UMASS AMHERST

SDP 18: EfficienSeat

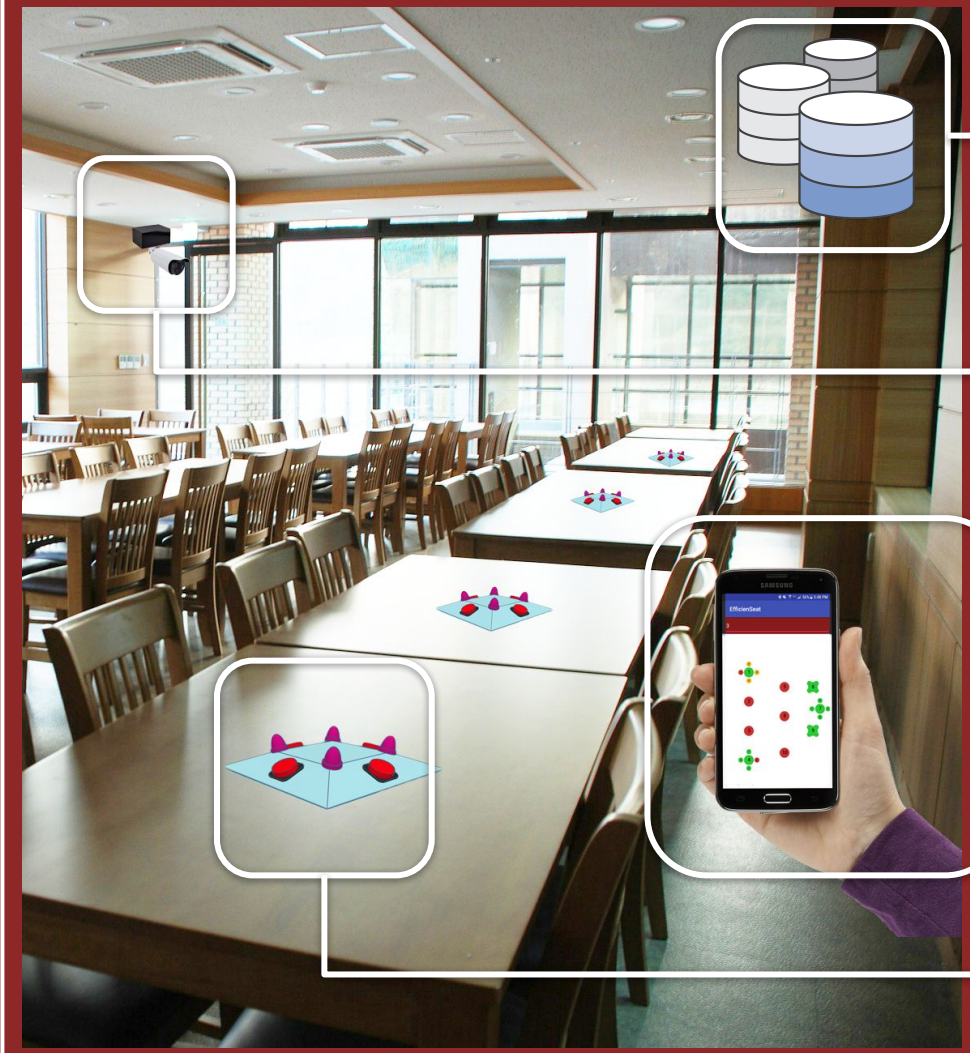
6 March 2018
Comprehensive Design Review

Team 26
Advised by Professor Tilman Wolf
Kristina Georgadarellis, Matthew Donnelly,
Dennis Donoghue, Aarsh Jain

Problem Statement

- Dining halls suffer due to seating inefficiencies
- A useful solution is convenient and saves time
  - Patrons need improved seat identification methods
  - Dining halls need an easily implementable and maintainable system

Our Vision for a New System



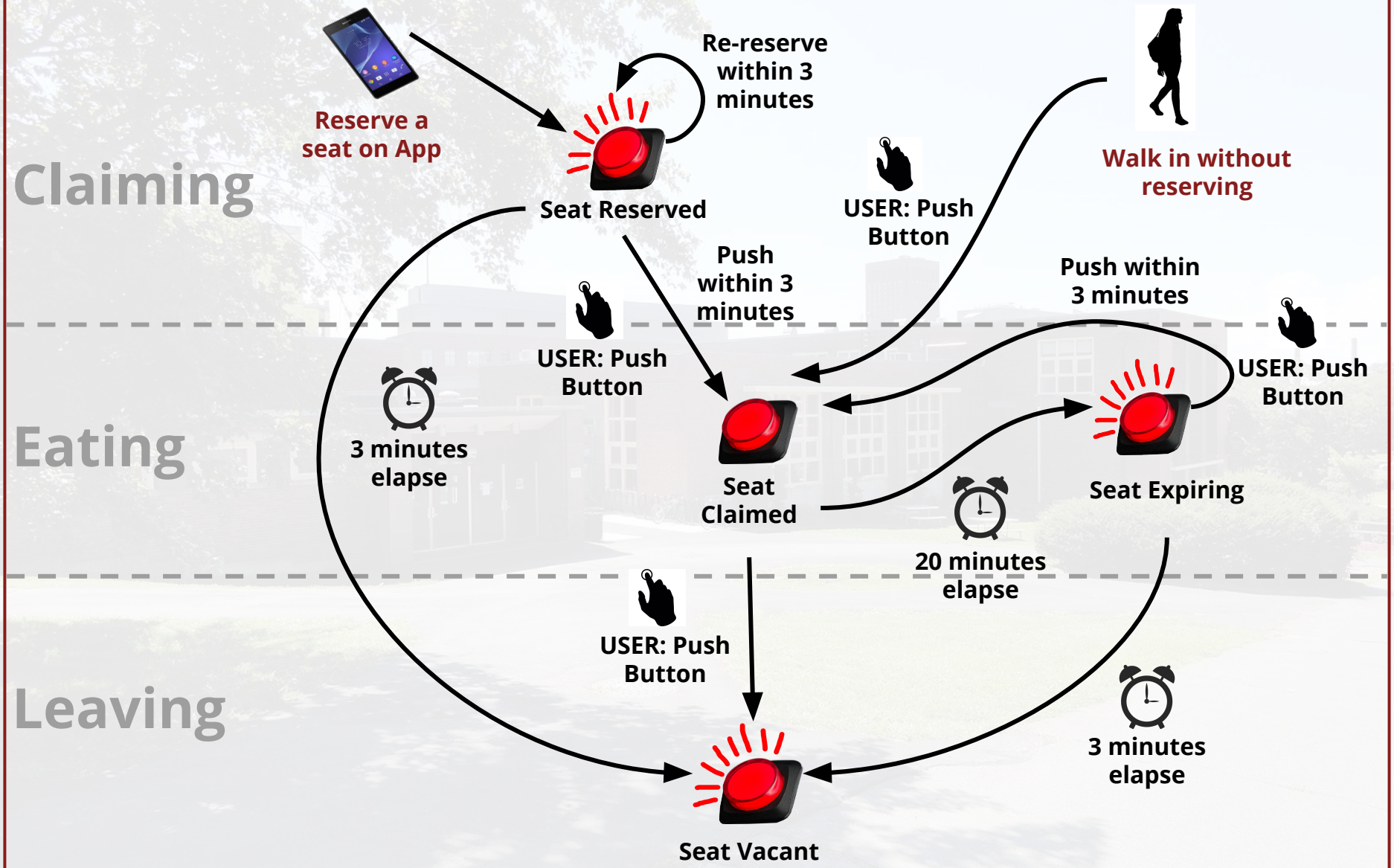
Database to act as hub for table seat data

Sky Unit for data relay and table positioning

App for mobile interaction with dining hall

Table Units with pushbuttons with four seats per table

User Interaction State Diagram



System Requirements

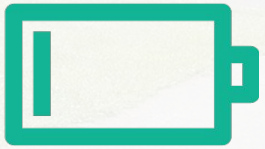
- Patrons should be able to:
  - Use an app to interact with dining hall
    - View a map of the dining hall
    - Make timed seat reservations
    - Search for seats by party size
  - Have choice of not using app
    - Table Units have physical user interface
- System implementation should be:
  - Easy to maintain
  - Safe around food

System Specifications

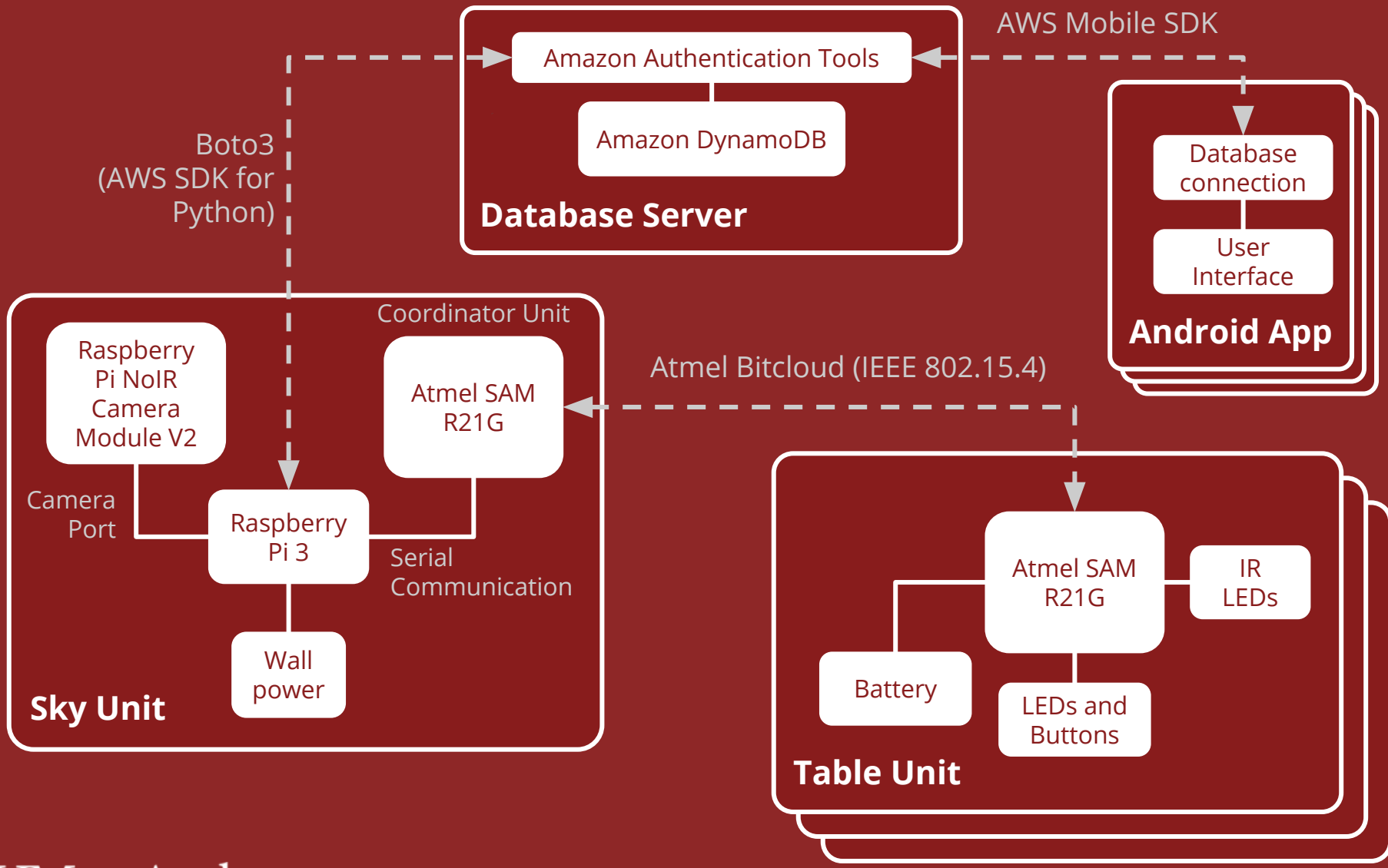
- Allows app users to search for available seats by party size through app
- Users will receive response from the app within 2 seconds
- **The app will generate a depiction of table locations accurate to ½ a table length**
- **Orientation of Table Unit detected within 90 degrees**
- Can function in a dining hall of several hundred seats
- **Table Unit is compliant with IPX4 (Splashproof)**
- Table Units should have a push button interface
- **Battery life of Table Unit should last a month**

Challenges

1. System Communication
   → Interfacing between four mediums
2. Data Synchronization
   → Table data agreement across system
3. Table Localization
   → Indoor positioning problem to find tables
4. Low power, Unobtrusive Nodal Units
   → High-functioning with low power costs



Block Diagram



CDR Deliverables

- Demonstrate full system communication
  - Hitting button on the Table Unit is reflected on the app
  - Reserving tables from app is reflected on the Table Unit
  - Conscientious data writing from both Sky Unit and App
- Prototype of table unit
  - Actual hardware for buttons and LEDs, battery
  - Case prototype design for Table Unit
- Demonstrate table localization ability
  - Configure a grid for a rectangular area
  - Determine position of a table unit

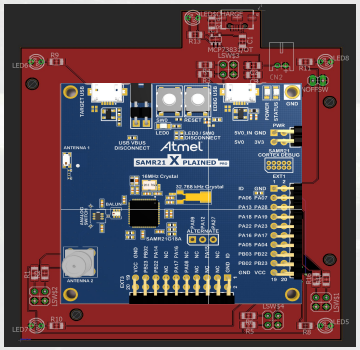
CDR Deliverables

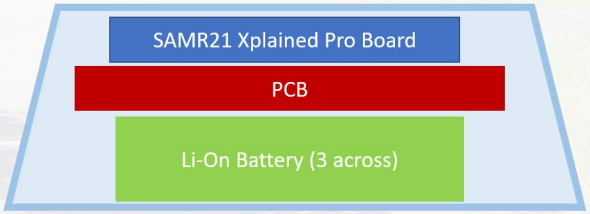
- Demonstrate full system communication
  - Hitting button on the Table Unit is reflected on the app
  - Reserving tables from app is reflected on the Table Unit
  - Conscientious data writing from Sky Unit and App
- **Prototype of table unit**
  - **Actual hardware for buttons and LEDs, battery**
  - **Case prototype design for Table Unit**
- Demonstrate table localization ability
  - Configure a grid for a rectangular area
  - Determine position of a table unit

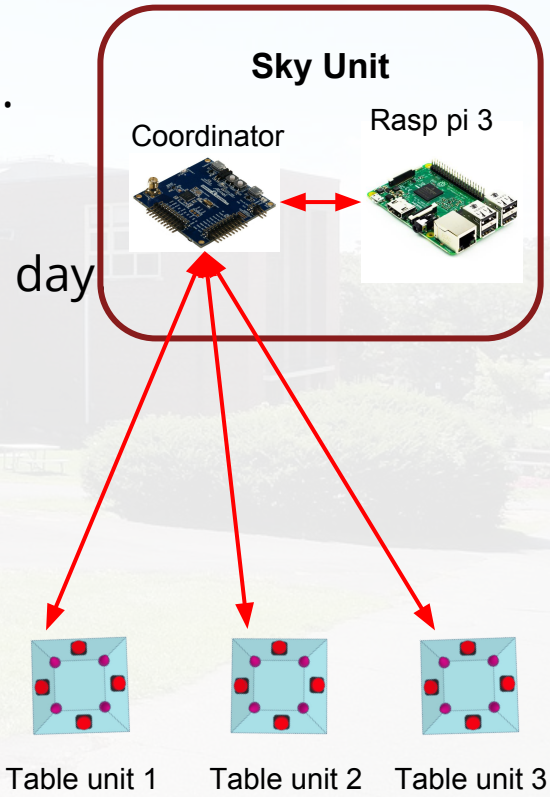
Low Power, Unobtrusive Modular Units

Table unit features:

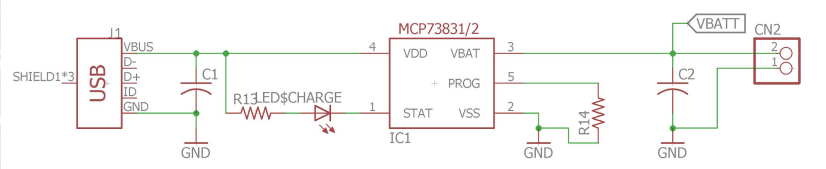
- Low powered chip with SoC for communication.
- Communication over IEEE 802.15.4 built on LwMesh.
- Powered using three 3400mAh batteries.
- Battery lasts up to 25 days with 14 hrs working each day
- Compact and modular design.



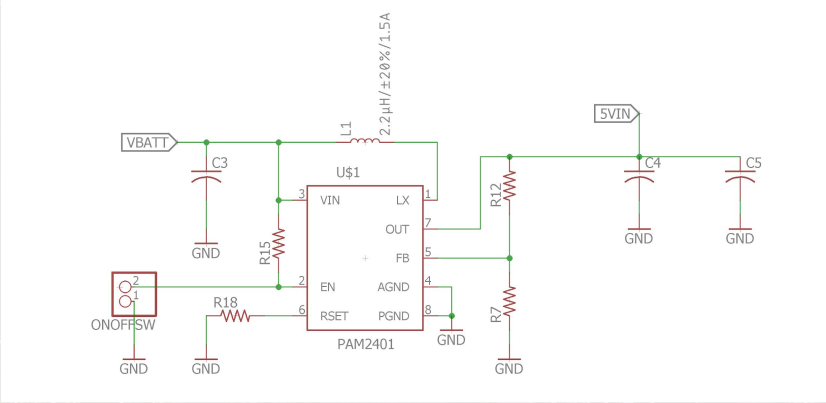




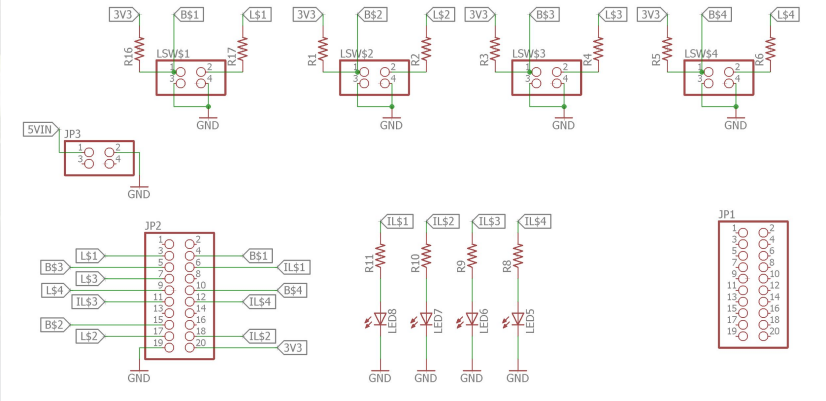
PCB Design



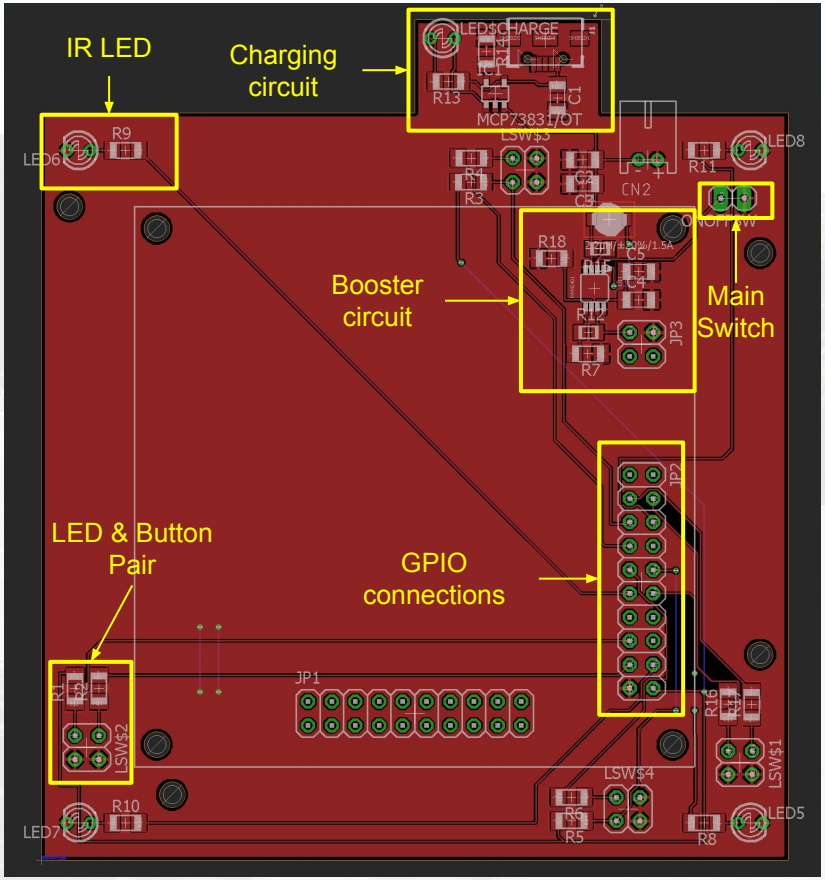
Charging Circuit



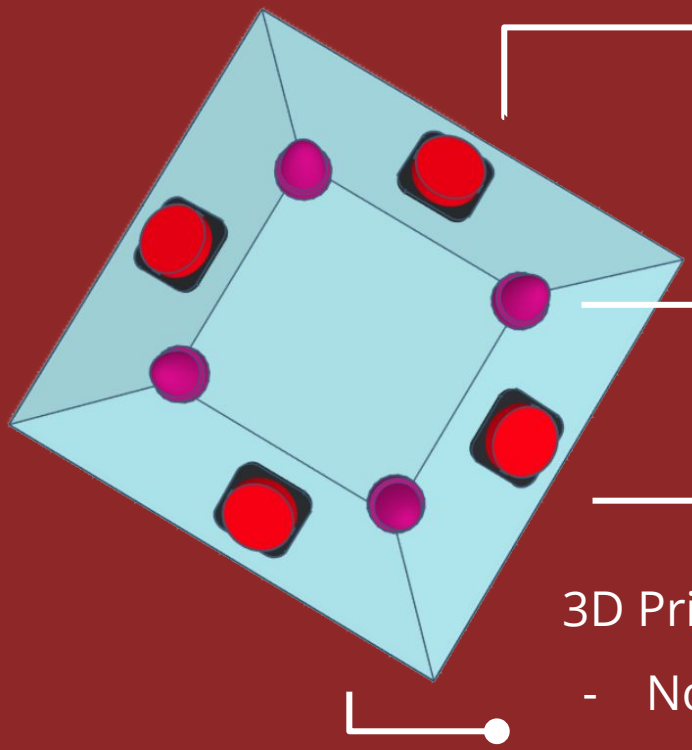
Booster Circuit



GPIO, Power and LED connections



Prototype of Case Design



LED Pushbutton

10mm IR LED

Off switch for non-operation

3D Printed Case

- No gaps between case and components
- Surface sanded and sealed to prevent bacteria growth
- Compact design - 10cm x 10cm

CDR Deliverables

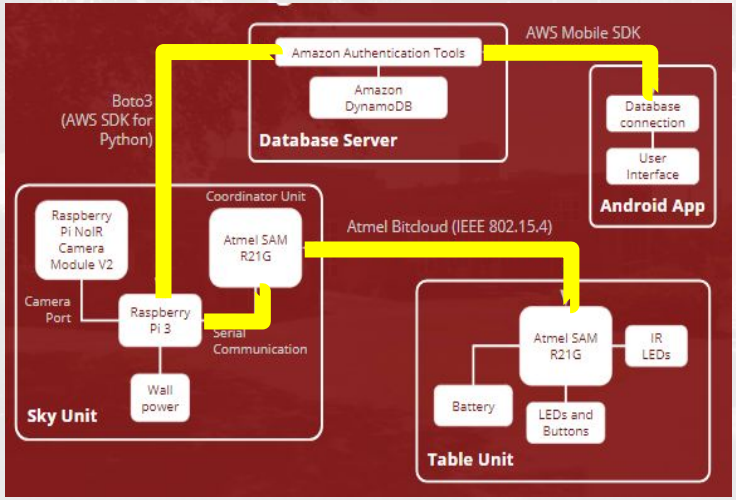
- Demonstrate full system communication
  - Hitting button on the Table Unit is reflected on the app
  - Reserving tables from app is reflected on the Table Unit
  - **Conscientious data writing from Sky Unit and App**
- Prototype of table unit
  - Actual hardware for buttons and LEDs, battery
  - Case prototype design for Table Unit
- Demonstrate table localization ability
  - Configure a grid for a rectangular area
  - Determine position of a table unit

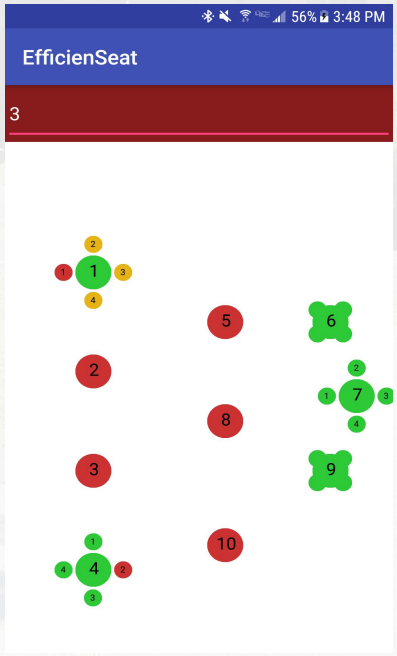
System Communication and Data Agreement

Challenge: Smooth system integration and state preservation

- Seat states must be reliably communicated throughout system
- States must be safely modified from two different systems (App, Table Unit)

Solution: Consistent conditional writing and efficient protocol





Locating Tables in the Room

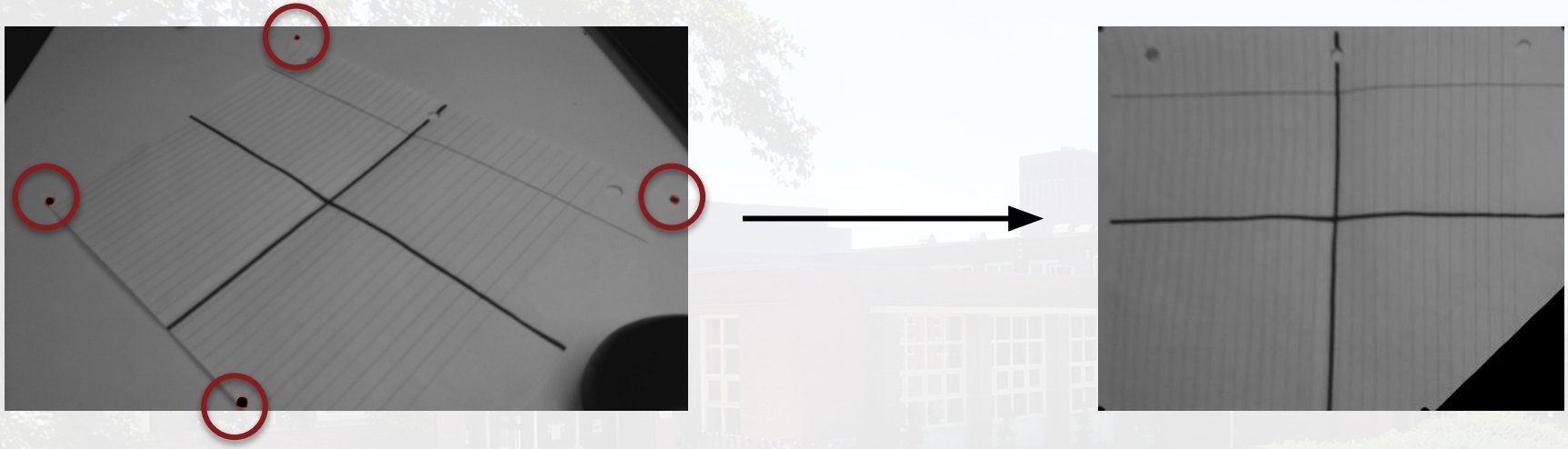
Challenge: Simple way to find table locations

- Accuracy: up to half a table length
- Speed: update every hour

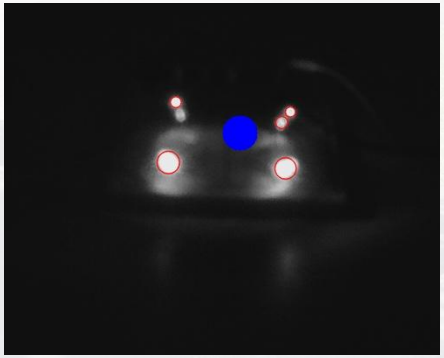
Solution: Use IR LEDs and camera to find the positions

The Process for Finding Tables

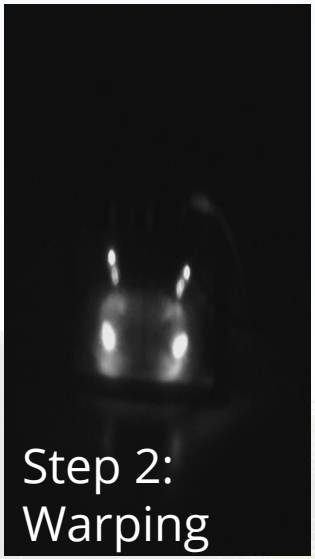
Step 1: Calibration - getting a “top down” view of room

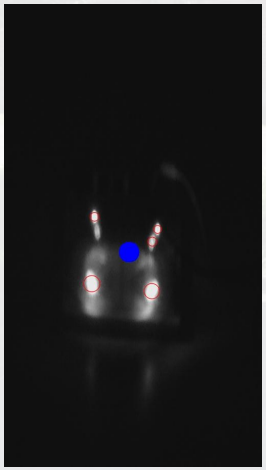


Step 2: Blob detection - find Table Units by IR LEDS one at a time



The Process for Finding Tables





Final Step: Blob Detection

Results of Test

Actual Position	Calculated Position
(20cm, 45.5cm)	(22cm, 48cm)

CDR Deliverables

- **Demonstrate full system communication**
  - **Hitting button on the Table Unit is reflected on the app**
  - **Reserving tables from app is reflected on the Table Unit**
  - Conscientious data writing from Sky Unit and App
- Prototype of table unit
  - Actual hardware for buttons and LEDs, battery
  - Case prototype design for Table Unit
- Demonstrate table localization ability
  - Configure a grid for a rectangular area
  - Determine position of a table unit

App reflected on Table Unit

Claiming

Eating

Reserve a seat on App

Re-reserve within 3 minutes

Seat Reserved

USER: Push Button

Walk in without reserving

Push within 3 minutes

USER: Push Button

Push within 3 minutes

USER: Push Button

3 minutes elapse

Seat Claimed

Seat Expiring

20 minutes elapse

USER: Push Button

3 minutes elapse

Seat Vacant

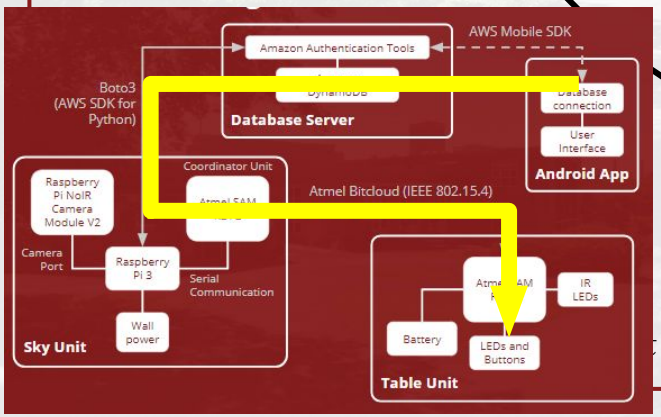


Table Unit reflected on App

Claiming

Eating

Reserve a seat on App

Re-reserve within 3 minutes

Seat Reserved

USER: Push Button

Walk in without reserving

Push within 3 minutes

USER: Push Button

Push within 3 minutes

USER: Push Button

3 minutes elapse

Seat Claimed

Seat Expiring

20 minutes elapse

USER: Push Button

3 minutes elapse

Seat Vacant

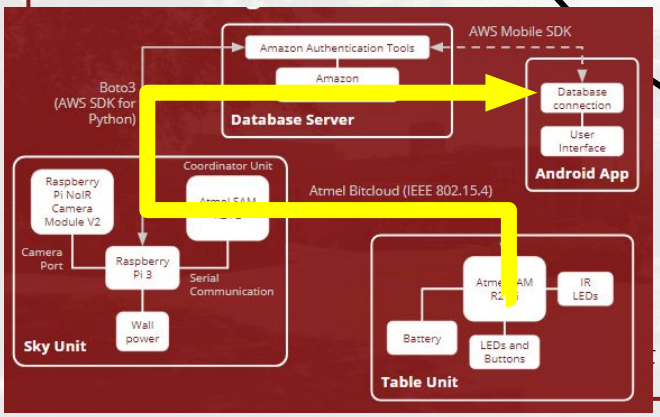
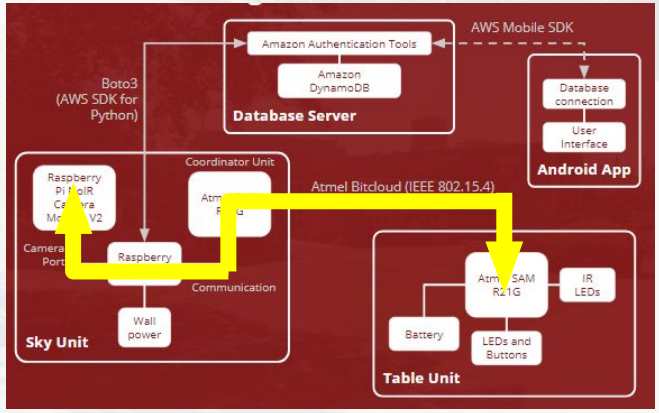


Table Localization Demonstration

Purpose: Find position of table given the length and width of a room

Demonstration Steps:

- Calibrated for marked area before presentation
- Run script to find coordinates of Point A
  - Display final result and coordinates
- Move IR LEDs to Point B and run script again



CDR Deliverables

- Demonstrate full system communication
  - Hitting button on the Table Unit is reflected on the app
  - Reserving tables from app is reflected on the Table Unit
  - Conscientious data writing from Sky Unit and App
- Prototype of table unit
  - Actual hardware for buttons and LEDs, battery
  - Case prototype design for Table Unit
- Demonstrate table localization ability
  - Configure a grid for a rectangular area
  - Determine position of a table unit

FPR Deliverables

- Deployable Table Unit
  - Case, PCB built and fully integrated
  - Case protects Table Unit from food/spills
- Demonstrate complete Table Localization ability
  - Improve algorithm for large scale implementation
- Complete and robust system operation
  - All parts fully integrated and working as one
  - Complete user App experience
  - Reservation and claim timers implemented

Questions?

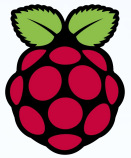
Questions to Consider I

- Power concerns - current power consumption
  - What we're going to do to reduce it
  - Quick analysis? Take power reading and calculate
- Timing coordination
  - Where will the timing take place?
- Session IDs
- Authentication and Confidentiality
- LED ON/OFF situation
  - The workers will shut system down after operation hours
  - 9-3, 6-10 7-12 it's busy 60% of the time OFF is taken final answer - it's an a e s t h e t i c choice, easier to see a lit up LED
  - Take google plot for evidence

Questions to Consider II

- Orientation of table unit
  - Know which LED is on
- Low battery indication, error
- Single reserve tables vs multi reserved tables?
  - If yes, how are we handling it?

Amazon DynamoDB



Raspberry Pi on Sky Unit

PCB Design

The following features are included in the PCB:

- Li-ion/Li-po charging circuit through micro USB.
- Booster circuit to provide board with 5V.
- LED and Button pair on each corner for buttons and status LEDs.
- IR LEDs on diagonal corners for table localization.
- Easy to mount under SAM R21.
- Charging status LED.
- ON/OFF slider switch.

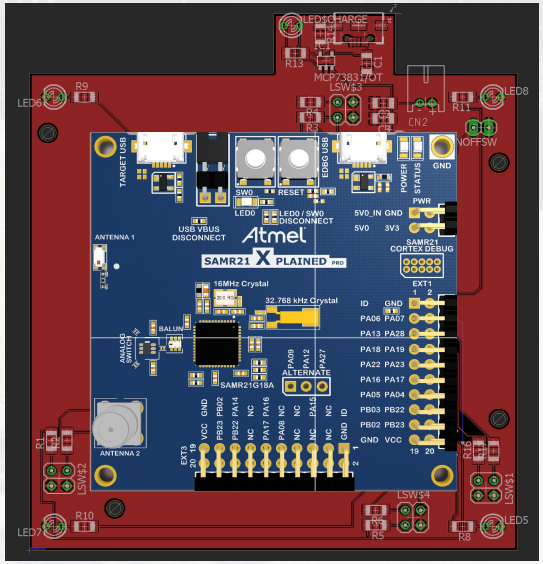


Table Unit

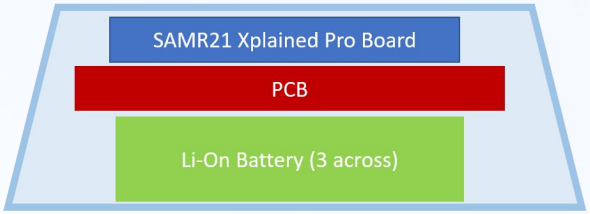
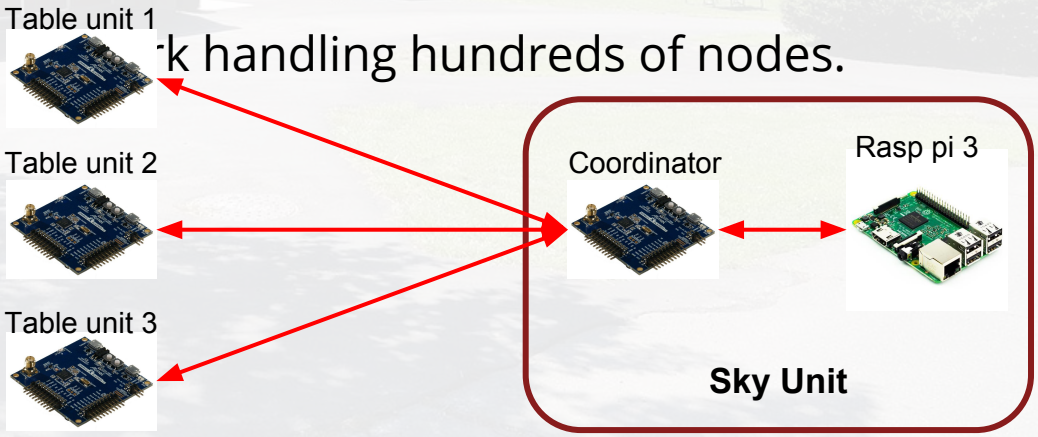


Table unit features:

- Low powered chip with SoC for communication.
- Communication over IEEE 802.15.4 built on LwMesh.
- System powered through three 3400mAh batteries.
- Battery capacity upto 25 days with 14 hrs working.
- Charging and boosting features included.
- Compact and modular design.
- Robust [illegible]k handling hundreds of nodes.



Timeline for User Interaction MOVE

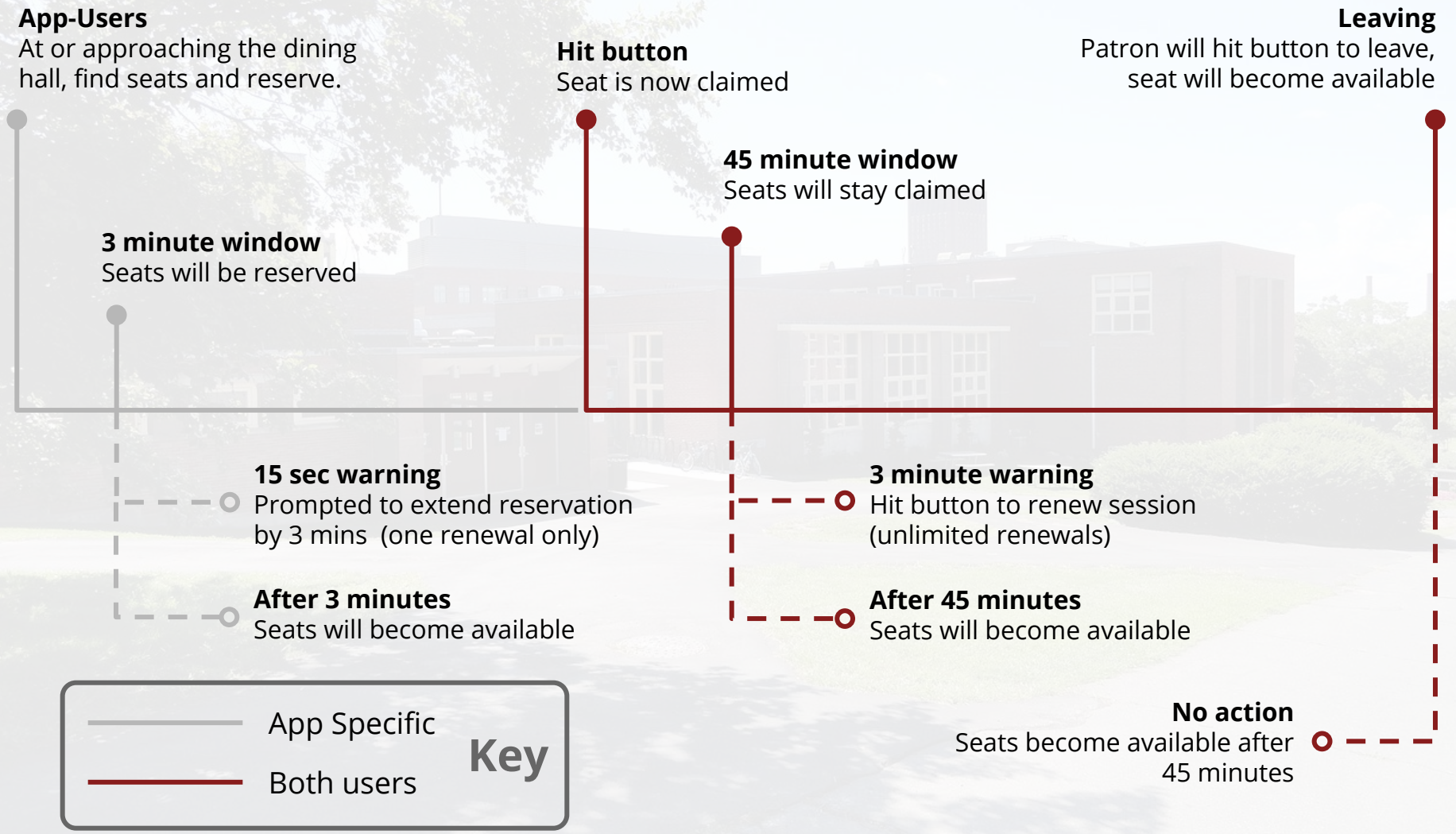
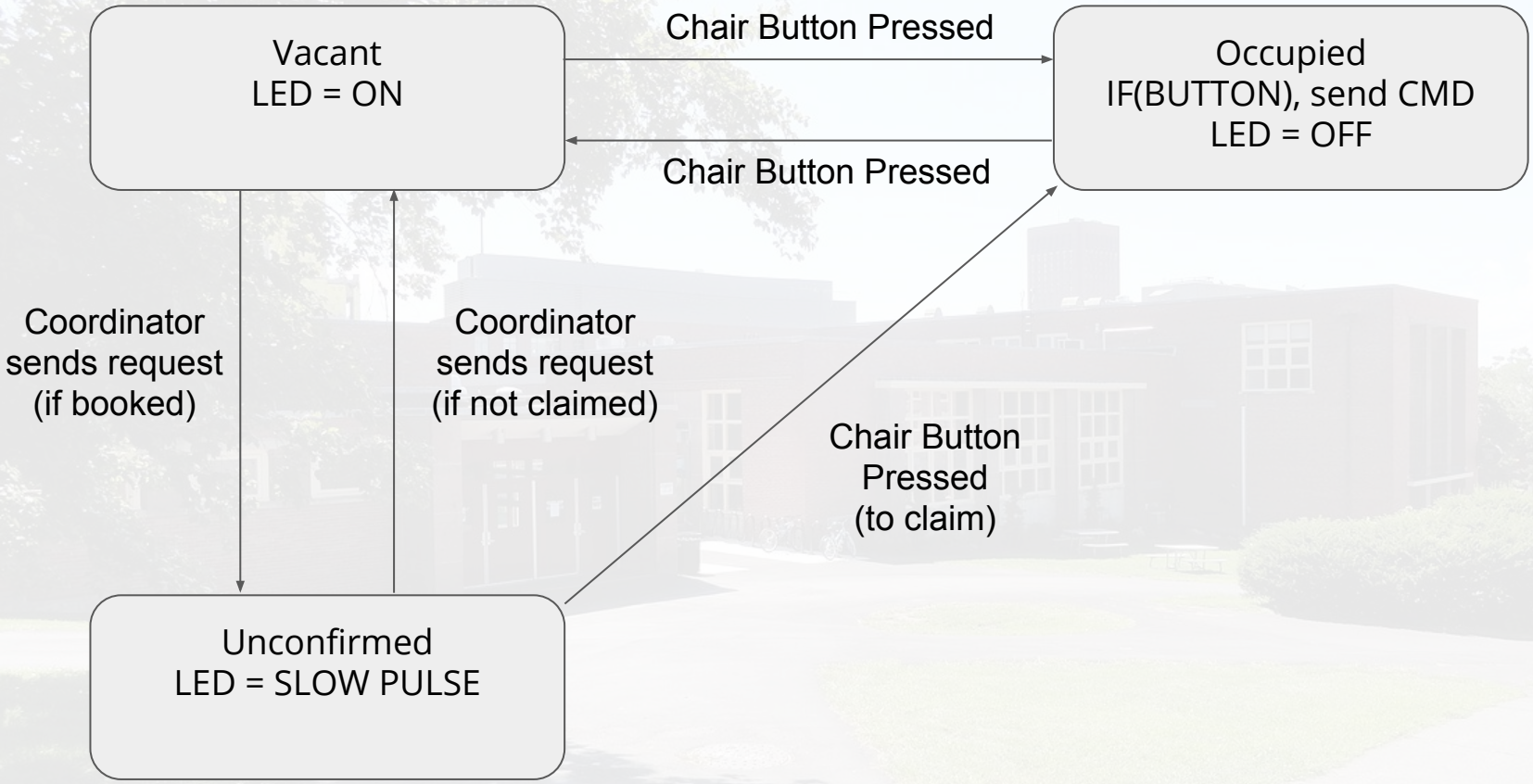


Table Unit: State Diagram



Table/Sky Unit Communication Protocol

Payload

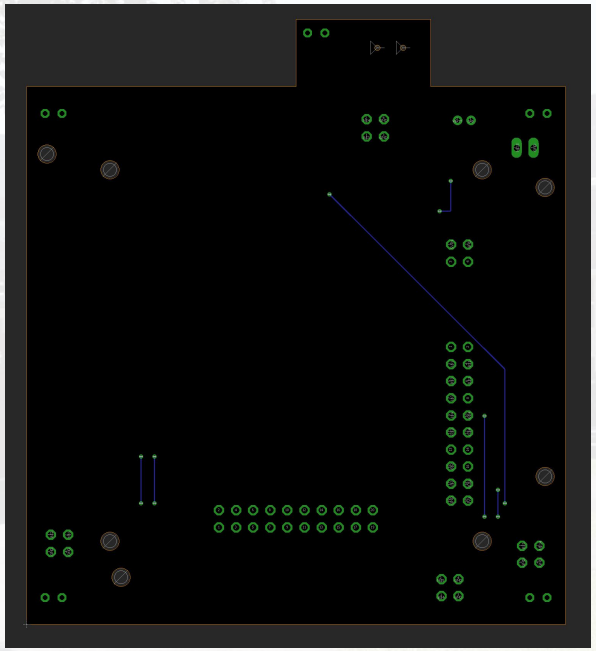
2 bytes	1 byte	4 byte
addr	**cmd**	**light state**

Definitions

ID	**Command**
0	Invalid
1	Change seat state
2	Change IR LED state
3	Wake up

ID	**Seat/IR State**
0	Vacant/on
1	Occupied/off
2	Reserved

Bottom Layer



Nominal User Interaction

Heading to dining Hall

Reserve a seat on App

Seat Reserved

3 min one-time refresh

3 mins to Confirm Reservation

USER: Push Button

At the Dining Hall

USER: Push Button

Unlimited refreshes

Seat Claimed or Seat Confirmed (if using app)

45 Minutes

Seat Reserved

Leaving Dining Hall

USER: Push Button

Seat Vacant