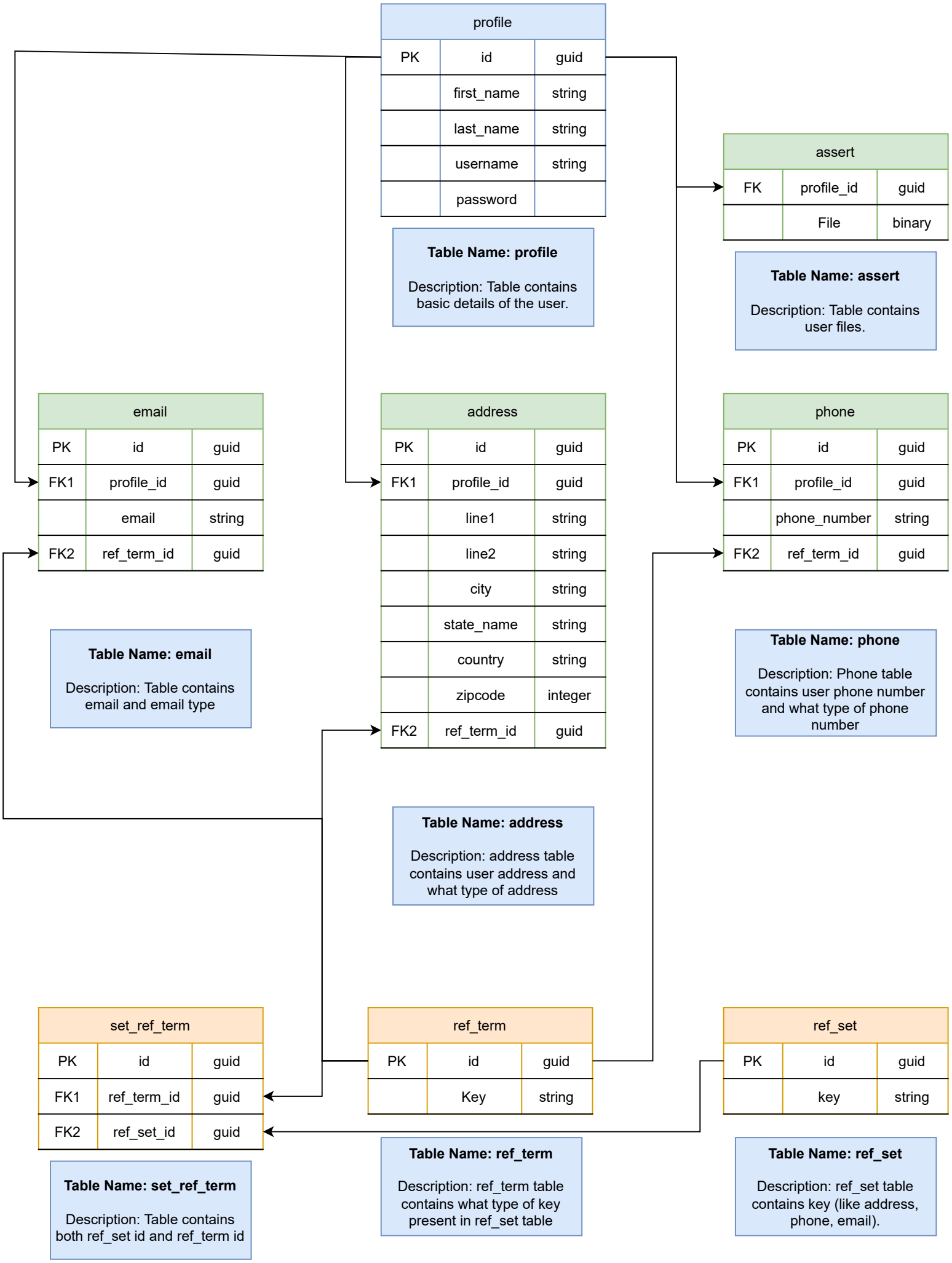
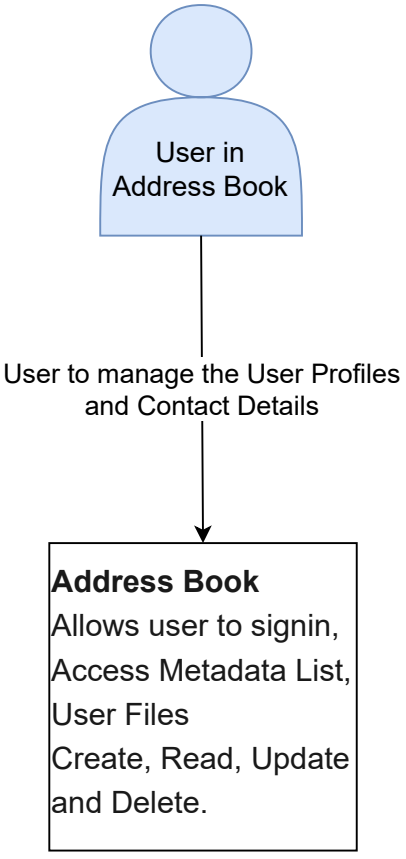
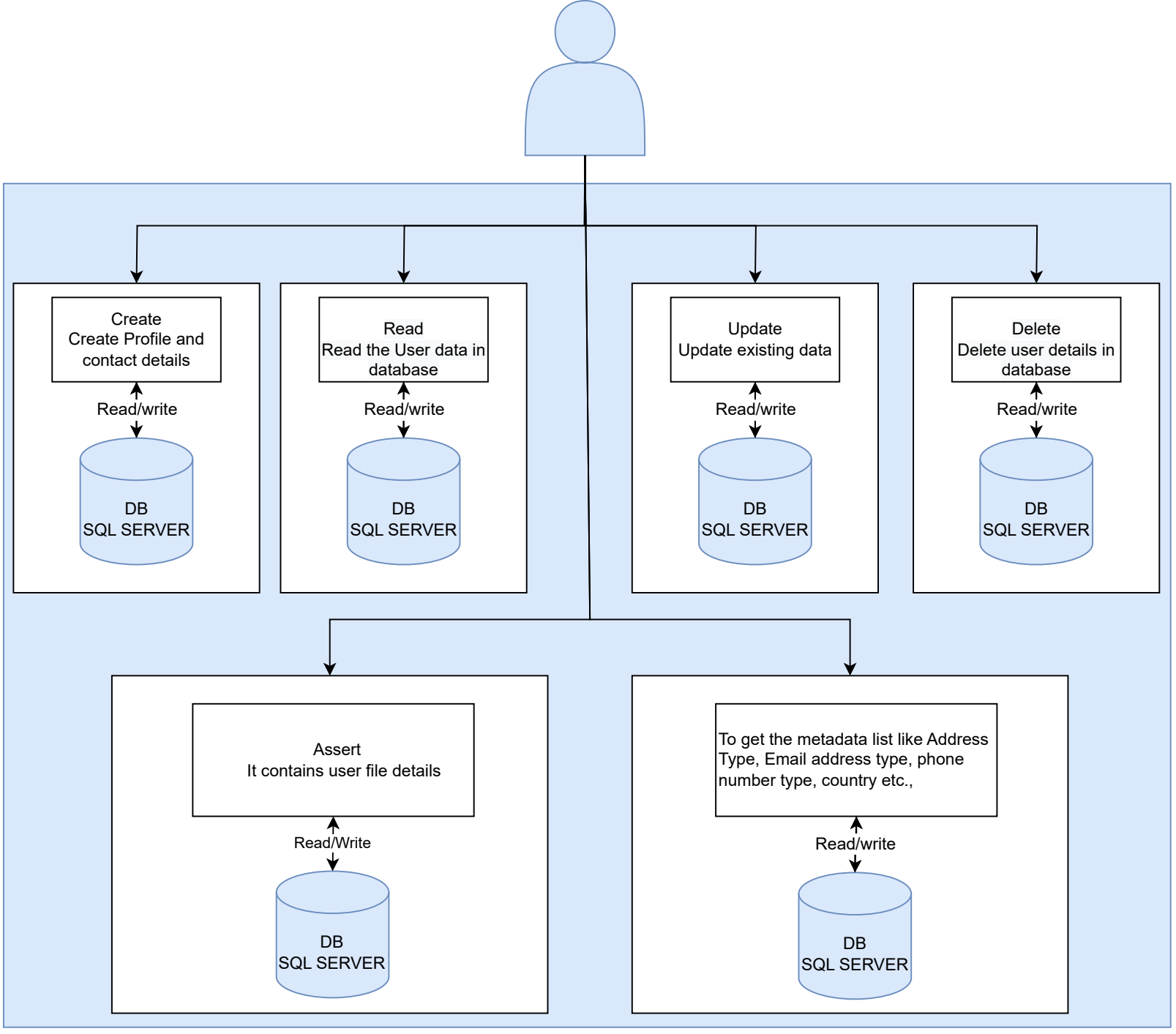
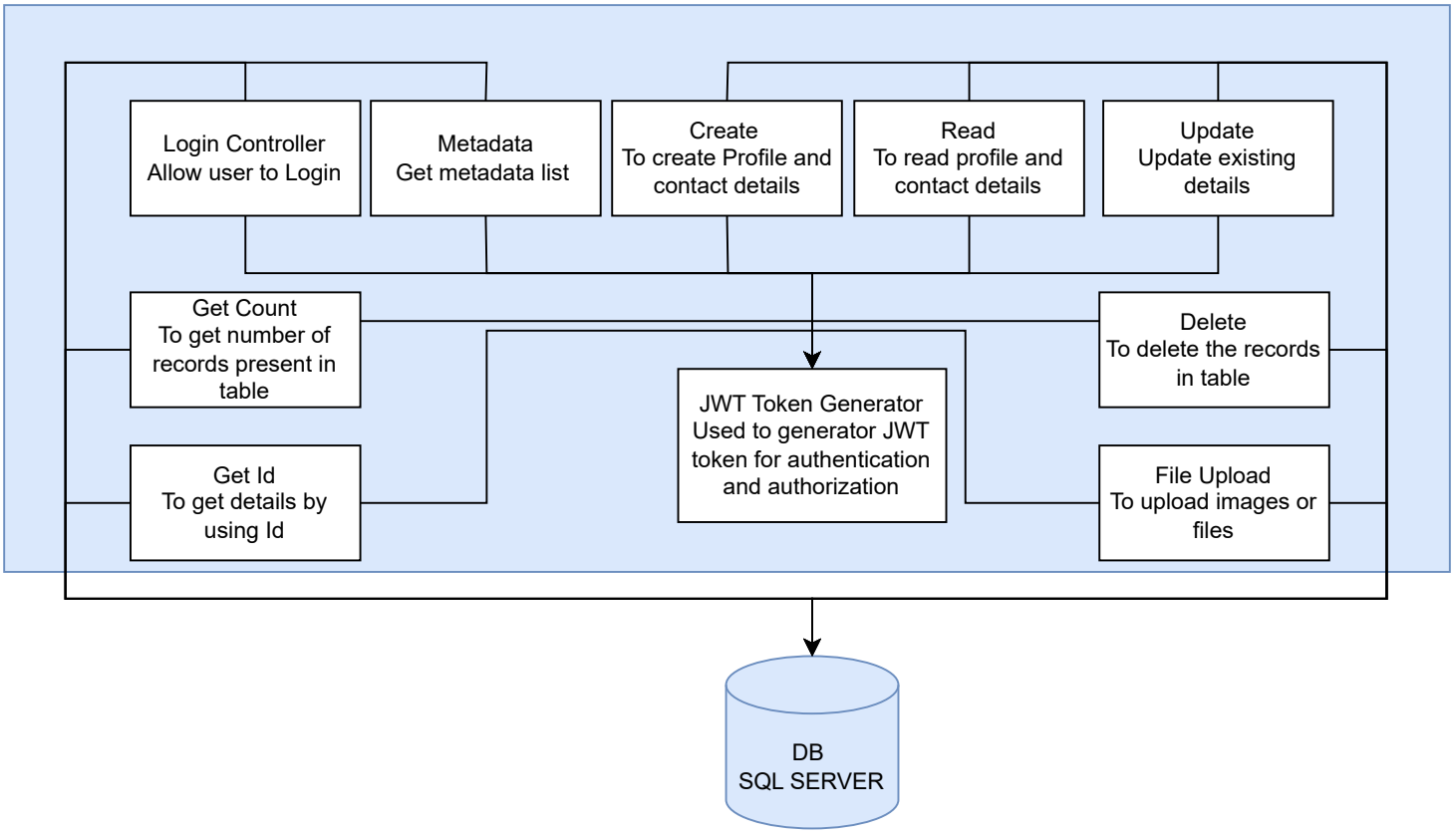

profile
PK id guid
first_name string
last_name string
username string
password
Table Name: profile
Description: Table contains basic details of the user.
assert
FK profile_id guid
File binary
Table Name: assert
Description: Table contains user files.
email
PK id guid
FK1 profile_id guid
email string
FK2 ref_term_id guid
Table Name: email
Description: Table contains email and email type
address
PK id guid
FK1 profile_id guid
line1 string
line2 string
city string
state_name string
country string
zipcode integer
FK2 ref_term_id guid
Table Name: address
Description: address table contains user address and what type of address
phone
PK id guid
FK1 profile_id guid
phone_number string
FK2 ref_term_id guid
Table Name: phone
Description: Phone table contains user phone number and what type of phone number
set_ref_term
PK id guid
FK1 ref_term_id guid
FK2 ref_set_id guid
Table Name: set_ref_term
Description: Table contains both ref_set id and ref_term id
ref_term
PK id guid
Key string
Table Name: ref_term
Description: ref_term table contains what type of key present in ref_set table
ref_set
PK id guid
key string
Table Name: ref_set
Description: ref_set table contains key (like address, phone, email).


User in
Address Book
User to manage the User Profiles
and Contact Details
Address Book
Allows user to signin,
Access Metadata List,
User Files
Create, Read, Update
and Delete.


Create
Create Profile and contact details
Read/write
DB
SQL SERVER
Read
Read the User data in database
Read/write
DB
SQL SERVER
Update
Update existing data
Read/write
DB
SQL SERVER
Delete
Delete user details in database
Read/write
DB
SQL SERVER
Assert
It contains user file details
Read/Write
DB
SQL SERVER
To get the metadata list like Address Type, Email address type, phone number type, country etc.,
Read/write
DB
SQL SERVER

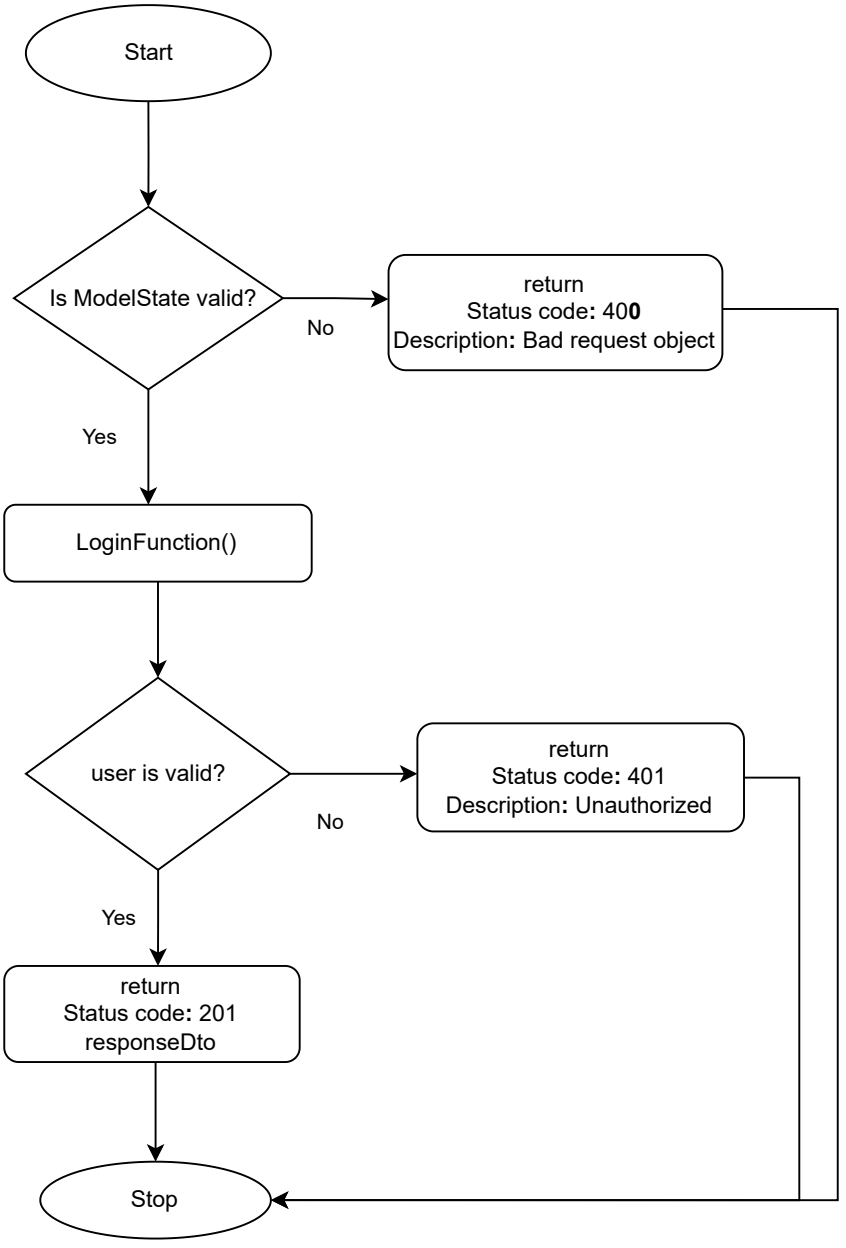

Login Controller
Allow user to Login
Metadata
Get metadata list
Create
To create Profile and contact details
Read
To read profile and contact details
Update
Update existing details
Get Count
To get number of records present in table
Get Id
To get details by using Id
JWT Token Generator
Used to generator JWT token for authentication and authorization
Delete
To delete the records in table
File Upload
To upload images or files
DB
SQL SERVER

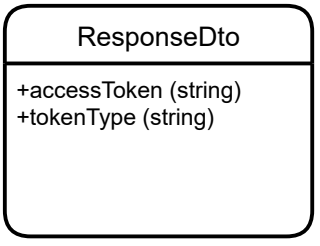
Login API

INPUT
+String username
+String password

URL:
/api/auth/signin

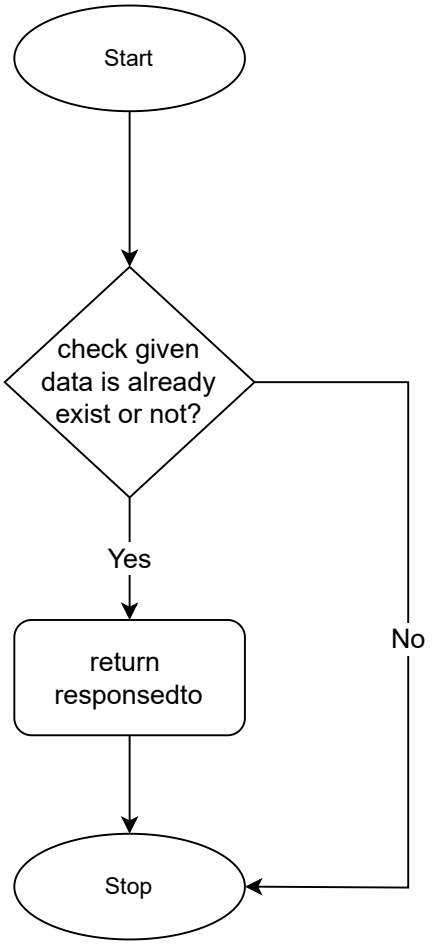
Method :
Post





LoginFunction Method

INPUT
+String username
+String password



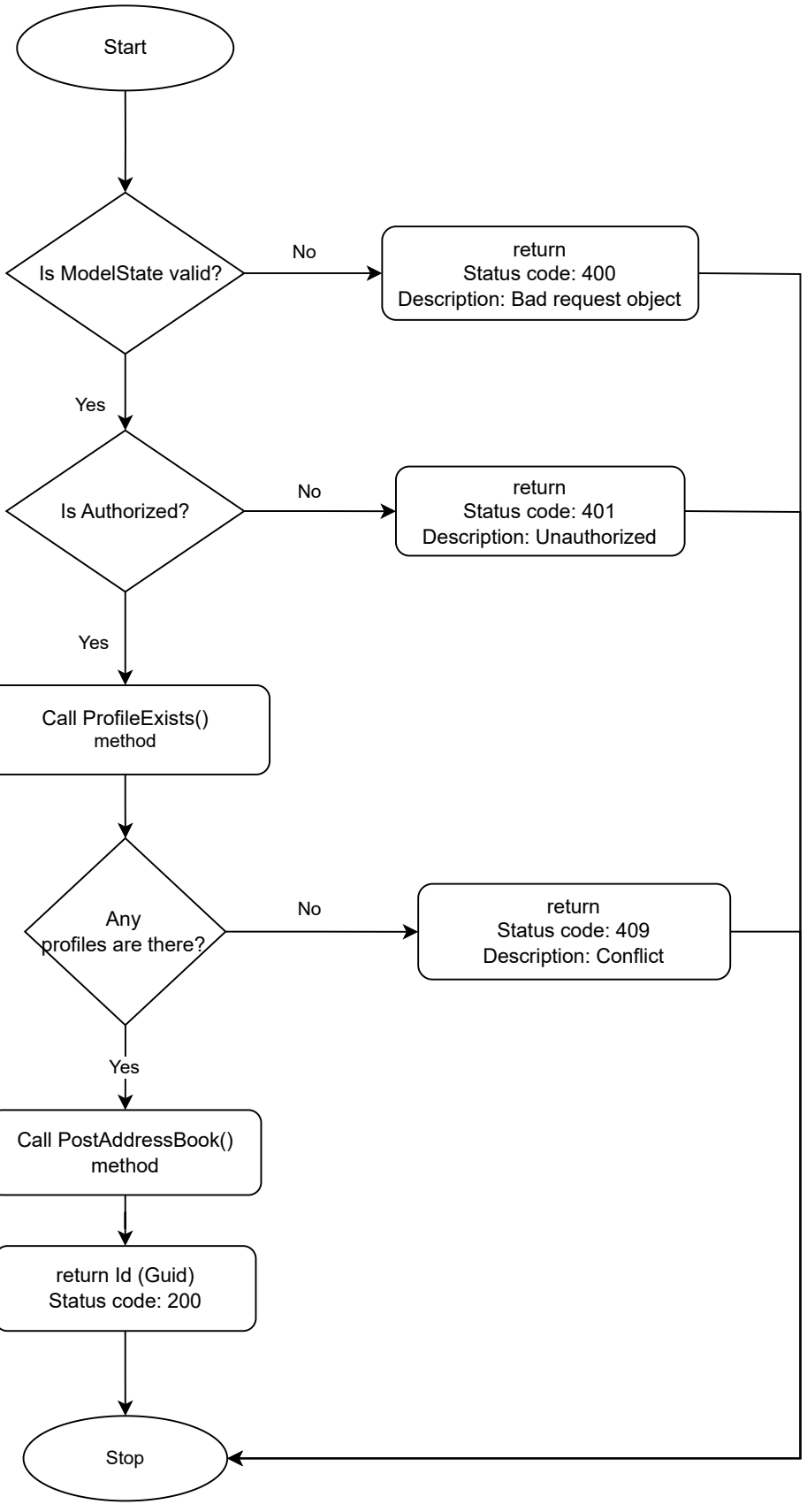
ResponseDto
+accessToken (string) +tokenType (string)

Create Address Book API

INPUT
+String firstname
+String lastname
+List <Email> body
+List<Address>body
+List<phone>body
+String AssestDto

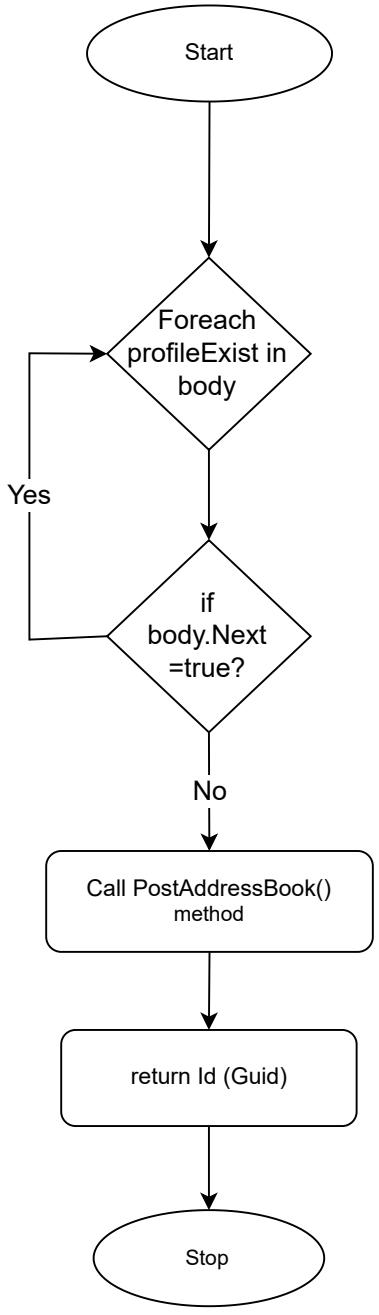
URL
/api/account

Method :
Post



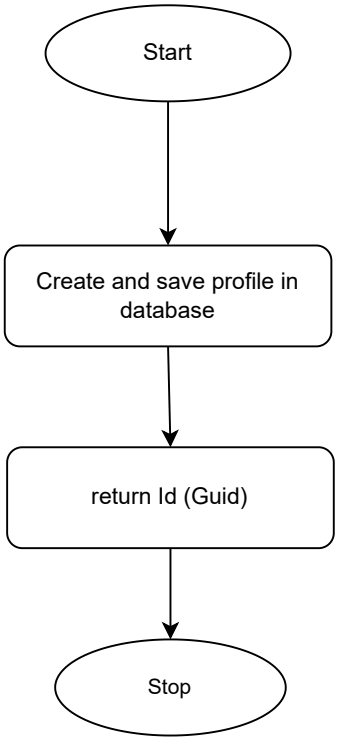
ProfileExists method

INPUT
+String firstname
+String lastname
+List <Email> body
+List<Address>body
+List<phone>body
+String AssestDto



PostAddressBook method

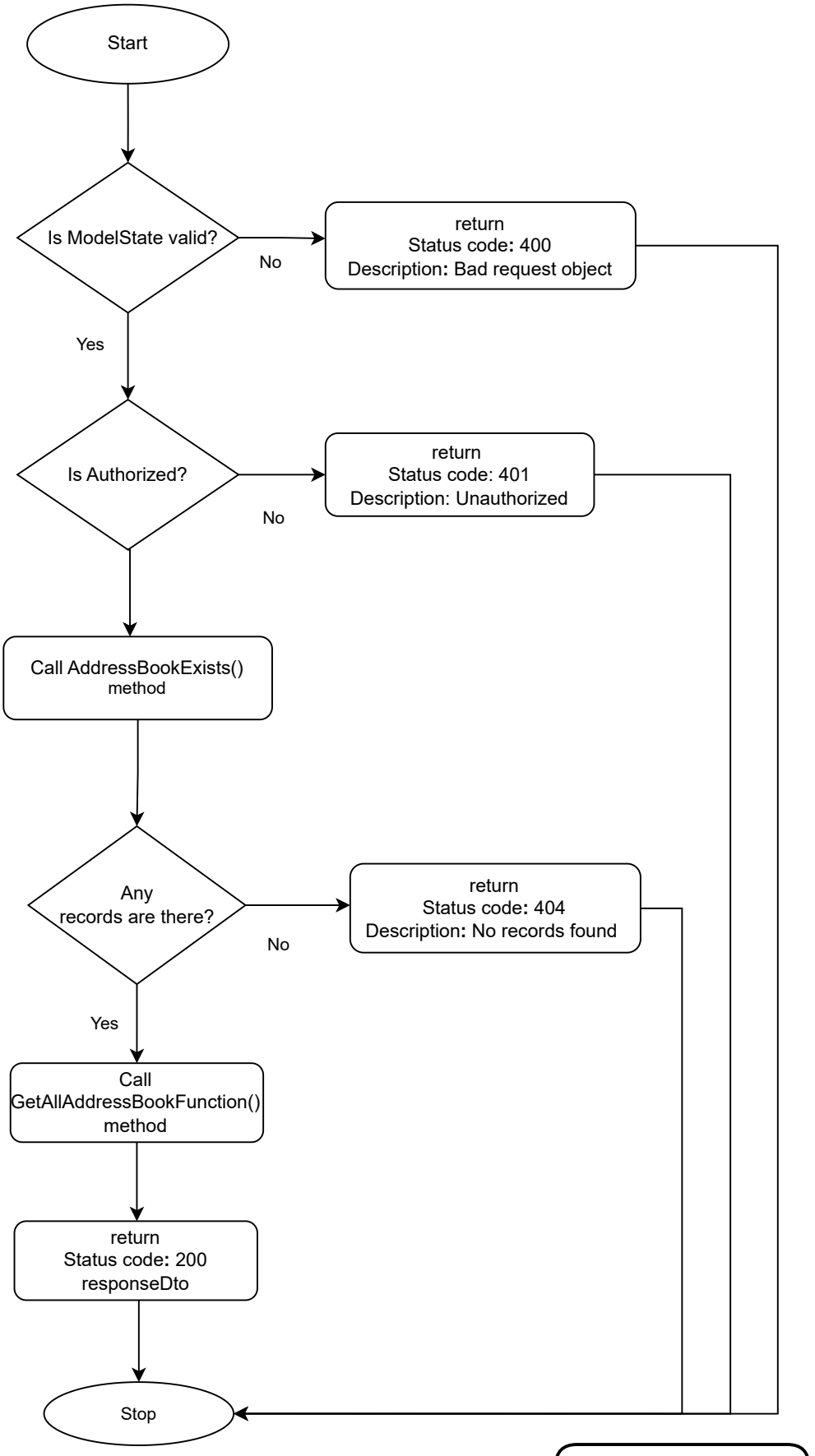
INPUT
+String firstname
+String lastname
+List <Email> body
+List<Address>body
+List<phone>body
+String AssestDto



Get All Address Book API

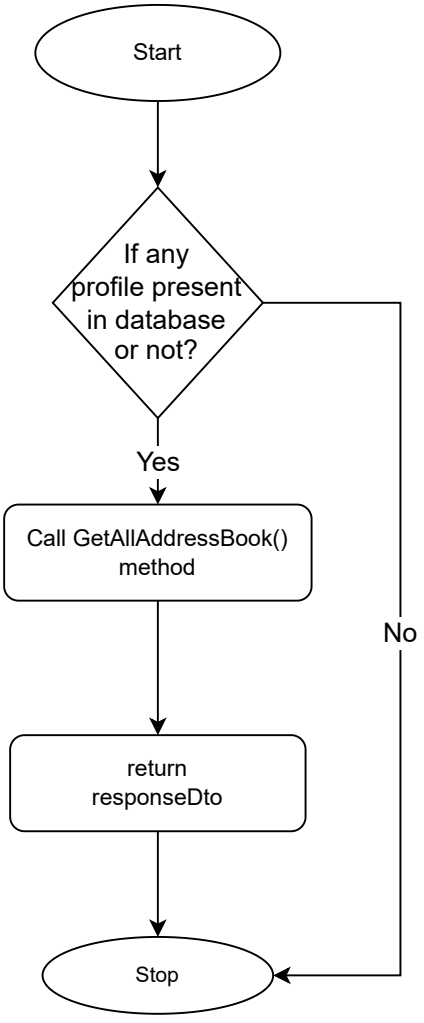
URL
/api/account

Method :
Get



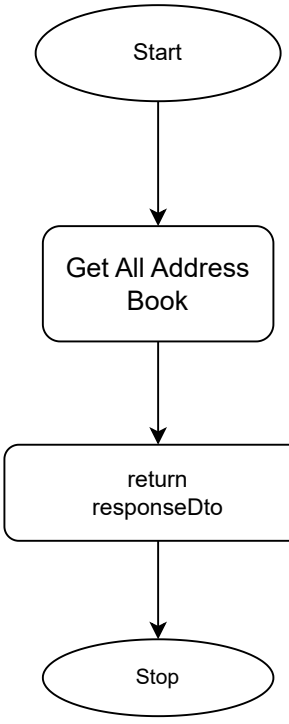
ResponseDto
+String firstname +String lastname +List <Email> body +List<Address>body +List<phone>body +String AssestDto

AddressBookExists



ResponseDto
+String firstname +String lastname +List <Email> body +List<Address>body +List<phone>body +String AssestDto

GetAllAddressBook

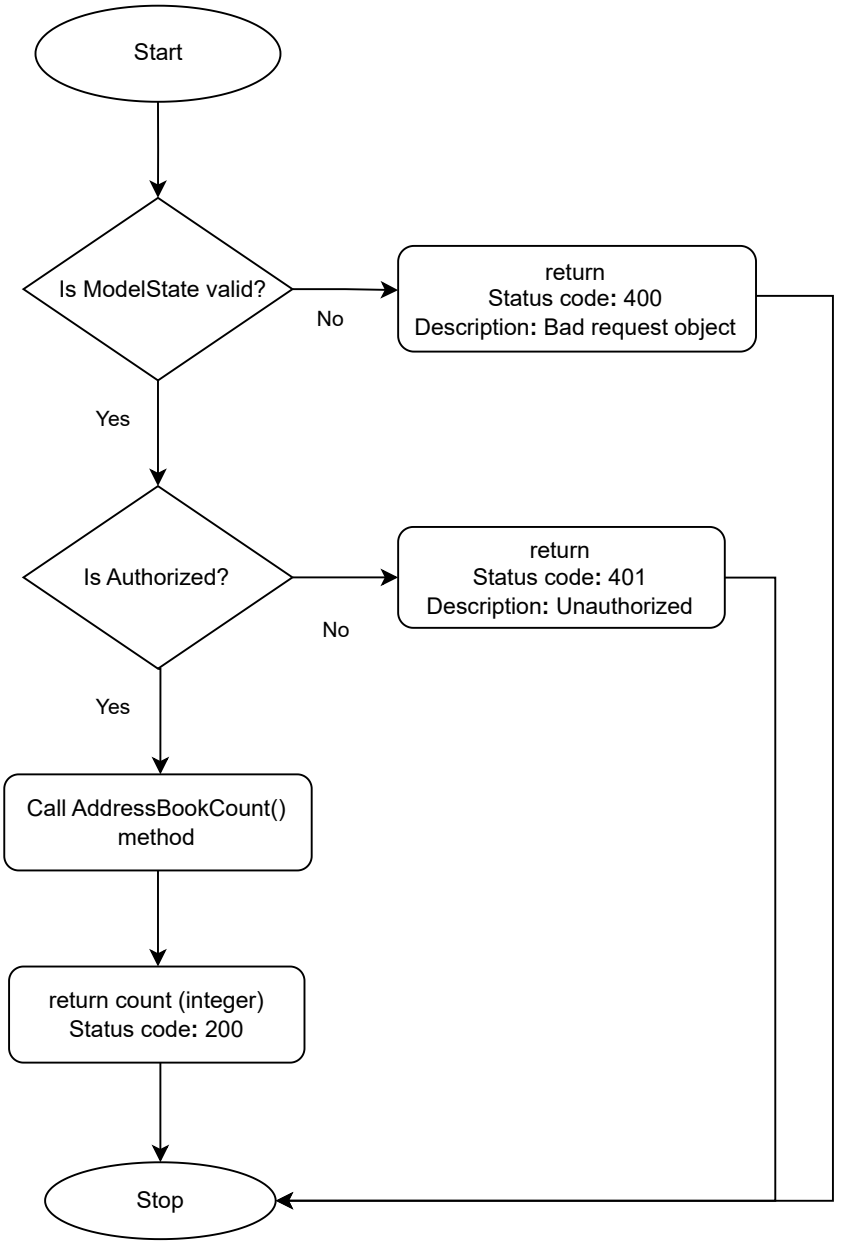


ResponseDto
+String firstname +String lastname +List <Email> body +List<Address>body +List<phone>body +String AssestDto

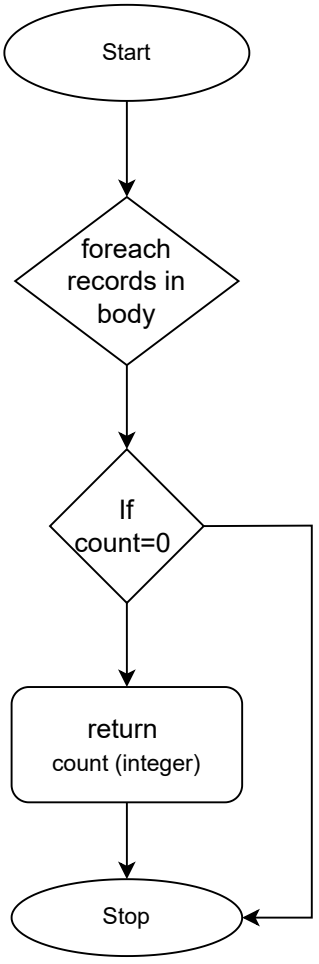
Get Count Address Book API

URL
/api/account/count

Method :
Get



AddressBookCount

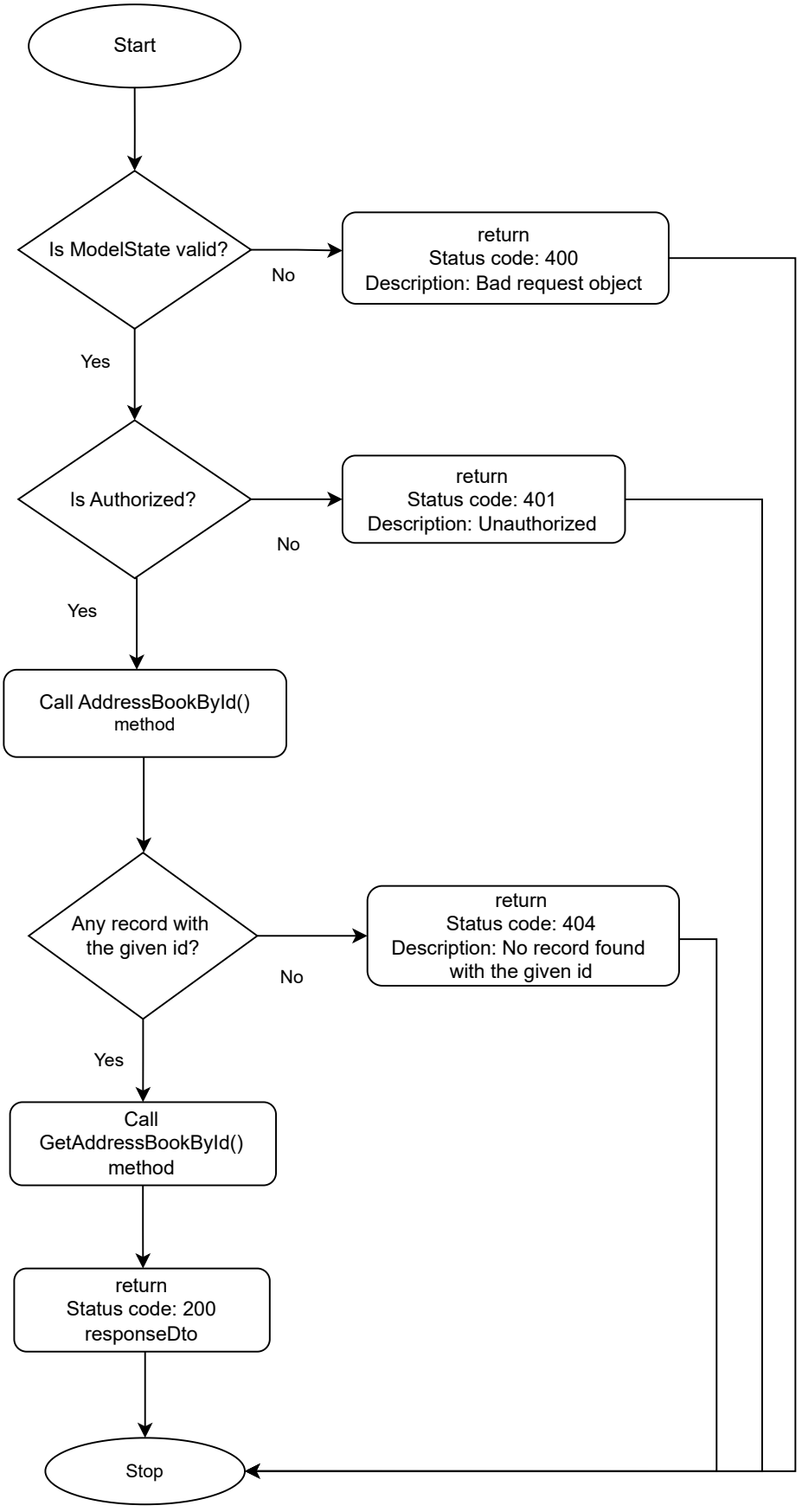


Get Address Book by Id API

Input:
+id (integer)

URL
/api/account/{id}

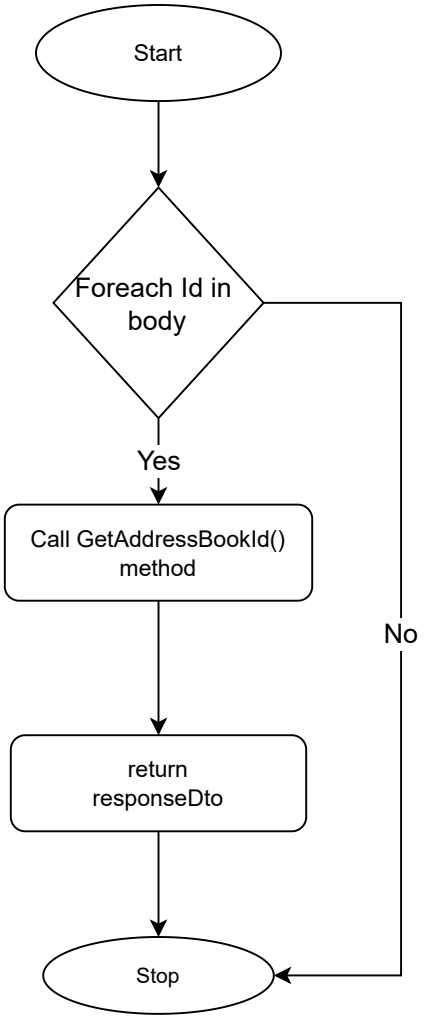
Method :
Get



ResponseDto
+String firstname +String lastname +List <Email> body +List<Address>body +List<phone>body +String AssestDto

AddressBookId method

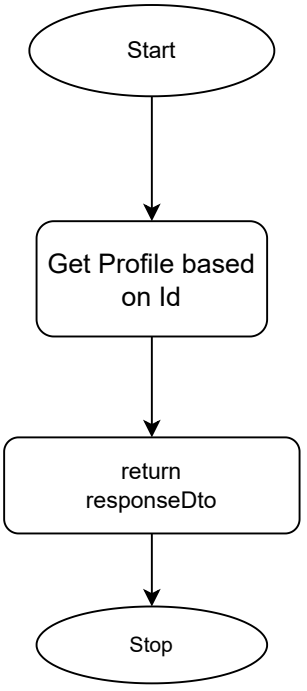
Input:
+id (integer)



ResponseDto
+String firstname +String lastname +List <Email> body +List<Address>body +List<phone>body +String AssestDto

Get AddressBookId method

Input:
+id (integer)



ResponseDto
+String firstname +String lastname +List <Email> body +List<Address>body +List<phone>body +String AssestDto

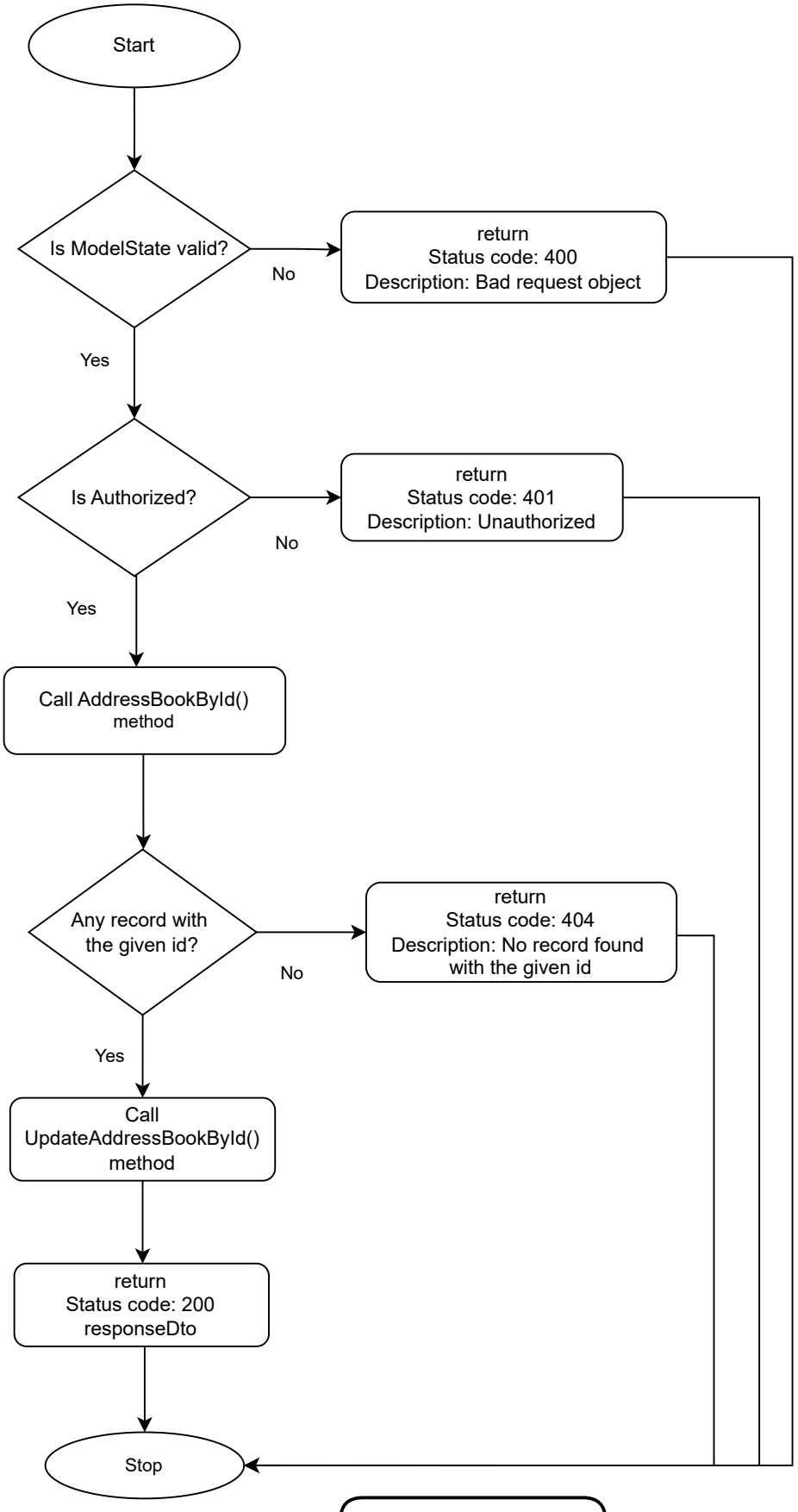
Update Address Book API

INPUT
+String firstname
+String lastname
+List <Email> body
+List<Address>body
+List<phone>body
+String AssestDto

URL
/api/account/{id}

Method :
Put

Input:
+id (integer)



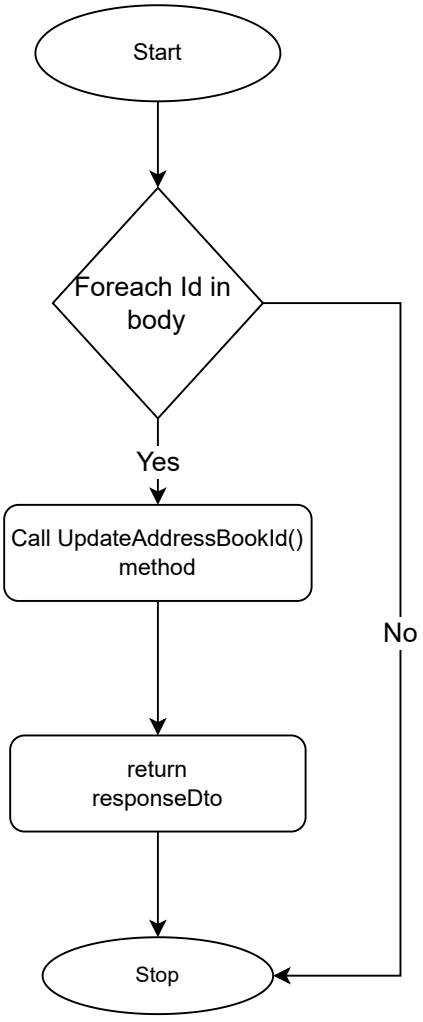
ResponseDto

+String firstname
+String lastname
+List <Email> body
+List<Address>body
+List<phone>body
+String AssestDto

AddressBookId method

Input:
+id (integer)

INPUT
+String firstname
+String lastname
+List <Email> body
+List<Address>body
+List<phone>body
+String AssestDto

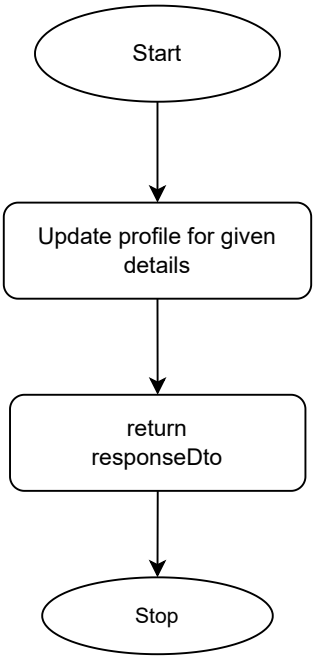


ResponseDto
+String firstname +String lastname +List <Email> body +List<Address>body +List<phone>body +String AssestDto

UpdateAddressBookId method

Input:
+id (integer)

INPUT
+String firstname
+String lastname
+List <Email> body
+List<Address>body
+List<phone>body
+String AssestDto



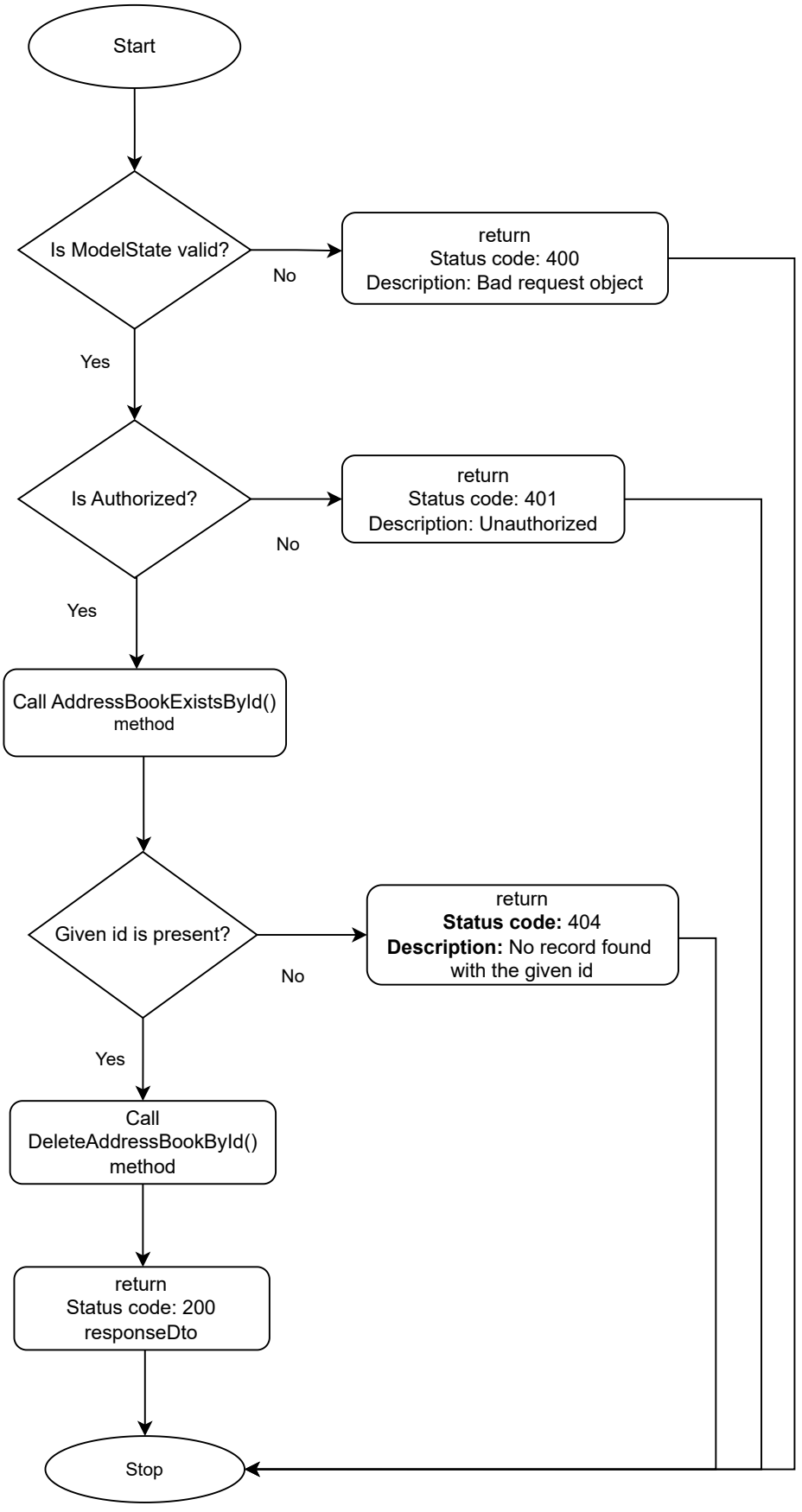
ResponseDto
+String firstname +String lastname +List <Email> body +List<Address>body +List<phone>body +String AssestDto

Delete Address Book API

Input:
+id (integer)

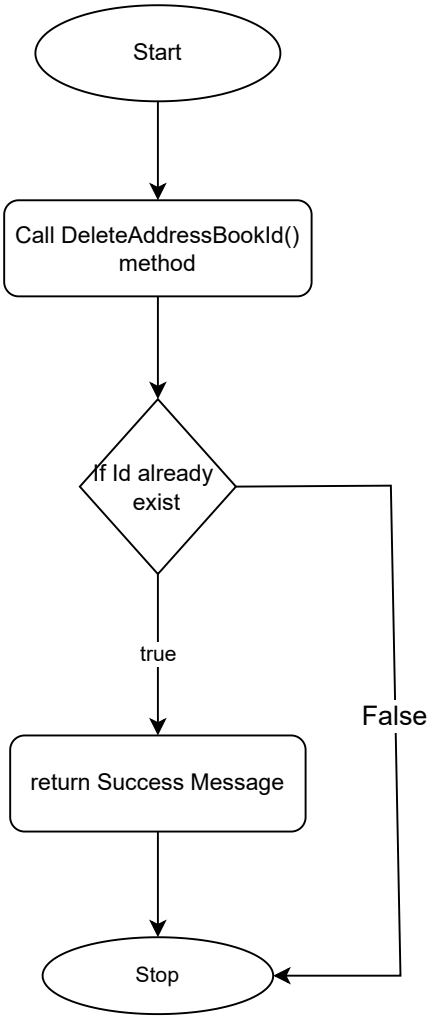
URL
/api/acoount.{id}

Method :
Delete



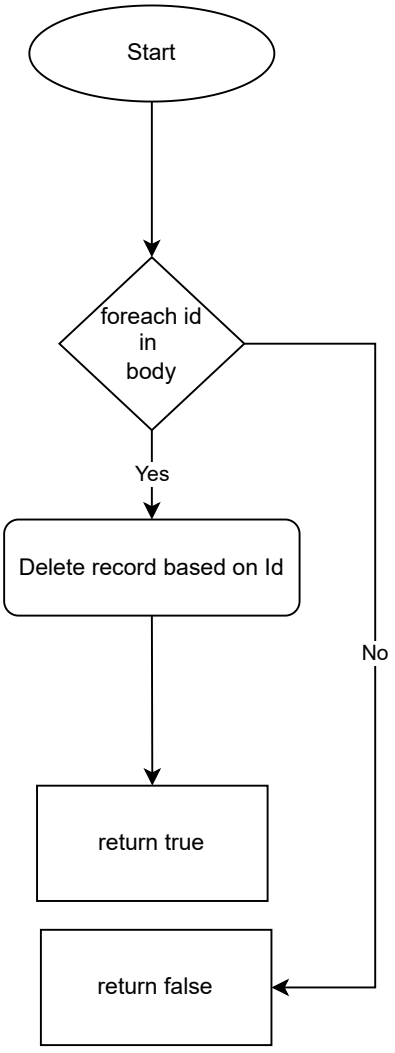
AddressBookExistById Method

Input:
+id (integer)



DeleteAddressBookId Method

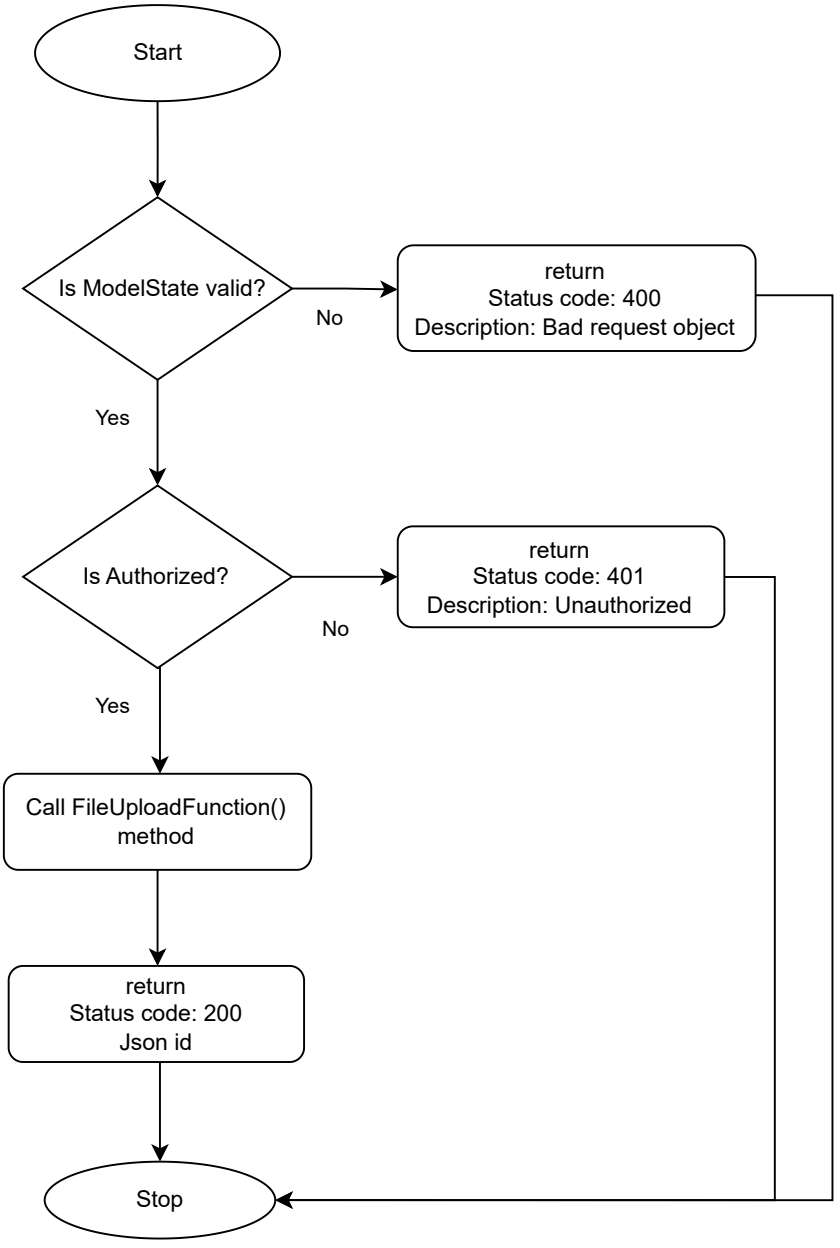
Input:
+id (integer)



File Upload API

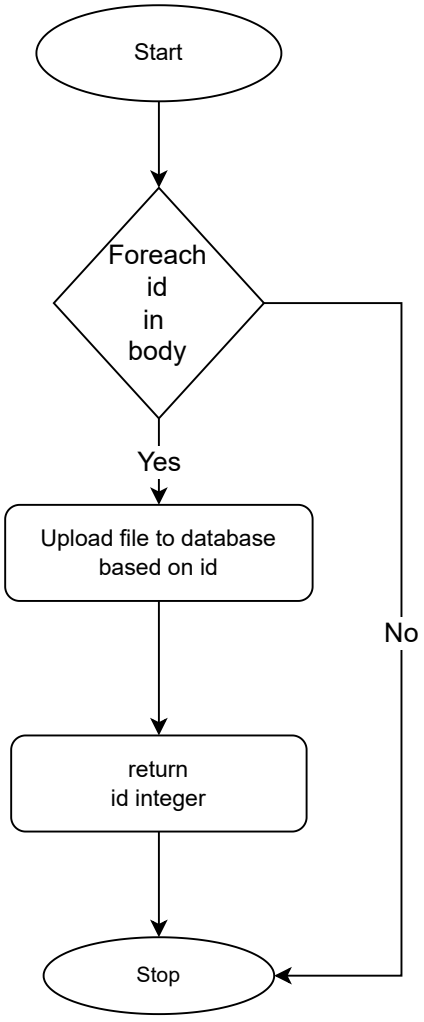
URL
/api/assert/uploadFile

Method :
Post



AddressBookExistById Method

Input:
+id (integer)

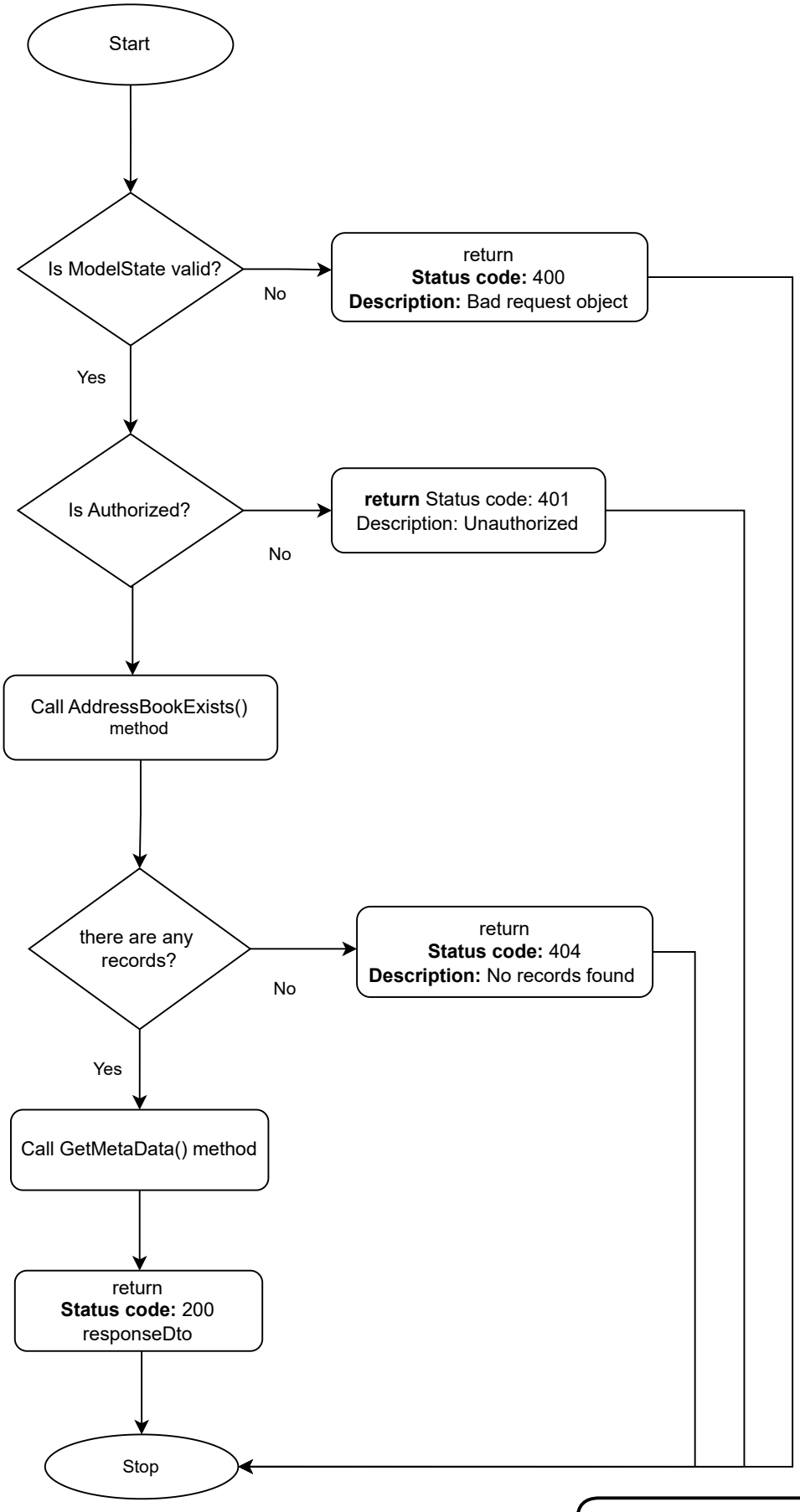


Meta data API

input
+string key

URL
/api/meta_data/ref_set/{key}

Method
get

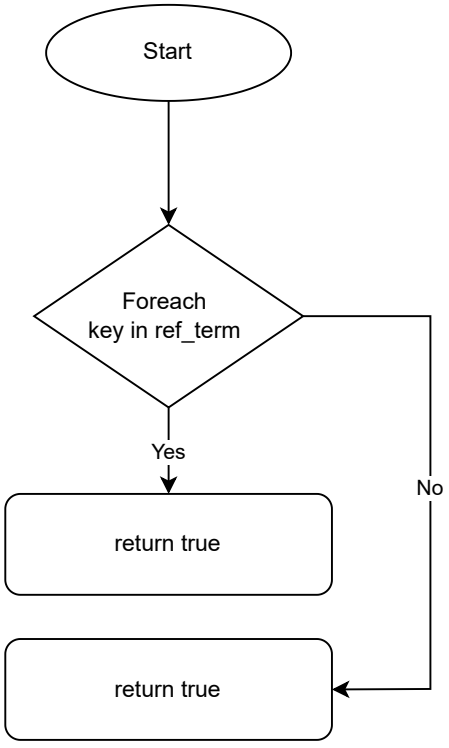


ResponseDto

+string id
+ string key
+ description

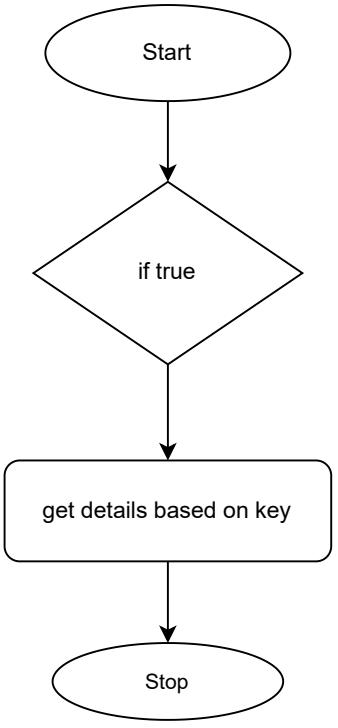
AddressBookExist

input
+string key

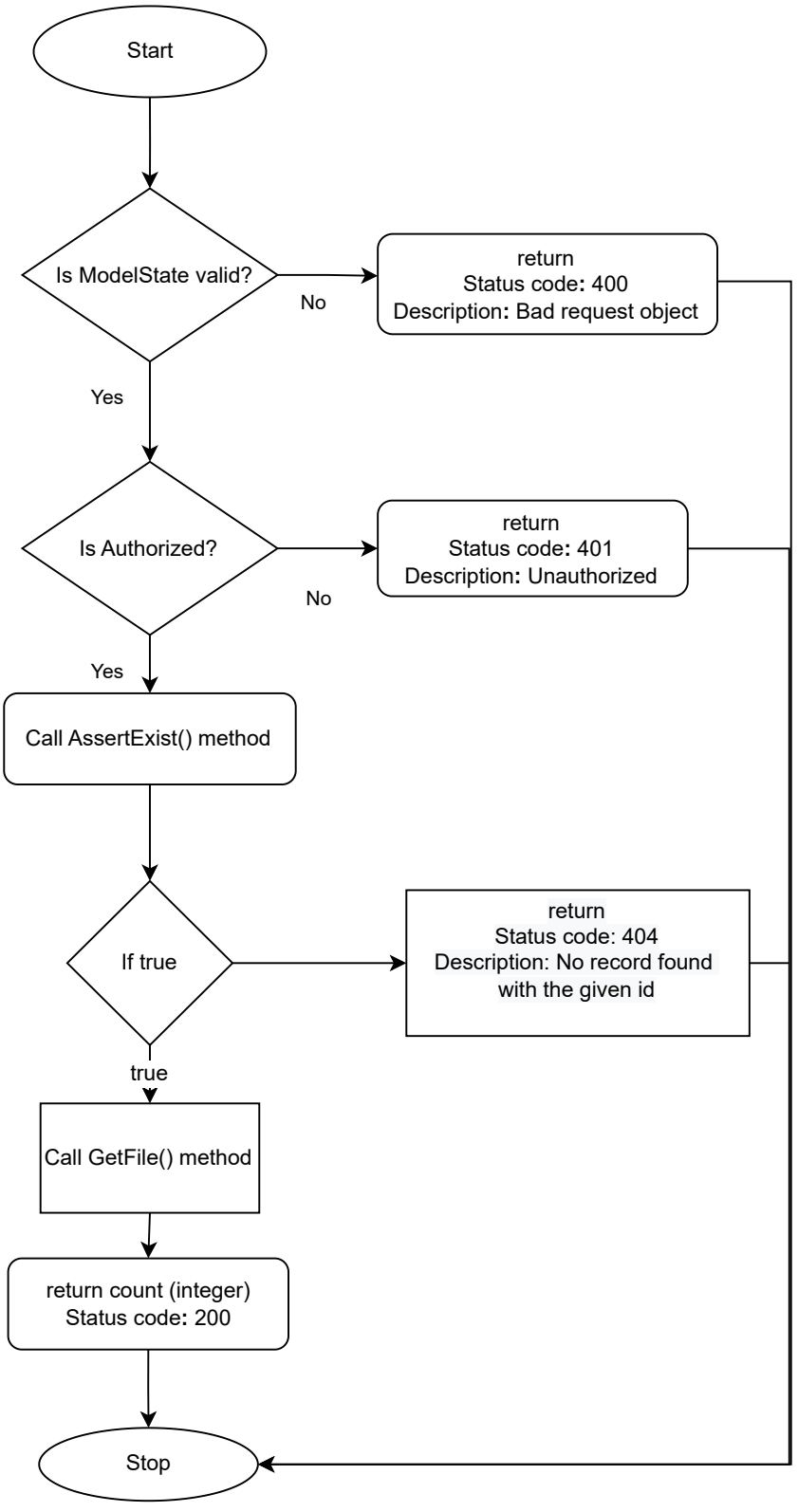


GetMetaData

input
+string key



Download API

URL:
/api/asset/{asset-id}

method
get