DocNo: 001.C.4:1

X-Note
System Design Model
Version 1.0

By:
X-Note Developers
2019-03

Group Member:
Jingyu Li
Du Liu
Yu Fan
Qiuxuan Ling
Shixuan Gu

Document Language:
English

Revision History

Date	Version	Description	Author
04/20/2019	1.0	Create and integrate	Jingyu Li

Contents

1. Introduction

1.1 Purpose

This document aims to list all the design models for the App X-Note, including use case diagrams, logic diagrams, realization diagrams, runtime diagrams and deployment diagrams. These design models are established on the requirement specification and system analysis, as a detailed illustration of the structure of the system. The following development of the system will proceed based on this document.

1.2 Background

(1) App name: X-Note
(2) Developers and users:
Developers: Jingyu Li, Du Liu, Shixuan Gu, Qiuxuan Lin, Yu Fan
Users: learners, especially students

1.3 Definition

All terms in this document are defined in Glossary.

1.4 Reference

(1) Object-Oriented Software Engineering: Using UML, Patterns, and Java (Third Edition). Pearson Education, Inc., 2010.
(2) 面向对象软件工程实践指南. 上海交通大学出版社, 2016.

1.5 Overview

This document involves use case diagrams, logic diagrams, realization diagrams, runtime diagrams and deployment diagrams. Use case diagrams shows the use case of the system. Logic diagrams mainly consists of the system architecture diagram, design class diagrams and realization of use cases. Realization diagrams construct component diagrams for every sub-system. Runtime diagrams show the process and threads of the system with class diagrams and component diagrams. Deployment diagrams show the deployment of the system into hardware and software. Those different models works together to illustrate the structure of the system.

2. Use Case View

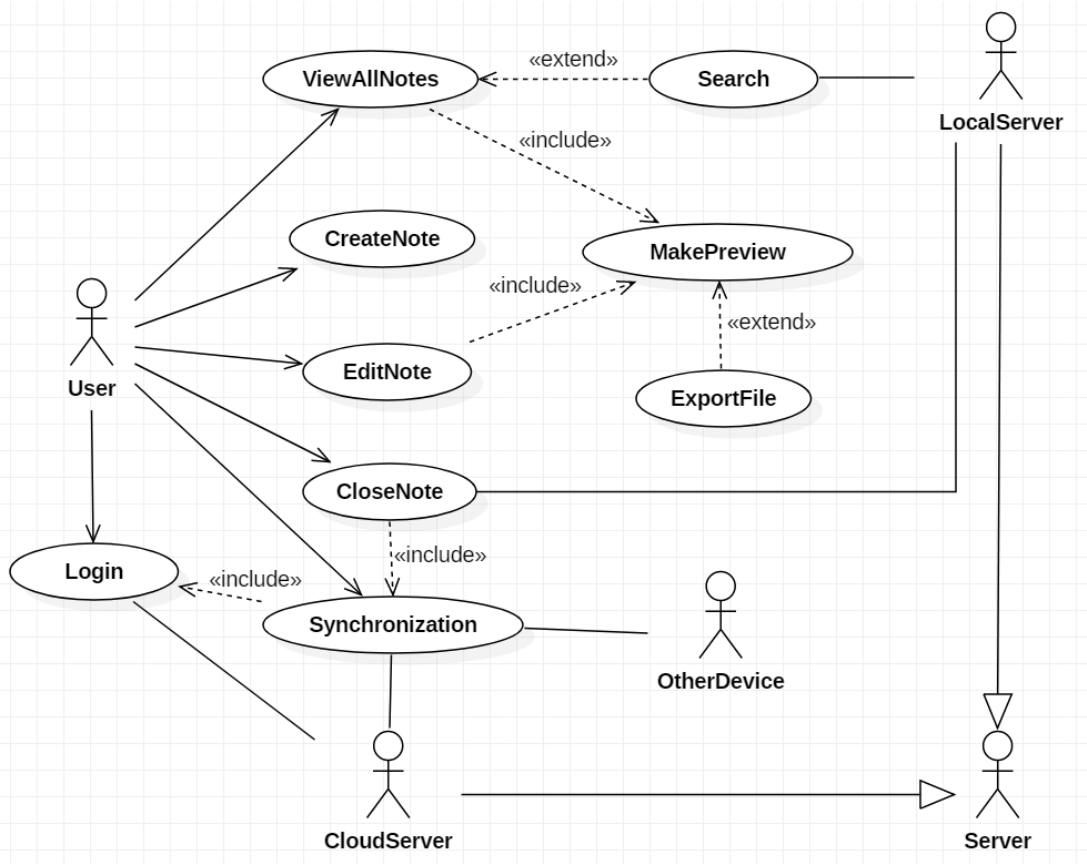

Figure 1: Use case diagram

3. Logic View

3.1 System structure

We divide the whole system into six sub-systems, which are UI, Note Management, User Management, Preview Management, Cloud Service and Local Service.

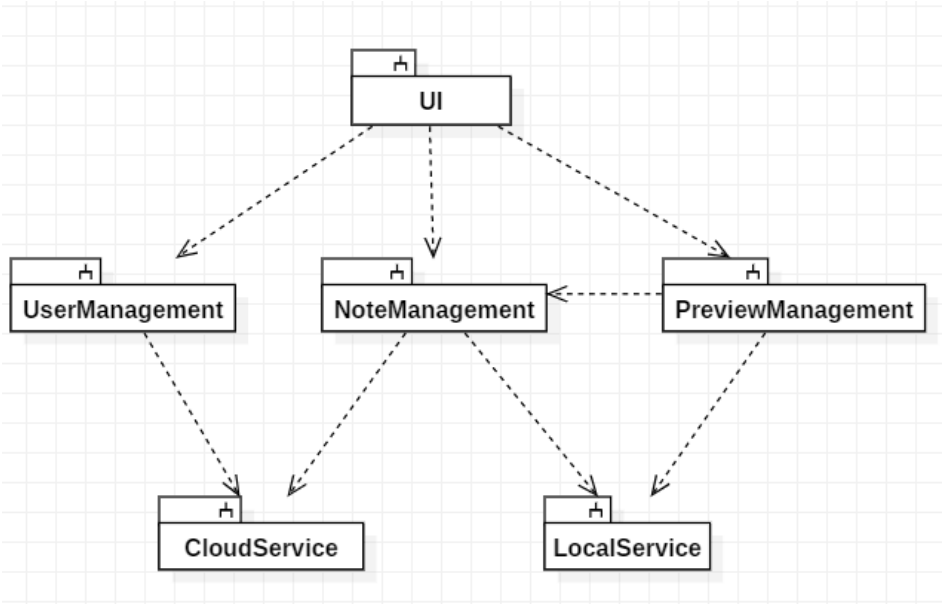

Figure 2: System structure

Class diagrams for each subsystem is listed as following:

(1) UI

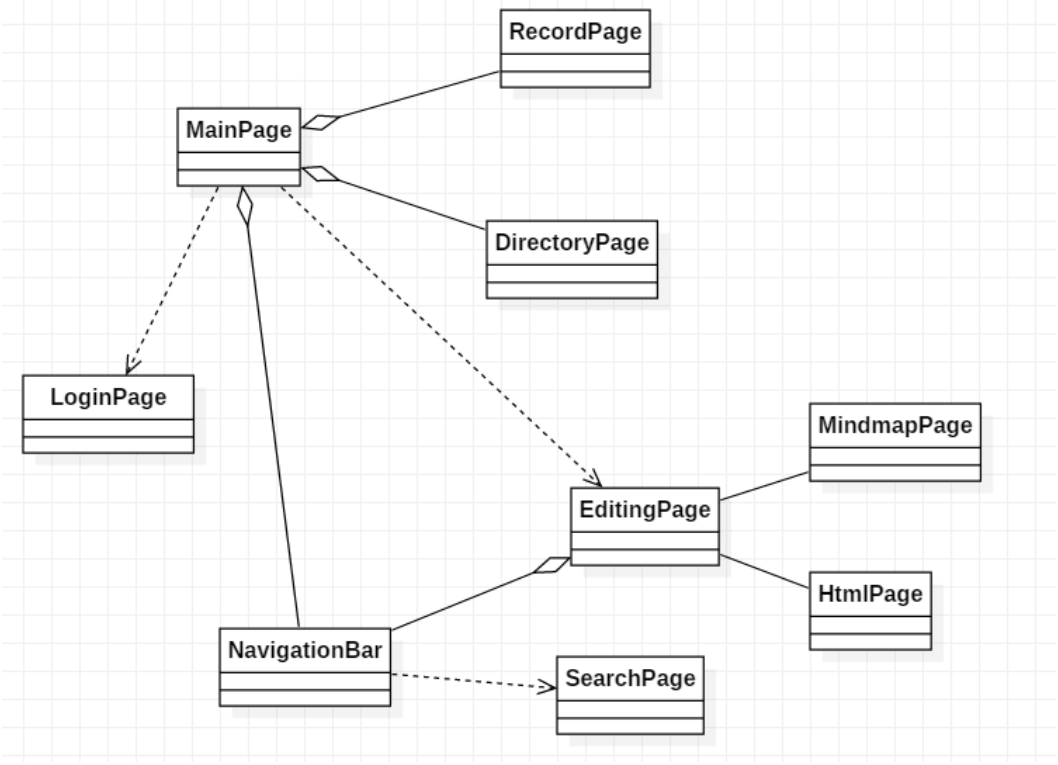

Figure 3: Subsystem class diagram of UI

(2) User Management

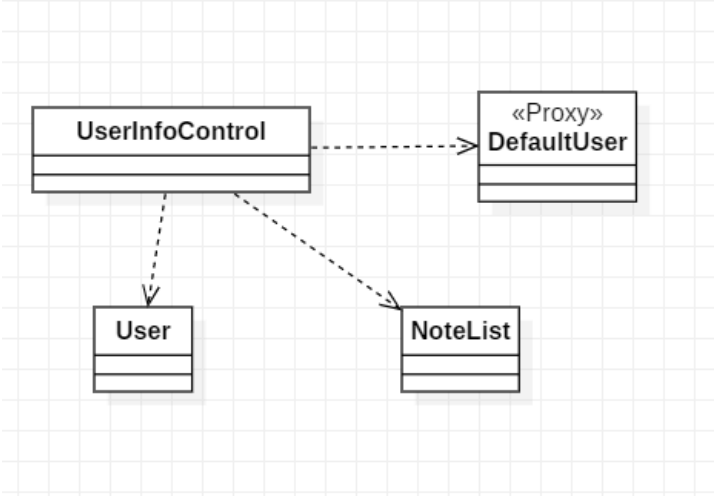

Figure 4: Subsystem class diagram of user management

(3) Note Management

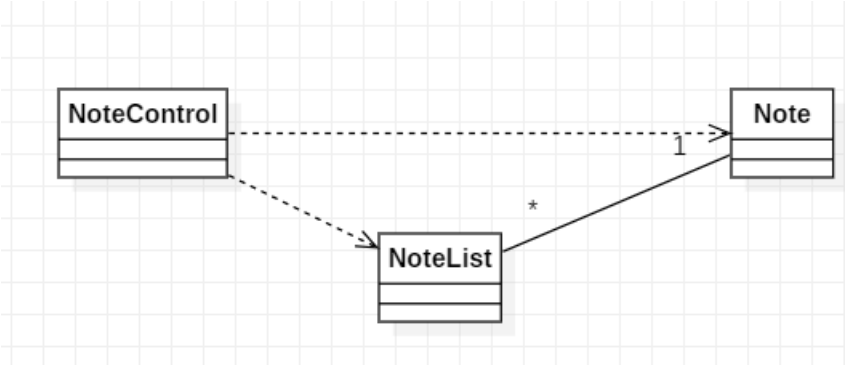

Figure 5: Subsystem class diagram of note management

(4) Preview Management

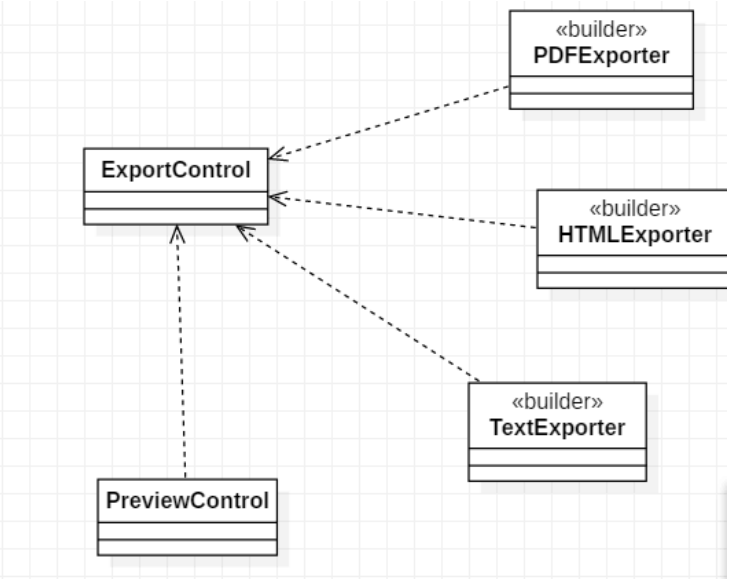

Figure 6: Subsystem class diagram of preview management

(5) Cloud Service

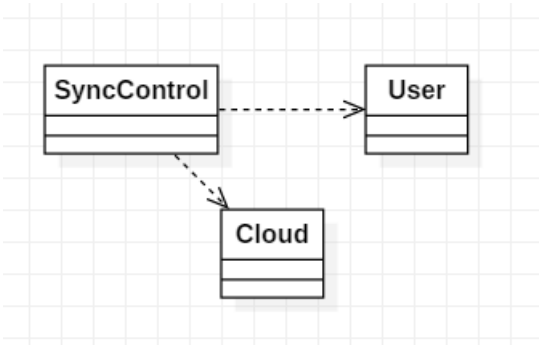

Figure 7: Subsystem class diagram of cloud service

(6) Local Service

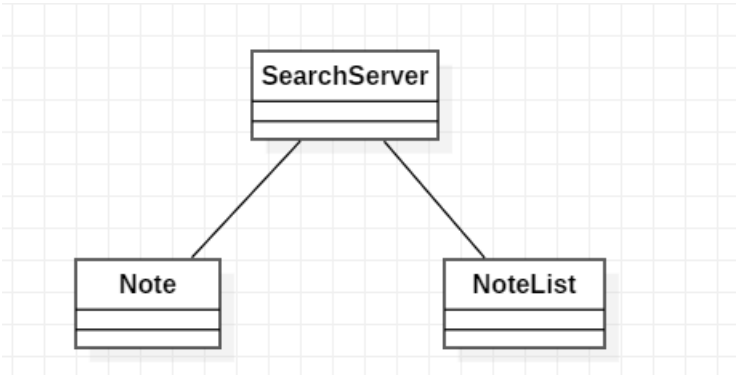

Figure 8: Subsystem class diagram of local service

3.2 Use Case Realization

a) Use case realization: ViewAllNotes

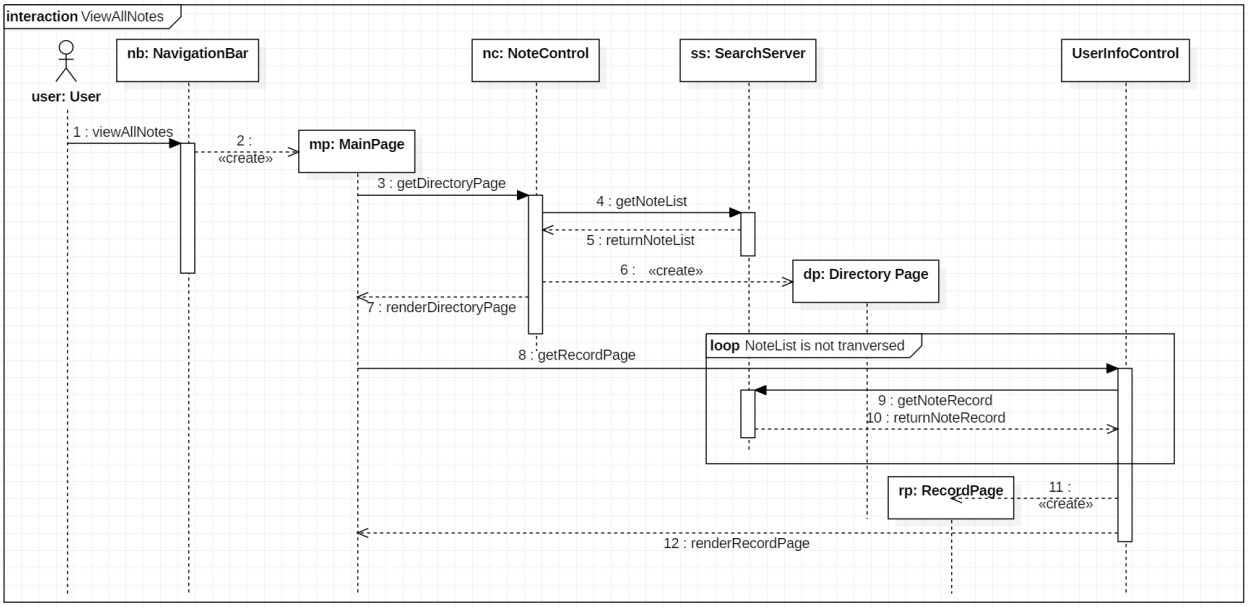

Figure 9: Realization of use case ViewAllNotes

b) Use case realization: Login

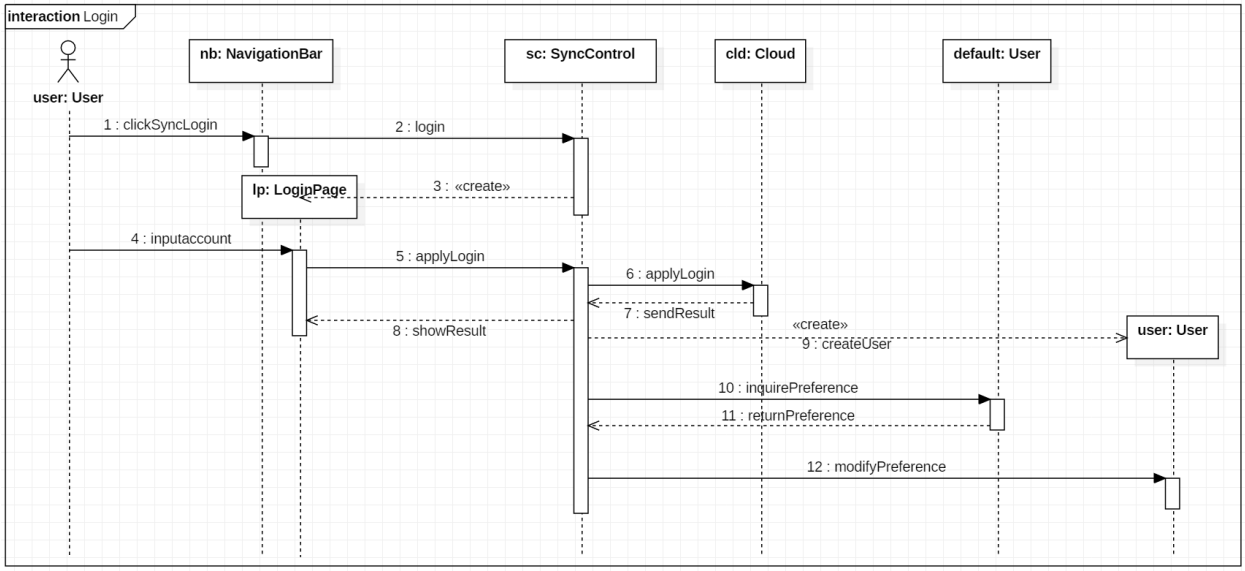

Figure 10: Realization of use case Login

c) Use case realization: Synchronization

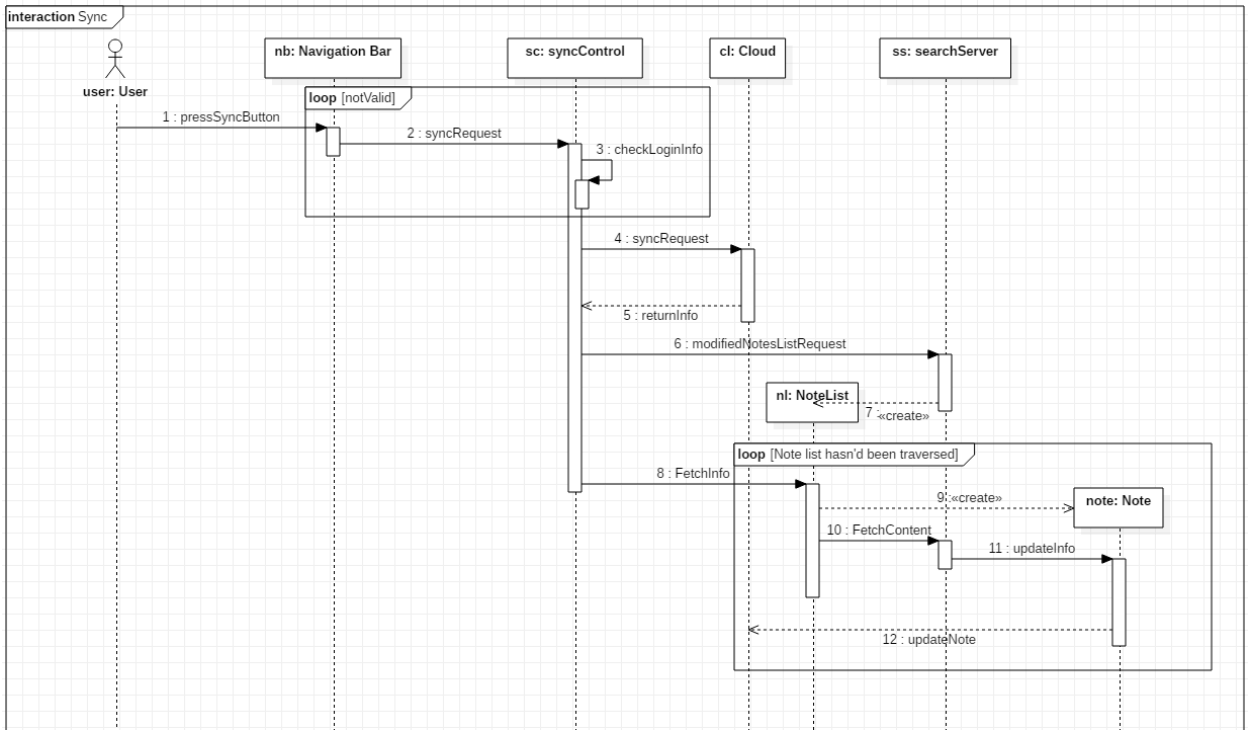

Figure 11: Realization of use case Synchronization

d) Use case realization: CreateNote

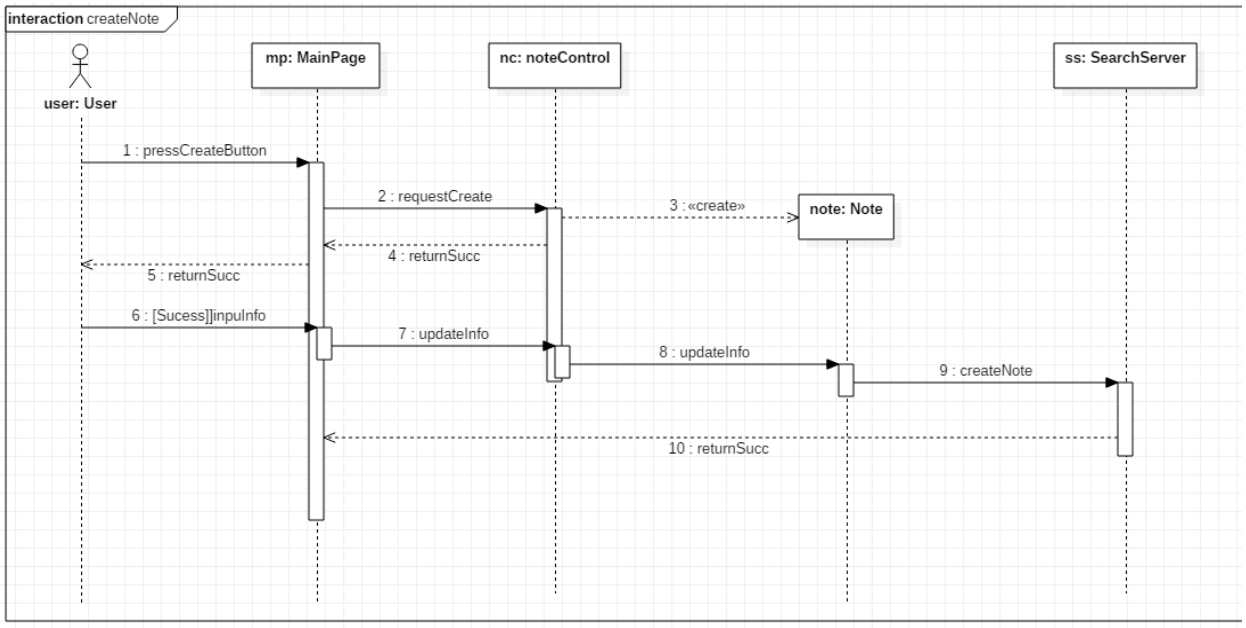

Figure 12: Realization of use case CreateNote

e) Use case realization: EditNote

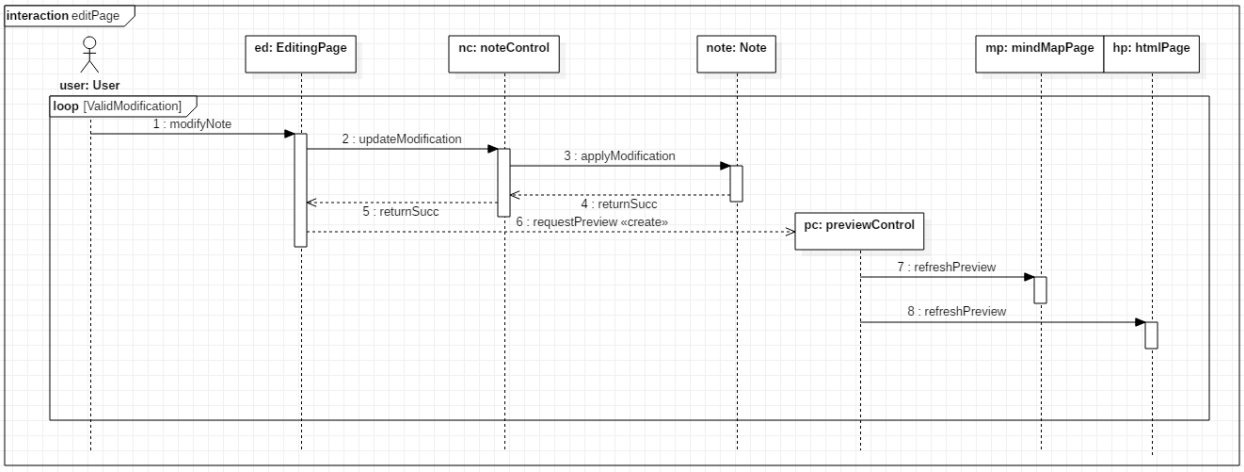

Figure 13: Realization of use case EditNote

f) Use case realization: CloseNote

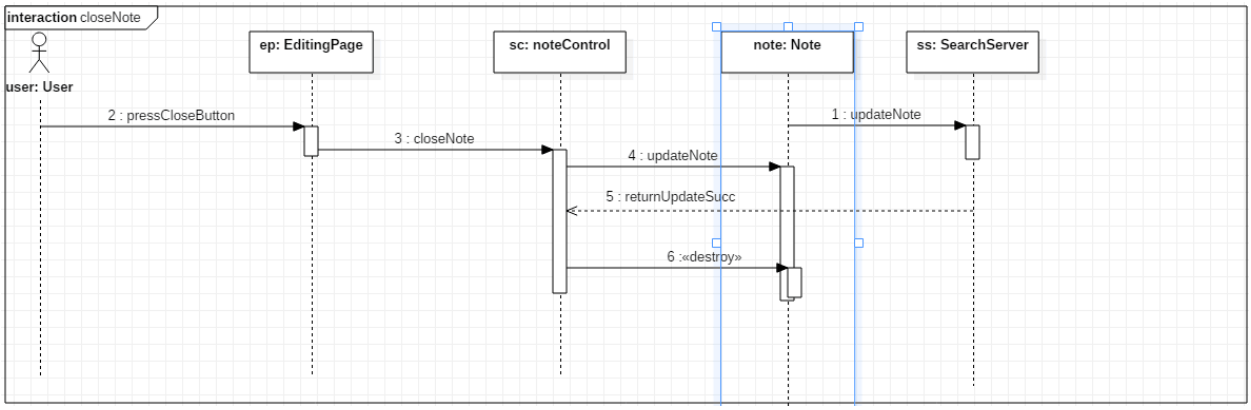

Figure 14: Realization of use case CloseNote

g) Use case realization: MakePreview

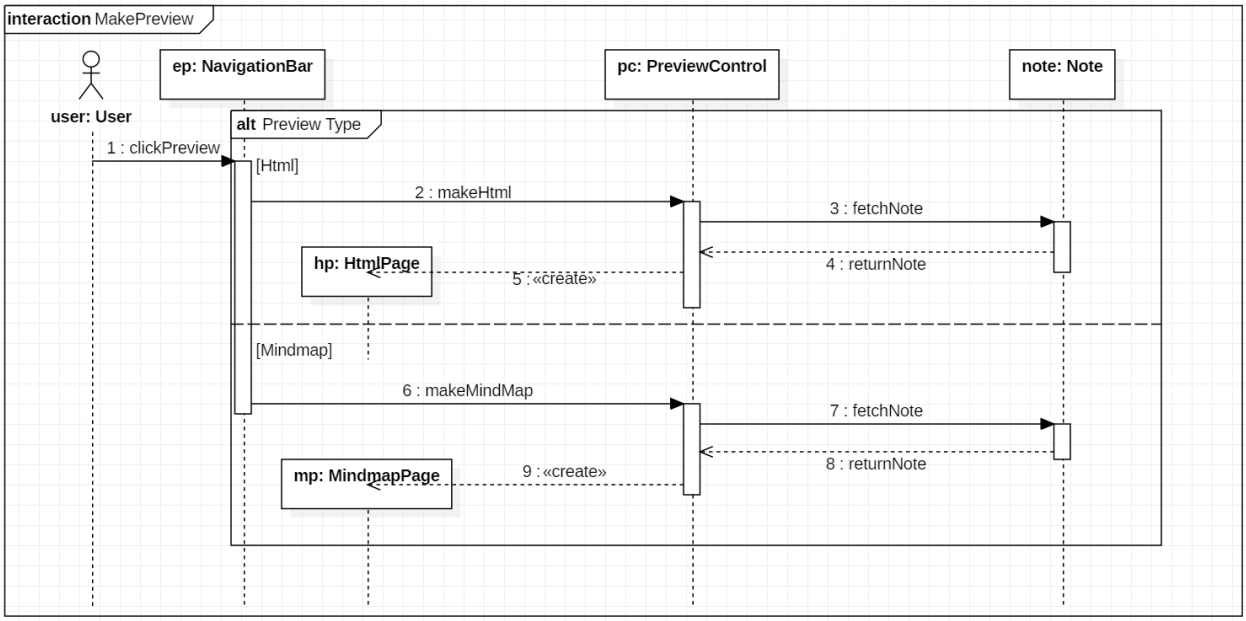

Figure 15: Realization of use case MakePreview

h) Use case realization: ExportFile

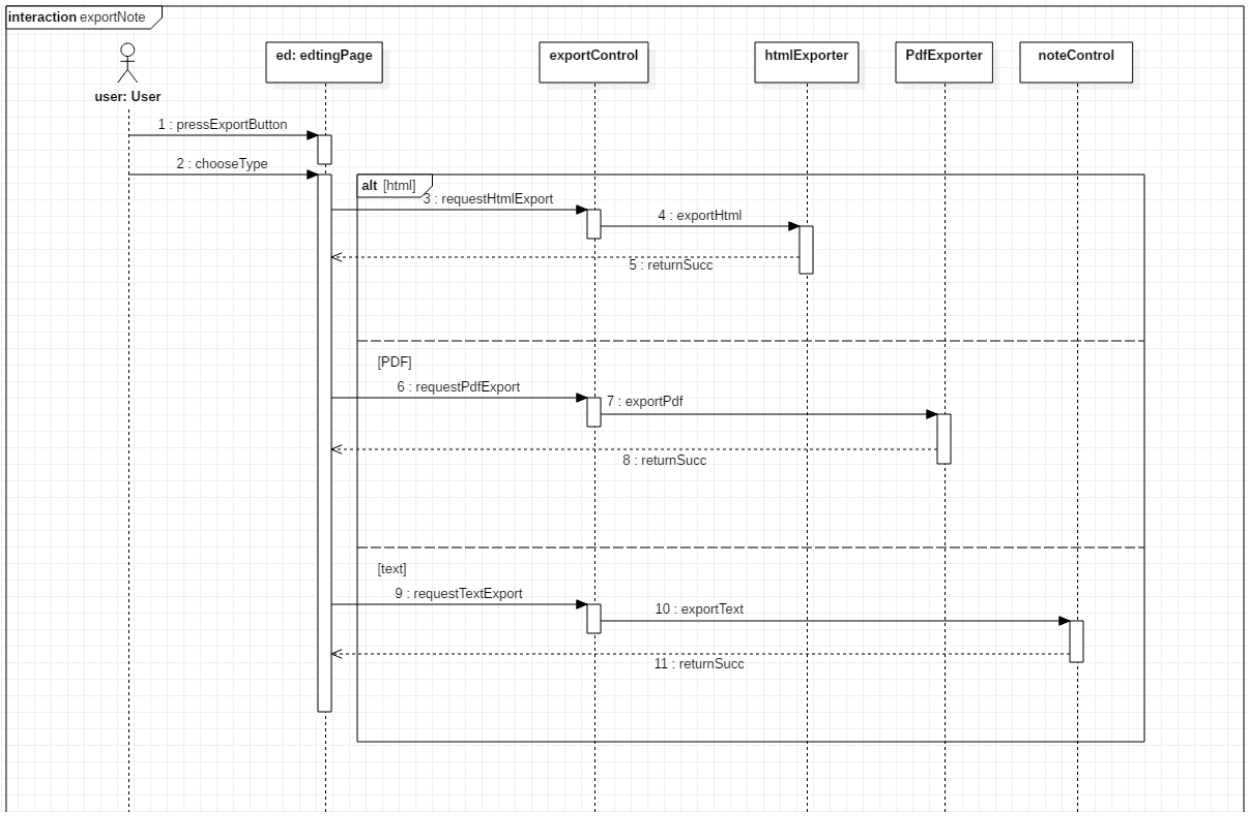

Figure 16: Realization of use case ExportFile

i) Use case realization: Search

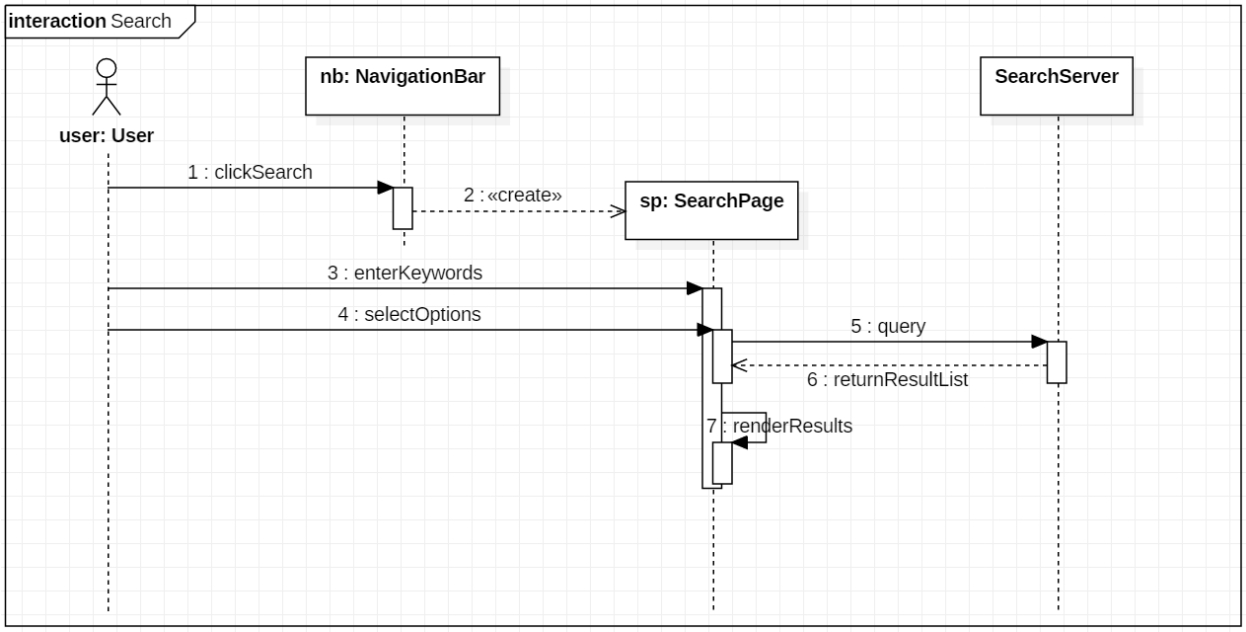

Figure 17: Realization of use case Search

3.3 Design Class Diagram

We list the design classes according to the subsystems they are in.

(1) UI design class

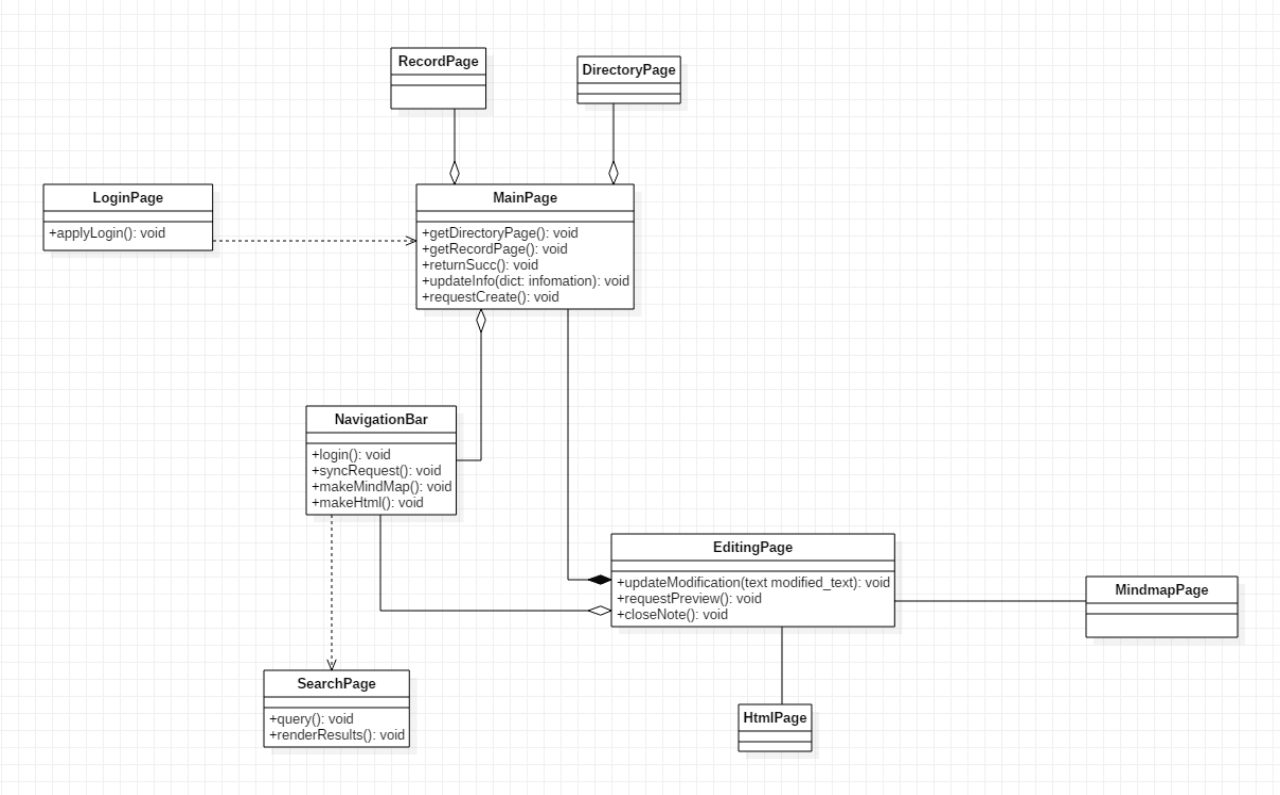

Figure 18: Design class diagram of UI

(2) User Management

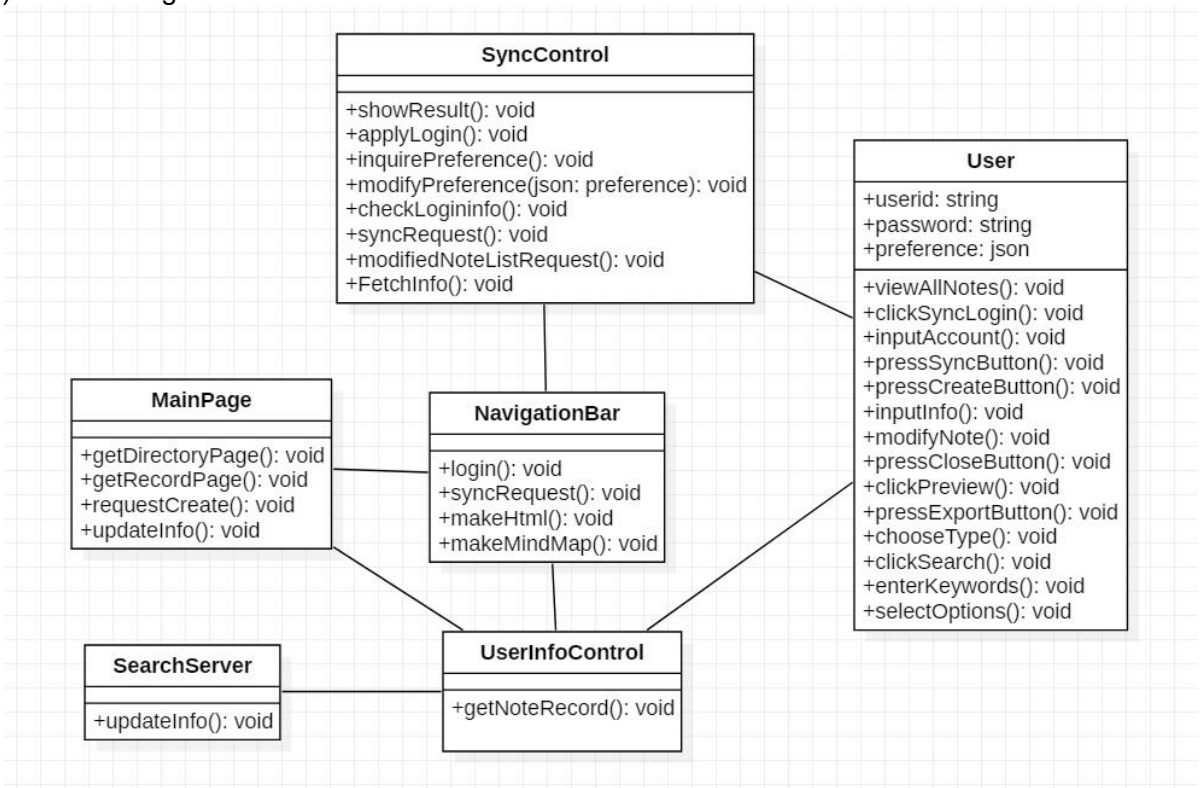

Figure 19: Design class diagram of user management

(3) Note Management

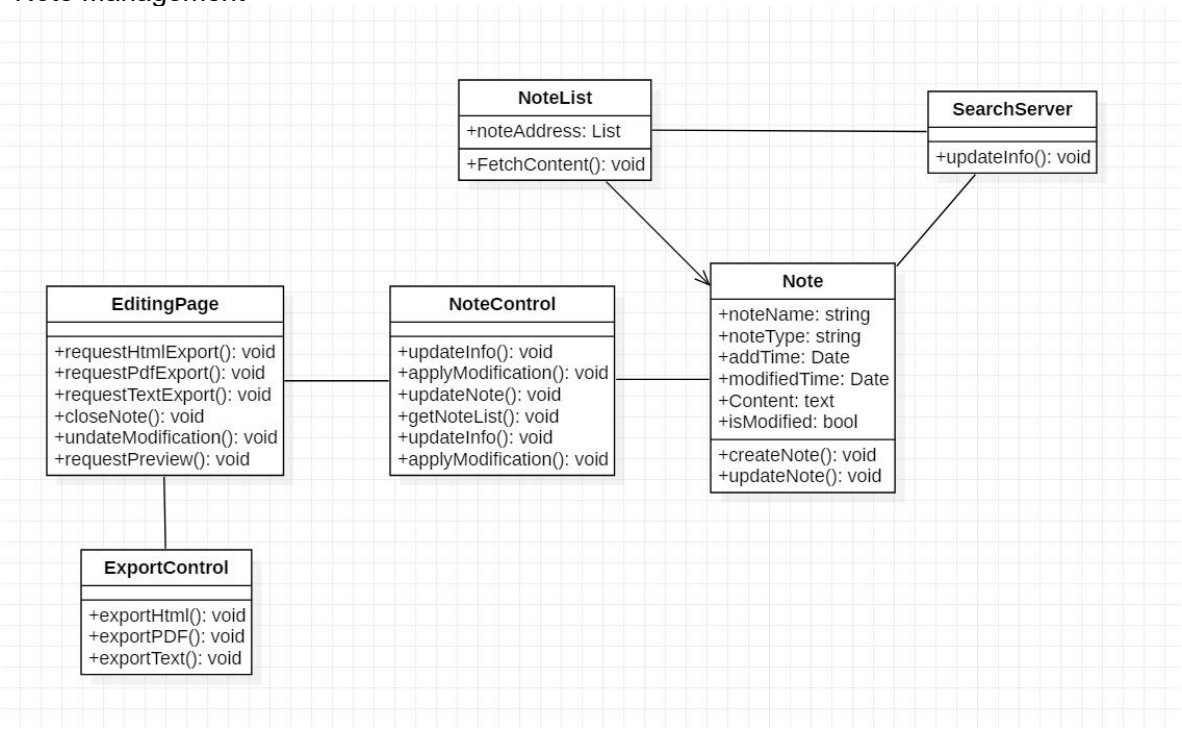

Figure 20: Design class diagram of note management

(4) Preview Management

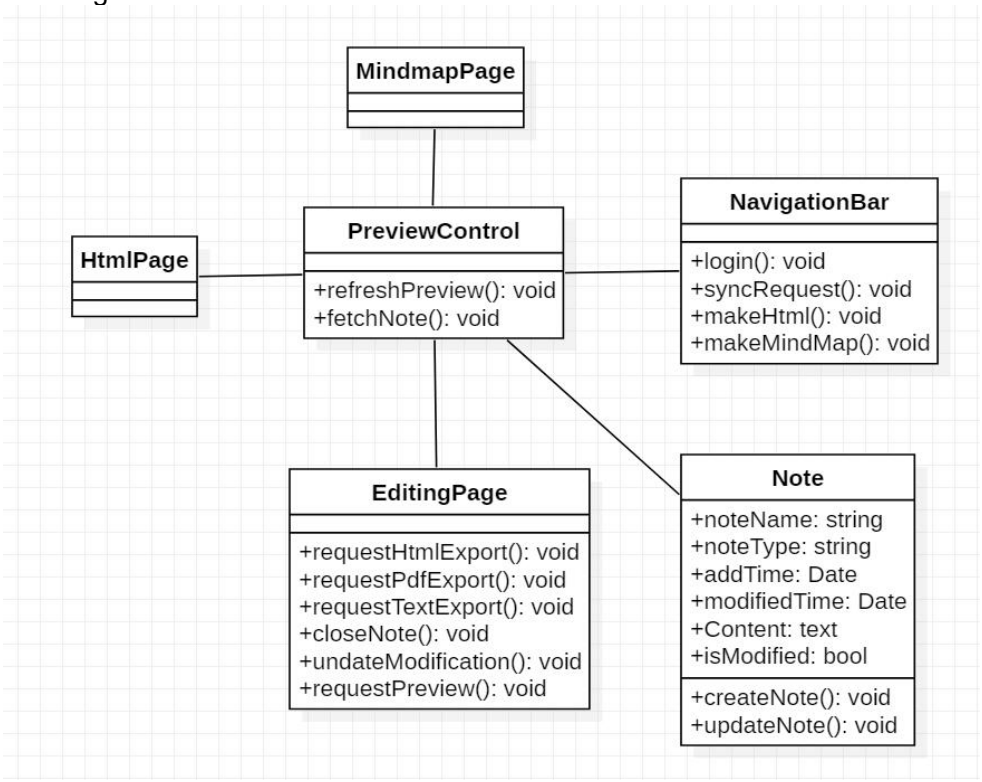

Figure 21: Design class diagram of preview management

(5) Cloud Service

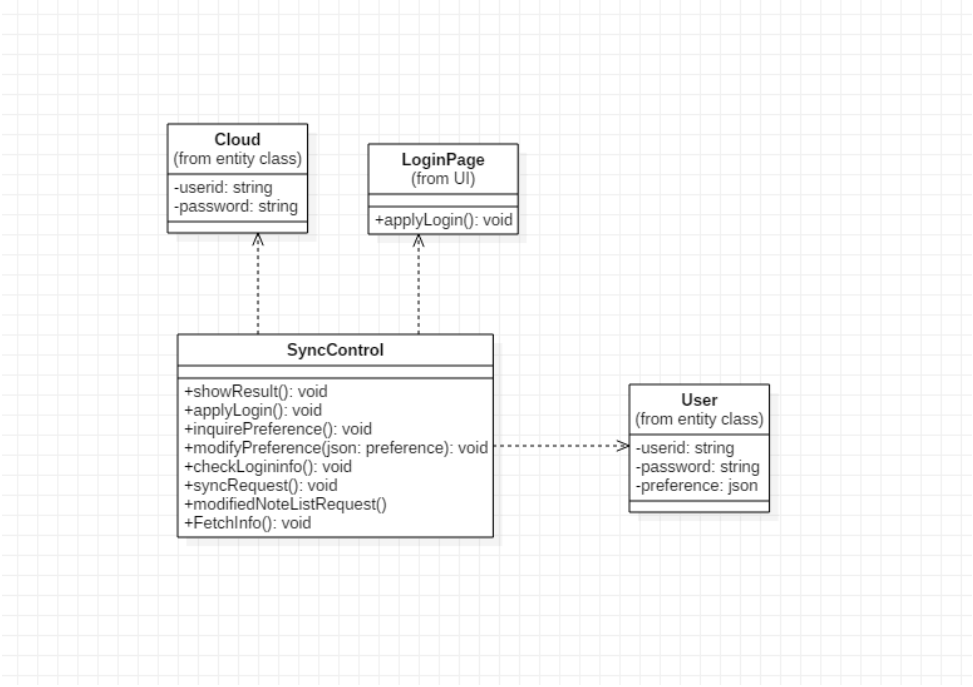

Figure 22: Design class diagram of cloud service

(6) Local Service

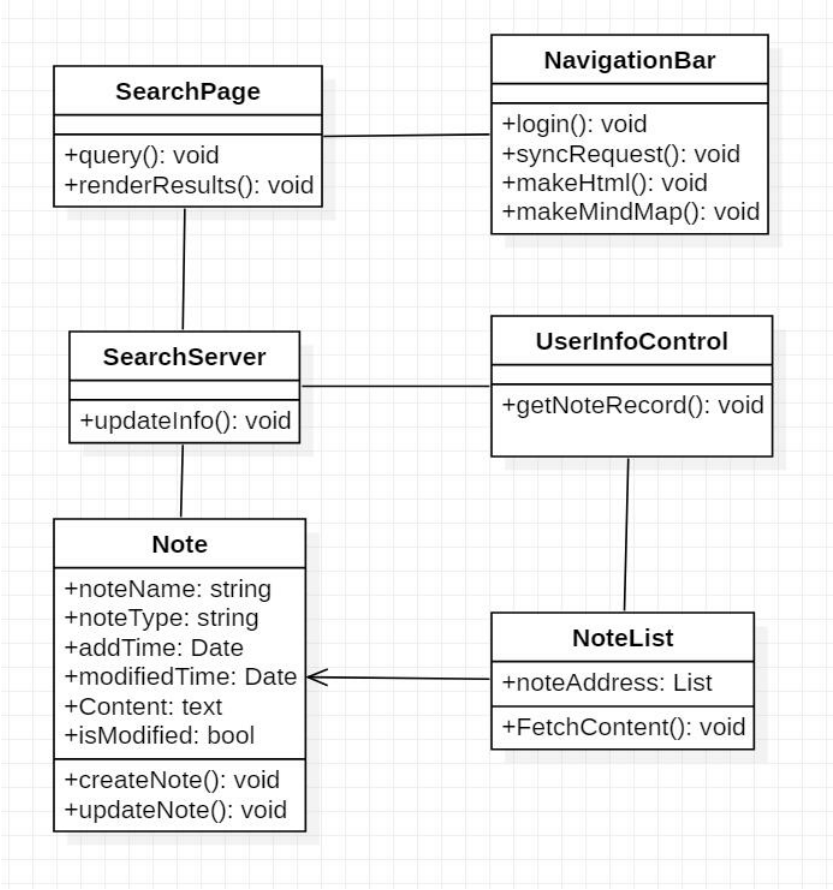

Figure 23: Design class diagram of local service

3.4 Other

3.4.1 State Diagram for Note

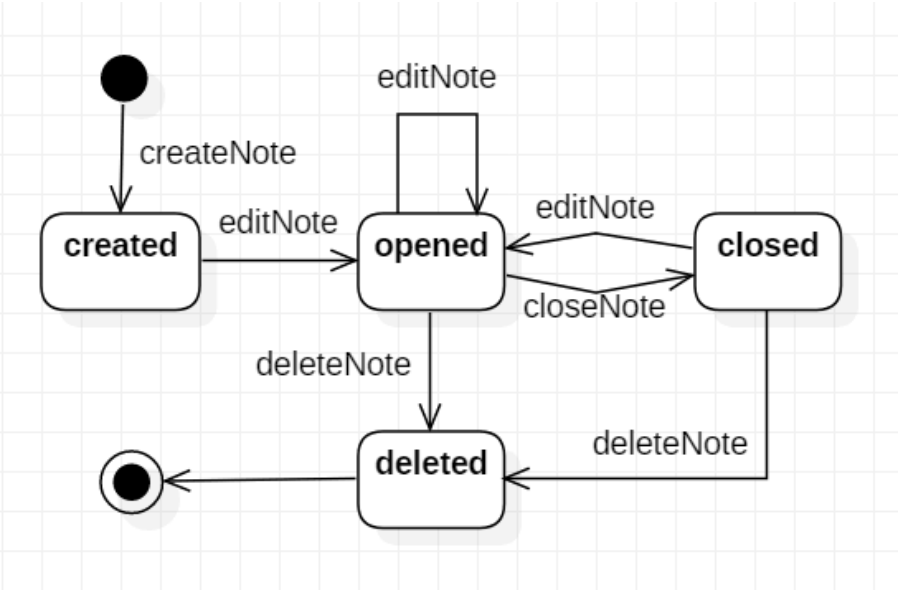

Figure 24: State diagram for Note

4. Implementation View

4.1 System in Development

(1) Subsystem UI

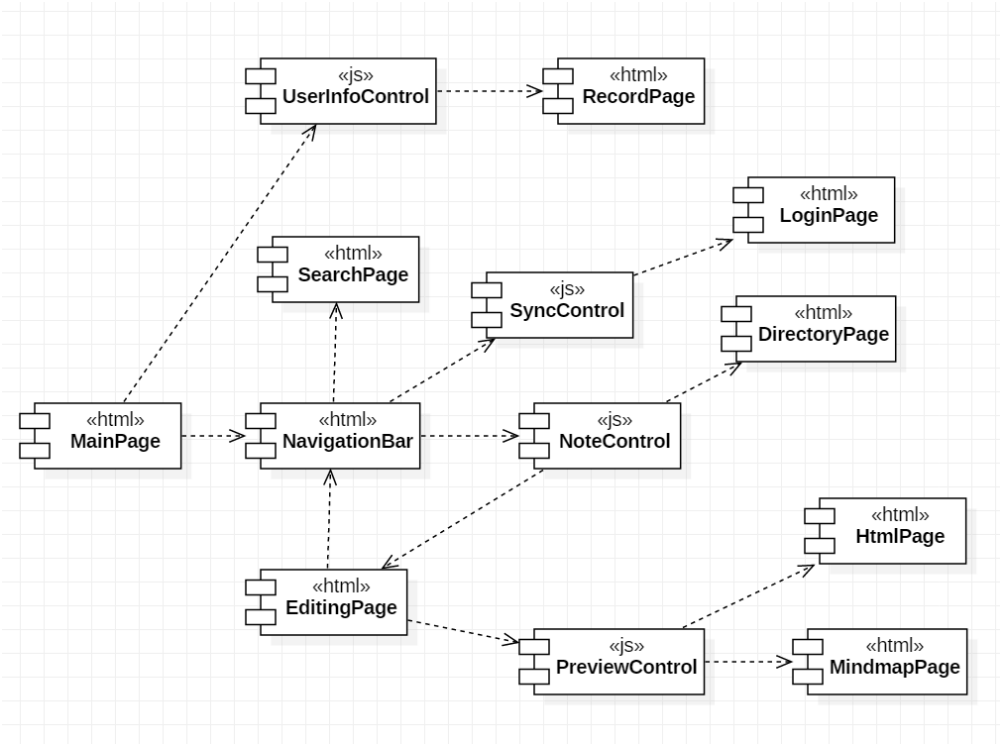

Figure 25: Component diagram of UI

(2) Subsystem User Management

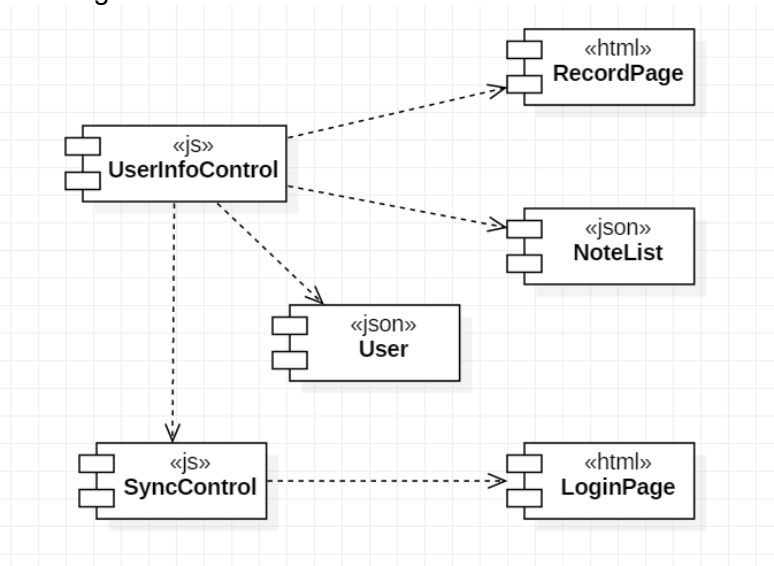

Figure 26: Component diagram of user management

(3) Subsystem Note Management

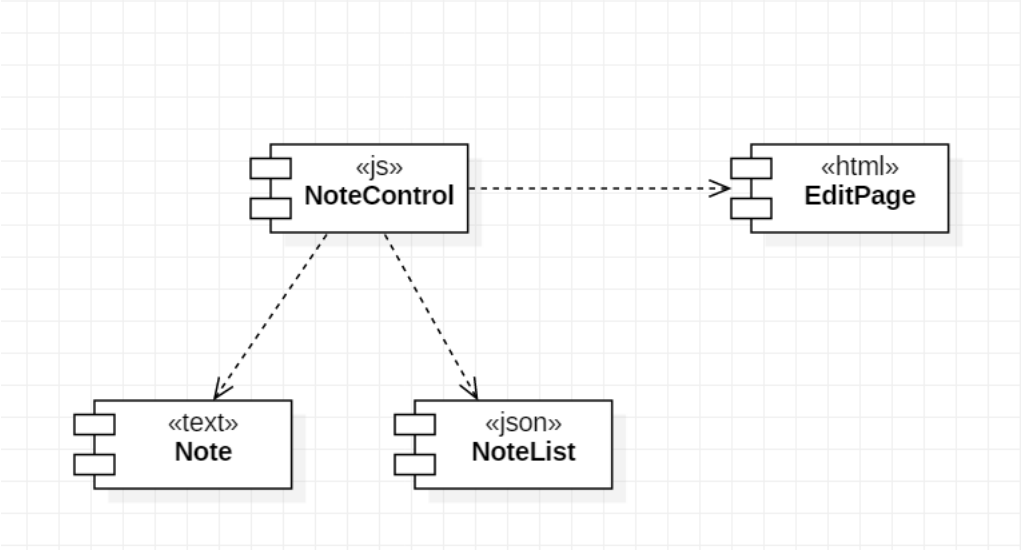

Figure 27: Component diagram of note management

(4) Subsystem Preview Management

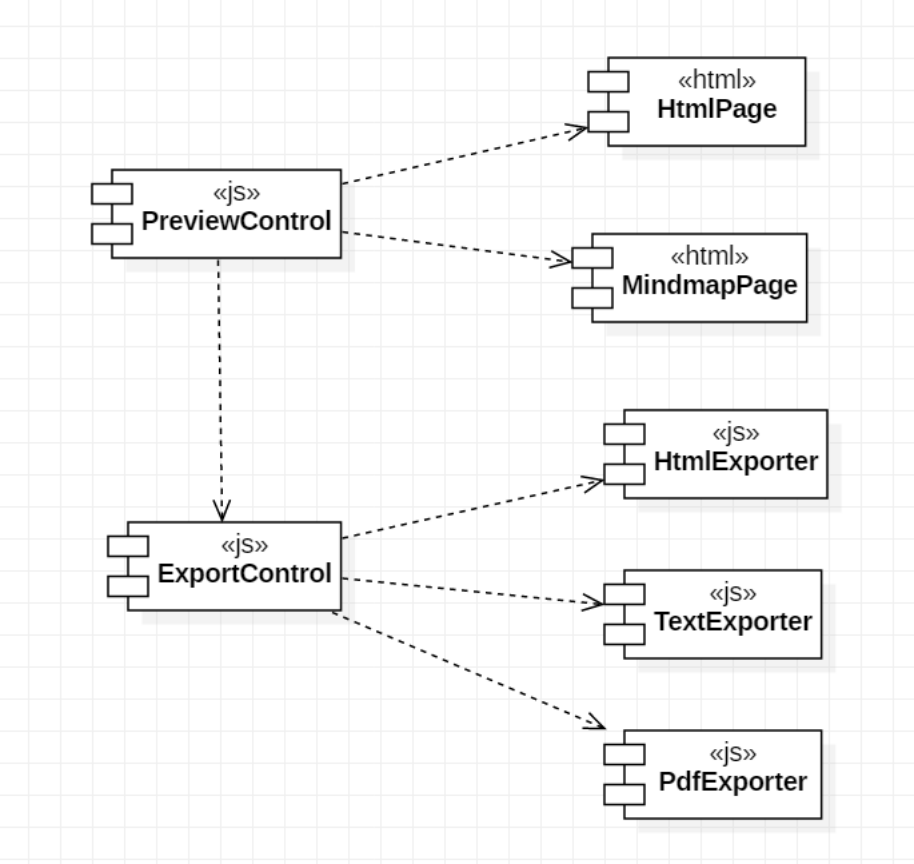

Figure 28: Component diagram of preview management

(5) Subsystem Cloud Service

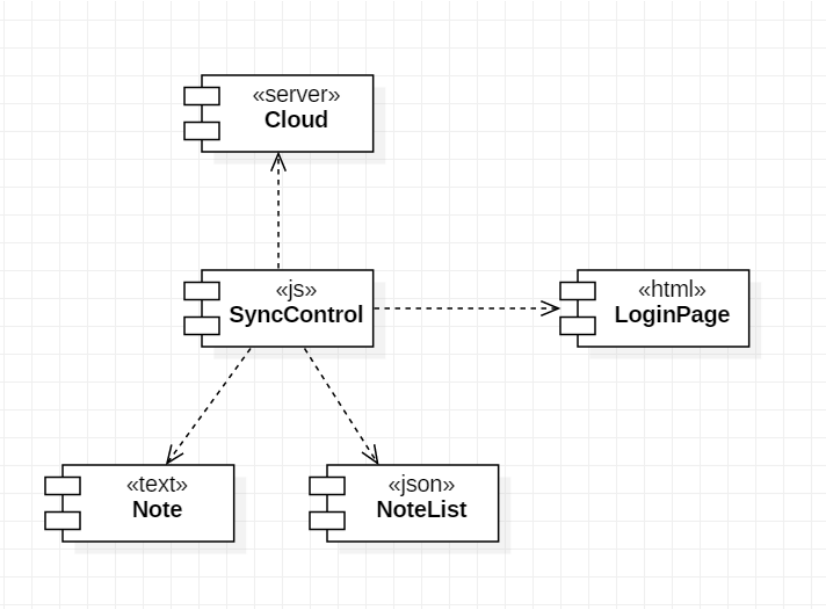

Figure 29: Component diagram of cloud service

(6) Subsystem Local Service

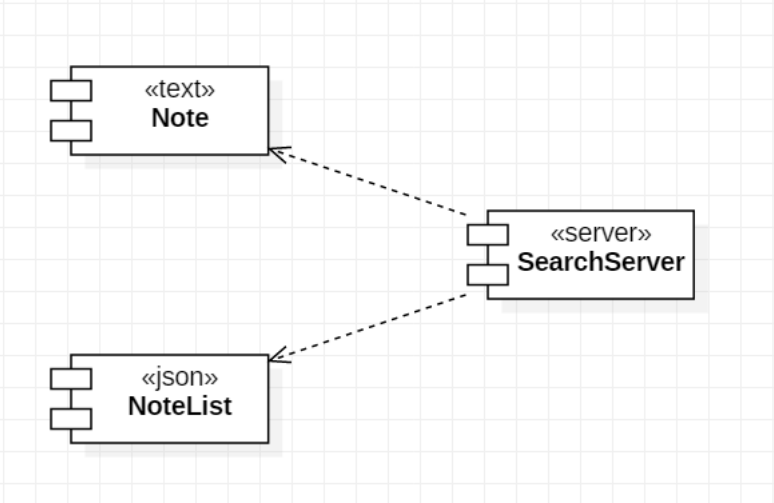

Figure 30: Component diagram of local service

4.2 Compiled System

Since Electron framework is based on Javascript, a script language, so it is interpreted at runtime. The compiled version is almost the same as the system in development. However, the framework defines an entrance to the system and it is shown below.

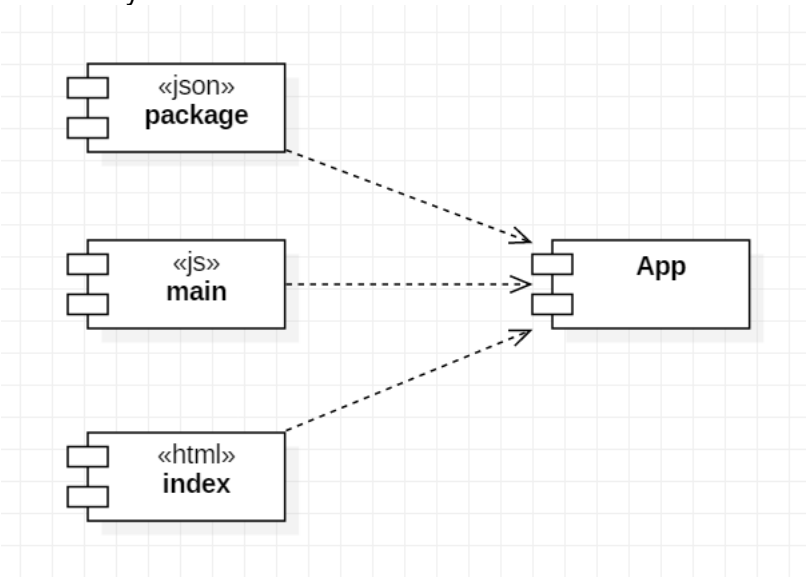

Figure 31: Component diagram for the entrance to the App

5. Process View

Since we will use Electron framework which creates two processes at runtime: Main Process to run the main app and Render Process to show UI, so we have two processes. We believe that some but not all the controllers should be assigned a thread.

For example, NoteControl, SyncControl and SearchServer should be assigned a thread so that synchronization can be automatically triggered when user is editing a note. UserInfoControl does not have a thread as it can be activated by the SyncControl. PreviewControl does not have a thread either, as it is activated by NoteControl. Those two control don’t need to be working all the time.

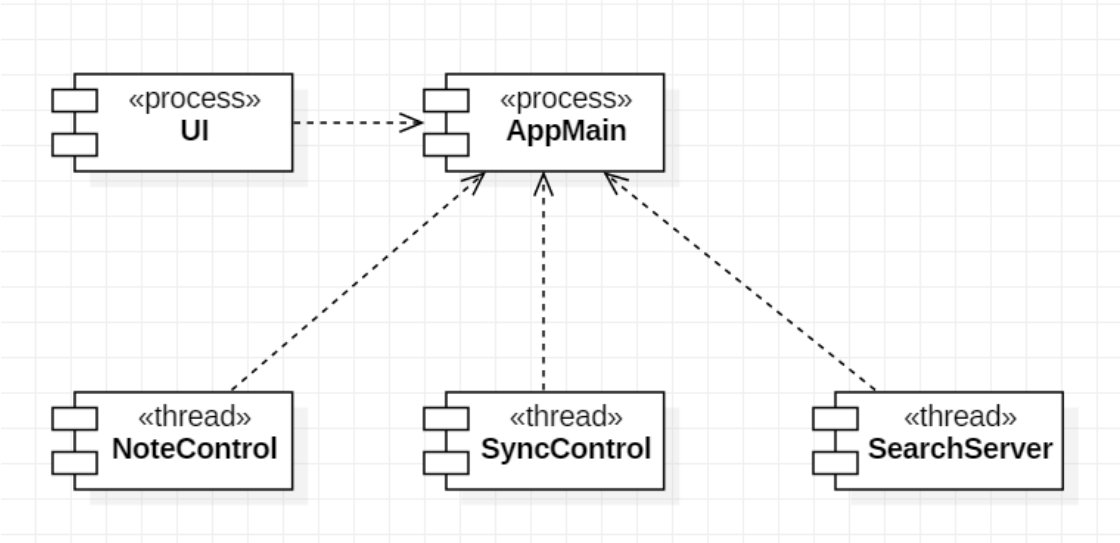

Figure 32: System runtime diagram

6. Deployment View

The system is a PC application, so the main App runs on a desktop computer. There will be a cloud server to ensure the cloud storage feature. Local service including local storage and search will be deployed on the same computer where the App runs.

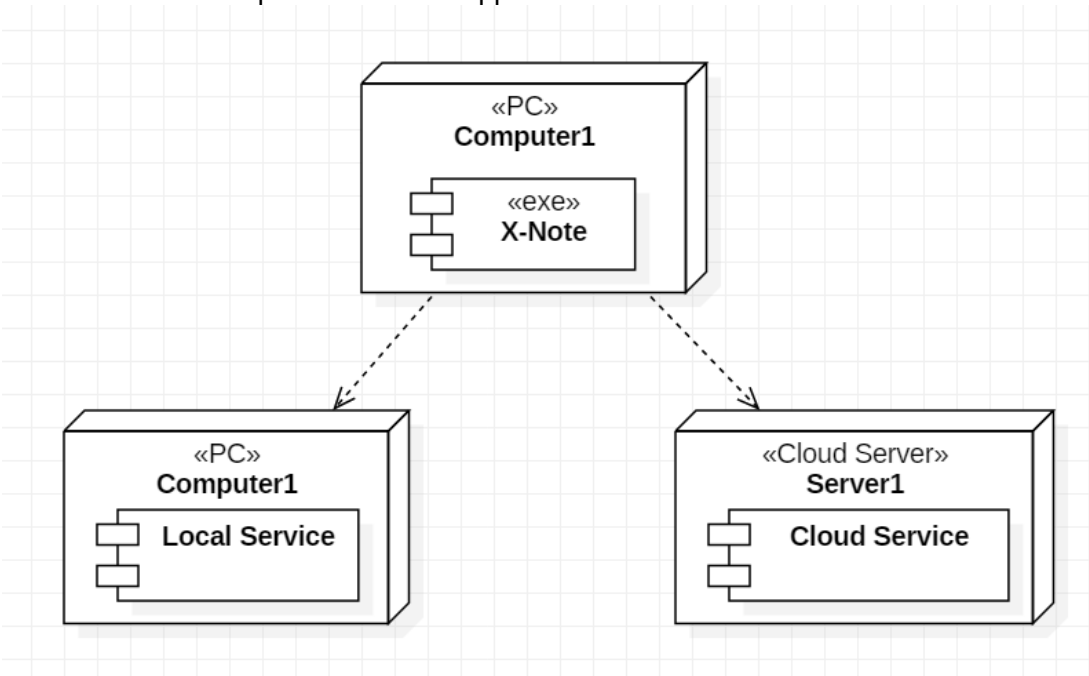

Figure 33: Deployment diagram