

Blood Bank:

Team Members : Hari Krishnan Raj Kumar, Lakshmi KS, Nidhi Kumar, Shristi Shrivastava

Basic functionality implemented:

- We have added the Blood donation functionality, This allows the hospital user to fill a form with required blood group, quantity required and description, which will be sent to the users.
- We have added the Event creation functionality, where the hospital user can create an event by entering the event details which will be displayed for the users.
- We have added the nearby hospitals functionality which helps the users filter nearby hospitals, this is done using the lat long calculation where we have developed an algorithm to find the closest ones to a particular latitude - longitude.
- It takes into account the 5-mile radius.
- We have added functionality to open google maps with a hospital's address which can be triggered by a button click by the user.
- Added functionality to list all donors.
- Added functionality to have different UI and functions based on User role - Donor/hospitals. The password is encrypted using the firebase standard encryption method.

Proper state management:

Please find an example explanation based on financial donations page, The same logic is implemented in all functions:

onCreate(): When the DonationsPage is created, the constructor is called to initialize the widget. Here, the DonationSource parameter is set to identify the source of the donation.

onStart(): The widget is being displayed, the build() method is called to create the UI elements.

onResume():

In Android, onResume() is called when the activity is ready for the user interaction. In our case, when the widget is rendered, the user can interact with it, such as entering the donation amount.

State Mutation (User Interaction):

When the user interacts with the widget, such as entering an amount in the text field, it triggers a state change. This is when the user interacts with UI elements in an Android app, triggering state changes in the activity or fragment.

setState() (UI Update): When the state changes, for example, when the user enters an invalid amount, setState() is called to update the UI to reflect the new state.

Async Operations (donate()):

Performing asynchronous operations, such as making a network request, is similar to

performing background tasks in Android, typically done in `onPostExecute()` of an `AsyncTask` or in a background thread. In our case, when the user initiates a donation The state management approach used in the `DonationsPage` widget is primarily handled through the `setState` method. This method is called whenever there's a change in the state of the widget, causing the UI to rebuild with the updated state.

The `TextEditingController` handles the `onPause` state.

Stateful Widget: `DonationsPage` is a stateful widget, indicated by its `StatefulWidget` superclass.

State Object: The stateful widget has a corresponding state object named `_DonationsPageState`, where the stateful logic resides.

State Variables:

amountController: A `TextEditingController` to control the input in the amount text field.

isValidAmount: A boolean variable to track whether the entered amount is valid or not.

State Mutation:

The `setState` method is called when the amount text field's value changes. Inside the `onChanged` callback of the text field, `setState` is invoked to update the `isValidAmount` variable based on the validity of the entered amount.

Remaining work to be completed for the next phase of the project:

Email notifications to the users when blood donation request is published.

Input validations using regex

Age validations for donors (min 18yo)

Validations for blood donations page.