Memorial University of Newfoundland **Fall Semester 2018**
Computer Science 2005 **Group 12**

Assignment 3

This assignment instructs us to "prioritize features of the system you are supposed to implement." As the application being developed is an activity logger app, we believe two of the most crucial parts of the app — its essence — are the user interface and the backend data storage and retrieval. Because of this, in this assignment to "create architecturally significant design," we have focused on developing the Java-based UI and engineering an efficient database structure for the backend that will enable easy retrieval of large data sets for a particular user.

We have decided to use a relational database and are using SQLite to implement this. SQLite was chosen for developmental simplicity and availability of a JDBC-SQLite driver.

A prototype of the relational database schema is provided in the tables below, with some sample data.

We also include some sample queries that we may run for various parts of the app.

Users

ID	EmailAddress	Name	DOB	Sex
1	jdoe@acme.com	Johnathan Doe	23-01-1992	M
2	goulding@contoso.com	Annalise Goulding	03-08-1947	F
3	neil@google.com	Neil Flemming	21-11-1989	M
4	avery.smith@mac.com	Avery Smith	12-12-1973	M

Table 1: The `User` table will keep track of what users we have in our system. The `ID` column is the primary key and will contain unique values for each user. This key will be used to relate rows in this table to rows in other tables.

Passwords

ID	UserID	PassHash	DateChanged
1	1	4bd44ba402231f35390c1ae3b76f0154	11-10-2017 13:17:08
2	1	7f61f924bcea42434fee55f3bdd05c5d	11-10-2017 14:01:56
4	3	18f8caa5842ff58d4a40701a1d377966	12-01-2018 07:23:45
3	2	634a3583084e873ccf31d58622707219	18-02-2018 08:32:45
5	3	75c625509efe6deca63084f971067634	19-10-2018 17:45:45

Table 2: We store the passwords in a separate table from the users so that we can maintain a password history of arbitrary length and can therefore enforce password history requirements when users change passwords. Again, the `ID` column is the primary key. The password is associated with a user by the second column, `UserID`. This approach also helps users remember their password, *f.e.*, if they type the old password, we can match that and remind them that they changed it past week.

Workouts

ID	UserID	WOType	Date	Duration	kCal
1	1	Run	18-04-2018	1:37:04.60	1,107
2	2	Weights	01-11-2018	1:59:36.78	3,114

Table 3: We log workouts to this table. As before, `ID` is the primary key and rows are associated with users by the `UserID` column. Entries in this table may be updated through the user interface if a user would like to correct data provided by a device's sensor, or if data is being provided to the app incrementally during a workout rather than in a lump-sum import.

Friends

ID	Sender	Receiver	SentDate	ConfirmedDate
1	1	2	18-02-2018 07:43.13.68	18-02-2018 07:45:12.34
4	3	4	12-08-2018 19:23:17.32	`NULL`

Table 4: This table keeps track of friends and pending friend requests. Confirmed friend requests will have a confirmation date not equal to `NULL`. A confirmation date equal to `NULL` indicates a friend request that has been sent but not accepted or declined yet. If a user declines a friend request or deletes a friendship, the corresponding row will be removed from the table. Here, rows 2 and 3 have been removed, indicating cancelled requests or terminated friendships.

Sample Query 1. We want to get all passwords for the user with email address neil@google.com.

```
SELECT Passwords.PassHash, Password.DateChanged from Users,
    Passwords WHERE Users.EmailAddress = 'neil@google.com';
```

This returns

PassHash	DateChanged
18f8caa5842ff58d4a40701a1d377966	12-01-2018 07:23:45
75c625509efe6deca63084f971067634	19-10-2018 17:45:45

Sample Query 2. We want the most recent social post for Annalise Goulding. We will definitely have this user's ID on hand as this is how we identify and keep track of users for everything. So we then query the following.

```
SELECT Status FROM SocialStatus WHERE UserID = 2 ORDER BY
    PostDate DESC LIMIT 1;
```

This returns the posting from 01-11-2018 14:48:32.12, "New 1RM in the gym today!!"

Devices

ID	UserID	MACAddress	DeviceName	DeviceModel	SerialNo
1	2	3d344f1294ca	Annalise's Watch	Apple Watch Series 2	K5J2WD4K
2	1	56dc8a3283c4	Johnathan Doe's FitBit	FitBit Charge 2	H45DC2456S
3	1	24c1c9ada464	John's Apple Watch	Apple Watch Series 3	KW5DC2NW

Table 5: Entries in this table are associated with a user's device(s). Here we record things like hardware MAC address, device model, device name. As before the `ID` column is the primary key for the table and `UserID` is how a device is associated with a user.

SocialStatus

ID	UserID	Status	PostDate
1	2	Had a great run along the lakeshore this morning and saw the most beautiful sunrise!	18-08-2017 09:12:09.98
2	4	First time in the pool this month; out of shape!	01-11-2018 13:04:12.92
3	2	New 1RM in the gym today!!	01-11-2018 14:48:32.12

Table 6: This table records optional status updates that users may post for their friends to see. Typically these will be about their workouts. Columns `ID` and `UserID` are as before.