

Deliverable 4: Chore Manager Android Application - Final Report

SEG 2105 B01

Fall 2017

**School of Electrical Engineering and Computer Science
University of Ottawa**

Course Coordinator: Dr. Miguel A. Garzón

	Group 22
Peter Lam Quach	8670663
Samuel Rennie	8881891
Alan Tran	8580760
Bilal Khalid	8589066

1. Introduction

The purpose of this report is to demonstrate the planning as well as the process taken in implementing our Chore Manager Application. The project was broken down into four deliverables, each playing a key role in the construction of this application. The first deliverable was to determine both functional and nonfunctional requirements for the application as well as to create use cases. Deliverable two was creating a Dynamic UML diagram for the app, as well as sequence and state diagrams, vital to our implementation. The third deliverable was the user interface for the application. Finally, the deliverable attached to this report is the source code and debug APK.

2. Contributions

Main Role	Group member	Deliverable 1	Deliverable 2	Deliverable 3	Deliverable 4
Back-End development	Alan Tran	35%	40%	10%	30%
Front/Back end development	Peter Lam	25%	10%	50%	30%
DB&F&B	Bilal Khalid	20%	20%	10%	20%
Front/Back end development	Sam Rennie	20%	30%	30%	20%

3. Corrections

Deliverable	Changes
Requirements	<ul style="list-style-type: none">• Removed requirement stating chores can be made to repeat• Added requirement about purchasing and deleting from the shopping list• Added requirement stating users must be able to filter resources by chore time• Added requirement stating users must be able to assign a type to a chore• Added post conditions to use-case and changed steps for creating a chore.
UML, Sequence and State Diagrams	<ul style="list-style-type: none">• Updated UML to reflect new methods and variables made during development, as well as fix errors in associations between classes• Changed state diagram to accurately reflect the implemented chore class• Changed sequence diagram to reflect app.

User Interface	<ul style="list-style-type: none"> • Updated UI to include deletion of buttons in grocery, material and shopping lists • Updated UI to show new views created during the implementation process • Updated Colour Scheme
-----------------------	--

4. Software Requirements

Functional Requirements

FR1	The system must allow a parent user to create a chore specifying the title and a description of the task.
FR2	The system must allow parent users to specify or change the deadline to any existing chore.
FR3	The system must allow users to delete or purchase items in the shopping list.
FR4	The system must allow parent users to specify the required materials for doing a chore.
FR5	The system must display to users the materials used for certain categories of chores
FR6	The system must allow users to log out.
FR7	The system must allow admin users to add or delete notes to a chore.
FR8	The system must allow a user to set a category when creating a chore
FR9	The system must allow users to filter out chores not assigned to them.
FR10	The system must display to all users the pantry and material items.
FR11	The system must display to users their chore completion status.
FR12	The system must allow all users to add a list of items that need to be purchased into the shopping list
FR13	The system must allow users to change their profile image, choosing from a selection of icons.
FR14	The system must display all users' total points in a menu.
FR15	The system must allow admin users to reset all data stored in the database.
FR16	The system must allow parents to assign unassigned chores to a child.
FR17	The system must allow users to sort chores by deadline or alphabetical order.

FR18	The system must prompt a user for their password to access their profile.
FR19	The system must not allow non-admin users to create or edit a chore.
FR20	The system must allow users to create an account using a username and password and to specify whether it is a parent or a child account.

Quality Requirements

QR1	The system must delete or reassign a task to a user in less than 8 clicks.
QR2	The system must switch user accounts in less than 5 clicks.
QR3	The system must update the pantry/cupboard and shopping list within 2 seconds.

Process Requirements

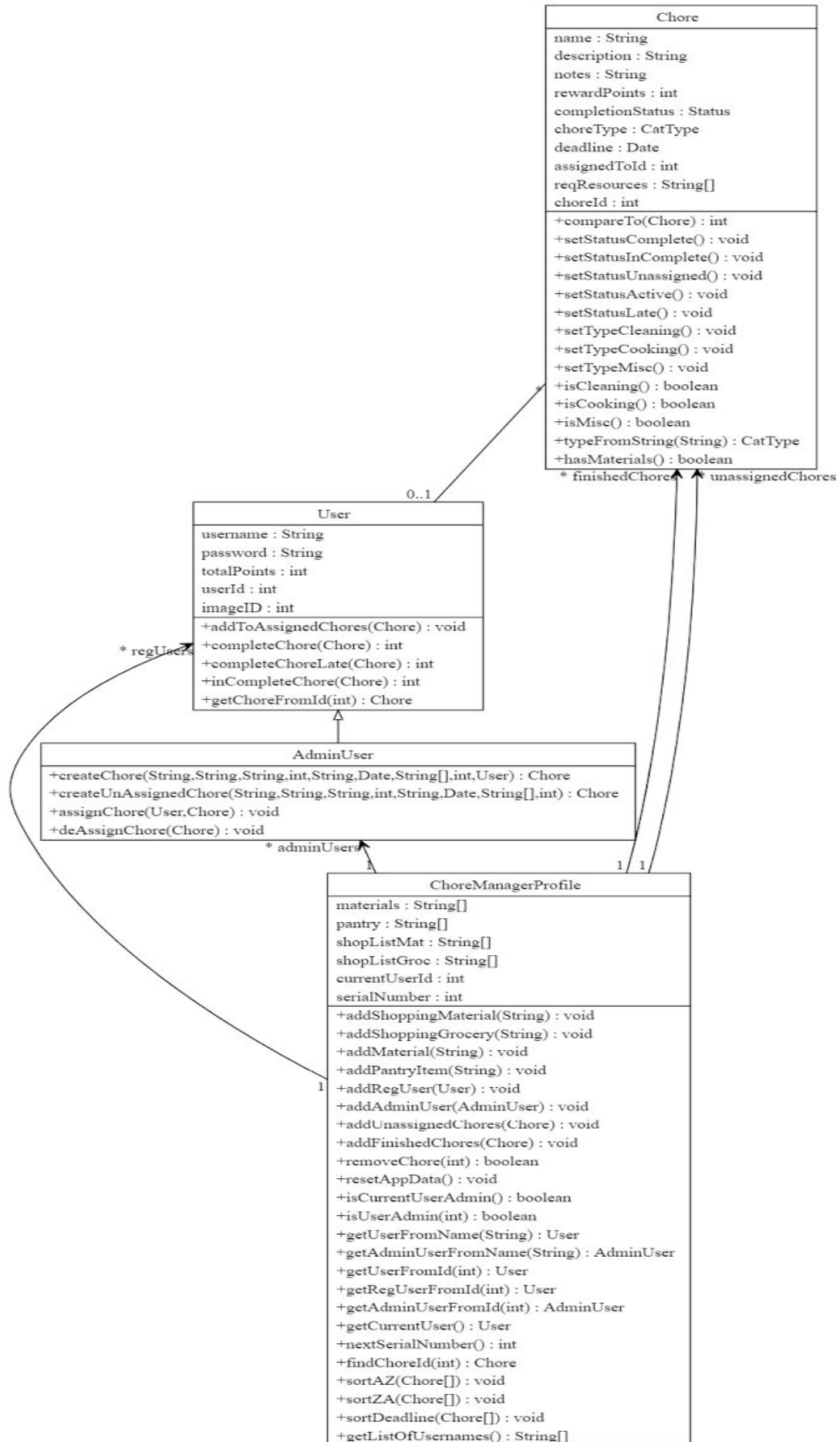
PR1	The systems use cases and requirements must be completed by November 2nd, 2017.
PR2	The system must be fully completed by December 6th, 2017.

Platform Requirements

PLR1	The system must be coded in Java and built using Android Studio 2.3.
PLR2	The system must be compiled using min SDK as Android 4.1 (API 16).
PLR3	The system must use Firebase as a database for storing and retrieving application data.

5. UML and Modeling

5.1 UML



5.2 Use Cases

Use Case: Add Chore

Actors: Admin-User (Parents)

Goals: Add chores and Assign who is responsible

Preconditions: User account must be an admin to add chores, only one person may be assigned to a specific chore

Postconditions: A chore will be created and assigned to a specified user.

Summary: User must be able to create a chore containing and end time, description, and assign a user to the chore

Steps:

Actor Actions	System Responses
1 . Click the chores button on the main menu	2. Switch view to choreActivity
3. Click on the 'Add Chore' button	4. 'Add Chore' dialog box appears
5. Enter chore name, deadline, description, note	
6. Click on assign a user spinner	7. Dropdown menu appears with users
8. Select which user	9. Dropdown menu closes
10. Click on type of chore spinner	11. Dropdown menu appears with types
12. Select type of chore	13. Dropdown menu closes
14. Click points spinner	15. Dropdown menu appears with points
16. Select how many points chore is worth	17. Dropdown menu disappears
18. Click on materials spinner.	19. Dropdown menu appears with materials
20. Select required materials	21. 'Materials' dialog appears on top of 'Add Chore'
22. Check mark applicable materials in the list	23. Dropdown menu disappears
24. Confirm selections	25. Close 'Add Chore' dialog 25. Create and store chore to the database and assign (if specified).

Use Case: Reset Data

Actors: Admin-User (Parents)

Goals: Reset all data in application

Preconditions: User account must be an admin

Postconditions: All data is wiped and user is brought to CreateUser menu.

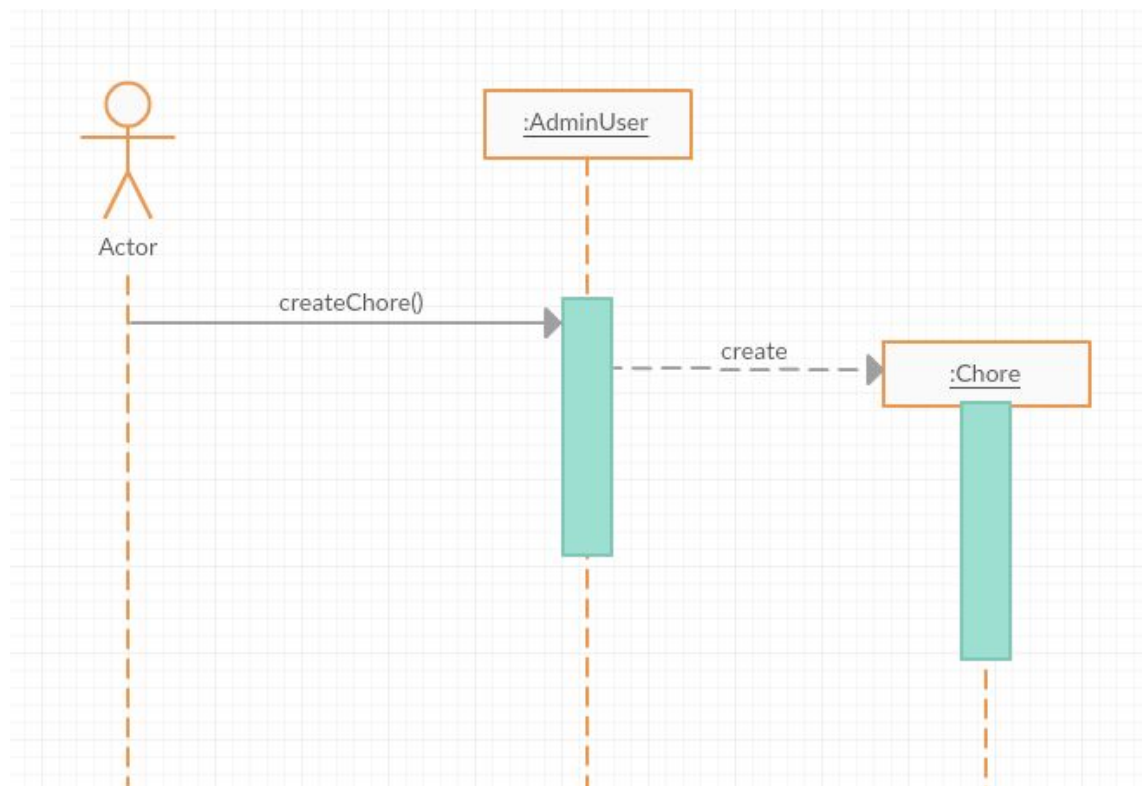
Summary: System resets all info, including all chores and associated information, while also deleting all accounts.

Steps:

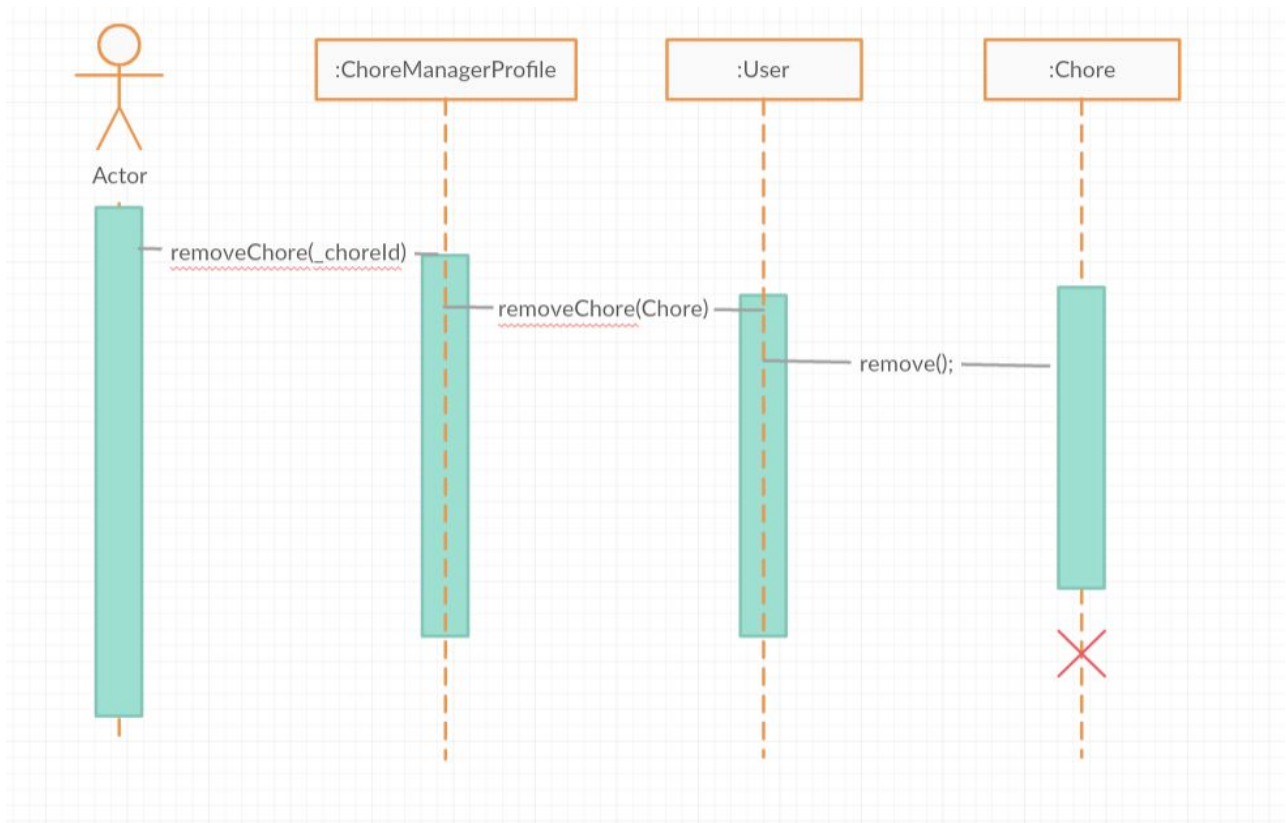
Actor Actions	System Responses
1 . Select settings within the main menu 2. Choose "Reset all data" within settings menu 4. Choose "Yes"	2. Switch to the settingsActivity view 3. Prompts user with a confirmation dialogue. 5. Close dialogue 6. Wipes database of all stored data 7. Switch to the SpecialUserCreationActivity view

5.3 Sequence Diagrams

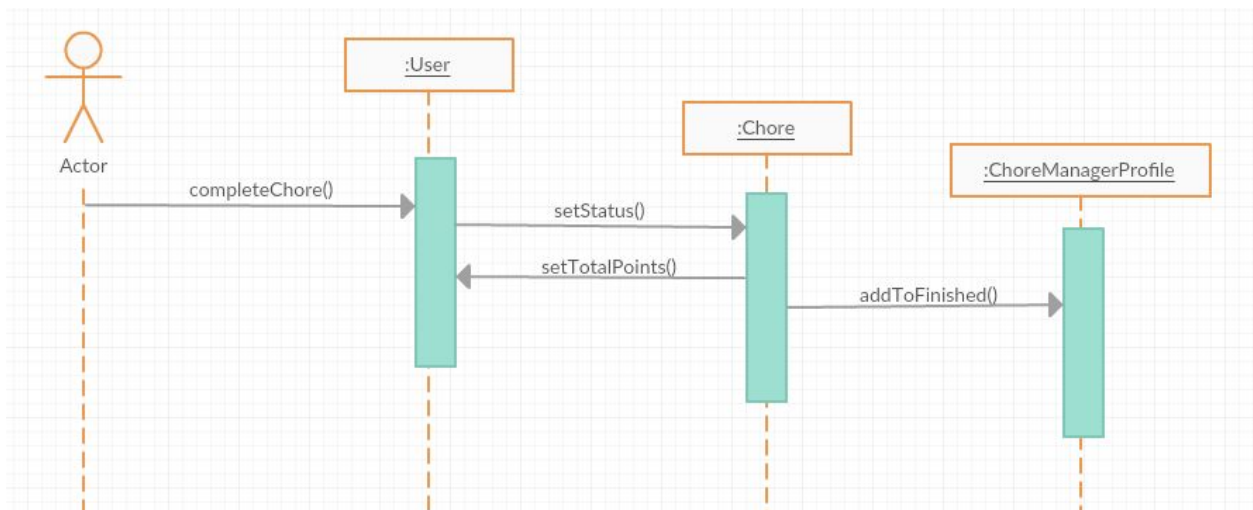
Creating a Chore



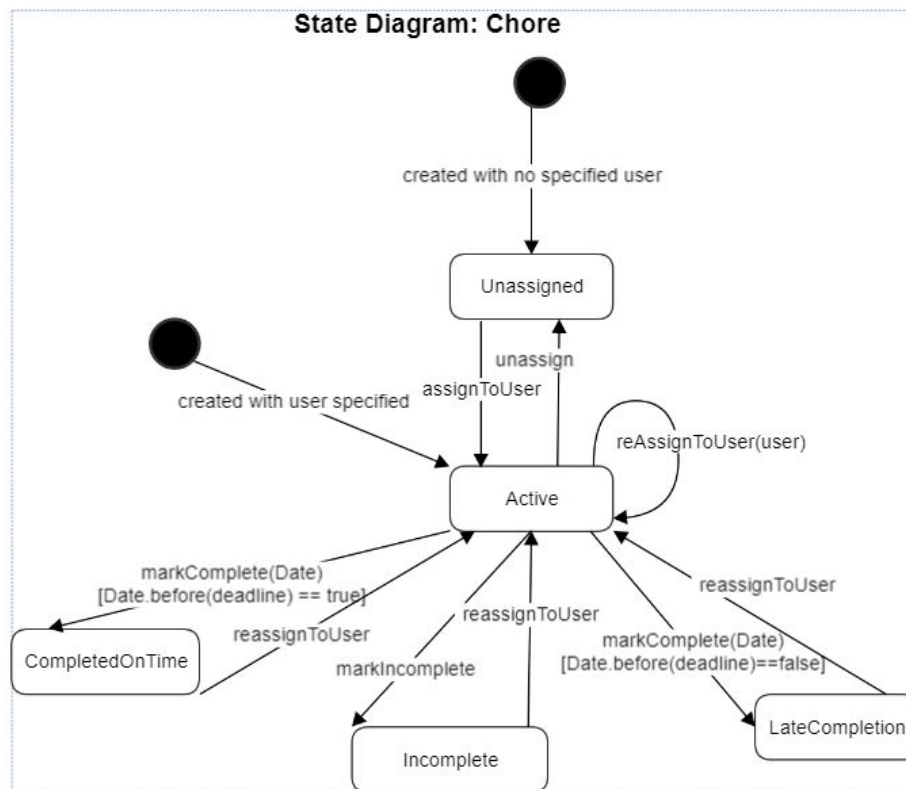
Deleting a Chore



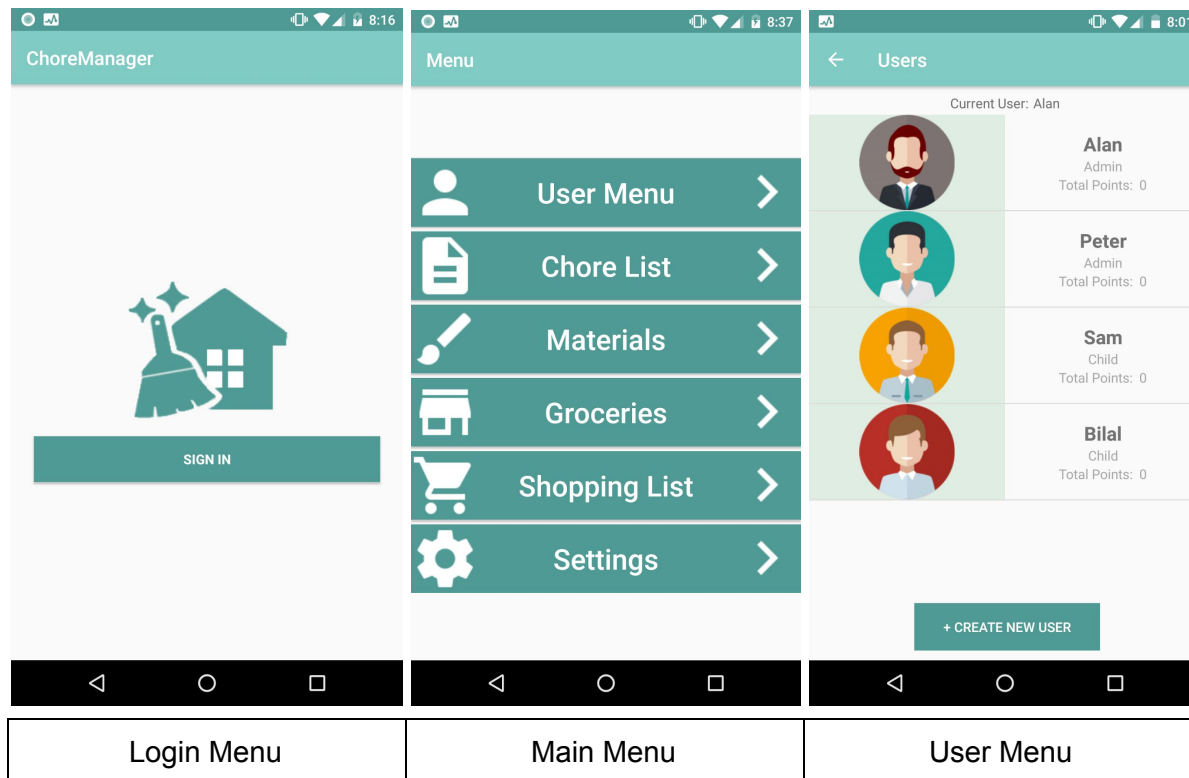
Complete Chore

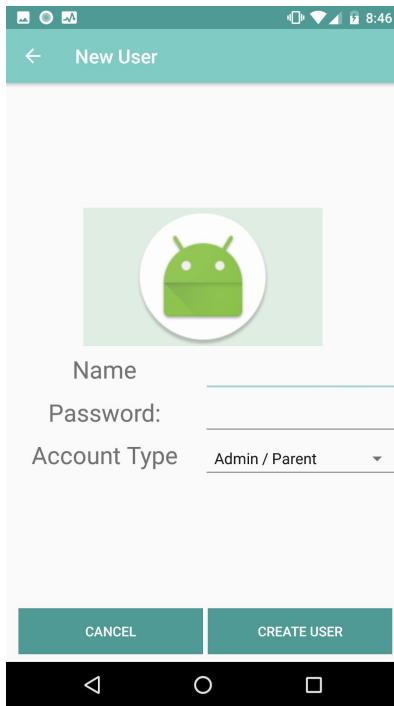


5.4 State Diagram




6. User Interface





New User



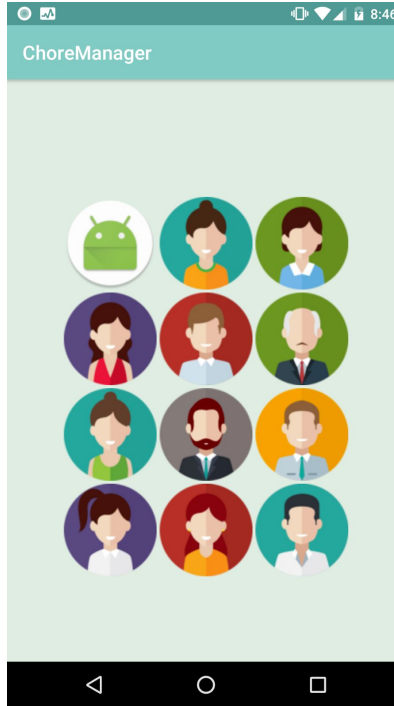
Name: _____

Password: _____

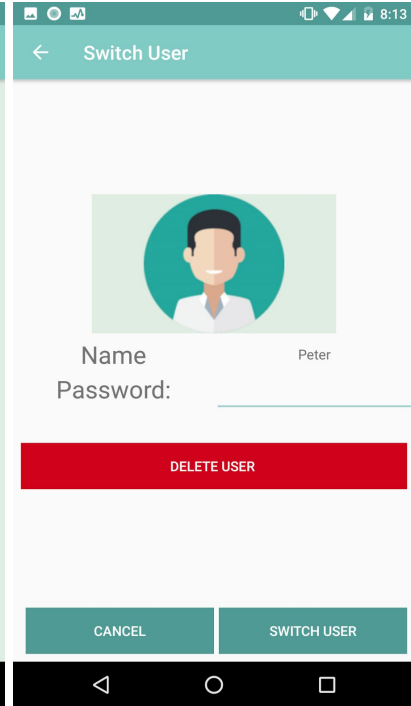
Account Type: Admin / Parent

CANCEL CREATE USER


New User Creation



Avatar Selection



Switch User



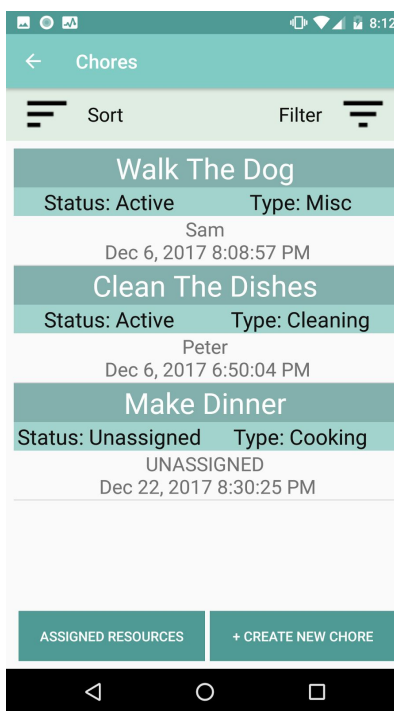
Name: _____

Password: _____

DELETE USER

CANCEL SWITCH USER

Switch/Delete User Menu



Chores

Sort Filter

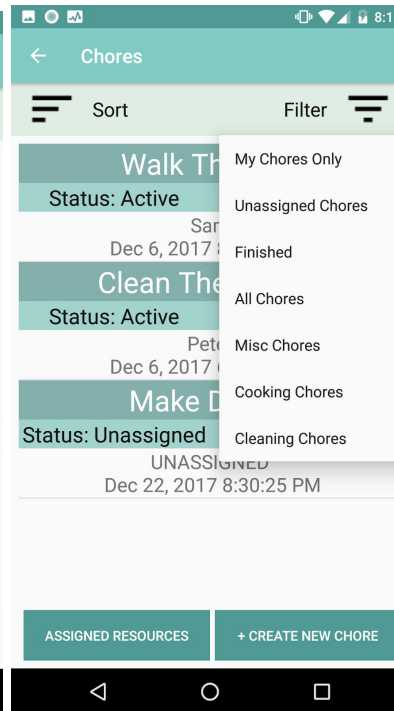
Walk The Dog
Status: Active Type: Misc
Sam
Dec 6, 2017 8:08:57 PM

Clean The Dishes
Status: Active Type: Cleaning
Peter
Dec 6, 2017 6:50:04 PM

Make Dinner
Status: Unassigned Type: Cooking
UNASSIGNED
Dec 22, 2017 8:30:25 PM

ASSIGNED RESOURCES + CREATE NEW CHORE

Chore List Menu



Chores

Sort Filter

Walk The Dog
Status: Active
Sam
Dec 6, 2017 8:08:57 PM

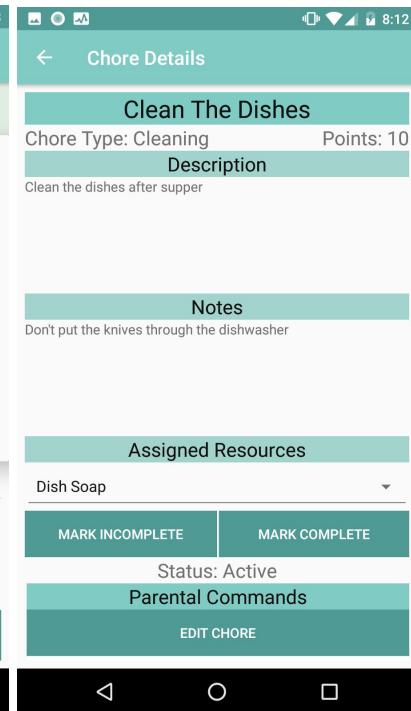
Clean The Dishes
Status: Active
Peter
Dec 6, 2017 6:50:04 PM

Make Dinner
Status: Unassigned
UNASSIGNED
Dec 22, 2017 8:30:25 PM

ASSIGNED RESOURCES + CREATE NEW CHORE

Filter menu options: My Chores Only, Unassigned Chores, Finished, All Chores, Misc Chores, Cooking Chores, Cleaning Chores

Filter Menu



Chore Details

Clean The Dishes

Chore Type: Cleaning Points: 10

Description
Clean the dishes after supper

Notes
Don't put the knives through the dishwasher

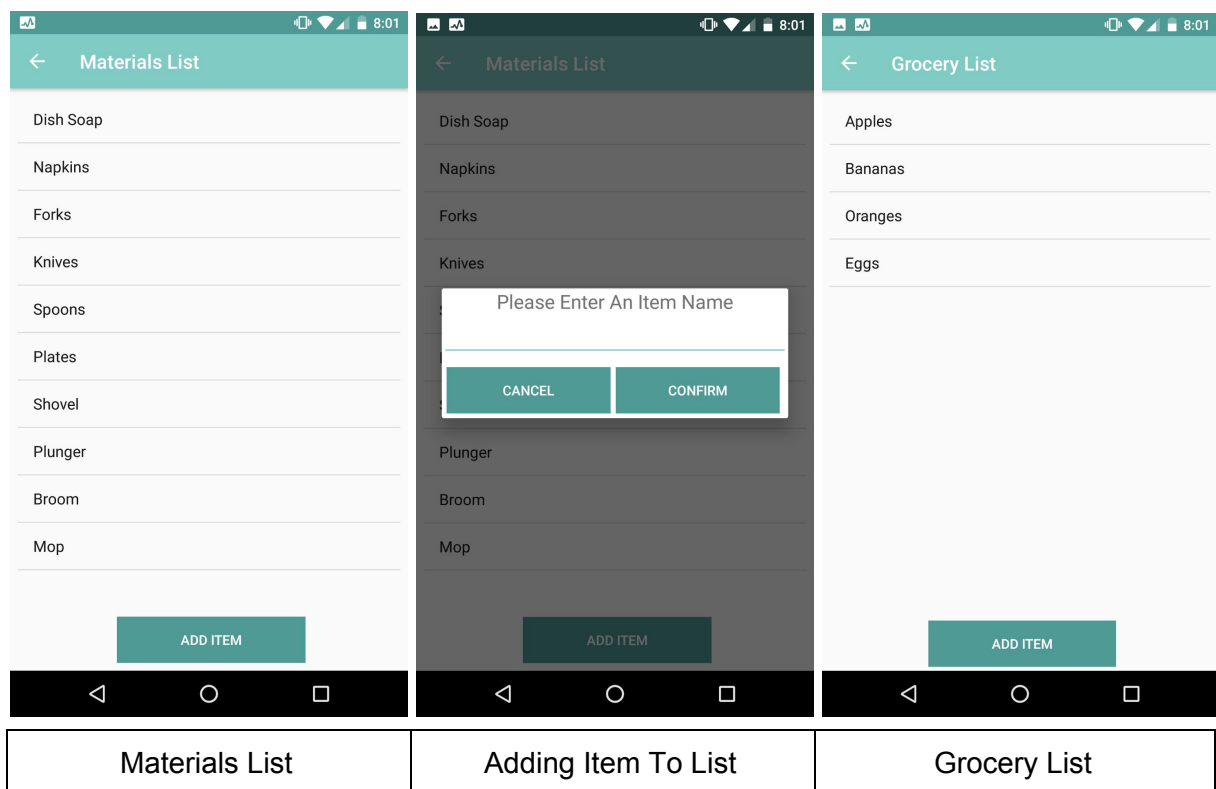
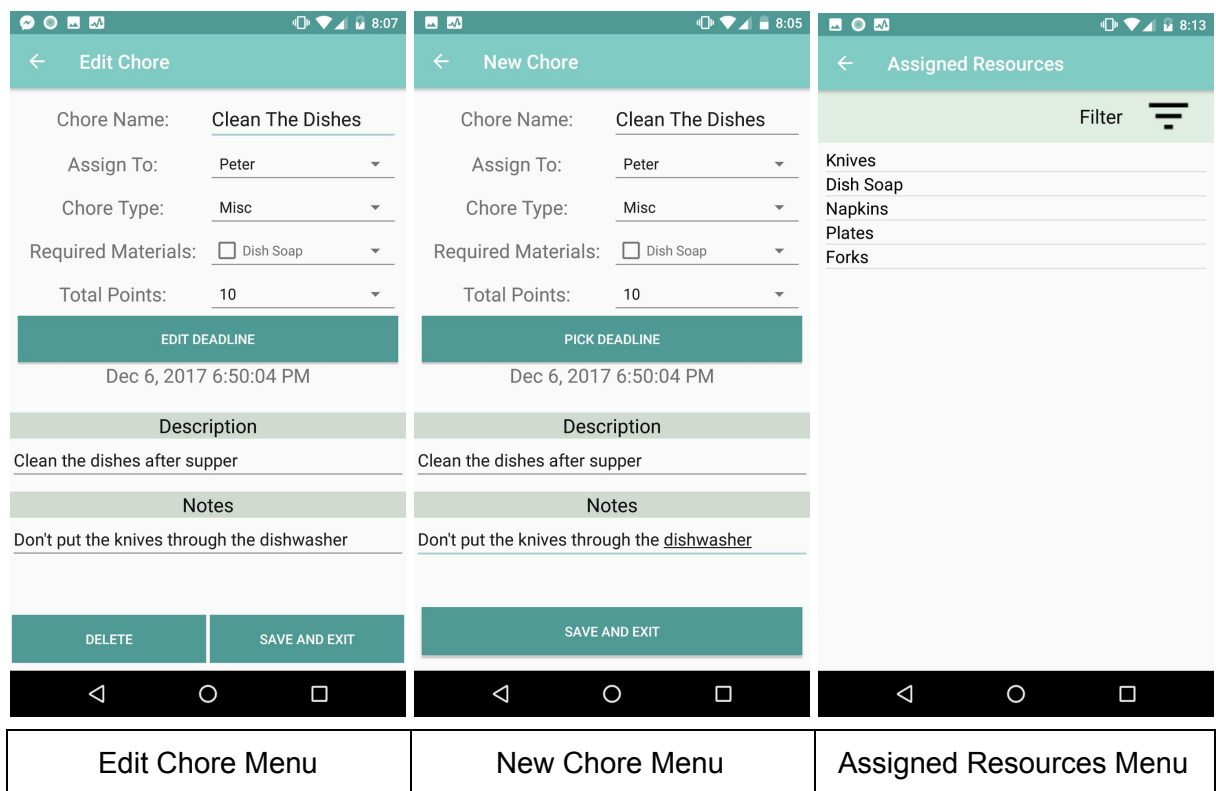
Assigned Resources
Dish Soap

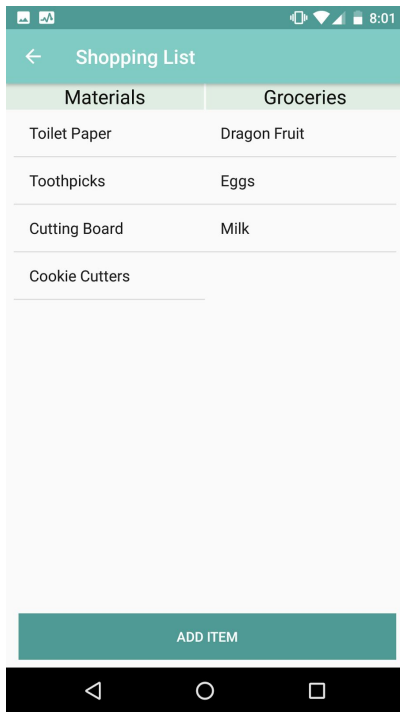
MARK INCOMPLETE MARK COMPLETE

Status: Active

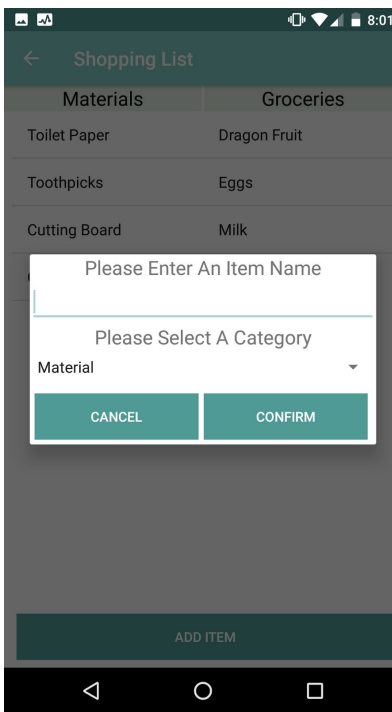
Parental Commands
EDIT CHORE

Chore Details

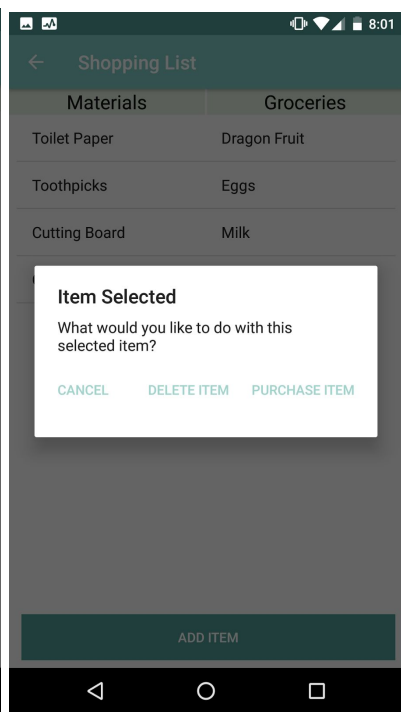




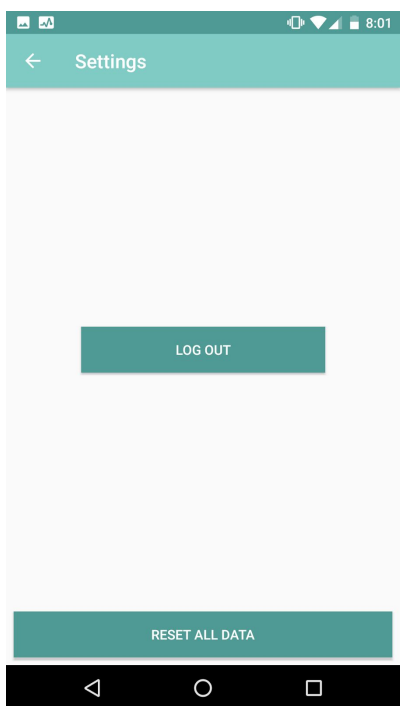
Shopping List



Adding Item to List



Item Selection



Settings Menu

7. Lessons Learned

Through the process of designing and implementing the Chore Manager application we learned a lot as a group. The first major lesson learned was to have reasonable expectations when designing software. Our original plans for the application proved overly ambitious causing features we wanted to implement to be cut due to time constraints. This project also did an excellent job of teaching us how to code in large groups. Through trial and error we found the divide and conquer approach to be the most efficient, at least in the context of this application. A suggestion for improving this project for future years would be setting the first three deliverables earlier. By forcing groups to have the UI implemented earlier groups would have more time to focus on coding the application.