Smart Wallet

the wallet just got smarter

Team:

1. Siddhartha Dutta (120040005)
2. Mihir Kulkarni (12D020007)
3. Shivam Garg (12D020036)
4. Aditya Nambiar (12D070012)

The problem

People are clueless about where their money is spent: restaurants, malls, lending, daily stuff.

Two things people always carry with themselves, watch and phone.

The phone got smart.
What about the Wallet ?

We make it super easy to keep track of your expenses by building an add-on (hookable card-sized screen and an app) that tracks payments and their location.

Project Description

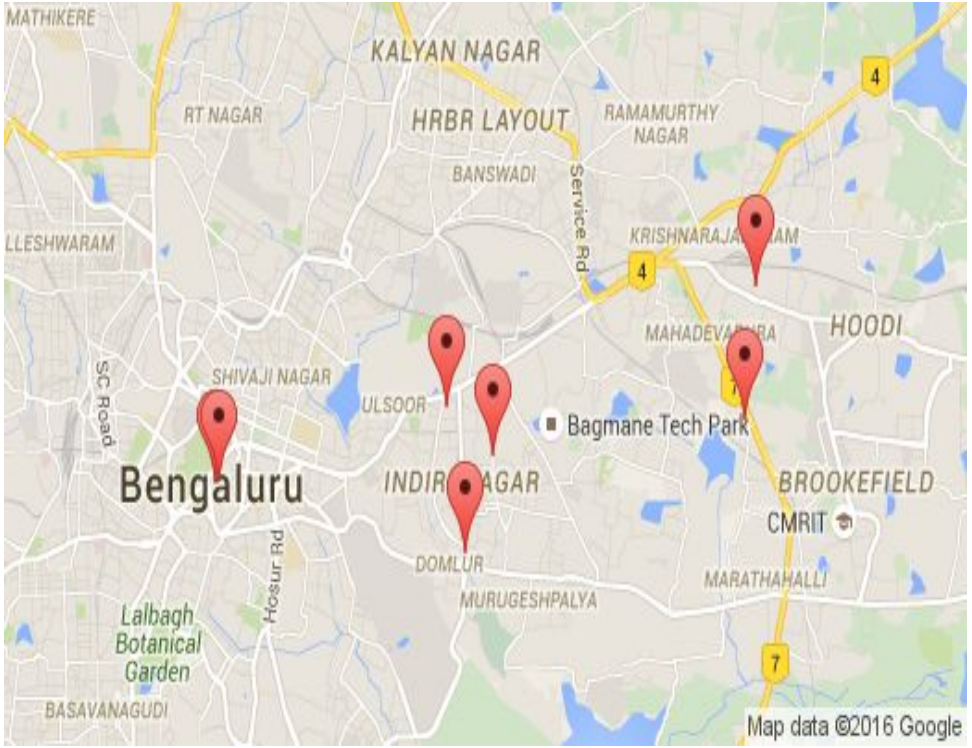
A small add-on for your wallet which can help you keep track of your cash transactions, act as a calculator and also gives you anti-theft security.

Step 1	Step 2	Step 3
Transaction	**Enter Data**	**Sync**
You buy an item for 100 bucks. You take out your wallet, pay the cash.	The proximity sensor in on the device realises the wallet is open. It switches on and prompts you to enter the amount	The data you enter is stored on the device. As soon as the wallet is connected to the phone, all data along with the location and timestamp is sent to the phone via BLE.

At the end of each month, you have a log of where you've spent and received cash.

Your mobile app also tells you where each transaction has been made.

Your cash flows are easily visible, and you can figure out where your major spending happens.



Task Specifications

- Android App:
  - Choose from a list of paired devices and connect to the right smart wallet
  - View data transmitted from smart wallet module(includes timestamp, amount, sent/taken info)
  - Allow edits to be made to data or add info like who borrowed or lended
  - View the place of transaction on a map
  - Make the data persist over restarting of app/ mobile
- Smart Wallet hardware module:
  - Make a calculator using TIVA board, matrix keypad and LCD screen
  - Integrating a bluetooth module
  - Storing data when hardware module not connected to mobile and transmitting it on reestablishing connection
  - Theft detection- Buzzing and notifying on wallet theft

Project Plan

a) **Plan followed to execute the task completion**

- Read and learnt android coding and usage of hardware peripherals(bluetooth, keypad, lcd, rtc)
- Initially planned to reverse engineer a small calculator but later due to some voltage problems, made our own calculator using lcd display, keypad and tiva processor
- Made a basic app and got the bluetooth h/w module to pair and communicate with the mobile bluetooth
- Separately got the keypad to work which required pull-down resistors and then got the lcd to work which required using pot, parallely added GPS and data editing support on the app
- Added support for real time clock and buffering of data, parallely made data persistent on the mobile filesystem
- Enclosed the entire hardware setup in an aesthetically appealing encasing

b) **Work division** critical tasks

Siddhartha and Aditya : focussed mainly on the app (software part)

Mihir and Shivam : focussed primarily on the embedded hardware part

Dates of completion

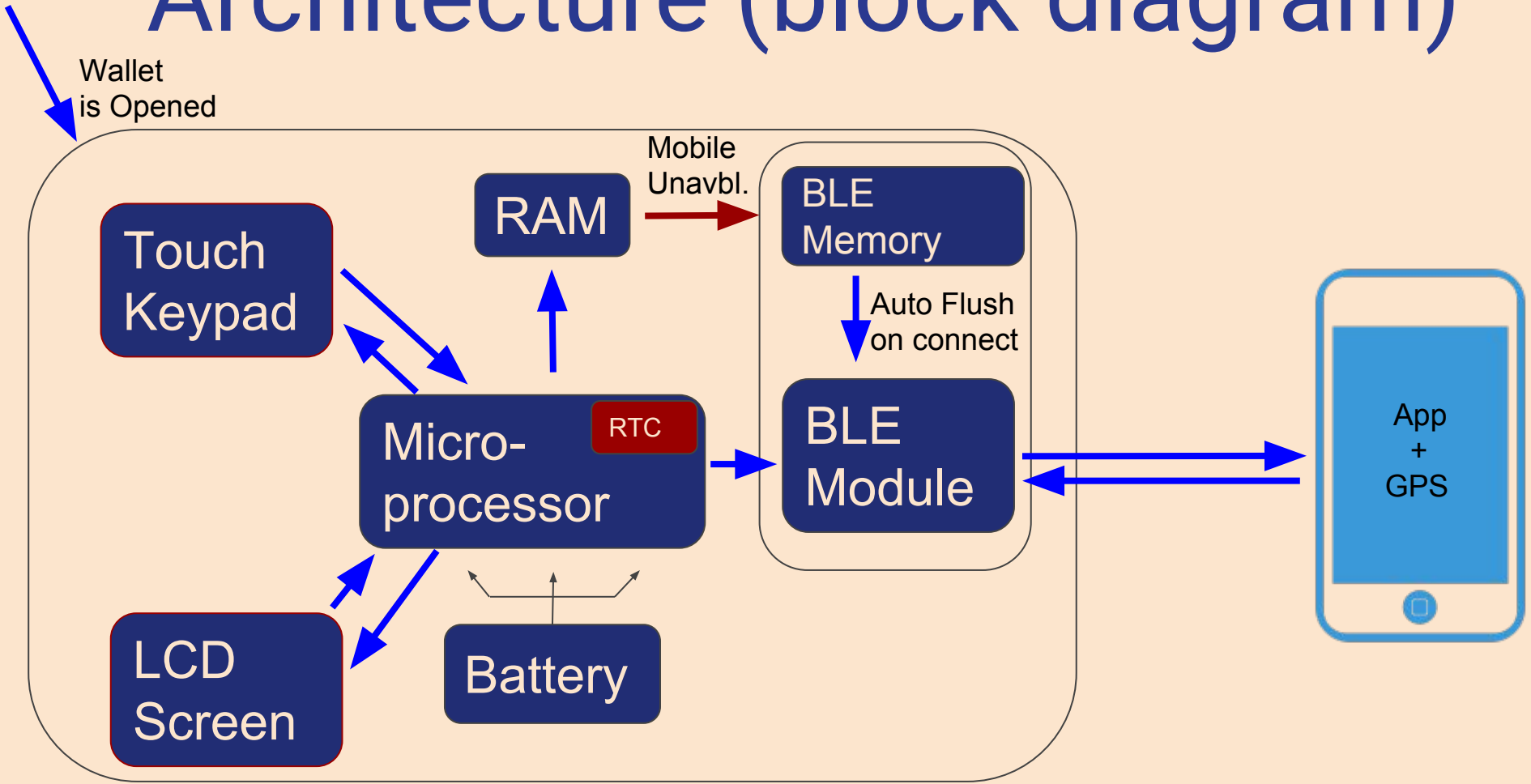
March 22: Demoed the first 3 points mentioned above

April 6: Demoed with last 3 points added

Critical Tasks

Accurately and reliably transmit amount and timestamp data for which rtc, keypad, lcd screen, bluetooth, the app and the communication between the app and our module was work perfectly

Architecture (block diagram)



States, signals and flow

- Initially, app and smart wallet are in idle state.
- Then, wallet gets a signal when any key is pressed on the keypad, and it goes to the state in which it performs arithmetic operation.
- When “send data” signal is received, it buffers the data in its memory, goes to “data buffered” state and checks whether connection is established with the app.
- If connection is established, data flows to app and wallet again goes to idle state. At this point, app goes from idle state to “parse data” state. In this state, app interprets the data, finds the location and display it. After this app again goes to the idle state.
- If at any time, connection is established, wallet sends the buffered data if it was in “buffered data” state and goes to idle state. And app processes this data as described in the last point.
- If at any time, connection between app and wallet is lost. App goes to “theft warning” state for some time and then again goes back to idle state.

Constraints

Size : Dimensions of a credit card with thickness margin of 4 cards

Power : Should last for a month

Cost : Should be affordable by middle class people

Time to market : 3 months

Risks/ Challenges

- Cards/ Online Wallets decreasing the usage of cash
- Increasing battery life
- Developing a design for women’s purses
- Creating a robust product to deal with the wear and tear of everyday use.

Innovation and Challenges

Innovation

- Entire smart wallet idea was innovative, not a startup idea picked from the west or elsewhere
- We didn't have enough keys, so used a combination of keys to represent money taken/ sent
- Used a real time clock so that correct timestamp can be logged even when not connected to mobile
- Buffering of data in hardware
- The android app code was written from scratch

Challenges

- We tried to reverse engineer a calculator but failed because voltage thresholds were low so made our own
- Matrix keypad required pull-down resistors to function properly so we had to integrate our own resistors
- Optimizing timeout for smooth user experience of keypad
- LCD screen required a pot(Figuring this and need of pull down resistor took lot of time)
- Two pins in the TIVA board are internally connected which is something we weren't aware of. We spent a lot of time debugging why the keypad and lcd worked individually but not together. We later solved it by using different pins.
- Had to configure the real time clock
- Had to learn android
- Made the app from scratch without using any existing code
- Interaction between the different background threads(bluetooth, UI, internet) led to frequent crashes. Debugging this was complicated.
- Learning Android Studio for the first time was a steep learning curve

Task Completed

Task: Making a calculator(keypad+screen+processor)
Problem: Tried reverse engineering an existing calculator, didn't work since voltage thresholds were low
Solution: Made our own using matrix keypad, lcd screen, TIVA processor

Task: Making the Matrix Keypad work
Problem: Wasn't working for a reason we discovered later, spurious characters and requirement of hard presses for typing was required
Solution: It required a pull down register which we integrated and then got it to work, timeout delays had to be very meticulously adjusted for a smooth user typing experience

Task: Getting LCD screen to work
Problem: Nothing was appearing on screen
Solution: Had to use a pot, changed resistance in order to improve contrast between screen and text color

Task: Getting screen and keypad to work together
Problem: They worked independently but didn't work together
Solution: Two pins that we were using were internally connected in TIVA, we changed to different pins

Task: Logging the correct timestamp
Problem: Initially we were using mobile time, so on no conn. with mobile incorrect time of transfer was logged
Solution: Used an RTC to log time in h.w module itself

Task Completed

Task: Getting bluetooth module to work, sending data(amount, timestamp, sent/ taken)
Problem: Wasn't working
Solution: Realized that the module was damaged, bought and used a new one

Task: Buffering data
Problem: Earlier data was transmitted only when the connection was established b/w mobile and module
Solution: Used some existing space within the bluetooth module to store data

Task: Displaying and editing data on mobile screen, buzzing & notifying phone on loss of wallet connection=>theft
Problem: App used to crash
Solution: Some UI issue that was eventually fixed, for bluetooth connection loss, we detect an interrupt

Task: Making data persistent
Problem: Earlier data would wipe out in closing app
Solution: Made a mini file system and stored data and meta data

Task: Showing map and gps data of where money was borrowed
Problem: App used to crash
Solution: Had troubles with background thread coordination, the way fragments and pop ups were handled etc.
Several rounds of debugging and trial and error solutions resulted in proper functioning

Tests

1. **Criteria:** Test that basic arithmetic operations can be performed on calculator
   Description: Test multiplication, addition, subtraction and division.
   Result: Passed

2. **Criteria:** Handling overflow in calculator
   Description: Do 1/0, 100000*100000
   Result: Passed. Output - “Overflow”

3. **Criteria:** Check that information is correctly transmitted to the app using bluetooth
   Description: Send data from smart wallet to app and see if it receives it
   Result: Passed

4. **Criteria:** Check that data is buffered in memory when bluetooth is not connected to app
   Description: Send data from smart wallet to app while it is not connected
   Result: Passed. On connecting bluetooth to app, data gets shown in the app.

Tests

5. **Criteria:** Test that keypad and lcd screen works
   Description: Press any key on keypad and it should show up on lcd
   Result: Passed

6. **Criteria:** Test that app shows the right location of transaction
   Description: Make any transaction and see if the location displayed is correct
   Result: Passed

7. **Criteria:** Check that theft detection works
   Description: Increase the distance between smart wallet and app
   Result: Passed. Mobile vibrates and app gives a notification.

8. **Criteria:** Check that data persists on the app even if the app is closed
   Description: Send some data to the app and then close the app
   Result: Passed. When app is reopened, data is still there.

9. **Criteria:** Check that edits made to the transaction details are reflected properly
   Description: Make some change to any trtransaction detail. Change the time or description.
   Result: Passed. After changes have been made, the new details are visible in the app.

Performance Metrics

1. Time taken to detect the key pressed on keypad and to show it on LCD screen.
   -- Of the order of ms. Smooth keypad experience.

2. Time taken to send the data from smart wallet to app.
   -- Of the order of ms. Unnoticable delay.

3. Time taken to detect an event of theft.
   -- Around 3-5 seconds.

Reusability

- The Android code is reusable and does not make use of any specific libraries, the code is free for use by anyone and adheres to standard android programming practices
- The code for hardware is specific to TIVA
- The logic that we have used for matrix keypad, bluetooth and lcd are all independent of each other. Apart from certain syscall they are easily extendable to any hardware.
- All code is functional and separated into different modules

Thanks !