

### **Tabellen:**

TABLE User (UserId int, UserName varchar(255), Feeling int, lastFeelingDate DATETIME)

TABLE Activity (ActivityId int, ActivityName varchar(255))

TABLE DoneActivities (UserId int, ActivityId int, Count int, Status int, lastStatusDate DATETIME)

TABLE FutureActivities (UserId int, ActivityId int, Status int, lastStatusDate DATETIME)

### **class Datenbank mit Methoden:**

#### **Verbindung auf- und abbauen**

private void connectDatenbank() throws Error

→ wird beim Erstellen eines Datenbank-Objekts automatisch aufgerufen

@ Exception

- Verbindung zur Datenbank kann nicht aufgebaut werden

void disconnectDatenbank()

→ bei Beenden des Skills

#### **User abfragen und ggf. anlegen**

int getUser(String UserName) throws Error

→ nach connectDatenbank-Methode

@ Exception

- Fehler vom MySQL Server zurückgegeben

@ String UserName

- erhalten durch Alexa

@ return

- UserId → UserName gefunden

- None → UserName nicht gefunden

int createUser(String UserName) throws Error

→ nach getUser-Methode returned None

@ Exception

- Fehler vom MySQL Server zurückgegeben

@ String UserName

- erhalten durch Alexa

@ return

- UserId

#### **User Feeling-Feld lesen und schreiben**

int getUserFeeling(int UserId) throws Error

→ vor Alexa-Abfrage: Wie geht es dir?

@ Exception

- Fehler vom MySQL Server zurückgegeben

@ int UserId

- erhalten durch getUser-Methode bzw. createUser-Methode

@ return

- 1 → „grün“

- 0 → „gelb“

- -1 → „rot“

- None → Feeling-Feld leer oder lastFeelingDate nicht heute

void setUserFeeling(int UserId, int Feeling) throws Error

→ nach Alexa-Abfrage: Wie geht es dir?

@ Exception

- Fehler vom MySQL Server zurückgegeben

@ int UserId

- erhalten durch getUser-Methode bzw. createUser-Methode
- @ int Feeling
  - „grün“ → 1
  - „gelb“ → 0
  - „rot“ → -1

### Activities Status-Feld lesen

int getFutureStatus(int UserId, String activity) throws Error

→ nach Nutzer-Antwort auf Alexa-Abfrage: Was machst du so?

- @ Exception
  - Fehler vom MySQL Server zurückgegeben
- @ int UserId
  - erhalten durch getUser-Methode bzw. createUser-Methode
- @ String activity
  - erhalten durch Alexa-Abfrage
- @ return
  - 1 → „grün“
  - 0 → „gelb“
  - -1 → „rot“
  - None → Status-Feld leer oder lastStatusDate nicht heute

int getDoneStatus(int UserId, String activity) throws Error

→ nach getFutureStatus-Methode returned None

- @ Exception
  - Fehler vom MySQL Server zurückgegeben
- @ int UserId
  - erhalten durch getUser-Methode bzw. createUser-Methode
- @ String activity
  - erhalten durch Alexa-Abfrage
- @ return
  - 1 → „grün“
  - 0 → „gelb“
  - -1 → „rot“
  - None → Status-Feld leer

### Activities Status-Feld schreiben

void setFutureStatus(int UserId, String activity, int status) throws Error

→ nach Nutzer-Antwort auf Alexa-Abfrage: Freust du dich? bzw. Ist das was für dich?

- @ Exception
  - Fehler vom MySQL Server zurückgegeben
- @ int UserId
  - erhalten durch getUser-Methode bzw. createUser-Methode
- @ String activity
  - erhalten durch Alexa-Abfrage
- @ int status
  - erhalten durch Alexa-Abfrage
  - 1 → „grün“
  - 0 → „gelb“
  - -1 → „rot“

void setDoneStatus(int UserId, String activity, int status) throws Error

→ nach Nutzer-Antwort auf Alexa-Abfrage: Hattest du Spaß?

- @ Exception
  - Fehler vom MySQL Server zurückgegeben
- @ int UserId

- erhalten durch getUser-Methode bzw. createUser-Methode
- @ String activity
  - erhalten durch Alexa-Abfrage
- @ int status
  - erhalten durch Alexa-Abfrage
  - 1 → „grün“
  - 0 → „gelb“
  - -1 → „rot“

### Aktivität vorschlagen

String getActivity(int UserId, int tries) throws Error

- @ Exception
  - Fehler vom MySQL Server zurückgegeben
- @ int UserId
  - erhalten durch getUser-Methode bzw. createUser-Methode
- @ int tries
  - Anzahl der Versuche Aktivitätsvorschlag zu geben
- @ return
  - Aktivität: Name der vorgeschlagenen Aktivität
  - None: User hat keine tries-viele Aktivitäten in DoneActivities eingetragen, die „grün“ oder „gelb“ sind

### FutureActivities auslesen

String[] getFutureActivities(int UserId) throws Error

→ nicht in Ablauf-Skizze berücksichtigt (abfragen ob gemacht wurde, ...)

- @ Exception
  - Fehler vom MySQL Server zurückgegeben
- @ int UserId
  - erhalten durch getUser-Methode bzw. createUser-Methode
- @ return
  - Aktivitäten: Array mit Namen der Aktivitäten
  - None: User hat keine Aktivitäten in FutureActivities eingetragen, die heute sind

### FutureActivity löschen oder in DoneActivities mit Status umschreiben

void deleteFutureActivity(int UserId, String activity) throws Error

→ nicht in Ablauf-Skizze berücksichtigt (Aktivität ist nicht mehr für heute geplant)

- @ Exception
  - Fehler vom MySQL Server zurückgegeben
- @ int UserId
  - erhalten durch getUser-Methode bzw. createUser-Methode
- @ String activity
  - erhalten durch Alexa-Abfrage

void moveFutureActivityToDone(int UserId, String activity, int status) throws Error

→ nicht in Ablauf-Skizze berücksichtigt (geplante Aktivität wurde gemacht)

- @ Exception
  - Fehler vom MySQL Server zurückgegeben
- @ int UserId
  - erhalten durch getUser-Methode bzw. createUser-Methode
- @ String activity
  - erhalten durch Alexa-Abfrage
- @ int status
  - erhalten durch Alexa-Abfrage
  - 1 → „grün“

- $0 \rightarrow \text{„gelb“}$
- $-1 \rightarrow \text{„rot“}$