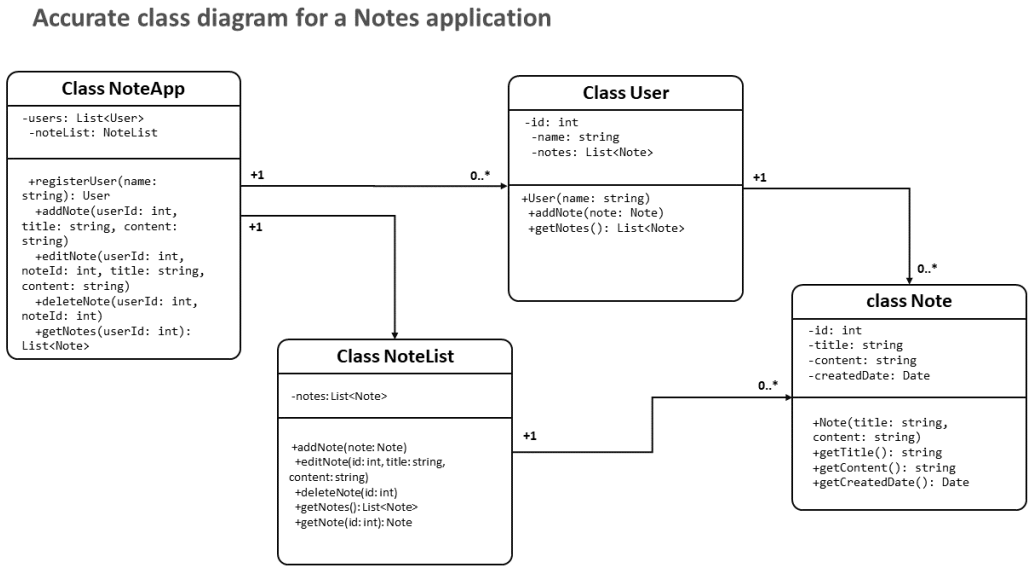

Below, I implemented the "NotesApp" class with python in the UML above...

In [ ]:
```python
class NoteApp:

  def __init__(self):
    self.users = []
    self.noteList = NoteList()

  def registerUser(self, name):
    user = User(name)
    self.users.append(user)
    user.id = len(self.users)
    return user

  def addNote(self, userId, title, content):
    user = self.getUser(userId)
    note = Note(title, content)
    self.noteList.addNote(note)
    user.addNote(note)

  def editNote(self, userId, noteId, title, content):
    self.noteList.editNote(noteId, title, content)

  def deleteNote(self, userId, noteId):
    self.noteList.deleteNote(noteId)

  def getNotes(self, userId):
    user = self.getUser(userId)
    return user.getNotes()

  def getUser(self, userId):
    return self.users[userId-1]
```

this is the full implementation of all the classes

In [ ]:
```python
from datetime import date

class Note:

  def __init__(self, title, content):
    self.id = None
    self.title = title
    self.content = content
    self.createdDate = date.today()

  def getTitle(self):
    return self.title

  def getContent(self):
    return self.content

  def getCreatedDate(self):
    return self.createdDate

class NoteList:

  def __init__(self):
    self.notes = []

  def addNote(self, note):
    self.notes.append(note)
    note.id = len(self.notes)

  def editNote(self, id, title, content):
    note = self.getNote(id)
    note.title = title
    note.content = content

  def deleteNote(self, id):
    self.notes.pop(id-1)

  def getNotes(self):
    return self.notes

  def getNote(self, id):
    return self.notes[id-1]

class User:

  def __init__(self, name):
    self.id = None
    self.name = name
    self.notes = []

  def addNote(self, note):
    self.notes.append(note)

  def getNotes(self):
    return self.notes

class NoteApp:
```

```
  def __init__(self):
    self.users = []
    self.noteList = NoteList()

  def registerUser(self, name):
    user = User(name)
    self.users.append(user)
    user.id = len(self.users)
    return user

  def addNote(self, userId, title, content):
    user = self.getUser(userId)
    note = Note(title, content)
    self.noteList.addNote(note)
    user.addNote(note)

  def editNote(self, userId, noteId, title, content):
    self.noteList.editNote(noteId, title, content)

  def deleteNote(self, userId, noteId):
    self.noteList.deleteNote(noteId)

  def getNotes(self, userId):
    user = self.getUser(userId)
    return user.getNotes()

  def getUser(self, userId):
    return self.users[userId-1]
```