

# *Pre-Task Questionnaire*

## *Design/Modeling Experience and Communication Skills*

### 1. Questions

Table 1 presents the questions of the pre-task questionnaire that are related to design as well as modeling experience and communication skills. It reports the id-number of the question, its description, experiment group (G: Graphical, T: Textual), and role of the participant (EXP: Explainer, REC: Receiver).

Q	Description	G	Role
1	How familiar are you with software design?	G T	EXP/REC EXP/REC
2	How familiar are you with UML modeling?	G	EXP/REC
3	How good are you in understanding UML models?	G	EXP/REC
4	How good are you in sense-making of UML models?	G	EXP/REC
5	How good are you in reading a textual document?	T	EXP/REC
6	How good are you in understanding a textual document?	T	EXP/REC
7	How good are you in sense-making of a textual document?	T	EXP/REC
8	How good are you in explaining your knowledge to others?	G T	EXP EXP
9	How good are you in building knowledge from conversing with others?	G T	REC REC

**Q:** Question, **G:** Group, **EXP:** Explainer, **REC:** Receiver

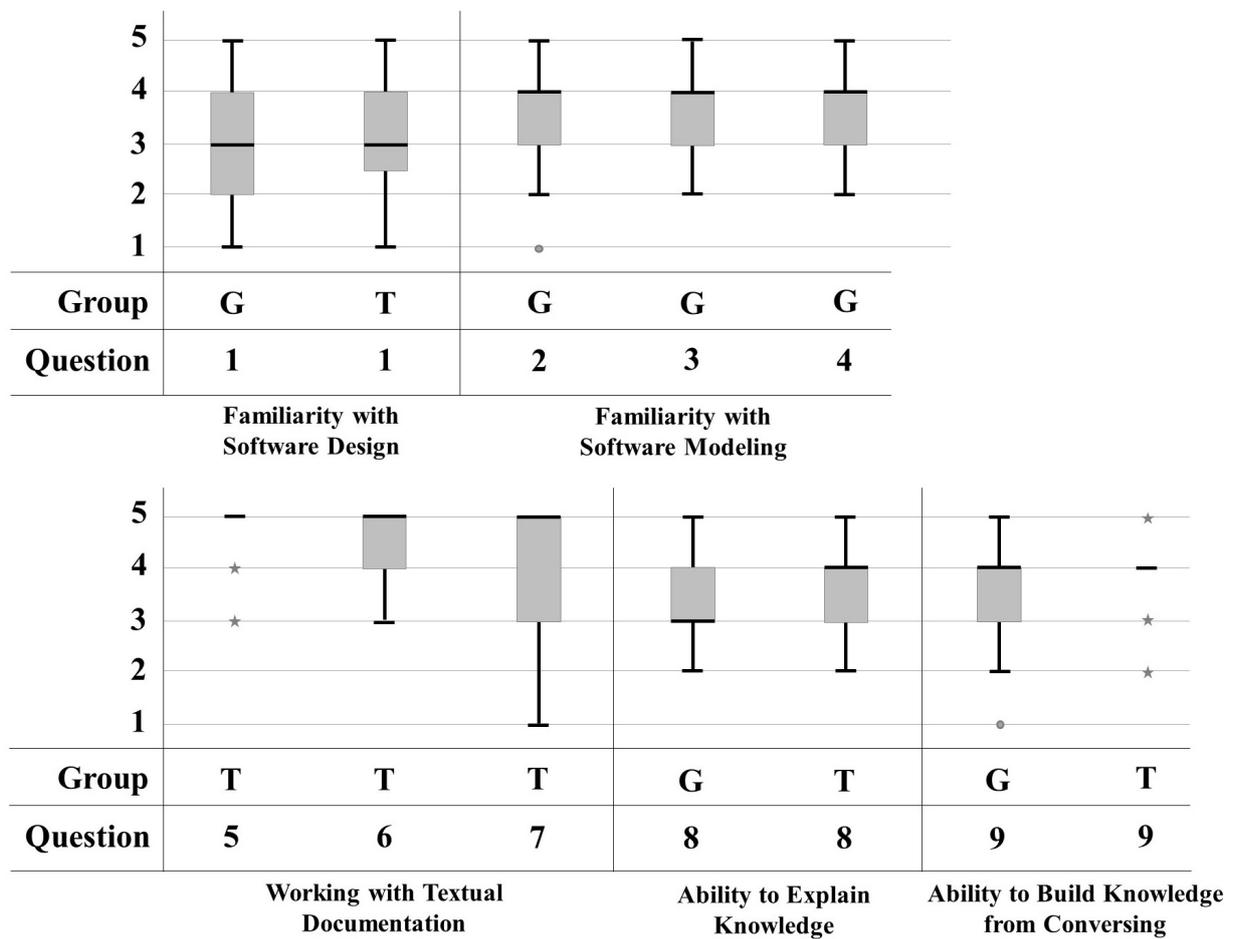
**Table 1:** Pre-Task Questionnaire: Design/Modeling Experience and Communication Skills

### 2. Results

The answers to the questions are presented as box-plots in Figure 1. We report the id-number of the question and the experiment group on the  $x$ -axis. The  $y$ -axis presents a 5-point *Likert* scale, where 1 is the lowest score and 5 is highest score. We find that:

- the participants are somewhat familiar with software design (median = 3).
- the participants are familiar with software modeling and good in understanding and sense-making of UML models (median = 4).
- the participants are very good in reading, understanding, and sense-making of textual documentation (median = 5).
- the *Explainers* in the group G are neither poor or good in explaining their knowledge (median = 3), while the *Explainers* in the group T are good in explaining their knowledge (median = 4).
- the *Receivers* of the two groups (G and T) are good in building knowledge from conversing with others (median = 4).

Moreover, by conducting the Independent Samples Mann Whitney U test, we observe that there is no statistically significant difference in the perceived design experience and communication skills of the participants between the two groups: G and T. Accordingly, we assume that perceived design experience and communication skills of participants are not influencing the results of the study.



**Figure 1:** Results of the Pre-Task Questionnaire on Design/Modeling Experience and Communication Skills

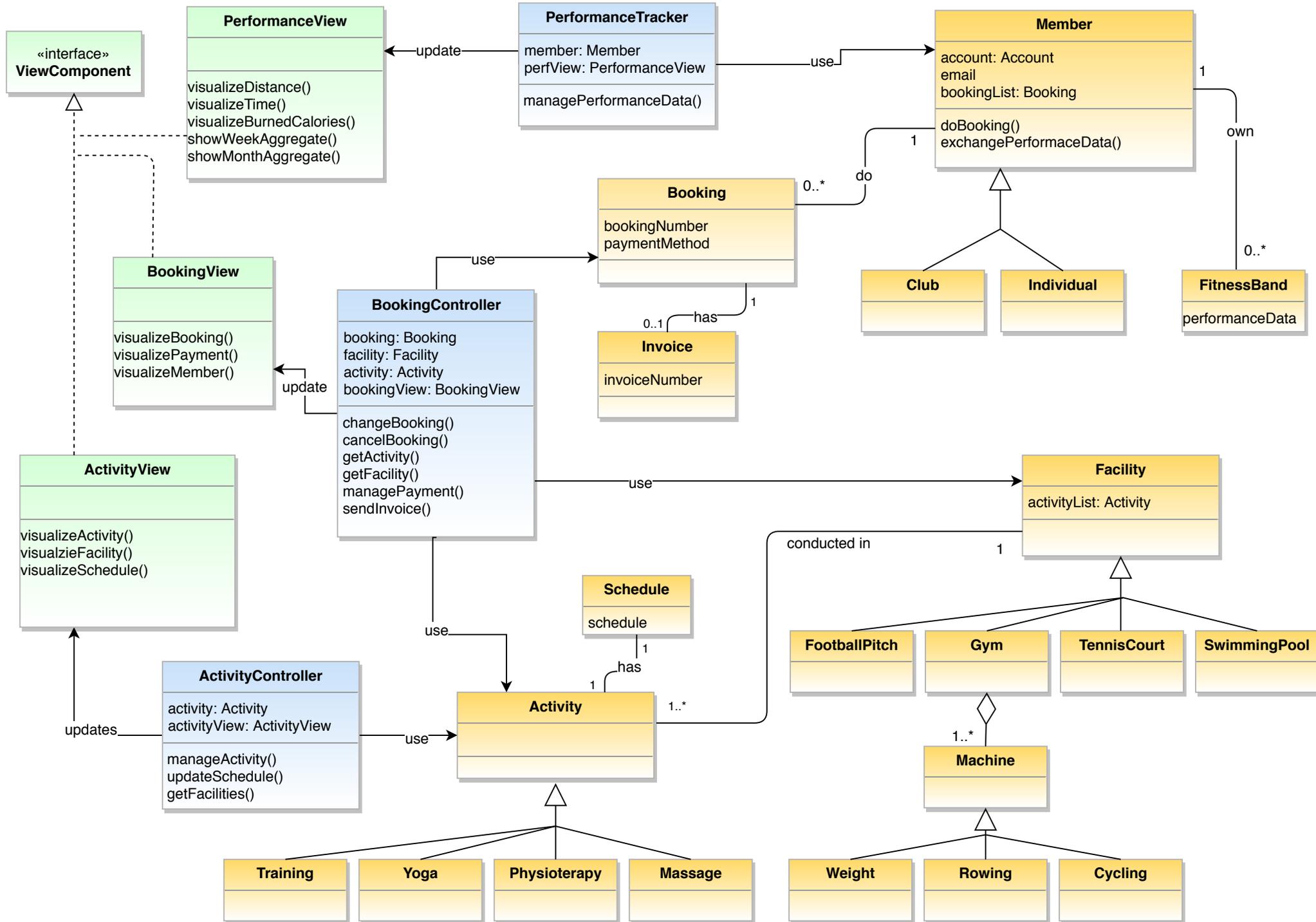
# The Fitness Paradise

The Fitness Paradise is offering a combination of sport activities. To this end, it has several facilities: an Olympic size swimming pool, a gym with several fitness machines (weight, rowing, cycles), a football-pitch, a tennis court.

The Fitness Paradise offers these facilities for rent for its clients. The Fitness Paradise accepts both individuals and clubs (e.g., companies and sport clubs) as members. The members have an account. In addition to the facilities, it also offers activities: training, yoga classes, physiotherapy and massages. The activities have a schedule that is updated every month. The Fitness Paradise wants to support booking via an app for smartphones.

The app presents the activity schedule for every day and week. Bookings can be paid with different methods. At the end of every month, an invoice is sent to members for any unpaid bookings over the past month. The system keeps track of the bookings. The app can also keep track of the fitness and performance of Fitness Paradise members. Examples of performance are for example: the distance cycled in a certain amount of time, calories burnt in a training session (but also aggregates over a week and month). For the fitness enthusiasts it is also possible to exchange such performance data which is collected by the member's fitness-band/watch.

# Structural Design Description of the Fitness Paradise App (Model-View Controller)



# Structural Design Description of the Fitness Paradise App

The software of the app is designed according to the Model-View Controller (MVC) design pattern.

## 1. ENTITIES

### I. Model

The model part includes of the following entities:

- **Member:** this entity provides information on user account, email and the list of bookings done by one member. Via this entity the member can do bookings and exchange performance data. The member can be a club or an individual.
- **FitnessBand:** this entity provides information on the performance data of the member.
- **Booking:** this entity provides information on the booking number and the payment method.
- **Invoice:** this entity provides information on the invoice number.
- **Facility:** this entity provides information on the list of activities conducted in one facility. The facility can be a football pitch, tennis court, swimming pool or gym. In the gym, there is at least one machine. The machine can be a: weight, rowing or cycling machine.
- **Activity:** This entity can be training, yoga classes, physiotherapy sessions, or massage sessions.
- **Schedule:** this entity provides information on the schedule of the activities.

### II. Controller

The controller part includes the following controllers:

- **Booking Controller:** this controller uses the data from the Booking, Facility and Activity entities. This controller updates the Booking View. The functionality of this controller is to change the booking, cancel the booking, get a facility, get an activity, manage the payments, and send the invoices.
- **Activity Controller:** this controller uses the data from the Activity entity. This controller updates the Activity View. The functionality of this controller is to manage the activities, update the schedule of the activities, and get the facilities where the activities are taking place.
- **Performance Tracker:** this controller uses the data from the Member entity. This controller updates the Performance View. The functionality of this controller is to manage performance data of the member.

### III. View

The view part includes three views that each implement the *View Component* interface:

- **Booking View:** this view provides options to visualize the booking, visualize the payment and visualize the member.
- **Activity View:** this view provides option to visualize the activity, visualize the facility and visualize the schedule of the activities.
- **Performance View:** this view provides option to visualize the covered distance and time, the burned calories, but also aggregates over a week and month.

## 2. RELATIONS

- **Member:**
  - One Member can do zero or many Bookings.
  - One Member owns zero or many FitnessBands.
  - One Member can be:
    - Club
    - Individual
  - The Member entity is used by the Performance Tracker.
- **FitnessBand:**
  - One FitnessBand belongs to one Member.
- **Booking:**
  - One Booking is done by one Member
  - One Booking has zero or one Invoice.
  - The Booking entity is used by the Booking Controller.
- **Invoice:**
  - One Invoice is related to one Booking.
- **Facility:**
  - One Facility can host one or many Activities.
  - One Facility can be:
    - Football pitch
    - Tennis court
    - Swimming Pool
    - Gym, which consists of one to many Machines, and one Machine can be:
      - Weight
      - Rowing
      - Cycling
  - The Facility entity is used by the Booking Controller.

- **Activity:**
  - One Activity has one Schedule.
  - One Activity is conducted in one Facility.
  - One Activity can be:
    - Training
    - Yoga class
    - Physiotherapy session
    - Massage session
  - The Activity entity is used by the Booking Controller.
- **Schedule:**
  - One Schedule belongs to one Activity.
- **Booking Controller:**
  - This controller uses the Booking, Activity and Facility entities.
  - This controller updates the Booking View.
- **Activity Controller:**
  - This controller uses the Activity entity.
  - This controller updates the Activity View.
- **Performance Tracker:**
  - This controller uses the Member entity.
  - This controller updates the Performance View.
- **Booking, Activity and Performance Views:**
  - These controllers implement the interface “View Component”.