

**Sprint 1 Submission**  
**Team Four Real**

■ **Team members**

Yu-Hsuan Lin, Yi-Jie Chou, LuoJia Zhao, Chienchia Chiu

■ **GitHub Repository/Project Board**

[https://github.com/zhaoluoJia/cs5500\\_group\\_project](https://github.com/zhaoluoJia/cs5500_group_project)

■ **Topic of the Project**

Exercise Manager

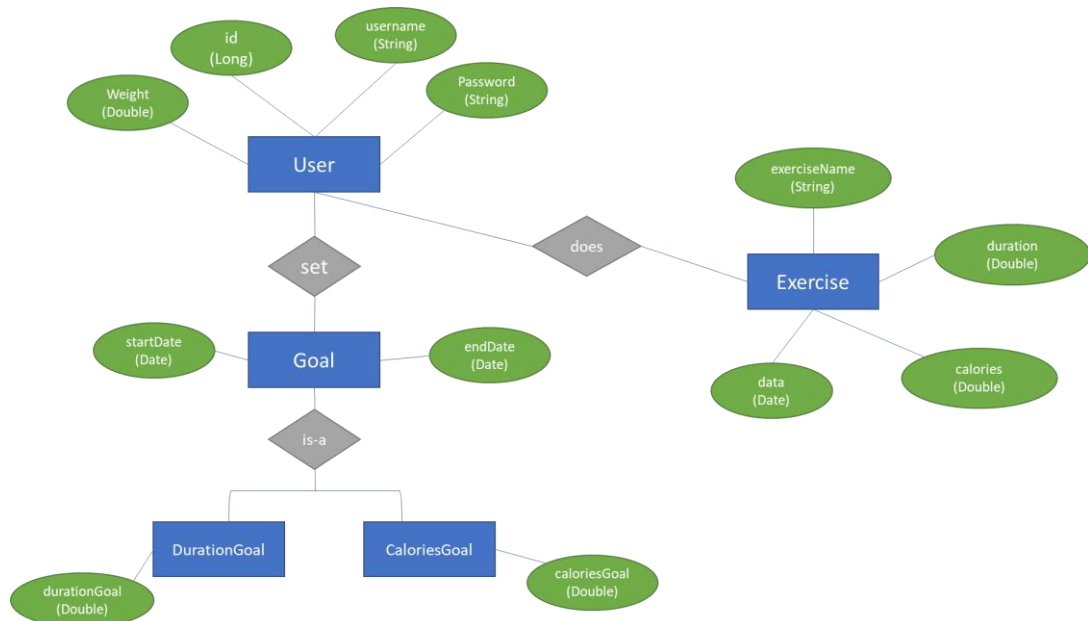
■ **Summary: What we have delivered**

User stories were modified after the meeting with TA on 2/14 (add the part to process input data), with business logic implemented in a Spring Boot Application to meet these stories. The database has been created to the MongoDB Cloud with storyline.json loaded and can be connected to our application. Unit tests are done on Models.

■ **User Stories & Delivery**

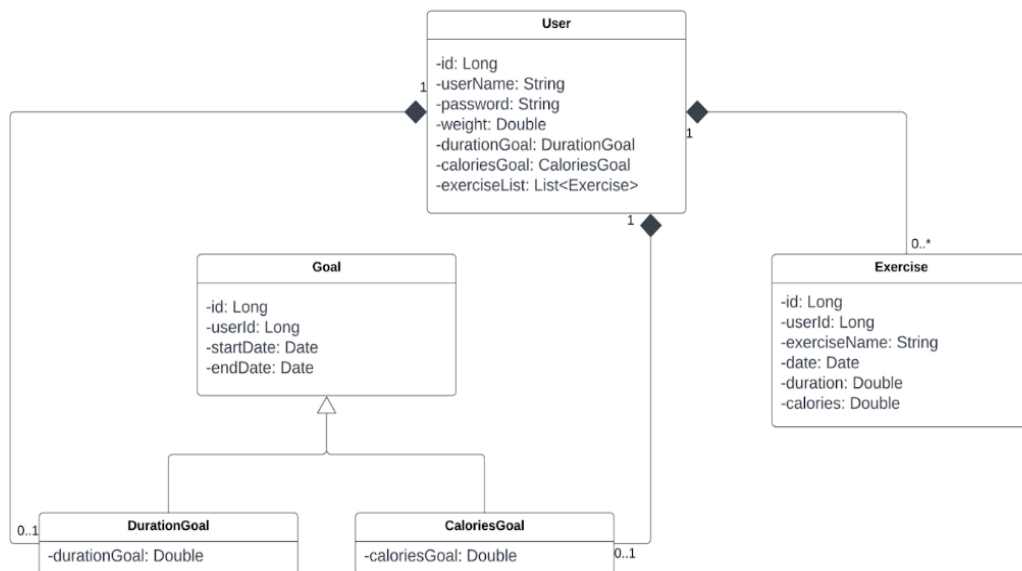
No	User Story	delivered	description
1	As an exercise lover, I want to get the calories burned in my activities by inputting the duration/type (walking/jogging/cycling/kayaking) of exercise and weight.	yes	In the model we have User and Exercise classes. A User object is created with attributes including weight, and Exercise. Exercise is created by a user with exercise name, date, and duration. User and Exercise CRUD operation along with the business logic supporting the calculation of calories are done in UserService, using a formula with MET. <a href="https://www.healthline.com/health/what-are-mets">https://www.healthline.com/health/what-are-mets</a>
2	As a person on a diet, I want to set a goal and compare the actual calories burned / actual duration run between a certain period.	yes	In the model, we created an abstract class Goal implemented by CaloriesGoal and DurationGoal. Implemented the business logic in UserService with CRUD operations of goal setting/query/update/deletion.
3	As an exercise lover, I want to get a weekly report about the past 7 days' exercise activities, with charts showing each day's total exercise duration/calories, and advice on which day I should exercise more.	yes	Analysis of the smallest calories/exercise duration between two dates is implemented in UserService. Planned to use API like Spring MVC to visualize each day's duration/ calories by charts when creating frontend.

## ERD



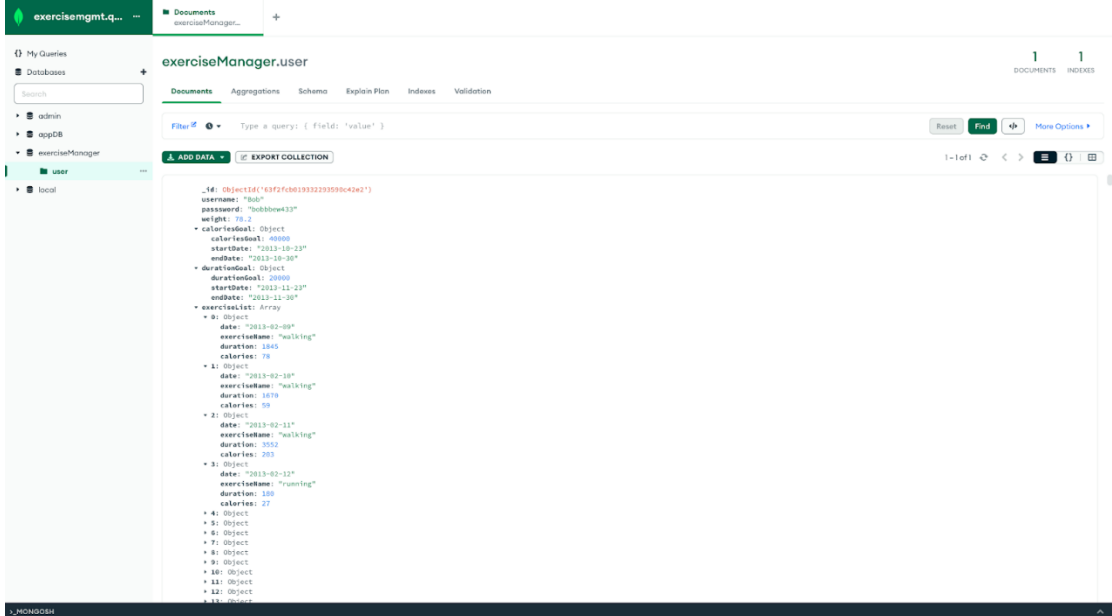
## UML

### Exercise Manager UML





Created MongoDB ExerciseManager on MongoDB Cloud (for ease of teamwork).  
The storyline.json has been imported into the database.



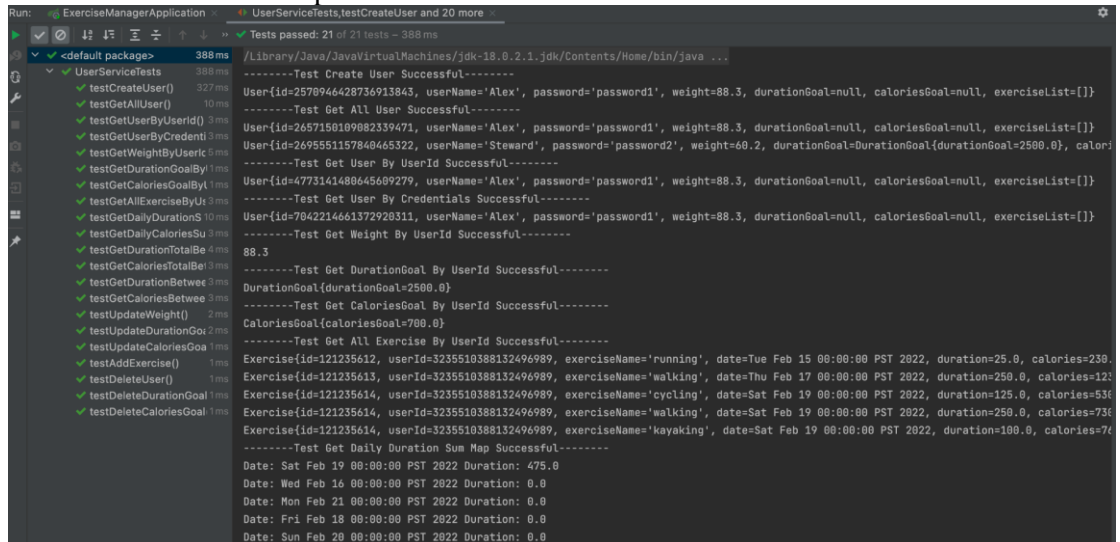
CRUD operations on the database can be successfully executed, with the repository and project configurations (maven dependencies/ application properties) set to connect to our database on the cloud. See the tests below.

## ■ Tests

We have done the unit tests to ensure we can do CRUD operations on our database. Some examples are presented below. The details (code) are on [Github](#).

The following screenshots are Service layer's test results. We've tested both normal cases and exceptions to ensure all business logic described in Sprint 1 user stories are satisfied.

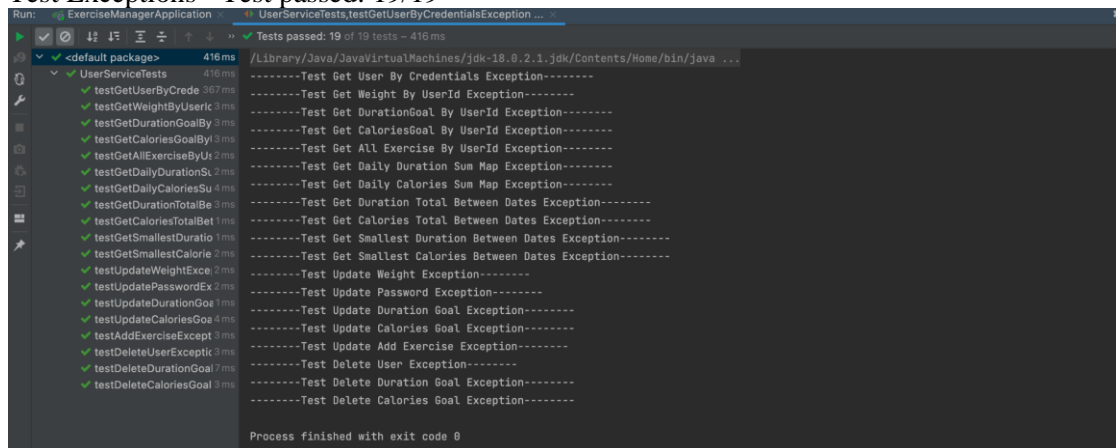
### 1. Test Normal Cases - Test passed: 21/21



```
Run: ExerciseManagerApplication, UserServiceTests.testCreateUser and 20 more
Tests passed: 21 of 21 tests - 388 ms

-----Test Create User Successful-----
User{id=2576946428736913843, userName='Alex', password='password1', weight=88.3, durationGoal=null, caloriesGoal=null, exerciseList=[]}
-----Test Get All User Successful-----
User{id=2657150109082339471, userName='Alex', password='password1', weight=88.3, durationGoal=null, caloriesGoal=null, exerciseList=[]}
User{id=2695551157840465322, userName='Steward', password='password2', weight=60.2, durationGoal=DurationGoal{durationGoal=2500.0}, caloriesGoal=730.0, exerciseList=[]}
-----Test Get User By UserId Successful-----
User{id=4773141488645689279, userName='Alex', password='password1', weight=88.3, durationGoal=null, caloriesGoal=null, exerciseList=[]}
-----Test Get User By Credentials Successful-----
User{id=7042214661372928311, userName='Alex', password='password1', weight=88.3, durationGoal=null, caloriesGoal=null, exerciseList=[]}
-----Test Get Weight By UserId Successful-----
88.3
-----Test Get DurationGoal By UserId Successful-----
DurationGoal{durationGoal=2500.0}
-----Test Get CaloriesGoal By UserId Successful-----
CaloriesGoal{caloriesGoal=700.0}
-----Test Get All Exercise By UserId Successful-----
Exercise{id=121235612, userId=3235510388132496989, exerciseName='running', date=Tue Feb 15 00:00:00 PST 2022, duration=25.0, calories=230.0}
Exercise{id=121235613, userId=3235510388132496989, exerciseName='walking', date=Thu Feb 17 00:00:00 PST 2022, duration=250.0, calories=121.0}
Exercise{id=121235614, userId=3235510388132496989, exerciseName='cycling', date=Sat Feb 19 00:00:00 PST 2022, duration=125.0, calories=53.0}
Exercise{id=121235614, userId=3235510388132496989, exerciseName='walking', date=Sat Feb 19 00:00:00 PST 2022, duration=250.0, calories=73.0}
Exercise{id=121235614, userId=3235510388132496989, exerciseName='kayaking', date=Sat Feb 19 00:00:00 PST 2022, duration=100.0, calories=7.0}
-----Test Get Daily Duration Sum Map Successful-----
Date: Sat Feb 19 00:00:00 PST 2022 Duration: 475.0
Date: Wed Feb 16 00:00:00 PST 2022 Duration: 0.0
Date: Mon Feb 21 00:00:00 PST 2022 Duration: 0.0
Date: Fri Feb 18 00:00:00 PST 2022 Duration: 0.0
Date: Sun Feb 20 00:00:00 PST 2022 Duration: 0.0
```

### 2. Test Exceptions - Test passed: 19/19



```
Run: ExerciseManagerApplication, UserServiceTests.testGetUserByCredentialsException and 18 more
Tests passed: 19 of 19 tests - 416 ms

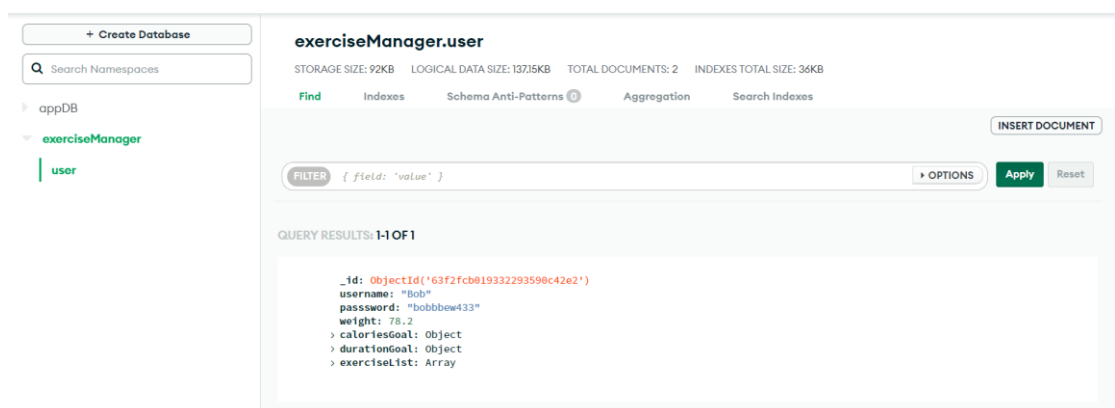
-----Test Get User By Credentials Exception-----
-----Test Get Weight By UserId Exception-----
-----Test Get DurationGoal By UserId Exception-----
-----Test Get CaloriesGoal By UserId Exception-----
-----Test Get All Exercise By UserId Exception-----
-----Test Get Daily Duration Sum Map Exception-----
-----Test Get Daily Calories Sum Map Exception-----
-----Test Get Duration Total Between Dates Exception-----
-----Test Get Calories Total Between Dates Exception-----
-----Test Get Smallest Duration Between Dates Exception-----
-----Test Get Smallest Calories Between Dates Exception-----
-----Test Update Weight Exception-----
-----Test Update Password Exception-----
-----Test Update Duration Goal Exception-----
-----Test Update Calories Goal Exception-----
-----Test Update Add Exercise Exception-----
-----Test Delete User Exception-----
-----Test Delete Duration Goal Exception-----
-----Test Delete Calories Goal Exception-----

Process finished with exit code 0
```

Moreover, to ensure that our operation does connect to our database, the following are the test results that are reflected in our database. (The details (code) are on [Github](#))

### 1. Test to create a user and check if it is created in the database.

database before



exerciseManager.user

STORAGE SIZE: 92KB LOGICAL DATA SIZE: 13715KB TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

FILTER { field: 'value' } OPTIONS Apply Reset

QUERY RESULTS: 1-1 OF 1

```
{
  "_id": ObjectId('63f2fcb019332293590c42e2'),
  "username": "Bob",
  "password": "bobbbow433",
  "weight": 78.2,
  "caloriesGoal": Object,
  "durationGoal": Object,
  "exerciseList": Array
}
```

## Test create user

```
no usages Elena Zhao
12 @SpringBootTest
13 class ExerciseManagerApplicationTests {
14
15     2 usages
16     @Autowired
17     private UserRepository userRepository;
18
19     no usages Elena Zhao
20     @Test
21     public void testCreateUser() {
22         User newUser = new User( userName: "Jose", password: "seg", weight: 77.6);
23         userRepository.save(newUser);
24         Optional<User> userFromDB = userRepository.findByUserName("Jose");
25         assertEquals(Optional.of(newUser).toString(), userFromDB.toString());
26     }
27 }
```

## database after

exerciseManager.user

STORAGE SIZE: 92KB LOGICAL DATA SIZE: 13715KB TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

FILTER { field: 'value' } OPTIONS Apply Reset

```
{
  "_id": "ObjectId('63f2fcb019332293598c42e2')",
  "username": "Bob",
  "password": "b0bbbnw433",
  "weight": 78.2,
  "caloriesGoal": Object,
  "durationGoal": Object,
  "exerciseList": Array
}
```

```
{
  "_id": 5805310040165204220,
  "username": "Jose",
  "password": "seg",
  "weight": 77.6,
  "exerciseList": Array,
  "_class": "com.exerciseMgr.exerciseManager.model.User"
}
```

2. Test to get the id from DB (expected 5805310040165204220) of a user (Jose) from database. Test passed.

```
no usages Elena Zhao
26 @Test
27 public void testGetId() {
28     User userFromDB = userRepository.findByUserName("Jose").orElse( other: null);
29     Long id = userFromDB.getId();
30     assertEquals(Optional.of(id), Optional.of( value: 5805310040165204220L));
31 }
32 }
```

Tests passed: 1 of 1 test - 907 ms

### 3. Test to delete the user with id 5805310040165204220

Database before

**exerciseManager.user**

STORAGE SIZE: 92KB LOGICAL DATA SIZE: 137.15KB TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes

INSERT DOCUMENT

FILTER { field: 'value' } OPTIONS Apply Reset

```
{
  "_id": ObjectId('63f2fcb019332293590c42e2'),
  "username": "Bob",
  "password": "bobbew433",
  "weight": 78.2,
  "> caloriesGoal": Object,
  "> durationGoal": Object,
  "> exerciseList": Array
}
```

```
{
  "_id": 5805310040165204220,
  "userName": "Jose",
  "password": "seg",
  "weight": 77.6,
  "> exerciseList": Array,
  "_class": "com.exerisemgr.exercisemanager.model.User"
}
```

```
42      no usages Elena Zhao *
43      @Test
44      public void testDeleteUser() {
45          userRepository.deleteById(5805310040165204220L);
46          assertEquals(userRepository.findByUserName("Jose"), actual: Optional.empty());
47      }
48  }
```

Database after

**exerciseManager.user**

STORAGE SIZE: 92KB LOGICAL DATA SIZE: 137.01KB TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes

INSERT DOCUMENT

FILTER { field: 'value' } OPTIONS Apply Reset

QUERY RESULTS: 1-1 OF 1

```
{
  "_id": ObjectId('63f2fcb019332293590c42e2'),
  "username": "Bob",
  "password": "bobbew433",
  "weight": 78.2,
  "> caloriesGoal": Object,
  "> durationGoal": Object,
  "> exerciseList": Array
}
```