Boda Boda -
Motorcycle taxis in East Africa

AGENDA

During this presentation :

- Introduction
- Significant Changes so far
- Features implemented
- Important Design Decisions
- High-level design
  - Main Parts & their Collaboration
  - Third party frameworks
- Detailed Software Design
- GUI Structure
- Client Feedback
- Plans for rest of the implementation

INTRODUCTION : Client

- **Client : OKAPI finance**
  - Bank the unbanked by providing :
    - Financial inclusion
    - Increasing access to financial services
    - Minimizing transaction charges
  - Global money transfer company
    - Offering affordable cash transfer services

INTRODUCTION : Product

BodaBoda : Mobile phone application in East Africa

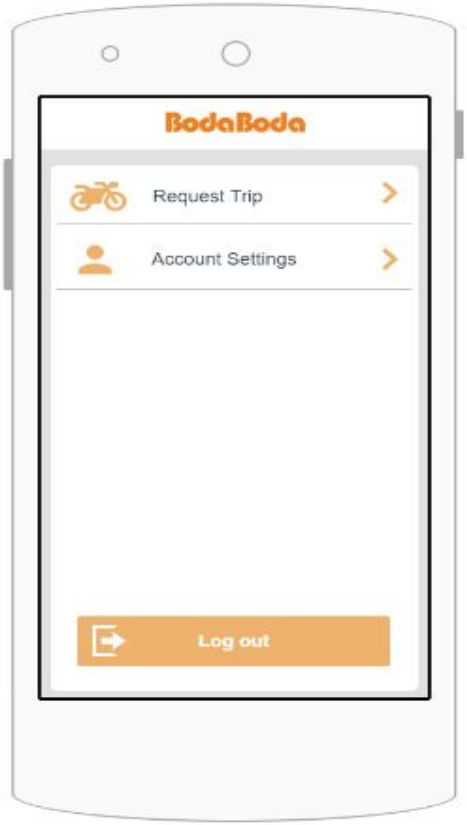
Goal : *Matching motorcycle taxi drivers and customers*

- **Customer :**
  - Customers could find the closest free driver
  - Find cheapest price
  - Know who is willing to drive them to their destination
- **Driver :**
  - Browse pending trip requests
  - Accept a request
- **App :**
  - Track the driven distance
  - Calculate a price
  - Handle the payment

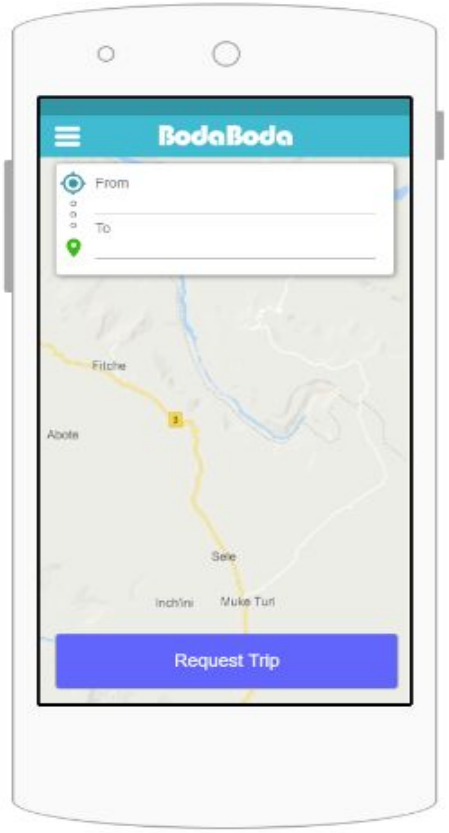
Significant Changes so far

- Changed the color scheme because the competitor is also using the same color.
- Customer main page is changed in a way that the customer can directly request a trip from there.
- Driver main page is changed in a way that the pending trip requests are shown directly, without extra navigation.
- Settings and secondary options are accessed from a sidebar menu.
- Features implemented

Customer Main Page

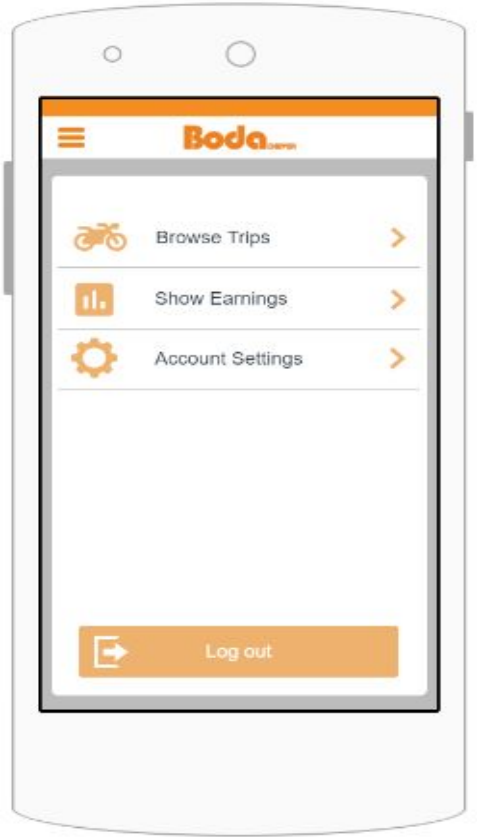


Before

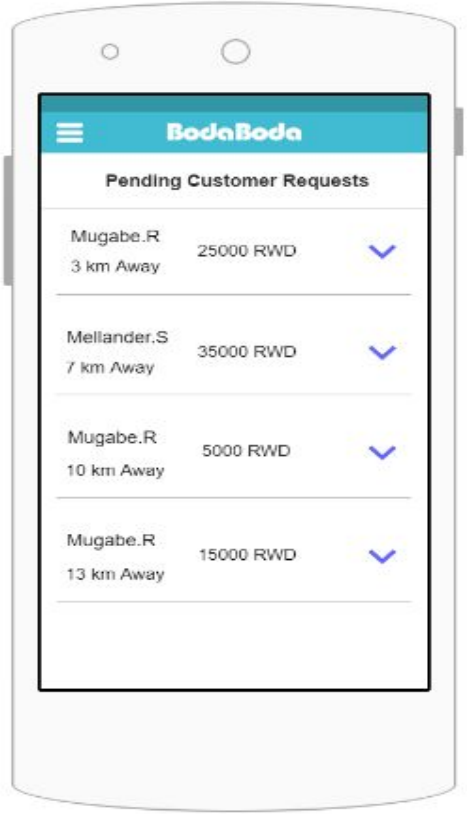


Now

Driver Main Page



Before



Now

Features Implemented

- Basic activity layout
- Auto-suggestion of location
- Server and database
- Established connection between front-end and back-end
- User-friendly design

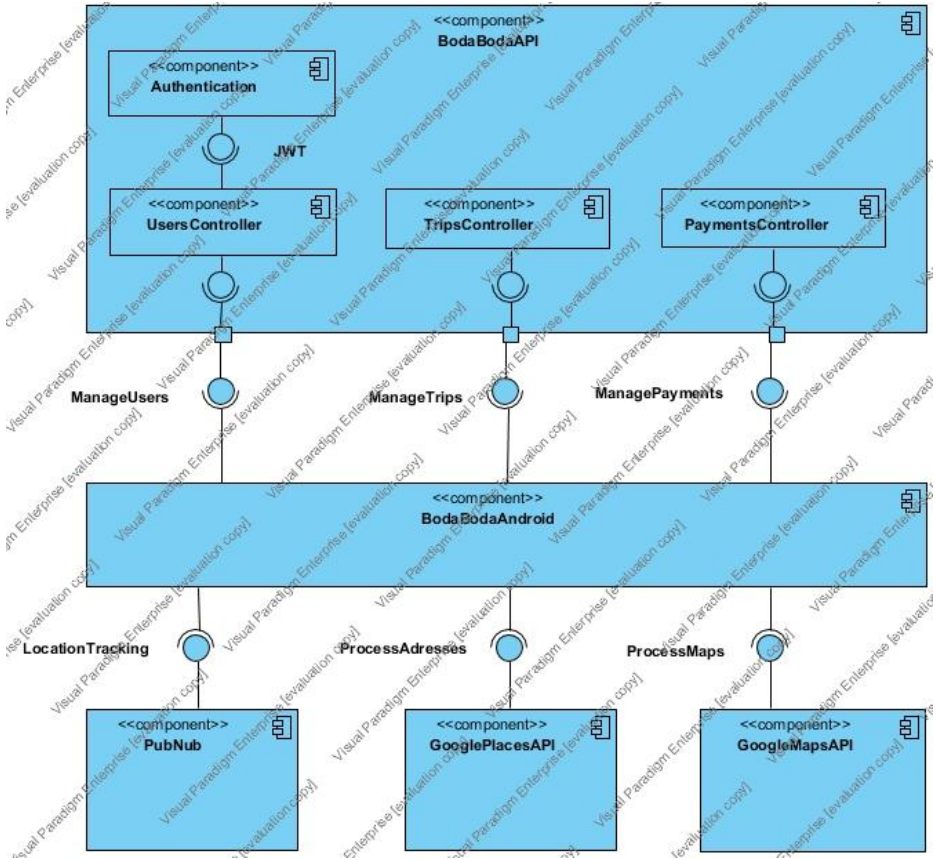
Important Design Decisions

- Creating the UI as simple as possible that can also be usable by the users with little experience with mobile applications
- Supporting the older android versions also so that users with older phones can use the application
- Sacrificed some of the aesthetics in exchange for usability
- Keeping as little computations as possible on the application and let the server do that work to save battery life
- Handling payments for users without bank accounts through Cash

High-Level Design

- Main Parts
  - Front-end android application
  - Back-end .NET Core server
  - mySQL Database

- 3rd Party Tools and Frameworks
  - Google Places API
  - Google Maps API
  - PubNub
  - Retrofit
  - Swagger



Detailed Software Design

- Front-end Design – Built as an Android application acting as an interface for the user
  - Uses phone's inbuilt GPS to track customer location
  - Google Places API to get the coordinates of the starting and destination locations of the trip
  - Geolocation information is sent to PubNub channel

- Back-end Design - Built as a .NET Core REST API providing storing, processing and authentication functionality to the system.
  - Models created
  - Services created
  - Controllers created
  - JSON Web Tokens used as an authentication standard
  - Entity Framework used as an object relation mapping tool
  - Swagger used for automation of API documentation

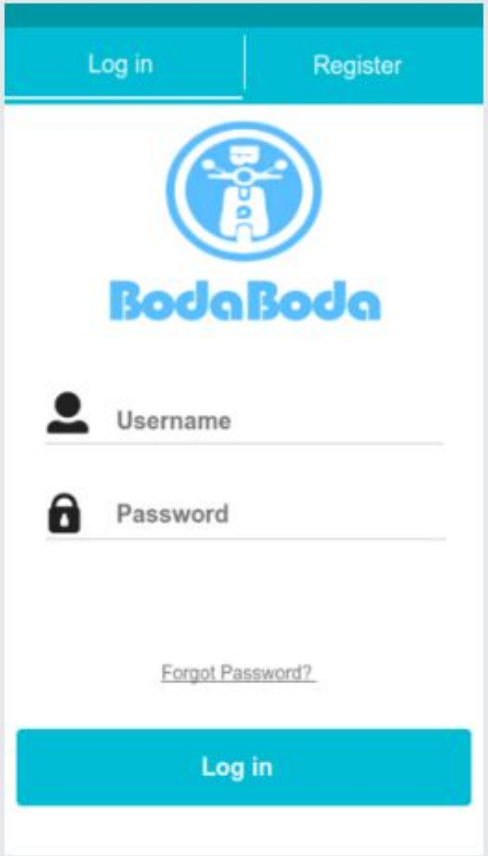
GUI Structure

- User Interface is structured in a way that favors usability over aesthetics

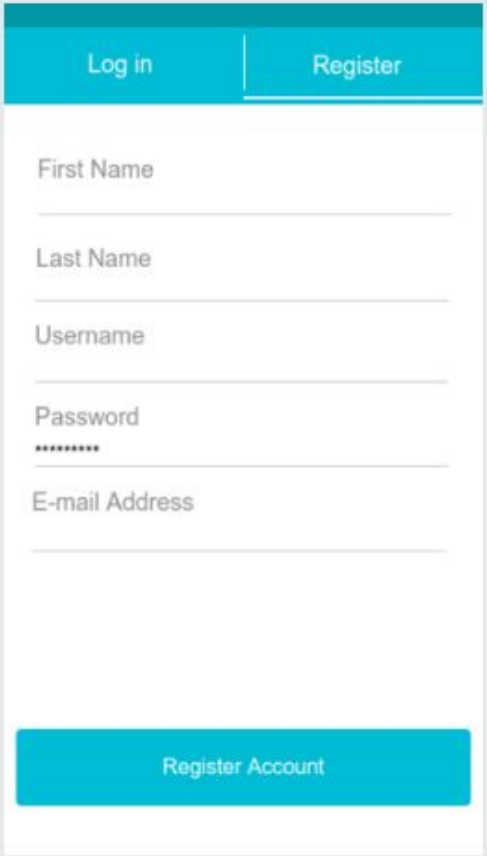
- Below are the interfaces that are part of the application
  - Guest Interface
  - Driver Interface
  - Customer Interface

Guest Interface

Loading Screen 2

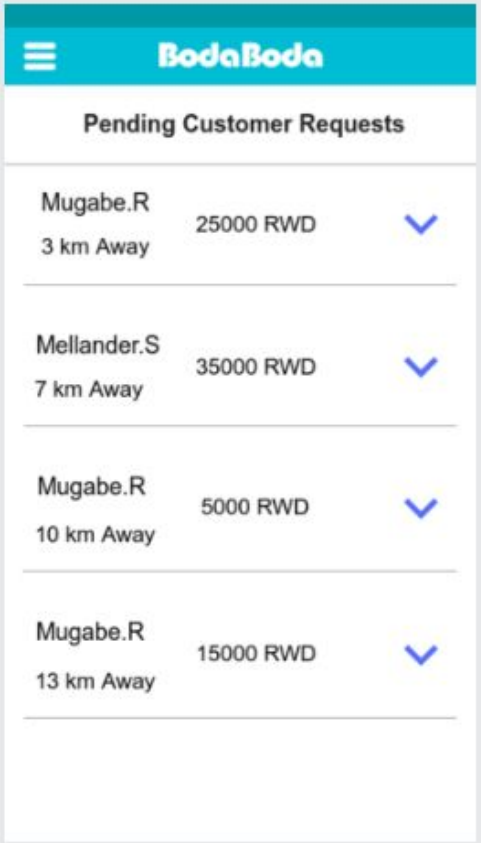


Login screen

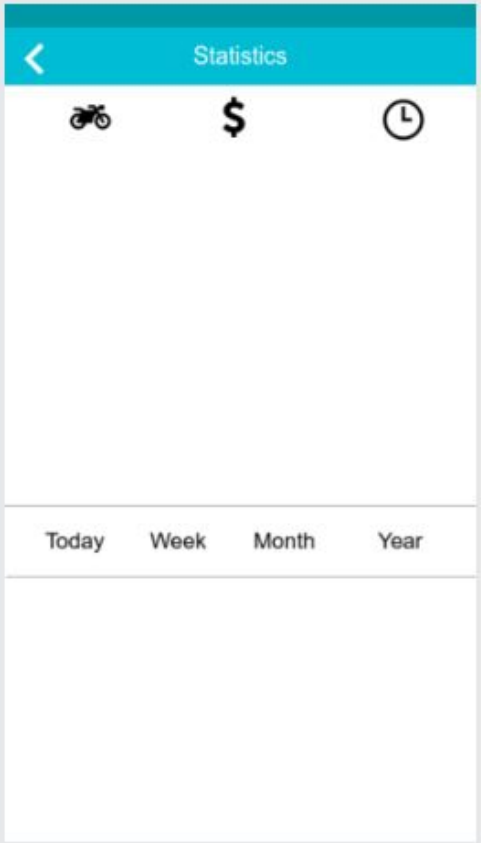


Register screen 2

Driver Interface - 1

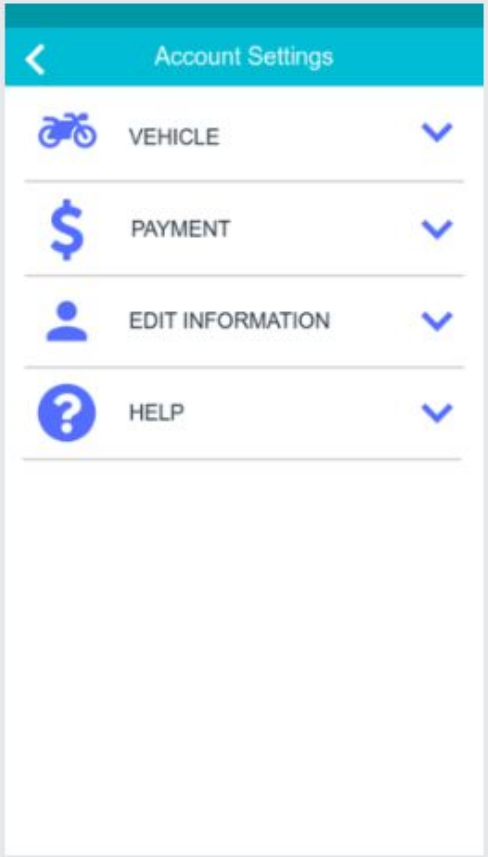


Driver Main2

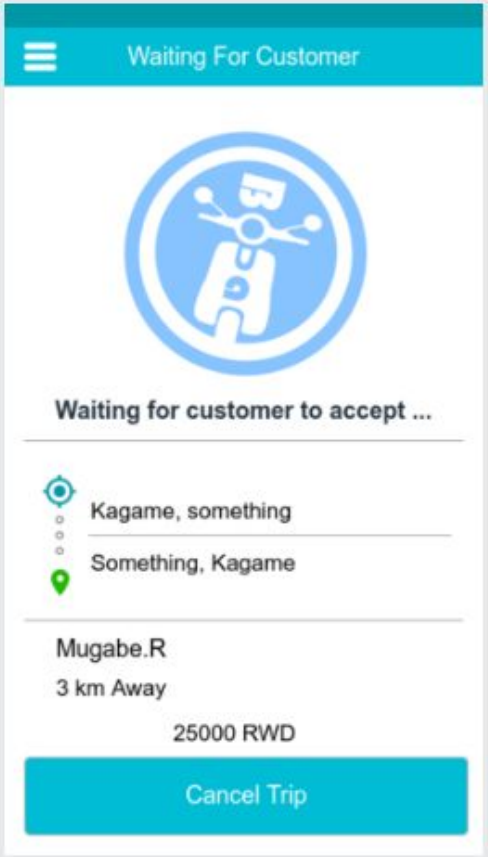


Driver: Show Earnings 2

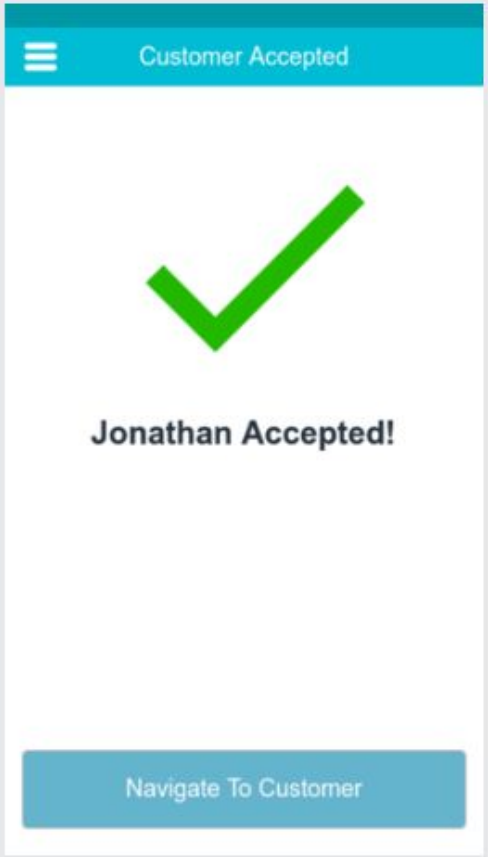
Driver Interface - 2



Driver: Account Settings 2



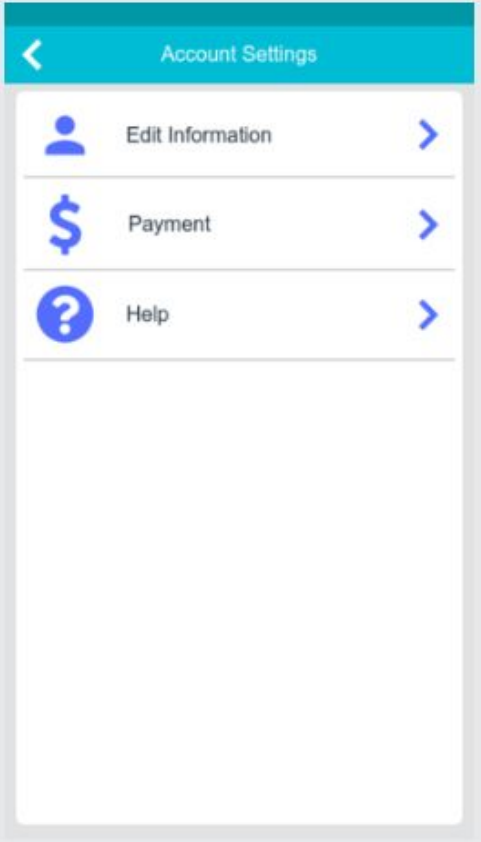
Driver: Trip 2nd step handshake 2



Driver: Trip 3d step handshake 2

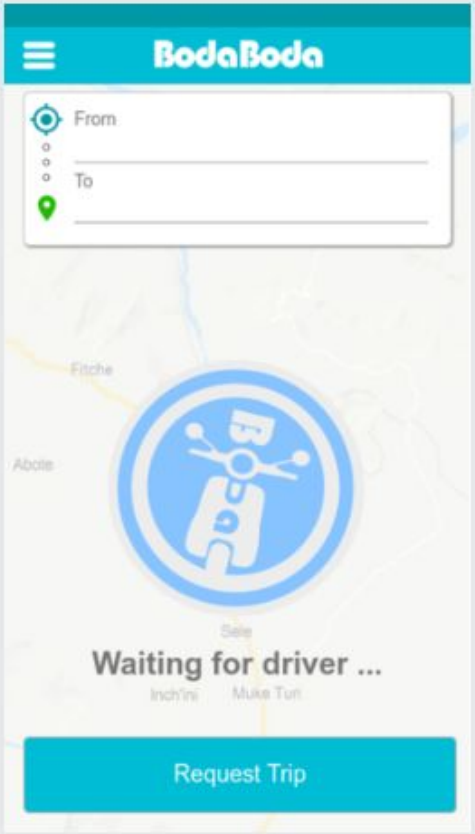
Customer Interface - 1

Customer:Request Trip

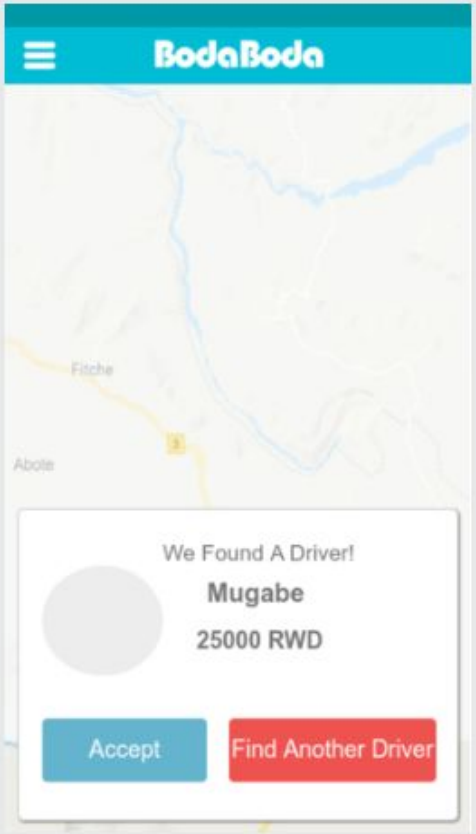


Customer: Account Settings

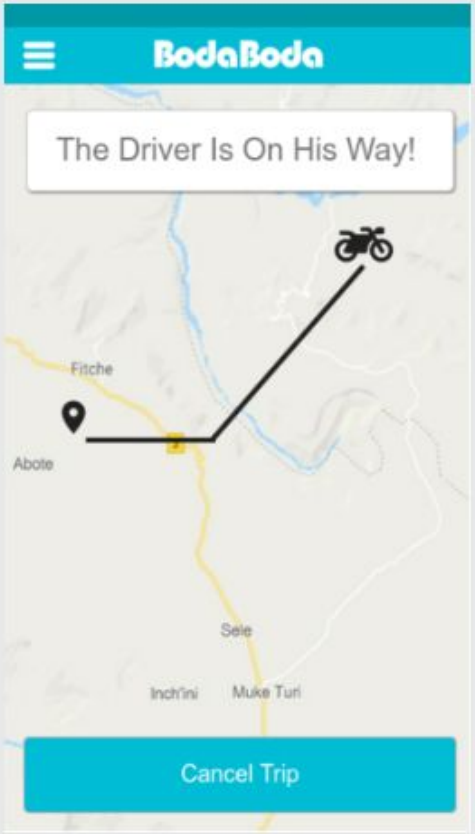
Customer Interface - 2



Customer: Trip 1st Step handshake



Customer: Trip 2st Step handshake 2



Customer: Trip 3rd Step Handshake

Client Feedback

- Suggested to visit the site material.io to follow the tips and guides for creating the user interface
- Price calculation - various methods discussion
  - Distance - time
  - Preemptive payment - GPS recalculation
- Statistics as a new requirement

Changes proposed by Client :

- Move the whole path from PubNub to the server after the trip is done
- Instead of "decline", the label should be renamed as "Find another driver“ in Customer screen

Plans for rest of Implementation

- Connecting Front-end with the Back-end
  - Accept/Find Another Driver
  - Browse Trips
  - Account Settings
  - Request Trip
- Navigate to Customer Location
  - Asynchronous communication between driver and customer
- Statistics
  - Create a new service for data mining and analytic purposes
  - RethinkDB


QUESTIONS ?