

Pro-Quest Services

Backend-Frontend Architecture

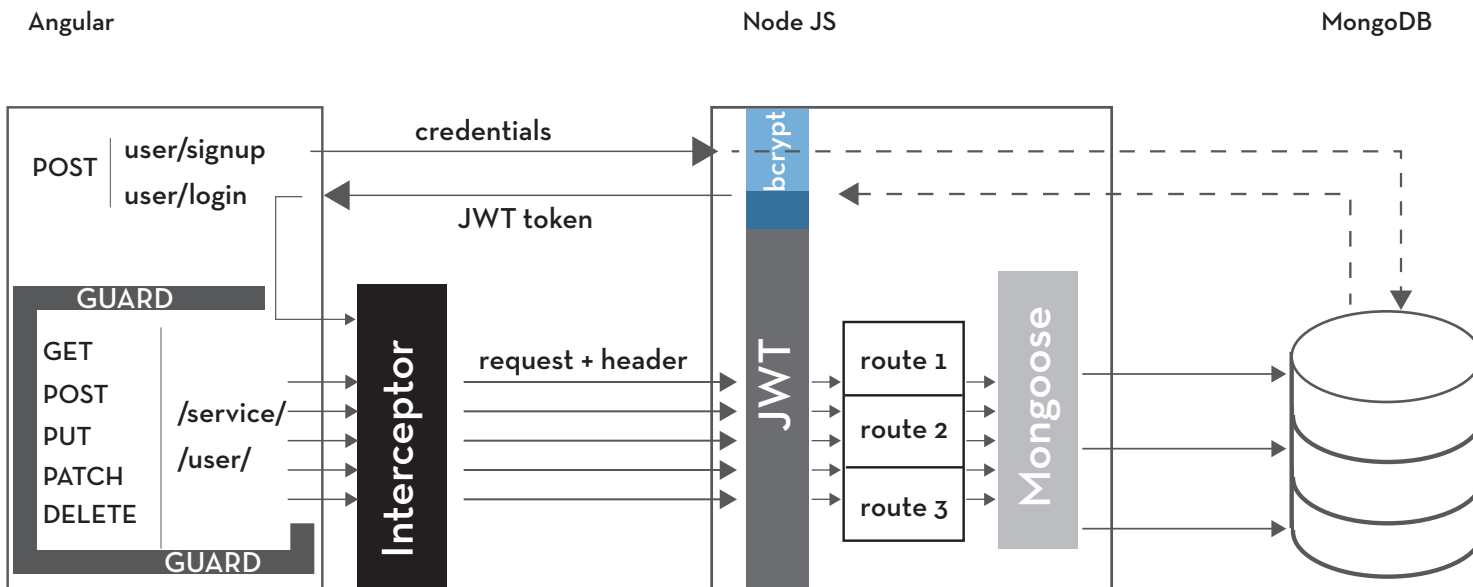
Mintesinot Gebre
Dec. 15, 2021

Pro-Quest Services

Frontend-Backend Integration

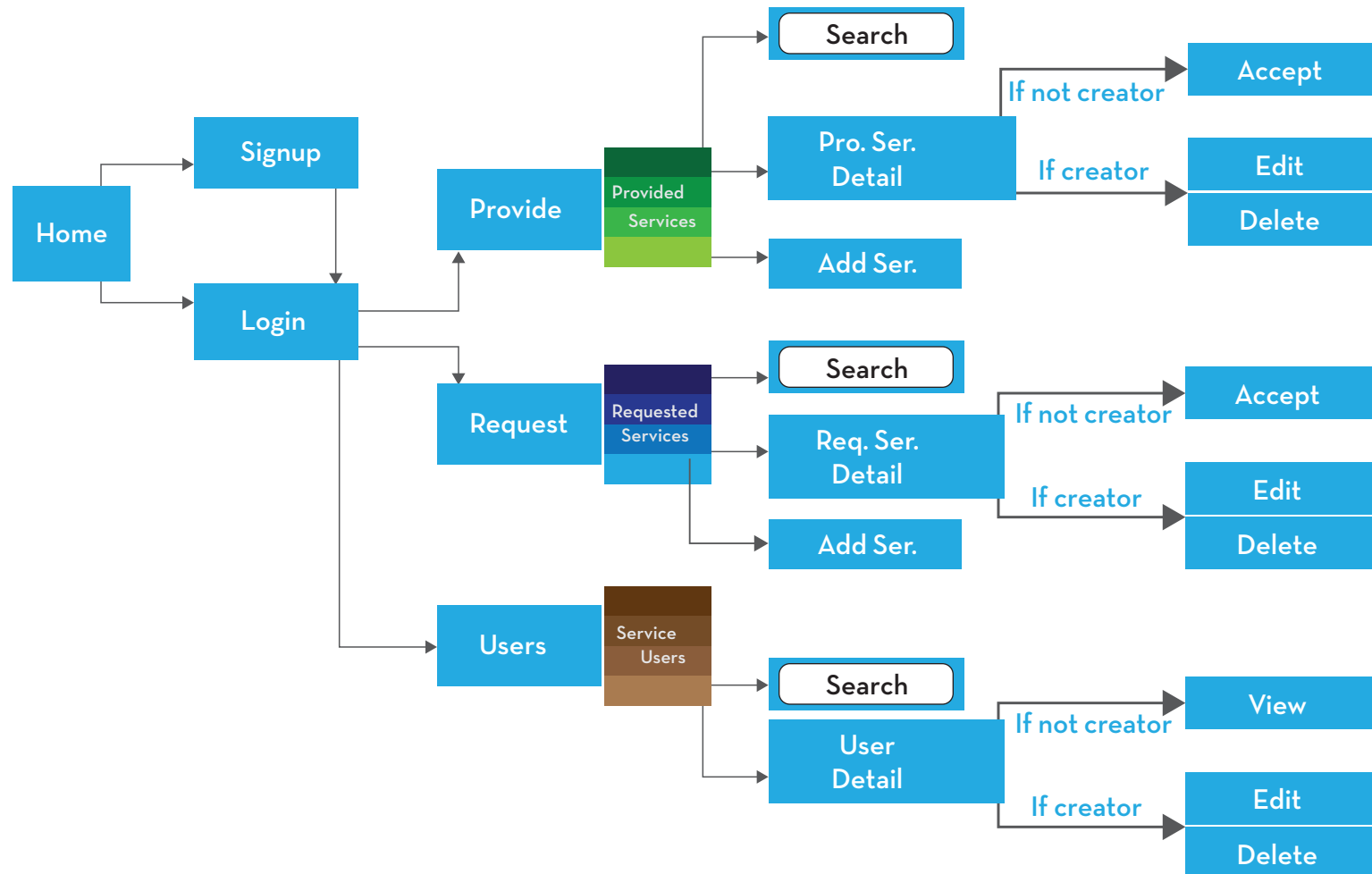
Frontend

Backend



Pro-Quest Services

Frontend

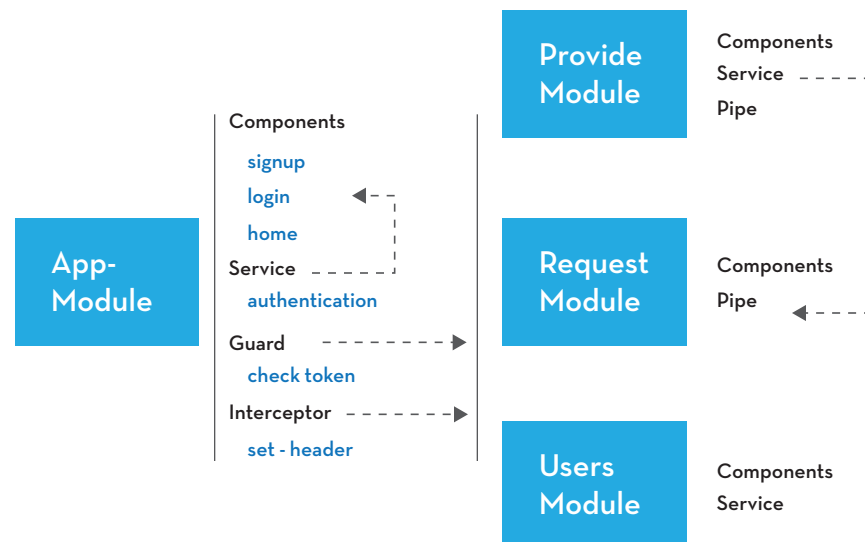


Pro-Quest Services

Frontend

root

children



Pro-Quest Services

Backend

“User” Route

Routes

POST /users/signup

*accepts user info & registers new users

POST /users/login

*accepts user login info sends jwt token

Get /users

*returns all users

Get /users/:email

*returns user with given email

Delete /users/:email

*removes user, with given email, from database

PUT /users/:email

*updates user, with given email, from database

User Schema

```
const UserSchema = new Schema({
  firstname: { type: String, required: true },
  lastname: { type: String, required: true },
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true },
  phone: { type: String, default: "*****" },
  address: {
    street: String,
    city: String,
    state: String,
    zipcode: String,
    country: String,
  },
  provided: { type: Array, default: [] },
  requestd: { type: Array, default: [] },
  accepted: { type: Array, default: [] },
  createdAt: {
    required: true,
    type: Date,
    default: Date.now(),
  },
});
```

Backend

“Service” Route

Routes

- **POST /services/:userId**
*accepts service info, registers service, updates creator user
- **POST /services/:id/comment**
*accepts comment & updates service
- **Get /services**
*returns all services
- **Get /services/provide**
*returns all provided-services
- **Get /services/request**
*returns all requested-services
- **Get /services/:id**
*returns service with given id
- **Get /services/:id/comments**
*returns array of comments of service with given id
- **Delete /services/:id**
*removes service using the given id
- **PUT /services/:id**
*updates service info
- **PUT /services/:userId/:serId**
*updates services and user, when user provides/requests
- **PATCH /services/:fullname**
*updates services, when user updates name

Service Schema

```
firstname: { type: String, required: true },
lastname: { type: String, required: true },
creatorId: { type: Number },
service: {
  title: { type: String, required: true },
  detail: { type: String, required: true },
  category: { type: String, required: true },
},
type: {
  type: String,
  enum: ["request", "provide"],
  required: true,
},
fulfilled: { type: Boolean, default: false },
hourlyPayment: { type: Number, required: true },
location: {
  city: { type: String, required: true, default: "Laurel" },
  coords: {
    latitude: { type: String, required: true },
    longitude: { type: String, required: true },
  },
  timestamp: { type: String, required: true },
},
comment: { type: Array, default: [] },
createdAt: {
  required: true,
  type: Date,
  default: Date.now(),
},
});
```