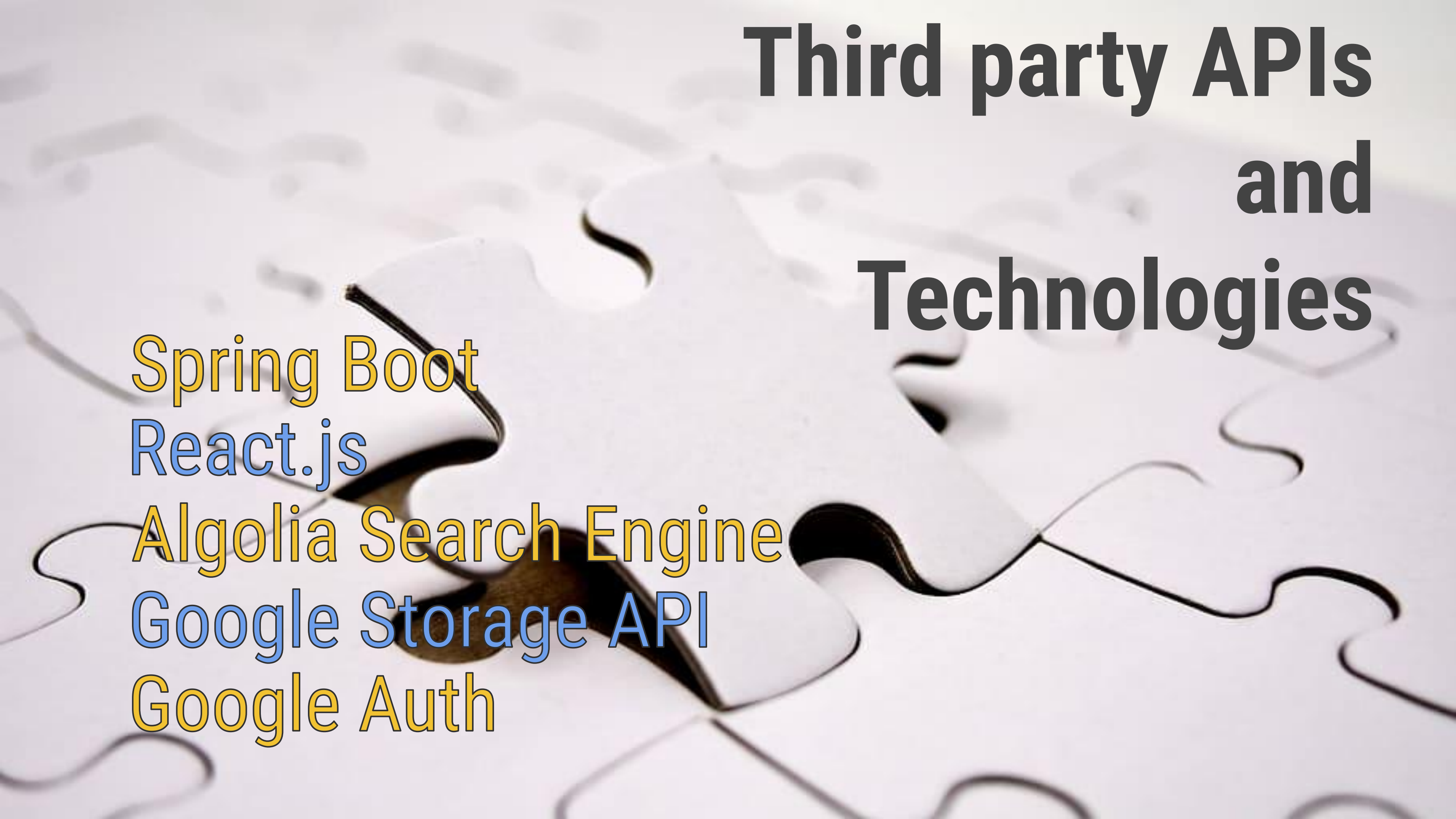
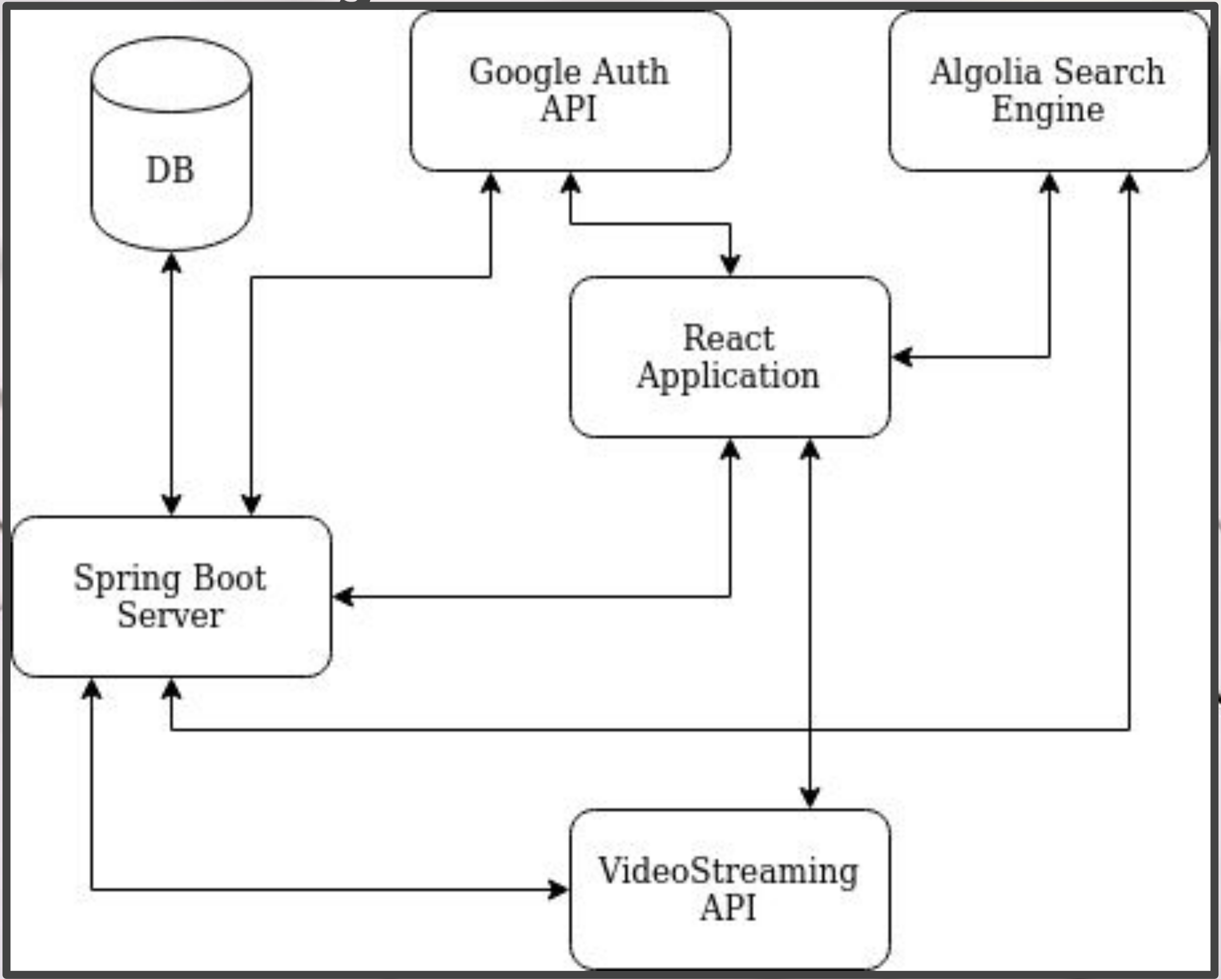
Third party APIs and Technologies

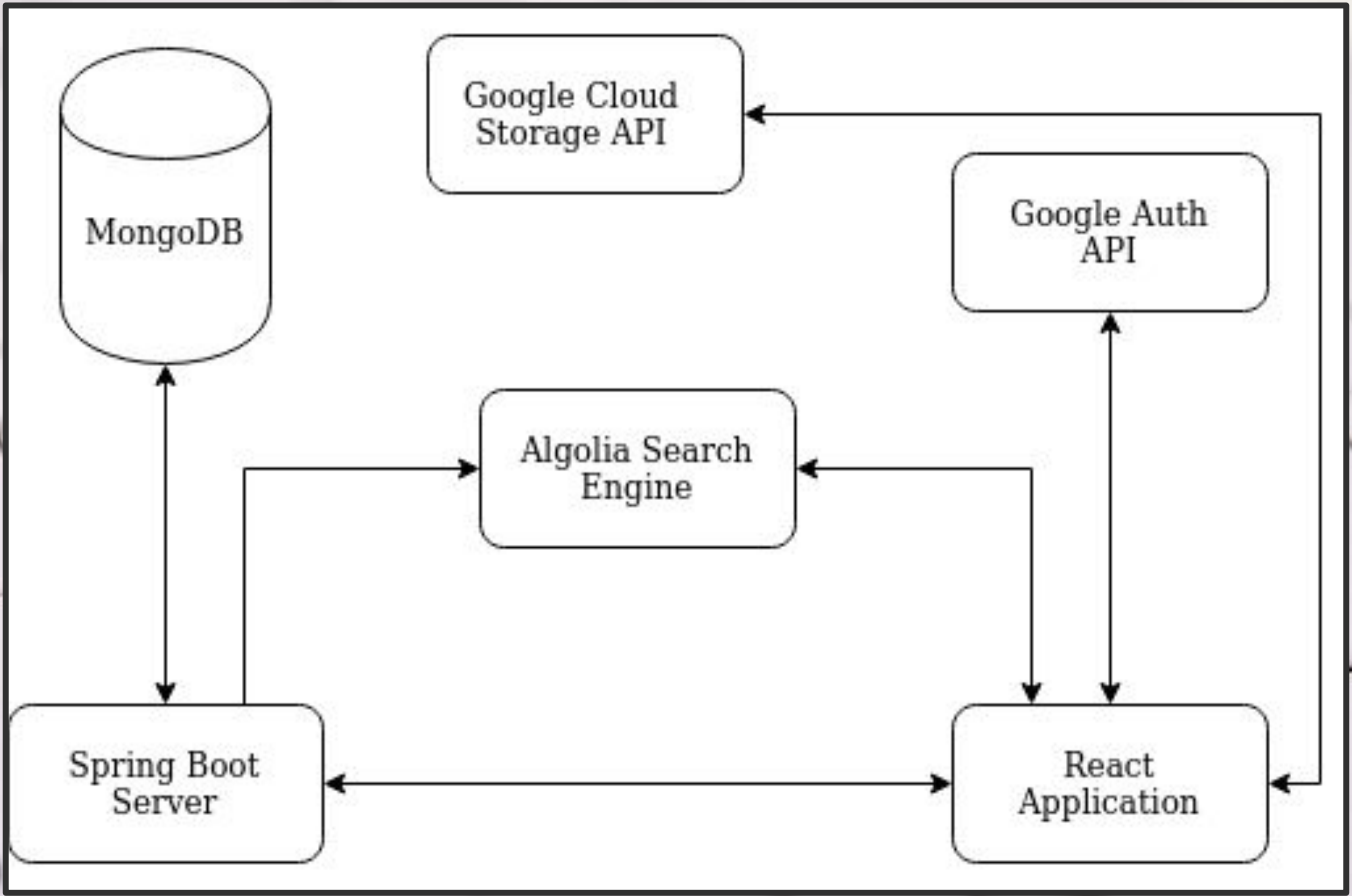


Spring Boot
React.js
Algolia Search Engine
Google Storage API
Google Auth

Original Architecture



Final Architecture - (MVC)



Technically Complex Problems

Real time search

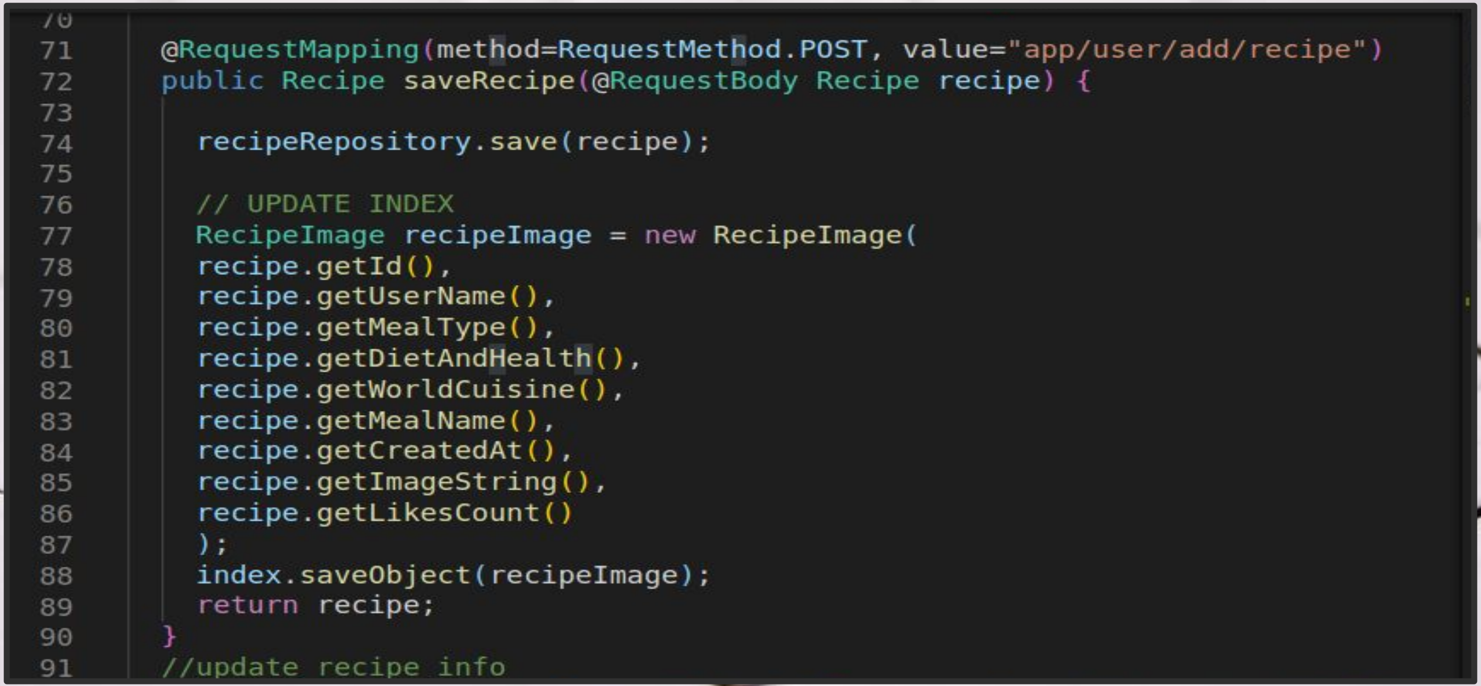
Saving and accessing images with URL links

Solutions

Algolia Search

Google Cloud Storage

Indexing Recipes To Algolia

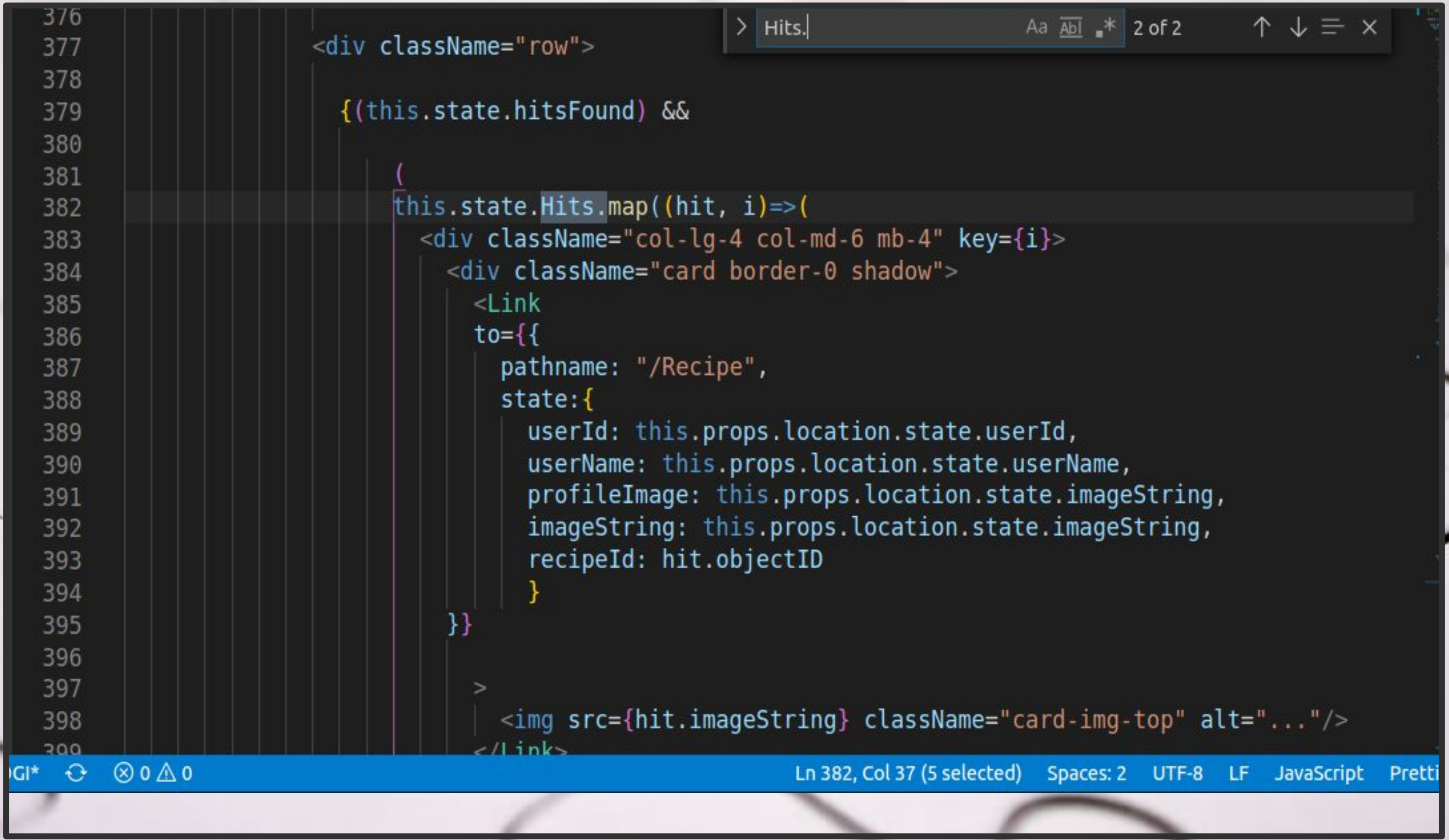


```java
@RequestMapping(method=RequestMethod.POST, value="app/user/add/recipe")
public Recipe saveRecipe(@RequestBody Recipe recipe) {

  recipeRepository.save(recipe);

  // UPDATE INDEX
  RecipeImage recipeImage = new RecipeImage(
  recipe.getId(),
  recipe.getUserName(),
  recipe.getMealType(),
  recipe.getDietAndHealth(),
  recipe.getWorldCuisine(),
  recipe.getMealName(),
  recipe.getCreatedAt(),
  recipe.getImageString(),
  recipe.getLikesCount()
  );
  index.saveObject(recipeImage);
  return recipe;
}
//update recipe info
```

Rendering Hits from Algolia



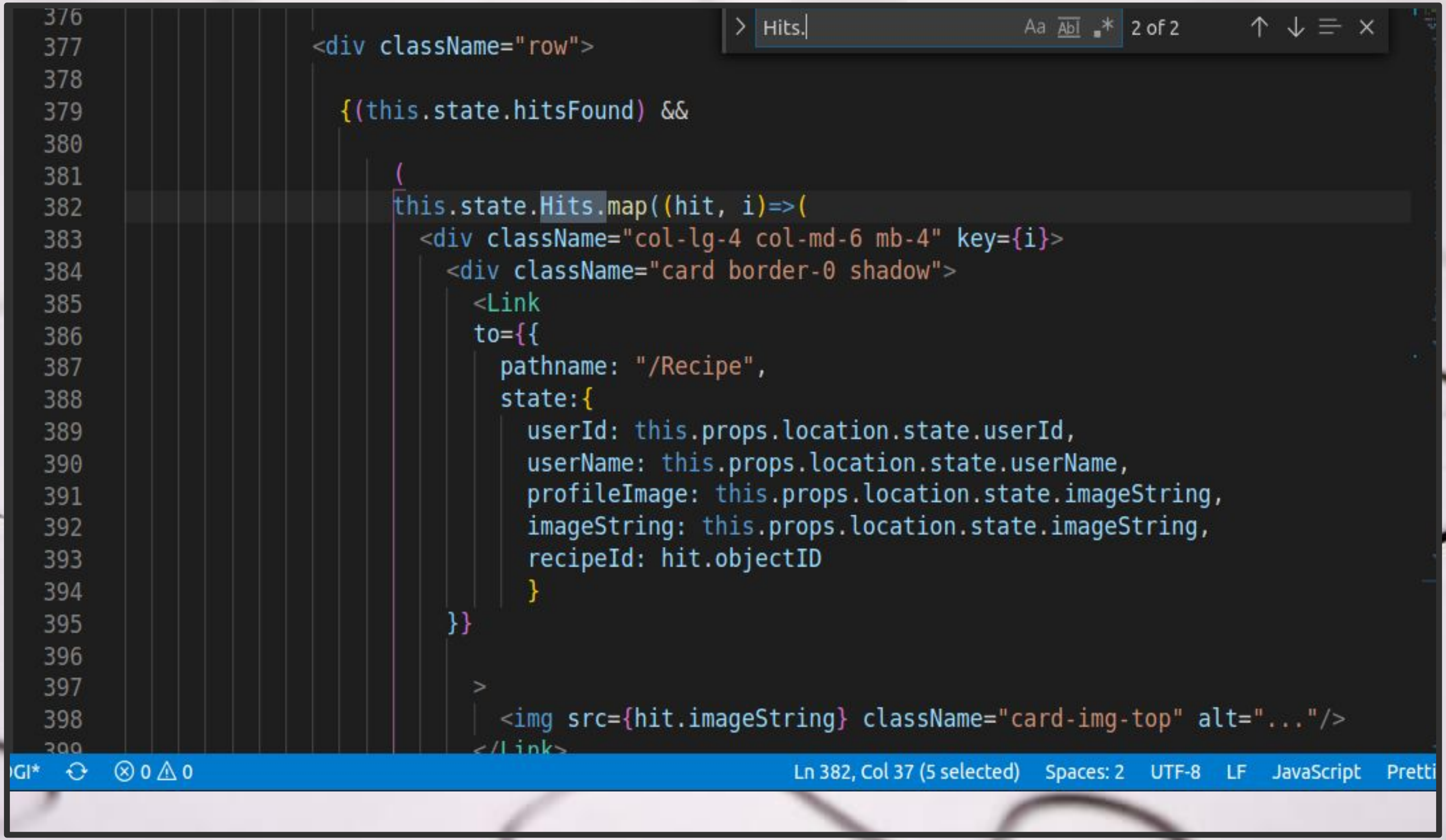
```
            <div className="row">

              {(this.state.hitsFound) &&

                (
                this.state.Hits.map((hit, i)=>(
                  <div className="col-lg-4 col-md-6 mb-4" key={i}>
                    <div className="card border-0 shadow">
                      <Link
                      to={{
                        pathname: "/Recipe",
                        state:{
                          userId: this.props.location.state.userId,
                          userName: this.props.location.state.userName,
                          profileImage: this.props.location.state.imageString,
                          imageString: this.props.location.state.imageString,
                          recipeId: hit.objectID
                          }
                    }}

                      >
                        <img src={hit.imageString} className="card-img-top" alt="..."/>
```

Saving Image To Firebase and Fetching URL



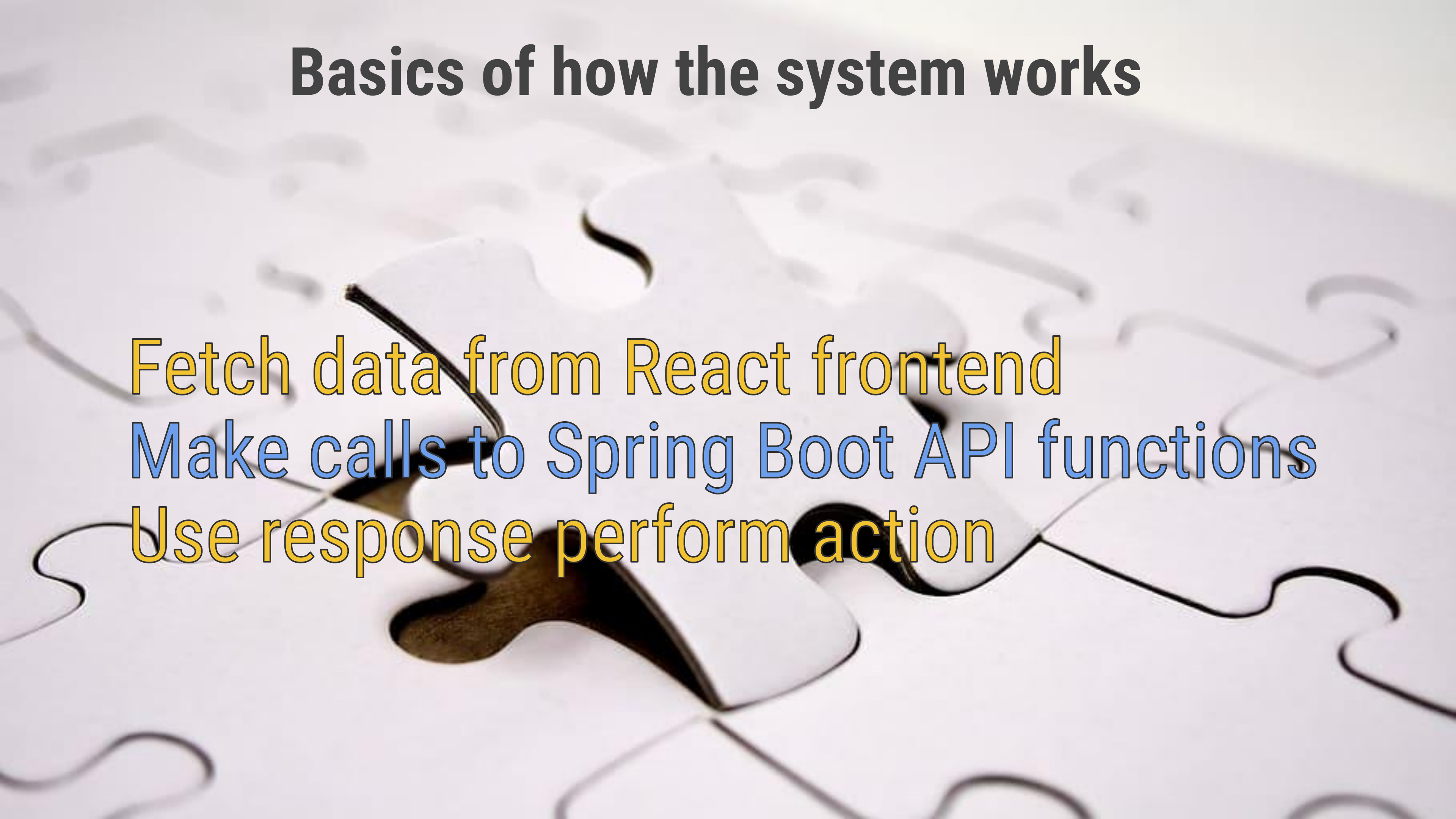
```
            <div className="row">

              {(this.state.hitsFound) &&

                (
                this.state.Hits.map((hit, i)=>(
                  <div className="col-lg-4 col-md-6 mb-4" key={i}>
                    <div className="card border-0 shadow">
                      <Link
                      to={{
                        pathname: "/Recipe",
                        state:{
                          userId: this.props.location.state.userId,
                          userName: this.props.location.state.userName,
                          profileImage: this.props.location.state.imageString,
                          imageString: this.props.location.state.imageString,
                          recipeId: hit.objectID
                          }
                    }}

                      >
                        <img src={hit.imageString} className="card-img-top" alt="..."/>
```

Basics of how the system works

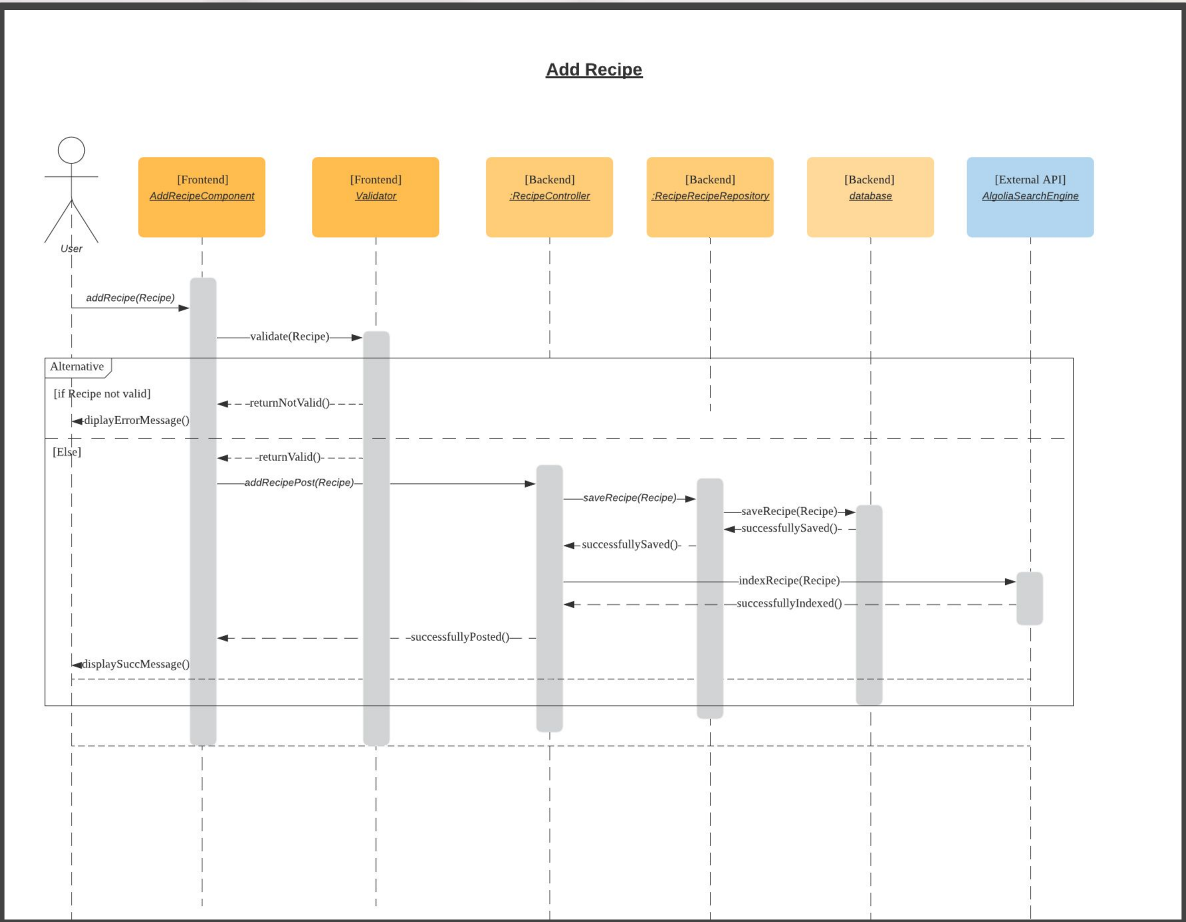


Fetch data from React frontend

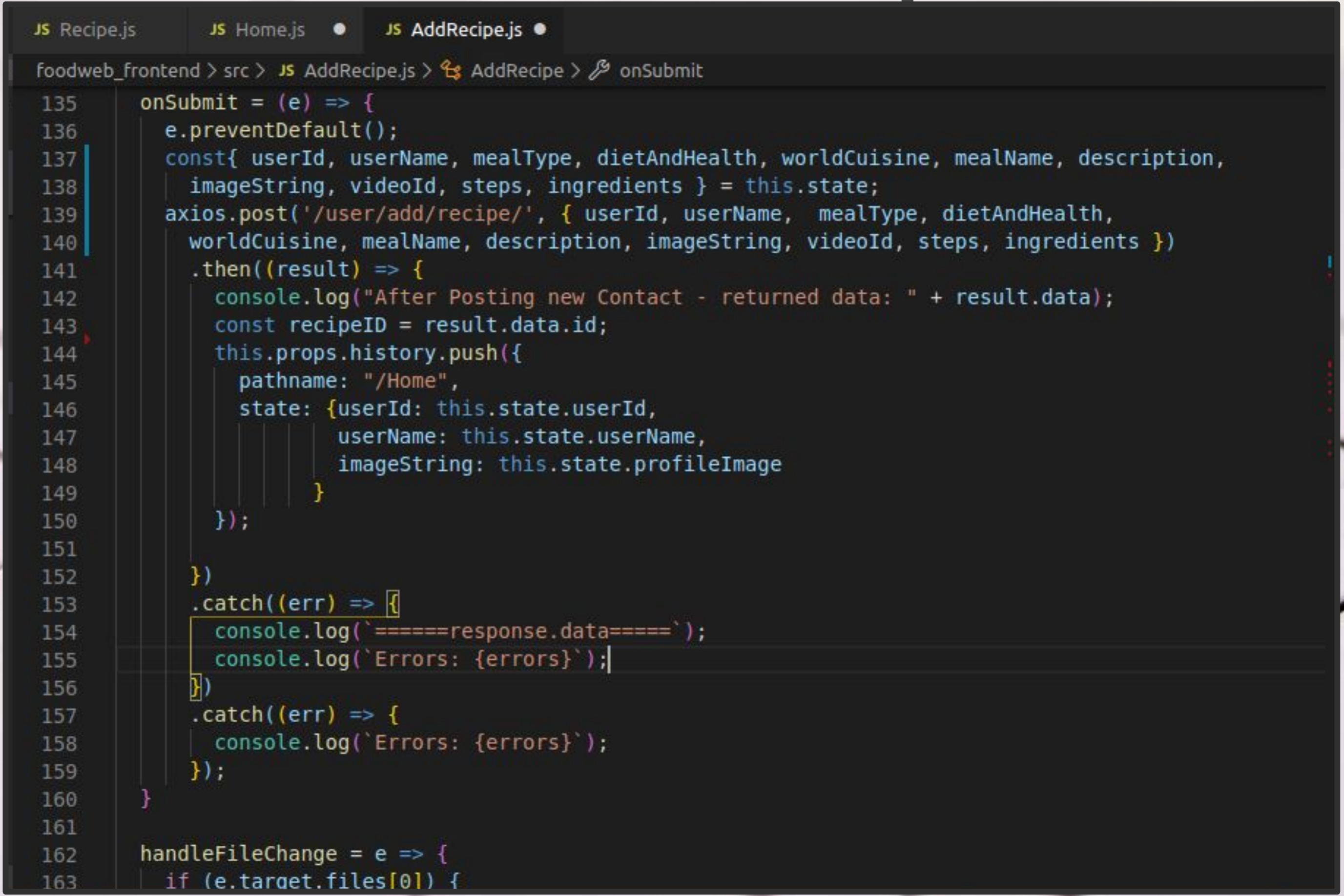
Make calls to Spring Boot API functions

Use response perform action

Example: Sequence Diagram



Frontend: Add Recipe

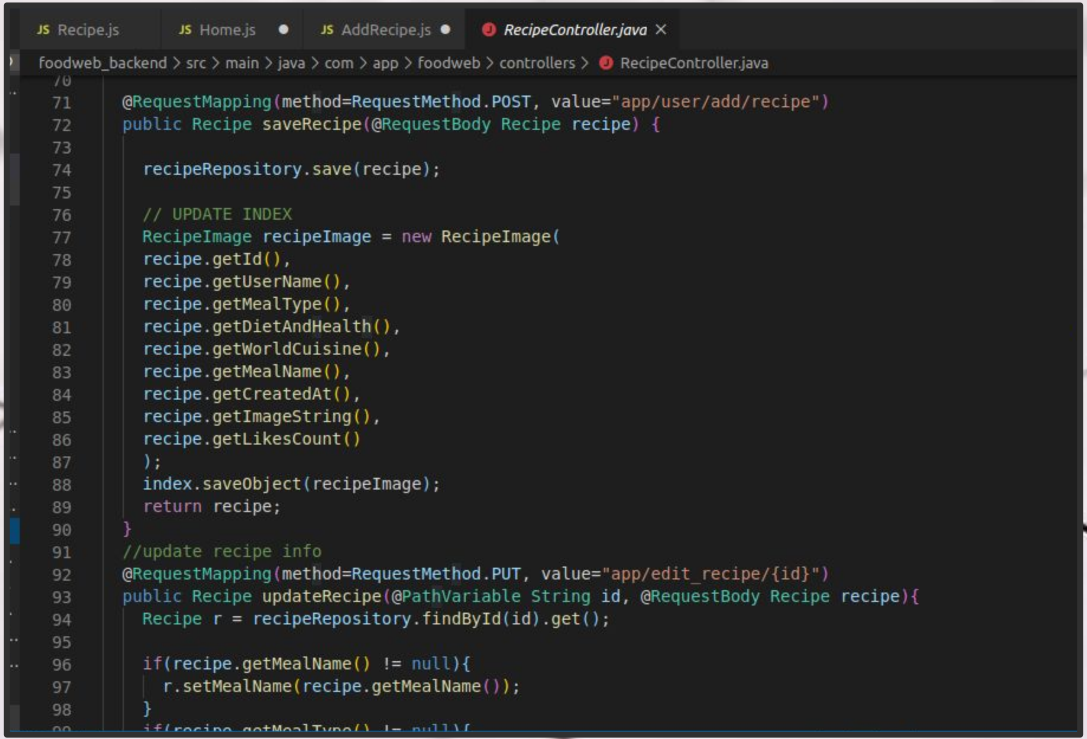

Recipe.js
Home.js
AddRecipe.js
foodweb_frontend > src > AddRecipe.js > AddRecipe > onSubmit

```
  onSubmit = (e) => {
    e.preventDefault();
    const{ userId, userName, mealType, dietAndHealth, worldCuisine, mealName, description,
      imageString, videoId, steps, ingredients } = this.state;
    axios.post('/user/add/recipe/', { userId, userName,  mealType, dietAndHealth,
      worldCuisine, mealName, description, imageString, videoId, steps, ingredients })
      .then((result) => {
        console.log("After Posting new Contact - returned data: " + result.data);
        const recipeID = result.data.id;
        this.props.history.push({
          pathname: "/Home",
          state: {userId: this.state.userId,
                  userName: this.state.userName,
                  imageString: this.state.profileImage
                }
        });

      })
      .catch((err) => {
        console.log(`======response.data=====`);
        console.log(`Errors: {errors}`);
      })
      .catch((err) => {
        console.log(`Errors: {errors}`);
      });
  }

  handleFileChange = e => {
    if (e.target.files[0]) {
```

Backend: Add Recipe



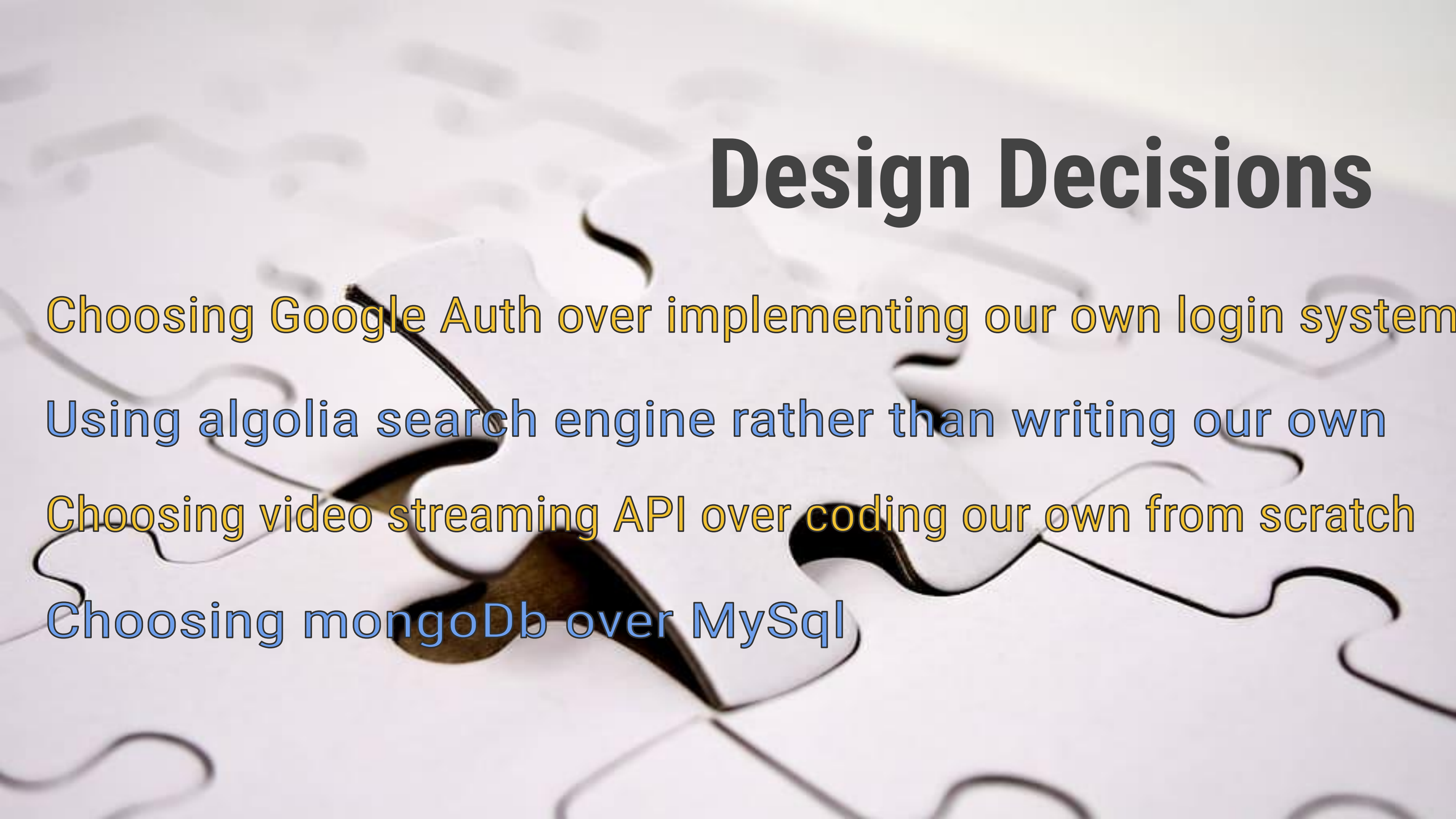
JS Recipe.js
JS Home.js
JS AddRecipe.js
RecipeController.java
foodweb_backend > src > main > java > com > app > foodweb > controllers > RecipeController.java

```java
  @RequestMapping(method=RequestMethod.POST, value="app/user/add/recipe")
  public Recipe saveRecipe(@RequestBody Recipe recipe) {

    recipeRepository.save(recipe);

    // UPDATE INDEX
    RecipeImage recipeImage = new RecipeImage(
    recipe.getId(),
    recipe.getUserName(),
    recipe.getMealType(),
    recipe.getDietAndHealth(),
    recipe.getWorldCuisine(),
    recipe.getMealName(),
    recipe.getCreatedAt(),
    recipe.getImageString(),
    recipe.getLikesCount()
    );
    index.saveObject(recipeImage);
    return recipe;
  }
  //update recipe info
  @RequestMapping(method=RequestMethod.PUT, value="app/edit_recipe/{id}")
  public Recipe updateRecipe(@PathVariable String id, @RequestBody Recipe recipe){
    Recipe r = recipeRepository.findById(id).get();

    if(recipe.getMealName() != null){
      r.setMealName(recipe.getMealName());
    }
```



Design Decisions

Choosing Google Auth over implementing our own login system

Using algolia search engine rather than writing our own

Choosing video streaming API over coding our own from scratch

Choosing mongoDb over MySql

Thank You !!!