e6 - waitr

Sophie Sacks, Sam Lavelle, Dina Razek

Use Case

The Scenario

- Hard to know how busy eateries are
- Students may miss an opportunity to grab food between classes
- Tracking wait times and making predictions

The Solution

- Combined hardware and software solution: two radio frequency identification (RFID) scanners + web application
- Scan in at beginning and end of line
- Web application for users to view with machine learning in backend

Use Case Requirements

- Build two RFID readers to accurately capture ID number and scan time
- AC-powered, OR battery powered scanner lasting **3 hours** or more
- Transmission from scanner to server within **70 milliseconds**
- Web application response time in under **2 seconds**
- Margin of error for wait time: **2 minutes**, or within **10%**
- Disregard ID numbers if 3 more ID numbers scan out before it
- Ability to keep track of up to **50 patrons**

Technical Challenges

Use Case Requirements		Technical Challenges
Transmission from scanner to server within 70 milliseconds	→	Connect a board (RPi) to the RFID readers to collect the live data and send to the server quickly
Web application response time in under 2 seconds	→	Update the web application wait time with new data consistently
Margin of error for wait time: 2 minutes, or within 10%	→	Build an ML model that considers past data when predicting wait times
Disregard ID numbers if 3 more ID numbers scan out before it	→	Capture all edge cases resulting from reader use

Solution Approach

The Solution

- 2 RFID readers
- A board to send the data from the readers to the server
- A web application to publish the wait time

The Materials

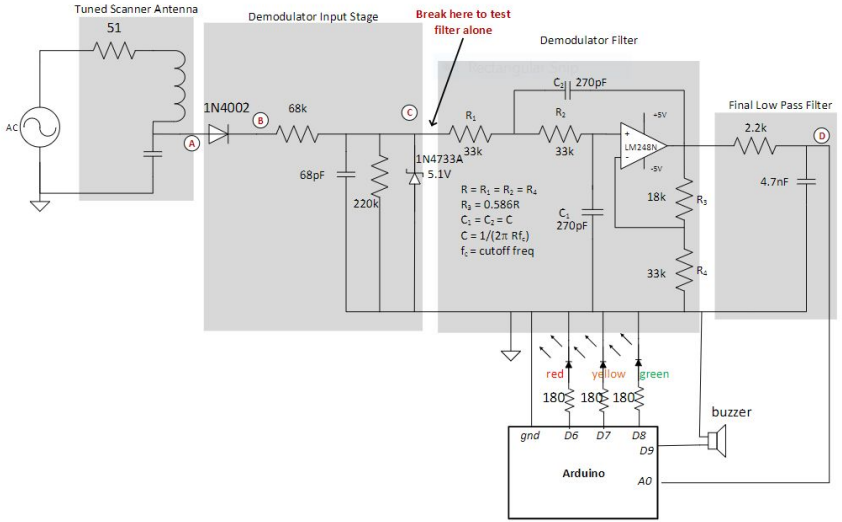
- RFID Readers
  - 2 Arduinos
  - Mag wire for inductor coils, possible 3D printed mount
  - Power source - battery or cable
- Board
  - Raspberry Pi
- Web application
  - AWS server
  - SQL database through Django
  - Python backend (ML library)
  - HTML/CSS/Javascript frontend

Solution Justification

- **Accuracy**: wait times based on direct input and output
- **Anonymity**: no way to identify the individual from the ID number
- **Experience**: lacking experience in computer vision to utilize cameras, but have experience in RFID readers

Material Decisions: Hardware

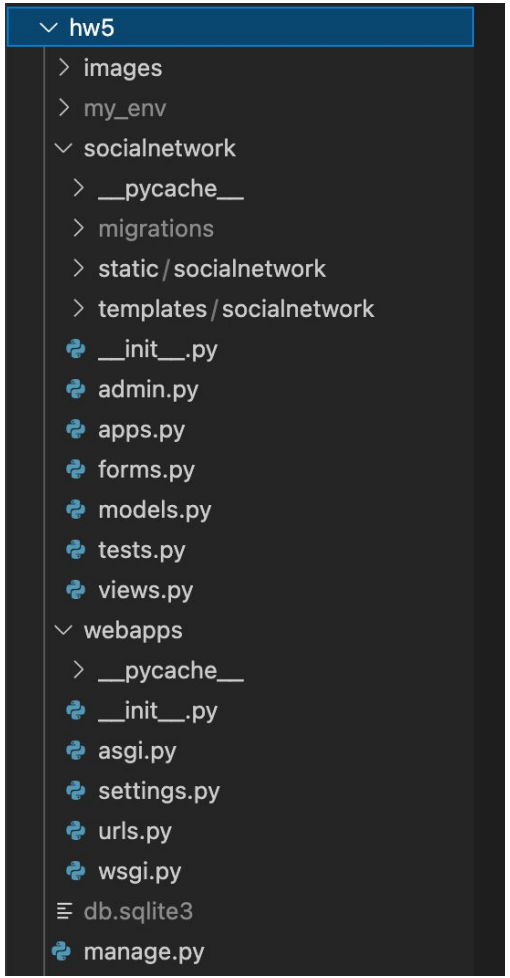
- RFID Reader
  - **Arduino**: based off 18-220 experience
  - **Mag wire for inductor coils**: based off 18-220 experience
  - **3D printed mount**: ability to place scanners in convenient locations
  - **Power source - battery or cable**: dependent on outlet availability, possible duration of batteries, and resources to change the batteries
- Board
  - **Raspberry Pi**: quick transmission of data over the Internet



Source: 18-220

Material Decisions: Software

- Web application
  - **AWS server**: easy to use, flexible, and secure
  - **SQL database through Django with input validation**: Protects against clickjacking, cross-site scripting (XSS), and SQL injections
  - **Python/HTML/CSS/Javascript**: have thorough experience



Testing, Verification, and Metrics

Tests

- Quantitative tests for accuracy of scanner time results
- Binary test for reader to board connection
- Binary test for board to server connection
- Unit testing of backend Python code
- Qualitative usability testing of the web application

Verification Goals

- Accuracy is within a 10% margin of error (iterative)
- Verify board is receiving data within 35 ms
- Verify web server is receiving data in time within 70 ms
- Ensure the product is beneficial, useful, efficient, and fulfilling its goals

Tasks and Division of Labor

- Sam
  - Research RFID scanner builds
- Sam
  - Design project connections (hardware / software) and implement (arduino, mcu…)
- Dina/Sophie
  - Set up software
    - Dina
      - Make github repo
    - Sophie
      - Set up Django
    - Dina/Sophie
      - Purchase a server hosted on AWS
- Sam/Dina
  - Build UI designs and logo
    - Dina
      - User testing on web app design
- Dina/Sophie
  - Backend: Create ML model
- Dina
  - Frontend: Build user interface

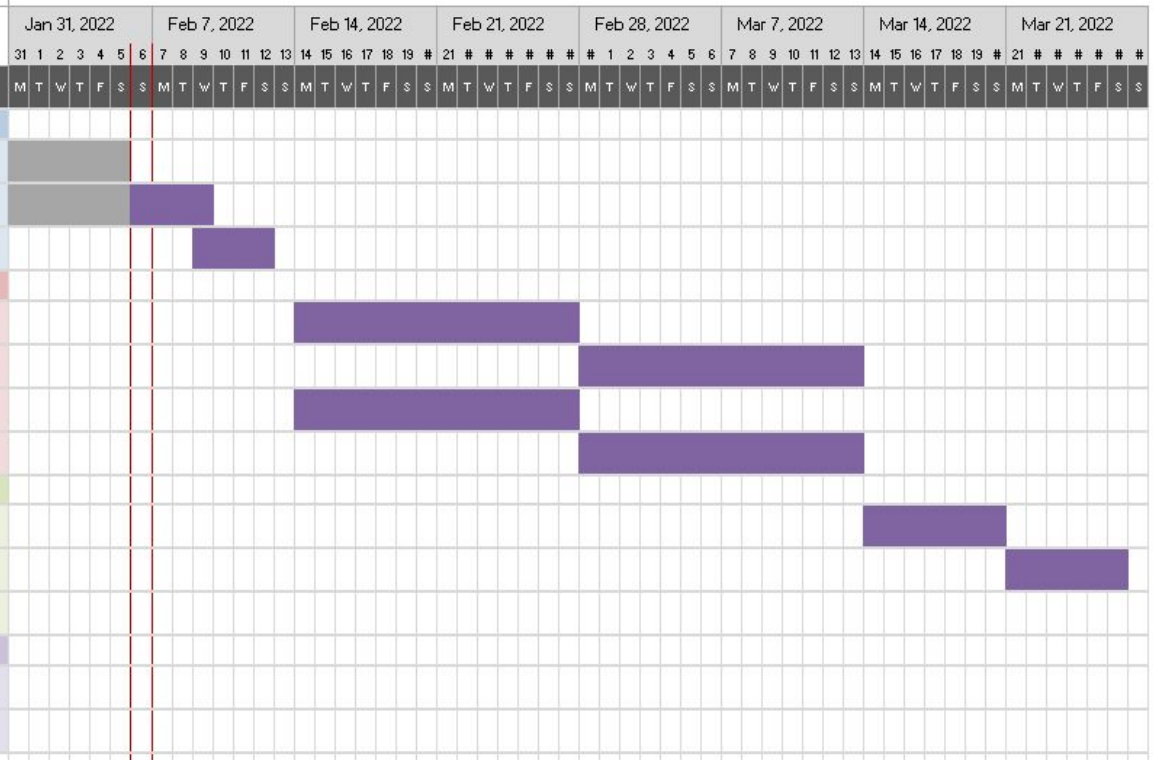
Schedule

18-500

Project Start: Sun, 1/30/2022

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Setup				
Set Up Website	Sam, Sophie	100%	1/31/22	2/5/22
Design Solution Implementation	All	60%	1/31/22	2/9/22
Order Parts	Sam	0%	2/9/22	2/12/22
Build Solution				
HW: circuit + arduin	Sam	0%	2/14/22	2/27/22
HW: Rpi	Sam	0%	2/28/22	3/13/22
SW: Frontend	Sophie, Dina	0%	2/14/22	2/27/22
SW: Backend	Sophie, Dina	0%	2/28/22	3/13/22
Testing				
HW ability to read CMU ID	Sam	0%	3/14/22	3/20/22
Ability to transmit from HW to SW	All	0%	3/21/22	3/28/22
Ability to calculate wait times from data	Sophie, Dlna	0%	3/28/22	4/4/22
Integration				
Integrate wait times into web app	Sophie, Dina	0%	4/5/22	4/19/22
Explore execution in eatery	All	0%	4/20/22	4/27/22



Impact

For Customers

Increased accessibility to food - no skipping meals because you don’t have the time

For Businesses

Decreased stress on eateries & staff, increased business during slow times

For the Future

Scalable to other businesses: La Prima, UC Package Pickup…