SikWare presents:

FML: Fix My Life

Design Document

Adam Pluth

Ken Logan

Brandon Mottor

04/04/2017

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Comment | Author | Date |
| 0.1 | \_init\_ | Adam Pluth | 02/16/2017 |
| 0.2 | Functional Requirements | Brandon Mottor | 03/01/2017 |
| 0.3 | Descriptions | Ken Logan | 03/01/2017 |
| 0.4 | Intro, Terminology,  Descriptions, System requirements | Adam Pluth | 03/01/2017 |
| 0.5 | Clean up, added Table of Contents, Appendices | Adam Pluth | 03/02/2017 |
| 0.6 | Activity Diagrams | Ken Logan | 03/02/2017 |
| 0.7 | Activity Diagrams | Brandon Mottor | 03/02/2017 |
| 0.8 | Formatting, correcting errors | Ken Logan | 03/21/2017 |
| 0.9 | Added Media/Pantry sequence diagrams | Brandon Mottor | 03/23/2017 |
| 0.10 | Added Chat/Progress sequence diagrams | Adam Pluth | 03/23/2017 |
| 0.11 | Added GroupManager, Notes, Calendar sequence diagrams | Ken Logan | 03/23/2017 |
| 0.12 | Added detailed class diagram | Brandon Mottor/Ken Logan | 03/26/2017 |
| 0.13 | Final proofread for submission | All | 04/04/2017 |

Table of Contents:

1. Preface 4
2. Intro 4
   1. Overview 4
   2. Terminology 4
      1. Living Group 4
      2. Karma 4
      3. Head of Household 4
3. Project Description 5
   1. Initialization 5
   2. Start Screen 5
   3. Users 5
   4. Sharing / Synchronization 5
   5. Rewards 5
   6. Inter-Communication 6
   7. Inventory / Ownership Log 6
   8. Scheduling / Reminders 6
   9. Notes List 6
4. Requirements 7
   1. Functional 7
      1. Calendar 7
      2. Notes List 7
      3. Shopping List 7
      4. Media Inventory 7
      5. Rewards 7
      6. Inter-Communication 7
   2. Non-Functional 7
      1. Admin User 7
      2. Synchronizer 7
      3. Data Manager 7
      4. Database 8
5. System Requirements 8
6. Appendix 9
   1. A-Use Case Diagram 9
   2. B-Class Diagram 10
   3. C-Activity Diagrams 11
      1. Chat 11
      2. Progress 12
      3. Notes 13
      4. Calendar 14
      5. Pantry 15
      6. Media 16
   4. D-Sequence Diagrams 17
      1. Chat 17
      2. Progress 18
      3. Notes 20
      4. Calendar 22
      5. Pantry 24
      6. Media 26
      7. GroupManager 28
7. Preface

Design document for an Android application titled “FML” or “Fix My Life.” This document outlines terminology of the app, platform requirements, functional and non-functional requirements, constraints, assumption, and diagrams.

1. Intro
   1. Overview

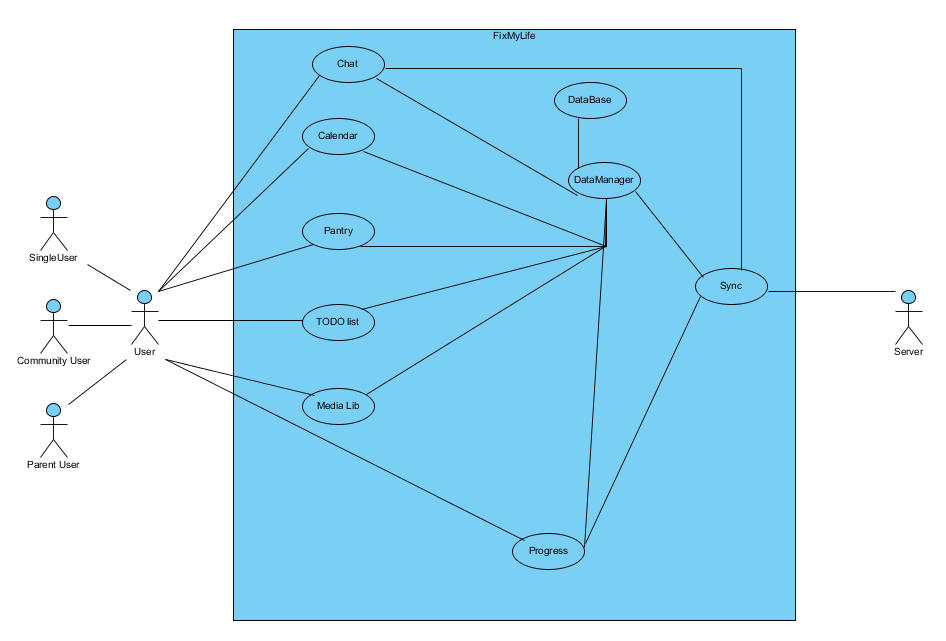
The intention of this app is to facilitate a happy and healthier and better organized lifestyle, either by yourself or with others.

The scope of the application is to provide users with an incentivized chore management system, a media and pantry item inventory system, a notes and task management system, and a living group chat system.

* 1. Terminology
     1. Living Group – group of users registered to the same living space or family.
     2. Karma – points awarded for events inside app. These point can have a positive or negative value.
     3. Head of Household(HOH) – the Admin user(s) for the living group.

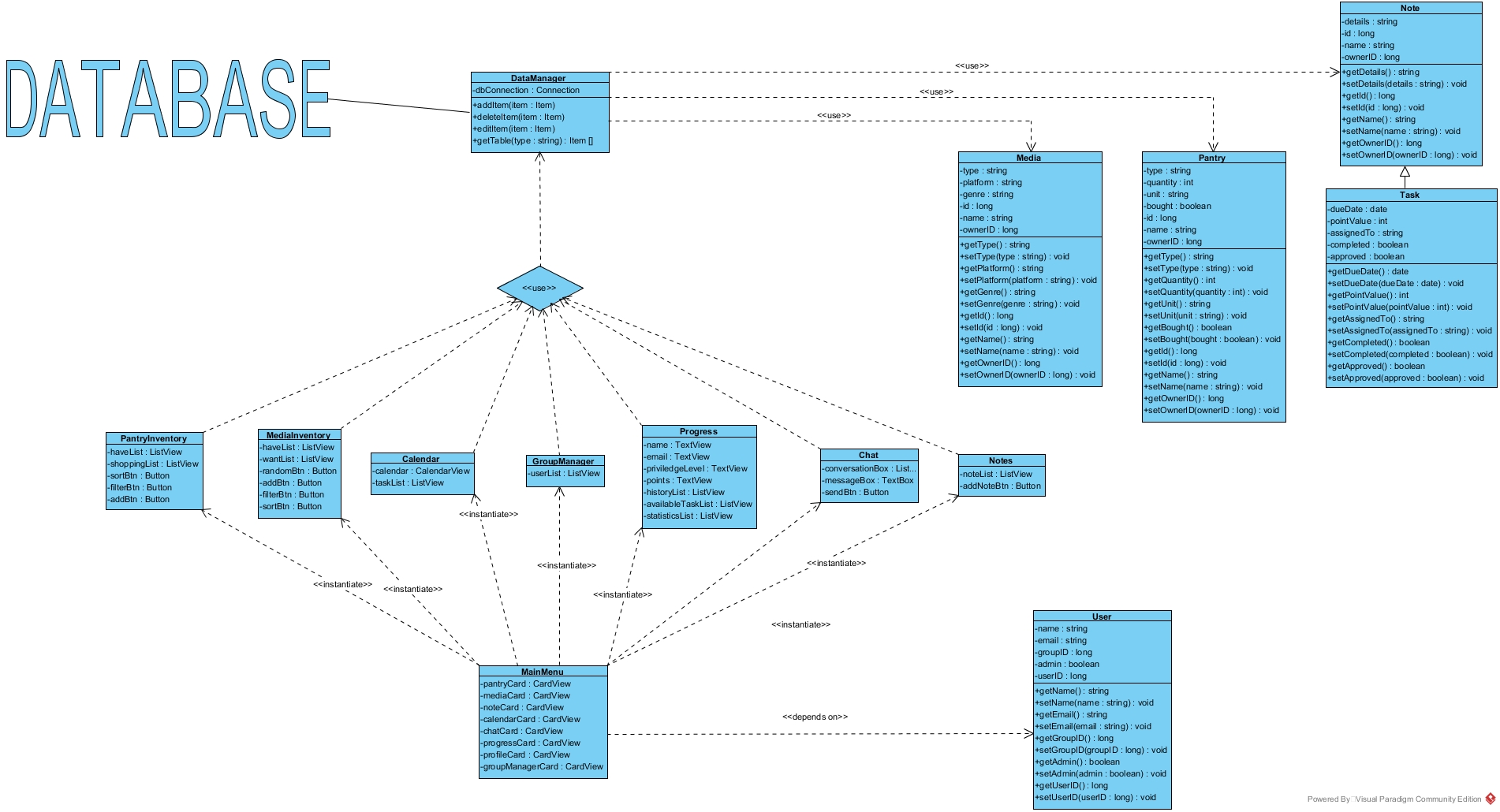
1. Project Description
   1. **Initialization:** When first ran, this app will ask for an existing user account or create a new one. After which the user will be asked if there are any persons living with the user and if they would like to participate in the experience. Steps will be taken to fit the user into their respective user category.
   2. **Start Screen:** Upon starting this App the user will be presented with a table that is auto populated with preset options.
   3. **Users:** the users will be classified into three separate entities. Single users will have free use of the app with automated encouragement. Another user entity is that of a community living area, such as a dorm/apartment, these users will be able to share information back and forth to provide more adequate living arrangements. The final case will be that of a parent child relationship in which the parent will have admin rights over the child and the child will have to follow certain protocols.
   4. **Sharing/Sync:**  the Data inputted by our users will be synchronized across all people in the living domain that use this app.
   5. **Rewards System:** The rewards section is designed to give a sense of achievement when utilizing the app. This section will have a progress bar which will balance the tasks accomplished with the ones that have not been accomplished with the ones that have. Positive “karma” points will be awarded to a person when an item is completed which will be exchanged for requests or unlocking mini games. There will be a history of tasks completed and points spend along with negative karma from missing a task.
   6. **Inter-Communication:** The chat activity will implement the SendBird messaging API, utilizing its group messaging feature for household wide chat.
   7. **Media and Pantry Inventory:** Users will be able to catalogue pantry and media inventory items. Both types of items will have unique characteristics that can be searched, sorted, or filtered. Items can be added to a “owned” list or a “wanted” list. Users will also be able to randomly select a media item for scenarios in which they cannot decide what to pick.
   8. **Calendar:** The scheduling menu will allow users to interact with a calendar to manage their events, appointments, tasks, and to-do items. Users can add new items or edit current items by interacting with the calendar, as well as changing the current calendar view to suit their needs. As an additional feature, there will also be an area to select items from your to-do list and add them to your calendar or get suggestions for free time to schedule new items.
   9. **Notes:** The notes activity will allow users to quickly add notes to the app. The notes can be upgraded to “tasks”, in which case they would have a user assigned to and an optional due date.

1. Requirements
   1. **Functional**
      1. *Calendar*: Organize and cleanly display events to the user. The ability to import tasks to this will be an additional feature. This will also provide a reminder when approaching time for event(s).
      2. *Notes List*: List of notes that are not important enough to warrant a task and a list of tasks that are assigned to users and have optional due dates. Notes will be able to be promoted to tasks.
      3. *Shopping List*: Dual Lists displaying items needed and items currently owned and in cabinets/refrigerator.
      4. *Media Inventory*: Similar to Shopping List, catalogs all media in house including games, movies, music, eBooks, etc… Also contains record of which person owns media, platform and other information about media items.
      5. *Reward**System*: this is to give users rewards to utilizing this application through various means.
      6. *Inter-Communication*: there will be a way of establishing a chat session for text, video, and Sticky Note communications.
   2. **Non-Functional**
      1. *Admin User:* this user(s) will set tasks and deadlines. Also, this user will be sent information to provide verification of the task to be confirmed and removed from the child’s chore list.
      2. *Data manager:* Responsible for pushing and retrieving Item objects to each phones local SQLite DB.
      3. *Database:* A SQLite database will be installed on the local devices which will contain all stored information, such as tasks, pantry items, media items, etc.
2. System Requirements
   1. Android handset with minimum SDK 5.0
   2. Internet access
3. Appendix
   1. A-Use Case Diagram



A user will open the app. Depending on the user type, the Main Activity will display a set view. From here the user will be able to open the Chat Activity, the Calendar Activity, the Pantry Activity, the Notes Activity, and the Media Activity. These activities will communicate with the DataManager to push changes to the respective local DBs.

* 1. B-Class Diagram

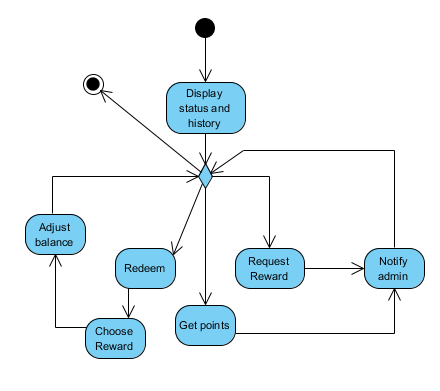


There are three main item types: MediaItem, PantryItem, and NoteItem. NoteItem has a subclass called TaskItem. The MainActivity will display certain tiles based off User privilege level, which will be stored alongside other relevant information in the User class. The MainActivity will be able to call PantryActivity, MediaActivity, CalendarActivity, NotesActivity, ChatActivity, and Progress/GroupManager Activity (depending on privilege level). The different activities will interact with the DB through the DataManager class, which will push and retrieve Item objects to or from the DB.

* 1. C-Activity Diagrams
     1. Chat

The chat activity will simply call the SendBird API, in which case a SendBird activity will be called and all interaction with our app will cease until the user exits the chat session with his group.

* + 1. Progress

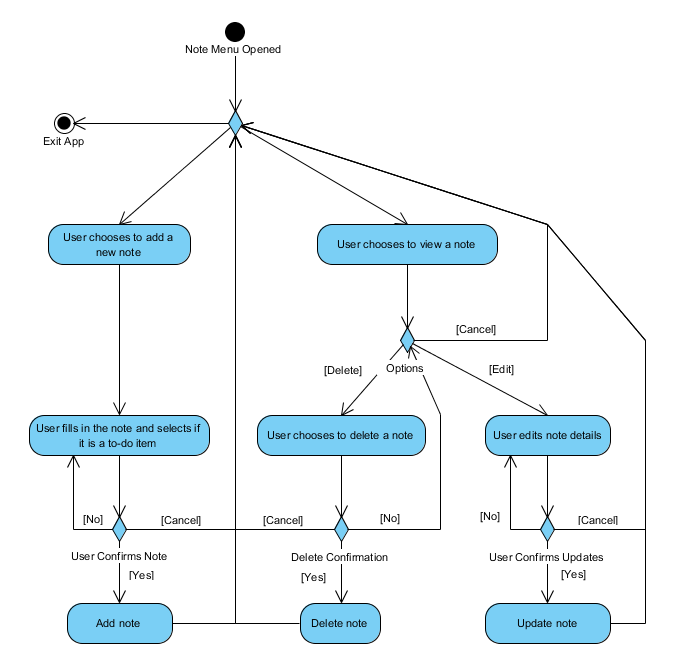


If user is SuperAdmin, the user is provided with a group management screen in which the user can select any user in the living group, including themselves. The profile will display available tasks, completed tasks, and statistics about completed tasks.

Admin users will be presented with their own user profile which will contain all elements of the user profile listed above.

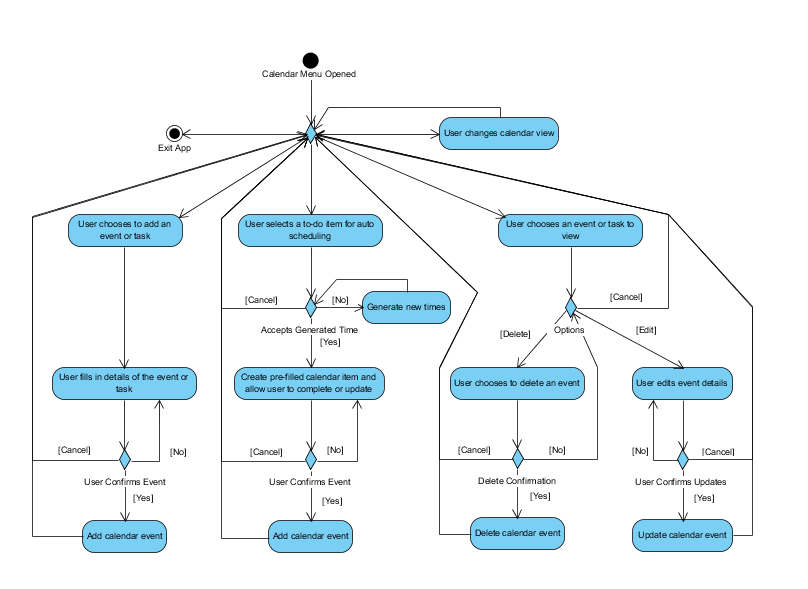
Child users will be presented with a rewards variant of the profile screen in which they will see preassigned tasks for point values, monitor total earned points, and a section for redeeming points for an item off the users wishlist.

* + 1. Notes



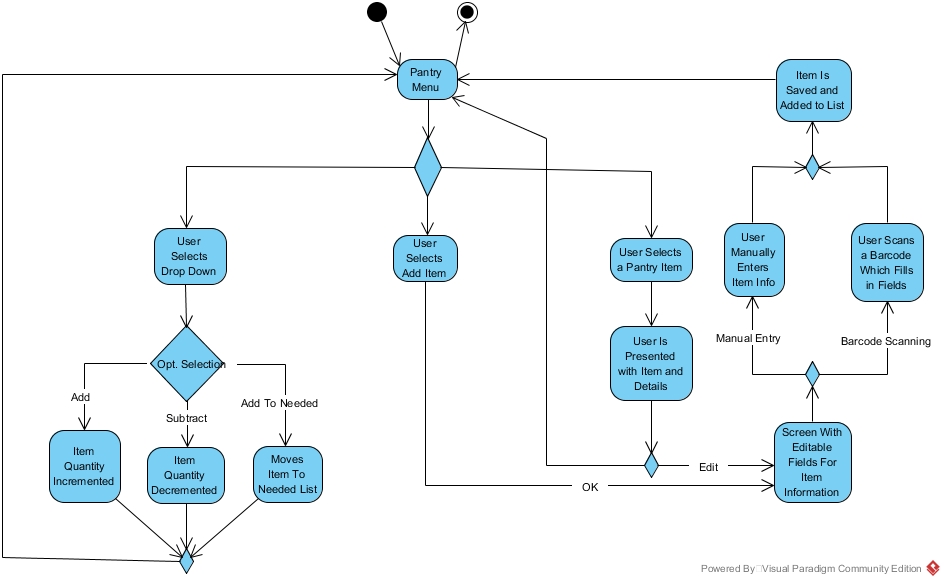
On activity creation, two list views will be populated with tasks and notes from the Local DB. If user clicks on an item, the relevant information will be displayed and the user will be given the option to update information or to promote a note to a task. The user will also be able to create a note or task object in which case the changes will be pushed to the DB and the respective list view will be updated.

* + 1. Calendar



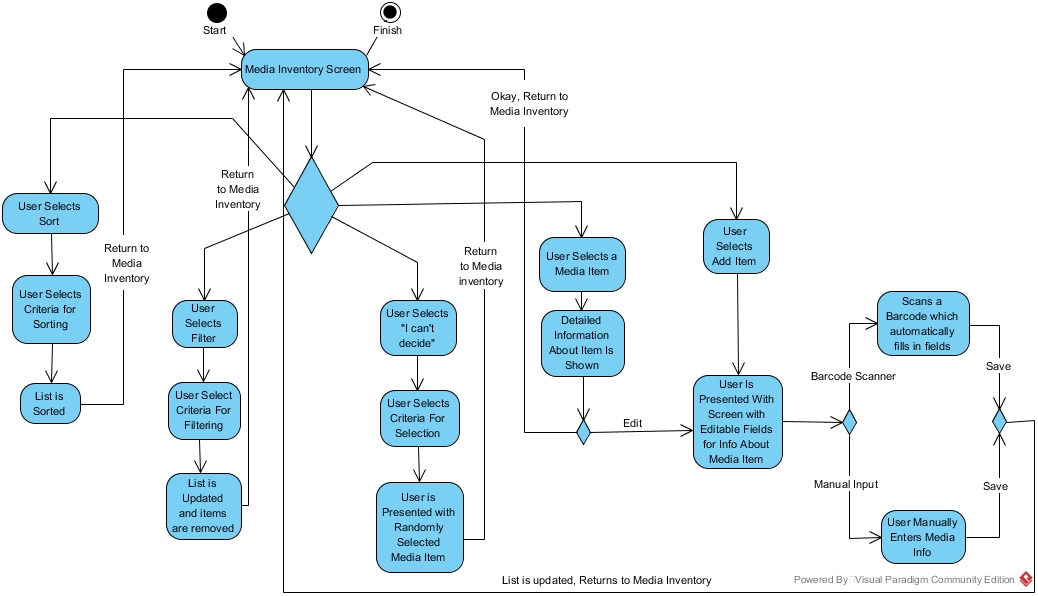
The user will be presented with a calendar view which contains tasks with due dates, and a list view with tasks that are not added to the calendar. On task selection, the user will be able to view and edit if needed, and promote the task to the calendar by adding a due date. The user will also be able to add and delete tasks/calendar events by using dialog boxed to input the relevant information, after which the Item object will be pushed to the DB and the respective views will be updated.

* + 1. Pantry



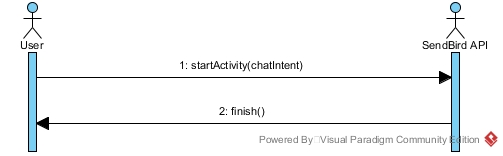
On activity creation, two list views will be populated with data from the local DB. If the user clicks on an item, a dialog box will appear with relevant information about the entry. If a user clicks add, the user will be presented with a dialog box where information can be added then saved to the DB. Filter button and sort button will allow users to filter/sort entries based on criteria.

* + 1. Media



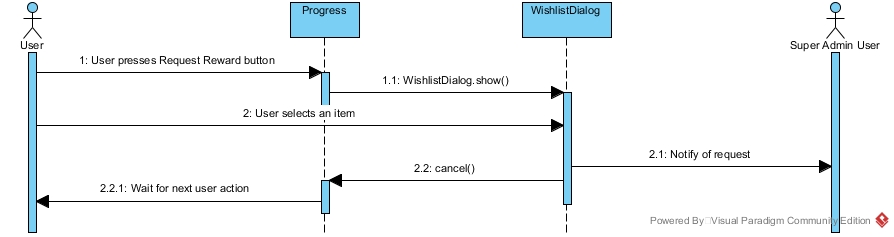
On activity creation, two list views will be populated with data from the local DB. If the user clicks on an item, a dialog box will appear with relevant information about the entry. If a user clicks add, the user will be presented with a dialog box where information can be added then saved to the DB. Filter button and sort button will allow users to filter/sort entries based on criteria. The Random button will return a randomly selected media item.

* 1. D – Sequence Diagrams  
     1. Chat



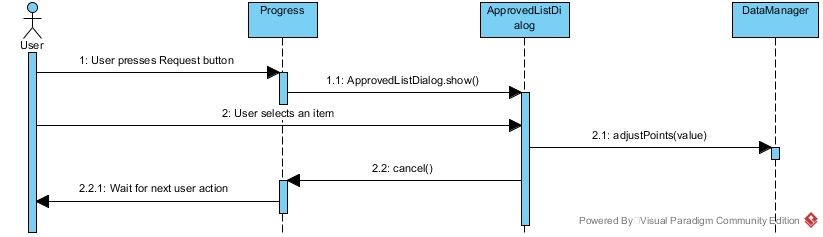
The MainActivity will call the ChatActivity which will call the SendBird API, at which point the API will handle all other user interactions. Upon exiting, user will be returned to MainActivity.

* + 1. Progress
       1. Request Reward



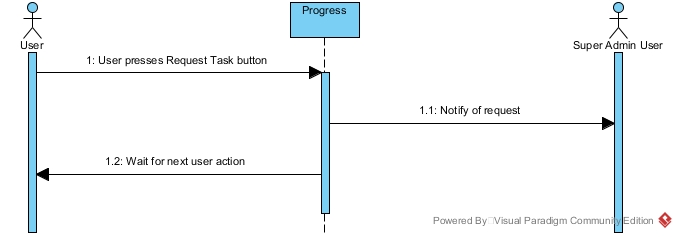
User chooses to request a reward, at which point the Progress Activity will open a dialog box where the user can input information about a desired reward. On save, the dialog box will notify the SuperAdmin user and add the item to the database of potential rewards, finally returning the user to the Progress Activity.

* + - 1. Choose Reward



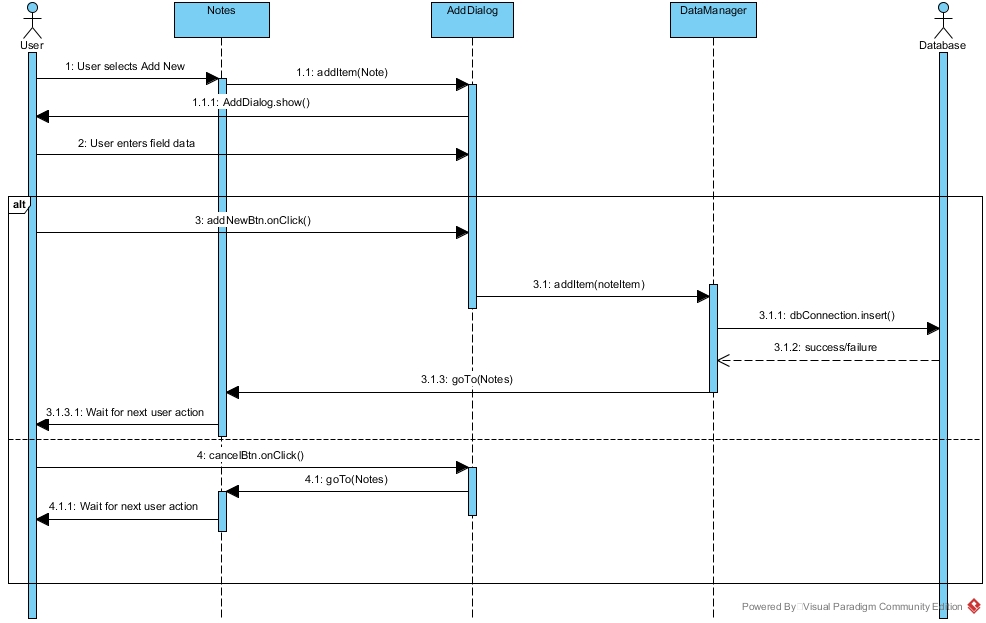
User requests a pre-approved reward. A dialog box containing a list of approved rewards will appear, from which the user can select an entry. On selection, if the user has enough points, the cost of the reward will be deducted, and the user will be returned to the progress activity.

* + - 1. Request Tasks



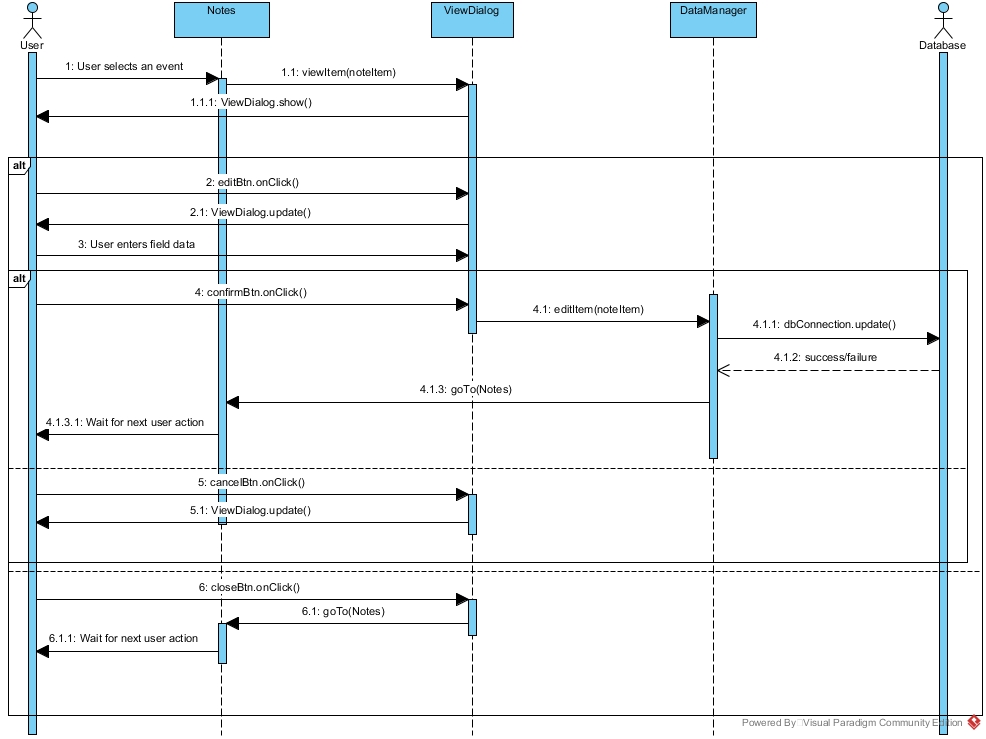
The user requests a task, in which case the Progress Activity will notify the SuperAdmin user that a task has been requested, then the user will be returned to the Progress Activity.

* + 1. Notes
       1. Add Note



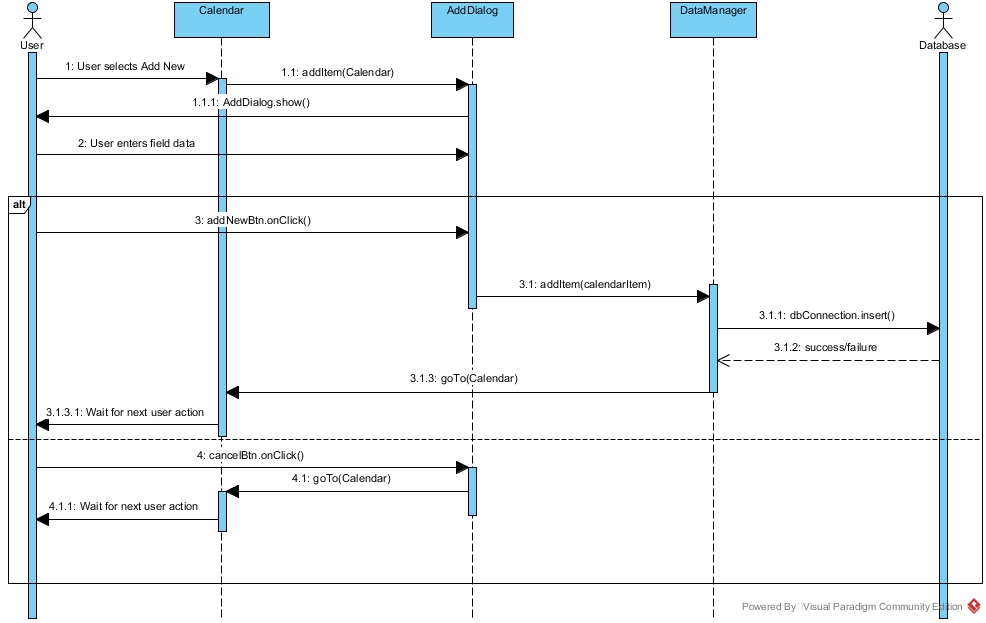
User selects add item, which brings up a dialog box with blank fields in which the user will input relevant information. If the user saves the item, the dialog box will call the DataManager which will push the item to the DB, then return to the Notes Activity. If the user cancels, they are returned to the Notes Activity.

* + - 1. View Note



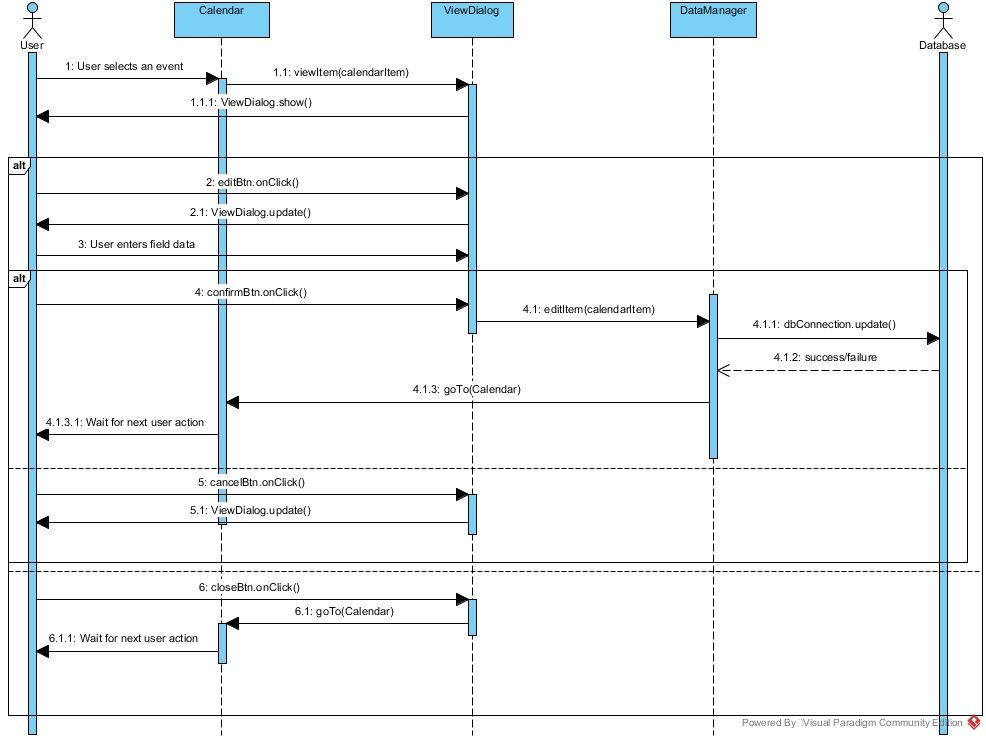
The user selects a note, which brings up a dialog box containing text fields containing relevant information about the note. If the user clicks edit, a different dialog box will appear with editable fields prepopulated with the already entered information. The user will make a change and select save, which will call the DataManager which will send the updated item to the database and return to the Notes Activity. If the user selects cancel, they are returned to the first dialog box. If the user selects cancel on the first dialog box, they are returned to the Notes Activity.

* + 1. Calendar
       1. Add Item



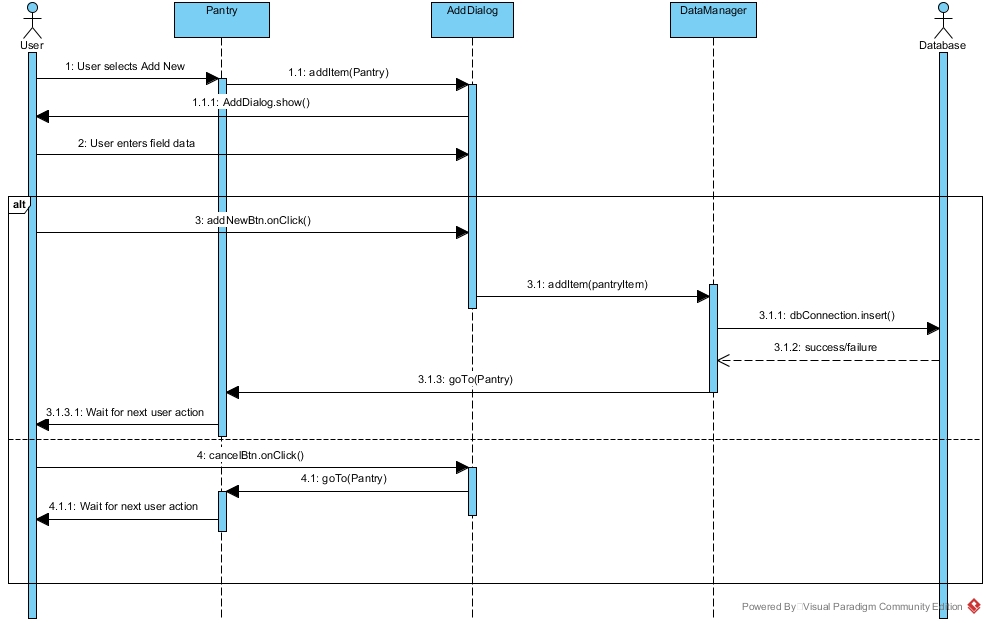
User selects add item, which brings up a dialog box with blank fields in which the user will input relevant information. If the user saves the item, the dialog box will call the DataManager which will push the item to the DB, then return to the Calendar Activity. If the user cancels, they are returned to the Calendar Activity.

* + - 1. View Item



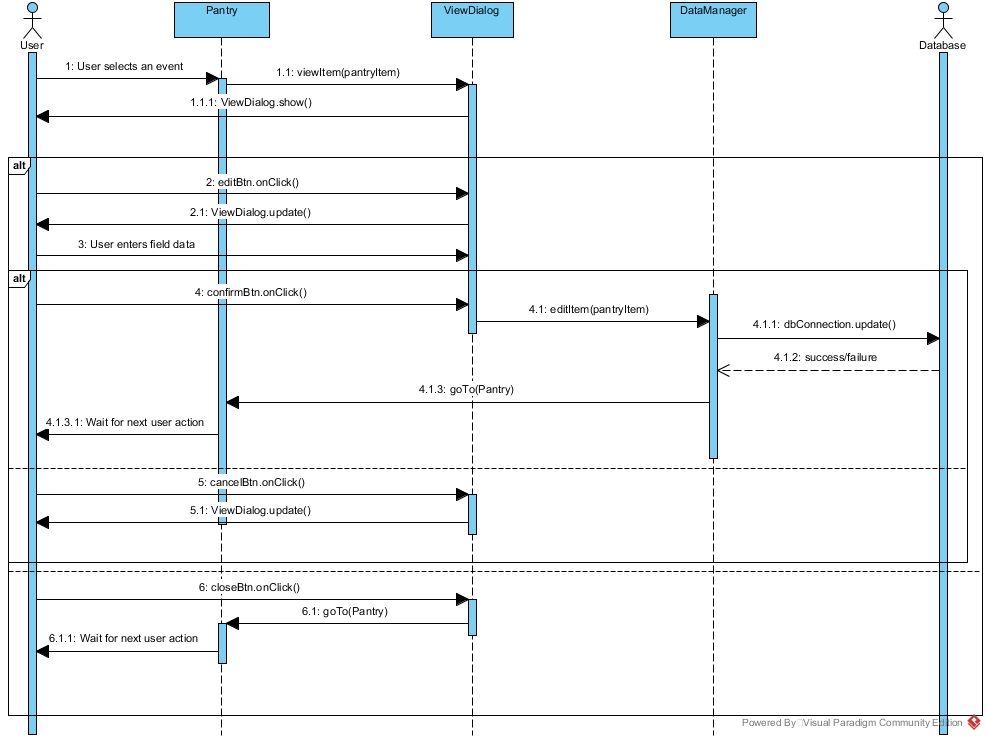
The user selects an event, which brings up a dialog box containing text fields containing relevant information about the note. If the user clicks edit, a different dialog box will appear with editable fields prepopulated with the already entered information. The user will make a change and select save, which will call the DataManager which will send the updated item to the database and return to the Calendar Activity. If the user selects cancel, they are returned to the first dialog box. If the user selects cancel on the first dialog box, they are returned to the Calendar Activity.

* + 1. Pantry
       1. Add Item



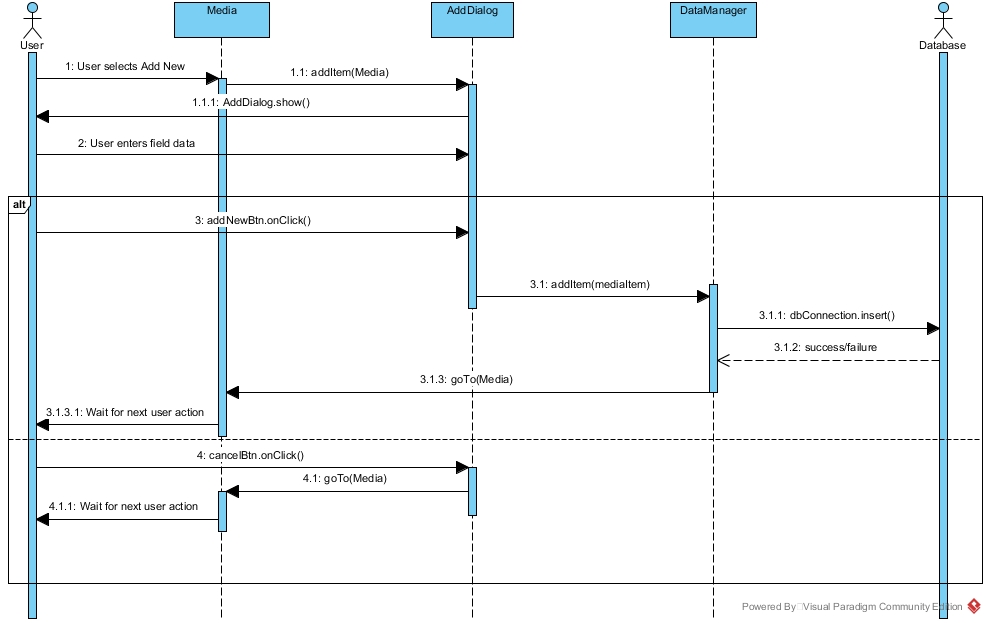
User selects add item, which brings up a dialog box with blank fields in which the user will input relevant information. If the user saves the item, the dialog box will call the DataManager which will push the item to the DB, then return to the Pantry Activity. If the user cancels, they are returned to the Pantry Activity.

* + - 1. View Item



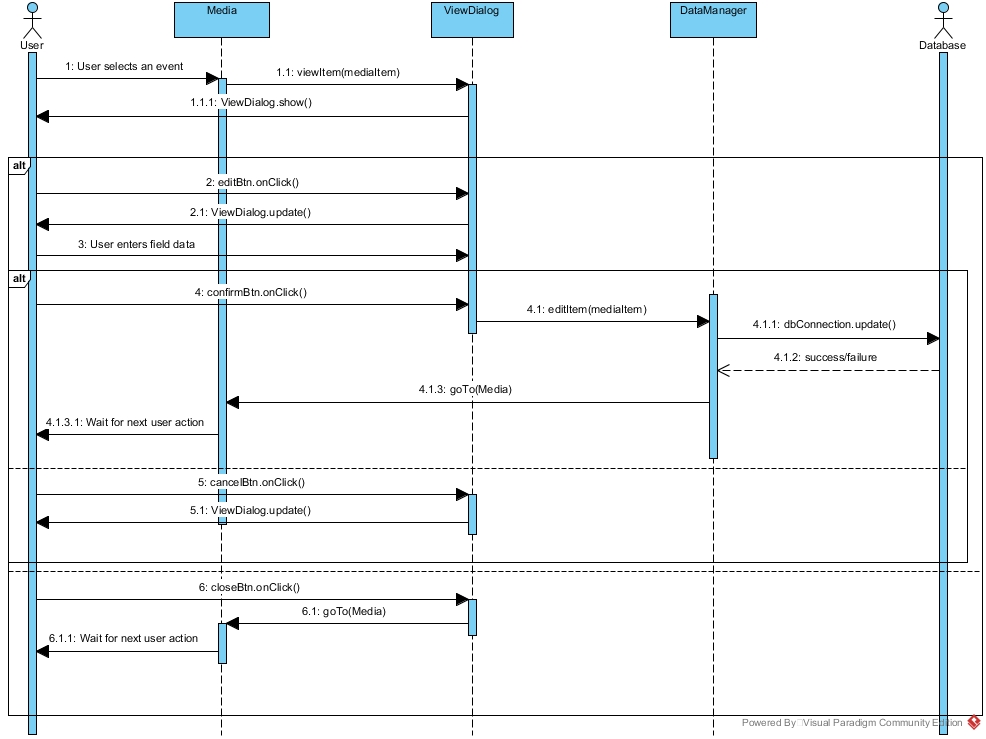
The user selects a pantry item, which brings up a dialog box containing text fields containing relevant information about the note. If the user clicks edit, a different dialog box will appear with editable fields prepopulated with the already entered information. The user will make a change and select save, which will call the DataManager which will send the updated item to the database and return to the Pantry Activity. If the user selects cancel, they are returned to the first dialog box. If the user selects cancel on the first dialog box, they are returned to the Pantry Activity.

* + 1. Media
       1. Add Item



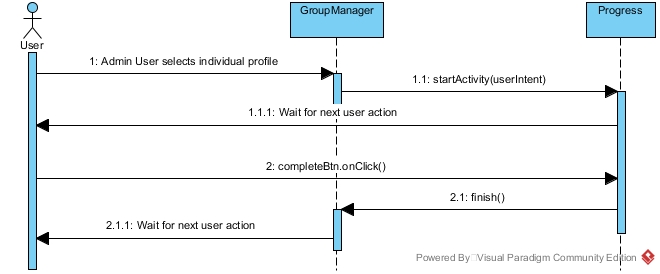
User selects add item, which brings up a dialog box with blank fields in which the user will input relevant information. If the user saves the item, the dialog box will call the DataManager which will push the item to the DB, then return to the Media Activity. If the user cancels, they are returned to the Media Activity

* + - 1. View Item



The user selects a media item, which brings up a dialog box containing text fields containing relevant information about the note. If the user clicks edit, a different dialog box will appear with editable fields prepopulated with the already entered information. The user will make a change and select save, which will call the DataManager which will send the updated item to the database and return to the Media Activity. If the user selects cancel, they are returned to the first dialog box. If the user selects cancel on the first dialog box, they are returned to the Media Activity.

* + 1. GroupManager
       1. SelectUser



The user is presented with a list of profiles in the living group. If the user selects a profile, the Progress Activity is started with that selected profile. When done, user is finished, they are returned to the GroupManager Activity.