

TODO List Manager on Steroids

Use Case Document

Team 3.09

1. Introduction

The goal of the project is to extend the existent TODO List Application on Android to a Web application and to provide sync functionalities between the Android and the Web application, thereby enhancing the user-experience.

2. Use Cases

2.1. Accounts

The mobile application user and the web application users have different use cases due to the distinct architecture with local and central databases.

2.1.1. Web Application

On the web platform the user has the following use cases regarding account information:

- Create a new account: this checks the availability of the specified username in the central database.
- Login: verifies information in central database.
- Logout
- Edit Account Information
- Delete Account

A use case diagram is provided below.

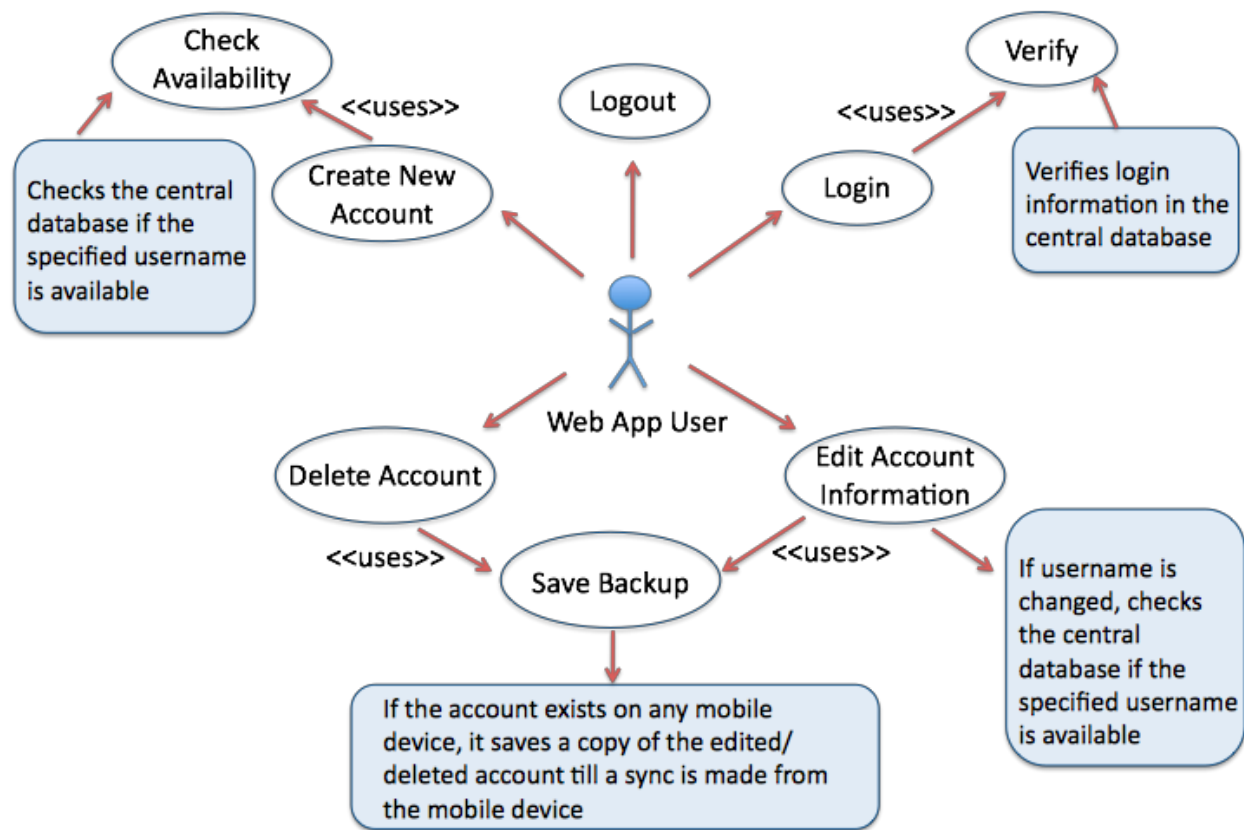


Figure 1: Accounts related use case diagram for web app user

Editing account information or deleting the account will not reflect on the mobile app till an implicit sync is made or the app is re-launched. In this time the user may make local changes to the account on the mobile app that has been deleted or edited in the central database. For this reason the changes are backed up till a sync is made.

2.1.1.1. Scenario: Creating a New Account from Web App

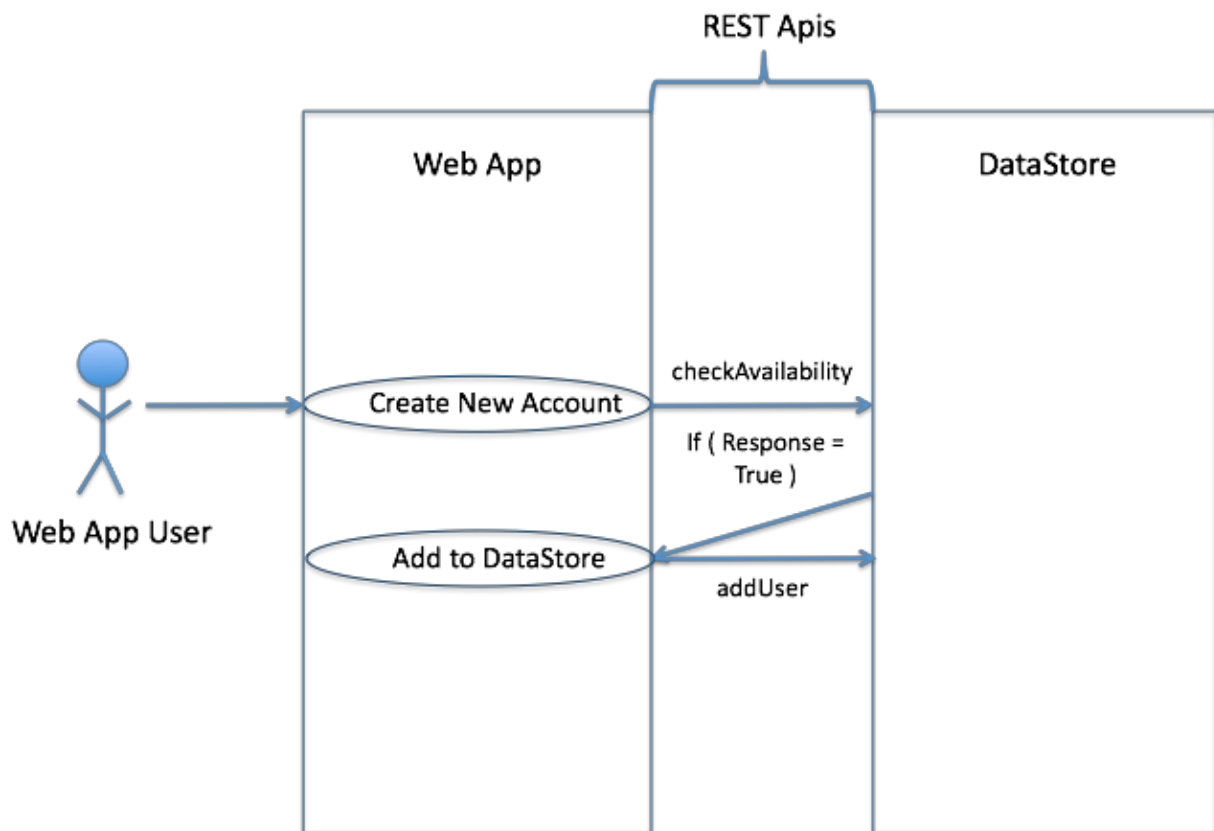


Figure 2: Creating new account from web app.

When the user tries to create a new account through the web application, the availability is checked in the DataStore. If the username is taken, the user is provided with an error message, else the user account is created in the DataStore.

2.1.1.2. Scenario: Login from Web App

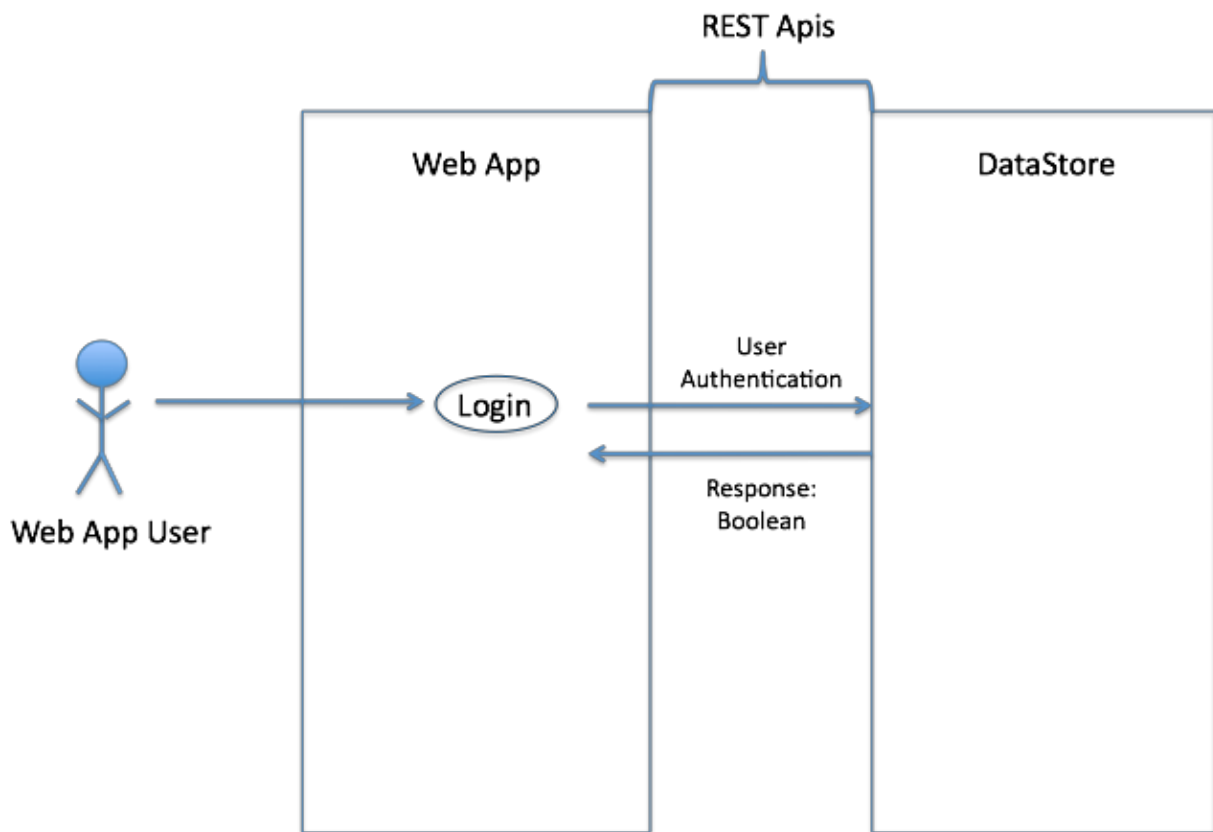


Figure 3: Login from web app.

Login from the web application requires user authentication from the DataStore. The ultimate goal is to have the user login with their Google accounts so all the authentication will be taken care of with the Google Apis.

2.1.1.3. Scenario: Edit/Delete from Web App

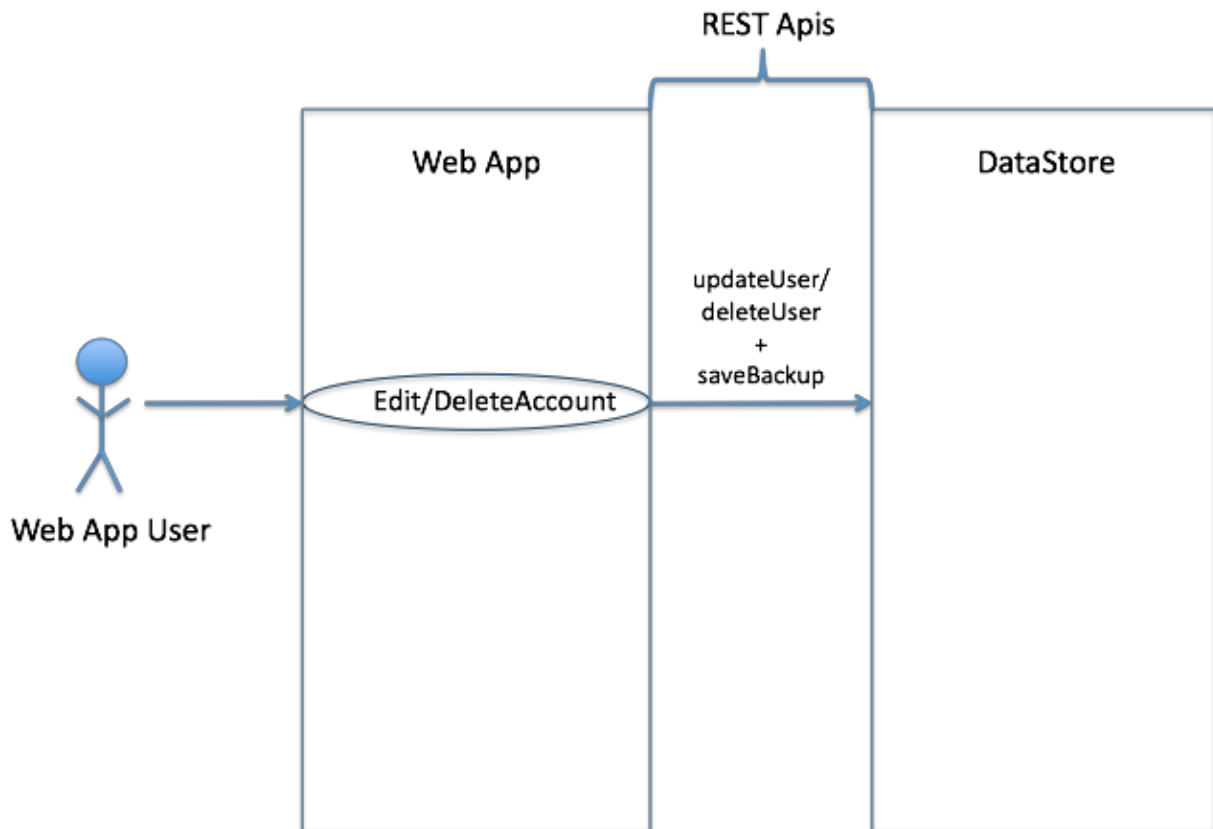
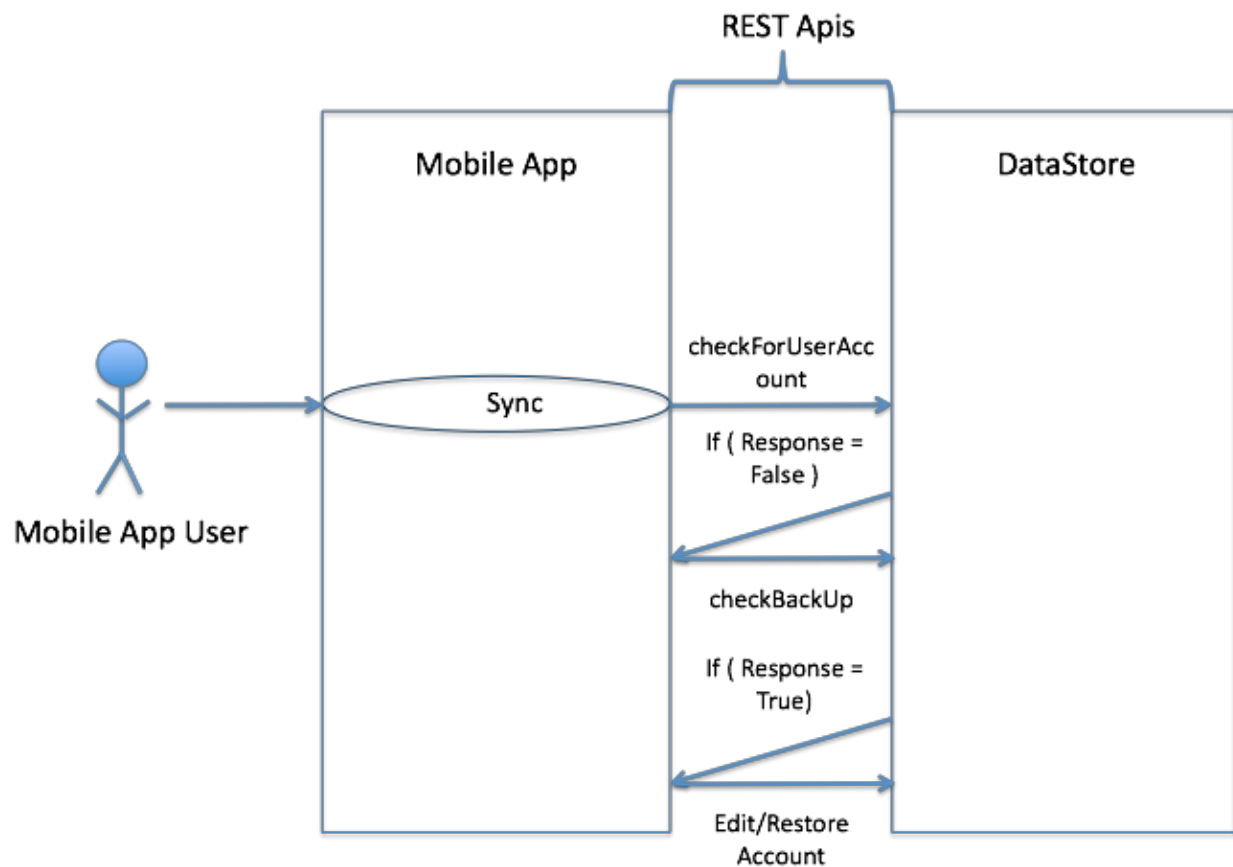


Figure 4: Edit/Delete from web app.

Editing and deleting an account can be tricky because the user on the mobile application may not sync immediately and therefore add tasks to an account that has been deleted from the Data Store. In this scenario, when the user does do a sync, the account will not be present in the Data Store. At this point The back-up stack will be checked and the account will be either deleted from the mobile (if no other changes were made after the delete) or restored on the web (if changes were made after the delete). And the user will be notified that the account was restored. This is displayed in the diagram below.



Note: If a deleted account is edited on mobile, the account is restored in the web

Figure 5: Sync from mobile after edit/delete from web app.

Alternatively, a user may edit the account information on the mobile side without syncing the edit account changes made on the web.

For the above scenarios the time stamp on both these changes will be seen and the latest one will be taken. If the time stamp happens to be the same, the DataStore version will take precedence.

Conflicts between the tasks are also handled in a similar fashion.

2.1.2. Mobile Application

On the mobile platform the user has the following uses cases:

- Create a new account: this checks the availability of the specified username in the central database.
- Login: verifies information in the local database.
- Logout
- Sync
- Edit Account Information
- Delete Account: deletes the account from both the local and central directory

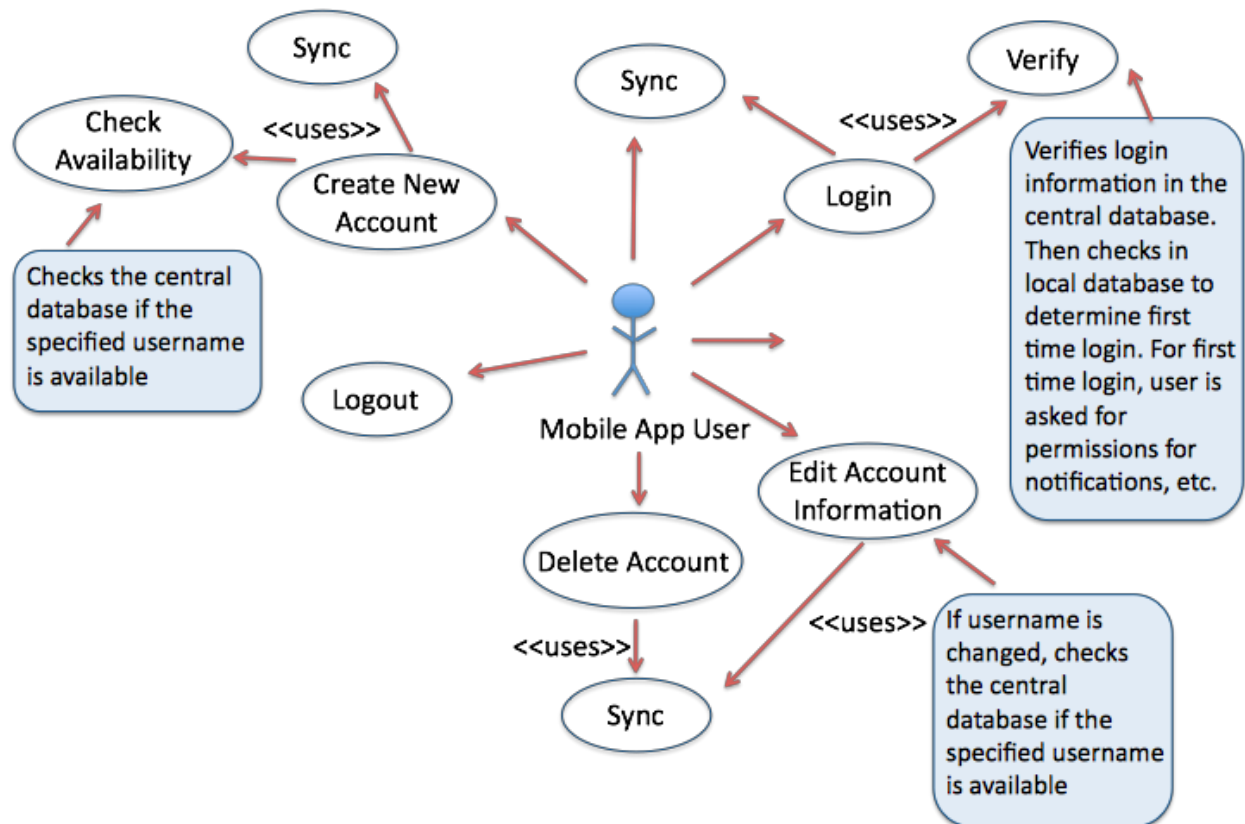


Figure 6: Accounts related use case diagram for mobile app user

2.1.2.1. Scenario: Creating a New Account from Mobile app App

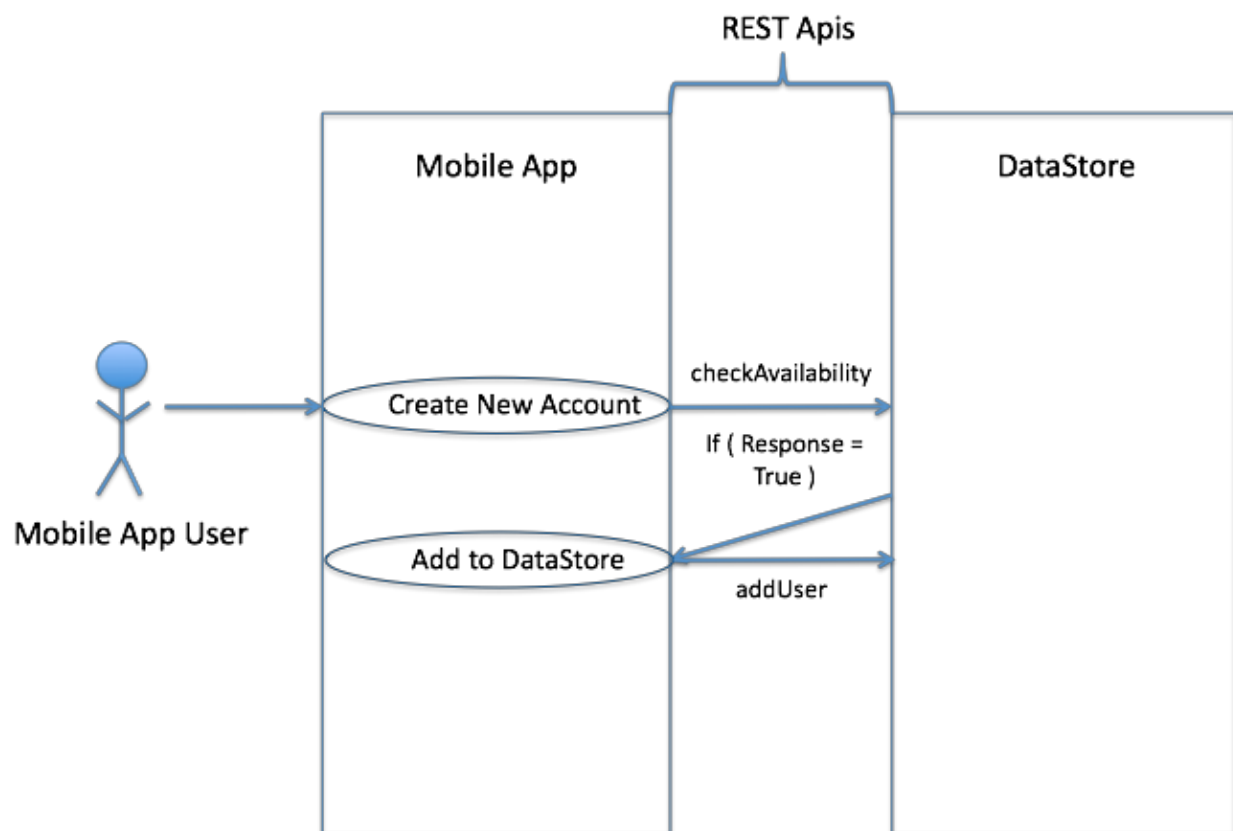


Figure 7: Creating new account from mobile app.

When the user tries to create a new account through the mobile application, the availability is checked in the DataStore. If the username is taken, the user is provided with an error message, else the user account is created in the DataStore.

2.1.2.2. Scenario: Login from Mobile app App

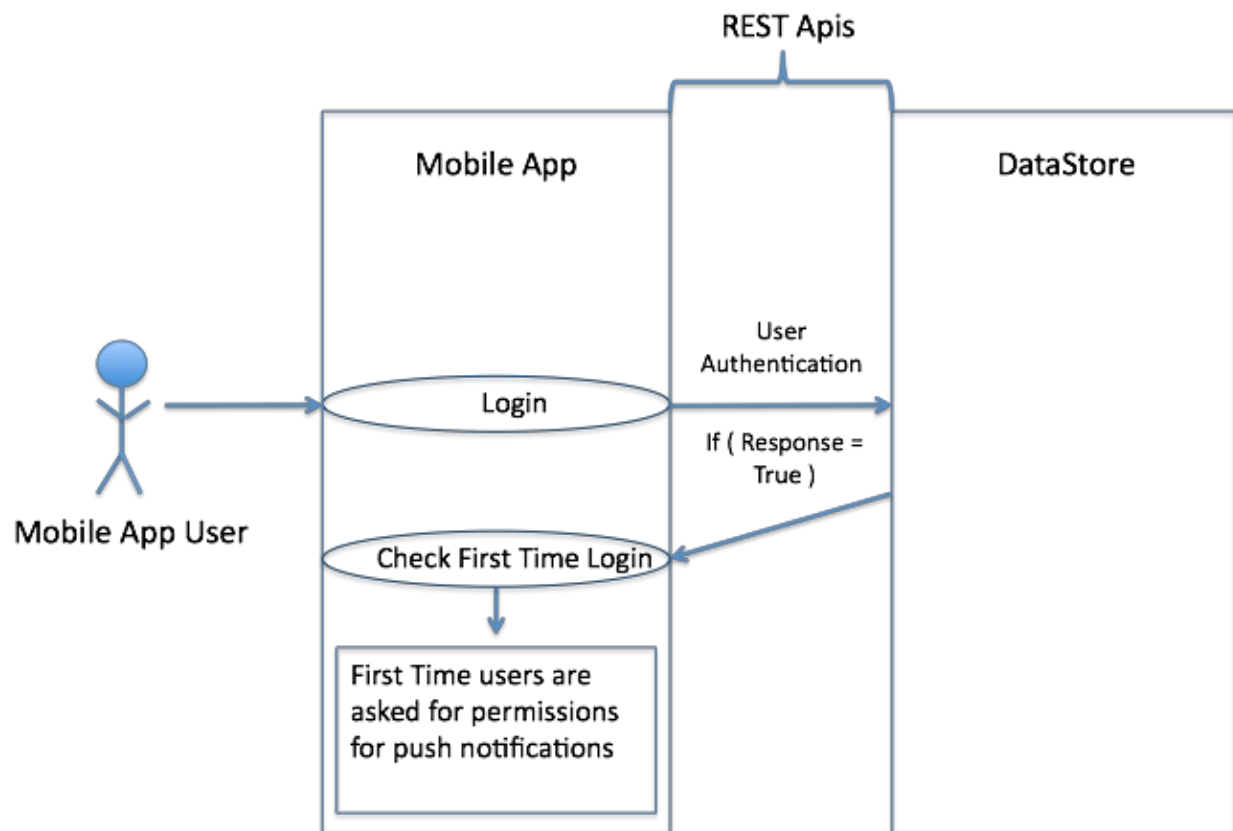


Figure 8: Login from mobile app.

Login from the mobile application requires user authentication from the DataStore. Once the user authentication from the DataStore is successful, the mobile application checks for first time login by looking up the user information in the local database. In case of first time login the user is asked for permissions for push notifications etc.

2.1.2.3. Scenario: Edit from Mobile app App

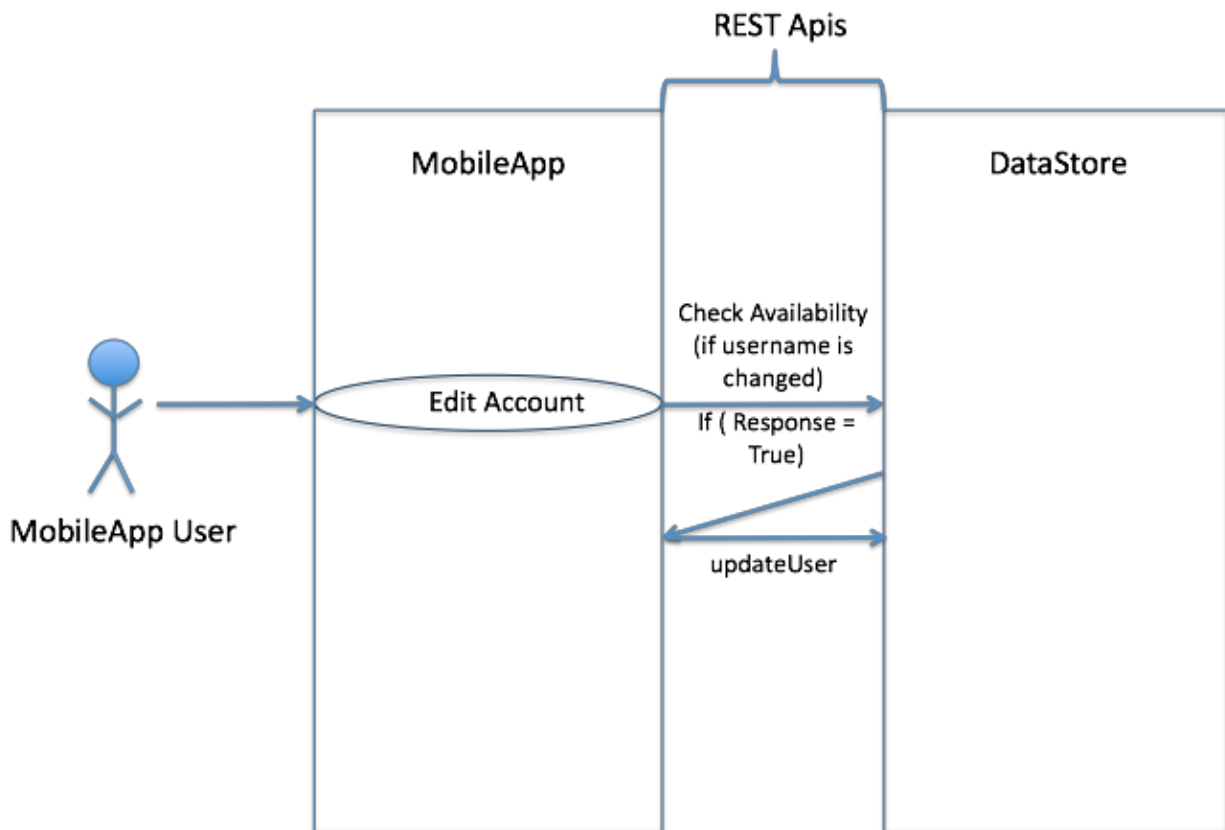


Figure 9: Edit account from mobile app.

Any edit action from the mobile side is immediately synced with the central database to avoid conflicts. If the username of the account is edited, the availability of that new username is checked.

2.1.2.4. Scenario: Delete from Mobile app App

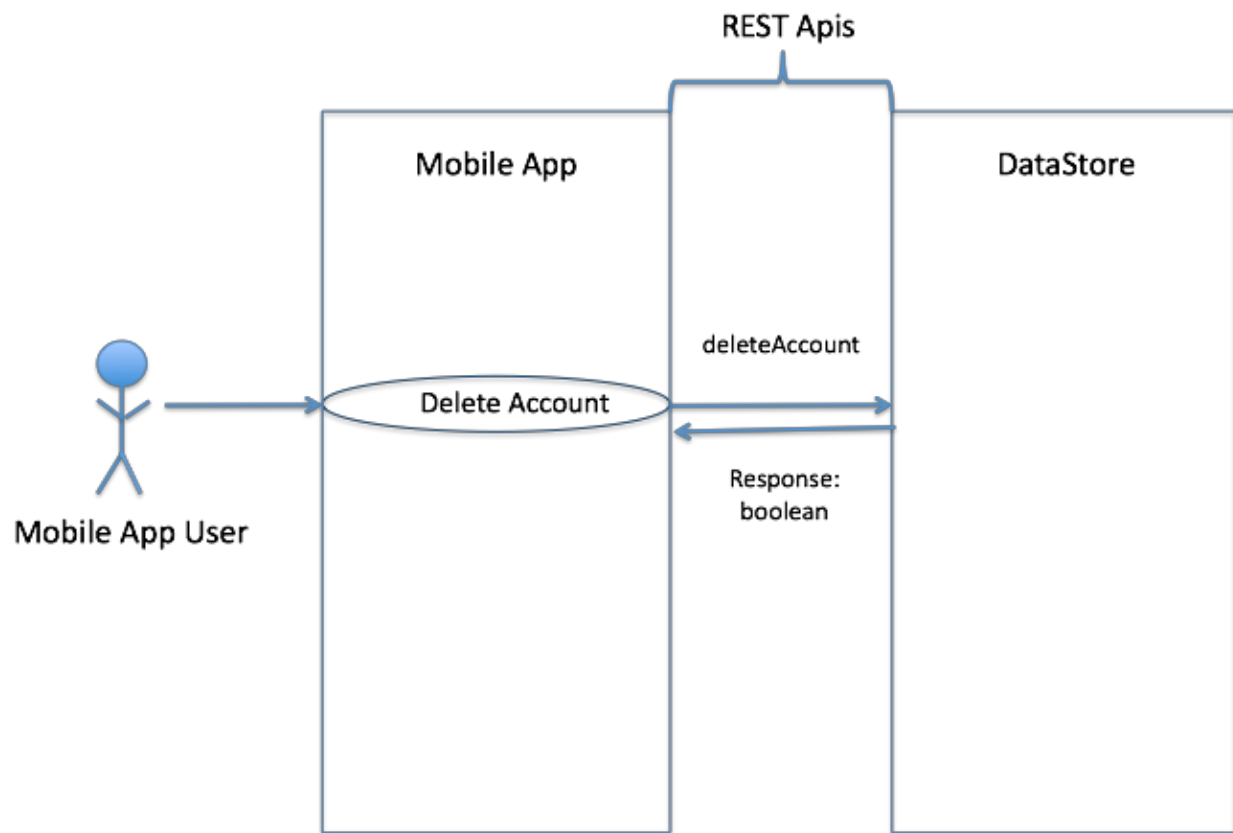
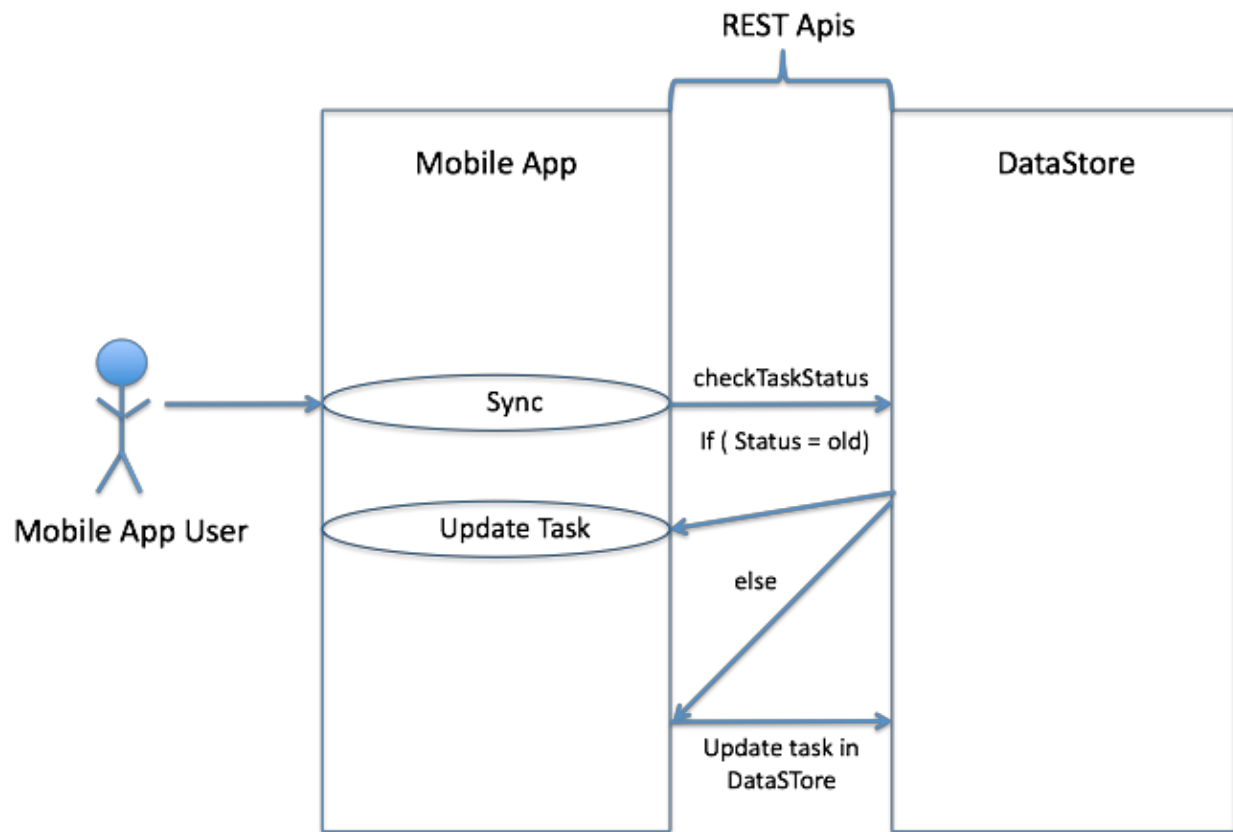


Figure 10: Delete account from mobile app.

Any delete action from the mobile side is immediately synced with the central database to avoid conflicts.

2.1.2.5. Scenario: Sync from Mobile app App



Note: If a deleted task is edited on mobile, the task is restored in the web

Figure 11: Sync account from mobile app.

Sync action is mostly for syncing tasks since changes to the account information is either synced immediately or handled with the back up stack. Note that if Google account login is implemented, these syncing functionality for accounts will not be needed.

For syncing tasks, the time stamp in both the local and central database is used to determine the latest copy.

2.2. Accounts

Both mobile app users and web app users have similar use cases. Both the users can

- Create tasks
- View tasks
- Edit tasks
- Check/uncheck tasks
- Hide/show completed tasks
- Delete Completed tasks
- Delete tasks

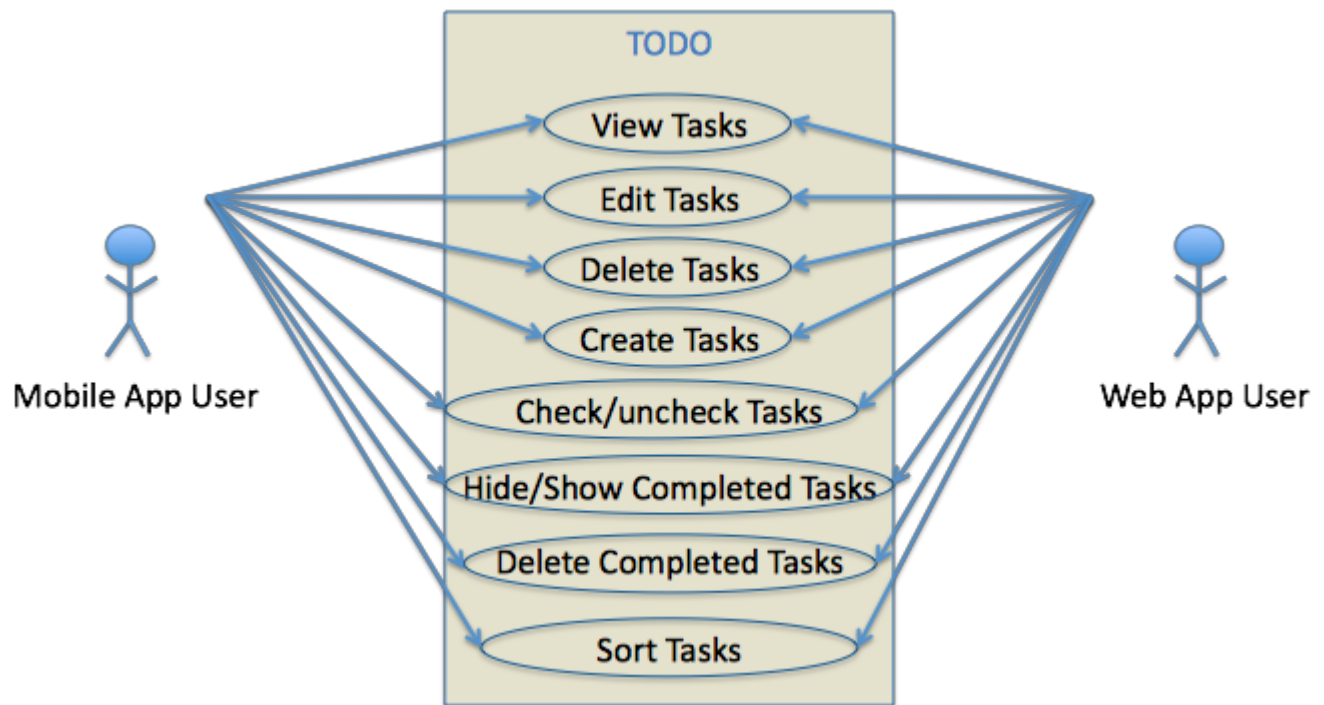


Figure 12: Task related use case diagram for mobile and web app user