# 

# CSC 431 FundsBook System Architecture Specification (SAS)

**Team 16**

|  |  |
| --- | --- |
| Luis Diaz | Programmer |
| Temuulen Ganbold | Programmer |
| Julio Ojalvo | Programmer |

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change Comments |
| 1 | 3/31/2021 | Luis Diaz, Temuulen Ganbold, Julio Ojalvo |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 

# 

# 

# 

# 

# 

# Table of Contents

[1.](#_3dy6vkm) System Analysis 6

[1.1](#_1t3h5sf) System Overview 6

[1.2](#_4d34og8) System Diagram 6

[1.3](#_2s8eyo1) Actor Identification 6

[1.4](#_17dp8vu) Design Rationale 6

[1.4.1](#_3rdcrjn) Architectural Style 6

[1.4.2](#_26in1rg) Design Pattern(s) 6

[1.4.3](#_lnxbz9) Framework 6

[2.](#_35nkun2) Functional Design 7

[2.1](#_1ksv4uv) Diagram Title 7

[3.](#_44sinio) Structural Design 8

[4.](#_2jxsxqh) Behavioral Design 9

# Table of Tables

<Generate table here>

# Table of Figures

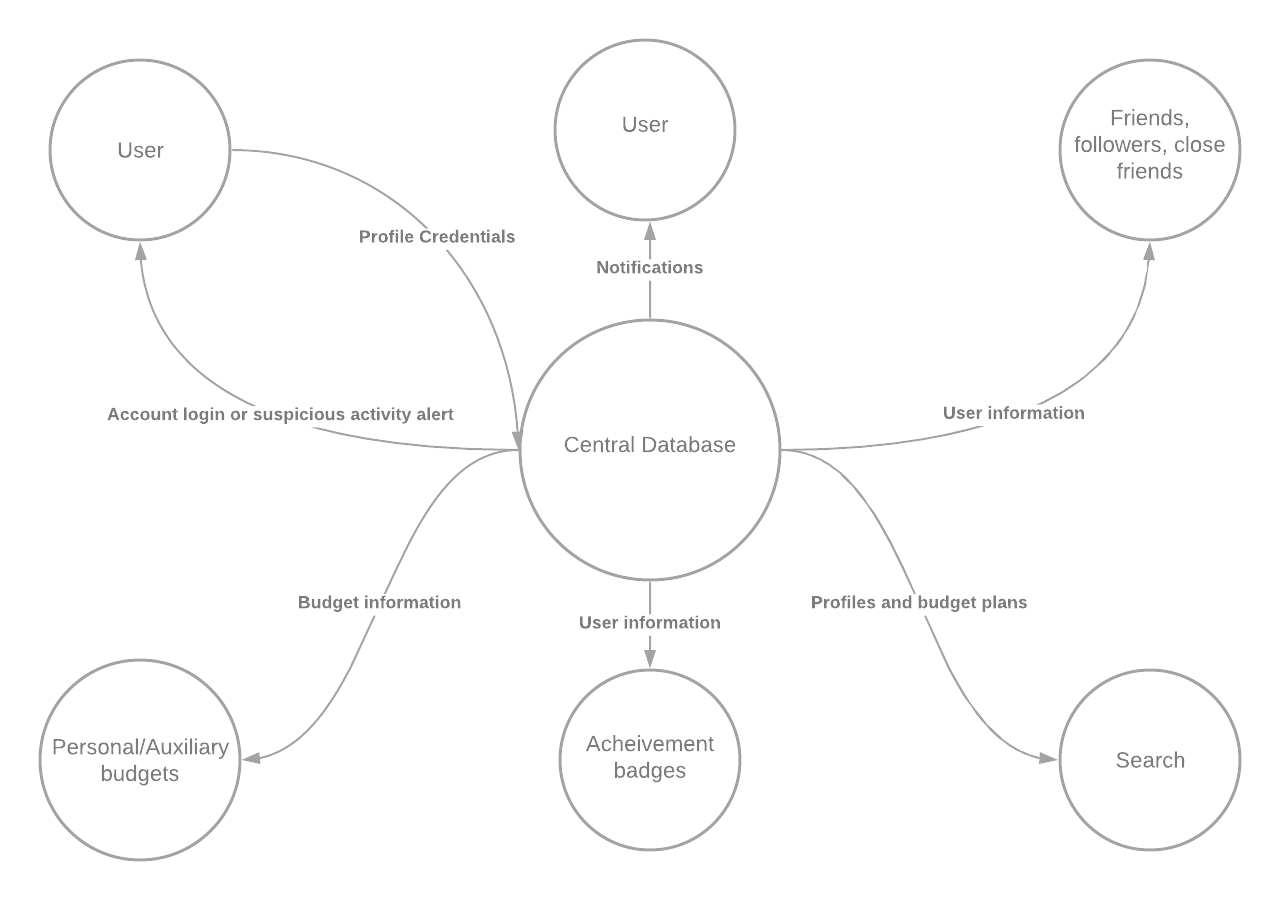
<Generate table here>

### 1. System Analysis

#### 1.1 System Overview

Our system will be a database-centric one where a centralized database containing the user’s profile, budget plans, and pertaining information will be accessible to other components and services.

#### 1.2 System Diagram

**

#### 1.3 Actor Identification

* Users
* Central database system
* client and server
* Devices
  + Any android devices
  + Any IOS device

#### 1.4 Design Rationale

##### 1.4.1 Architectural Style

This will be a database-centric architecture style. A central database will store all necessary information except private banking information which will be stored locally. All functional and non-functional requirements will access this central database if necessary, for their process.

##### 1.4.2 Design Pattern(s)

**Creational Design Pattern**

* Factory

**Structural Design Patterns**

* Facade
* Adapter
* Decorator

**Behavioral Design Patterns**

* Command

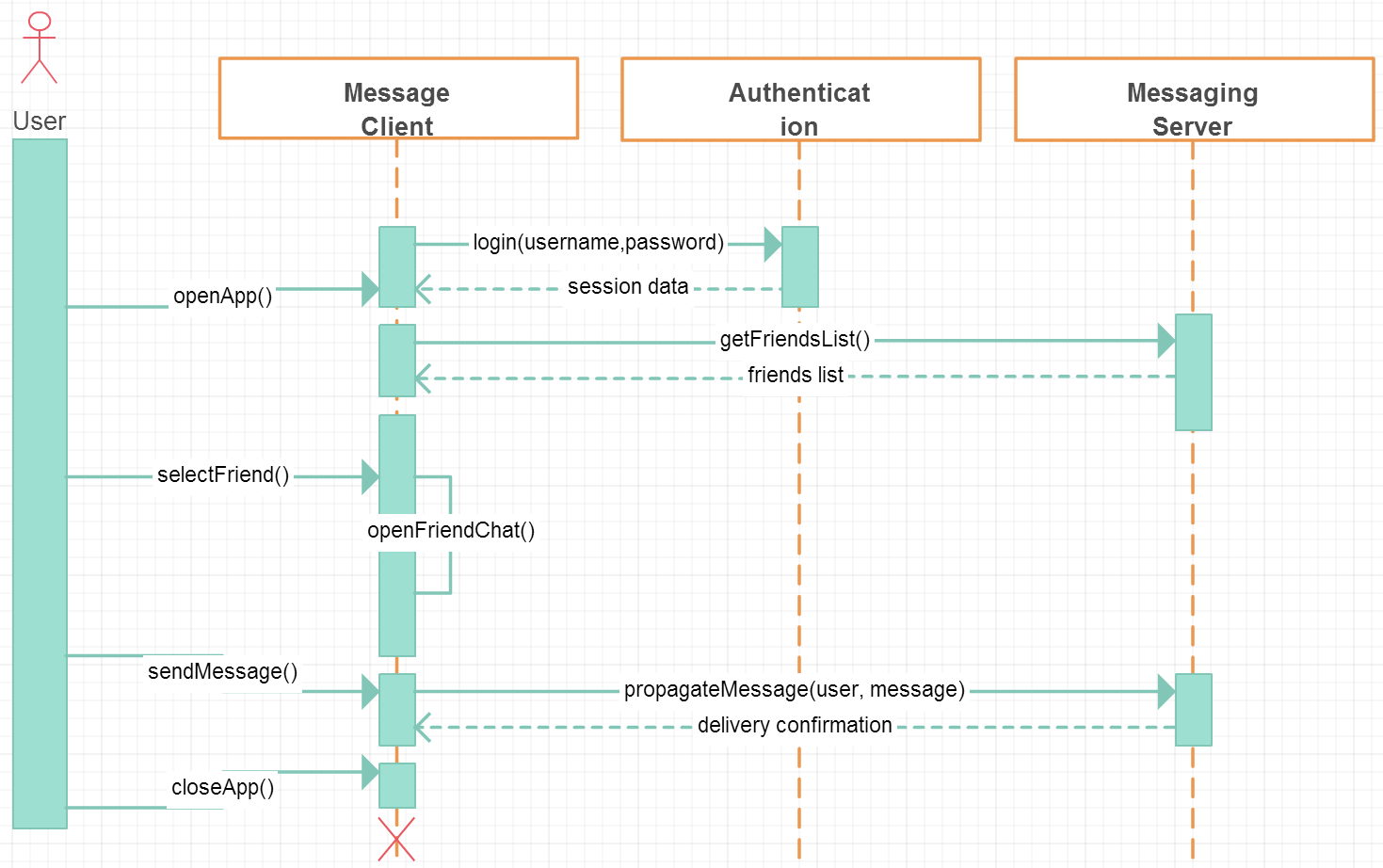
##### 1.4.3 Framework

If we were to use any framework, I believe that SPRING: Java + Hibernate + MySQL/PostgreSQL, would be the best applicable framework to our application. Using this framework would allow us to create our central database and allow us to query any information needed as it is requested by the user. Spring would also allow us to have a more secure application as its authentication feature allows us “to integrate with industry-standard security schemes and deliver trustworthy solutions that are secure by default”, which is essential in our app as there can be instances where a user may link their bank accounts to application.

### 

### 2.Functional Design

#### 2.1 Diagram Title



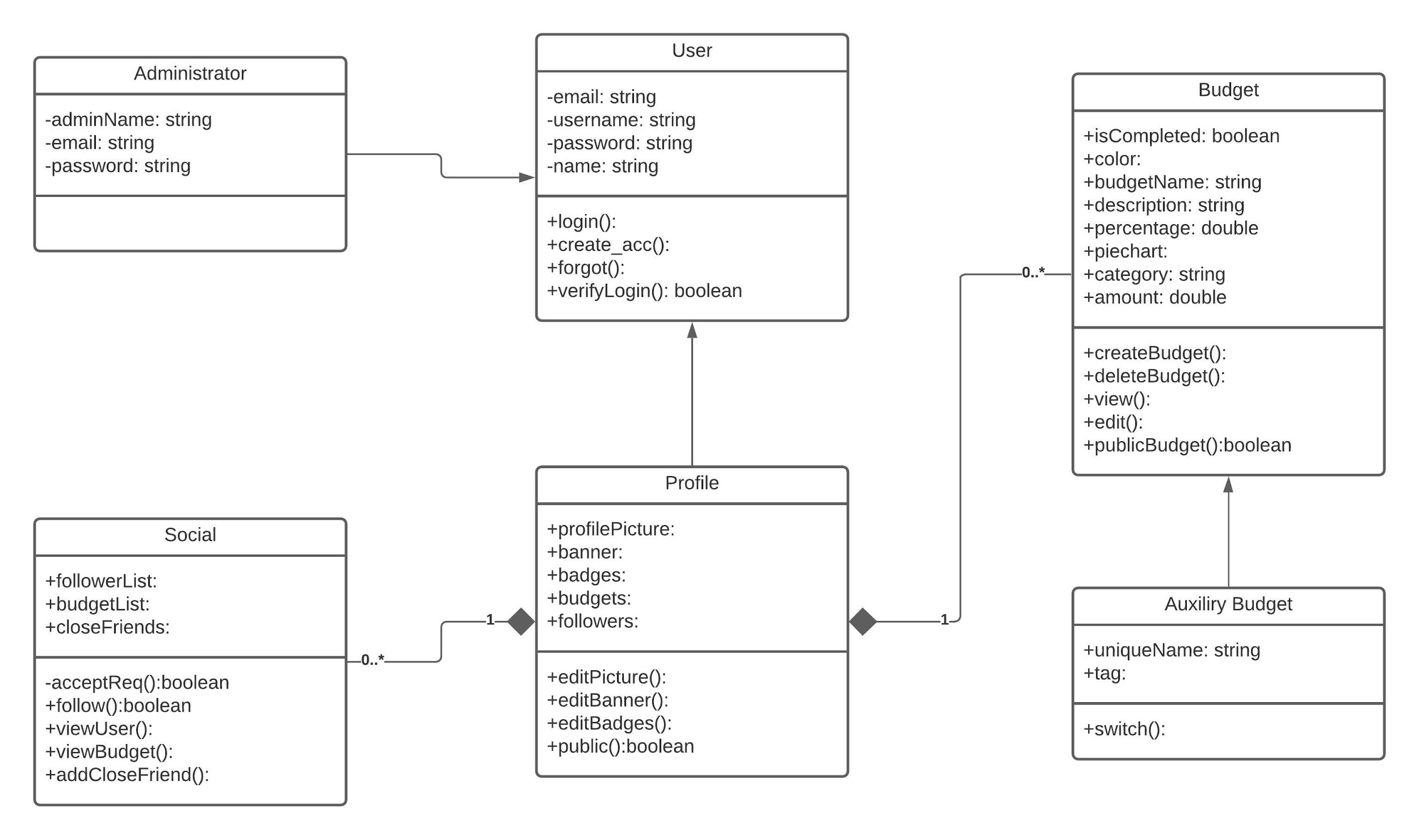
### 

### 

### 

### 3. Structural Design

Following is our class diagram representing our system:



### 4. Behavioral Design