

**Student:** Gabriel Fernandez, Florida International University  
**Mentors:** Mohsen Taheri, Vladan Lavolic, Ranjeet Deshmukh  
**Instructor:** Masoud Sadjadi, Florida International University

## System Design



## Problem

**Life is complex:** full of problems & tasks that we must solve simultaneously, and it can get overwhelming. This is true in particular for students, who must sometimes juggle with project deadlines, studying, work, family, and any number of other activities.

## Solution

**Neat** is an application that aims to make it easier to manage life's problems through the tried and true Agile principles.

With neat, people can categorize their goals & list the tasks required to fulfill them in a timely manner. In return, neat provides valuable progress feedback through smart algorithms.

### My core tasks for neat 1.0:

- Develop a secure & efficient backend to store & manipulate data
- Develop a REST API for data transfer
- Create an UI/UX mockup for future frontend development

## Current System

**Status:** The backend is live on an AWS server, fully accessible from a browser. The iPhone application runs natively in the device, is fully connected to the backend, and has many features implemented such as login, view dashboard, and add assignments

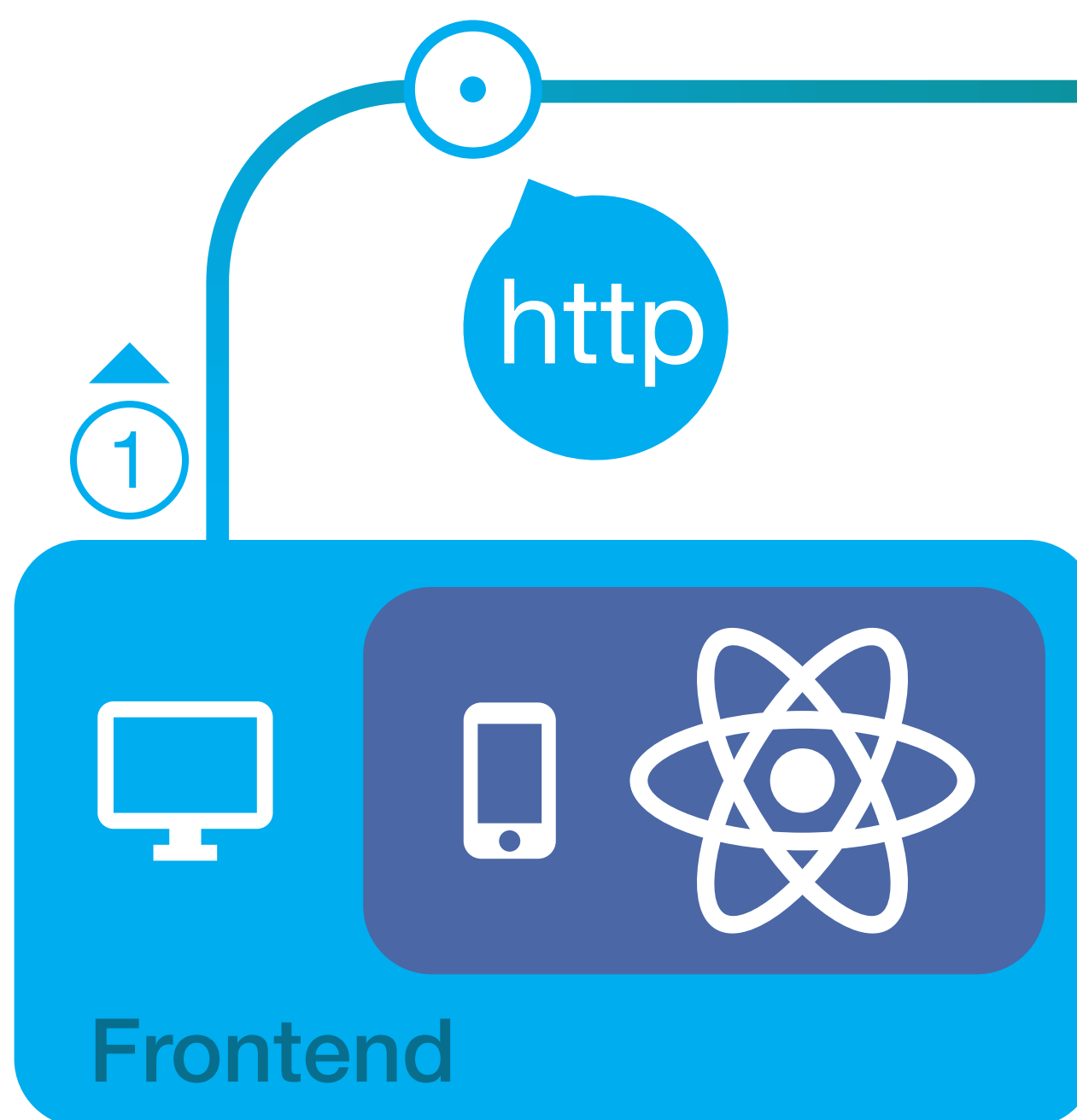
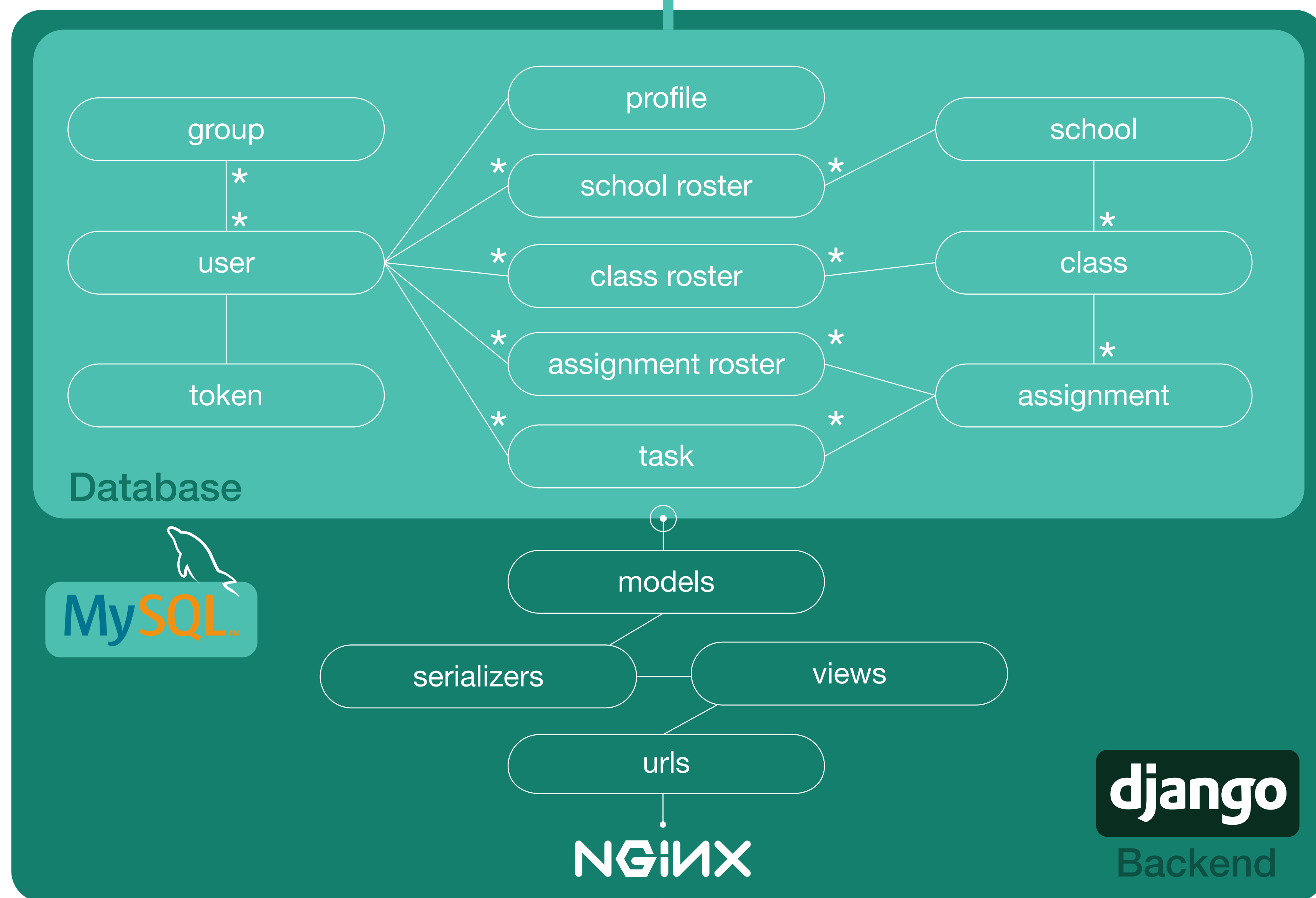
### My contributions to the current backend system:

- Token & session authentication
- Passwords securely stored with PBKDF2 / SHA256 hash
- E-mail verification & e-mail password reset support
- Filtering & automatic user data retrieval based on token
- REST API with endpoints to get, add, & update models
- Smart progress algorithm

## Acknowledgement

The material presented in this poster is based upon the work supported by **Mohsen Taheri, Vladan Lalovic, & Ranjeet Deshmukh**. I am helpful for the help received from my team members:

- Fernando Campo
- Nelson Cruz
- Pierre Jimenez
- Pachev Joseph
- Justin Lopez
- Giselle Pacheco



**Validation** is done by the backend. If the requests are invalid or cause data integrity exceptions, proper error messages are returned in the response.

### Sample valid response:

```
{
  "url": "http://52.87.176.128/api/assignment/1/",
  "name": "Science Fair Project",
  "startDate": "2016-11-20T17:05:05.174682Z",
  "dueDate": "2016-11-23T00:00:00Z",
  "classFK": "http://52.87.176.128/api/class/1/",
  "expected": 0.7187178949952164,
  "progress": 0.6666666666666666,
  "smart status": "Slightly Behind"
}
```

## Mockups

