Class User - class representing the data stored in the database

Public Data Members -

name

email

Password

DOB

Gender

Public Methods - User constructor with all values

Class UserDatabase - class handling database operations

Public Method -

void createUser(const User& user) - create a user in the database

DefaultDataPopulator - class populating default data for a user

Public Method -

void populateDefaultData(const User& user) - Implementation to populate default data for the user

DirectoryManager - class handling directory creation

Public Method -

void createUserDirectory(const User& user) - Implementation to create a directory for the user

NotificationService - class handling user notifications

Public Methods -

void notifySuccess(const User& user) - Implementation to send success notification

void notifyError(const User& user, const std::string& errorMessage) - Implementation to send error notification

SignupService - class for the signup process

Private Data Members -

UserDatabase userDatabase

DefaultDataPopulator dataPopulator

DirectoryManager directoryManager

NotificationService notificationService

Public Methods -

void signup(const std::string& name, const std::string& email) - Creating a new user

-------> Flow of SignUP API

User newUser( name, email, password);

try {

// User Addition in Database

userDatabase.createUser(newUser);

// Populate default data

dataPopulator.populateDefaultData(newUser);

//Create user Directory

DirectoryManager.createUserDirectory(newUser);

// Notify user of success

notificationService.notifySuccess(newUser);

} catch (const std::exception& e) {

// Notify user of error

notificationService.notifyError(newUser, e.what());

}

FLOW -

SignupService signup\_service;

signup\_service.signup("John Doe", [john@example.com](mailto:john@example.com), “mypassword”);