YEDITEPE UNIVERSITY
CSE 344 PROJECT
DESIGN REPORT

CSE STUDENTS:

Ali Abbasi Dolatabadi
Gamze Nur Erdem
Ece Tipici
Elif Dikmen
Muhammet Ramazan Dolaş

COURSE INSTRUCTOR:

Associate Prof. Dr. Mert Özkaya

Table of Content

1.INTRODUCTION

1.1 Purpose of the Document

This document aims to provide a concise explanation of class diagrams, covering their characteristics and methods. It will discuss the affiliations, compositions, aggregations, relationship names, and multiplicities of relationships between classes. Additionally, the document will examine dynamic models such as state, sequence, and activity diagrams. It will also address software architecture using UML Package Diagrams and UML Component Diagrams. The goal is to thoroughly explain each component of the software product, as well as the underlying mechanisms of class operations and the overall program architecture.

1.2 Purpose of the System

The purpose of this app is to facilitate an appointment system for students seeking face-to-face meetings with their professors. The app aims to serve university students as a centralized platform for scheduling meetings with professors. It strives to simplify the appointment scheduling process, eliminating the need for extensive email communication and reducing the risk of overlapping appointments.

In summary, this appointment scheduling app empowers students by streamlining communication with professors, eliminating the hassle of email back and forth, and ensuring a smooth scheduling process. By offering a centralized platform accessible through mobile devices, the app allows students to manage their time effectively and prioritize their academic success.

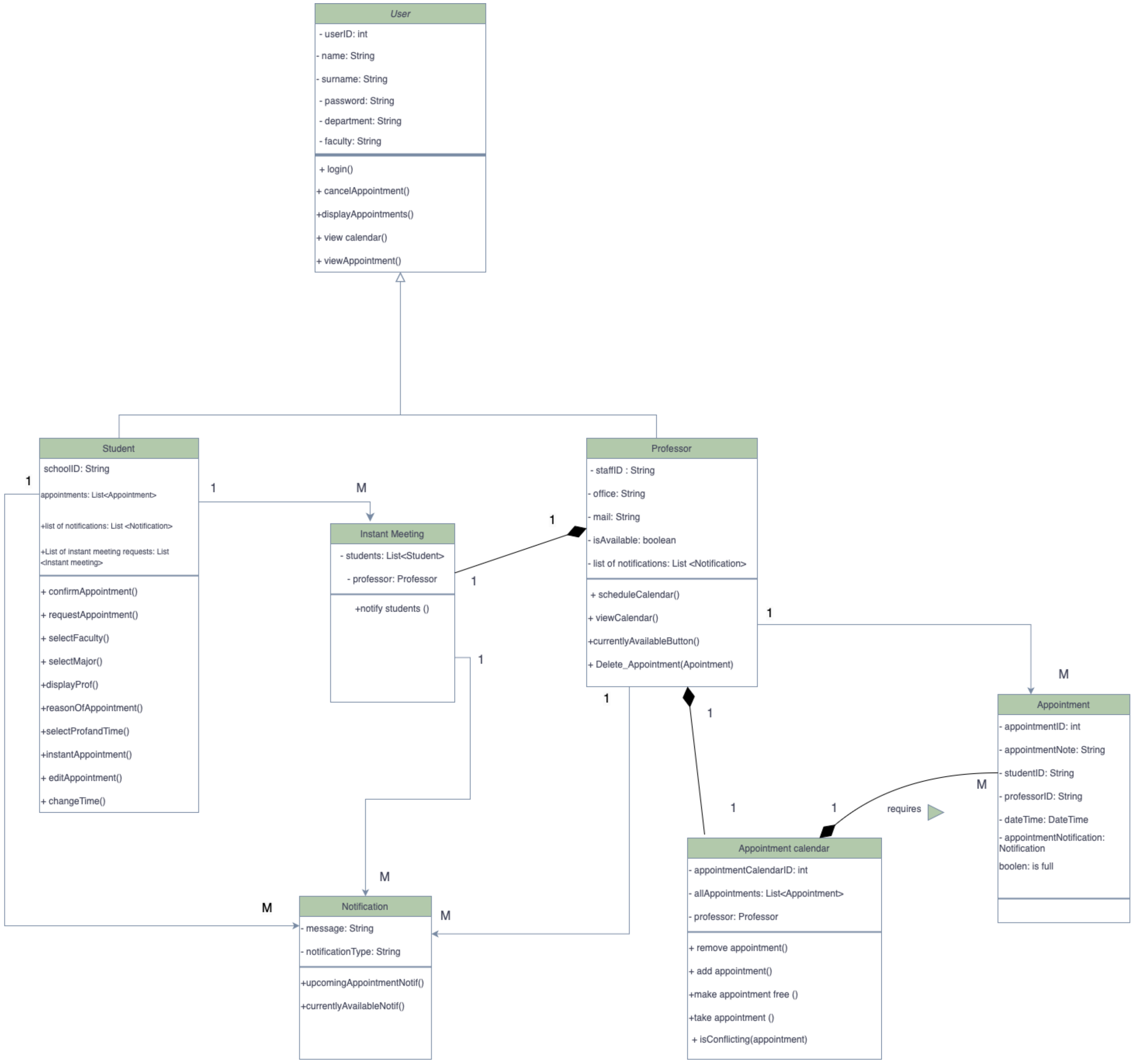
1.3.Structure of the Document

Information regarding classes and their characteristics, processes, and relationships in terms of associations, compositions, aggregations, relationship names, and multiplicities will be provided in another section of the paper. Following that, the document's "Dynamic Models" section, which includes “Sequence Diagrams," "State Diagrams," and "Activity Diagrams," is present. Under the heading "Software Architecture," after the section on "Dynamic Models," there will be "UML Package Diagram" and "UML Component Diagram," then "Entity Relationship Diagram" and "Glossary & References.”

2. DETAILED CLASS DIAGRAM

2.1.Class Diagram

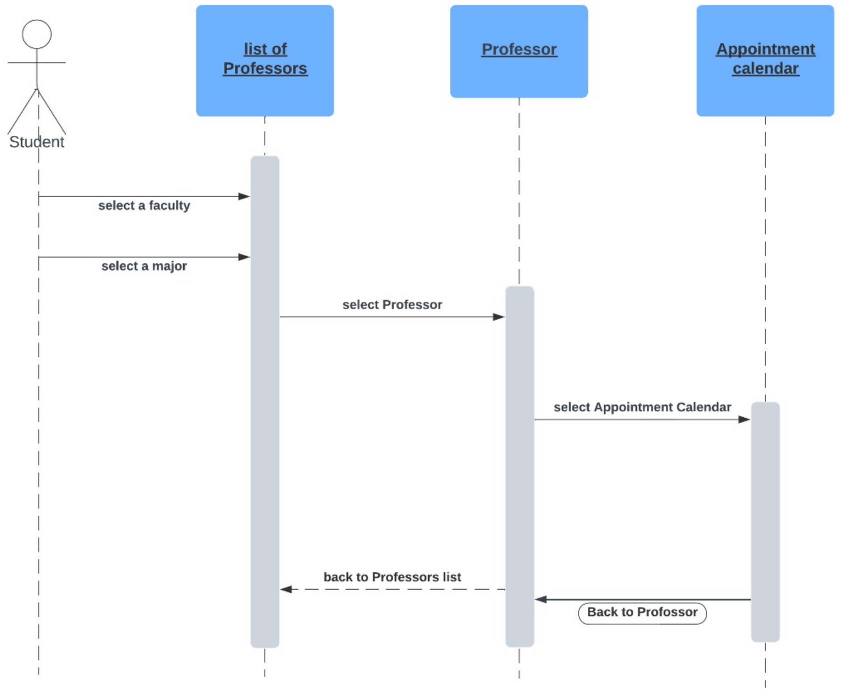
2.1.1 & 2.1.2 Detailed Classes and Detailed Relationships



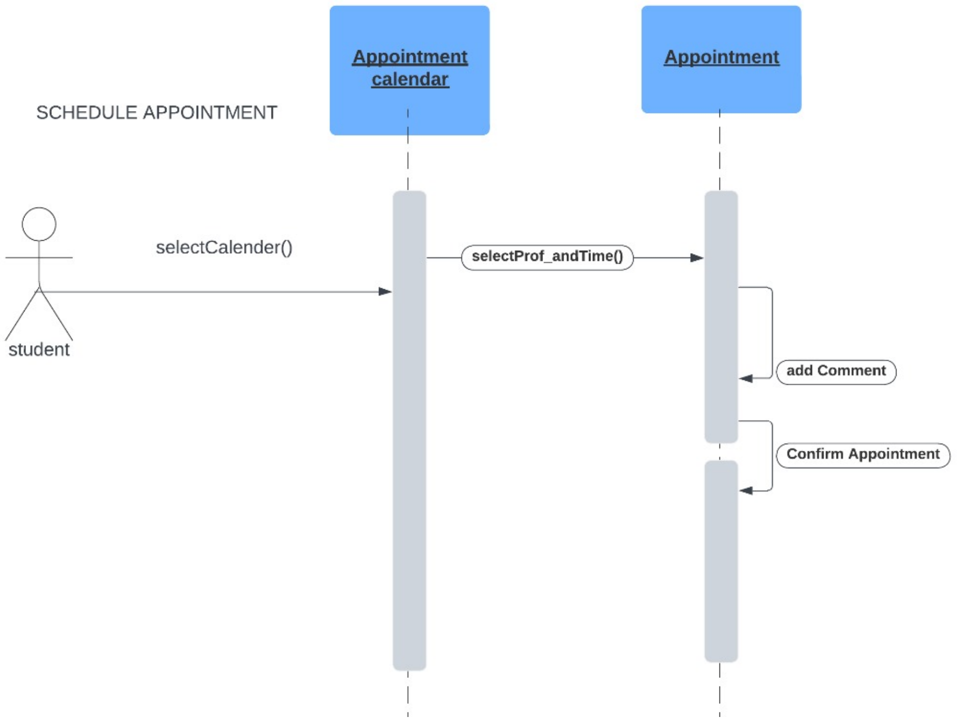
3.DYNAMIC MODELS

3.1) Sequence Diagrams

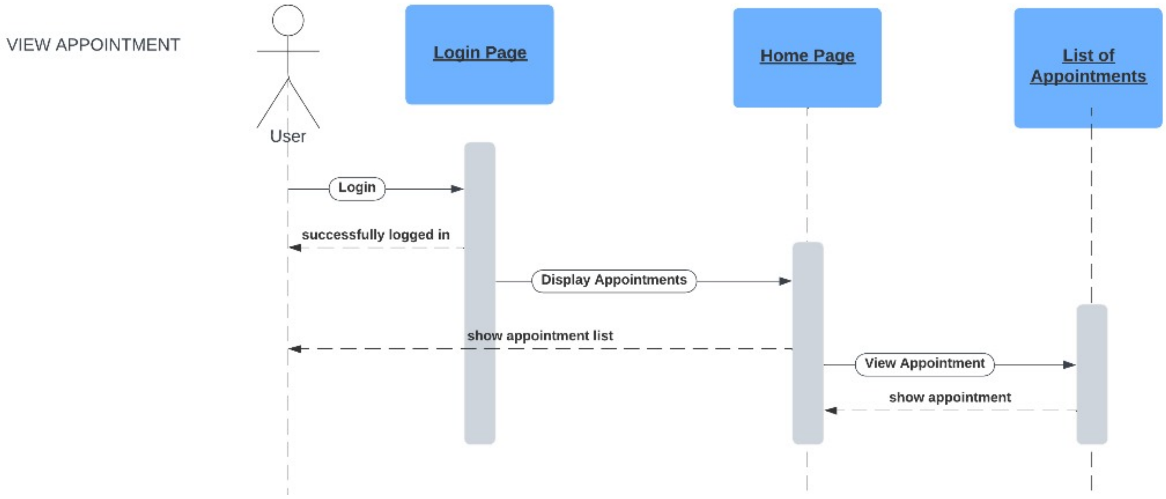
View Calendar:



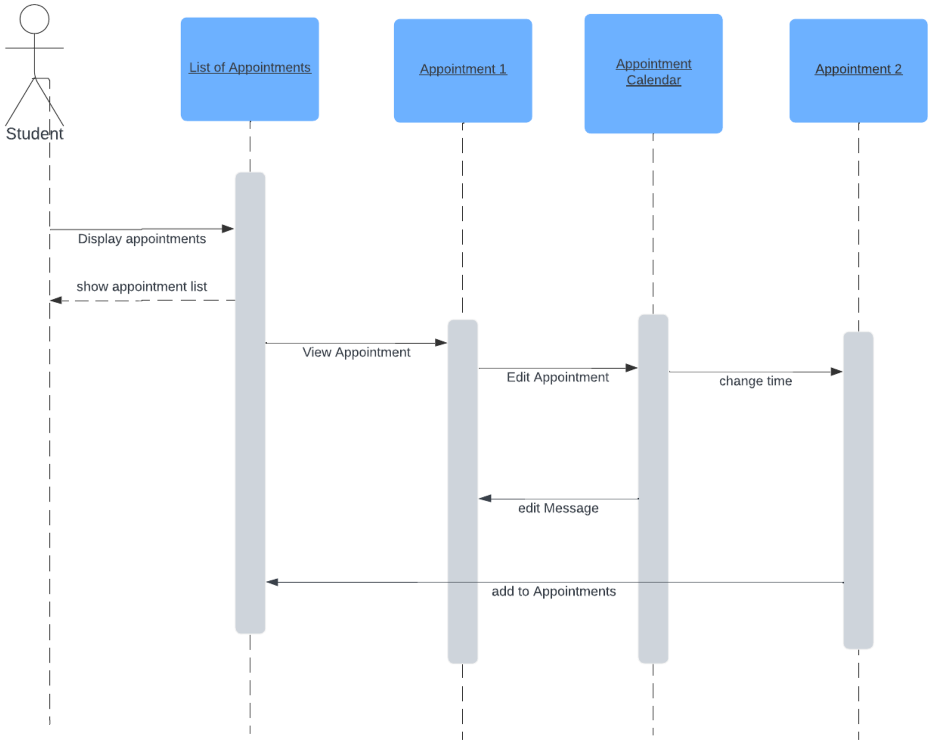
Schedule Appointment:



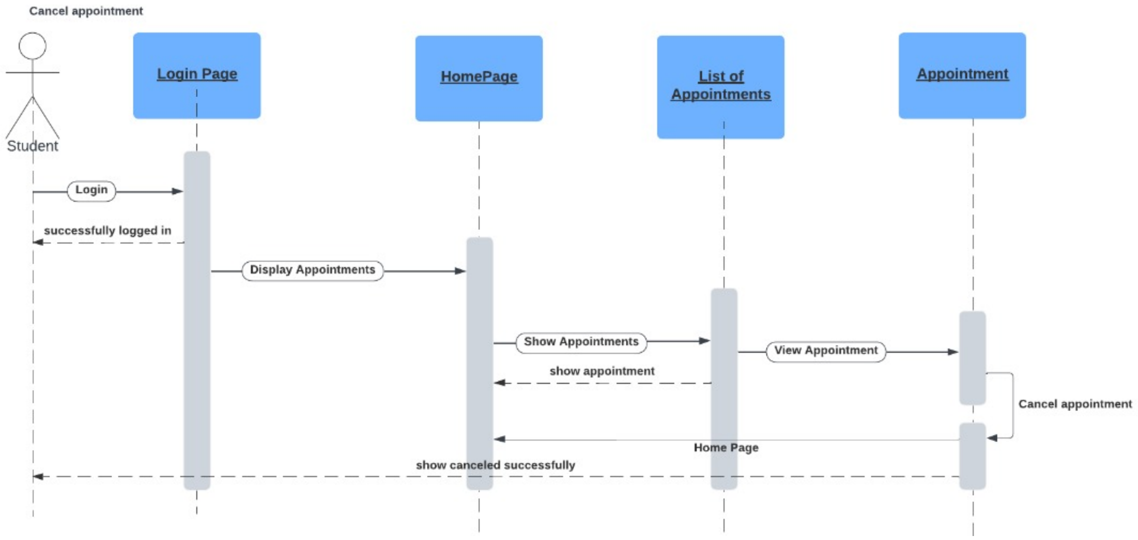
View Appointment:



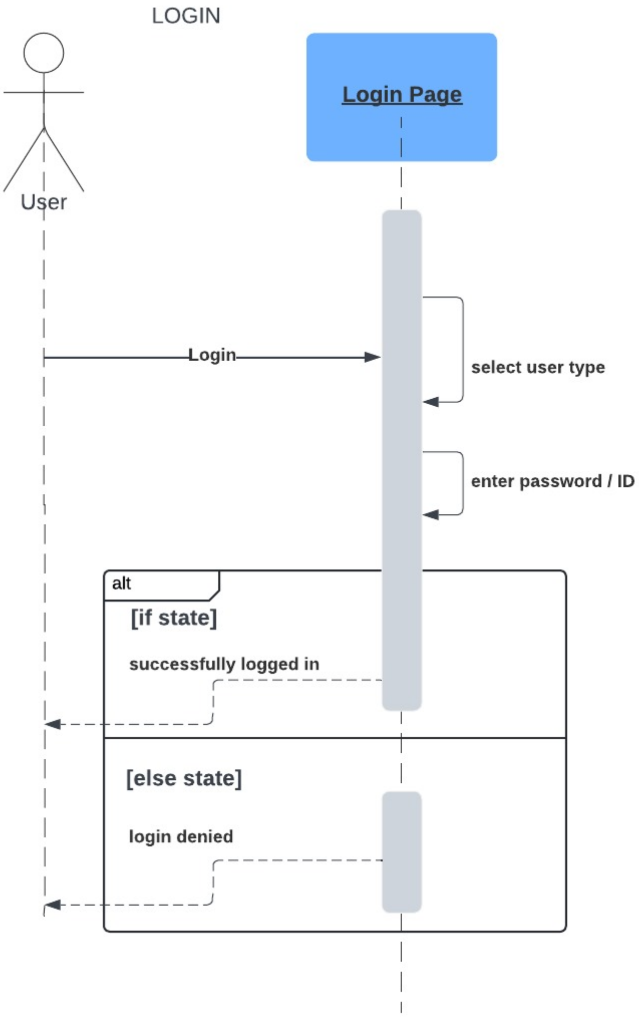
Change Time:



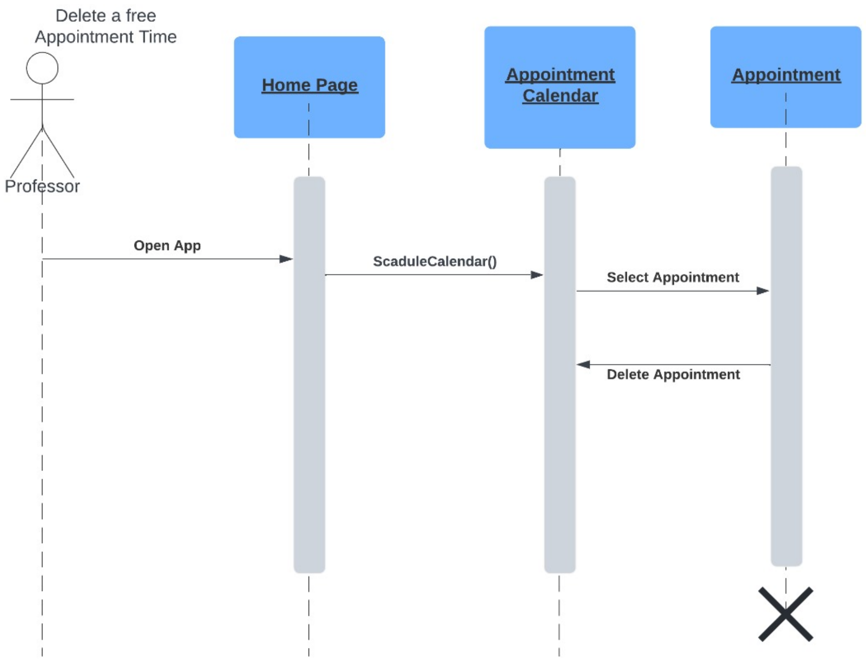
Cancel Appointment:



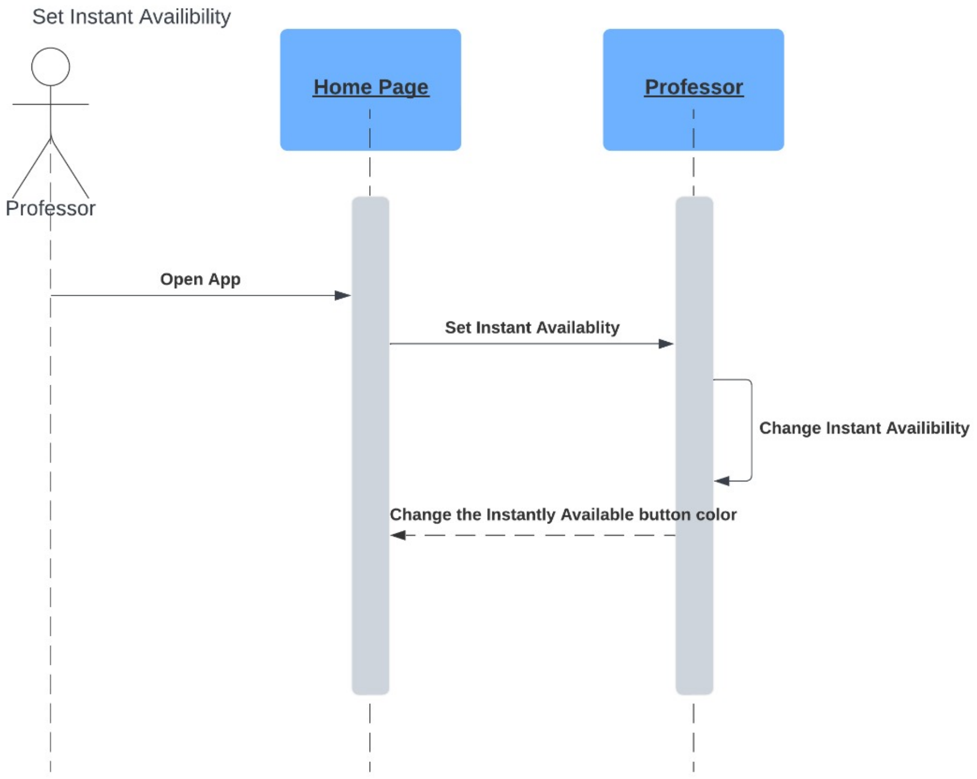
Login:



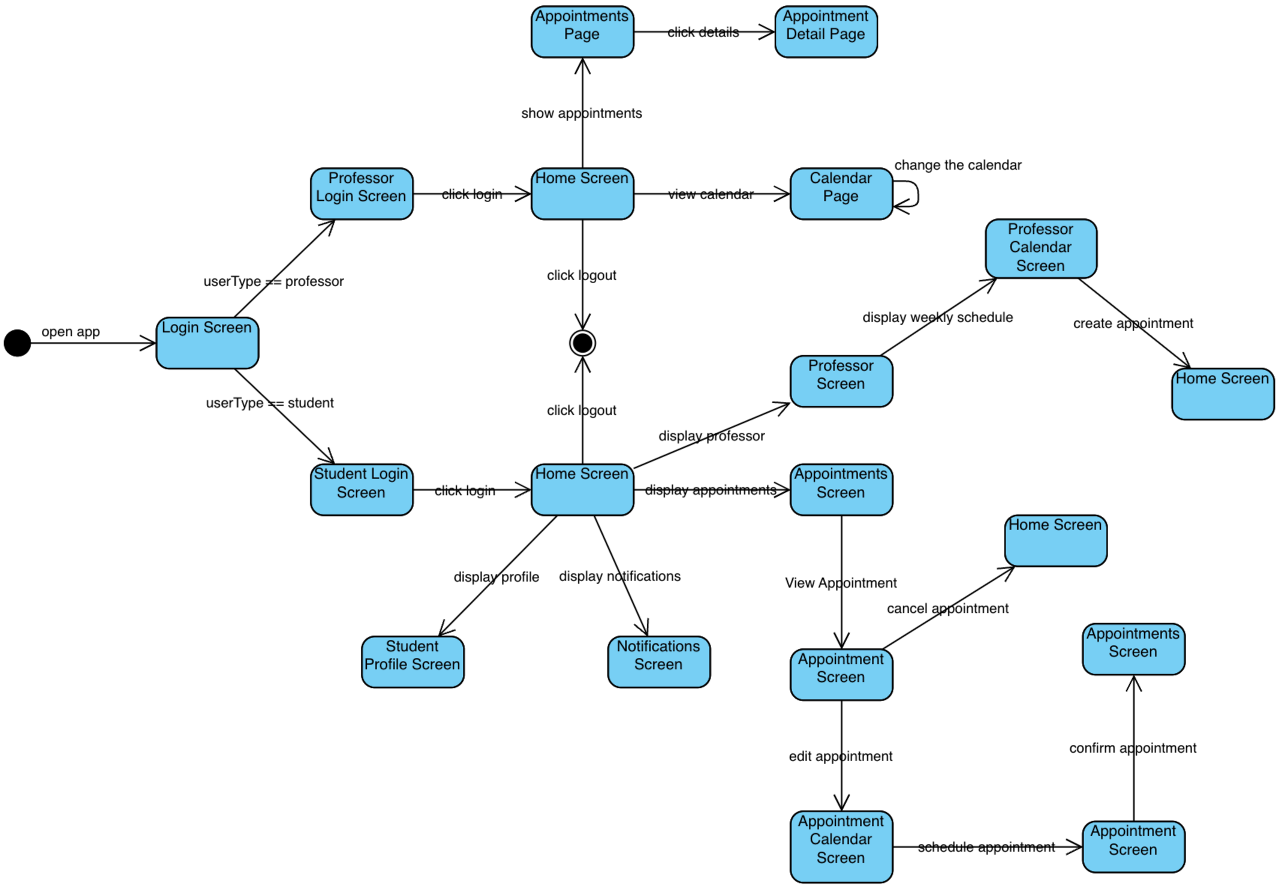
Delete Appointment Time



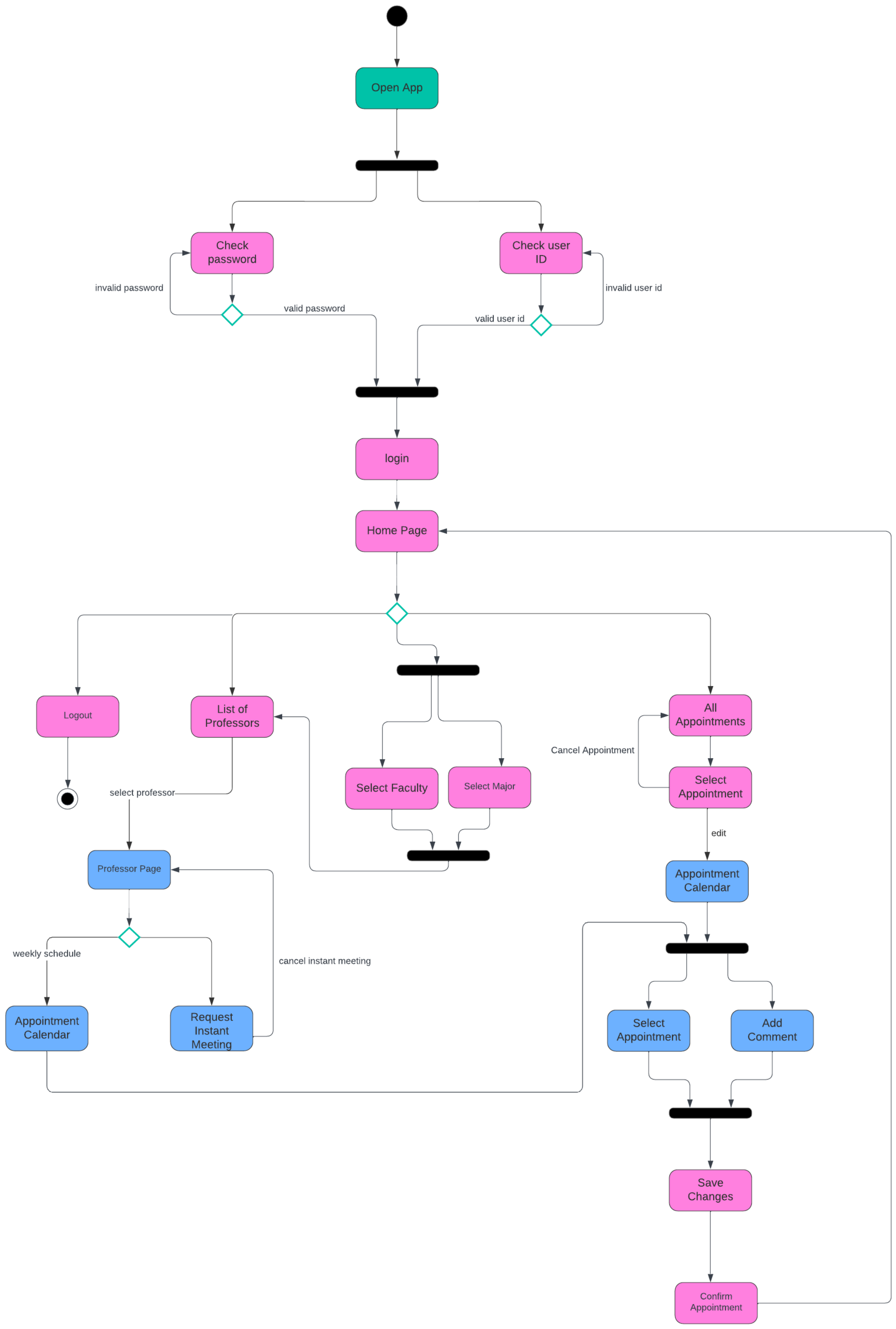
Set Instant Availability



3.2) State Diagram

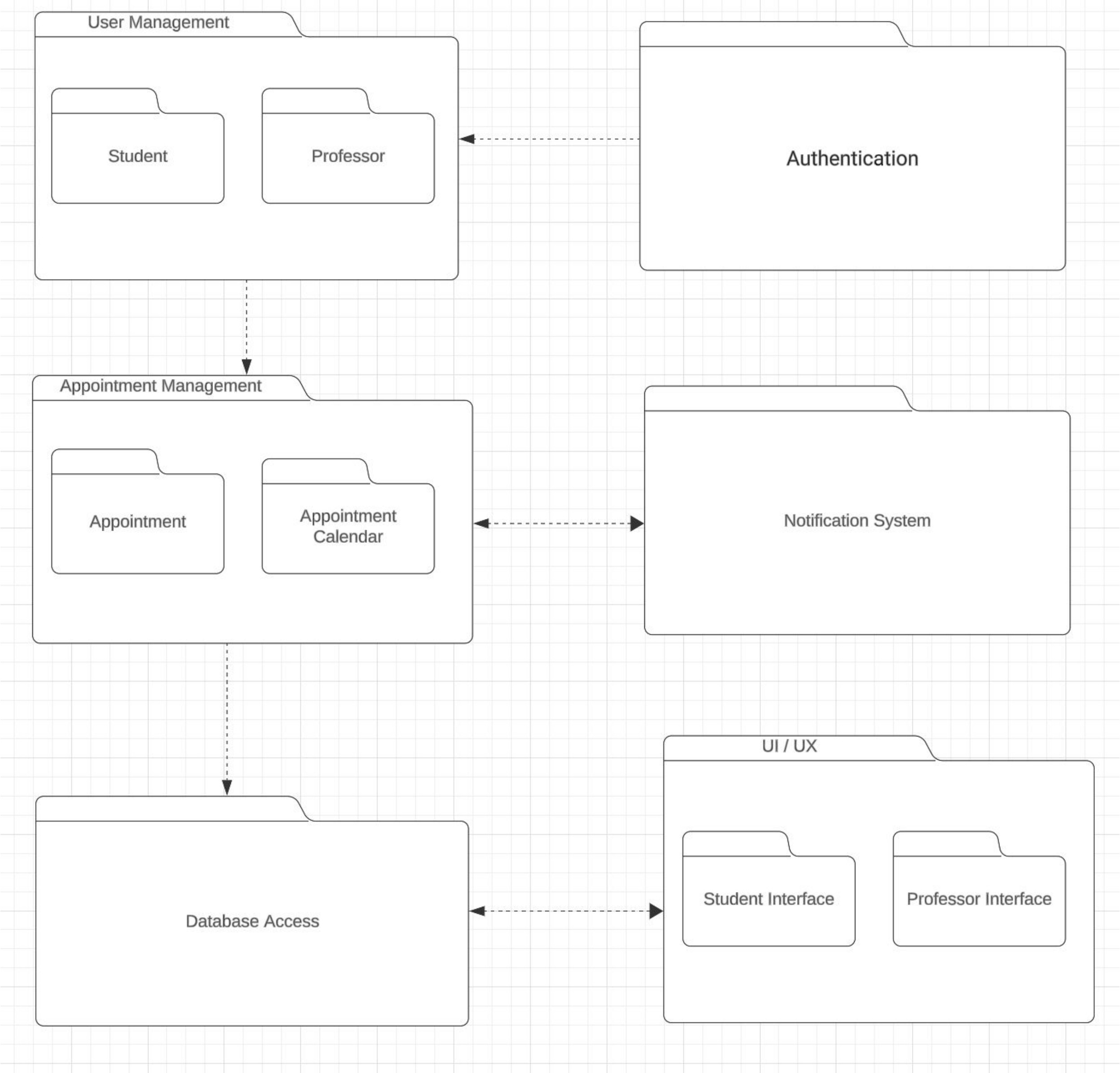


3.3) Activity Diagram

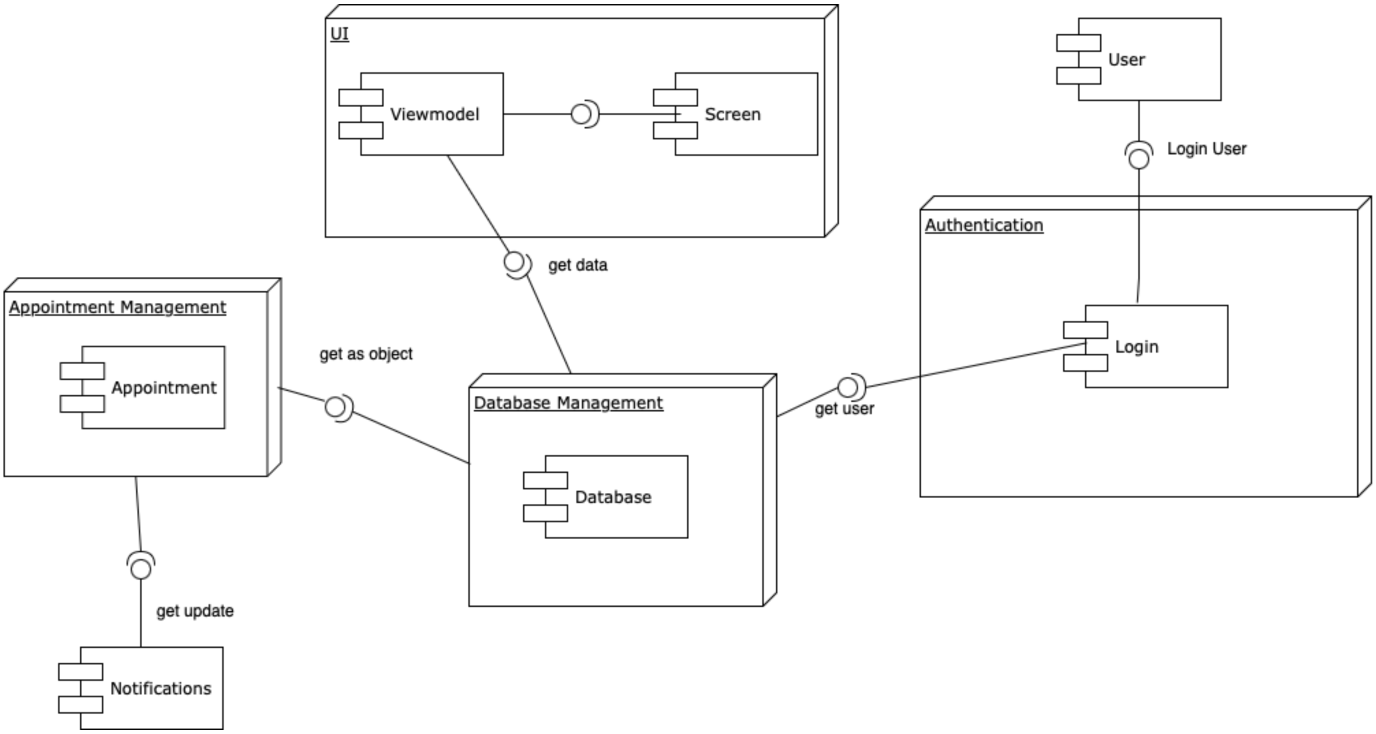


4) SOFTWARE ARCHITECTURE

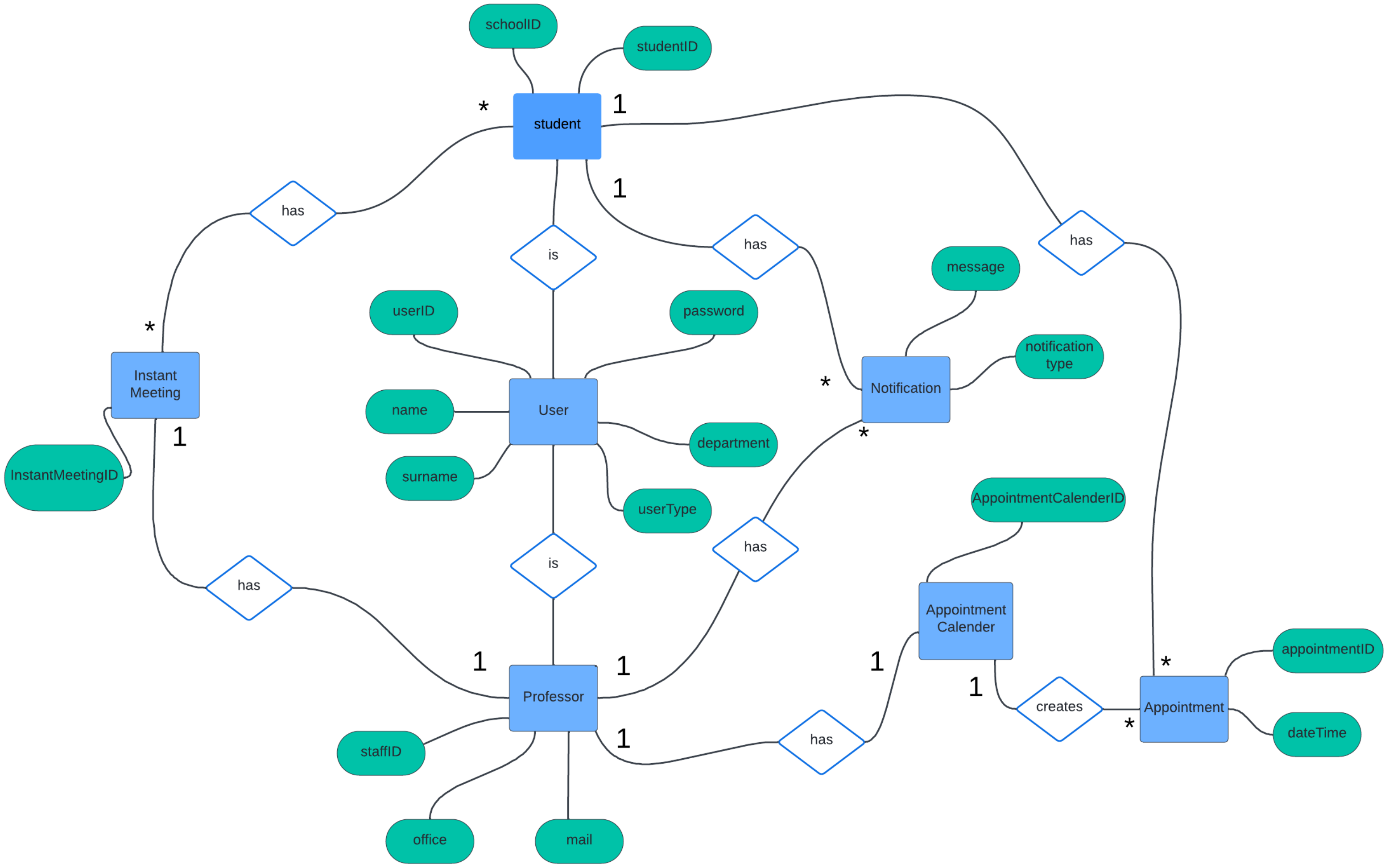
4.1) UML Package Diagram



4.2) UML Component Diagram



5) ENTITY RELATIONSHIP DIAGRAM



6. Glossary and References

6.1. Glossary

Activity Diagram: an activity diagram provides a view of the behavior of a system by describing the sequence of actions in a process.
Aggregation: an aggregation relationship shows a classifier as a part of or subordinate to another classifier.
Associations: is a relationship between two classifiers, such as classes or use cases, that describes the reasons for the relationship and the rules that govern the relationship.
Attribute: An attribute describes a range of values for that data definition. Attributes describe the structure and value of an instance of a class.
Component Diagram: Component diagram breaks down the actual system under development into various high levels of functionality. Each component is responsible for one clear aim within the entire system and only interacts with other essential elements on a need-to-know basis.
Multiplicity: Multiplicity in UML allows to specify cardinality -number of elements - of some collection of elements.
Package Diagram: is used as a central view to represent a software system compile-time logical architecture.
Retrieve: Searching for, locating, and returning data.
Sequence Diagram: Sequence Diagrams are interaction diagrams that detail how operations are carried out.
State Diagram: State diagrams to illustrate the dynamic view of a system. They are especially important in modeling the behavior of an interface, class, or collaboration.

6.2. References

1.Use Case Specification Guideline – Best Tips & Guidance for 2023 – Business Analyst Mentor: https://businessanalystmentor.com/use-case-specification-guidelines/

2.Visual Paradigm International. (n.d.). Free Class Diagram Tool. Retrieved March 28, 2023,from/https://online.visual-paradigm.com/diagrams/solutions/free-class-diagram-tool/

3.Figma: https://www.figma.com/file/IaVnFHV7N0hzAtHUEARFRz/CSE344-Project?type=design&node-id=0-1&mode=design&t=KFbRripkCa9bGKvB-0

4.Draw.io:https://app.diagrams.net/#G1pHcvFi9DQkICMOT3tRsra-x0RBE6hNbj%23%7B%22pageId%22%3A%22C5RBs43oDa-KdzZeNtuy%22%7D

5.Lucidchart.app:https://lucid.app/lucidchart/16d38b0a-2ae0-459c-95d7-afb6a67d8c1e/edit?invitationId=inv_e1f043f9-ed9d-4548-8bfc-072804dc571c&page=0_0#