GitHub Username: nadajp

Little Talkers

Description

Little Talkers is an app for recording children's first words and expressions. It makes it easy to quickly record these important memories by automatically inserting date and location, and allowing the user to perform audio recordings of the child. Additional details about the expression can be filled in by the user at any time and the expression can easily be shared with friends and family. Phrases are kept organized in a list which can be sorted alphabetically or chronologically. For those never-ending toddler questions or the first time a child answers when someone asks their name, there is a separate 'Question and Answer' category.

Intended User

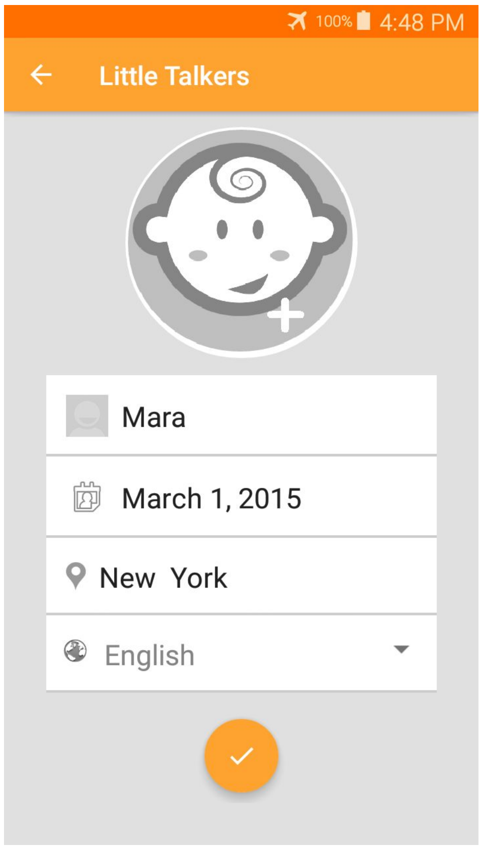
This app is for parents and caretakers of small children who are starting to talk.

Features

- Two categories: word/phrase and question/answer
- Current date, location and child's default language are automatically inserted for every phrase
- Location is automatically detected and filled in. If it is unavailable, default location is inserted
- Automatically filled-in fields can be modified by user at any time
- Optional fields include translation (in case it's not obvious what the child was trying to say), to whom they said it, and additional notes
- Audio-record the child - audio file will be stored on your phone for easy export or it can be shared directly through the app
- View all expressions in a list and sort chronologically or alphabetically
- Share an expression with friends and family via text or email
- Support for multiple children
- See the child's progress by clicking on the word details page, where there is a list of all previous versions of the word, sorted chronologically
- Export data as a CSV file, then email to yourself or your family for easy viewing in any spreadsheet application
- Sync all data to the cloud

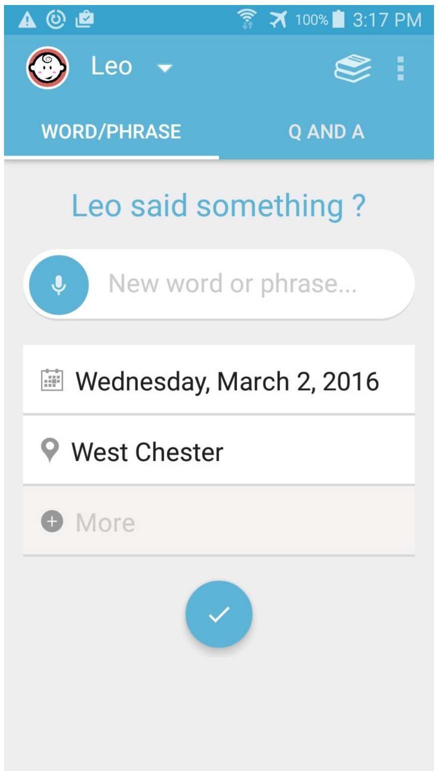
User Interface Mocks

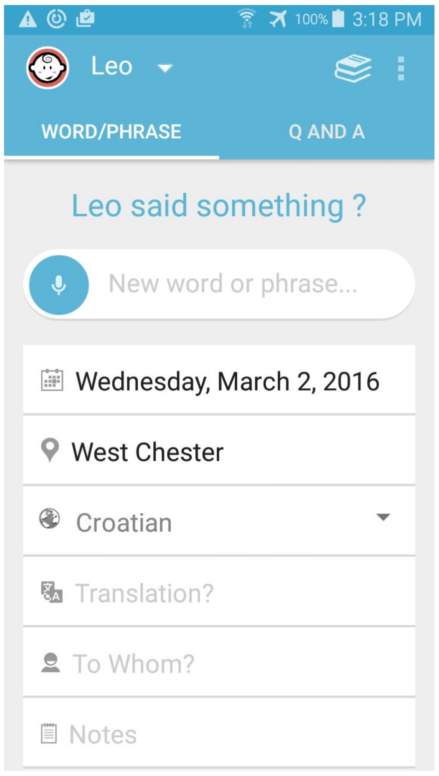
Screen 1



Insert details for a child, including default language, which will be automatically selected for each new word that's inserted. The user will be able to change language in a dropdown list for individual words. The birthdate is used to keep track of the child's age.

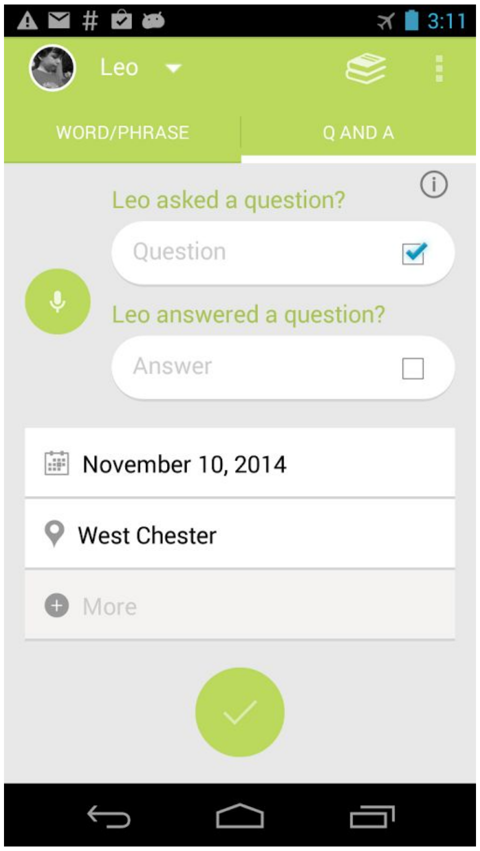
Screen 2





Add Word Screen (shown along the expanded view that includes optional details). User enters word details. The mic button allows user to record a word. Clicking on the next tab switches to Q and A mode. The spinner on the toolbar allows the user to switch to a different child and clicking on the child's picture shows their profile with word counts. (Screen 7)

Screen 3

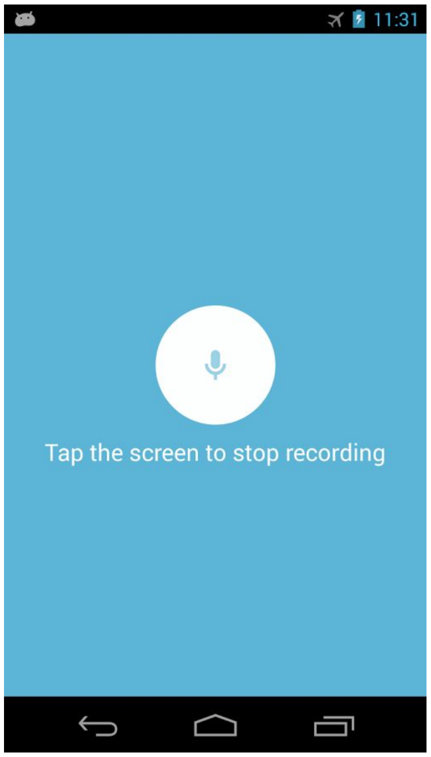


November 10, 2014

West Chester

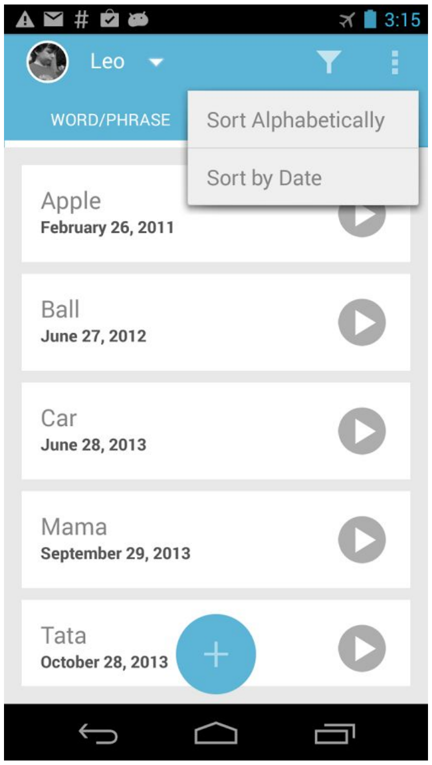
Q and A mode allows the user to enter a question and (optional) answer, then check of the box that indicates which part of the conversation was spoken by the child.

Screen 4



User is currently recording. The mic icon animates to indicate recording in progress and a tap on the screen stops the recording and takes user back to the Add New screen.

Screen 5



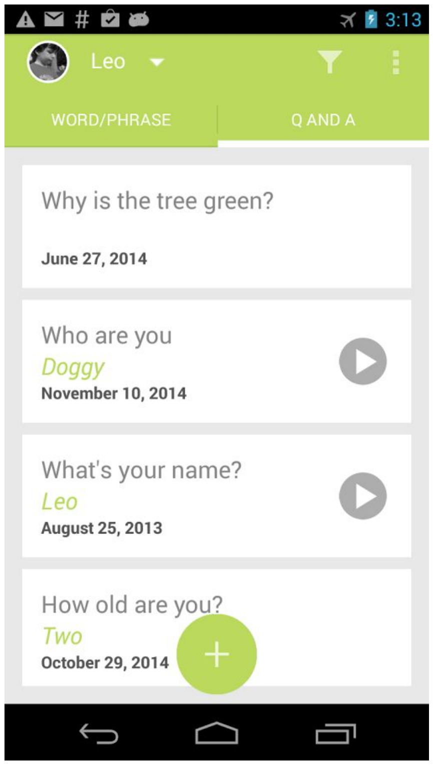
Apple
February 26, 2011

Ball
June 27, 2012

Car
June 28, 2013

Mama
September 29, 2013

Tata
October 28, 2013



Why is the tree green?
June 27, 2014

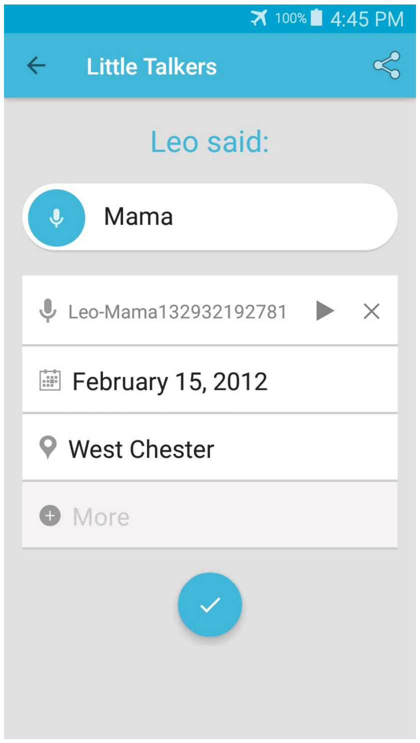
Who are you
Doggy
November 10, 2014

What's your name?
Leo
August 25, 2013

How old are you?
Two
October 29, 2014

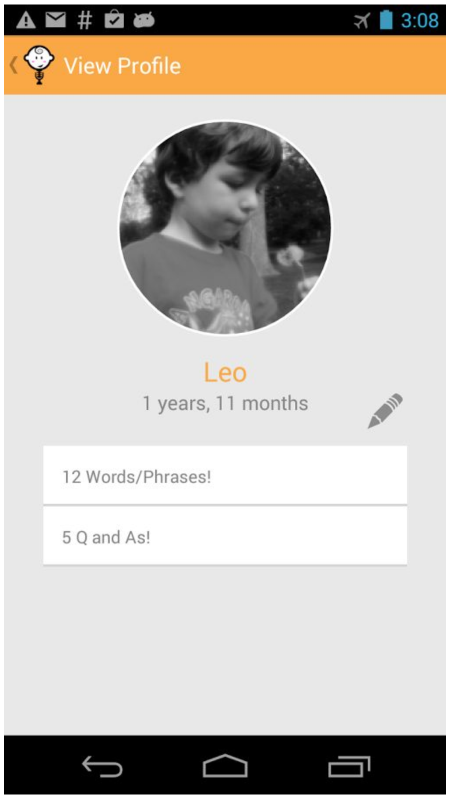
Dictionary views shows all currently recorded words or qa's, along with their recordings, which can be played back directly from the list. Clicking on the plus button takes the user back to the Add New screen, and clicking on a word takes them to the detail view where they can make edits or share the word.

Screen 6



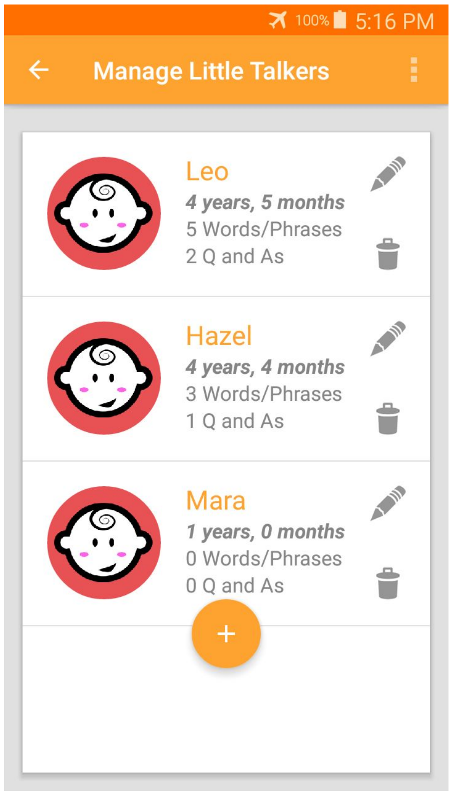
View Details Screen, where the word's details can be viewed and edited, and the word can be shared via share button in the action bar.

Screen 7



View Profile Screen shows the child's age and progress

Screen 7



Leo
4 years, 5 months
5 Words/Phrases
2 Q and As

Hazel
4 years, 4 months
3 Words/Phrases
1 Q and As

Mara
1 years, 0 months
0 Words/Phrases
0 Q and As

Manage Little Talkers allows user to add and manage the children

Key Considerations

How will your app handle data persistence?

The app will use an SQLite database and a Content Provider for data persistence. Audio files will be stored in a folder in the phone’s public storage.

Describe any corner cases in the UX.

The user navigates between Word and QA modes by sliding between tabs. To switch to list viewing, the user clicks on the dictionary icon in the action bar, and to go back to Add New screen, the user clicks on the Plus (Add New) button. When viewing an existing word or a child’s profile, a back button is provided in the action bar to take the user back to the main screen. To stop recording and go back to the Add Word screen, the user taps anywhere on the screen.

Describe any libraries you'll be using and share your reasoning for including them.

- Google Play Services for location detection and admob
- Jackson for JSON support when sending/receiving server data
- Android support libraries, including Material Design support library

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Create an app module and a server module
- Configure libraries

Task 2: Implement UI for Each Activity and Fragment

Create all the above listed UIs inside fragments so they can later be adapted for tablet

- Build UI for adding a child
- Build UI for adding a new word, and one for displaying the words (a list)
- Repeat for QA, reusing any common parts
- Build a custom action bar with tab layout
- Add a ViewPager and connect with tab layout to allow switching between Word and QA
- Add spinner with placeholder for child's image to action bar
- Build UIs for managing children and viewing their profiles
- Connect all UIs and ensure the navigation is user-friendly
- Create resources for buttons and icons that will be used and add them to the UIs
- Implement transitions between activities for supported systems

Task 3: Implement UI for Tablets

Decide on tablet design and implement above UIs accordingly

- Sketch out all screens as they should be displayed on a tablet
- Create master-detail views where appropriate
- Create a dimens.xml file for tablets and adjust margins and dimensions accordingly

Task 4: Set up database and content provider

- Design the database scheme
- Implement database and content provider
- Test the database and content provider to ensure all basic database operations are functioning by creating and running unit tests

Task 5: Connect UI to the content provider

Use the content provider to save a child's information, and then to save new words and load them into the list view

- Implement validation for all user-entered fields
- When the check button is clicked in the Add Kid UI, use content provider to insert the child into the database
- When the add button is clicked in the Add Word UI, use content provider to insert the word into the database
- Use a loader to load all words into the list view
- Use a loader to load default fields for a child in the Add New screen
- Write instrumentation tests to test all components of the UI

Task 6: Implement audio recording

- Investigate popular audio formats for Android
- When the user clicks record, record the audio and animate screen to show that the audio recording is taking place
- When the user clicks again, stop recording, create a new temporary audio file in the Little Talkers folder of public file storage
- When the user saves the phrase, rename file to a unique filename that describes the phrase and insert filename into the database

Task 7: Implement support for multiple children

- Populate action bar spinner with children's names using a loader
- When a new child is selected, notify all fragments so they can update the data accordingly

Task 8: Implement Server

- Create a Google App Engine app with a Datastore to store all components of the app: user info, kid details, word and QA details
- Using Google Cloud Endpoints, create an API that exposes functions for inserting, updating and deleting data from the Datastore

Task 9: Implement SyncAdapter

- When app starts, connect to server and schedule periodic sync
- Implement all components of a SyncAdapter and connect it to the content provider
- In onPerformSync, check which updates have not yet been performed, then upload latest changes to the server

Task 10: Implement Location Services

- When app starts, connect to location services and retrieve current location
- Insert current location in the Add New screen
- If unavailable, insert default location for the child

Task 11: Implement Admob

- When app starts, connect to Admob and load into bottom of screen

Task 12: Implement Widget

- Design and implement a widget that will allow quick access to the Add New interface, including the mic for quick recordings

Task 13: Test

- Create and run unit tests as needed to cover all aspects of code
- Test on a variety of devices with different API levels, screen sizes and resolutions
- Use all app functionalities, rotate screen, insert unexpected data, and use back buttons to ensure that the app always behaves as expected
- Fix bugs and handle all exceptions appropriately

Task 14: Finalize App

- Provide and organize all remaining resources and graphics
- Sign and export app
- Store keystore and passwords in repository and refer to keystore by relative path

---

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"