GitHub Username: pedrofraca

BabyDiary

Description

When you have a baby you spend a lot of time just taking photos or recording videos, most of the time those videos are lost on the gallery and it's hard to find those, same happens with your baby weight and height history, in the best case you will have a book where you have noted all the figures, but you are not carrying this book all the time and checking a specific figure for a specific date can be hard. And what about vaccines? Do you know the exact date when your baby receives a vaccine?. With the help of this app you can track all this information, store it on your mobile device and query whenever you want.

Intended User

Parents, who wants to track his baby/ies information and media.

Features

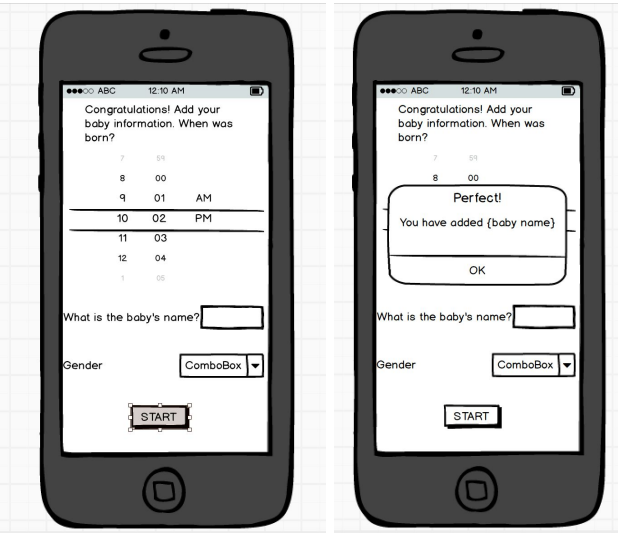
Main features of the app:

- Take a photo of your baby and add a date or add a description.
- Record a video of your baby in a special moment.
- Save the weight and height of the baby for a specific moment.
- Have a list of all the vaccines the baby has received.
- Search in all the records for a specific word, tag or date.
- Record an audio of your baby saying something.
- Review all the information stored in the app navigating through the timeline.
- Attach the location to a specific event to identify where the event has happened.
- View all the events on a map.

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



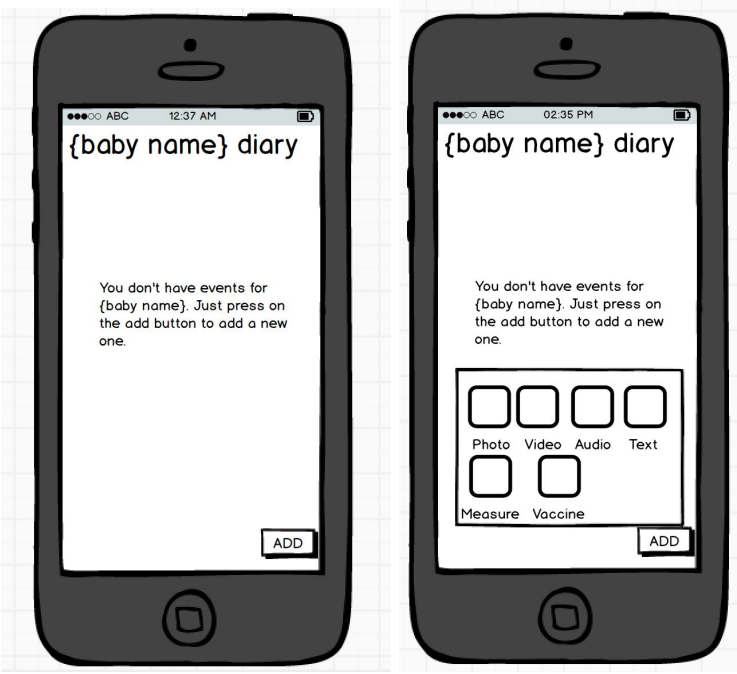
This screen will appear after the app is opened. Taking into account that this is the first screen and we don't have any baby in the app added we will ask to the user to create one.

We can use the same screen to add all the attributes or we can create a screen for each attribute. For each baby we are going to ask for:

- Date of birth
- Baby's name
- Gender

An alert dialog/snack bar should appear once the baby has been created correctly.

Screen 2



This screen will appear once you have created a baby, and when you don't have any action created for that baby. We alert to the user that the baby doesn't have actions and we recommend to create a new one. The add button should be a Floating Action Button.

When the user clicks on the add button a popover similar to the one that whatsapp shows to add an attachment to a chat will be shown.

Possible Events to add are:

- Photo
- Video
- Audio
- Text
- Measures
- Vaccine

The location can be attached to every event as an optional parameter (it will be prefilled with the user current location) also an extra description can be also added to each event. For the Measure and Vaccine event also an image or video can be also attached, this will be really helpful to store the serial number of the vaccine as an attachment, or to record a video of the baby on the bascule.

Screen 3

This is the description for the timeline screen filled with events.

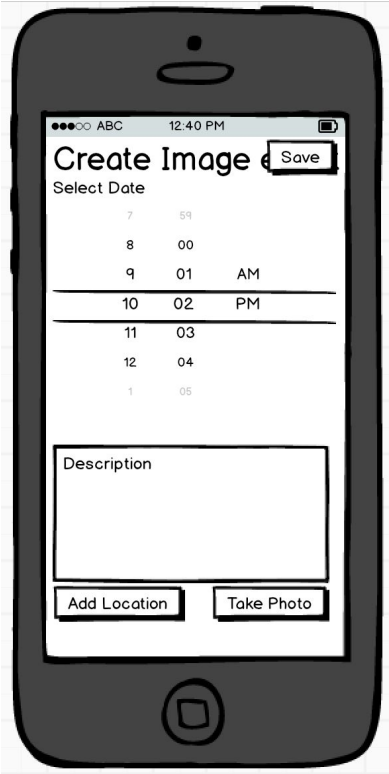
A search button will be available on the toolbar and the user can search by an event, the search will be done against the event description.

Once an event has been added it should appear on the timeline screen as a card. We should show always the date when the event has happened in all cards and the relevant information for that specific event:

- If the event contains an image we should show a preview of the image, as image background. This will generate a colorful timeline.
- For a text event we should show the text as the title of the card.
- If the event is an audio, the user should be able to play it from timeline.
- For a vaccine event, we should show the name of the vaccine and the date when was applied to the baby.
- For a measure event, we should show in big the measure for the height and the weight at the same time we can show a comparison against the previous measurements.

This screen will have also the option to open a DrawerMenu with some additional options, please take a look to the task that describes this screen implementation for further details.

Screen 4



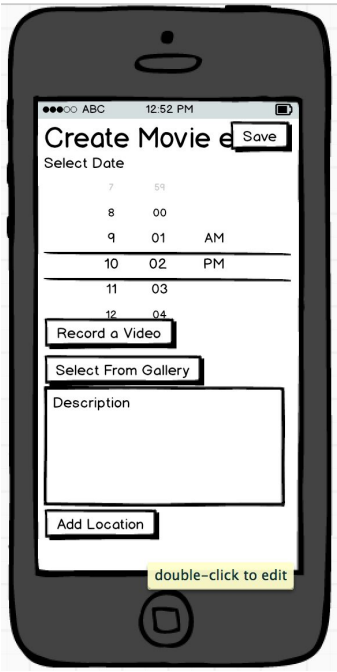
This screen is used to create an image event. The image can be choose from the gallery or from the camera directly.

The DatePicker component will be filled by default with the current date, the user can also add a description to the event using the EditText on the screen.

The save button should be on the ToolBar.

Pressing the Add Location button a new screen will appear and the user will be able to select the current location or find a place to select.

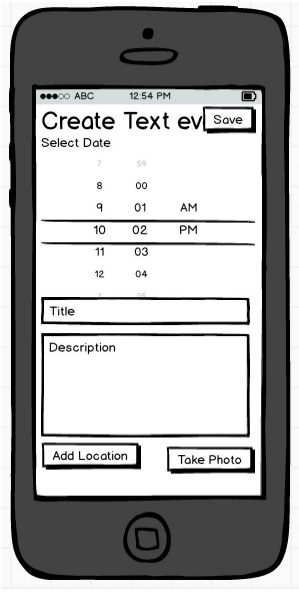
Screen 5



This screen is used to create a video event. The video can be recorded or just choose it from the gallery.

As in the image event we can add a location and set a description.

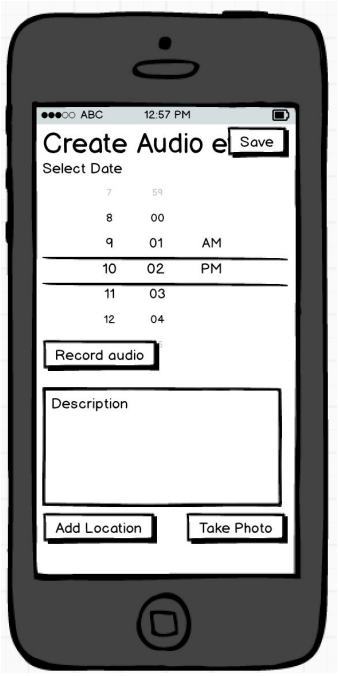
Screen 6



This screen is used to create a text event.

As in the image event we can add a location and set a description, and we can also take a picture and attach to the event.

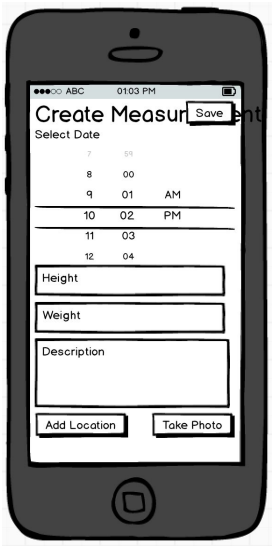
Screen 7



This screen is used to create an audio event.

As is the image event we can add a location and set a description, and we can also take a picture and attach to the event.

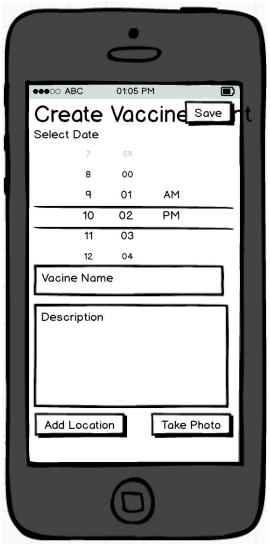
Screen 8



This screen is used to create an measurement event.

As in the image event we can add a location and set a description, and we can also take a picture and attach to the event.

Screen 9



This screen is used to create a vaccine event.

As in the image event we can add a location and set a description, and we can also take a picture and attach to the event.

Screen 10

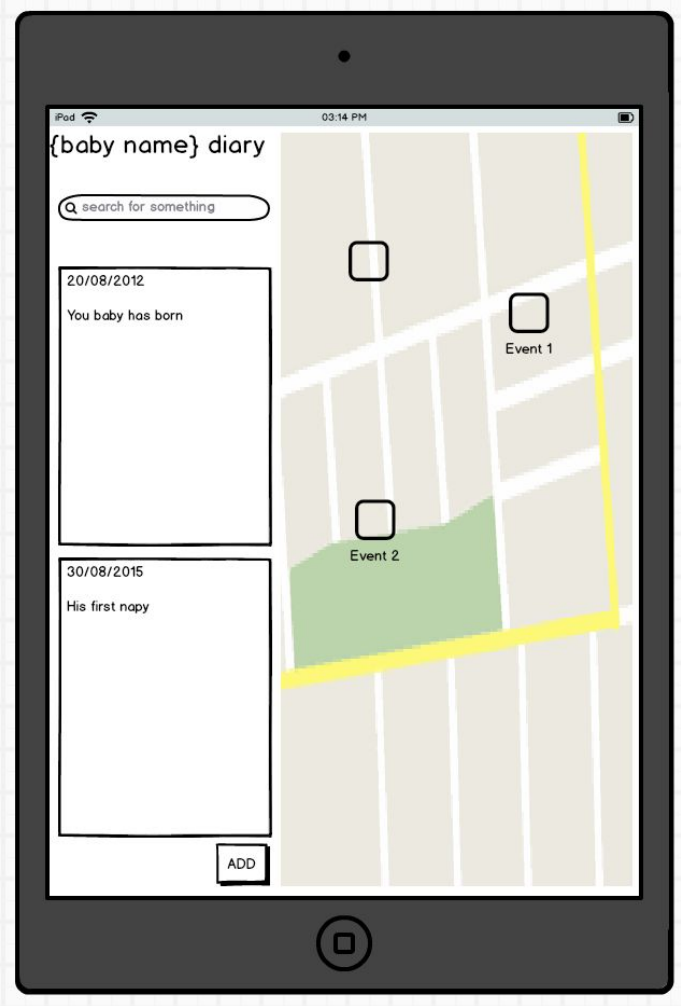
This screen will appear when the user clicks on the add location button on every event.

By default the current device location will be selected. And the user also has the opportunity to search for a location entering a text and select it from the list.

Screen 11

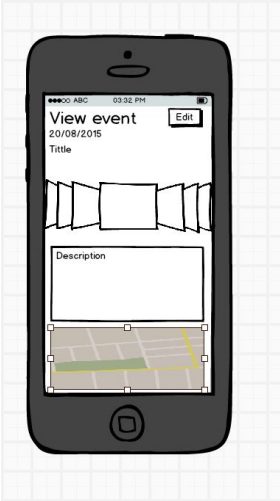
This screen will place all the events on the map, just to navigate through all the evest and to see where those event happened.

Screen 12



This is how the main screen should appear on a tablet. With the timeline on the left and the map with all the events on the right.

Screen 13



This screen is created to show an event detail. Here the user will be able to navigate through all the media attached to that specific event or take a look to the location where the event happened. The location will appear on the map.

Key Considerations

How will your app handle data persistence?

With a local database managed by a content provider.
The videos and the photos will be stored on an app folder.

Describe any corner cases in the UX.

The app will send monthly notifications to the user, just saying that the baby has one month of life more and suggesting to include some data on the app. If the user clicks on a specific notification the timeline will be opened and the user can create a specific event from there.

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

- Google Maps library to show a map view with all the events.

- https://github.com/BoD/android-contentprovider-generator Contentprovider generator to generate the first version of the content providers.
- Google ad mob, we want to make some money with the app. ;).
- Google Places API. We need it to use on the select location functionality.

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

As we are going to use the google maps SDK we should create a google maps key to use into the app and prepare the project with the required libraries dependencies.

For the content provider we should create here all the files needed for the content provider generator and generate all the needed classes.

We should create also an app on google ad mob and generate some ads to include in the app.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for CreateBabyFragment, this fragment contains all the information needed to create a Baby:
    - DatePicker to select the date of birth.
    - EditText to enter Baby's name.
    - Spinner to select the baby gender.
    - Button to confirm the creation of the baby.
- Build UI for CreateBabyActivity, this activity will be launched the first time the app is opened and will load the CreateBabyFragment inside.
- Build UI for the TimelineFragment, contains a RecyclerView where all the events are listed. If no events on the app we shouldn't show the Fragment and a message alerting to the user to create new events using the FloatingActionButton.
- Build UI for the ViewEventFragment, this fragment will show the details for a specific event.
- Build UI for the MapEventFragment. This Activity will appear only if the device is a mobile device, and will be launched when the user clicks on a card on the TimelineFragment.

- Build UI for the TimelineActivity, this activity will show only the TimelineFragment if it's on a mobile device, if the app is opened on a tablet we should show the TimelineFragment on the left side of the screen while on the other part we should load the MapEventFragment with all the events loaded.
- Build UI for the CreateEventFragment, this fragment will be responsible for creating events. And it should contain fields
- Build UI for the SelectLocationFragment, this fragment contains a MapView with the selected
- Build UI for the ViewEventFragment. This frament will show all the information related to an event.

Task 3: Implement logic to insert on the database the information for a baby

- Connect the CreateBabyActivity with the database. Just inserting the information for a Baby on the database.
- Add logic to show only this activity if is the first time the app is launched or if the user press on the menu to add a baby.

Task 4: Implement logic to load information from the events on the TimelineFragment.

- Connect TimelineFragment with the database and to load all the information related with events.
- As TimelineFragment has a RecyclerView we should draw each event differently depending on the event type.
- This fragment must be able to notify to the container Activity that the user has clicked on any event.
- The events on the TimelineFragment should appear on inverse chronological time, same as Facebook app does.
- Implement logic to search by text description on any event.

Task 5: Implement TimeLineActivity

- Implement logic to distinguish if the app it's opened on a mobile device or on a tablet. If it's opened on a mobile device, it should load only the TimelineFragment, on the other hand if it's opened on a tablet it should load the TimelineFragment (by default the first time we enter on a tablet the first element on the RecyclerView should be selected) on

the left and the MapEventFragment on the right for the selected event with all the events loaded.
- This activity should contain a FloatingActionButton, when clicked the Activity to create a new event should be launched.
- Implement (Take a look to Screen 3) logic to open a DrawerMenu to enter to the apps menu to open other Activities. Options that this menu should contain are:
    - Change to another baby, if the user has more than one baby, on the DrawerMenu should appear a sub menu for each baby. We should read from the content provider how many babies we have on the database and his names.
    - Button to create a new baby. Clicking on this button the CreateBabyActivity will be opened.
    - Button to access to the MapEventActivity. Where the user can see all the events in a map. This option should not appear on a tablet.

Task 6: Implement CreateEventFragment

- Implement all the logic for this fragment to support all the event types.
- One or more images can be attached to an event.
- Implement the logic to save the event information on the database.
- Implement the logic to save the media related to this event and link internally to the event.
- Implement a snackbar or another notification to tell to the user that we have saved the event correctly.

Task 7: Implement SelectLocationFragment

- Implement MapView to show the current location.
- Implement the Places Api communication to query for places
- This fragment should notify to the activity which contains it about the location the user has selected.

Task 8: Implement MapEventFragment

- Implement the logic to load all the events on the map
- Implement the MapView to connect correctly using the maps api.

Task 9: Implement Service to send monthly notifications to the user.

Implement a service to send notifications to the user every month. Alerting that his baby has one month more of life and suggesting to the user that he should create an event to celebrate it or something.

Implement the PendingIntent to launch the TimelineActivity when the user clicks on the notification.

Task 10: Implement ViewEventFragment

- Implement the logic to show the event information on this fragment.
- Show all the media attached to this event.
    - Show a set of images like a gallery. Allowing to the user to do zoom in or zoom out.
    - Allow to reproduce to video attached to the event.
- Implement a MapView to show the event location.
- Launch the CreateEventActivity if the user clicks on the edit button on the toolbar.