

UMASS  
AMHERST

# SDP 18: EfficienSeat

Preliminary Design Review  
18 October 2017

**Team 26**

**Advised by Professor Tilman Wolf**

Kristina Georgadarellis, Matthew Donnelly,  
Dennis Donoghue, Aarsh Jain

# Background Information

- Dining halls have inefficient “first come, first served” seating - problem during busy times
- Patrons must commit and pay for entry before knowing state of seating
- Leads to congested walkways with patrons spending unnecessary time searching for seats
- Wastes patron’s time and lowers efficiency of dining hall staff that need to use walkways



# Assess Needs

## Patron Needs

- Be able to efficiently locate seating at dining halls
- Be able to know seating status of entire dining hall at a glance

## Dining Hall Needs

- Adaptation to system should be simple and painless
- Be able to implement a cost effective, easy to maintain system

# Our Solution: An Overview

## Three main components

- Camera system with imaging and thermal sensing mounted on ceiling
- Small interactive unit on the table with which patrons interact
- Phone app

## Three clients to keep in mind

- Patrons who use our app (referred to as Users)
- Patrons who do not (referred to as Non-Users)
- Dining hall staff

# Our Solution: Alternatives

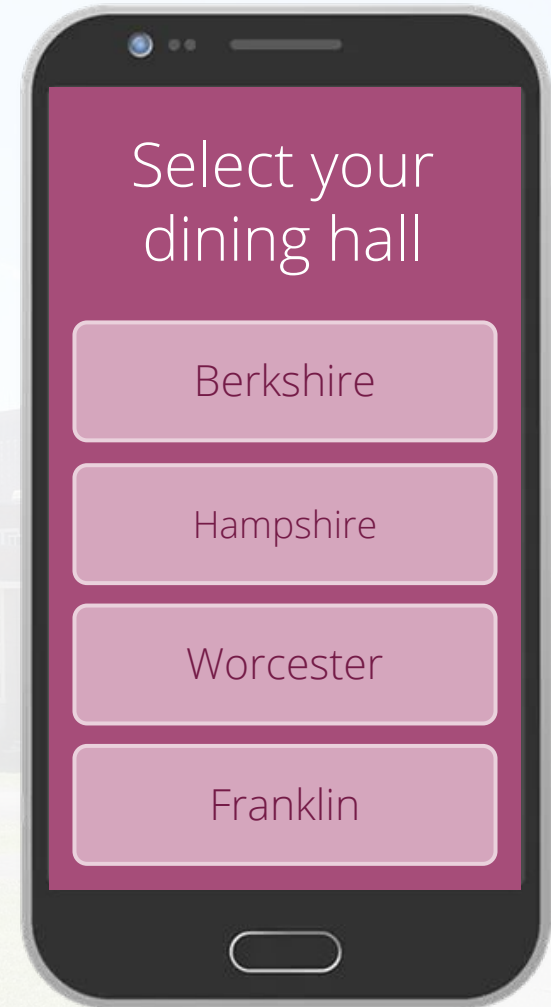
- Passive indicators to replace the table unit like IR reflector strips
- Different options for the camera like infrared, live imaging
- Restaurant reservation system
- Sensing sections of the dining hall rather than seats
- App access from home



# Our Solution: Scenarios

## Let's walk through a scenario

- Four friends want to go to dinner
- Open the app and select the dining hall



# Our Solution: Scenarios

## Let's walk through a scenario

- Four friends want to go to dinner
- Open the app and select the dining hall

## At the Dining Hall

- Users can find seats based on party size
- Camera keeps track of empty seats
- Button system on table assists Users and Non-Users



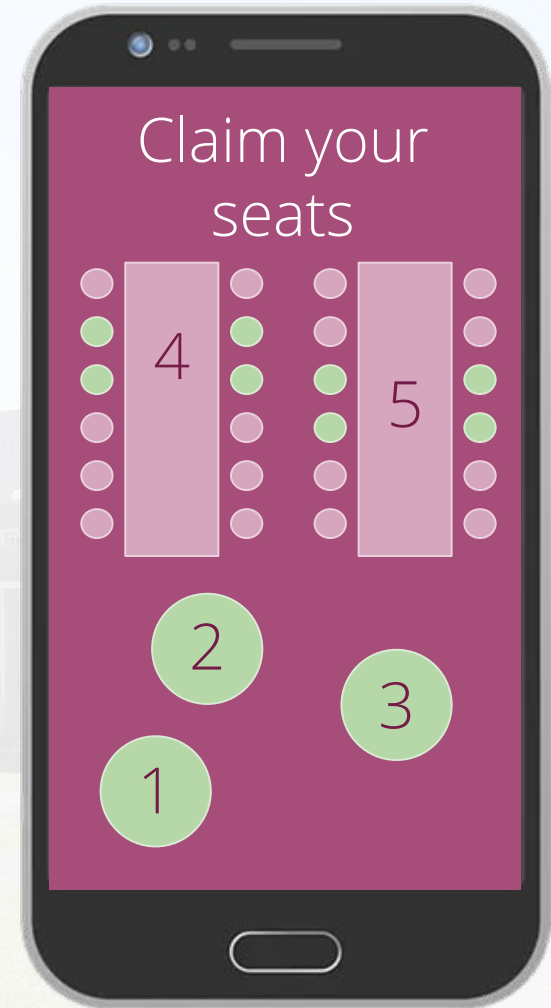
# Our Solution: Scenarios

## Getting Seated

- App finds list of available seats and tables according to party size
- Once selected, the seats are reserved for 2 minutes
- Go to the seat and press the button to claim

## If no seats are available

- Prompted to add request to queue
- A notification pops up when seats ready





# Our Solution: Scenarios

## Seating Priority System

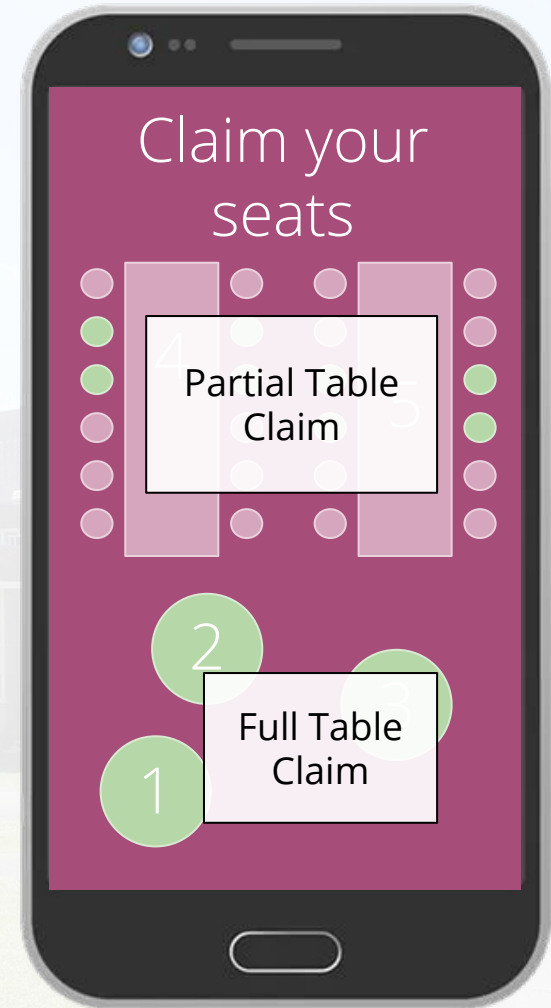
- Two types of table claiming: Full and Partial
- Big parties have full table priority over single sitters
- Non-users alter table priority

## Reporting Errors

- Available seat or table was actually taken

## Non-User Interaction

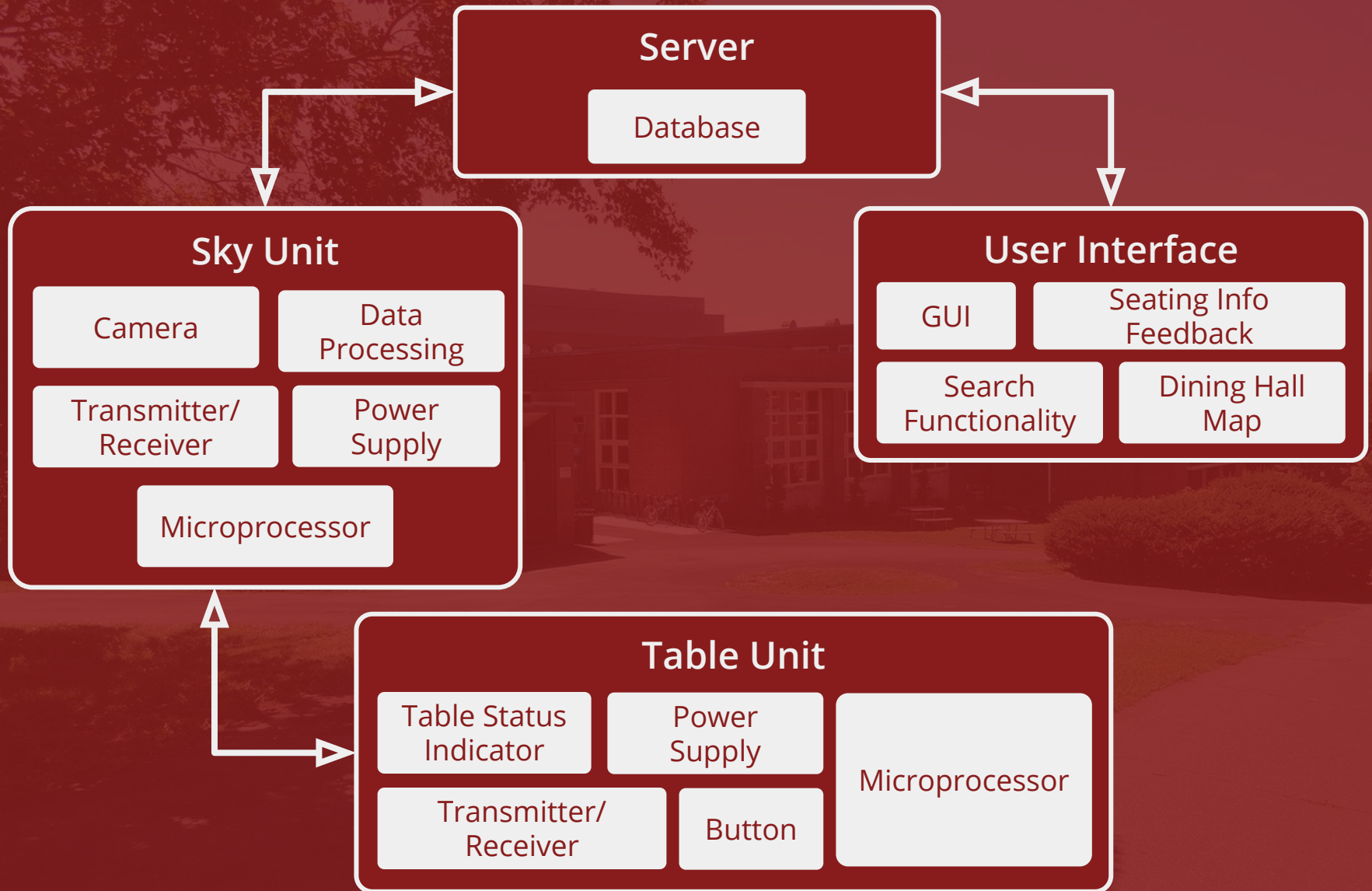
- LED on table unit indicates table status
- Only sit at tables that indicate “available”
- Encouraged to push button to claim table

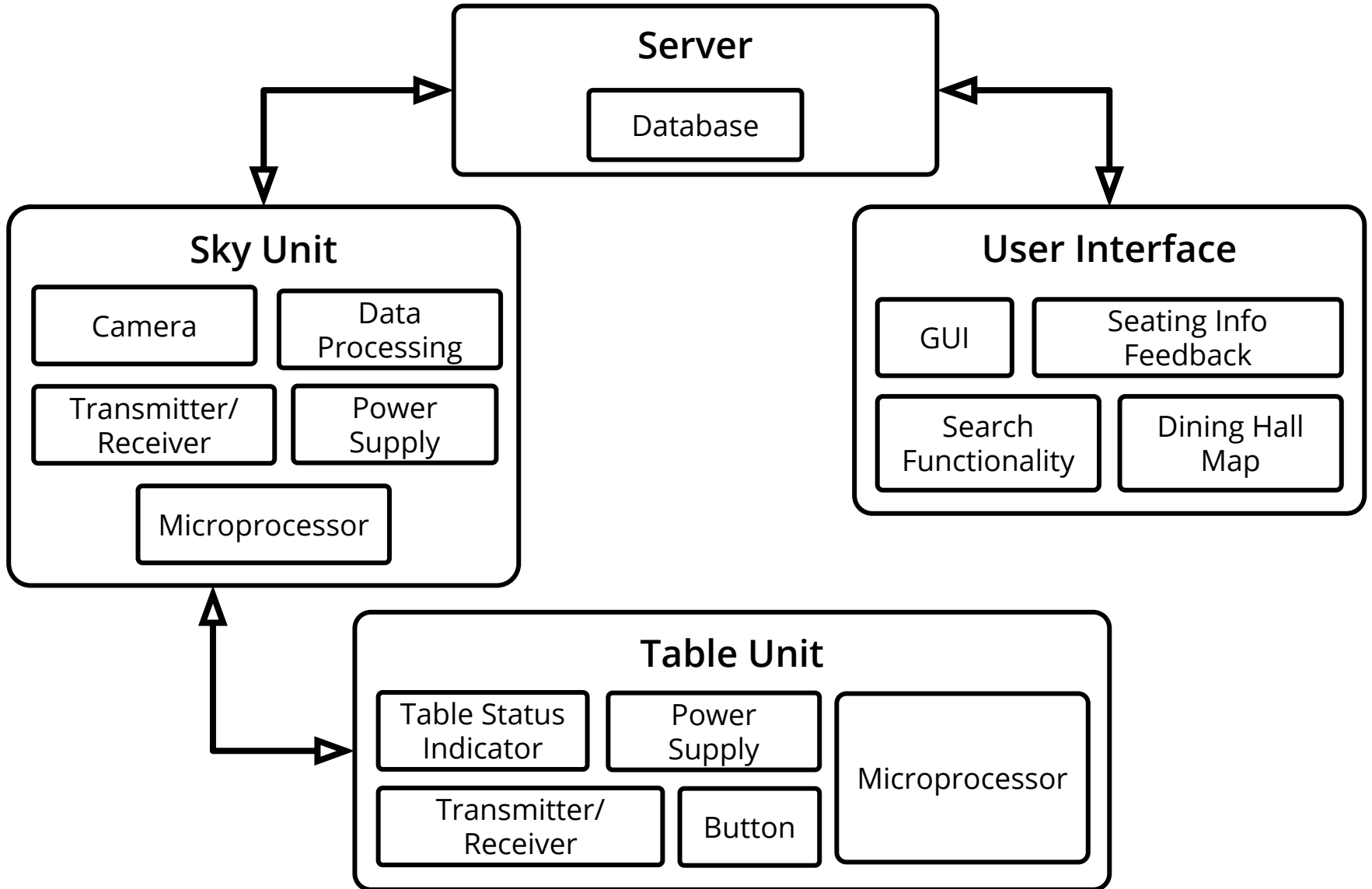


# System Specifications

- Allows Users to search for available seats by party size through app
- Users will receive response from app within 2 seconds
- For parties up to the size of the largest table, achieve 90% success in finding seats correctly if available
- Function in dining rooms of several hundred seats
- Table unit is spill-proof and low maintenance
- System accommodates Non-Users

# Block Diagram





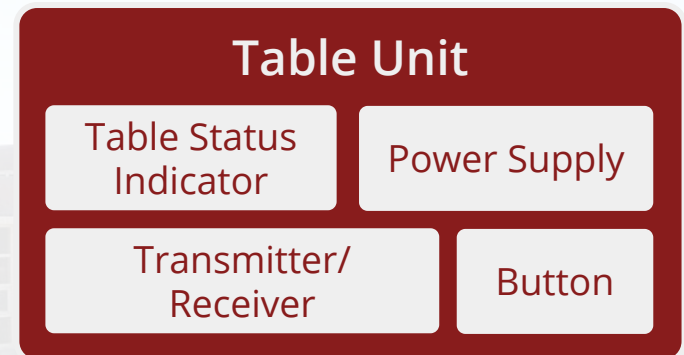
# Subsystems: Table Unit

## Requirements

- Must be low power
- Accommodate both Users and Non-Users
- Communicate with sky unit
- “Dining Hall” resistant

## Implementation

- LED push button to indicate table status
- Solar/Rechargeable battery for power supply
- RF receiver and transmitter



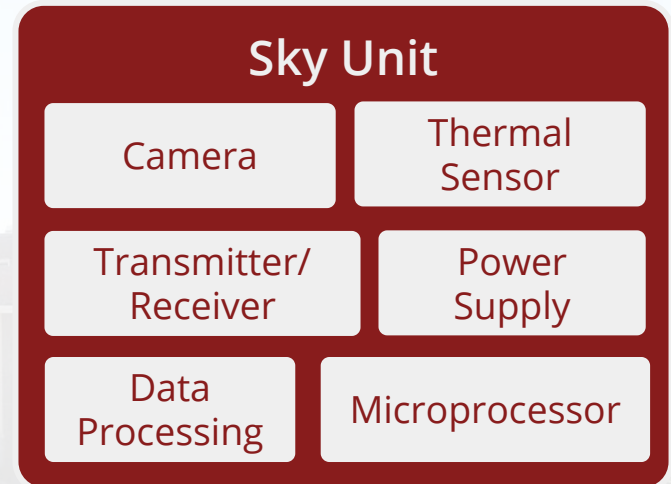
# Subsystems: Sky Unit

## Requirements

- Encompass ~100 seats
- Easy to install in dining hall
- Accurately identify empty seats
- Communication with table unit/server
- Pre-process camera data

## Implementation

- Visual and thermal cameras
- Uses wall power
- RF and WiFi communication
- Microprocessor



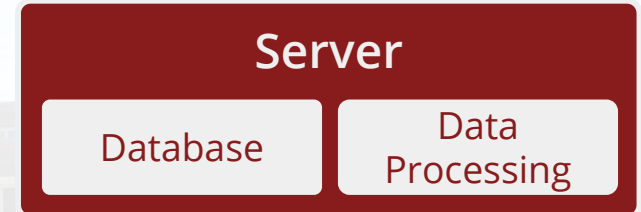
# Subsystems: Server

## Requirements

- Capable of storing data and handling App requests
- Handle 100+ concurrent requests
- Assist in processing of Sky Unit data

## Implementation

- Implemented on scalable rented Amazon Servers
- Possible to split into a database and application server



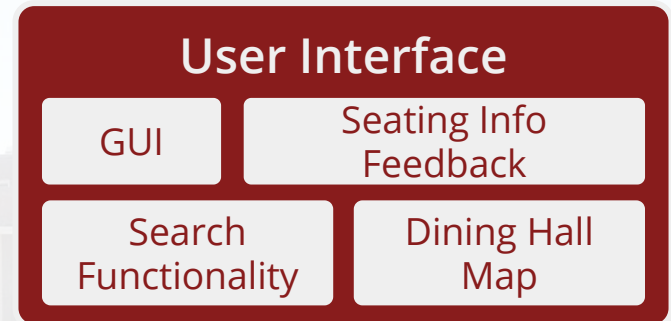
# Subsystems: User Interface

## Requirements

- Takes less than 2 seconds to load or refresh
- Quick response time
- Minimalistic layout for ease of use

## Implementation

- Android platform
- Android Studio
- Java





# MDR Deliverables

## Table Unit Power Supply

- Demonstrate a power supply that can support the table unit over the course of its daily operation

## User Interface

- Demonstrate that the app can run a table search and display results

## Empty Seat Analysis

- Demonstrate that the camera system can detect seated patrons and objects

## Table Unit

- Demonstrate that the table unit can cycle through all states

# Team Responsibilities

## Dennis

- App, Server, search algorithm, and sky unit data processing

## Aarsh

- Hardware for table unit (LED button, power supply)

## Matt

- Hardware for camera unit (Visual camera, thermal camera, microprocessor)

## Kristina

- Sky unit data processing

**Questions?**



# Back-up Slides

# State Machine - User Search

