

## Scenario

JZ Fitness is a fitness app designed and tuned towards restrictions of the covid-19 lifestyle. It is directed towards home overall athletic improvement while under the pandemic lockdown. JZ Fitness allows the user to download and register within the app, where, after registration, the user's registration details will be stored within our database. JZ Fitness is a completely user driven application where after the registration process, he/she is able to freely explore and navigate any options available by us. There is a built-in filter system open to the user's personal preferences. The user can specify any categories listed below and apply the filter method:

- Navigate to the filter system displayed within the app
- Check any boxes that apply to your preference
- Click apply at the bottom
- The results should correlate with user input
- Once results have loaded successfully, find what you want from the filtered results

## Adding Objects

Noun Phrases	Object	Attribute	Actors	Irrelevant
App, filter system, registration	Important Concepts			
User, app, database			Important Actors	
Process, restrictions, covid-19, home, application, pandemic, lockdown				Irrelevant noun phrases (part of explanations).
Apply, input, filter system, results		Features part of the app		
Dark/light theme, notification	Potential concepts			

system				
--------	--	--	--	--

## *Adding Classes*

<b>User</b>	This class consists of all user driven attributes related to the app like the registration process, filter system and any changes that can be made within the app by the user.
<b>Application</b>	This class consists of all features presented within the application, for example (filter system, progress tracking etc)
<b>Database</b>	This class consists of the database attributes for JZ Fitness such as the user inputs and user data

## *Adding Attributes*

All of the attributes shown within the classes section are primitive data types and cannot be productively broken down into sub classes.

- User: Primitive
- Application: Primitive
- Database: Primitive

## *Adding Relationships*

User to Database - The user “reports” to the database

Application to User The application “displays” the results to the user

Database to Application - The database “triggers” the application to open