SLUber

Blake Braun, Ryan Murphy, Eric Ramos

Capstone 1

3 Main Pieces -

1. Client Application
2. Server
3. Dispatcher Application

**Client Application -**

* User should be able to open mobile website (eventually part of the SLU app) and be greeted with information about SLU Ride’s operating hours, the number of rides currently in the queue, and a button to request a ride
  + If the current time is not within SLU Ride’s operating hours, a message should be shown indicating this, and the user should be unable to request a ride.
* When the user goes to request a ride, they will be asked for standard information to be sent to a SLU Ride dispatcher
  + Name - text field
  + Banner ID Number - text field - limit to 9 numbers
  + Phone Number - text field - limit to 10 numbers
  + Starting & Ending Locations
    - Initially a dropdown list of preset locations. In the future, utilize GPS to find the user
  + Number of passengers
  + Additional notes to be sent to the dispatcher
* Ideally, the app should have the capability to remember some of this information once it has been entered once (such as name, banner ID, and phone number)
* The user should be able to edit their ride request once it has been placed (such as if they need to change their pickup location,) and should also be able to cancel it.
* When the ride is being sent to the user’s pickup location, the user should receive a notification to their phone indicating this.
* Future development could include:
  + Showing a map of where the driver is located when en route
  + Giving an estimated time of arrival
  + Allowing the user to call/message their specific SLU Ride driver
  + Allowing the user to leave feedback about their ride
* Probable Technologies
  + React.js / AngularJS

**Server**

* Database server
  + Store current rides & all past rides
* Application server
* Web server
  + Host both mobile site and dispatcher site
* Future development could include a routing algorithm based on origin and destination locations of ride requests
* Probable Technologies
  + MongoDB - database for rides
  + Spring - Serve webpages, REST Endpoint for Mongo

**Dispatcher application**

* When a ride request comes in, it should be situated at the bottom of the current queue, showing all information about the ride
  + Name
  + Banner ID number
  + Phone number
  + Starting & ending locations
  + Number of passengers
  + Additional notes to be sent to the dispatcher
  + Time request received
* The dispatcher should be able to edit ride requests with the following:
  + Unit dispatched (driver number)
  + Time the ride arrived
  + Time the ride is cleared (ended)
  + Response time (can be calculated automatically)
* In addition, the dispatcher should also be able to:
  + Send a message to the user, such as a notification when the ride is on the way
    - This latter notification should be a simple button that sends a predefined message
  + Clear rides out of the queue when they are finished
  + Generate reports of rides for a given day, week, month, year, or all time
  + Manually add a ride request from someone who calls the SLU Ride phone number
    - They should need to enter the same information as is on the mobile site
  + Rearrange rides in terms of priority or location
* The site should also:
  + Have a section for rides currently in the queue, and a section for all rides that can be filtered by the day, week, month, year, or all time
  + Automatically move rides to the ‘all rides’ section once they are complete
  + Keep track of how many rides are in the queue and report this to the mobile site
* Future development could include showing the dispatcher where all vans are on a map, automatically routing vans based on location, allowing the dispatcher to keep a list of who is currently driving and assign drivers to rides