Automated Cat Feeder

Design Review
Matt Smith, Chris Ranc, Shabab Siddiq
1. The user should be able to wirelessly connect to the Automated Cat Feeder system.
2. The user should be able to add and remove pets from the feeder system.
3. The user should be able to assign a feeder to a specific pet.
4. The user should be able to modify a pet’s feeding time and food amounts.
5. The user should be able to monitor a pet’s overall dietary habits.
6. The overall ability to feed one to several animals independently.
7. The capacity for a multi-day food supply.
8. The feeders should only dispense dry food.
9. The device should be capable of long term reliability for continued operation.
10. The device should have a level of dispensing precision to work with different animals.
11. The app should be able to track and analyze eating behavior.
12. The device can easily be set up wirelessly.
13. Multiple devices should be easily set up for scalability.
<table>
<thead>
<tr>
<th>Marketing Requirements</th>
<th>Engineering Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,5</td>
<td>A. Software to keep track of multiple animals and differentiate behavior based on animals being served</td>
</tr>
<tr>
<td>2</td>
<td>B. Database to store data for mobile app that presents in human readable format</td>
</tr>
<tr>
<td>1</td>
<td>C. Wireless operation (aside from power) to prevent curious animals from chewing on wires</td>
</tr>
<tr>
<td>3</td>
<td>D. Ability to wirelessly communicate to supplementary devices that act as simple feeders while the master unit manages smart features</td>
</tr>
<tr>
<td>1,12</td>
<td>E. Supplementary and master feeders respectively need a controller with wireless peripherals or peripherals to integrate with a wireless module</td>
</tr>
<tr>
<td>7</td>
<td>F. High capacity food container of at least 0.5 gallons</td>
</tr>
<tr>
<td>10</td>
<td>G. Precise dispensing control. Dispensing depends on food size, ranging from small to large kibble.</td>
</tr>
<tr>
<td>9</td>
<td>H. Long-term reliability for automatic feeding for at least 1 week</td>
</tr>
</tbody>
</table>
Concepts

- Feeder mechanism
- User interfaces
  - Web or Smartphone application
- Controller Configurations
Concepts: Considered Feeding Mechanics

Corkscrew Dispenser
Concepts: Considered User Interfaces

Web App

Mobile App
Concepts: Considered Controller Configurations

Master Controller

Supplementary Controller
Design Overview

- ESP Stack
- Pi Stack
- Android Stack
- Physical Design
  - Motors
  - 3D printing
  - Design Exploded View
  - Control Electronics
  - Identification components (RFID)
ESP-8266 Stack

- ESP12S development board
- Micropython
- Picoweb web framework--modified
- i2C compatibility for peripheral driving
ESP-8266 Stack

Network Devices: -Master Feeder

Picoweb Server Modified

Btree Storage

Electronics: -motor driver -RFID reader
Raspberry Pi ZeroW

- Built in wireless functionality
- Raspbian lite
- Simultaneous access point and managed connection
  - using HostAPD and DNSMasq for networking
- Bare-bones HTTP server + local application server
- MongoDB
- i2C compatibility for peripheral driving
Raspberry Pi ZeroW

Network Devices:
- Smartphone
- Secondary feeder

HTTP Server

Application Server

MongoDB

Electronics:
- Motor driver
- RFID reader
Android Stack

- Kotlin used for app development.
- User-friendly user interface for managing pet data and feeding schedules.
- Chosen due to minimal developer costs (it’s free!) and availability of test devices.
- Essentially configures the feeder.
Physical design

- Auger dispenser
- Integrated RFID reader in practical location
- Removable bowl and hopper for easy cleaning
- Strong conduit to protect power cable
- 3D printed chassis for rapid prototyping
Physical Design
# Gantt Chart

<table>
<thead>
<tr>
<th>Task</th>
<th>Start</th>
<th>End</th>
<th>Duration</th>
<th>% Done</th>
<th>Work Days</th>
<th>Days Left</th>
<th>Days Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>100%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.2</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>75%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.3</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>50%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.4</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>25%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.5</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>0%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.1</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>100%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.2</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>75%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.3</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>50%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.4</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>25%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.5</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>0%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3.1</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>100%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3.2</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>75%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3.3</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>50%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3.4</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>25%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3.5</td>
<td>Mon 11/13</td>
<td>Thu 11/17</td>
<td>5</td>
<td>0%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.1</td>
<td>Tue 11/13</td>
<td>Fri 11/17</td>
<td>5</td>
<td>100%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.2</td>
<td>Tue 11/13</td>
<td>Fri 11/17</td>
<td>5</td>
<td>75%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.3</td>
<td>Tue 11/13</td>
<td>Fri 11/17</td>
<td>5</td>
<td>50%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.4</td>
<td>Tue 11/13</td>
<td>Fri 11/17</td>
<td>5</td>
<td>25%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.5</td>
<td>Tue 11/13</td>
<td>Fri 11/17</td>
<td>5</td>
<td>0%</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Current Day**: Red
- **Available work days**: Blue
- **Days completed**: Grey
# Bill of Materials

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Cost ($)</th>
<th>Our Cost ($)</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android OS Device</td>
<td>200.00-500.00</td>
<td>0.00</td>
<td>Already Owned</td>
</tr>
<tr>
<td>Raspberry Pi Zero W</td>
<td>10.00</td>
<td>0.00</td>
<td>Cana Kit 3-5 business days</td>
</tr>
<tr>
<td>Micro SD Card</td>
<td>7.80</td>
<td>0.00</td>
<td>Amazon 2-day shipping</td>
</tr>
<tr>
<td>ESP8266 (Feather)</td>
<td>17.12</td>
<td>0.00</td>
<td>Amazon 2-day shipping</td>
</tr>
<tr>
<td>Stepper Motor</td>
<td>15.00-20.00</td>
<td>15.00-20.00</td>
<td>Amazon 2-day shipping</td>
</tr>
<tr>
<td>Drive belt</td>
<td>6.00-12.00</td>
<td>6.00-12.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>3D Chassis and drive components (including revisions)</td>
<td>40.00-50.00</td>
<td>40.00-50.00</td>
<td>RIT Printers (9AM-(PM))</td>
</tr>
<tr>
<td>MFRC522</td>
<td>12.00</td>
<td>12.00</td>
<td>Amazon 2-day shipping</td>
</tr>
<tr>
<td>Misc. connecting hardware</td>
<td>10.00-40.00</td>
<td>10.00-40.00</td>
<td>Amazon</td>
</tr>
<tr>
<td><strong>Total Cost</strong></td>
<td><strong>83.00-134.00</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Risks

- **Hardware (Physical)**
  - Multiple units increases complication
    - Reuse as many components as possible to simplify the design.

- **Hardware (Electronic)**
  - Motor sizing, driver components
  - RFID range and placement

- **Software (Android)**
  - Seamless interaction with hardware on network
  - User-Friendly and Functional UI
  - Lack of experience with the development environment

- **Software (Embedded)**
  - Software that can interact over AP without issues
Testing

- Hardware
  - Measuring feeder mechanism.
  - Observing regular feeding schedules to ensure reliable operation.

- Software
  - Concurrency testing for each feeder
    - Master feeder will be tested against user input and virtual supplementary feeders for input while operating peripheral hardware
    - Supplementary feeder will be tested against virtual requests of a master feeder while operating peripheral hardware
Questions?