

## Sprint 2 Scoping

### Team Four Real

#### ■ Team Members

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

#### ■ GitHub Repository

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

#### ■ Project Board

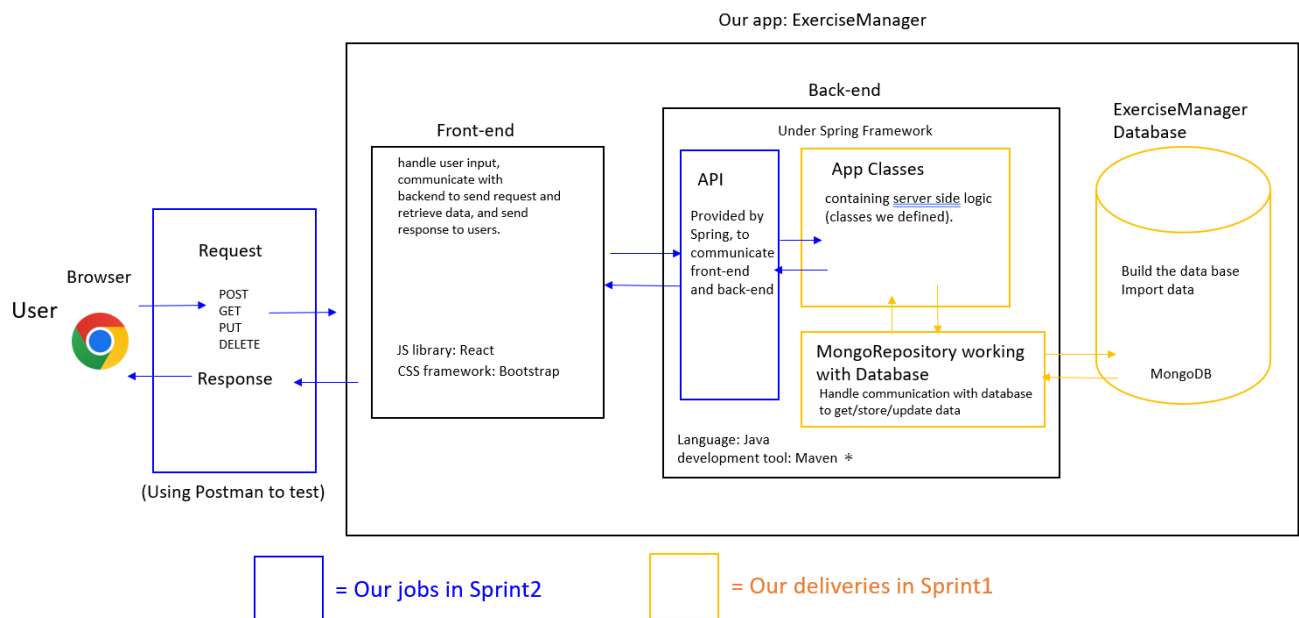
<https://github.com/users/zhaoluoJia/projects/1/views/3>

#### ■ Project Name

Exercise Manager

#### ■ Sprint 2 Scope

In Sprint 2, we're going to implement the REST API that allows user to query data from the website as well as to put new data into the system.



\* Maven is used to manage the dependencies of a Spring application, such as the Spring framework itself and any other libraries that the application needs.

#### ■ Sprint 2 Tasks

##### 1. Fix Sprint 1 bugs

- Duplicate usernames are not allowed to be added into the database

##### 2. Implement Controller for backend

- File structure:

ExerciseManager/src/main/java/com.exerisemgr.exercisemanager/controller/UserController

b. Methods to implement:

```
@RestController
@RequestMapping("/api/users")
public class UserController {

    @Autowired
    private UserService userService;

    /**
     * Create the User.
     * @param user: the user object including all information.
     * @return: the User.
     */
    @RequestMapping(value="/register", method = RequestMethod.POST)
    public User createUser(@RequestBody User user) {
        // TODO implementation.
        return null;
    }

    /**
     * Get the User by userId
     * @param userId: the userId.
     * @return: the User.
     */
    @RequestMapping(value="/{userId}", method = RequestMethod.GET)
    public User getUserById(@PathVariable Long userId) {
        // TODO implementation.
        return null;
    }
}
```

```

/**
 * Get the User by user information.
 * @param user the user object.
 * @return User object
 */
@RequestMapping(value = "/login", method = RequestMethod.GET)
public User getUserByCredentials(@RequestBody User user) {
    // TODO implementation.
    return null;
}

/**
 * Get the weight of the user by userId.
 * @param userId userId.
 * @return user's weight.
 */
@RequestMapping(value = "/{userId}/weight", method = RequestMethod.GET)
public Double getWeightById(@PathVariable Long userId) {
    // TODO implementation.
    return null;
}

/**
 * Get the DurationGoal object by userId.
 * @param userId userId.
 * @return user's DurationGoal object.
 */
@RequestMapping(value = "/{userId}/durationGoal", method = RequestMethod.GET)
public DurationGoal getDurationGoalById(@PathVariable Long userId) {
    // TODO implementation.
    return null;
}

```

```

/**
 * Get the CaloriesGoal object by userId.
 * @param userId userId.
 * @return user's CaloriesGoal object.
 */
@RequestMapping(value = "{userId}/caloriesGoal", method = RequestMethod.GET)
public CaloriesGoal getCaloriesGoalByUserId(@PathVariable Long userId) {
    // TODO implementation.
    return null;
}

/**
 * Get a list of all Exercise objects of the user by userId
 * @param userId userId.
 * @return a list of Exercise objects.
 */
@RequestMapping(value = "{userId}/exercises", method = RequestMethod.GET)
public List<Exercise> getAllExerciseByUserId(@PathVariable Long userId) {
    // TODO implementation.
    return null;
}

/**
 * Get a map of each date to that day's duration sum by user id and start date, end date.
 * @param userId userId.
 * @param startDate the start date (Date).
 * @param endDate the end date (Date).
 * @return a Map mapping each day to its duration sum.
 */
@RequestMapping(value = "{userId}/dailyDurations", method = RequestMethod.GET)
public Map<Date, Double> getDailyDurationSumMap(@PathVariable Long userId, @RequestParam Date startDate,
    @RequestParam Date endDate) {
    // TODO implementation.
    return null;
}

```

```

/**
 * Get a map of each date to that day's calories sum by user id and start date, end date.
 * @param userId userId.
 * @param startDate the start date (Date).
 * @param endDate the end date (Date).
 * @return a Map mapping each day to its calories sum.
 */
@RequestMapping(value = "{userId}/dailyCalories", method = RequestMethod.GET)
public Map<Date, Double> getDailyCaloriesSumMap(@PathVariable Long userId, @RequestParam Date startDate,
    @RequestParam Date endDate) {
    // TODO implementation.
    return null;
}

/**
 * Get the total duration between the given start date and end date.
 * @param userId userId.
 * @param startDate the start date (Date).
 * @param endDate the end date (Date).
 * @return the total Duration between start date and end date.
 */
@RequestMapping(value = "{userId}/durationTotalBetweenDates", method = RequestMethod.GET)
public Double getDurationTotalBetweenDates(@PathVariable Long userId,
    @RequestParam Date startDate,
    @RequestParam Date endDate) {
    // TODO implementation.
    return null;
}

```

```

/**
 * Get the total calories between the given start date and end date.
 * @param userId userId.
 * @param startDate the start date (Date).
 * @param endDate the end date (Date).
 * @return the total calories between start date and end date.
 */
@RequestMapping(value = "/{userId}/caloriesTotalBetweenDates", method = RequestMethod.GET)
public Double getCaloriesTotalBetweenDates(@PathVariable Long userId,
    @RequestParam Date startDate,
    @RequestParam Date endDate) {
    // TODO implementation.
    return null;
}

/**
 * Get the smallest duration between the given start date and end date.
 * @param userId userId.
 * @param startDate the start date (Date).
 * @param endDate the end date (Date).
 * @return the smallest duration between start date and end date.
 */
@RequestMapping(value = "/{userId}/smallestDurationBetweenDates", method = RequestMethod.GET)
public Double getSmallestDurationBetweenDates(@PathVariable Long userId,
    @RequestParam Date startDate,
    @RequestParam Date endDate) {
    // TODO implementation.
    return null;
}

```

```

/**
 * Get the smallest calories between the given start date and end date.
 * @param userId userId.
 * @param startDate the start date (Date).
 * @param endDate the end date (Date).
 * @return the smallest calories between start date and end date.
 */
@RequestMapping(value = "/{userId}/smallestCaloriesBetweenDates", method = RequestMethod.GET)
public Double getSmallestCaloriesBetweenDates(@PathVariable Long userId,
    @RequestParam Date startDate,
    @RequestParam Date endDate) {
    // TODO implementation.
    return null;
}

/**
 * Update weight for the user.
 * @param userId userId.
 * @param weight the weight the user wants to change to.
 */
@RequestMapping(value = "/{userId}/weight", method = RequestMethod.PUT)
public void updateWeight(@PathVariable Long userId,
    @RequestParam Double weight) {
    // TODO implementation.
}

/**
 * Update password for the user.
 * @param userId userId.
 * @param password the password the user want to change to.
 */
@RequestMapping(value = "/{userId}/password", method = RequestMethod.PUT)
public void updatePassword(@PathVariable Long userId,
    @PathVariable String password) {
    // TODO implementation.
}

```

```

/**
 * Update duration goal for the user.
 * @param userId userId.
 * @param startDate the starting date of the duration goal.
 * @param endDate the ending date of the duration goal.
 * @param durationGoal the duration goal the user want to change to.
 */
@RequestMapping(value = "{userId}/durationalGoal", method = RequestMethod.PUT)
public void updateDurationGoal(@PathVariable Long userId,
    @RequestParam Double startDate,
    @RequestParam Double endDate,
    @RequestParam Double durationGoal) {
    // TODO implementation.
}

/**
 * Update calories goal for the user.
 * @param userId userId.
 * @param startDate the starting date of the calories goal.
 * @param endDate the ending date of the calories goal.
 * @param caloriesGoal the calories goal the user want to change to.
 */
@RequestMapping(value = "{userId}/caloriesGoal", method = RequestMethod.PUT)
public void updateCaloriesGoal(@PathVariable Long userId,
    @RequestParam Double startDate,
    @RequestParam Double endDate,
    @RequestParam Double caloriesGoal) {
    // TODO implementation.
}

```



```

/**
 * Create new exercise in the exercise list for the user.
 * @param userId userId.
 * @param exerciseName the exercise the user want to add.
 * @param date the date of the exercise.
 * @param duration the duration of the exercise.
 */
@RequestMapping(value = "/{userId}/exerciseList", method = RequestMethod.POST)
public Exercise createExercise(@PathVariable Long userId,
    @RequestParam String exerciseName,
    @RequestParam Date date,
    @RequestParam Double duration) {
    // TODO implementation.
    return null;
}

/**
 * Delete the user.
 * @param userId userId.
 */
@RequestMapping(value = "/{userId}", method = RequestMethod.DELETE)
public void deleteUser(@PathVariable Long userId) {
    // TODO implementation.
}

/**
 * Delete the duration goal for the user.
 * @param userId userId.
 */
@RequestMapping(value = "/{userId}/durationGoal", method = RequestMethod.DELETE)
public void deleteDurationGoal(@PathVariable Long userId) {
    // TODO implementation.
}

```

```

/**
 * Delete the calories goal for the user.
 * @param userId userId.
 */
@RequestMapping(value = "/{userId}/caloriesGoal", method = RequestMethod.DELETE)
public void deleteCaloriesGoal(@PathVariable Long userId) {
    // TODO implementation.
}

```

### 3. Test Controller

We're planning to use Postman to create and send API requests to our backend system. Please see below our test plan in details.

#	Test Method	Operation	Test URL	Request Body	Expected Test Result
1	public User createUser(@RequestBody User user)	POST	http://localhost:8080 /api/users/register	{ "userName": "Leo", "password": "p@assWord7", "weight": 77.8 }	New user with userName Leo added to the database
2	public User getUserByCredentials(@RequestBody User user)	GET	http://localhost:8080 /api/users/login	{ "userName": "Leo", "password": "p@assWord7", "weight": 0 }	User with userName Leo returned
3	public ResponseEntity<List<User>> allUser()	GET	http://localhost:8080 /api/users	N/A	All users in the database returned
4	public User getUserById(@PathVariable Long userId)	GET	http://localhost:8080 /api/users/63f2fcb01 9332293590c42e2	N/A	User with userId 63f2fcb019332293 590c42e2 returned
5	public Double getWeightByUserId(@PathVariable Long userId)	GET	http://localhost:8080 /api/users/63f2fcb01 9332293590c42e2	N/A	Weight with userId 63f2fcb019332293 590c42e2 returned
6	public DurationGoal getDurationGoalByUserId(@PathVariable Long userId)	GET	http://localhost:8080 /api/users/63f2fcb01 9332293590c42e2	N/A	DurationGoal with userId 63f2fcb019332293 590c42e2 returned
7	public CaloriesGoal	GET	http://localhost:8080 /api/users/63f2fcb01	N/A	CaloriesGoal with userId

	getCaloriesGoalByUserId(@PathVariable Long userId)		9332293590c42e2		63f2fcb019332293590c42e2 returned
8	public List<Exercise> getAllExerciseByUserId(@PathVariable Long userId)	GET	http://localhost:8080/api/users/63f2fcb019332293590c42e2/exercises	N/A	All exercises with userId 63f2fcb019332293590c42e2 returned
9	public Map<Date, Double> getDailyDurationSumMap(@PathVariable Long userId, @RequestParam Date startDate, @RequestParam Date endDate)	GET	http://localhost:8080/api/users/63f2fcb019332293590c42e2/dailyDurations?startDate=2023-02-22&endDate=2023-02-28	N/A	The daily duration sum map from 2023-02-22 to 2023-02-28 with userId 63f2fcb019332293590c42e2 returned
10	public Map<Date, Double> getDailyCaloriesSumMap(@PathVariable Long userId, @RequestParam Date startDate, @RequestParam Date endDate)	GET	http://localhost:8080/api/users/63f2fcb019332293590c42e2/dailyCalories?startDate=2023-02-22&endDate=2023-02-28	N/A	The daily calories sum map from 2023-02-22 to 2023-02-28 with userId 63f2fcb019332293590c42e2 returned
11	public Double getDurationTotalBetweenDates(@PathVariable Long userId, @RequestParam Date startDate, @RequestParam Date endDate)	GET	http://localhost:8080/api/users/63f2fcb019332293590c42e2/durationTotalBetween?startDate=2023-02-22&endDate=2023-02-28	N/A	The duration total from 2023-02-22 to 2023-02-28 with userId 63f2fcb019332293590c42e2 returned
12	public Double getCaloriesTotalBetweenDates(@PathVariable Long userId, @RequestParam Date startDate, @RequestParam Date endDate)	GET	http://localhost:8080/api/users/63f2fcb019332293590c42e2/caloriesTotalBetween?startDate=2023-02-22&endDate=2023-02-28	N/A	The calories total from 2023-02-22 to 2023-02-28 with userId 63f2fcb019332293590c42e2 returned
13	public Double getSmallestDurationBetweenDates(@PathVariable	GET	http://localhost:8080/api/users/63f2fcb019332293590c42e2/smallestDurationBetweenDates	2023-02-22 (startDate)  2023-02-28 (endDate)	The smallest duration from 2023-02-22 to 2023-02-28 with userId

	Long userId,@Request Param Date startDate,@Requ estParam Date endDate)				63f2fcb019332293 590c42e2 returned
14	public Double getSmallestCalor iesBetweenDates (@PathVariable Long userId,@Request Param Date startDate,@Requ estParam Date endDate)	GET	http://localhost:8080 /api/users/63f2fcb01 9332293590c42e2/s mallestCaloriesBetw eenDates	2023-02-22 (startDate)  2023-02-28 (endDate)	The smallest calories from 2023-02-22 to 2023-02-28 with userId 63f2fcb019332293 590c42e2 returned
15	public void updateWeight(@ PathVariable Long userId,@Request Param Double weight)	PUT	http://localhost:8080 /api/users/63f2fcb01 9332293590c42e2/w eight	100.0 (weight)	update userId 63f2fcb019332293 590c42e2 weight to 100.0
16	public void updatePassword( @PathVariable Long userId,@PathVaria ble String password)	PUT	http://localhost:8080 /api/users/63f2fcb01 9332293590c42e2/{ {password}}	N/A	update userId 63f2fcb019332293 590c42e2 password
17	public void updateDurationGo al(@PathVariable Long userId,@RequestP aram Double startDate,@Reque stParam Double endDate,@Reques tParam Double	PUT	http://localhost:8080 /api/users/63f2fcb01 9332293590c42e2/d urationGoal	3000.0 (durationGoal)  2023-02-22 (startDate)  2023-02-28 (endDate)	update userId 63f2fcb019332293 590c42e2 durationGoal to 3000.0 from 2023-02-22 to 2023-02-28

	durationGoal)				
18	public void updateCaloriesGoal(@PathVariable Long userId,@RequestParam Double startDate,@RequestParam Double endDate,@RequestParam Double caloriesGoal)	PUT	http://localhost:8080 /api/users/63f2fcb01 9332293590c42e2/caloriesGoal	250.0 (caloriesGoal)  2023-02-22 (startDate)  2023-02-28 (endDate)	update userId 63f2fcb019332293 590c42e2 caloriesGoal to 250.0 from 2023-02-22 to 2023-02-28
19	public Exercise createExercise(@PathVariable Long userId,@RequestParam String exerciseName,@RequestParam Date date,@RequestParam Double duration)	POST	http://localhost:8080 /api/users/63f2fcb01 9332293590c42e2/exerciseList	walking (exerciseName)  2023-02-22 (startDate)  2023-02-28 (endDate)	new walking exercise with userId 63f2fcb019332293 590c42e2 from 2023-02-22 to 2023-02-28
20	public void deleteUser(@PathVariable Long userId)	DELETE	http://localhost:8080 /api/users/63f2fcb01 9332293590c42e2/	N/A	delete user with userId 63f2fcb019332293 590c42e2
21	public void deleteDurationGoal(@PathVariable Long userId)	DELETE	http://localhost:8080 /api/users/63f2fcb01 9332293590c42e2/durationGoal	N/A	delete durationGoal with userId 63f2fcb019332293 590c42e2
22	public void deleteCaloriesGoal(@PathVariable Long userId)	DELETE	http://localhost:8080 /api/users/63f2fcb01 9332293590c42e2/caloriesGoal	N/A	delete caloriesGoal with userId 63f2fcb019332293 590c42e2

#### 4. Code quality

We are going to utilize one or more [Software code quality tools](#) to ensure our code quality.