CS 442 – MOBILE APPLICATION DEVELOPMENT

FINAL REPORT **DATE: 12/06/2016**

Project Name: Smart Bar

Group members: Adeeb Ahmed, Shreyas Moudgalya, Sumedha Gupta, and Prasanna Shanmuganathan

1. Project's general description

App category: Fitness and Health

This app will be designed to be a companion app to a “Smartbar”. What is Smartbar? It’s a weightlifting bar that can record real time data about a weightlifters workout. The mobile app would take data produced by the smart bar and seamlessly keep track of a lifters workouts and overall progress.

2. Feature/requirement list

1. Onboarding - OAuth sign up or basic authentication
2. Profile page - Essentially homepage where user will see their statistics, include steps walked
3. Data visualization - Workout analytics page, shows graphical view of progress with some filter features
4. Journal page - Calendar or List View to see when user worked out with someone, and what workouts the user did. Notes can be added, reminders
5. Workout with friends - Set time to workout with friends, integrates with calendar, add new friends and can view friends profile.

The user should be able interact and do some action within each page, no static pages.

3. UI transition flow

→ Journal → Notes
Sign up/Login screen → Profile Page → Data visualization
→ Workout with friends → View friends profile

4. Elevator Pitch

Our team has given elevator pitch to TA Radha and explained her all the features of our app. She has also given us some suggestions, which we have included in our features list. She was impressed with the project idea and approved it.

5. UI Design and Screen Layouts

Layouts:

a. **Main Screen:**
Presents login and sign up option for the users.

b. **Profile Screen:**
Shows the profile of the user and contains all the modules related to user workouts. E.g. Journal Module, Data Visualization, Workout With Friends and Add new friends.

c. **Connect to smartbar screen (NEW FEATURE)**
This screen will allow the user to connect to their smartbar via bluetooth, for visual purposes only. No real smartbar exists at this time and mock data will be used.

d. **User Journal Screen:**
Provide the workout history of the user and allow the user to switch to the calendar view of the workouts. It also allows the user to search workouts on basis of a exercises performed.

e. **Calendar View Screen:**
Allow the user to see all the past and future workouts highlighted and allow the user to go to a particular date.

f. **Add notes Screen:**
On click of a particular date from the calendar view the user is able to see all the workouts on that date and can add or edit notes for the same.

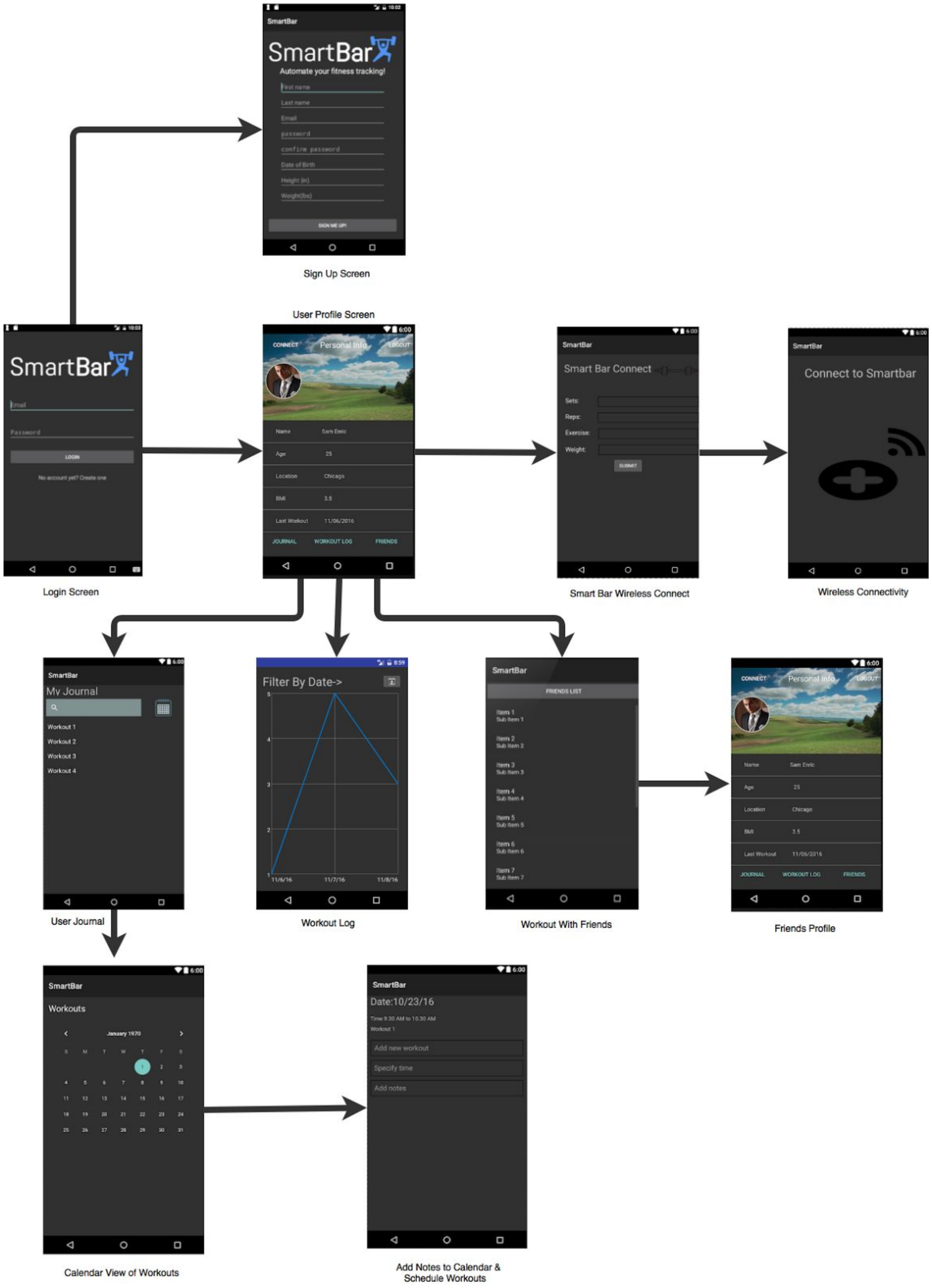
g. **Data Visualization Screen:**
Workout analytics page, shows graphical view of progress with some filter features.

h. **Workout With Friends Screen:**
Allow the user to set workouts with friends

i. **Add Friends:**
Allow the user to add new friends by providing the name.

The screen layouts and their transition flow is shown below:

- **Transition Flow**



6. Individual work assigned and completed

Prasanna Shanmuganathan :

Week 1

1. Designed the basic UI sketches for all the screens.
2. Designed the layouts for the Journal module, which includes the Journal page, Calendar View, setting future workouts and adding, notes to the calendar.
3. Designed the Use case diagram for the application.(Fig.1)
4. Designed the database structure and diagram for journal module.(Fig.2)
5. Created class files for workout and exercise entity.
6. Implemented User Journal Activity and User Journal Fragment for displaying the list of workouts.
7. Implemented Workout DbHelper and Workout Reader Contract classes for displaying the workout history using SQLite.
8. Working on Implementation of Calendar View Activity to display the past and upcoming workouts the user has set.

Week 2

1. Updated the use case diagram to add the new use case for Smartbar connect feature.(Fig.1)
2. Updated the database diagrams to include all the tables, their attributes,references and constraints.(Fig.2)
3. Designed the transition flow diagram for all the screens in the app.(Fig.a)
4. Designed the UI and layout pages for the new screens for connecting the app to the smartbar.
5. Implemented DbHelper and Reader Classes for the workout, exercise and calendar tables.
6. Implemented the feature to show the list of past workouts on the user journal page.
7. Implemented fragment User Journal Fragment to contains the list of workouts from the database.
8. Integrated SQLite to retrieve the data from the workout table and display it on the screen.
9. Updated class files for workout and exercise entity to add new attributes.
10. Created class file for calendar notes entity which includes attributes such as workout date, workout id ,notes, workout time.

Week 3

1. Designed the Calendar View Screen on which the calendar layout of the workouts will be shown.
2. Created a Fragment based layouts for user journal fragment and the calendar fragment.
3. Configured and implemented Caldroid Calendar View on the Calendar screen.
4. Implemented DatePicker on the Calendar Screen.
5. Synchronize the DatePicker with the Caldroid calendar.
6. Implemented the highlighting of dates in the calendar view for the past and future workouts.
7. Implemented base framework for the database.
8. Met with team to resolve the merge conflicts on the git repository and updated the repository.
9. Designed the test cases layout.
10. Done testing for the Journal Screen and the Calendar Screen and added test cases for the same.
11. Updated database design to add modify some attributes and to add some tuples.

Week 4

1. Done POC on Firebase and MLab to implement a synchronized backend as SQLite was not centralized.
2. Integrated Firebase in to the app.
3. Mapped the SQLite tables to the tree like structure in firebase.
4. Interacted with firebase in the User Journal Fragment to fetch all the workouts from firebase, Calendar Fragment to get the workout dates to highlight the past and future workouts and in Notes Fragment to fetch and update the notes for the workouts.
5. Implemented Splash Screen for the app.
6. Added search functionality in the workout history.
7. Implemented Notes Fragment to get all the notes from the firebase for a particular workout and allow the user the user to edit and update the notes.
8. Used Expandable List view to display workout history.
9. Implemented reminders features to set reminders for future workouts.
10. Served as Phase Manager.

Adeeb Ahmed - Phase manager

1. Came up with app idea and overall vision.
2. Designed UI and logo for smartbar and signup and login pages.
3. Designed database table structure for login and sign up
4. Setup github workflow, led team by dividing work

5. Implemented functionality signup and login using sqlite3
6. Added connect to smartbar new screen

Week 2:

1. Met with group members and designed database
2. Implemented database using sqlite3 (android.database.sqlite)
3. Created mock data generation sql files (create, insert, delete)
4. Using executeSQLscript() to interface w database from java
5. Connecting app with database on login, signup, and smartbar pages

Week 3:

1. Fully implemented login function
2. Made modifications to database
3. Working with sqlite3 and ADM
4. Implementing signup functionality and adding test cases
5. Helped with documents and UI flow
6. Served as Phase manager

Week 4:

1. Fully implemented login tests and functionality
2. Fully implemented signup tests
3. Fully implemented connect to smartbar pages
4. Setup ui tests using espresso
5. Switched backend from slqite to firebase

Sumedha Gupta:

1. Designed User interface sketches for the App flow.
2. Designed the Layouts for the Data Visualization module which include Workout analytics page, shows graphical view of progress with some filter features.
3. Used Graph feature in android to implement data analytics(Workout info).
4. Created Classes to manipulate the content in the graph based on the workout data.
5. Working on modifying the graph data based on DATE filters using the Calendar view.
6. Implemented Data fetch for the manipulation of graph.
7. Enhanced the structure of Profile Page using Custom View(UI)
8. Implemented DbHelper and FriendReader classes for displaying the ProfileInfo and FriendProfile using SQLite.
9. Generated Graph based on sets and reps of the User logged in
10. Implemented the entire functionality on Firebase Platform.
11. Created Entity classes to retrieve data objects from FireBase.
12. Improvised design on the graph structure.

13. Manipulation of the graph was filtered based on dates.
14. Improvised the graph by fetching data for 5 days.
15. Graph view design was modified for the proper visualization of progress
16. Fully implemented Graph generaton test
17. Fully implemented Profile Page Generation
18. Workout progress was screen on Graph view page
19. Designed the test cases.

7. Progress

1. Contemplated the no. of screens, features to be added on each screen and UI designing.
2. Designed basic UI sketches for most of the app screens.
3. Thought about some extra features to add e.g. posting your workout details on Facebook, twitter, changing the unit of measure of the workout, privacy settings to set the privacy of the profile whether it will be private, shared with friends or public.
4. UI Design of almost all the screens is completed.
5. Started basic implementations of features.
6. Completed the basic functionality for most of the features.

8. FAQ:

1. **Will the app be specifically for the smart bar?**
   Yes users could also input the data manually in the “workout” feature if they do not have a smartbar.
2. **How to import the data produced by the smart bar?**
   For the purpose of this assignment we will use mock data so no actual smartbar will be required.

9. Use Case Diagram

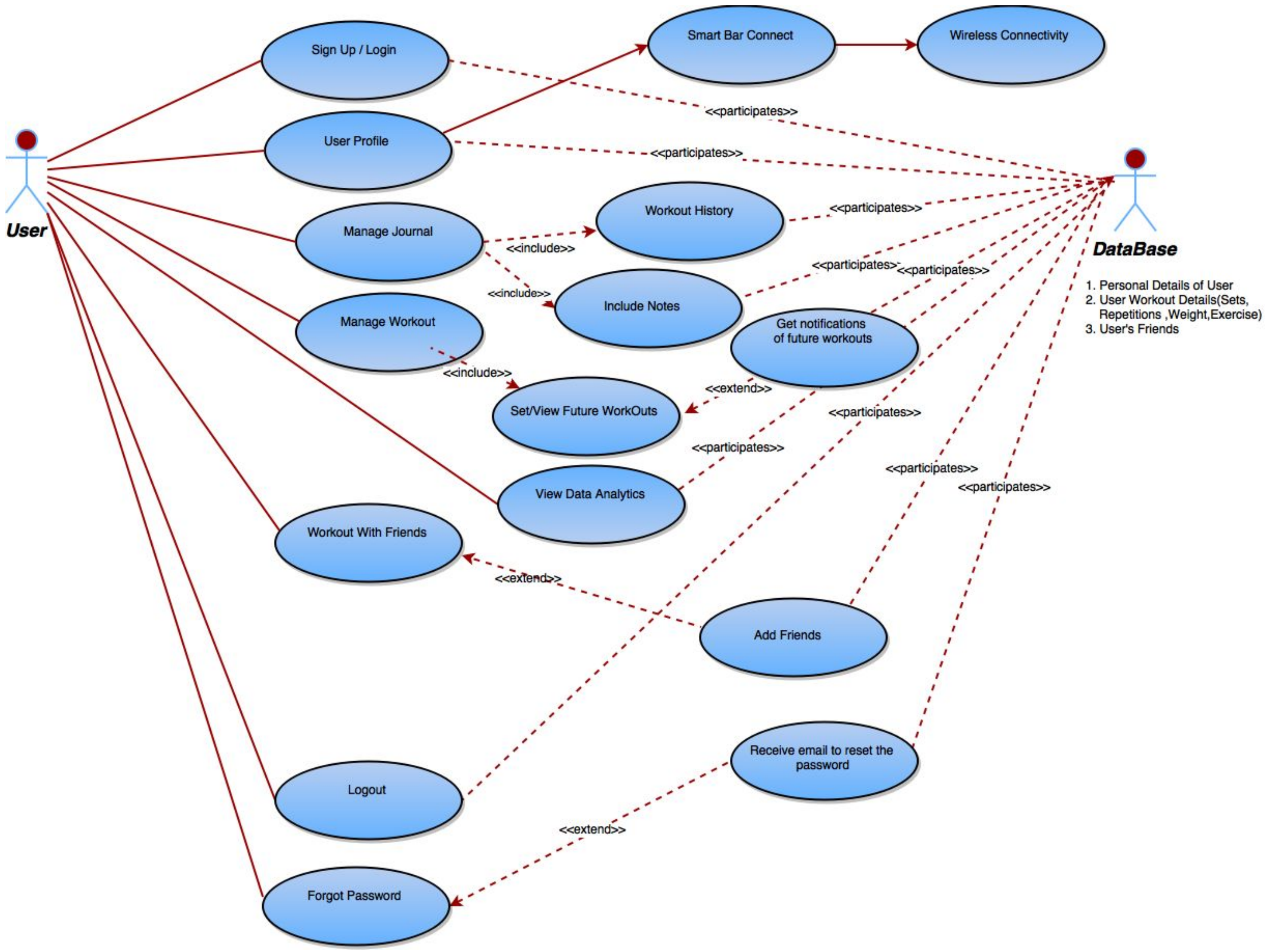


Fig.1

10. Database Diagram for Journal Module

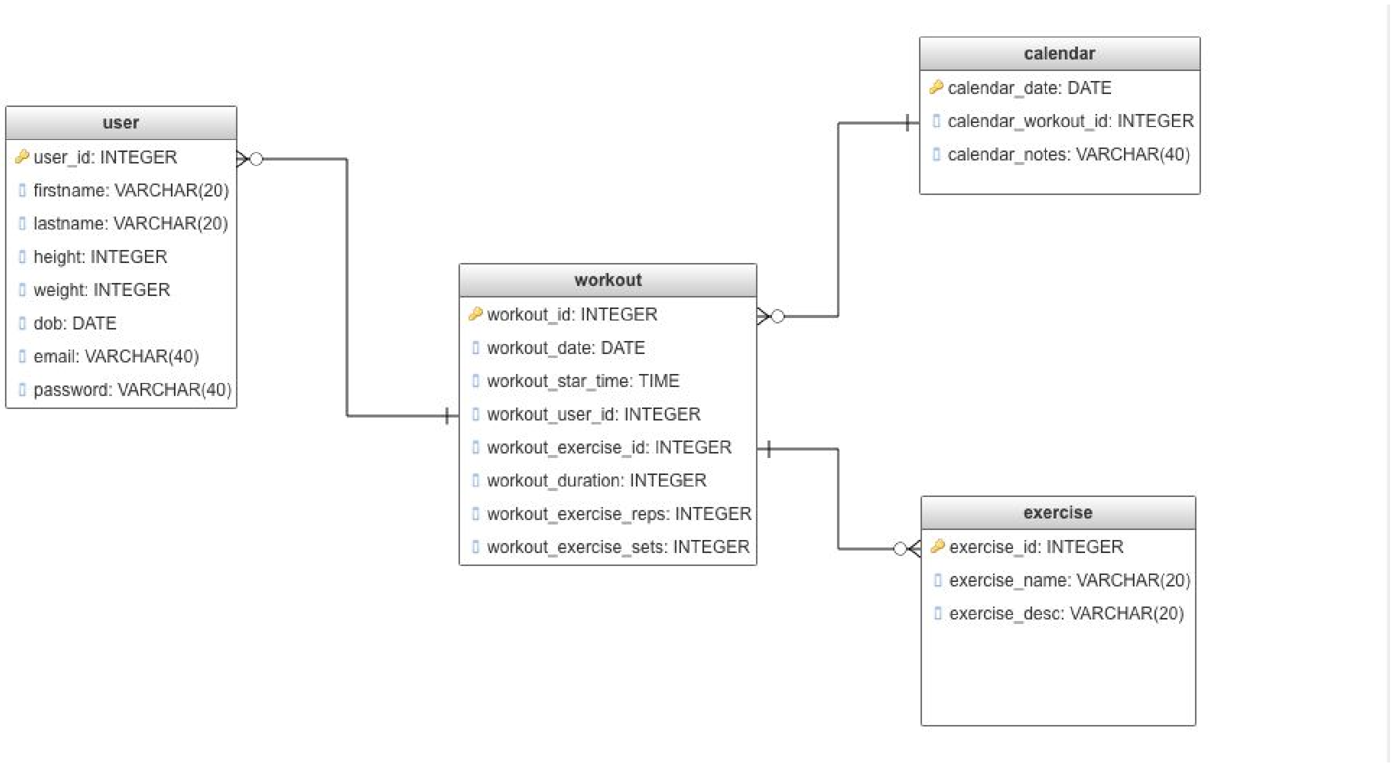


Fig.2

Database schema:

```
CREATE TABLE IF NOT EXISTS `Users` ( `userid` INTEGER PRIMARY KEY AUTOINCREMENT,
`firstname` TEXT NOT NULL, `lastname` TEXT NOT NULL, `height` INTEGER NOT NULL, `weight`
INTEGER NOT NULL, `dob` TEXT NOT NULL, `email` TEXT NOT NULL, `password` TEXT NOT
NULL );


CREATE TABLE "Workouts" ( `workoutid` INTEGER PRIMARY KEY AUTOINCREMENT, `date`
TEXT NOT NULL, `start` TEXT NOT NULL, `end` TEXT NOT NULL, `sets` INTEGER NOT NULL,
`reps` INTEGER NOT NULL, `weight` INTEGER, `exercise` TEXT NOT NULL, `fk_userid` INTEGER
NOT NULL, FOREIGN KEY(`fk_userid`) REFERENCES `Users`(`userid`) );
```

```
CREATE TABLE IF NOT EXISTS `Notes` ( `noteid` INTEGER PRIMARY KEY AUTOINCREMENT,
`notes` TEXT, `calanderdate` TEXT NOT NULL, `fk_workoutid` INTEGER NOT NULL, FOREIGN
KEY(`fk_workoutid`) REFERENCES `Workouts`(`workoutid`) );
```

```
CREATE TABLE IF NOT EXISTS `Friends` ( `friendid` INTEGER PRIMARY KEY
AUTOINCREMENT, `fk_userid` INTEGER, FOREIGN KEY(`fk_userid`) REFERENCES Users(userid)
);
```