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

Abstract
EfficienSeat 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

Database Server Task
- Manage seat information

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

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

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.

Specifications

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

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

Database

**Database storage includes:**
- ID and seat status
- Location
- Orientation

**Capabilities:**
- Conscientious data access
- Auto scaling capability

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
- 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

<table>
<thead>
<tr>
<th>ITEM</th>
<th>QTY</th>
<th>PER UNIT</th>
<th>ITEM</th>
<th>QTY</th>
<th>PER UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Unit</td>
<td>1</td>
<td>4.73</td>
<td>Sky Unit</td>
<td>1</td>
<td>60</td>
</tr>
<tr>
<td>SamR21 Microcontroller</td>
<td>1</td>
<td>2.70</td>
<td>Raspberry Pi 3</td>
<td>1</td>
<td>35</td>
</tr>
<tr>
<td>PCB</td>
<td>1</td>
<td>20</td>
<td>Pi3 Camera</td>
<td>1</td>
<td>35</td>
</tr>
<tr>
<td>SMD Components</td>
<td>50</td>
<td>25</td>
<td>Pi3 Heat Sinks</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Pushbuttons w/ LED</td>
<td>4</td>
<td>6</td>
<td>190 nm IR Filter</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>IRLED</td>
<td>4</td>
<td>2.8</td>
<td>190 nm IR Filter</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Wires</td>
<td>26</td>
<td>0.2</td>
<td>Power Adapter</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>Ribbon Cable</td>
<td>1</td>
<td>2.5</td>
<td>Case</td>
<td>58g</td>
<td>1.5</td>
</tr>
<tr>
<td>Battery Holders</td>
<td>2</td>
<td>5.3</td>
<td>Case</td>
<td>58g</td>
<td>1.5</td>
</tr>
<tr>
<td>Batteries</td>
<td>4</td>
<td>32</td>
<td>Total (Sky Unit)</td>
<td>156.7</td>
<td></td>
</tr>
<tr>
<td>Case</td>
<td>174</td>
<td>3.1</td>
<td>Grand Total</td>
<td>258.33</td>
<td></td>
</tr>
<tr>
<td>Total (Table Unit)</td>
<td>101.63</td>
<td></td>
<td>AWS</td>
<td>&lt;1meg/sec</td>
<td></td>
</tr>
</tbody>
</table>