PROJECT DESIGN SPECS

Project Description

Our lives as students at Dartmouth depend immensely on our access to information. Whether we are checking the dining menu for the day or events occurring at the Hop, we spend a significant amount of time looking up information. Our project, Timely, seeks to eliminate this access time. Timely predicts the information you might need and delivers it exactly when you need it. The apps predictive abilities are empowered by both continuous sensing data and archived biorhythmic data collected from the user's smartphone.

The continuous sensing data enables predictions based on real time information of the user collected from the phone sensors. For instance, information collected may include the user's current location from location sensor(GPS or Network) data or their current physical activity based on accelerometer data. From the collected location data, Timely can then find and inform the user of events occurring around the current location. In special locations, Timely will provide particular useful information such as the day's menu when in the dining hall. And from the collected accelerometer data, Timely can deliver stats like the current speed, altitude and amount of calories burned during a physical activity (running, cycling, hiking, etc).

We'll take advantage of the archived biorhythmic data to initialize several components of the scheduling functionality. The app auto-schedules sleep and meals, and we can use previous data to schedule these based on patterns in habitual behavior in context the context of the changing day to day schedule. For example, if a user tends to eat at Collis for 30 minutes at 1pm daily, the app will automatically schedule that lunch. If the user's mood has been stressed for an extended period of time, the app can parse the campus events list and for events that advertise de-stressing. If the user's mood has changed, the app will suggest similar events for the specific user in the future. Or if the user has noted that they are distracted in the PACO app and the microphone detects loud noise during study time, Timely will suggest to auto-book a study room. Thus, the app continuously calibrates to meet the needs of the user.

Top-level System Components:

The primary component is the continuous sensing application that processes data from smartphone sensors. As with Google Now, the platform for displaying information is decoupled from the background functionality, which is essential for proper decomposition. This decoupling will make it easy to plug and play different activities and functionalities into the app without changing the platform user interface.

The other component is an external API coded in Node.js, Coffeescript, interfacing with a Postgres database. The nature of this application necessitates that server response times be quick and efficient because we need to get the information to the user at the exact time they need it. Thus, we chose the Node.js architecture because of the multiple requests coming from the application to the server. Node.js allows the server to handle multiple IO via asynchronous callbacks, so response times are faster. We also cache the pages every two hours so server JSON responses will be returned under 0.2 seconds. Our servers are currently set up for zero-time deployment and can be deployed with Heroku. We see possibilities in extending the Postgres database into a spatial one, i.e. PostGIS. So far, the following functionalities have been implemented:

- Bringing up live menu of FoCo menu: we make POST requests to the FoCo nutrition menu and parse the JSON returned. We can currently retrieve Breakfast, Lunch and Dinner menus when the user passes by FoCo -- detected with the accelerometer and GPS.

  http://timely-api.herokuapp.com/menus

- Mapping real-time events onto the Google Map: we have scraped the RSS feed from campus list-servs. We could extend this functionality to any type of list-serv, not just campus events. http://timely-api.herokuapp.com/events.
- Reverse-geocoded building name: Given a latitude / longitude, the API will return a building name. This is essential for displaying user-friendly names instead of hard-to-understand floating point numbers.
- Finally, if data on the API changes, Google Cloud Messaging will push the notifications onto the phone so the phone does not need to continuously poll the API. This is important because we want to mitigate polling the server too often so as to maximize battery life.

Description of Classes

On the following page is a source tree of our current src/ directory. At a high level, the MainActivity holds the Google Map fragment which the user interacts with. Click listeners are set for the map and for each of the Google Now-esque cards that will display information. Practically every API request will need to be an asynchronous task that runs on its own thread. Thus, Google Calendar requests and API requests each have their own classes.

Source Tree:

```
├── src
│   └── dartmouth
│       └── timely
│       ├── AsyncEventsInsert.java
```

- Uses the Google Calendar API to insert an event that has been scraped by the campus list-serv of events and mapped onto Google Maps.

```
│       ├── AsyncEventsPost.java
```

- POST request that accesses the events API to retrieve a listing of events from the campus list-serv via a RSS feed.

```
│       ├── AsyncLoadEvent.java
```

- Uses the Google Calendar API to load your events from Google Calendar

│ ├── AsyncMenuPost.java

- GET requests that accesses the nutrition menu from FoCo, boloco and the Hop.

│ ├── Globals.java

- Globals file that describes the global variables used int he application.

│ ├── MainActivity.java

  - Holds Map fragment, Google OAuth and API accesses.

│ ├── NotificationReceiverActivity.java

- Activity for the Notification and sets up the Content View

│ ├── NowLayout.java

  - Layout for Google Now-esque card that shows the user what they need before they need it. Used in the XML.

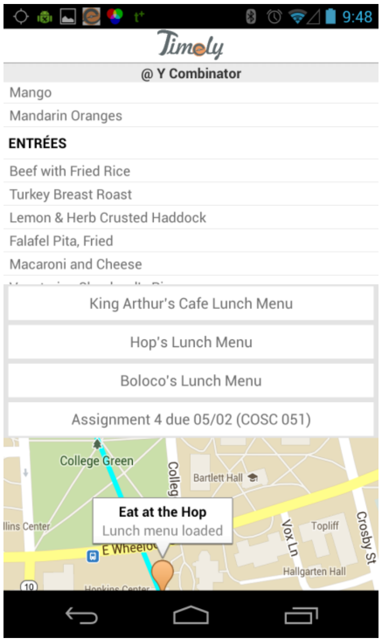
│ ├── calendarOnClickListener.java

  - Listener/handler for Calendar menu item

│ └── lunchOnclickListener.java

  - Listener/handler for Lunch menu item

UI Design


9:48

Timely

@ Y Combinator

Mango

Mandarin Oranges

ENTRÉES

Beef with Fried Rice

Turkey Breast Roast

Lemon & Herb Crusted Haddock

Falafel Pita, Fried

Macaroni and Cheese

King Arthur's Cafe Lunch Menu

Hop's Lunch Menu

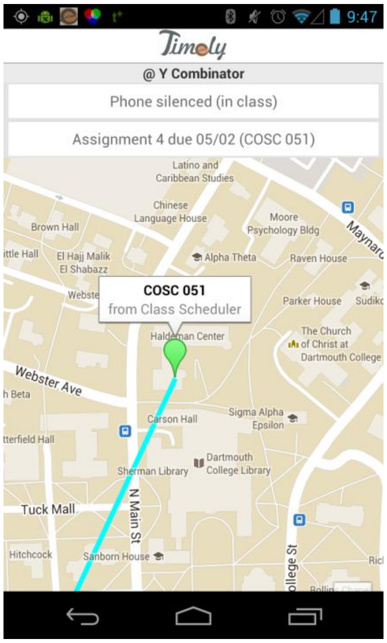
Boloco's Lunch Menu

Assignment 4 due 05/02 (COSC 051)

Eat at the Hop

Lunch menu loaded

Figure 1.


9:47

Timely

@ Y Combinator

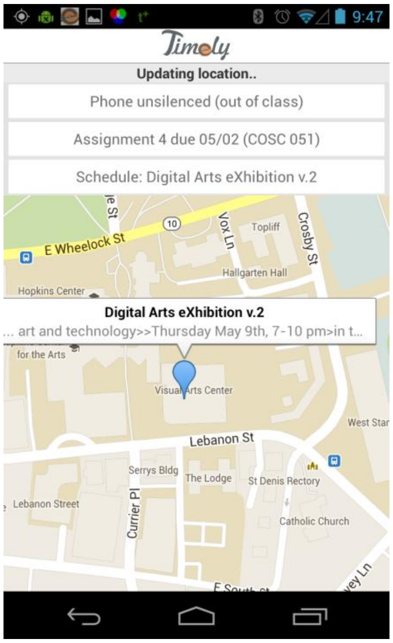
Phone silenced (in class)

Assignment 4 due 05/02 (COSC 051)

COSC 051

from Class Scheduler

Figure 2.


9:47

Timely

Updating location..

Phone unsilenced (out of class)

Assignment 4 due 05/02 (COSC 051)

Schedule: Digital Arts eXhibition v.2

Digital Arts eXhibition v.2

... art and technology>>Thursday May 9th, 7-10 pm>in t...

Figure 3.

The UI screendumps in the figures above represent the UI design we are pursuing for the Timely application. The UI focuses on the Google Map to display the user's current location as well as locations of interest around campus. There are markers as well as text to clearly display user information. The top portion of the screen is dedicated to displaying activities, events, menus, and other information of interest to the user based on the location and past preferences of the user.

Figure 1 shows the dropdown menus for the dining halls on campus. The menu for the 1953 Commons Dining Hall is currently being explored in the first figure.

Figure 2 displays a feature of the Timely application – Phone Silencing. Timely detects the user entering one of his or her class and automatically silences the phone. The notification of the phone being silenced is displayed. There is a notification of an assignment due for the same class that is being displayed.

Figure 3 continues on displays the un-silencing of the phone when the user’s class is over. In addition to the assignment notification, Timely also displays events happening on campus, such as the Digital Arts eXhibition, all of which is shown in Figure 3.