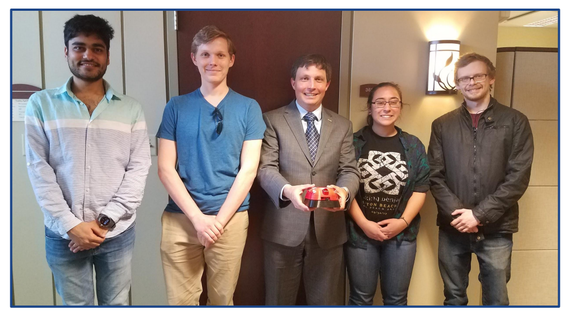
EfficienSeat

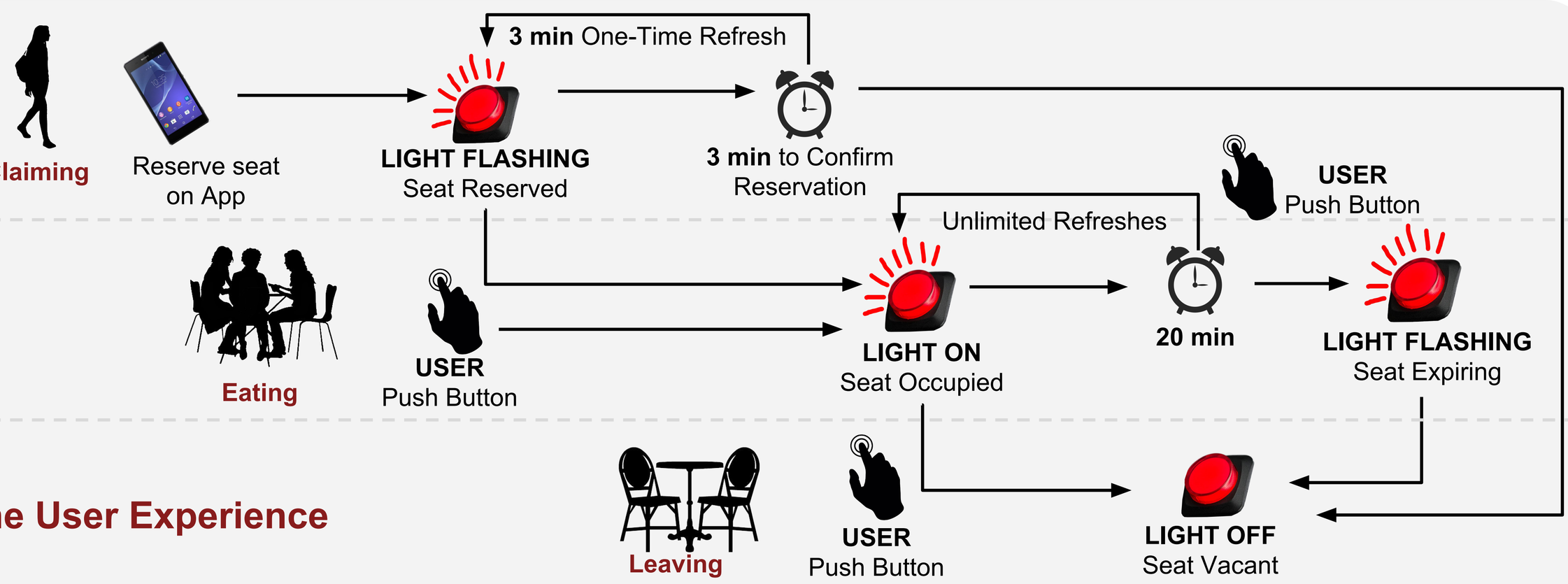
**Matthew Donnelly, Dennis Donoghue,
Kristina Georgadarellis, and Aarsh Jain
Faculty Advisor: Prof. Tilman Wolf**



Abstract

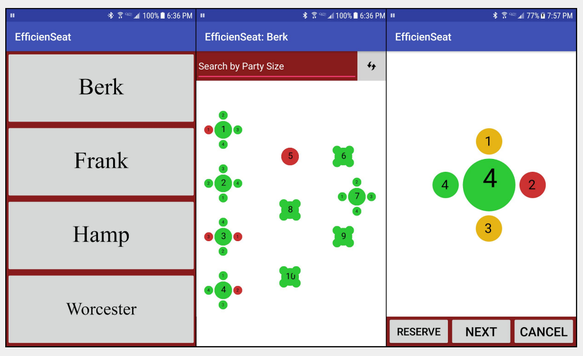
EfficenSeat is our solution to overcrowded dining halls. A mobile app displays a map of tables where users can view occupancy and reserve seats. Inside the dining hall, users can claim their seats by pushing a button on a modular unit aboard each table. This system aims to save the user time, while not disrupting the normal flow of operation.

The User Experience



Android App Tasks

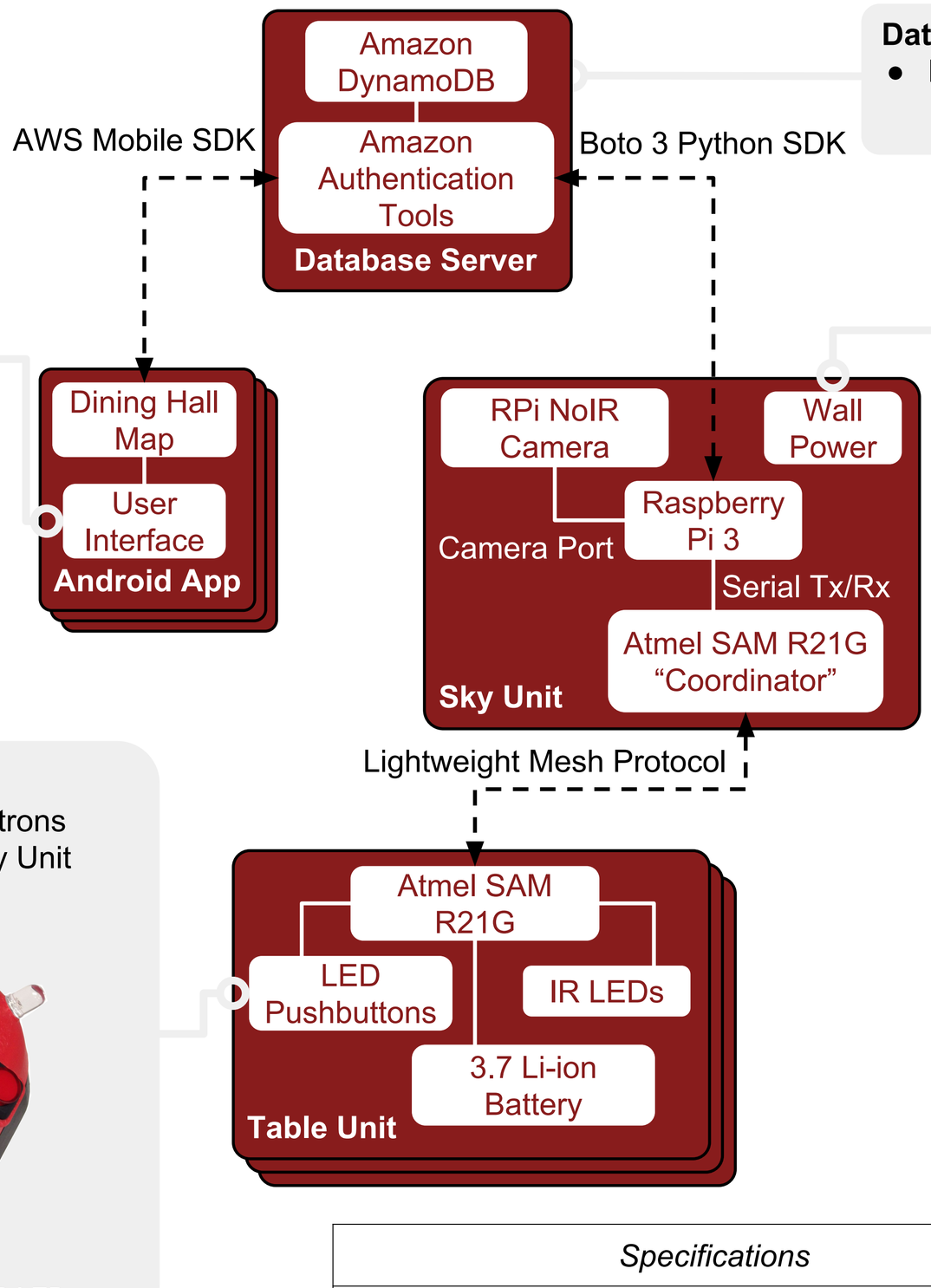
- Implement table search algorithm
- Render tables and seats
- Send seat reservation requests



Key:

Green: Open
Yellow Reserved
Red: Claimed, Other Users' Reservations

Different screens on the Mobile App



Database Server Task

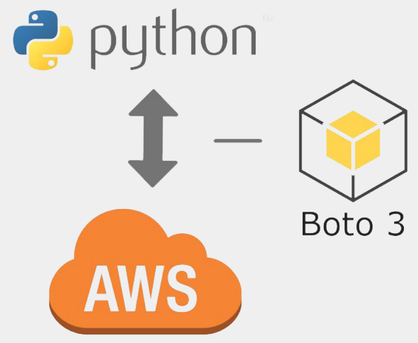
- Manage seat information

Sky Unit Tasks

- Table localization
- Communication with Table Unit
- Handle seat claiming requests

← 3D Printed Sky Unit Case

Sky Unit perspective of IR LEDs →



Python interaction with AWS via Boto 3 SDK

Table Unit Tasks

- Physical interface for patrons
- Relay table status to Sky Unit

Table Unit with pushbuttons, IR LEDs

Specifications	*Results*
Allow users to search by party size	Search support for party size ≤ 4
Updated map displayed within 2 seconds	Average Response is 38.6 ms
Table locations are accurate to ½ a table length	Centimeter precision
Can support > 100 seats	System is scalable at cost
Table unit is IPX4 (Splash-Proof)	Flexible Table Unit shell
Patrons can claim seats by pressing a button on the Table Unit	Interactive Table Unit
Weekly battery replacement	Battery Power lasts 6 days

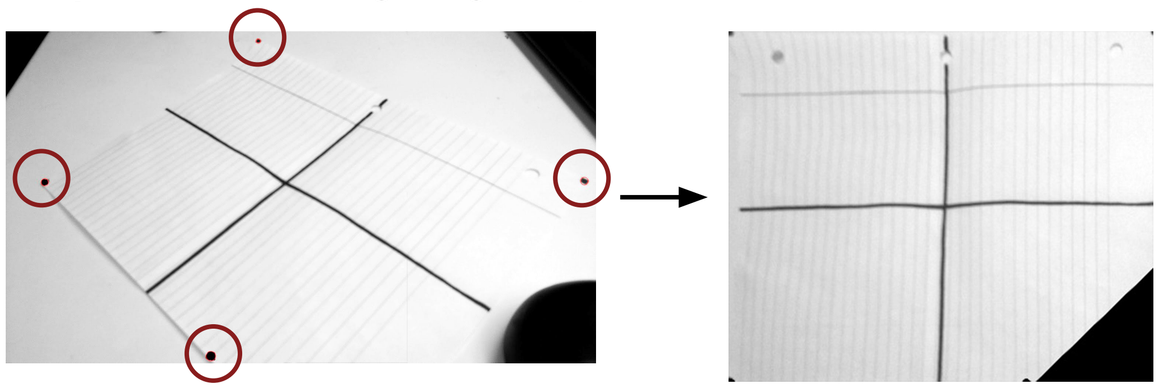
Acknowledgements

We would like to thank Professors Hollot and Soules for their continuous feedback, and Professors Goeckel, Kelly, and Holcomb for their expert advice. Also, many thanks to Fran Caron and M5 staff for their time and resources, especially Shira for taking care of our 3D prints. Finally, we would like to thank Professor Wolf for his invaluable time and dedication in making our vision a reality.

Table Localization

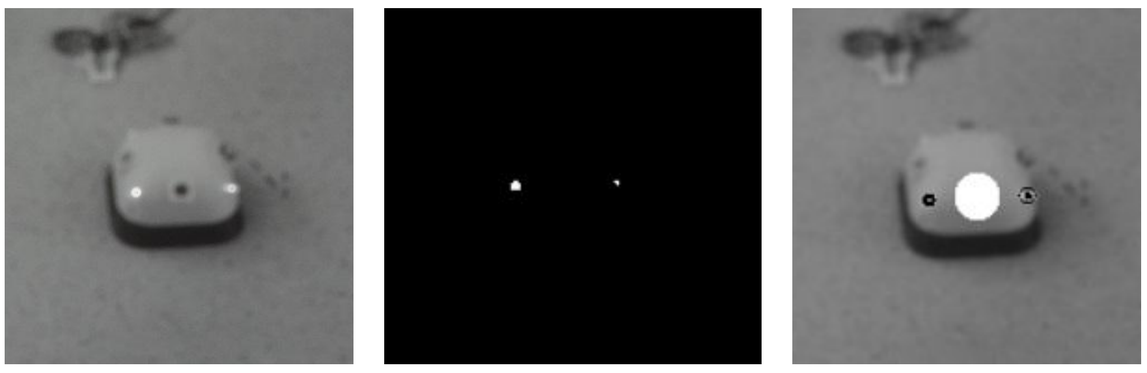
Purpose: Table position is needed for accurate map depiction

Step 1: Calibration - getting a “top down” view of room



1. Corners of dining hall are identified using the IR LEDs
2. Perspective transform takes 3D to 2D

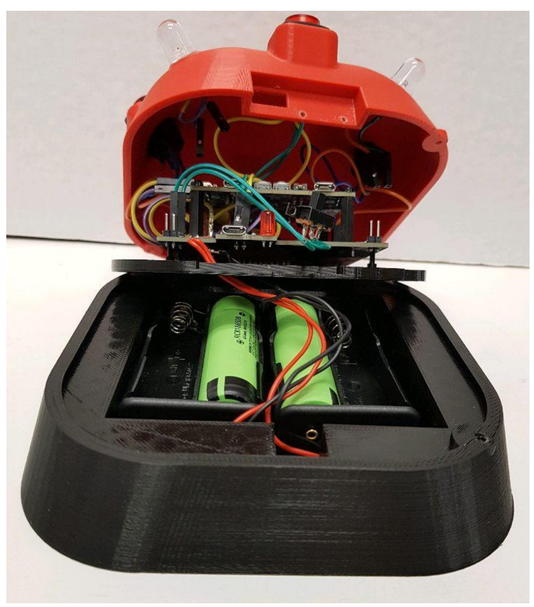
Step 2: Image Processing - find Table Units by IR LEDS

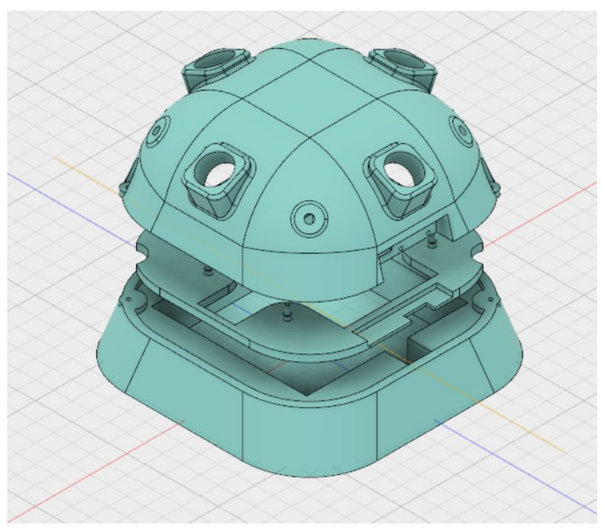


1. IR LEDs blink 2 times and camera captures ON/OFF states
2. Binary Thresholding and Gaussian Blur filter all images
3. 1st image searched for IR LED candidates using contouring
4. Compared with two other images to isolate Table Unit

Performance: Tested placing the unit in different locations

Table Unit Internals





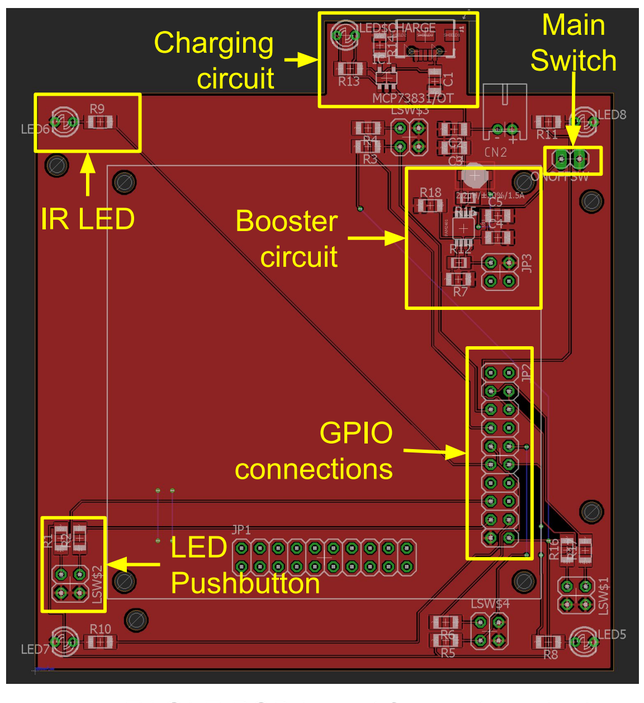
Top: *CAD model of Table Unit Case*
Left: *Printed case with components*

The Breakdown:

- Overall power consumption = 330mW
- Four 3400mAh batteries = 6 days of continuous operation

Low power, RF capable evaluation board driving Table Unit

Rechargeable Li-ion Batteries power the Table Unit



EAGLE PCB layout for custom design. Aboard is a charging circuit

Database

DynamoDB

amazon web services

Database storage includes:

- ID and seat status
- Location
- Orientation

Capabilities:

- Conscientious data access
- Auto scaling capability

tableID	tableStatus	seat1	seat2	seat3	seat4	tableX	tableY	tableAngle
7	0	0	0	0	0	600	450	45
8	1	0	0	0	0	350	400	0
10	2	0	0	0	0	350	150	0
3	0	1	0	1	0	100	300	225
2	0	0	0	0	0	100	500	135
9	0	0	0	0	0	550	300	0
4	0	5172	1	5172	0	100	100	315
6	0	0	0	0	0	550	600	0
1	0	1	0	0	0	100	700	45
5	0	1	0	2	2	350	600	45

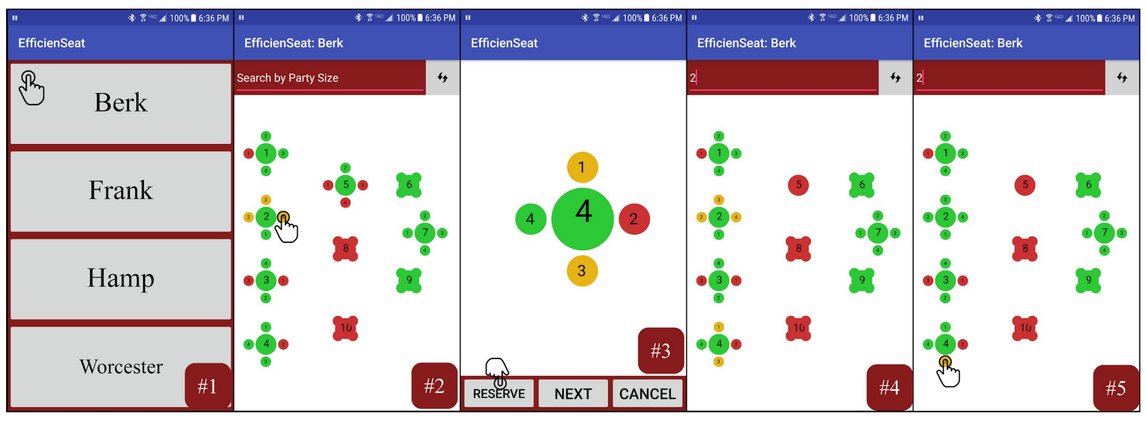
Mobile Application

Android Studio App

- Renders interactive dining hall map to user
  - Rendered through a purpose-built custom view class
  - Automatic display scaling for device compatibility
  - User touches translated to click coordinates on display
- Communicates safely with DynamoDB
  - Utilizes Amazon CognitoCredentials system for DB access
  - Reservations are made safely through conditional writing
    - If local and server data do not match, write is cancelled
  - Table data updated to match database...
    - On a timed interval
    - On user interaction including:
      - refresh button press
      - party size search
      - table reservation operations

Sample User-App Interaction

1. Select dining hall
2. Tap to select seats in map mode
3. Reserve seats by party size search
4. Return to map mode and review reserved seating
5. Tap reserved seats to unreserve manually



Overall Cost

ITEM	QTY	PER UNIT	ITEM	QTY	PER UNIT
Table Unit			**Sky Unit**		
SamR21 Microcontroller	1	4.73	Xplained Pro SamR21	1	60
PCB	1	20	Raspberry Pi 3	1	35
SMD Components	50	25	Wires	3	0.2
Pushbuttons w/ LED	4	6	Pi 3 Camera	1	28
IRLED	4	2.8	850 nm IR Filter	1	10
Wires	26	0.2	Pi 3 Heat Sinks	3	7
Ribbon Cable	1	2.5	Power Adpater	2	15
Battery Holders	2	5.3	Case	58g	1.5
Batteries	4	32	**Total (Sky Unit)**		**156.7**
Case	171g	3.1	**Grand Total**		**258.33**
Total (Table Unit)		**101.63**	AWS	<1req/sec	Free