



DRIVEWAY

Parking

Screen Sketches

1_CW_7

Varun Advani
Jia Ze Chong
Lindsay Timko
Alex Volkov

Renter

- ❖ find nearby listings as per address request
- ❖ make reservations based on listing availability
- ❖ view the list of upcoming reservations
- ❖ update or cancel upcoming reservations
- ❖ message hosts / review notifications
- ❖ update profile information, including financial information

Admin

- ❖ approve new host listings
- ❖ resolve disputes between renters and hosts
- ❖ verify renter / host government IDs
- ❖ remove users from the system
- ❖ monitor / delete offensive reviews

Host

- ❖ add a new listing
- ❖ accept reservations
- ❖ view the list of upcoming reservations
- ❖ approve changes or cancel upcoming reservations
- ❖ message hosts / review notifications
- ❖ update profile information, including financial information

Requirements

- ❖ support for multiple types of users (renters / hosts / admins)
- ❖ support for unlimited number of renters / hosts
- ❖ real-time updates of search results and messages
- ❖ user-friendly interface
- ❖ fast response to user interaction with any UI element (less than 0.5 seconds per request)

User: keeps information regarding each user's account

- ❖ UserID – (randomly generated)
- ❖ Email address (unique)
- ❖ Password (not null)
- ❖ Name (not null)
- ❖ Phone number (not null)
- ❖ Credit card information
- ❖ Bank account information
- ❖ User type (renter or host—updated based on login choice)
- ❖ *Primary key:* UserID

Property: holds information regarding each location available for rental

- ❖ PropertyID (randomly generated)
- ❖ Address (not null, unique)
- ❖ Cost per hour
- ❖ Host – foreign key to UserID in **User** table
- ❖ *Primary key:* PropertyID

Bookings: stores information regarding books

- ❖ ReservationID (randomly generated)
- ❖ PID – foreign key to PropertyID in **Property** table (not null)
- ❖ RenterID – foreign key to UserID in **User** table (not null)
- ❖ Check in date and time (not null)
- ❖ Check out date and time (not null)
- ❖ *Primary key:* ReservationID

Admin: stores information regarding administrative users

- ❖ AdminID (randomly generated)
- ❖ Email address (unique)
- ❖ Password (not null)
- ❖ Name (not null)
- ❖ Phone number
- ❖ *Primary key:* AdminID

Reviews: contains information regarding reviews

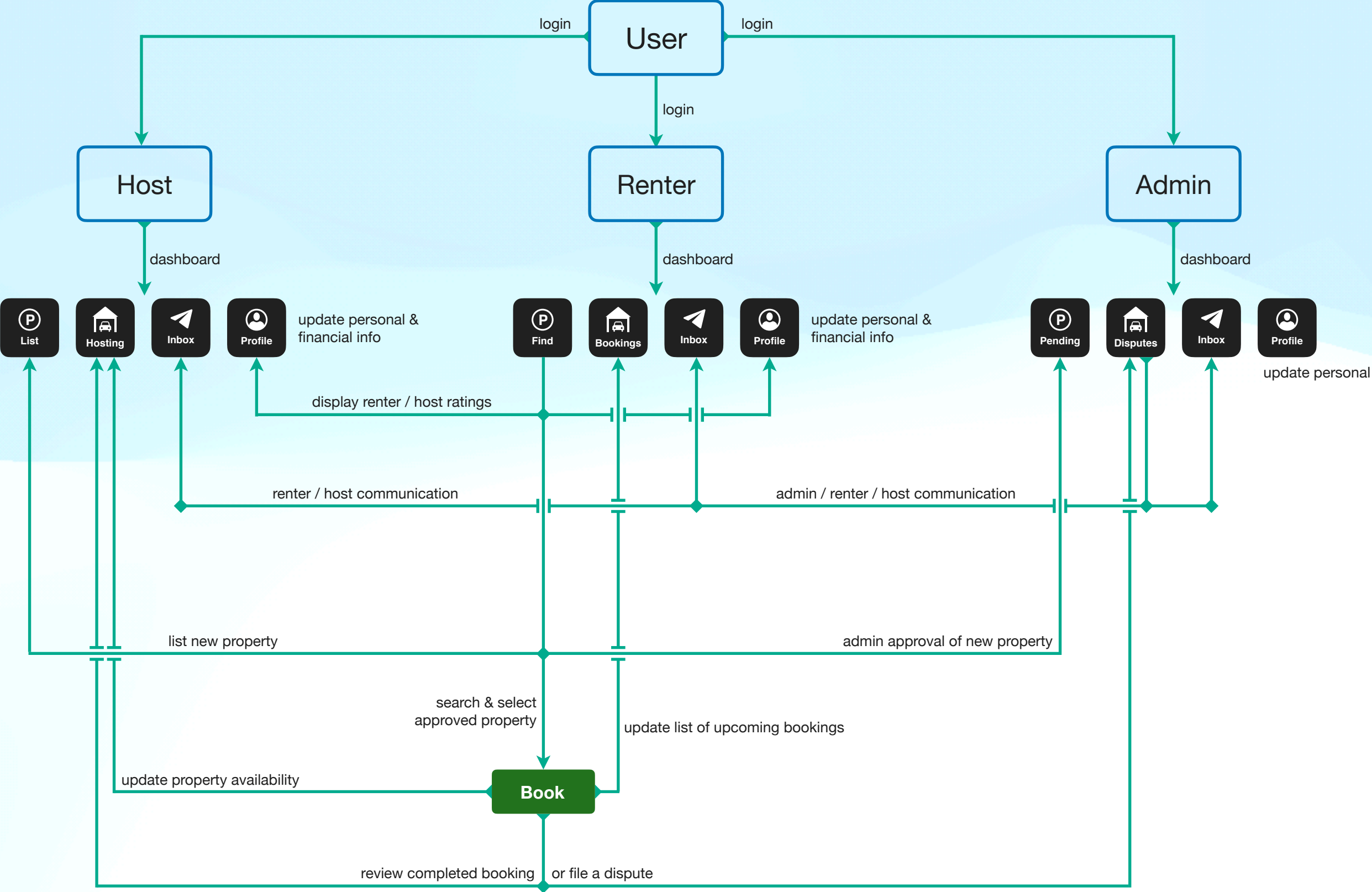
- ❖ RID – foreign key to ReservationID in **Bookings** table
- ❖ Accommodation rating (not null)
- ❖ Safety rating (not null)
- ❖ Responsiveness rating (not null)
- ❖ Comments
- ❖ *Primary key:* RID

Messages: contains all incoming and outgoing messages for inbox

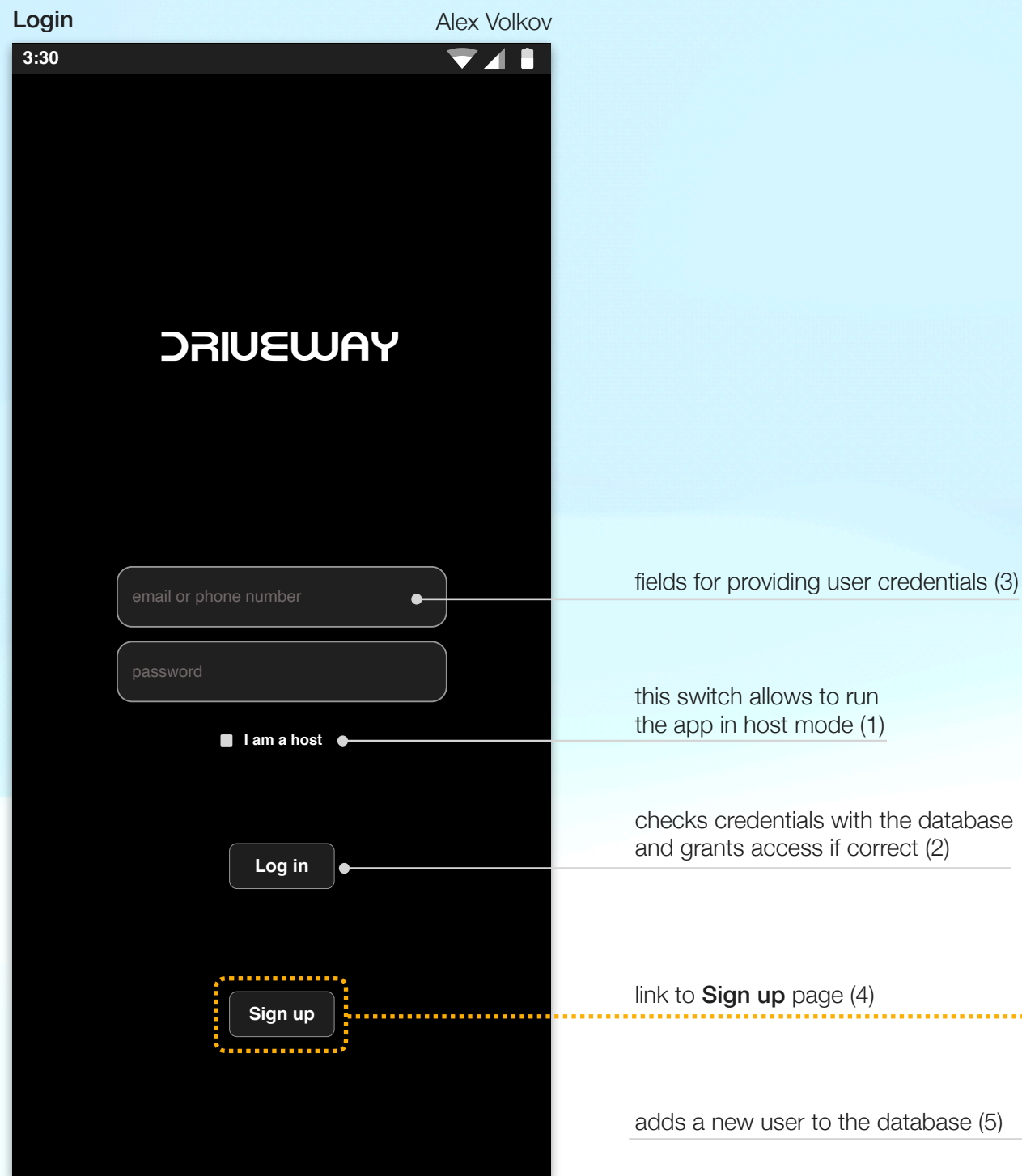
- ❖ MessageID (randomly generated)
- ❖ Sender – foreign key to UserID in **User** table (not null)
- ❖ Receiver – foreign key to UserID in **User** table (not null)
- ❖ Date sent (not null)
- ❖ Message body (not null)
- ❖ *Primary key:* MessageID

Disputes: contains information regarding

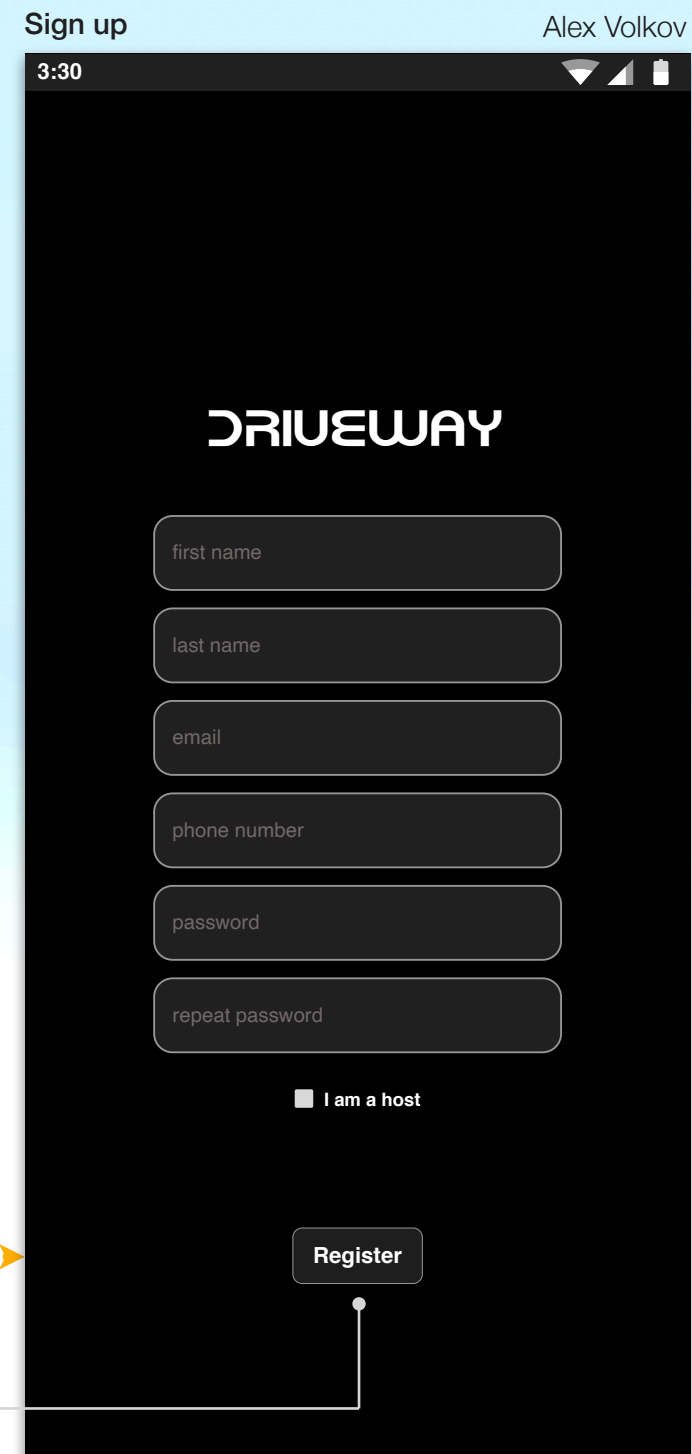
- ❖ RID – foreign key to UserID in **User** table
- ❖ Admin – foreign key to AdminID in **Admin** table (not null)
- ❖ Resolved (not null)
- ❖ *Primary key:* RID



Screen Prototypes

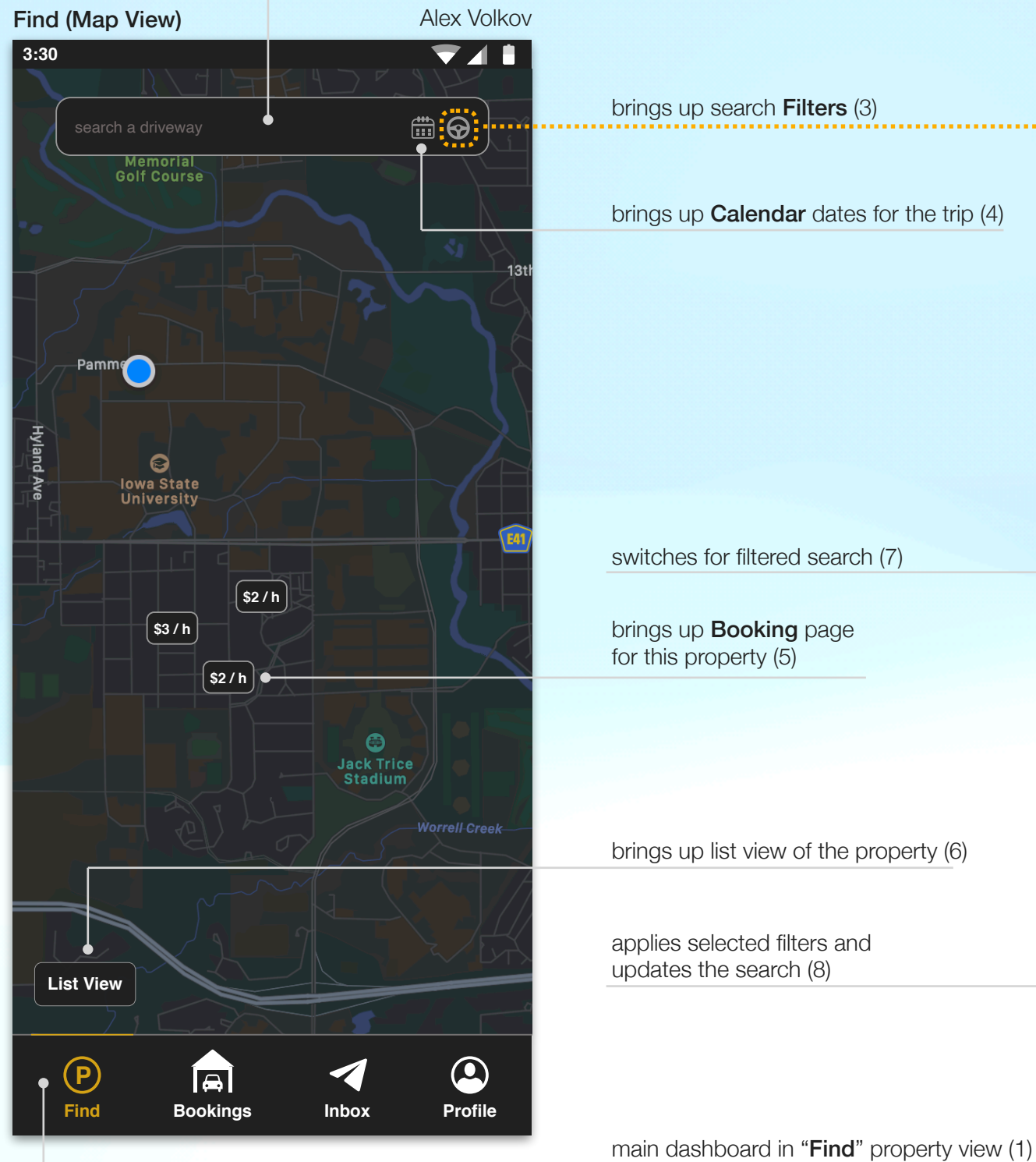


Login screen is important not only to authenticate a user but also for determining the application mode, i.e. renter, host, or administrator. By default, the user will be logged in as a renter. However, **I am a host** switch (1) triggers the host mode. If the administrator credentials are provided, the app will run in the admin mode, regardless of the status of the switch. When pressed, the **Log in** button (2) sends a request to the server to verify the correctness of the credentials (3), allowing access upon a successful match. A new user must press the **Sign up** button (4) to register.

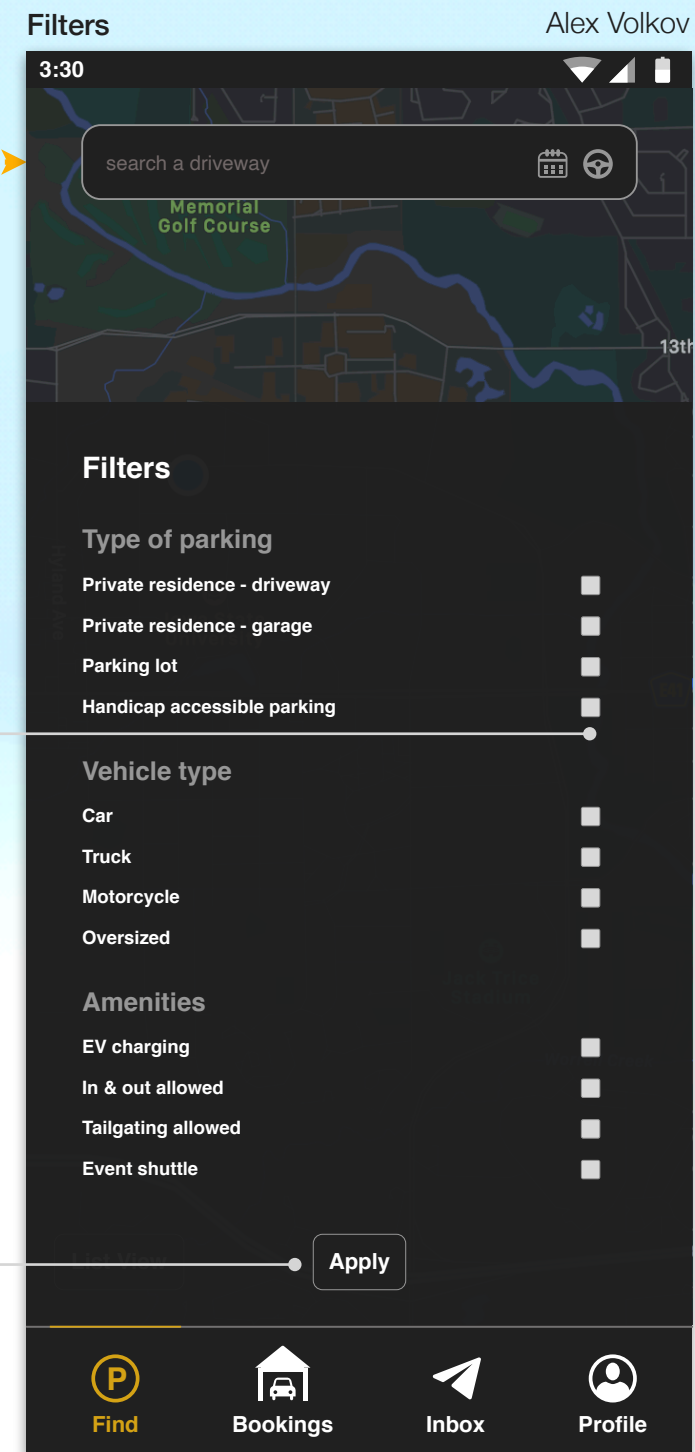


Sign up screen allows the user to register to use the services of the app. First name, last name, email, phone number, and password (repeated for error-checking) are required. Email is the primary key for the user, so a verification link will be sent to the provided email to confirm it. A user can also check **I am a host** switch to log in to the host mode. When pressed, the **Register** button (5) creates a new user if the user with the provided email doesn't already exist.

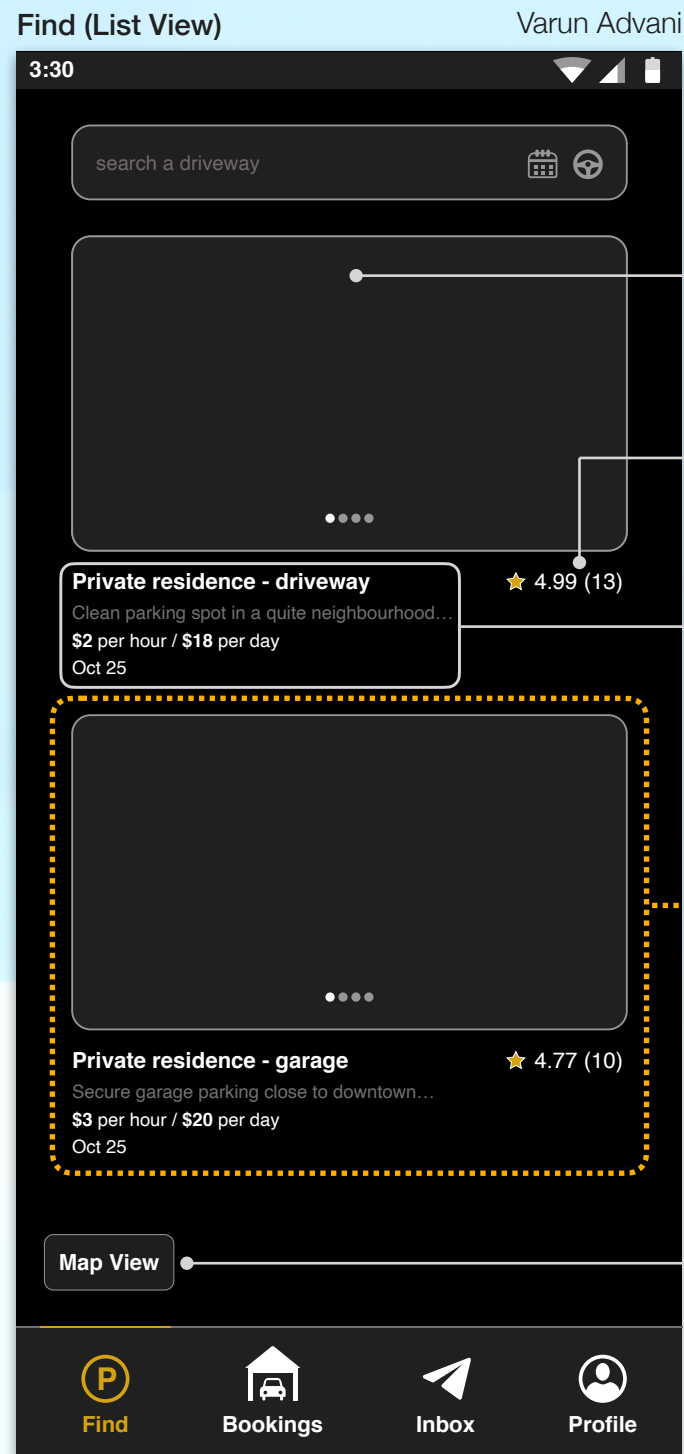
search field to find closest property by the given address (2)



Find (Map View) (1) view of the main dashboard is where most of the app logic is happening. A user can search listed property by providing an address in the search field (2). A user can also specify search parameters by selecting **Filters** (3) and providing trip dates in the **Calendar** (4). Properties (5) show up on the map as rates per hour. A user can switch from the current map view to the **List View** (6) by pressing the appropriate button.



Filters (3) view is used to specify a search of the driveway by the type of parking (private residence - driveway, private residence - garage, parking lot, handicap accessible parking), vehicle type (car, truck, motorcycle, oversized), and amenities (EV charging, in & out allowed, tailgating allowed, event shuttle) (7). To update the search with the specified parameters, a user needs to press **Apply** button (8).



brings back to search (10)

property pictures carousel (1)

average rating (reviews #) (2)

type, name, rate, and relevant date displayed for each property (3)

amenities from the search filters (6)

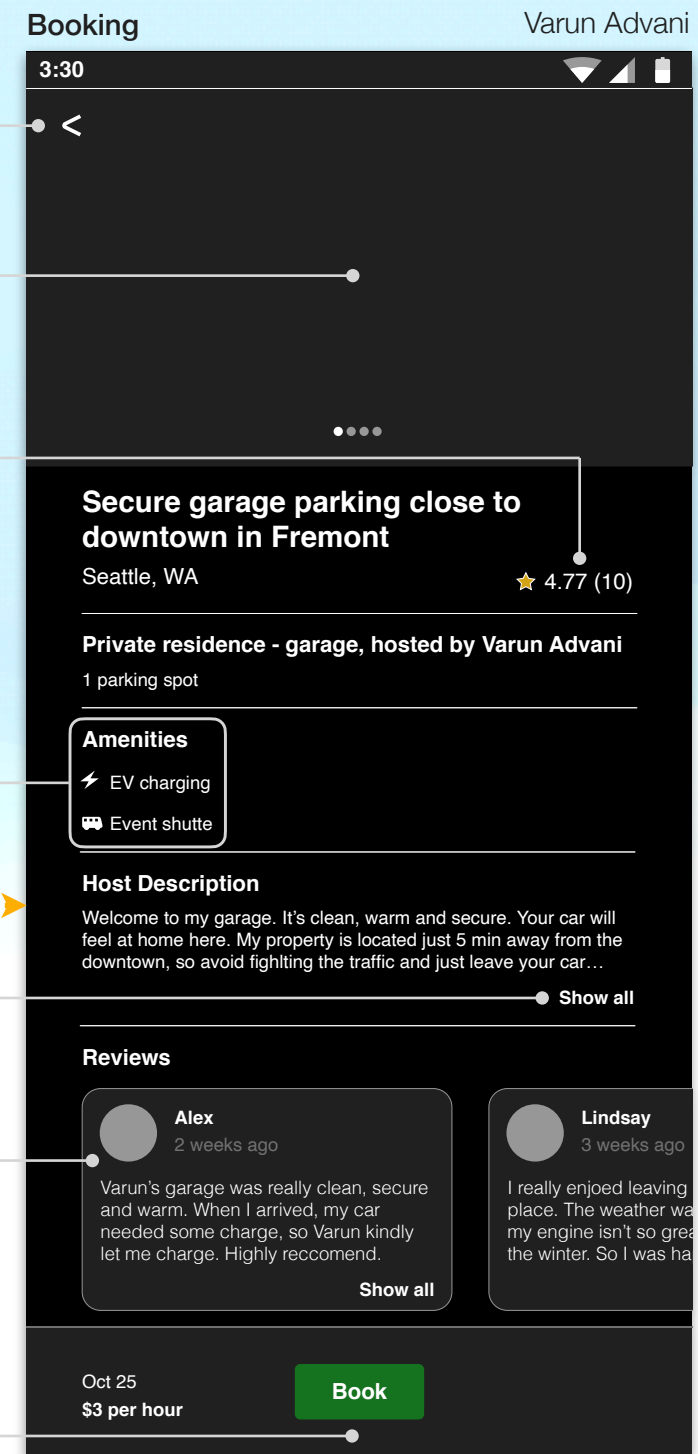
brings up **Booking** page for this property (5)

brings up full property description (7)

recent reviews (8)

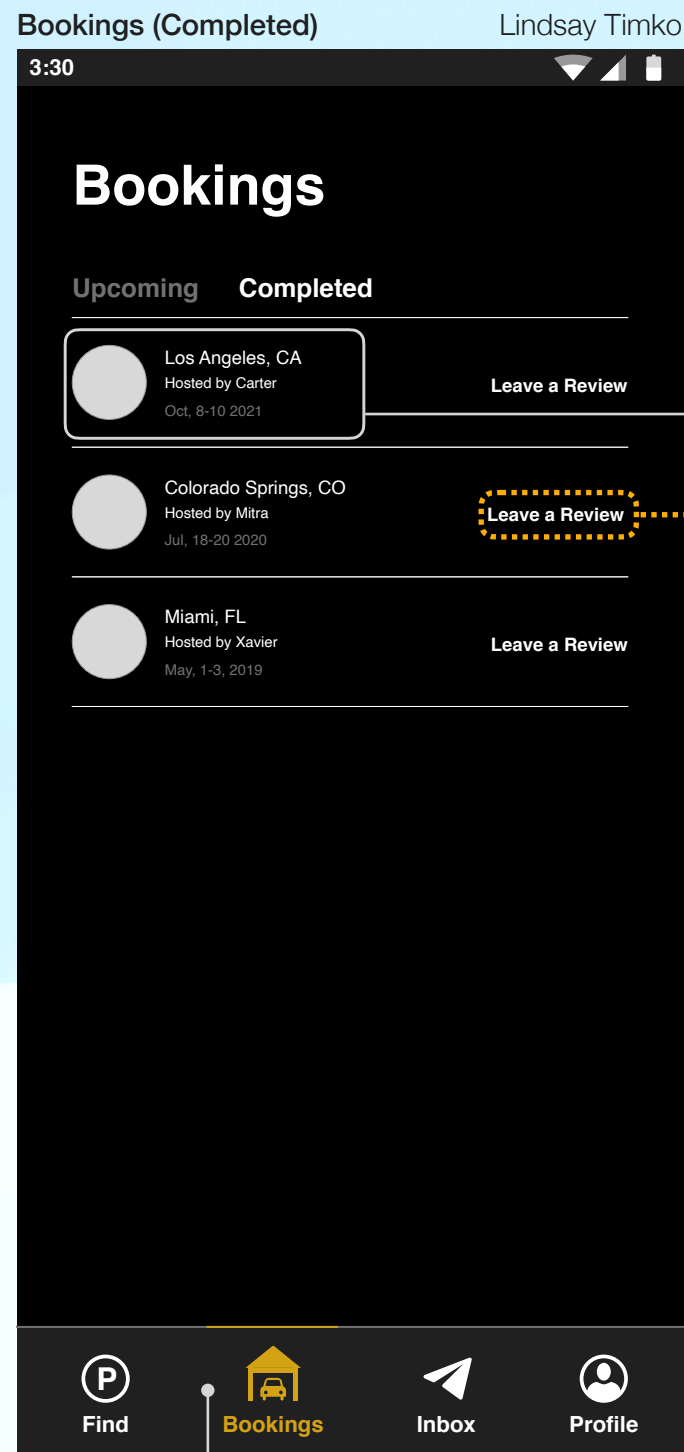
brings up map view of the property (4)

brings up booking review page (9)



Booking page includes an enlarged property pictures carousel (1), title of the property, average ratings with a number of reviews (2), type of parking, available amenities (6), host description (7), and recent reviews (8). A user can book the current property by pressing **Book** button (9). A left arrow (10) on top brings the user back to the search mode.

Find (List View) view of the main dashboard offers a more detailed overview of the search results. It includes a property pictures carousel (1), average rating with a number of reviews (2), type of parking, name, rate, and relevant date displayed for each listing (3). A user can switch back to the map view by pressing **Map View** button (4). A user can also select a listing by pressing on any of the listing UI elements (5).



brings up trip summary (2)

brings up **Review** page (3)

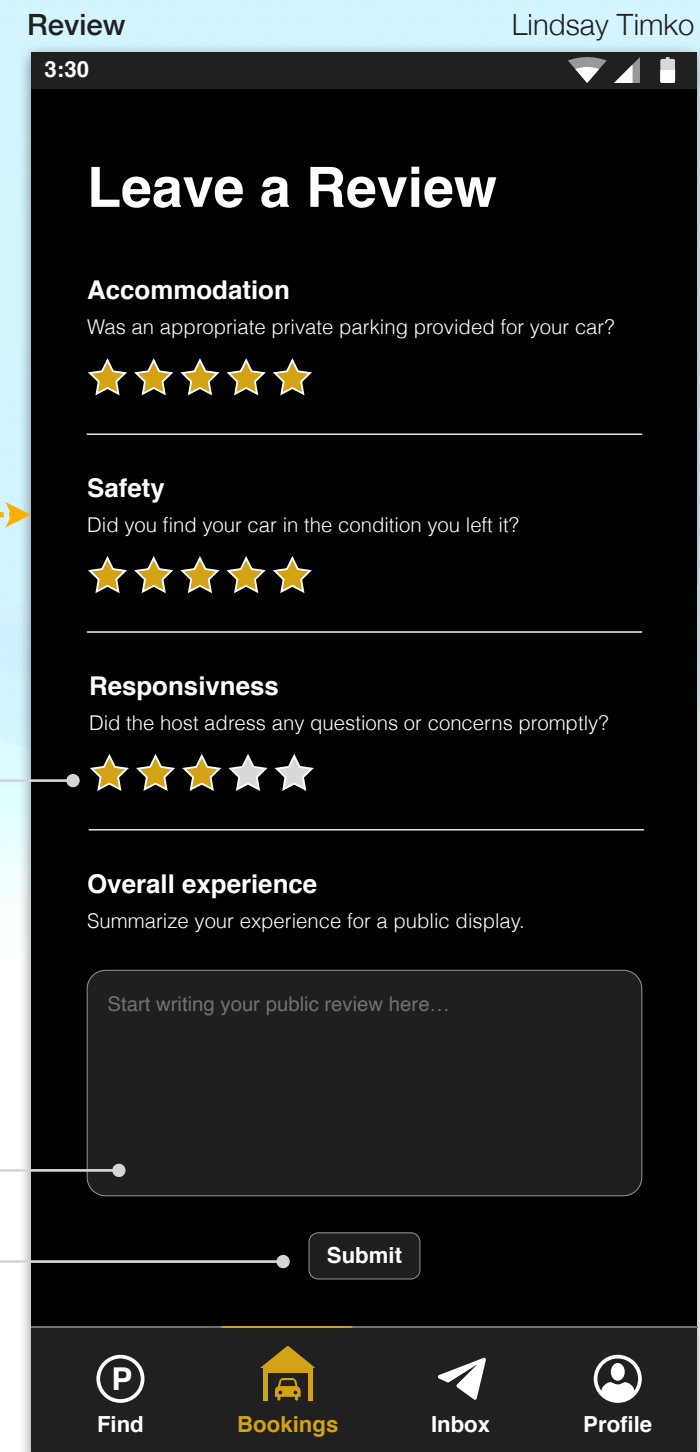
interactive star ratings (4)

field for public review (5)

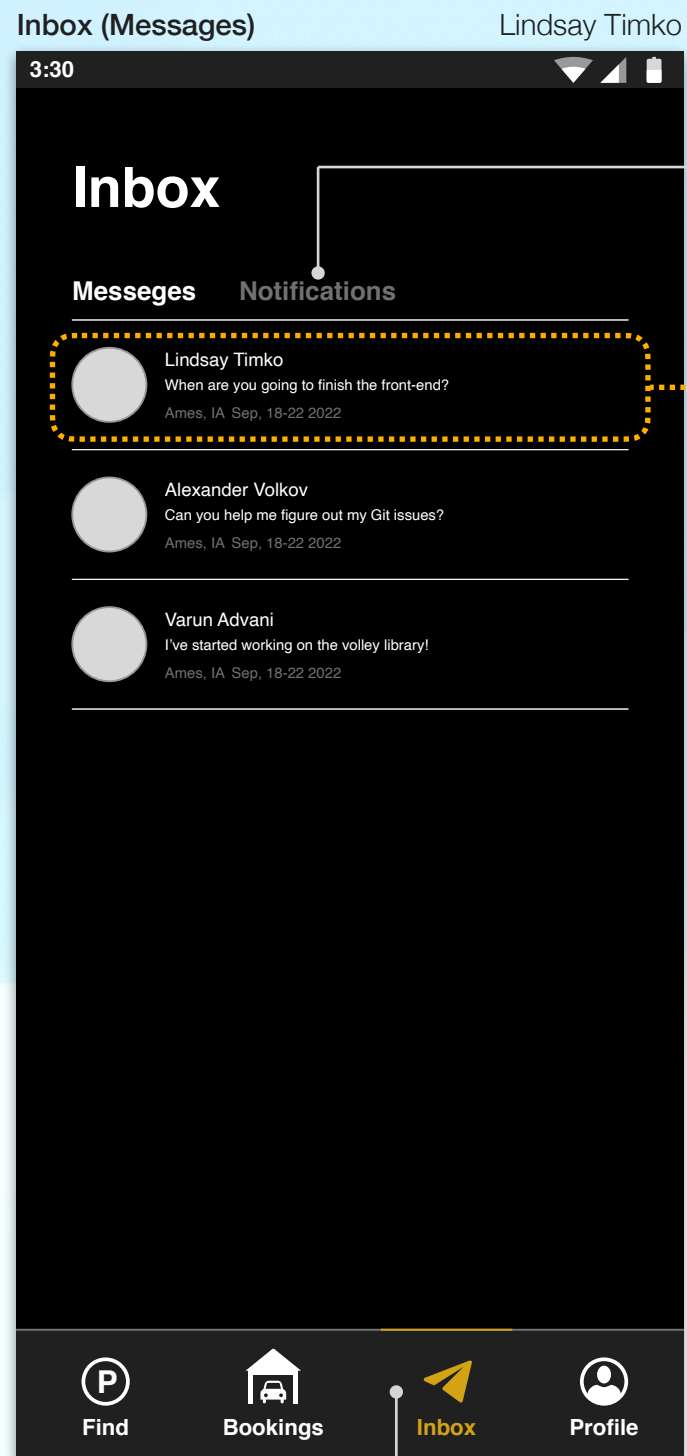
submission of the review form (6)

main dashboard in **"Bookings"** view (1)

Bookings (Completed) (1) view of the main dashboard summarizes the completed trips. A user can either bring up a trip summary (2) by tapping on any of the UI elements corresponding to each property or **Leave a Review** (3) for the property by pressing an appropriate button. On the trip summary page (not shown here), a user can file a complain if something goes wrong with the trip.



Review (3) page allows a user to give feedback about their experience after completing a trip. A user can use interactive star ratings (4) to answer questions from the following three categories: accommodation (Was an appropriate parking provided for your car?), safety (Did you find your car in the condition you left it?), and responsiveness (Did the host address any questions or concerns promptly?). A user can also leave a public review (5) and **Submit** their review (6) by pressing an appropriate button.



generates a URL link to this booking for sharing (5)

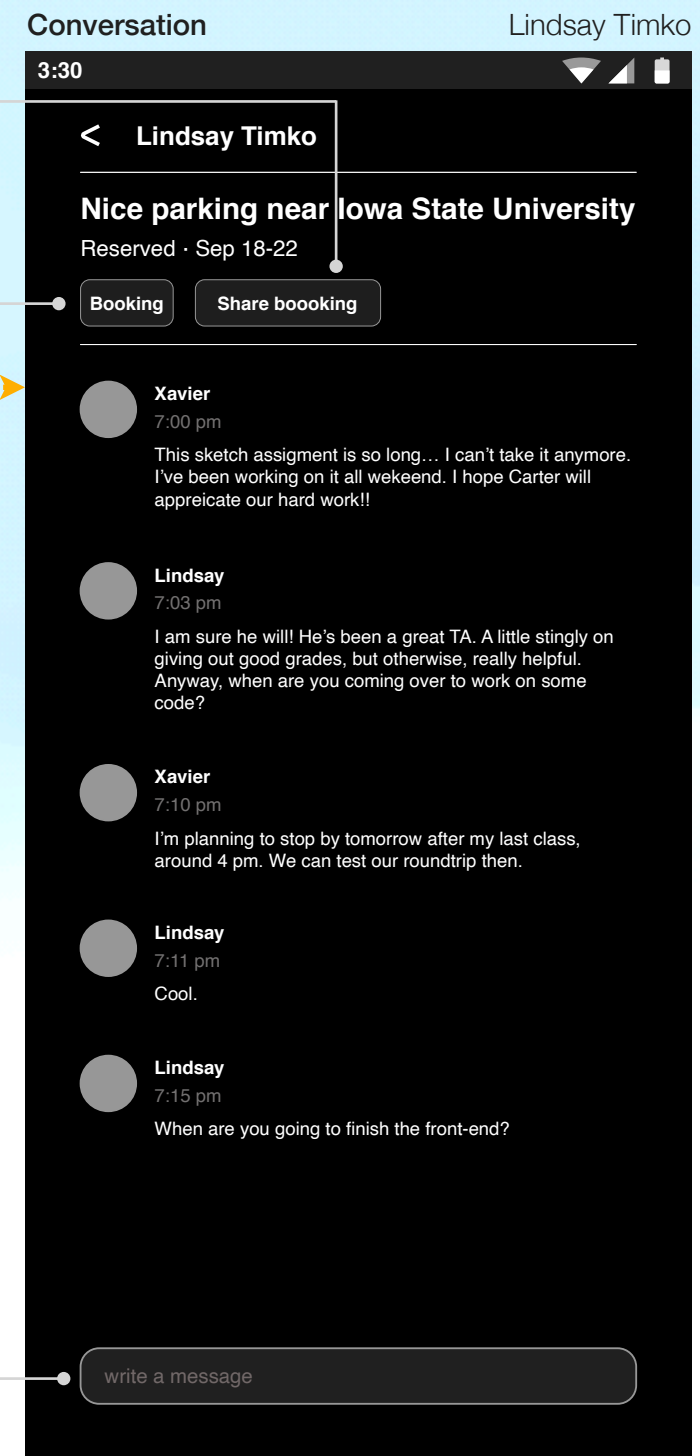
brings up **Notification** page (3)

brings up booking confirmation page (4)

brings up search **Conversation** page (2)

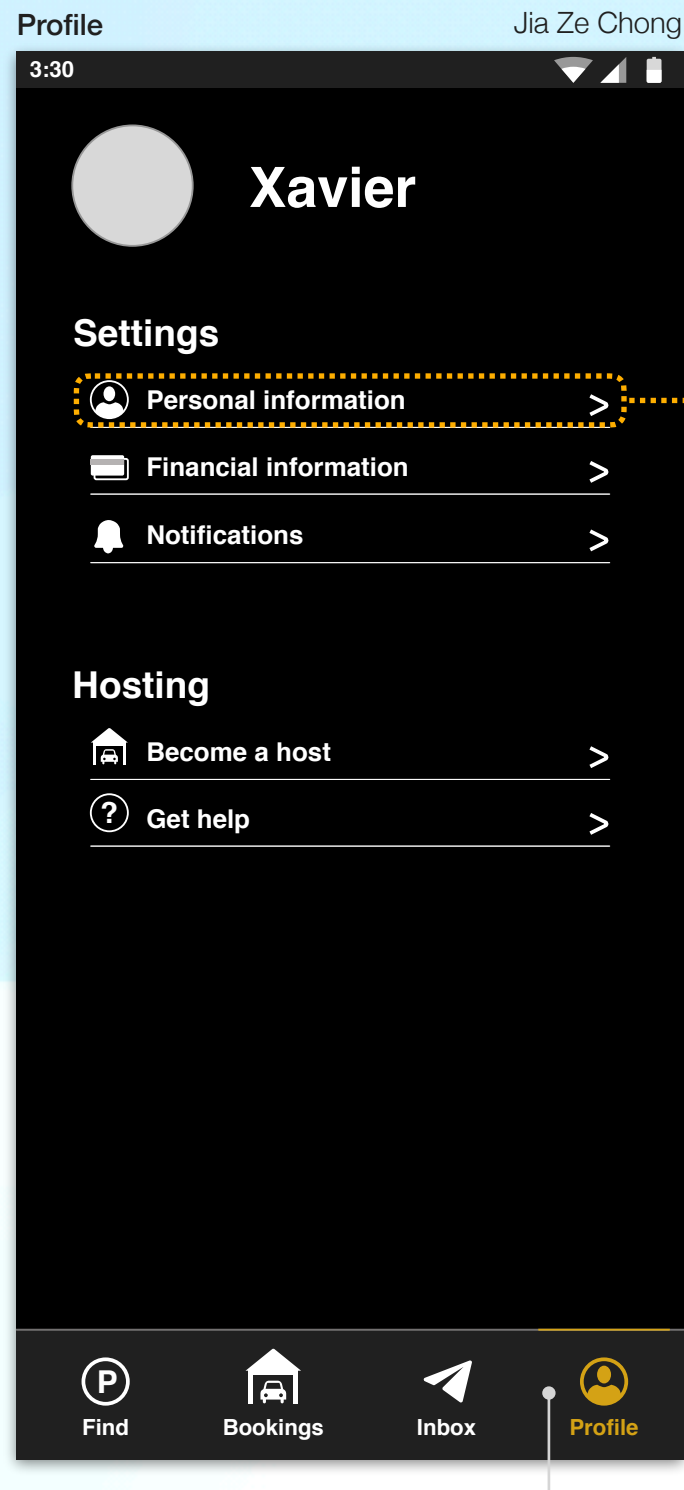
field for typing a message (5)

main dashboard in **"Inbox"** view (1)



Inbox (Messages) (1) view of the main dashboard brings up a list of all the conversations. Each conversation has an addressee name, a preview of the most recent message, as well as the trip details associated with the conversation (2). A user can open a conversation by tapping on any of the UI elements for each message. A user can also access recent **Notifications** (3) by tapping an appropriate button.

Conversations page provides a booking summary with the ability to access the **Booking** (4) confirmation or **Share booking** (5) by tapping appropriate buttons. When **Share booking** is pressed, a URL link of the current listing is generated and can be shared by text, email, or social media. A user can type a message in an appropriate field (5) at the bottom of the screen.



saves the updates to the database (8)

brings up search **Personal Info** page (2)

fields for updating personal info (3)

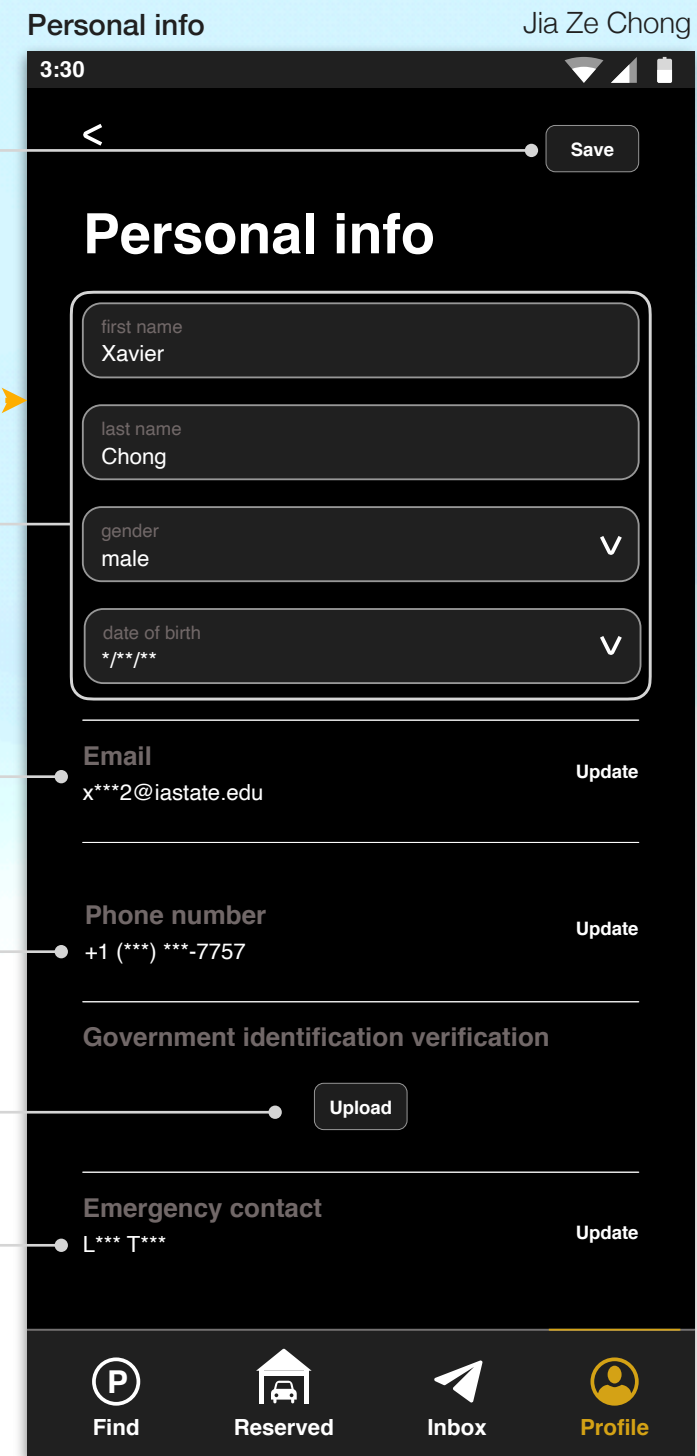
email update (4)

phone number update (5)

prompts to upload a photo
of government-issued ID (6)

emergency contact update (7)

main dashboard in **"Profile"** view (1)



Personal info page allows a user to update their first name, last name, gender, date of birth (3), as well as email (4) and phone number (5). A user must also provide a government-issued ID to be verified by the administrator before any properties can be booked. A user can upload a picture of their government-issued ID by pressing the **Upload** (6) button. A user can also add an emergency contact (7). To **Save** (8) the changes, a user needs to press an appropriate button.

Profile (1) view of the main dashboard brings up an account summary for a user, including personal information, financial information, notifications settings, as well as general information about how to become a host and get help. A user can update **Personal information** (2) by tapping an appropriate UI element.