

GitHub Username: evelynzayas

First Everything

Description

Doing something for the first time can be thrilling! Keep track of those special life moments in one place so that you can re-live them at a swipe of your finger! Whether you want to save your adventurous moments or those precious moments of your little loved ones, the First Everything app makes it fun and easy to capture the what, when, and where of significant life events. Record it as text, image, sound and video - whatever works for you and your moment.

Intended User

Persona #1: a proud parent wanting to capture baby's firsts.

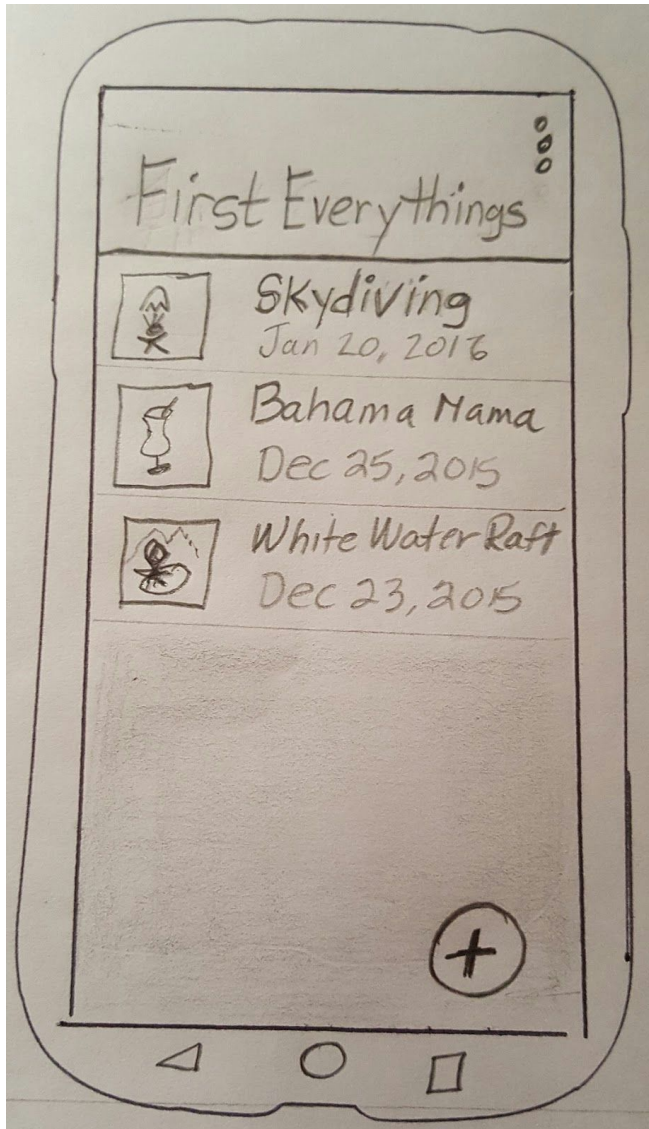
Persona #2: anybody who wants a quick and easy way to keep track of "first times" and significant life moments in one place for quick recall, review and share

Features

- Saves relevant information about a life moment in one place: what, where, when, and with who in multimedia - text, images, audio, video, geo
- Scroll through life moments
- Share life moments

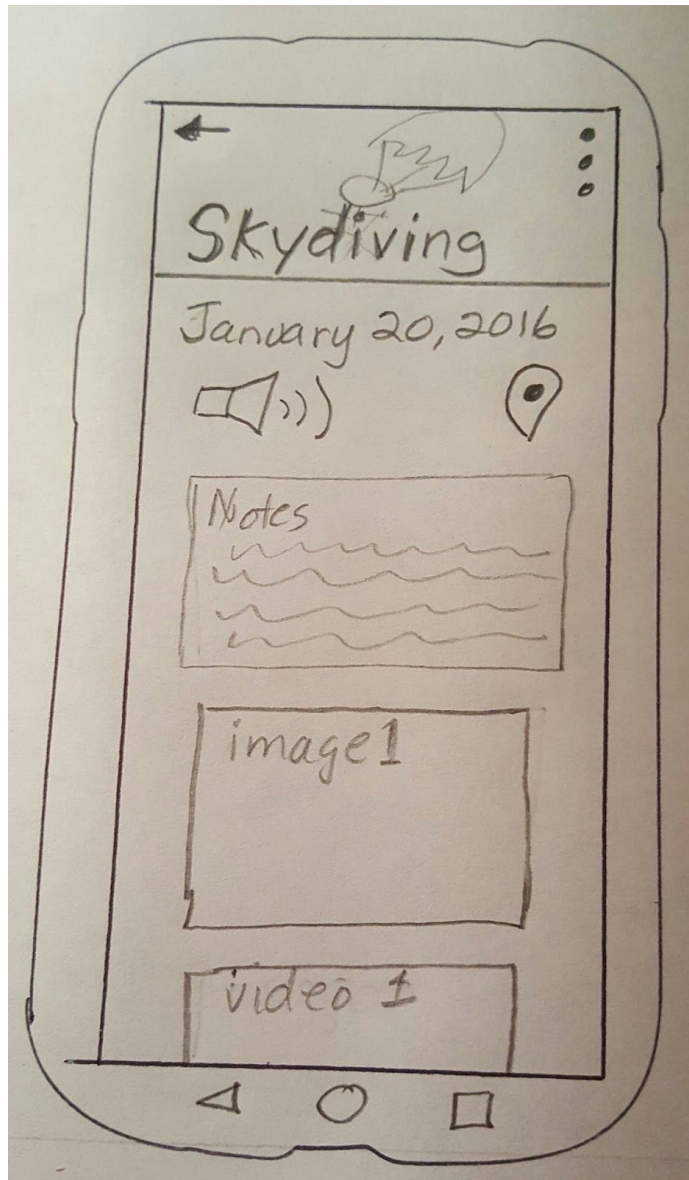
User Interface Mocks

Home Screen



The home screen is a scrollable list that will display the life events as they are entered. There is a FAB to add a new life event. The overflow menu will provide access to an about and a help modal window.

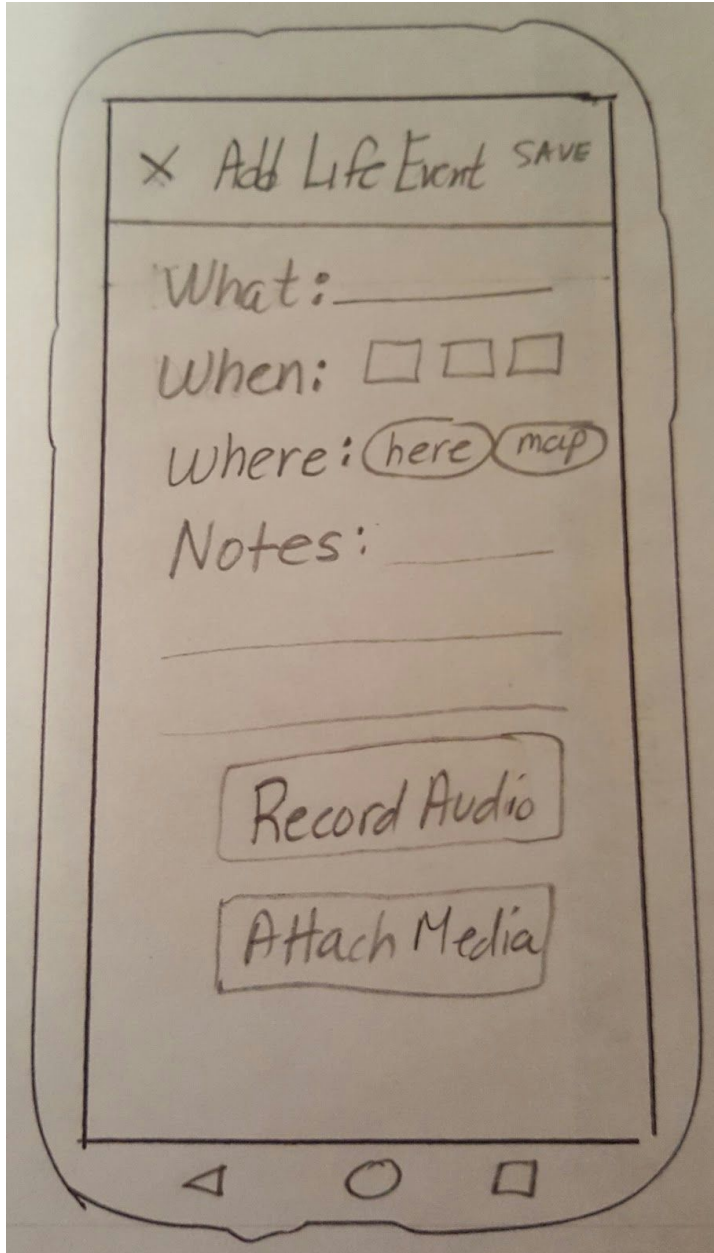
Detail view of a Life Event



The detail view screen will include all of the data entered through the Add Life Event screen. It will have a collapsible toolbar and include the name of the (free form text) event name, date, an icon to open the audio player to listen to optional recording, an icon to open Google Maps that shows the location of the event, the (free form text) journal-like entry, and the images and/or videos that were associated with this record. The overflow menu provides access to the share option (edit option in future release).

Addendum: there will be a video icon between the speaker and geo pin icons to launch media player. There will be an ImageViewer so users can swipe through if multiple images.

The 'Add Life Event' screen



The Add Life Event screen will include data entry fields to include a text field for the name, a data picker, buttons that will set the current location or open Maps to pin a location, multi-line text field for journaling, a button to open the voice recorder, and a button to open the media selection. The toolbar has a save or cancel option.

Key Considerations

How will your app handle data persistence?

For this first version, the life event data (i.e. text, location data, image file local path) will be stored locally on the device using SQLite. I envision future versions to allow users to upload the data to their Google Drive for backup purposes.

Describe any corner cases in the UX.

This app will use a master-detail structure. The home screen will contain the list of life events that the user can scroll through and selecting a life event from the list will launch the detail view of it. Media attached to the detail view, such as an image, video or sound file, will launch as a modal window. Each image will be displayed with a swipe gesture. For a video or sound file, standard media player controls will be displayed. The UI action to see the location of the event will open Google Maps with the location pinned on the map. The back button will close the modal window or go back to the list. In detail view, the overflow menu will provide the options to edit or share the event. On the home screen, there will be a floating action button for adding a new event, and an overflow menu with about and help options. On the 'add event' screen, there will be text fields and buttons for data entry, with save and cancel options on the top toolbar. Saving a new event will then show a toast or snackbar stating success, and the user will be returned to the home screen. Canceling out of a new event will prompt the user to confirm the discard action, and then return the user to the home screen.

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

Design support library to have access to material design elements.
Google Play Services for mapping.
SQLite for local data storage.

Next Steps: Required Tasks

Task 1: Project Setup

- create new project with activity fragment
- configure libraries and add dependencies

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity, which is the main screen which will query the database and return the results in a ListView.
 - UI includes FAB for adding a life event
 - overflow menu: about, help/how-to
 - each shown in a dialog
 - corner case: if no life events are found (new installation), display a toast encouraging input.
- Build UI for Detail Life Event View, which will be launched via Intent when user clicks on a specific life event in the ListView. The UI will display the following information:
 - life event (text)
 - date of event
 - speaker icon to launch via Intent to put up simple player UI to listen to audio recording
 - video icon to launch via Intent Simple Media player (NOTE: the mockups don't show this)
 - map icon to launch via Intent to put up Google map window to show pinned location
 - notes (text)
 - image switcher
 - overflow menu: share
- Build UI for Add Life Event. The UI will prompt for the following information:
 - life event (text)
 - date of event (picker)
 - buttons to annotate capture of geo info
 - notes (text)
 - button to launch audio recorder
 - button to launch multimed media picker (images, video)
 - SAVE and CANCEL options
- Build UI for Simple Media Player (audio or video)
 - Play/Pause Button - (Displays "Pause" when media is currently playing & displays "Play" when a playback is stopped)
- Build UI for Simple Map Viewer
 - TBD

Task 3: Implement SQLite DB

- create DB statements

- insert into DB statements
- query DB and return results statements
- integrate results with ListView through adapter

Task 4: Testing

- create and run the test cases
- user alpa testing

Task 5: Optimize end to end experience for tablet

- Create alternative layouts
- Migrate the existing UI flow to use a master-detail structure

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**"