

SmartBus Structure

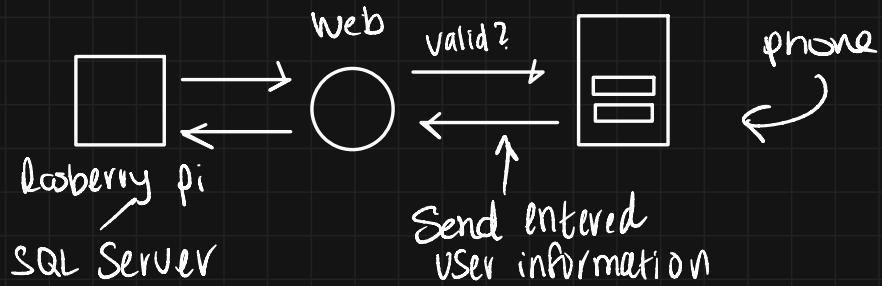
Application - Part 1

1) Login



> must be an authorized user

2) Authorized user



> checks in with database /SQL Server to see if user information is stored.

3) if (user exist)

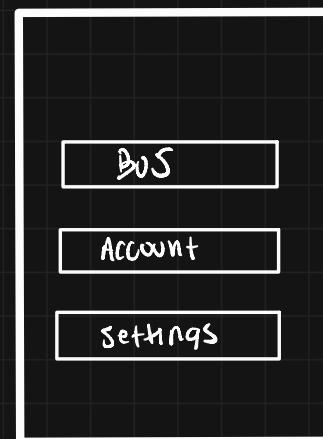
↳ access to Application

else if (user DNE)

↳ keep application locked

7 might look into 2 step authentication

4) Parent Interface



> interface - Parent

- Bus
 - ↳ user is only allowed to view bus that they are assigned to
- Account - parent
 - ↳ list view of children tied to there account

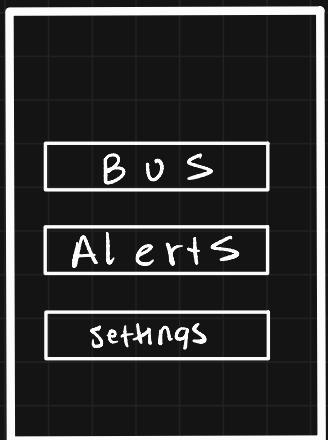
• Setting

- ↳ Logout
- ↳ View
 - Dark mode ?
 - size of font ?

↳ notification settings

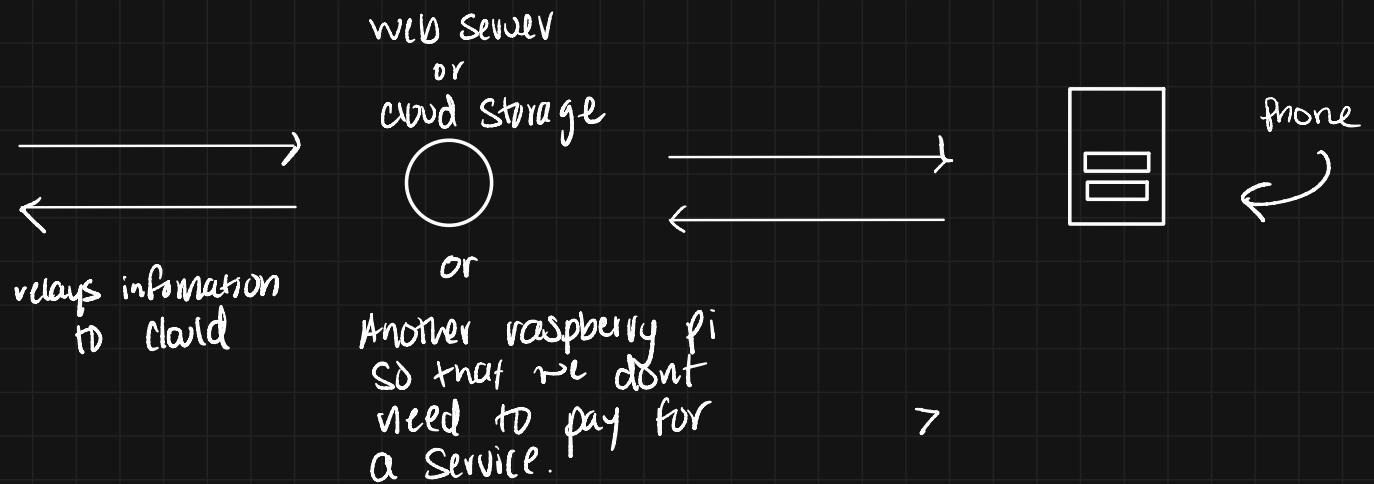
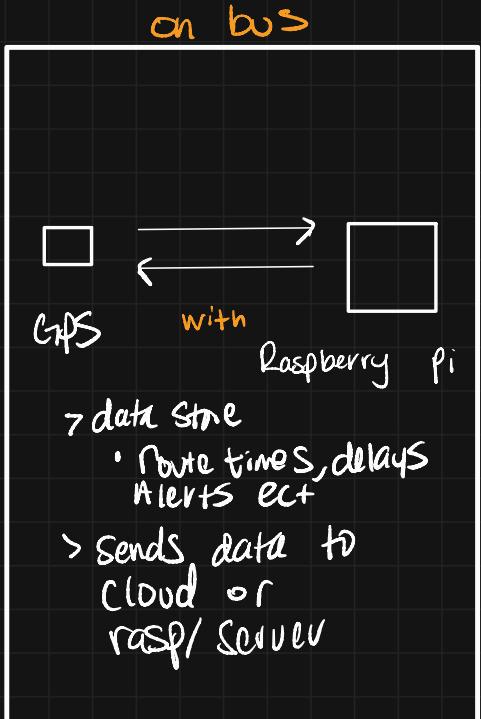
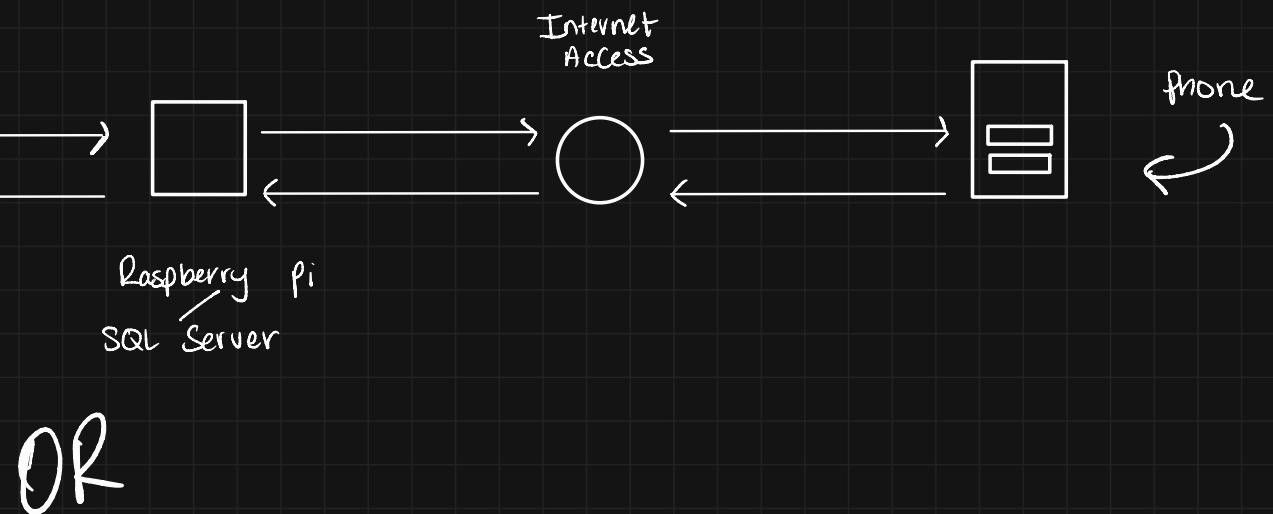
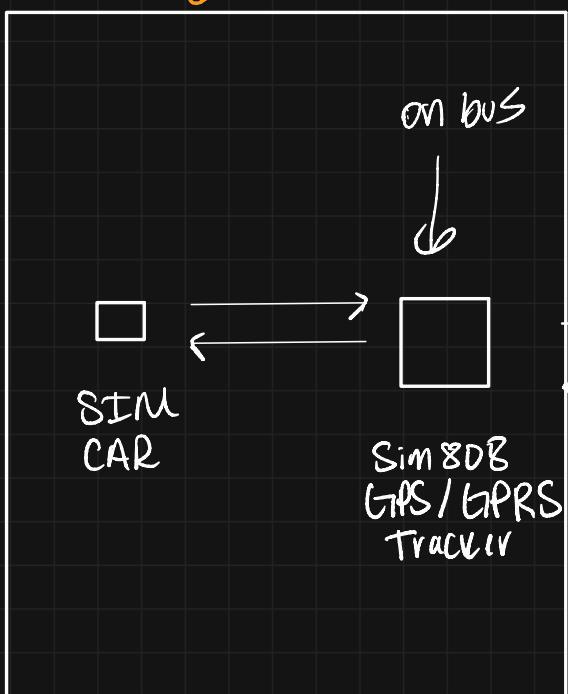
- bus Alerts
- weather Alerts
- maintenance Alerts
- delayed Alerts

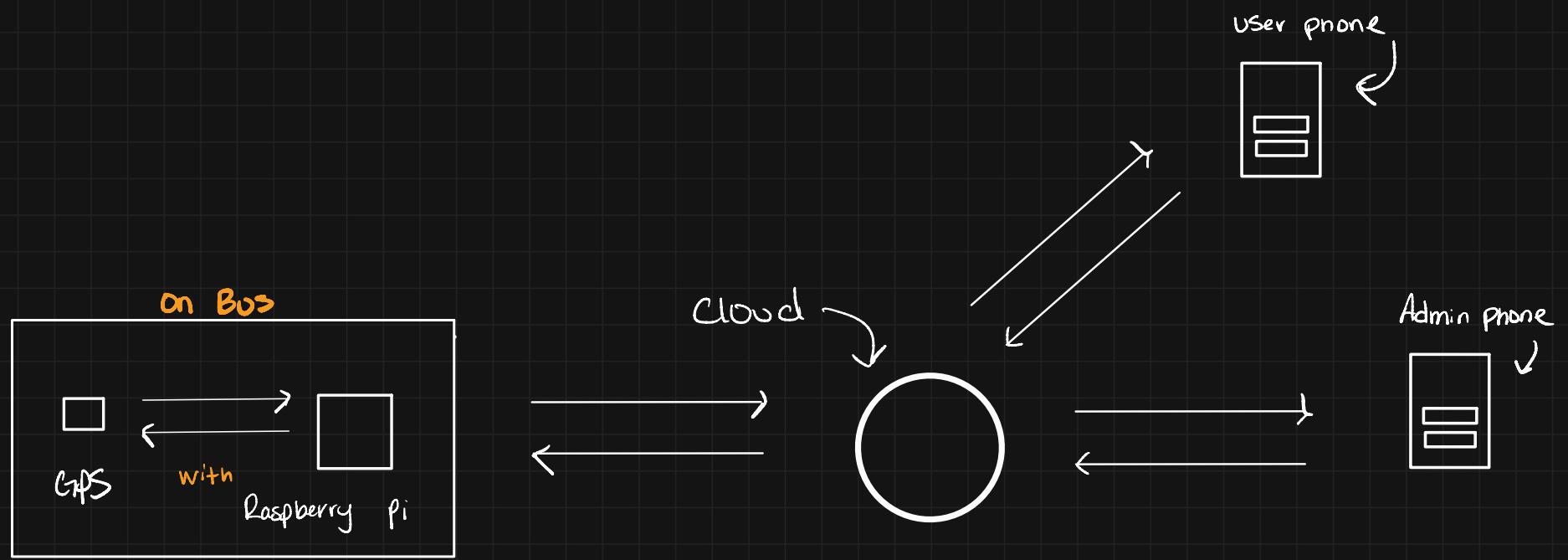
5) Admin interface



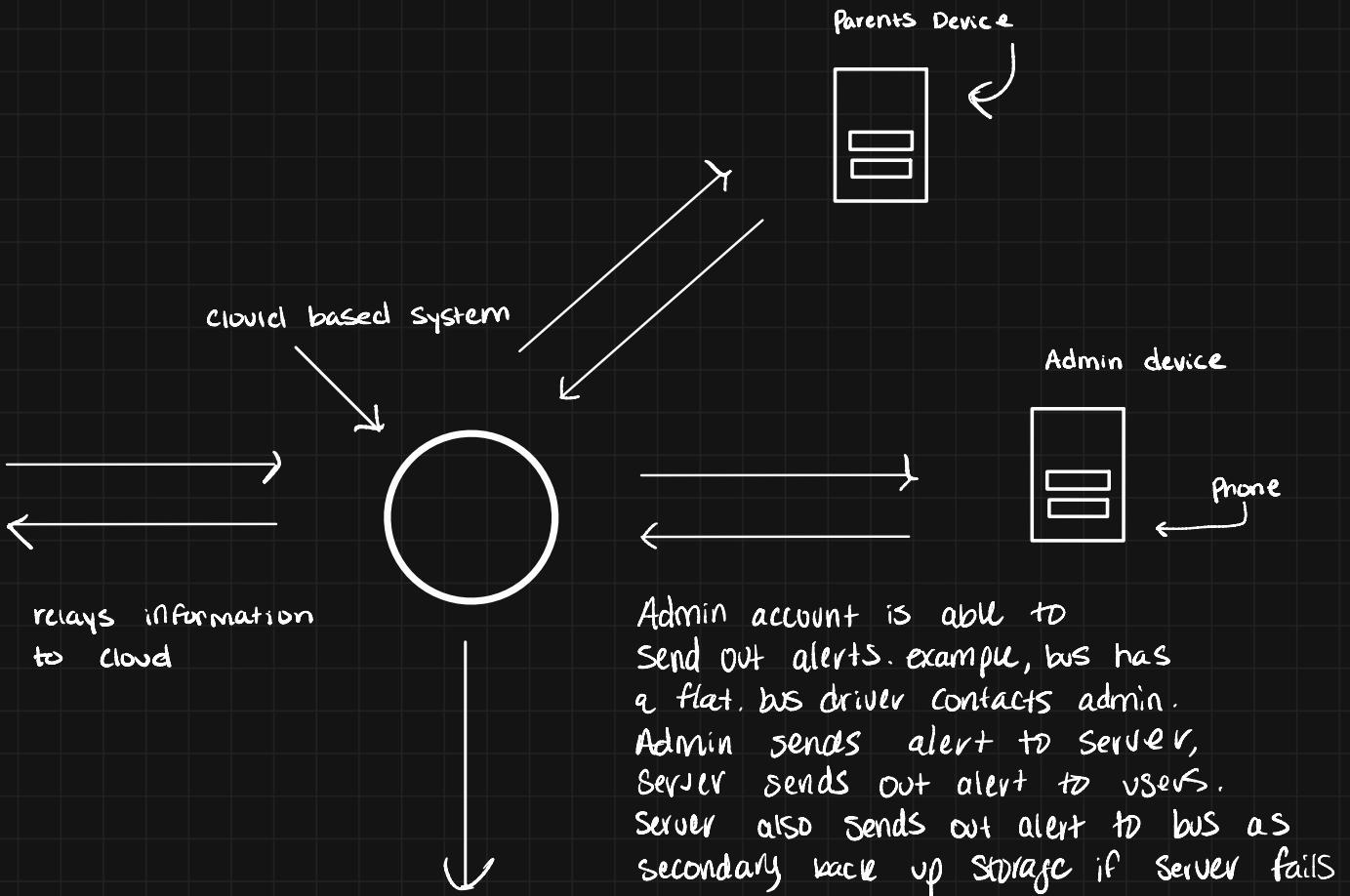
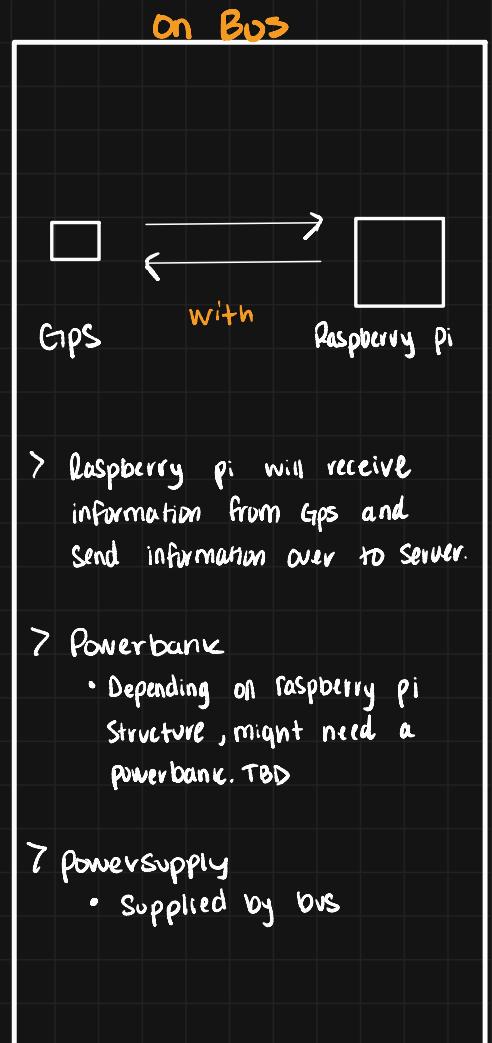
> Admin interface

- **BUS**
 - ↳ Active buses → Map / Listview
 - ↳ inactive buses → list view
- **Alerts**
 - ↳ Bus number and Alert / issue
- **Settings**
 - ↳ Logout
 - ↳ View
 - Dark mode ?
 - Size of font ?
 - ↳ notification Settings
 - bus Alerts
 - weather Alerts
 - maintenance Alerts
 - delayed Alerts





SmartBus Application Structure



* Materials subject to change depending on design

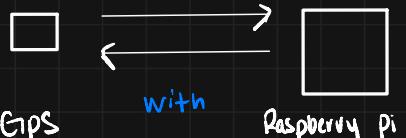
SmartBus Application Structure

Hector Vazquez
x875c9b3

* name of application Subject to Change

back end

on Bus



> Raspberry pi will receive information from Gps and send information over to server.

> Powerbank

- Depending on raspberry pi Structure, might need a powerbank. TBD

> PowerSupply

- Supplied by bus

cloud based system

code

Data Retrieval

External Data

code

Data Retrieval

front end

View code

UI

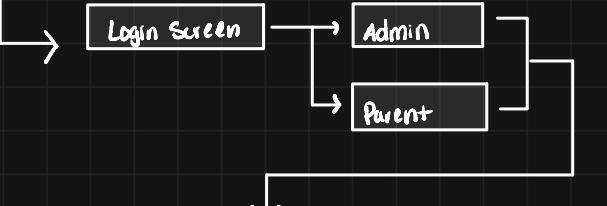
Data

Data Changes

event code

linked

user input



* UI Depends on User

Smart bus UI

