University of California, Santa Cruz

Project Phase 2 Report

Qian Cao, Dailon Dolojan, Zifan (Jason) Zhao, Tiantong Zhang, Jordan Hernandez

TIM 50

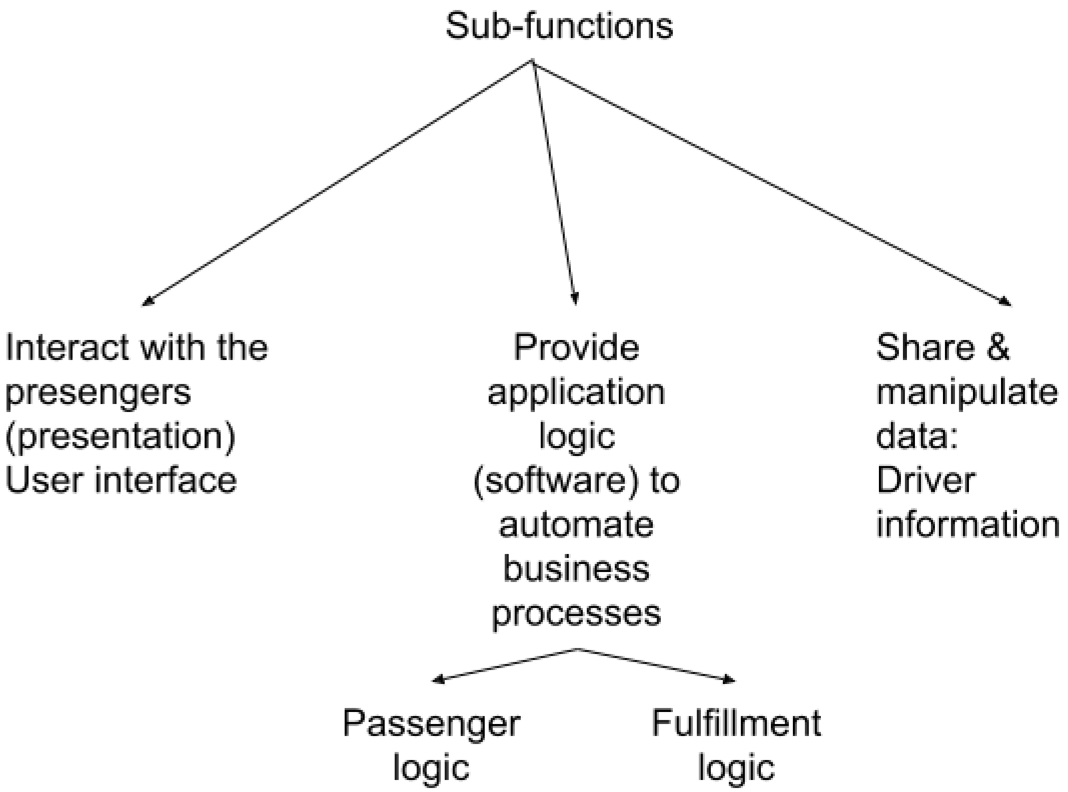
7/10/2017

Professor Subhas Desa

Task 1: Design the software architecture

- Step 1: Define the business or organization problem that needs to be solved
    - Instead of letting the application to complete the ordering process and randomly select driver we are offering the right to let our customers to pick the driver they want or select a particular type/gender of driver that they prefer or feeling comfortable with. The purpose of giving them them the right to choose their driver is beneficial for both side of driver and customers.

- Step 2: Define the necessary business process that need to be automated by the software application


Sub-functions
Interact with the presengers (presentation) User interface
Provide application logic (software) to automate business processes
Share & manipulate data: Driver information
Passenger logic
Fulfillment logic

- Step 3: Translate the business processes into the requirement for the softwares:
    - Log into uber server via app.
    - Present the homepage to the user
    - Present the map on screen where user can set pin pick up point and type the destination.
    - Provide the different car types for user to select

  - Present the payment method for user to verify.
  - Present the cost of the trip to the user, and also the time it takes to the destination
  - Provide the options for user to choose a particular type/gender of driver
  - Present the list of drivers who meets the requirement and are available to pick the order at that time
  - Provide the options for user to select a driver.
  - Present the completed order details includes the time the drivers takes to arrive the user's location.
  - Present the map while the ride is going on
  - Provide the rating method for user to rate for the driver
  - Present the ride completed page

- <u>Step 4: Design the software architecture</u>
  - Layer 1: Client layer
    - Enable the Graphic User Interface (GUI) for the client (user or passenger) on the client's smartphone
  - Layer 2: Ride logistics presentation layer
    - Create the GUI for the client
    - Present the necessary information
      - Map which use the user's location as the center
      - Destination blank
      - Payment selection
      - Car type selection
      - Drivers' profiles
      - Confirm ride order button

- Layer 3: Rider fulfillment application layer
  - Perform the application logic
    - Passenger transaction layer
      - Enable passengers to interact with the driver profiles
      - Include ride order entry
      - Payment
    - Fulfillment layer
      - Interaction with banks
      - Interaction with drivers
- Layer 4: Database management server layer
  - Manipulate and share data
    - Rider: the customer/passenger's information, such as Such as rider's name, address, payment method, phone number, email address, and the driver who served him.
    - Driver: includes information about the drivers. Such as driver's name, address, salary(25% of the cost of the trip + salary counted by working-hours), phone number, email address, car he used for serving the ride.
    - Trip: includes information about the trip/ride/order. Such as rider's information, driver's information, trip's destination, cost, and status(completed or cancelled).
    - Travel: includes up-to-date traffic, map, and GPS.
    - Payment: includes the information about transaction, such as the rider's payment, and the driver's commission.

Software architecture diagram

Task 2: Design the hardware architecture

- Step 5: Design the hardware architecture
    - Layer 1:
        - Enable data customers, banks, and orders to be processed from the Graphic User Interface (GUI) to the web server.
    - Layer 2:
        - Host necessary information (user’s location, drivers’ profiles, and so on)
        - Facilitate the visualization, data entry, communication with the application and output of results
    - Layer 3
        - Host the application logic
            - Passenger logic
                - Passenger information
                - Interacting with the driver profiles
                - (Including ride order entry, payment)
            - Fulfillment logic
                - Interaction with banks
                - Interaction with drivers
    - Layer 4
        - Manipulate and share data
            - Passengers
            - Drivers
            - Orders

Hardware Architecture Diagram

Task 3: The preliminary database and preliminary network architectures.

We follow micro-services architecture catering to mobile and web apps, here goes the preliminary Database and preliminary network architectures.

Preliminary Database:

- Software View:

- Hardware View:

Preliminary network architectures:

1. Data link layer(n-to-n).
2. Network layer(h-to-h).
3. Transport layer

4. Application layer

Task 4: List of Available Implementation Options

• **Geolocation & Mapping Software**:

a.) List of Software

Below is a list of commercially available **GIS software** that our company has considered implementing.

1. ArcGIS (Esri)
2. Geomedia (Hexagon Geospatial)
3. Global Mapper (Blue Marble)
4. SuperGIS (Supergeo Technologies)

b.) Software-Specific Selection Process (selection process specific to the type of software)

Currently, our application primarily uses iOS' CoreLocation framework and Google Maps because these softwares are already implemented in the mobile devices our company targets. This makes use and navigation of the software much simpler than if we were to use outside companies' softwares. However, our company may seek to become less dependent on Apple and Android's softwares, so having these listed softwares as alternatives may prove beneficial to our future business decisions.

• **Push Notification & SMS Software**:

a.) List of Software

Below is a list of commercially available **SMS software** that our company has considered implementing.

1. Textedly
2. Mobiniti
3. Avochato
4. MessageMedia
5. SimpleTexting

b.) Software-Specific Selection Process

Currently, our application uses Twilio as its primary SMS software. As for Push Notification software, our company uses Apple Push Notification Services and Google Cloud Messaging. These two softwares continue to be our best options because they are already implemented within their respective mobile devices (Apple, Android) and remain the most compatible and user-friendly with these mobile devices and our application. However, our company may look to implement additional *or* alternative softwares should we find a software that better suits our business needs.

- **Payment Integration**:

a.) List of Software

Below is a list of commercially available **Payment Processing software** that our company has considered implementing.

1. Zoho Checkout
2. PDCflow
3. Recurly Subscription Billing
4. EBizCharge

b.) Software Selection Process

Currently, our application uses Braintree, one of the leaders in the mobile payment industry, as its primary payment software. Our company also uses Paypal's Card.io service for credit card scanning on iOS. PayPal has proven to be a useful asset to our company because of its integration with our application, and its reputation in the payment industry. However, our company may look to implement alternative *or* additional softwares to our application should we find a software that better suits our business needs.

- **Software Pertaining to our Current Business Problem**

Our current business problem deals with offering customers the option to select a driver based on their preferences. We are looking for a software that allows our application to display a driver's "profile" or information to our users.

Overall Software Selection Process

In looking to switch from the softwares we currently use, we are looking for software that can help lower spending without being detrimental to our current business processes. Softwares that are highly reputable within their respective industries would be our safest options, but any software that can offer features we can implement to improve our application would also be valuable to our company. Above all else, softwares that can fulfill our current (and future) business needs, or can help expand the functionality and utility of our application will rank highly in our software selection process.

https://yalantis.com/blog/uber-underlying-technologies-and-how-it-actually-works/