Testr

Project Demo and Design Pattern Analysis

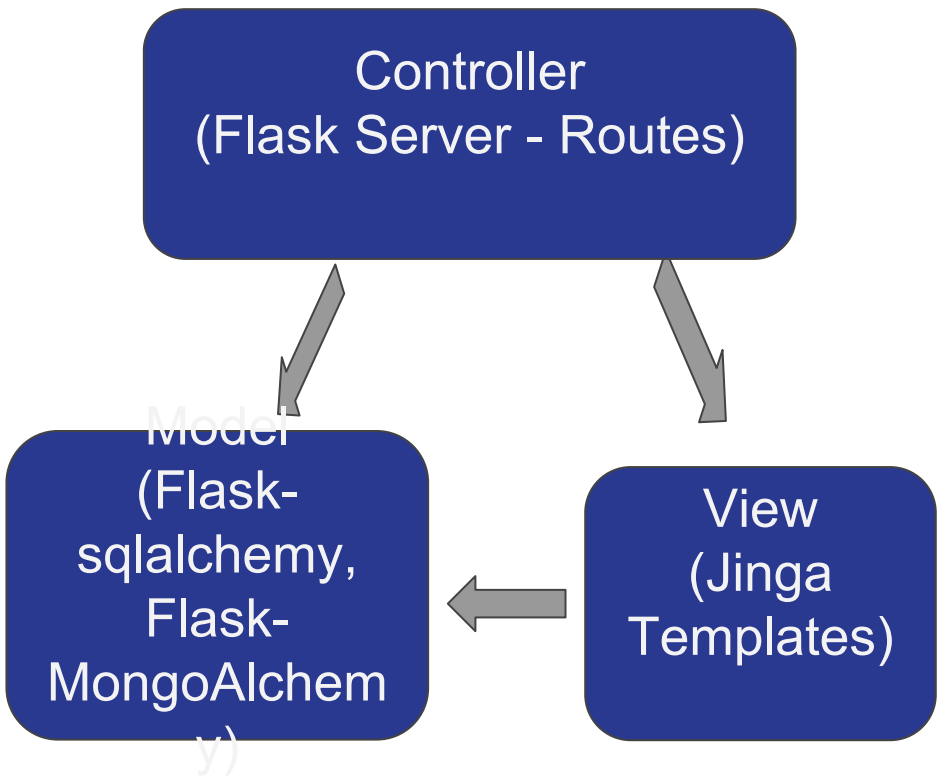
Karthik Kannan
Mahesh Kumar Ravindranathan
Nikhil Mahendra

Project Demo

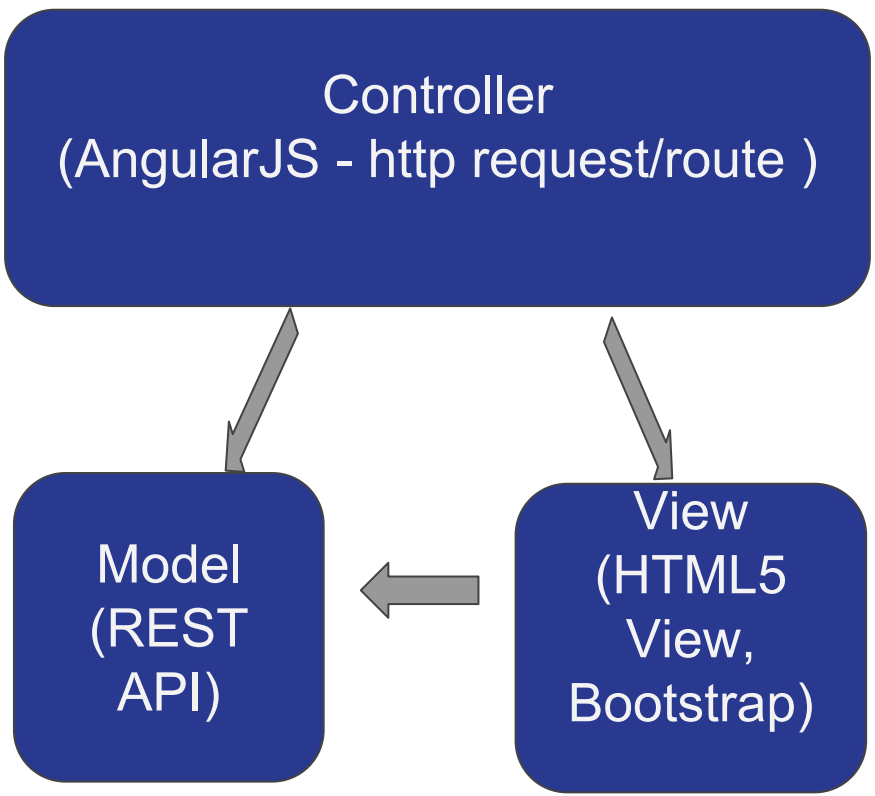
Technologies Used

- Python Flask for Backend
- AngularJS for Frontend
- MySQL Database for storing user credentials
- MongoDB for storing the application data
- Jinga based Templates
- Bootstrap for CSS
- Triangle Library to co-ordinate between the templates created using Jinga with Angular
- Flask-sqlalchemy and Flask-MongoAlchemy for ORM

Architectural Pattern - Model View Controller (MVC), REST API



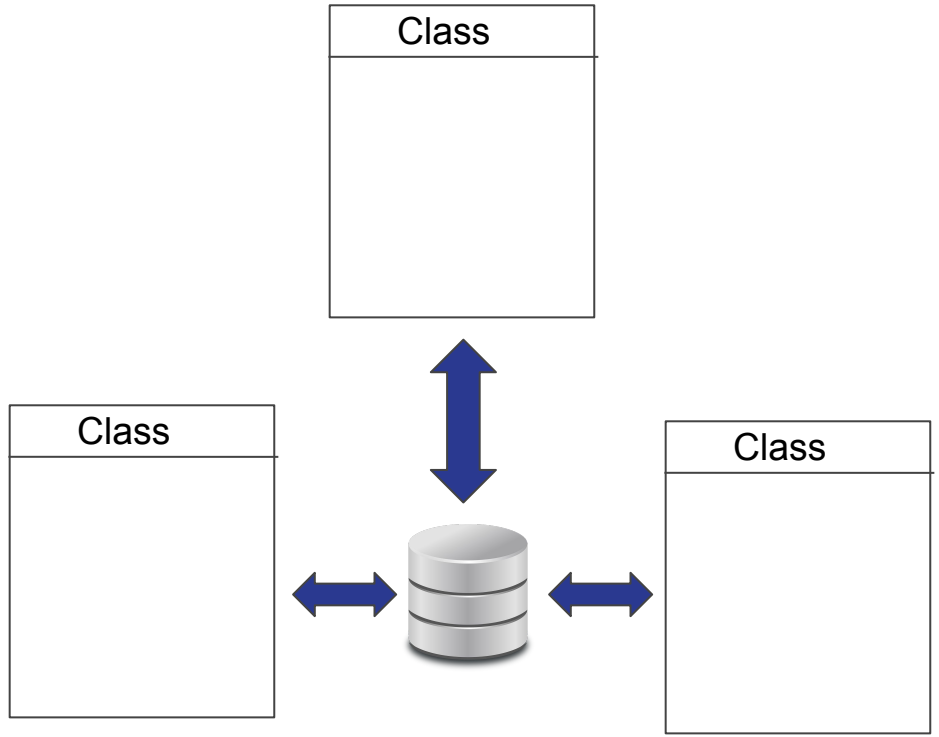
Server Side MVC



Client Side MVC

ORM

- To map tables to classes in our code
- ORM framework (SQLAlchemy)
- ODM framework (MongoAlchemy)
- Decouples the object model and database schema from the beginning.



Design Patterns

Factory Method

- Creation of Questions.
- Since every question can have different type of answers.
- The type of answer defines the kind of options provided for question.

Strategy

- Auto-grading.
- Every answer has a different condition for it being correct.
- The strategy patterns abstracts the details of grading.

Observer Pattern

- Student notification mechanism
- Notify the student that only few minutes are left for the test to end.

What we learnt

- Initially thought in terms of processes(sequence diagrams)
- Class diagrams makes it easier to visualize how components interact with each other.
- Design patterns are a really helpful guide to good practices.
- Using design patterns in Python isn’t as straightforward

Thank You