

The context of our app includes five important features: Reservations, Contact, Activity Data, Emergency Notifications, and the GPS/Mapping system.

The Reservation system of the app is the most vital and the main purpose of the app. This will collect user placed or cancelled reservations and update the reservation database with the information that will be vital for the actual usage of the other features. This system will also keep track of the daily scheduled shuttle runs that users can pick to reserve from.

The Contact System allows for the reserved riders and the driver to contact each other in the case of delays or any other emergency events that occur.

The Activity Data system is for administrators in order to monitor the actual number of active users for the app as well as demand for the shuttle based on time and day. This data can help the administrators adjust the shuttle schedule.

The Emergency Notifications system will allow for Pace University and the Transportation Department send banner notices when the app is open regarding any changes to the schedule due to any events/holidays/weather difficulties.

The GPS/Mapping system allows for users to track the location of the shuttle if any delays due to traffic or other events cause the shuttle to arrive late. There can also be coordination between students and professors. For example, professors can check the app to see if a student is late to their class because their shuttle is stuck in traffic.

**Activity Model of Placing a Shuttle Reservation**

Check bus schedule

Verify available arrival/ departure times

Check bus attendance

Counts available seats left

Adds user to attendance

Add new reservation

Some behavior this app would follow for a user’s request would be to add a reservation in the app’s reservation system.

The reservation system would first see all of the shuttle’s available times for the day so that the assigned time the reservation is for is correct. If the shuttle already departed before the reservation was processed the reservation should get rejected.

Next the shuttle’s attendance is checked to make sure if the shuttle can fit the new reservation, if it is full then the reservation should be rejected.

After that the reservation is officially placed by adding the user into the shuttle’s attendance with their first name at the bottom of the list for others using the app to see.

Interaction Model:

A picture containing box and whisker chart

Description automatically generated

This interaction model for user registration is a sequence diagram. The user enters account information – name, email, password, cell phone number (optional) in the app’s user interface. The application checks the entered email against valid pace emails. If invalid, the app alerts the user. If valid, the app sends an email to the user with a verification code, notifies user of email, and prompts entry of code. If user enters valid code, account creation is successful. User information is stored in the database and the user has access to application functionalities.

Structural model:

Account

**Name**

**Email**

**Password**

Id#

Phone number

Privilege level

Pace verification status

Email verification status

CreateAccount()

AdjustpLevelAccount()

DeactivateAccount()

VerifyAccount()

The Account class manages the accounts of the users. It consists of name, email, password, account id number, phone number and privilege level. There are also values to track the process of verification; pace verification status noting if the email is valid and email verification status validating the coded users entered. The account class also has the createAccount function to instantiate an account, an AdjustpLevelAccount to promote or demote other accounts if own privilege level is high enough. VerifyAccount runs before/or will be called by createAccount and ensures that the user is a valid pace email user. DeactivateAccount removes access to app’s functionalities when users are no longer associated with pace or choose to do so.

Team hearts (Sean – Context; Dean – Behavioral; Anna – Interactive&Structural)