Design Document for <<PENGU>>

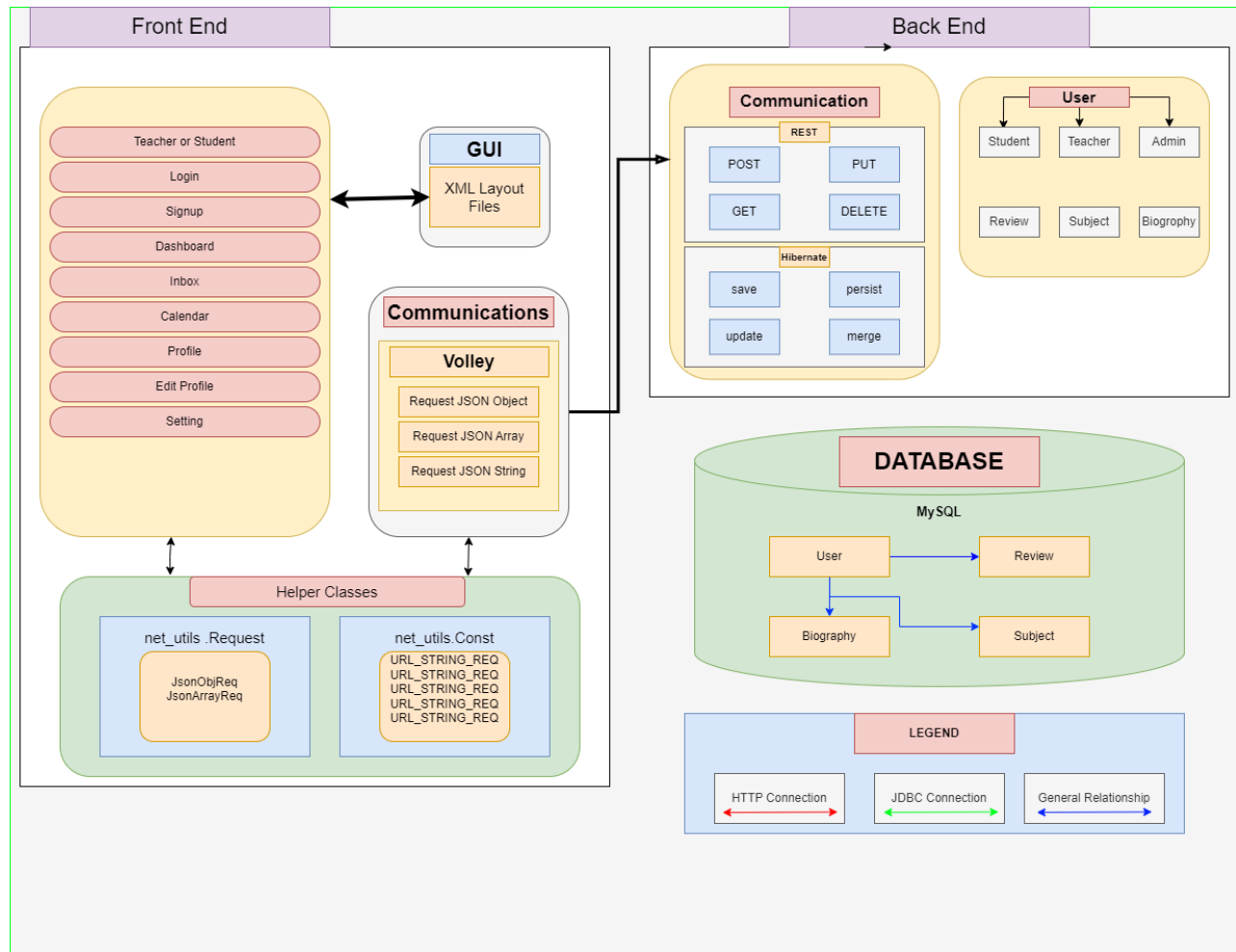Group <MS3_5>

Selim Erenay: % 25

Merve Erenay: % 25

Haad Cheema: % 25

Ahmet Okur: % 25

Block Diagram

Front End

Teacher or Student
- Login
- Signup
- Dashboard
- Inbox
- Calendar
- Profile
- Edit Profile
- Setting

GUI
XML Layout Files

Communications
Volley
- Request JSON Object
- Request JSON Array
- Request JSON String

Helper Classes

net_utils .Request
JsonObjReq
JsonArrayReq

net_utils.Const
URL_STRING_REQ
URL_STRING_REQ
URL_STRING_REQ
URL_STRING_REQ
URL_STRING_REQ

Back End

Communication

REST
- POST
- PUT
- GET
- DELETE

Hibernate
- save
- persist
- update
- merge

User
- Student
- Teacher
- Admin
- Review
- Subject
- Biogrophy

DATABASE
MySQL
- User
- Review
- Biography
- Subject

LEGEND
- HTTP Connection
- JDBC Connection
- General Relationship

Frontend (currently implemented)

SignIn (User)

• Create account generates a page with the following elements:

o EditText: UserID

o EditText: Password

o EditText: Email

o Button: CreateAccount

• Upon clicking the button 'CreateAccount' the values of the UserID, Password, and Email are sent as a POST request to the server.

Login (Admin, User, Business)

• Create account generates a page with the following elements:

o EditText: UserID

o EditText: Password

o Button: Login

• Login Screen takes user input of their username and password, and if that username and password is valid, it will take the user to the main menu, where they are logged in as that user.

AccountList (Admin)

• AccountList generates a page with the following elements:

o Button: Account List

o Table: Accounts

• Loading the this page or clicking the button 'Account List' completes a GET request from the server into the dynamic table 'Accounts'

Backend

The backend uses mappings to update the database based on information sent to the given mappings' URLs.

which include:

• Post: send information on an item to be added to the database.

• Get: request information, often with an identifier for the specific item requested from the database

• Put: send information to update a specific item in the database

• Delete: send an identifier to delete a specific item from the database

Controllers

The controllers contain the mappings for communication between frontend and the database.

These include:

• User: Contains the above mappings to create users, which contain one-to-one
relationships with the below student, teacher, and admin type users.

• Student: Students have one to one relationship with teachers and are able to write reviews or biography.
They also have the function to login as well as any user.

• Teacher: Teachers are given reviews from the students. so they have a one to one relationship with
students.

• Admin: This has one to one connections with students, and teachers as well as the access to all parts of the
tables which are the reviews, subjects, biography. Only admins can create subjects at the moment.

Table Relationships

user
- 🔑 id INT
- ◇ password VARCHAR(255)
- ◇ email_id VARCHAR(255)
- ◇ name VARCHAR(255)
- ◇ role INT
- Indexes ▶

biography
- 🔑 id INT
- ◇ age INT
- ◇ bio VARCHAR(255)
- ◇ image VARCHAR(255)
- ◇ user_id INT
- Indexes ▶

subject
- 🔑 subjectid INT
- ◇ subject_name VARCHAR(255)
- Indexes ▶

review_id
- 🔑 id INT
- ◇ actual_review VARCHAR(255)
- ◇ written_by_id INT
- ◇ written_for_id INT
- Indexes ▶