**Software Architecture Specification**

March 25, 2020

Team 4 (Mariem Bchir, Liam Carriker, Joe Kane, Mary Lasater, Lina McDermott, Caroline Sigl)

1. System and Architectural Context

1.1. Rationale

The project’s architecture is largely defined by module views and component and connector views. The architecture utilizes the client-server style. The client-server style best suits this project as it allows the client to make requests to the server. Among the considered and rejected styles are the shared data style and the publish-subscribe style. Though the system contains a data repository and data accessors, client-server better suits the project than the shared data style. This is because the client-server style emphasizes the calls the client makes to the server, which we prefer over the two-way communication used in the shared-data style. The publish-subscribe style was also rejected, as this style has the server serve more as a broadcasting object rather than as a way to access the database to pull specific information.

1.2. Scope

This document covers the software architecture specifications for the entirety of the first version (pilot) of a completely new application. The developers’ main concerns in regards to this document are feasibility and clarity of the system. The developers wish for the system to be readable and maintainable for any future development. The primary concern of any users of the system is the reliability of the system.

1.3. Definitions, Acronyms, and Abbreviations

C&C- Component and Connector View

IP - Internet Protocol

UDP - User Datagram Protocol

UI - user interface

Activity - main Android application interface component

Fragment - an independent Android component that may be used by an Activity

ViewModel - Android class designed to store and display UI data

1.4. Behavior

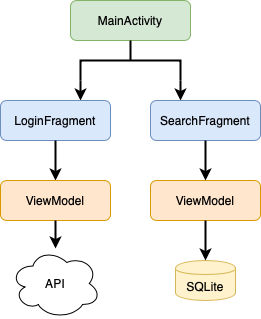
Within the system, the client makes requests to the server to access information stored in the database. The server accesses the database and communicates the information back to the client. From an Android perspective, the system utilizes one MainActivity which depends on Fragments and ViewModels to access and display information.

2. Architecture Views

2.1. Views

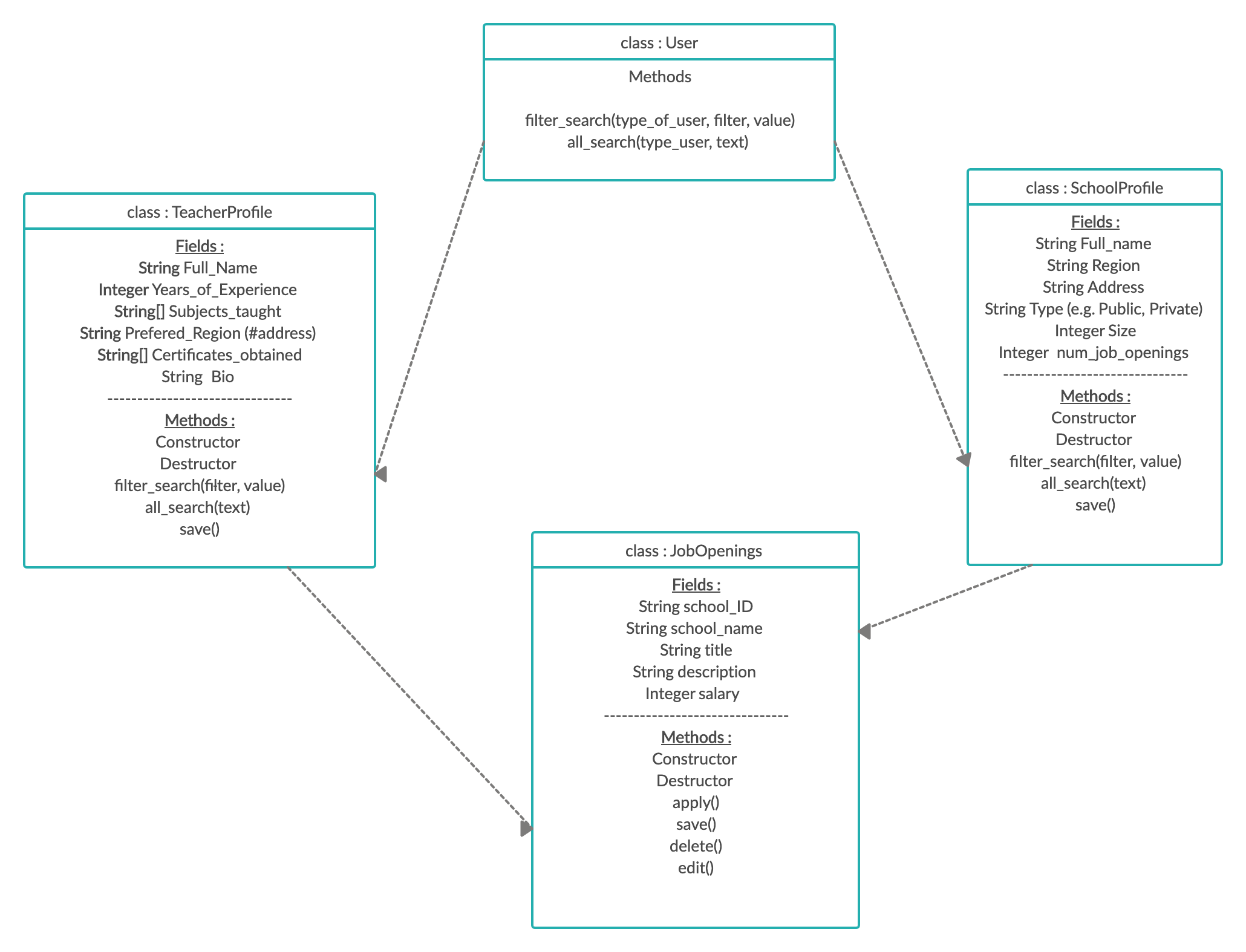
2.1.1. Module Views

2.1.1.1. Android Module View



In the diagram above, the Android Module View for the system is depicted. The application will consist of one main activity, which depends on two fragments. Both fragments will depend on ViewModels. In both cases, there may be multiple ViewModels for the different modes of the application (school user and teacher user). However, regardless of the mode, the Login ViewModel(s) will depend on an API to authenticate users. Additionally, regardless of the mode, the Search ViewModel(s) will depend on SQLite to show search results.

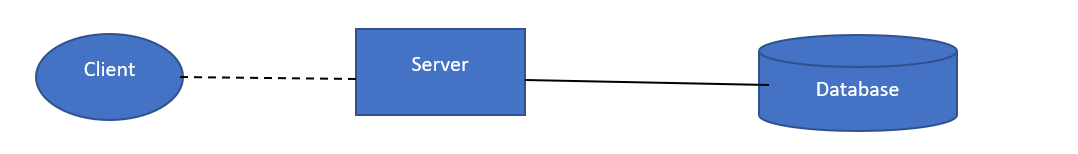
2.1.1.2. Class Module View



In the diagram above, a class diagram is depicted. This is the system’s second module view.

2.1.2. C&C Views

2.1.2.1. Main Components

In the diagram above, we have identified the three main components of our application and how they connect to each other. The client is connected to the server by a request-reply connector and the server is connected to the database by a database access connector. 

2.2. Element Catalog

2.2.1. Client- the Android application client. Makes requests to the server for any information needed.

2.2.2. Server- communicates with the database. Retrieves information requested by the client.

2.2.2.1. At the moment, the data format that will be used is unknown.

2.2.2.2. Since the amount of data that we will be transferring is very small there are fewer chances for packets to get lost. Hence we are using IP-UDP as our internet protocol. In case a packet is dropped the client may make the request again without much cost.

2.2.3. Database- stores the information on schools and teachers. Uses SQLite.

2.2.4. MainActivity - the main Android Activity that will open when the application opens. Depends on Fragments and ViewModels to display information.

* The ViewModels will depend on the various classes depicted in the diagram in section 2.1.1.2.

3. Across Views Description

3.1. Views

3.1.1. Android Module View

3.1.1.1. The ViewModels use the connectors specified in the C&C View to access data in the database.

3.1.1.2. The ViewModels access information from the TeacherProfile and SchoolProfile classes.

3.1.2. Class Module View

3.1.2.1. User class used in the Client component.

3.1.2.2. TeacherProfile and SchoolProfile used in Server and Database components.

3.1.2.3. JobOpenings class used in the database server.

3.1.3. C&C View

3.1.3.1. Client is the Android MainActivity and related fragments.