

# Functional Design



“To optimise every gymnast training-goals in a fun, safe and efficient way”

# Preface

- a. Pitch
- b. Goal application
- c. User-Stories
- d. Use-Case Descriptions
- e. User Requirements
- f. Conceptual Design
- g. UML-Diagram
- h. Flow Diagrams
- i. Wireframe
- j. Acceptance test // nog toevoegen



## Updated in version 2

- Use-Case-Description when the End-Goal is reached by a gymnast.
- Updated version of the conceptual model: less is more with a little more abstraction.
- Added user-story for working on multiple training routes simultaneously.
- Improved lay-out
- Use-Case-Diagram update #2: To the point en cut away unnecessary items.
- Added Flow Diagrams: use case start exercise + abstract start finish route.
- Added Wireframe : Exercise

## Updated in version 3

- Use-Case-Diagram #3: Focus on most important basic functions
- Deleted Flow-Diagram starting-Use-Case
- Added new and improved Flow-Diagram in response to the given feedback
- Updated Conceptual model
- Updated most important use-case-stories and requirement in green and deleted the green with the others.
- Updated Use-case-discription 002

## Updated in version 4

- 4 acceptance tests of the most important use-cases.
- Updated Conceptual model: more to the point flow of concepts // deleted 2 start-endgoal
- Updated most important user requirement.

## Pitch

My application focuses on an optimal individual learning trajectory for a gymnast. An application that guides an athlete to achieve a desired goal. The application provides multiple courses and within each course multiple intermediate steps and training options to the aspired goal. It uses reflection from the user to suggest alternative routes that better suit the athlete's current skill-level with a clever algorithm. It can also suggest optimal training-tips to complete exercises with technical beauty. The application distributes points for each completed task. Points that are connected to an individual user. This stimulates the training process en makes it fun and competitive.

Goal 1: Give insight in the methodology of multiple core-gymnastic-elements.

Goal 2: Raise the enthusiasm of training.

Goal 3: Prevent unnecessary injuries.

Goal 4: Bundling knowledge of different methodologies.

<b>Actors</b>	<ul style="list-style-type: none"><li>• Gymnast</li><li>• Trainer</li></ul>
<b>Use Cases</b>	<ul style="list-style-type: none"><li>• Login with individual account</li><li>• View earned points</li><li>• Select training-route</li><li>• Extract user information</li><li>• Suggest optimal training-route</li><li>• Suggest suitable training-exercises</li><li>• Distribute points for completing exercises</li></ul>

## User Stories

Gymnast	"I want to log in into a secure environment where my data is safe and cannot be altered by other people. "
Athlete	"I want to see my total earned points at all times at my screen."
Athlete	"In the application is want an easy-to-follow user-interface without unnecessary information. This also has to translate to the exercises ...Easy to follow, empowered with visual sources and an attractive design."
Athlete	"If I don't get an exercise quit right, I want the system to guide me through a different training-route. One that's suits me better than the original one."
Trainer	"If I login into the application, I want the option to immediately look at the complete program for each exercise-route."
Athlete	"If I get stuck on an exercise I want the option to revisit the previous exercise, continue to train on the current exercise or continue to the next exercise if I want to complete and continue the course. "
Athlete	"If I get stuck on an exercise I want an alternative option or tips how to train properly to succeed in the original exercise."
Athlete	"I want to simultaneously work on multiple training-goals in the application."

## Use Case Discriptions

Use case name	<b>UC-001: Continue Next Exercise</b>
Scenario	Gymnast completes exercise after exercise and is on the trajectory of the original training-route.
Brief Description	The gymnast completes the exercises prior to the last one. The application continues to the next exercise till reaching the last End-Goal-Exercise. The gymnast completes a trainingroute by completing the last goal of the route. He/She is following the standard route that is given with all the requirements of the exercises.
Actors	Gymnast
Pre-Conditions	Gymnast has given user-input on multiple occasions that has helped to guide her in the optimal trainingroute. The gymnast has almost completed the route and there are only a few exercises left before completing the selected course.
Flow of Activities	<ol style="list-style-type: none"> <li>The gymnast continues a course within the system.</li> <li>The gymnast completes the for-last exercise.</li> <li>Before continuing to the last Exercise, the gymnast is asked for some feedback and user-input about the course.</li> <li>The gymnast gets a thorough description of the last exercise with the needed requirements that he/she must perform. The preference is given to grade the exercise with a trainer.</li> <li>After performing the exercise the gymnast is asked to select the option that is best suited for the performance. If the performance meets the requirements of the exercise. The route is completed.</li> <li>The gymnast earned extra point for completing an entire trainingroute and the system marks the selected route as complete. This way the gymnast can revisit every part of the route, without earning the points.</li> </ol>
Related User-Cases	<b>UC-002</b> <b>UC-003</b>

Use case name	<b>UC-002: Go to previous exercise</b>
Scenario	Can't complete the suggested exercise.
Brief Description	The user follows a course and is shown the next exercise. However, the exercise is too difficult or the execution is flawed.
Actors	Gymnast
Pre-Conditions	The user has completed the previous exercise
Flow of Activities	<ol style="list-style-type: none"> <li>The gymnast selects a course in the system.</li> <li>The gymnast answers some questions about his/her starting level.</li> <li>All goes well and the gymnast is completing exercise after exercise within the course. Then he comes across an exercise that he finds difficult.</li> <li>The gymnasts takes time to train with the exercise, but discovers that he can't complete the exercise with the given norms. He finds it too difficult.</li> <li>After giving some user-input, the algorithm gives the user the option to revisit the previous exercise to improve his/hers technique.</li> </ol>

	l. After practising with the previous exercise, the gymnast is ready for the original exercise till he/she reaches his goal.
Post-Conditions	Successful Completion. a. The gymnast can continue the course. b. The gymnasts earned points for completing the alternate exercise.
Related User-Cases	<b>UC-003</b>

<b>Use case name</b>	<b>UC-003: Suggest Practice for current Exercise</b>
Scenario	Can't complete the suggested exercise, due to technical or physical shortcomings.
Brief Description	The system has selected an exercise within a selected course. However, the exercise is too difficult or the execution is flawed. The gymnast however, wants to complete the current (original) exercise within the route.
Actors	Gymnast
Pre-Conditions	Gymnast has given initial user-input at the start of the course so the system can estimate the optimal starting exercise.
Flow of Activities	<ol style="list-style-type: none"> <li>The gymnast selects a course in the system.</li> <li>The gymnast answers some questions about his/her starting level.</li> <li>The algorithm selects the optimal starting exercise</li> <li>All goes well and the gymnast is completing exercise after exercise within the course. Then he comes across an exercise that is nu as simple as the previous ones.</li> <li>The gymnasts takes time to train with the exercise, but discovers that he can't complete the exercise. His trainer points out that he is making some technical mistakes. His coordination is off en his arms are not in the correct posture.</li> <li>The gymnast asks the algorithm for help and selects the option: Suitable training-exercises.</li> <li>The gymnast is asked for some input about his technical problems. The algorithm than selects multiple exercises that he can do to increase his technical level for the exercise.</li> <li>The gymnast returns to the exercise and finds that the original exercise is way more easy than before. He completes the level and can continue the course to the End-Goal.</li> </ol>
Post-Conditions	<ol style="list-style-type: none"> <li>Successful Completion <ol style="list-style-type: none"> <li>The gymnast has technically increased his skill-level.</li> <li>The gymnast earned points for training with the exercises.</li> <li>The gymnast can continue to the next (original) exercise.</li> </ol> </li> </ol>
Related User-Cases	<b>UC-002</b>

## User Requirements

Requirement	Category	MOSCOW
The admin of the system can see the current number of users and the public information of these users.	Business Functional	C
The user is able to log in to the system with a personal id and password.	Stakeholder Non-functional	C
The user can create a personal profile within the application.	Stakeholder Functional	C
The user can select a course from the system within an easy to follow menu.	Stakeholder Functional	M
The user can give user-input as a response to clear en easy to follow questions from the algorithm.	Stakeholder Functional	S
The user can see his/her earned points at all times within the application.	System Non-functional	S
The user can select an course with a clear goal. The course has 5 to 20 exercises in it. The user can train with an exercise and has the option to complete the exercise. Based on a short feedback-cycle, the user continues to the next exercise, stays with the current or goes to the previous exercise within the course.	System Functional	S
The user has the option to select an alternative training route that better suits his/her skill-level. This route is suggested by the algorithm, based on the given input by the user.	System Functional	M
The user has the option to select multiple training-exercises to prepare for the original exercise. These training-exercises are suggested by the algorithm as an optimal choice.	System Functional	M
The user is distributed points for completion an exercise, training or the End-Goal. The points must be flexible distributed based on: difficulty, repetition, category.	System Functional	M
The user can revisit an unlocked exercise as many times as he/she wants. However, the system only gives points for the exercise the first time it is completed. (With an exception of the training-exercises).	System functional	M
The user can work on multiple training-goals at the same time within the application. The system remembers all the progress the user makes on the different training routes.	System	C

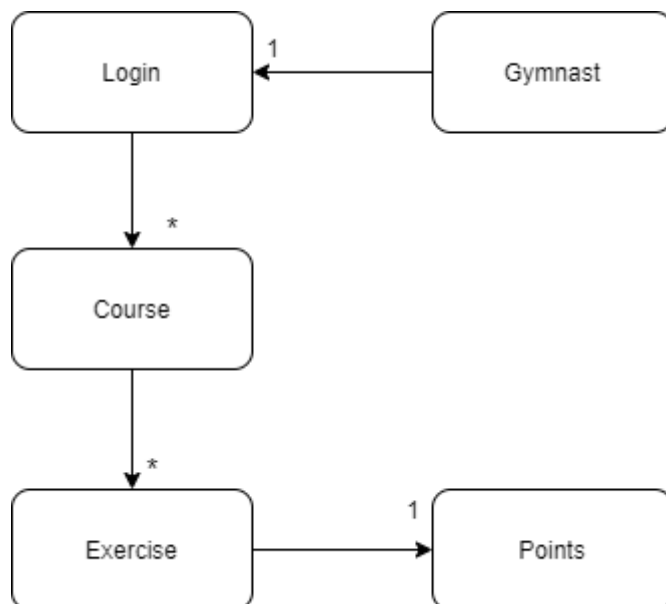
## Conceptual Model

\* Version #4

In the model below I represent the concepts in my application that guides a user to an aspired course.

First the user has to login for the system to know who is using the application (points distribution, history exercises etc.) Then the user can select multiple courses. Each course has multiple exercises in them.

The user can train with an exercise and can continue to the next by completing it. He/she has to validate the exercise with the given requirements. The system then will distribute points for completion and the user is brought to the next exercise in the course. This process continues till the user completes the course. By completing a course, the user reaches his goal.



## UML diagram

The diagram below explains the different interactions between the use-cases in the application. A user must log-in every time he uses the app. The system will verify the password and it will show an error if the user gives an incorrect input. When in the system, the user can view the available courses. Subsequently a user can select a course of choice.

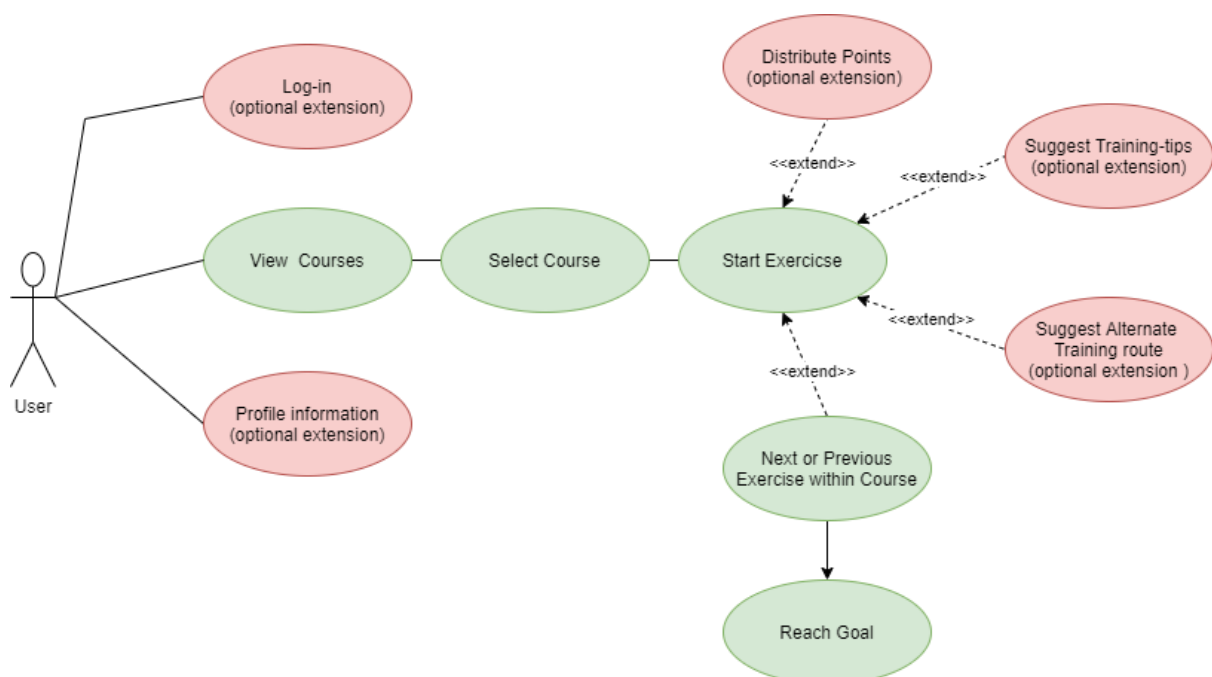
A user is asked for input, before the first exercise starts. The algorithm will select the best starting-exercise suited for the users level. When the user performs the first exercise, he/she gets multiple options, depended on how well the exercise is going. After practicing the exercise, the user is asked for some input. According to the answers, the algorithm will suggest or decide 3 possible outcomes.

1. Complete exercise and go to the next. This will continue to the end-goal of the course.
2. Suggest alternate training-route that is better suited for the user.
3. Suggest one or multiple training-exercises for the original exercise.

The user gets a set of points for completing an exercise. The points may vary, depending on the difficulty or length of the exercise.

The user also gets point for completing (extra training-exercises). Depending on the difficulty or length, repetition and execution the points may vary.

\* This is version #3. In this diagram I put the feedback to work on focussing on the main function of the application. "Choosing a goal and reach that goal in a course". I'm going to design this base of the application en mentioned the optional Use-Cases for extension.



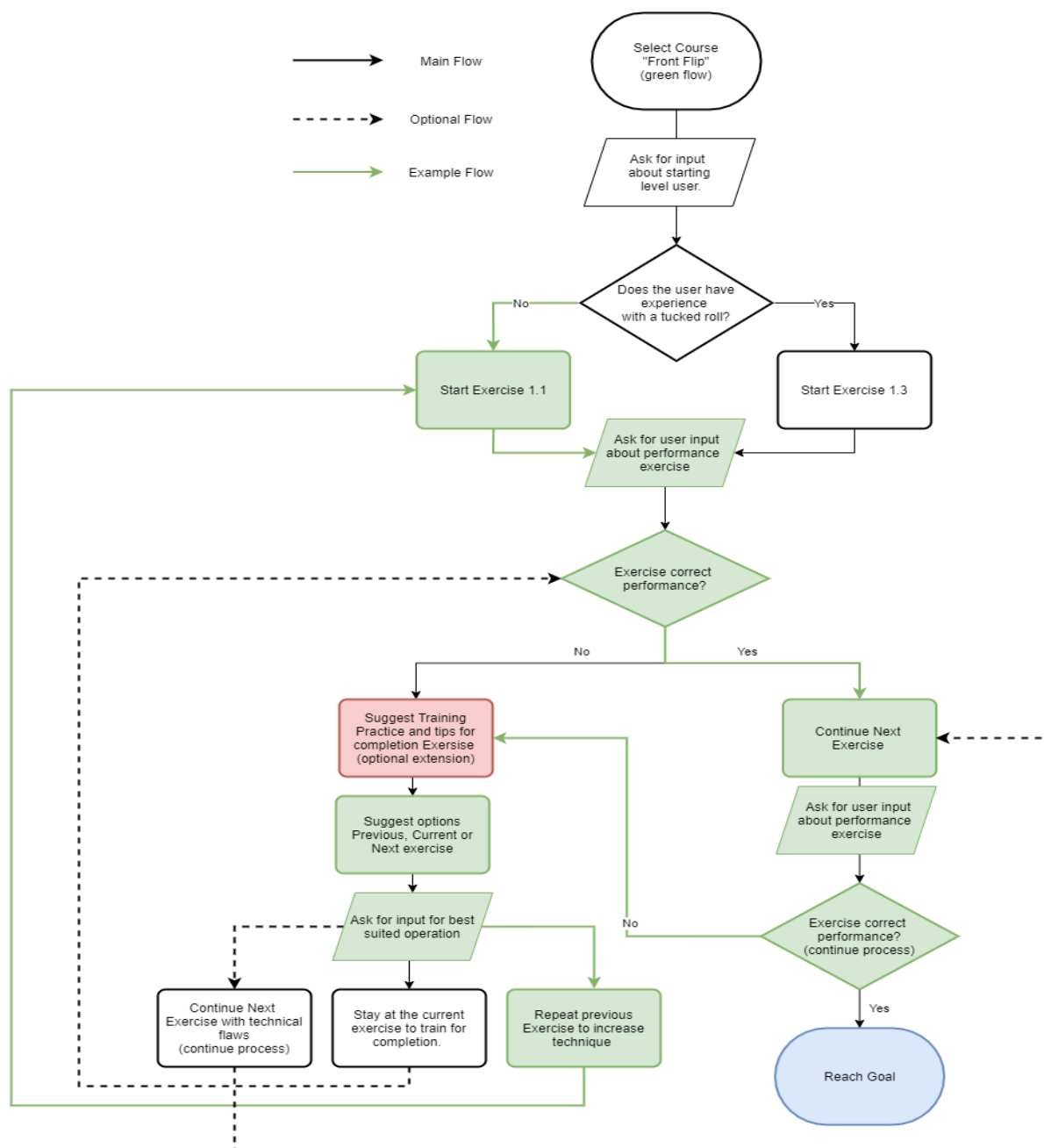


## Flowchart Diagram: A more clear view of following a course (continue, next, or repeat exercise)

In this flowchart you can see the whole process from starting a course, till reaching the goal of the course.

### \*Update version #3

I deleted the previous models of the abstract and starting User-Cases. In stead I've represented a flowcase-diagram where you can see a clear example of following a chosen course with a goal at the end. I added a green route to give an example of a chosen route in this course. The dotted lines are other optional operations. As you can see I deleted the alternate training-routes for now and give the optional training-tips Use-Case a red colour. This is to represent an extension of the application in a later stage.



## Wireframe

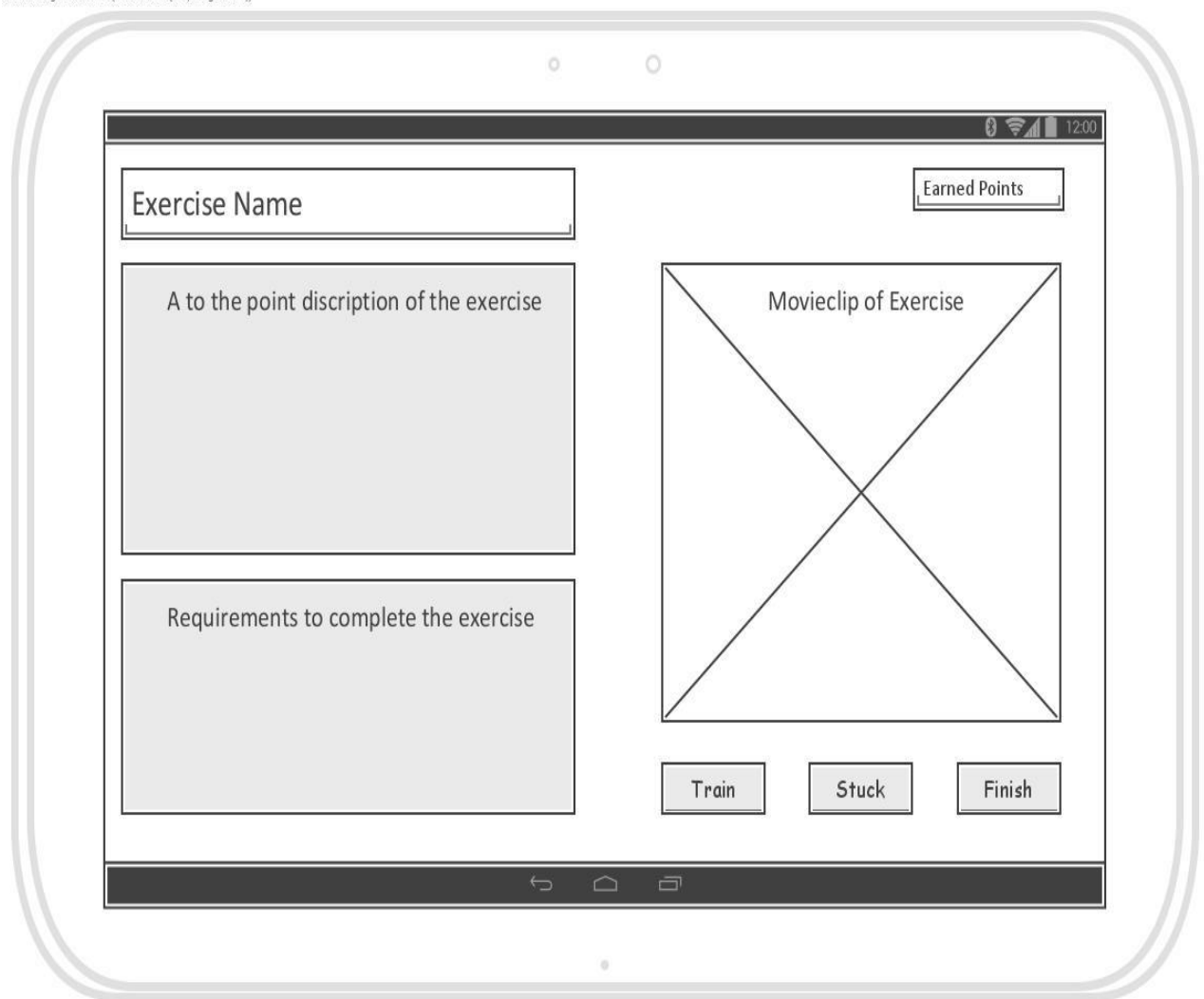
In the wireframe below you can see how an exercise is displayed. There is a balance between information, clarity and layout. The most important use-cases are displayed in the buttons below. If a gymnast finishes the exercise. He/She can continue to the next.

The button “Train” will execute the use-case for practice routines that can be used for the given exercise. The button “Stuck” will suggest the best option for the route.

This can be:

- Proceed to an alternative route.
- Go back to a previous exercise.

Visual Paradigm Professional (this is the first version of the software)



## Acceptance testing

Test id	TC-001
Test scenario	Click "Complete"
Test case description	Click on complete exercise in course
Test steps	<ol style="list-style-type: none"> <li>1. Click on the "Complete" button.</li> <li>2. Open pop-up window with validation requirements exercise.</li> <li>3. Click on "Complete" button.</li> </ol>
Test data	
Pre-condition	<ol style="list-style-type: none"> <li>1. User must have internet-connection.</li> <li>2. User must have valid login to execute test.</li> </ol>
Post-condition	<p>User gets the next exercise in the course in a new window.</p> <p>User gets points distributed to the databank.</p> <p>Completed exercise is added to users ArrayList completed exercises.</p>
Expected result	Successful opening of the next exercise in the course after validating requirements of the current exercise in a pop-up window.
Actual result	
Status	
Executed by	
Executed date	
Commands	

Test id	TC-002
Test scenario	Go back exercise
Test case description	Click on "Go back" button to go to the previous exercise in the course.
Test steps	<ol style="list-style-type: none"> <li>1. Click on "Go Back" button.</li> </ol>
Test data	
Pre-condition	<ol style="list-style-type: none"> <li>1. User must have internet-connection.</li> <li>2. User must have valid login to execute test.</li> </ol>
Post-condition	<p>Open previous exercise while closing current exercise in the same window.</p> <p>After completing previous exercise again, this exercise is not added to the ArrayList of the user. Since the exercise is already in it.</p>
Expected result	The previous instance of the exercise is opened. User is not distributed points.
Actual result	
Status	
Executed by	
Executed date	
Commands	

Test id	TC-003
---------	--------

Test scenario	Distribute points
Test case description	User gets points for completing an exercise
Test steps	<ol style="list-style-type: none"> <li>1. User clicks on "Complete" button.</li> <li>2. User answers the feedback questions.</li> <li>3. User clicks again on "Complete" in pop up window</li> </ol>
Test data	
Pre-condition	<ol style="list-style-type: none"> <li>1. User must have internet-connection.</li> <li>2. User must have valid login to execute test.</li> </ol>
Post-condition	User is distributed a number of points that is stored in the dataset.
Expected result	The dataset gets the distributed points in the correct parameter of the gymnast.
Actual result	
Status	
Executed by	
Executed date	
Commands	

Test id	TC-004
Test scenario	Verify login function
Test case description	User gets points for completing an exercise
Test steps	<ol style="list-style-type: none"> <li>1. Enter valid username.</li> <li>2. Enter valid password.</li> <li>3. Click on login</li> </ol>
Test data	
Pre-condition	<ol style="list-style-type: none"> <li>3. User must have internet-connection.</li> <li>4. User must have valid login to execute test.</li> </ol>
Post-condition	A green text with "Successful" login appears en user is taken to the welcome-page.
Expected result	The user is taken to the Welcome Page and the Gymnast that logged in can be used across the whole system till he/she logged out.
Actual result	
Status	
Executed by	
Executed date	
Commands	