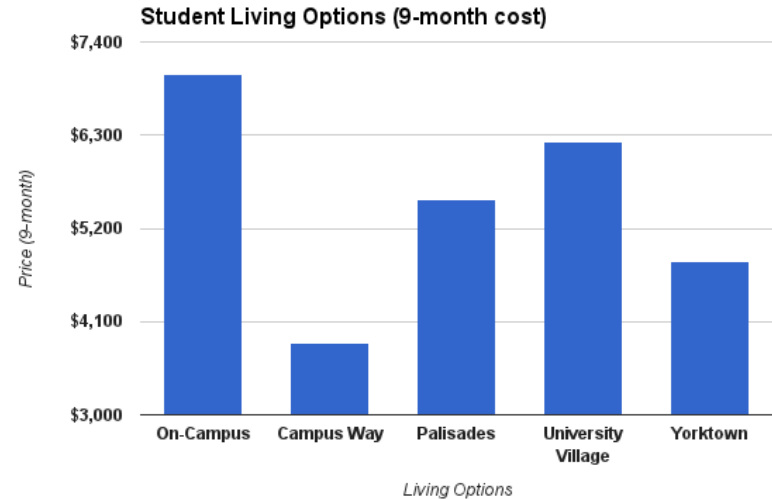


# College Living

Blake Dunn, Derek Overlock, Lumin Shi

# System Definition

- \* On-campus housing at UA is expensive
- \* Off-campus housing in Tuscaloosa is cheaper (9 month price of rent)
- \* College students need a safe and convenient way to interact



# System Scope

- \* College Living will help localize and refine the search for roommates and apartments
- \* Users will initially answer a set of questions
  - \* Answers are used to rate compatibility with other users
- \* User and Apartment Information will be stored on remote database
- \* Apartment Contacts will use web-service
- \* Users can find other users to interact with

# System Domain



Mobile  
Operating System  
(Android)

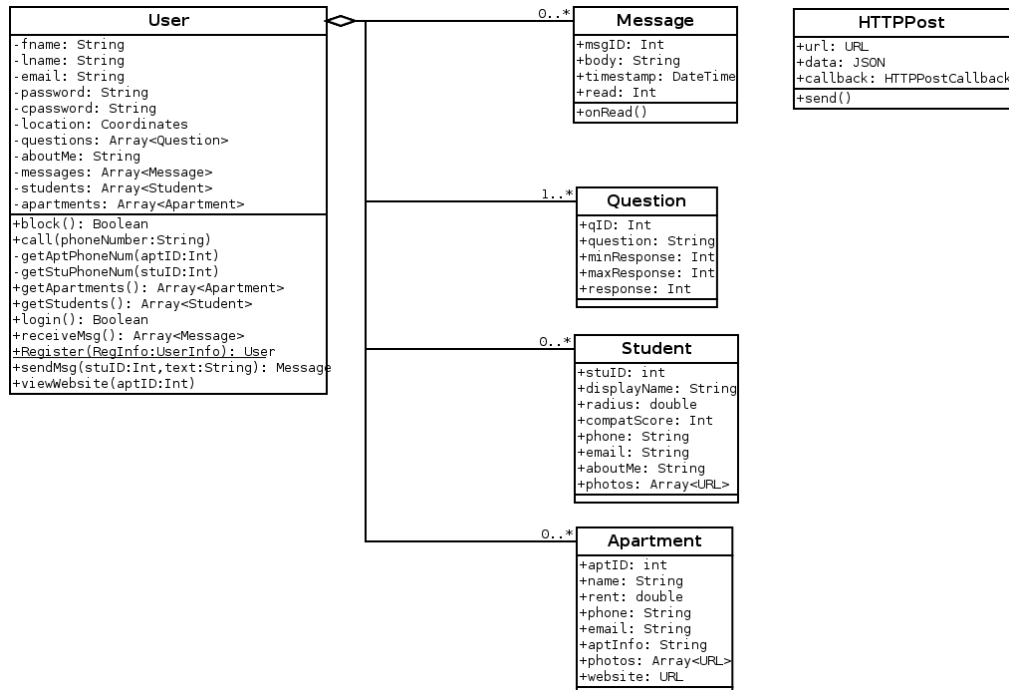


Web Server  
with Server Side Language  
(Apache w/ PHP)



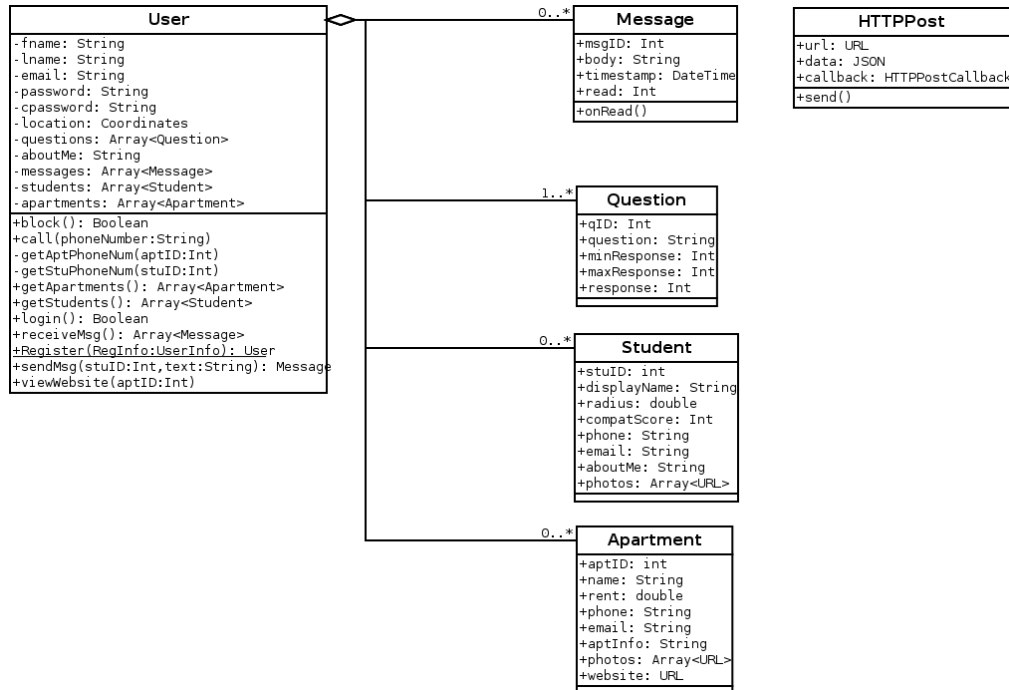
Database  
(MySQL)

# Class Diagram



- \* **User Class**
  - \* Logged-in user
  - \* Contains local apartments, students, and messages
  - \* Methods (Logging In, Registering, Messaging, etc.)
- \* **Student/Apartment Class**
  - \* Name, distance, profile, photos, etc.
  - \* Get/Set functions
- \* **Message Class**
  - \* Object for message transferring

# Class Diagram (cont'd)



## \* Question Class

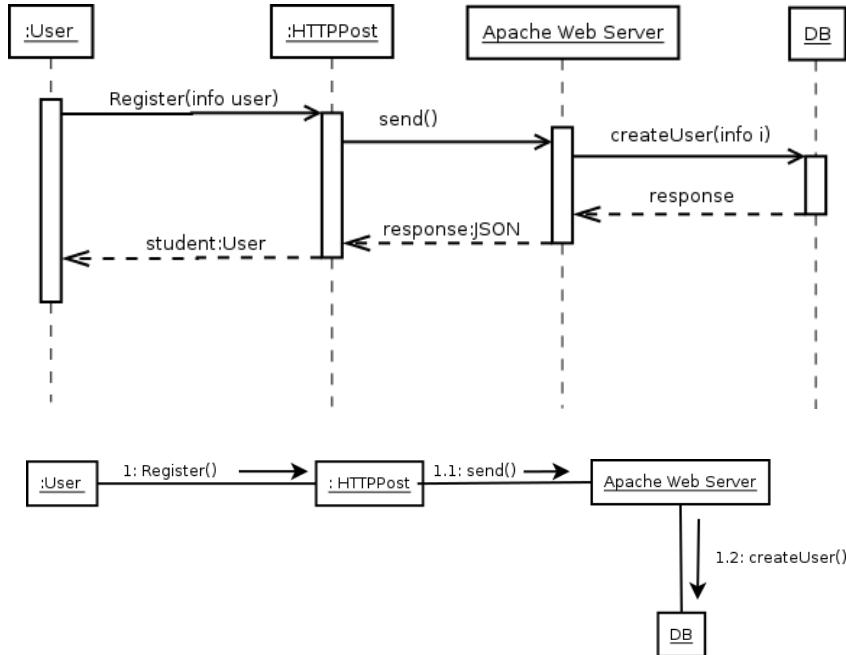
- \* Questions in questionnaire
- \* Used with registration
- \* Used for determining compatibility

## \* HTTPPost

- \* Bridge between device and server
- \* HTTP POST request with JavaScript Object Notation (JSON) body
- \* Response will be JSON decoded string

# Seq. & Com. Diagrams

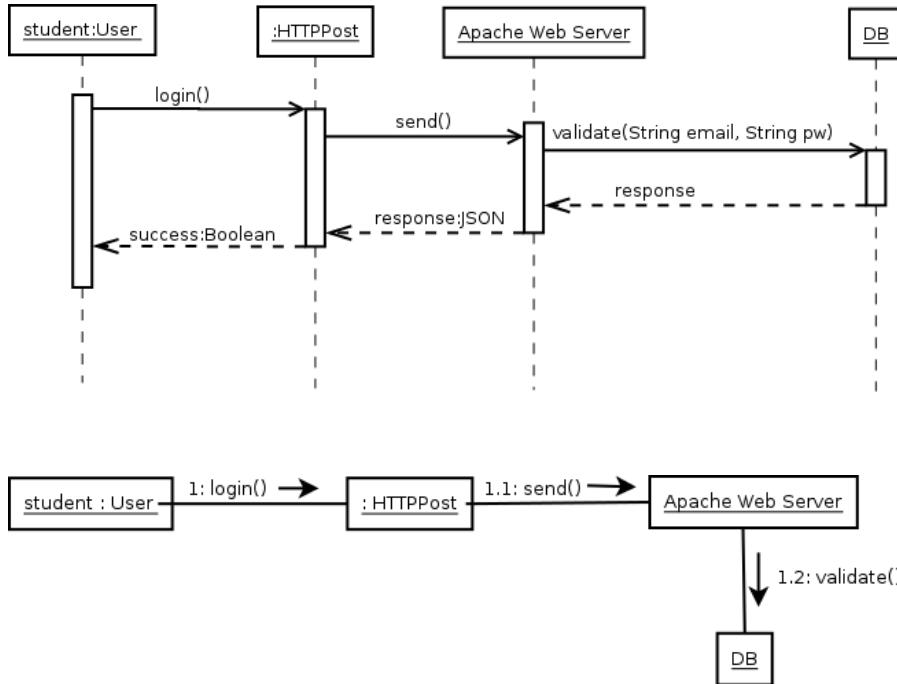
## User Registration



- \* `User.Register()` will send user information to server
- \* Server will create user
- \* `User.Register()` will return initialized User object

# Seq. & Com. Diagrams

## User Logging In

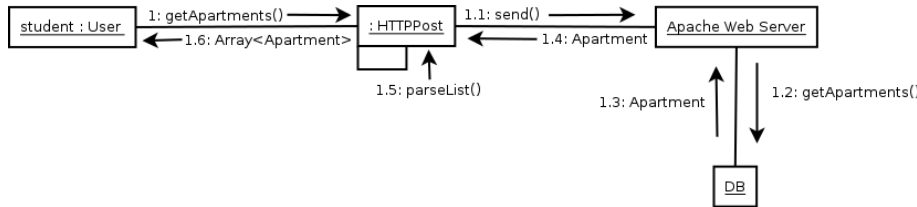
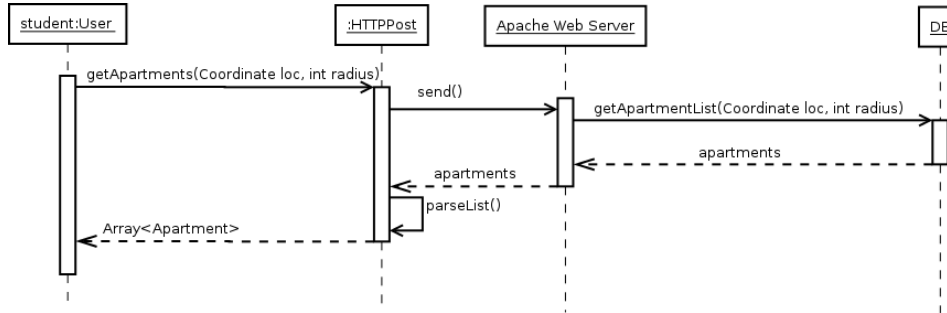


- \* User calls login with username and password
- \* HTTPPost sends request to server
- \* Server validates login with DB
- \* Returns success boolean



# Seq. & Com. Diagrams

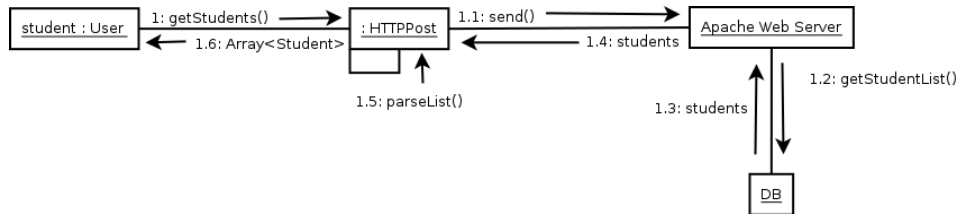
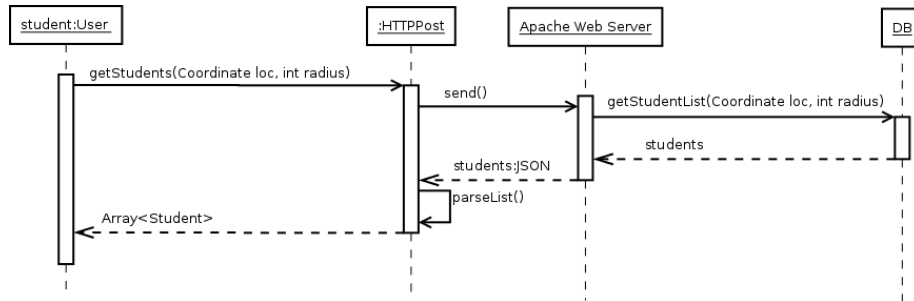
## Retrieving Local Apartment List



- \* User calls `getApartment()` with current location and search radius
- \* HTTPPost sends request to server
- \* Server queries DB and pulls local apartment records, returns to User

# Seq. & Com. Diagrams

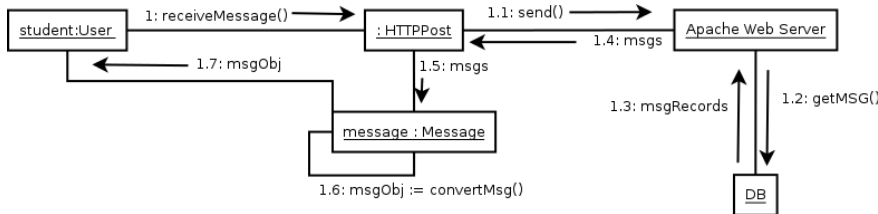
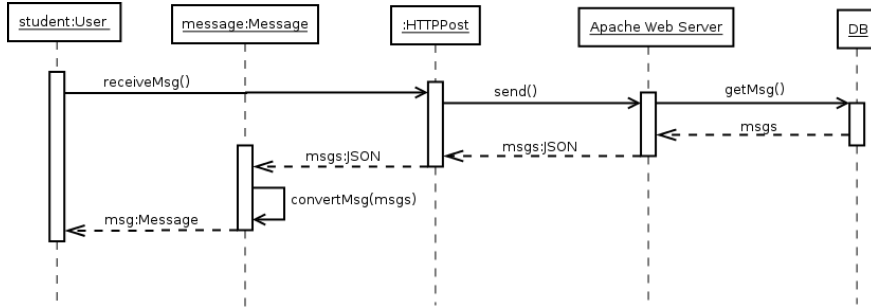
## Retrieving Local Student List



- \* User calls **getStudent()** with current location and search radius
- \* HTTPPost sends request to server
- \* Server queries DB and pulls local student records, returns to User

# Seq. & Com. Diagrams

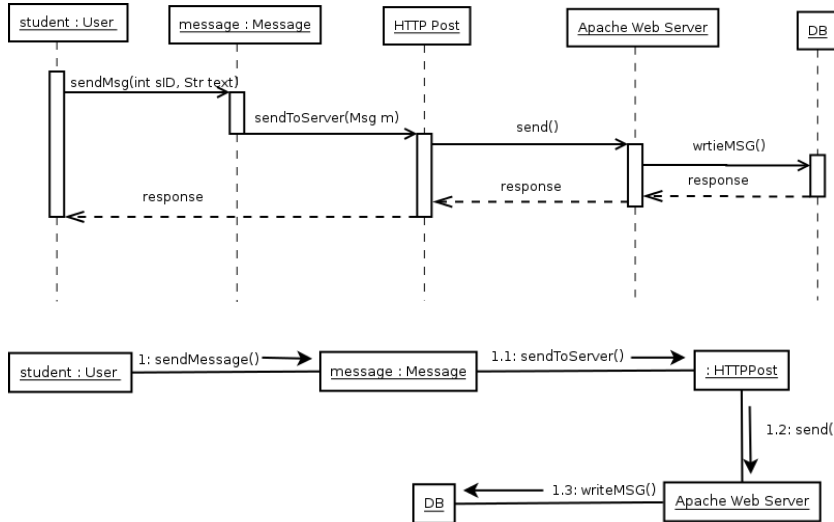
## Retrieve New Messages



- \* User will call receiveMsg()
- \* HTTPPost will send request to server
- \* Server will get new messages from DB
- \* Server returns new messages to User

# Seq. & Com. Diagrams

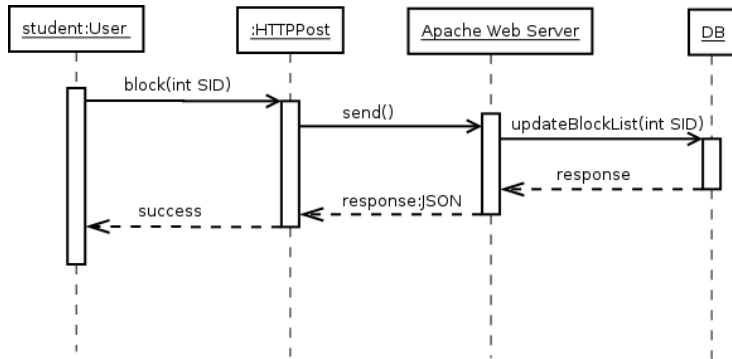
## Sending Messages



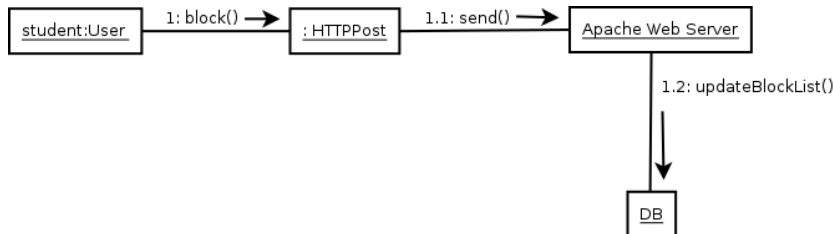
- \* User will call `sendMsg()` with student ID and message content
- \* HTTPPost will send message to server
- \* Server will post message to DB for other user to retrieve

# Seq. & Com. Diagrams

## Blocking a User

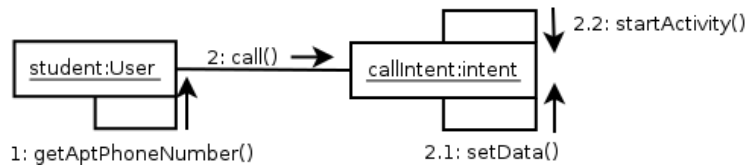
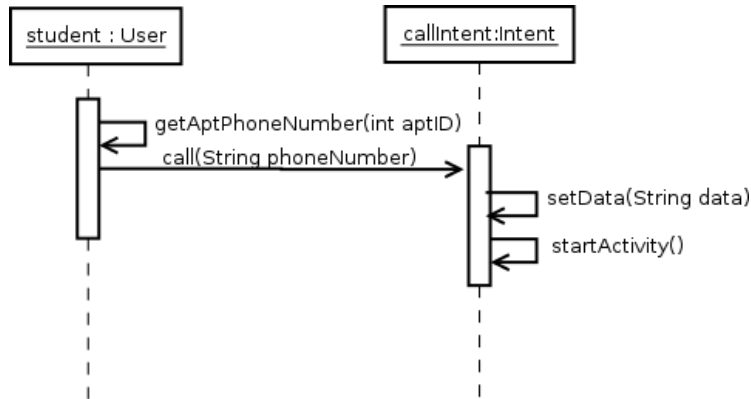


- \* User will call `block()` with student ID passed in
- \* HTTPPost will send request to server
- \* Server will update DB to reflect block



# Seq. & Com. Diagrams

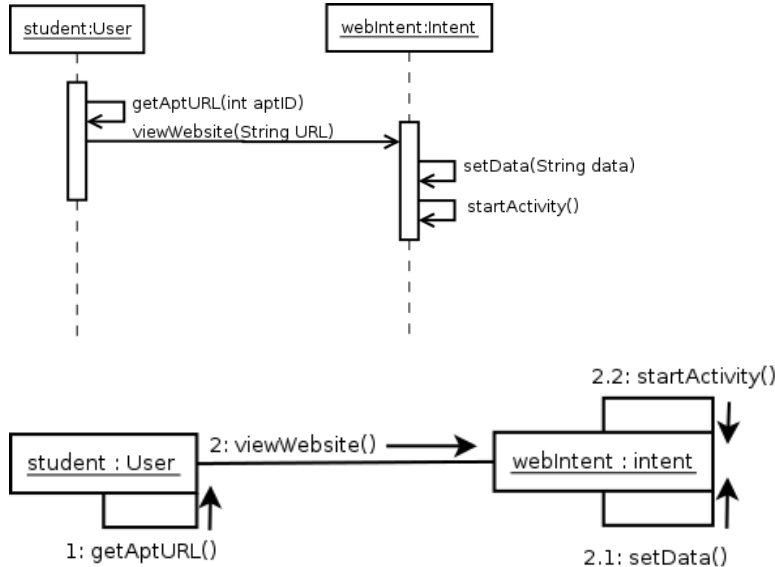
## Calling an Apartment



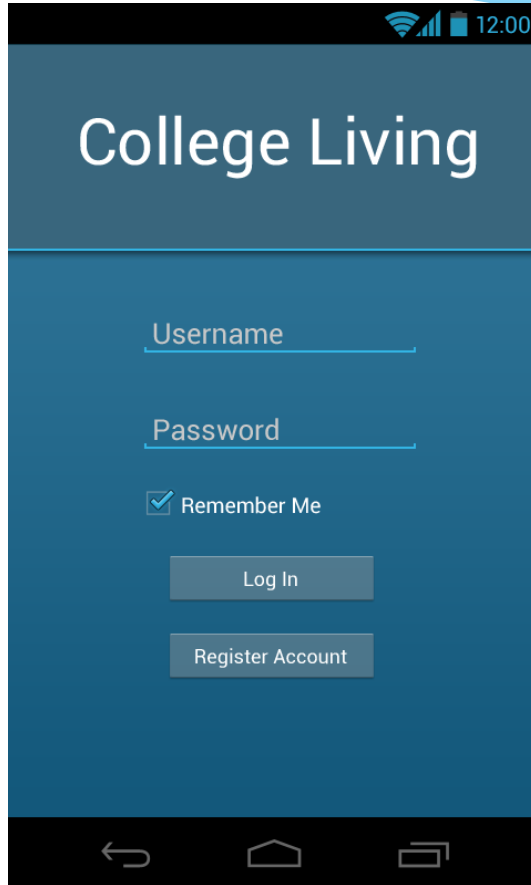
- \* User calls `getAptPhoneNumber()` with apt ID to get phone #
- \* User calls `call()` with apt. phone #
- \* Intent is used to start call activity

# Seq. & Com. Diagrams

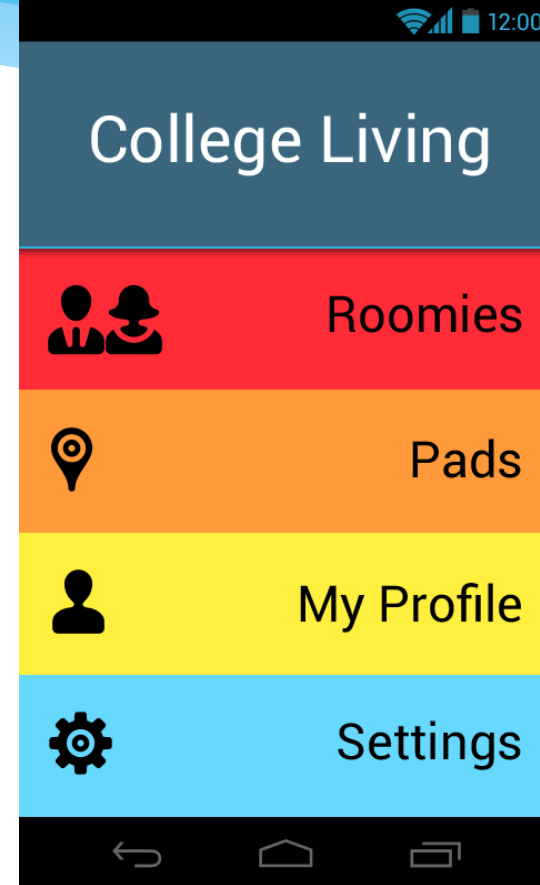
## Viewing Apartment Website



- \* User will call `getAptURL()` with apt. ID
- \* User will call `viewWebsite()` with the URL
- \* `webIntent` is initialized with apartment URL

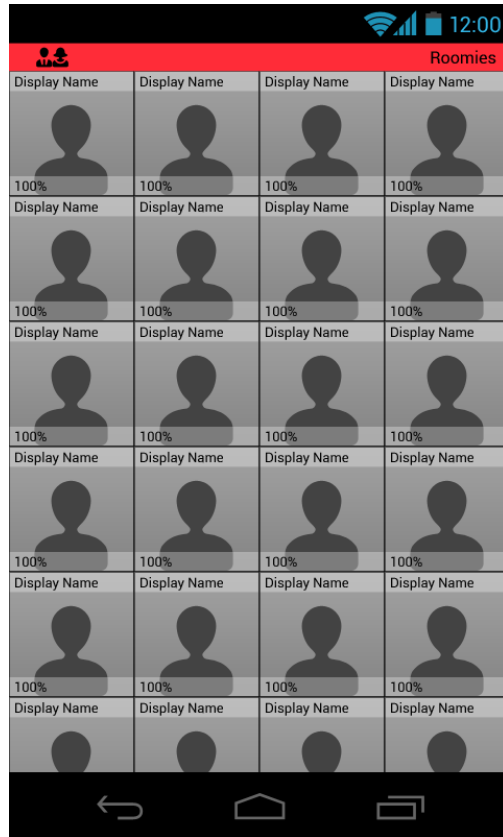


Log In Screen



Home Screen





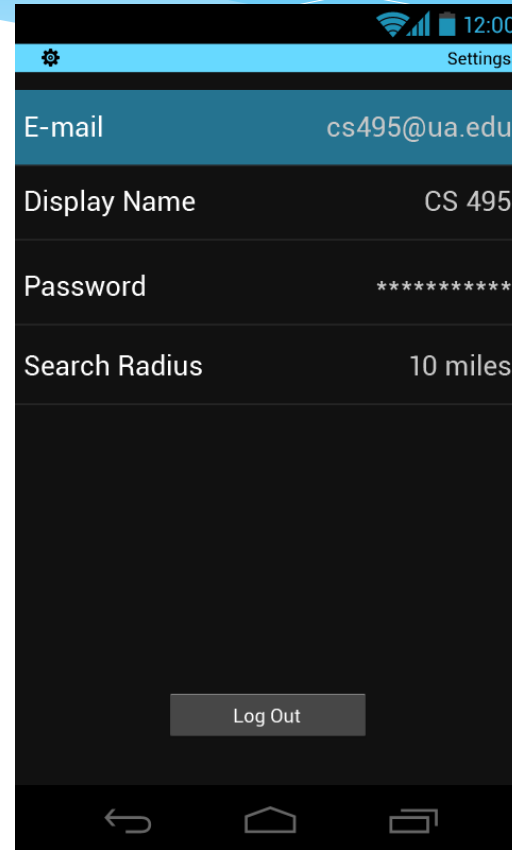
Roomies Screen



Pads Screen



General Profile Screen



Settings Screen