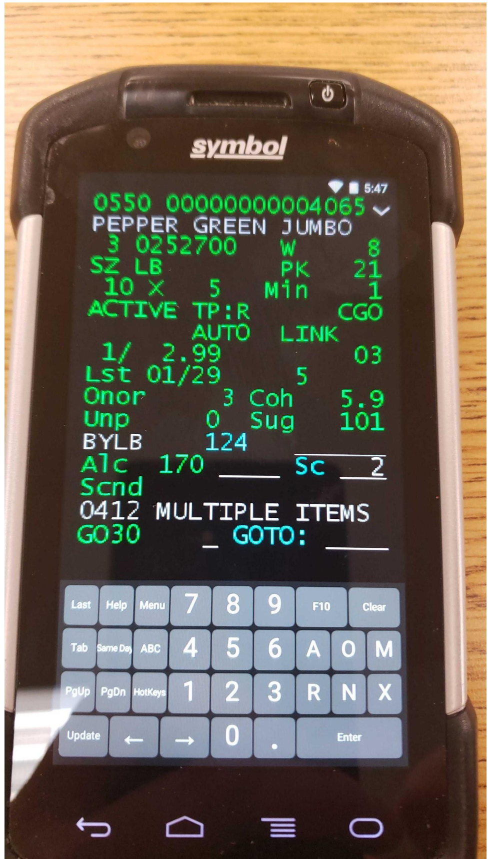
MIMS

Mobile Inventory Management System

Patrick Richeal, Brandon Ostasewski,
Dominic Marandino, John Donahue, Matthew Finnegan,
Paul Sigloch, Sagarika Kumar, and Steven Bruman

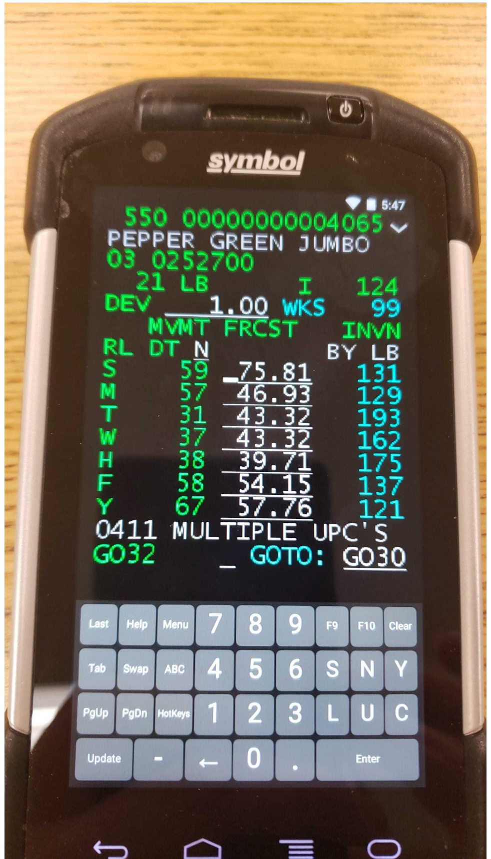
Senior Project, Spring 2019
Dr. Baliga

Current App in use and its flaws



```
0550 00000000004065
PEPPER GREEN JUMBO
 3 0252700       W      8
SZ LB            PK    21
 10 X    5     Min      1
ACTIVE TP:R           CGO
       AUTO   LINK
 1/   2.99             03
Lst 01/29         5
Onor        3 Coh     5.9
Unp        0  Sug     101
BYLB       124
Alc   170 ____  Sc  __2
Scnd
0412 MULTIPLE ITEMS
GO30        _ GOTO: _____
```

symbol
Last Help Menu 7 8 9 F10 Clear
Tab Same Day ABC 4 5 6 A O M
PgUp PgDn HotKeys 1 2 3 R N X
Update ← → 0 . Enter



```
 550 00000000004065
PEPPER GREEN JUMBO
03 0252700
  21 LB          I    124
DEV ___1.00 WKS        99
   MVMT FRCST        INVN
RL DT N              BY LB
S     59  _75.81      131
M     57   46.93      129
T     31   43.32      193
W     37   43.32      162
H     38   39.71      175
F     58   54.15      137
Y     67   57.76      121
0411 MULTIPLE UPC'S
GO32      _ GOTO: GO30
```

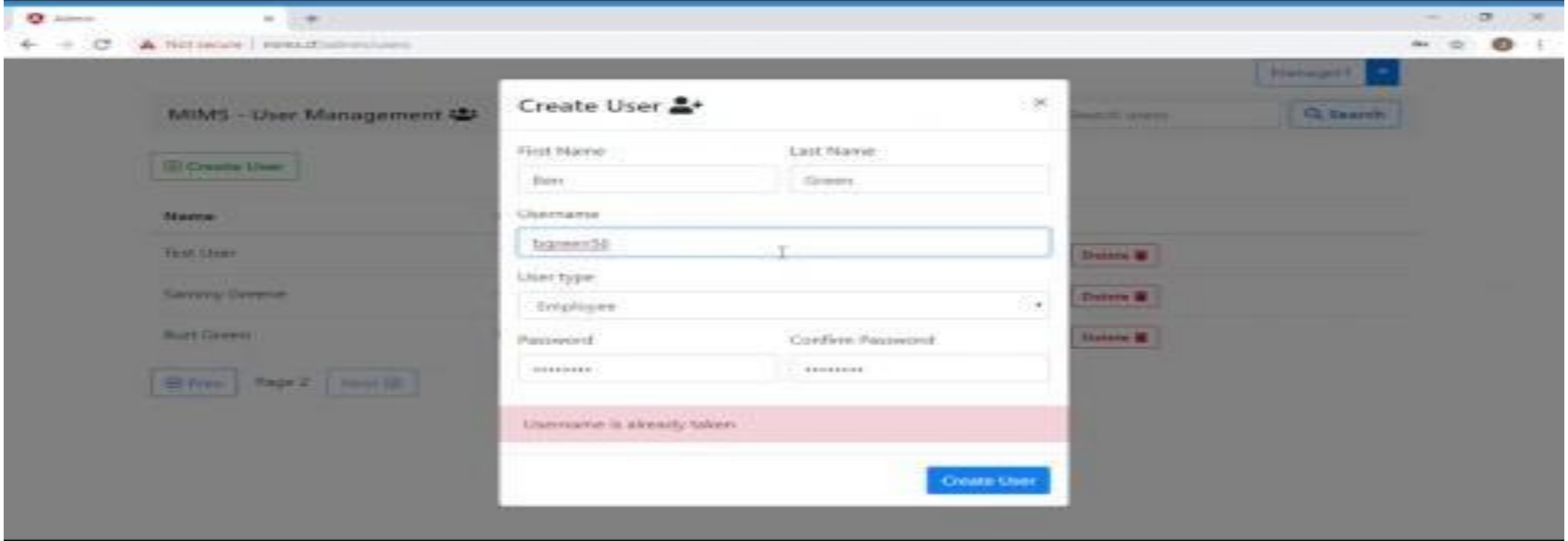
symbol
Last Help Menu 7 8 9 F9 F10 Clear
Tab Swap ABC 4 5 6 S N Y
PgUp PgDn HotKeys 1 2 3 L U C
Update - ← 0 . Enter

Features

- Admin website to allow managers to create employees to login to the app
- App for employees to use to search products with various criteria
- Viewing details of a product
- Adjusting inventory of a product
- Sales movement system to view past product sales history
- Future sales prediction system to view predicted future sales

Stretch goals

- Analytic tools to visualize product sales
- In app barcode scanner to look up products
- Conversions to from imperial to metric system
- Chat system



Test User

Sammy Greene

Burt Green

First Name: Ben

Last Name: Green

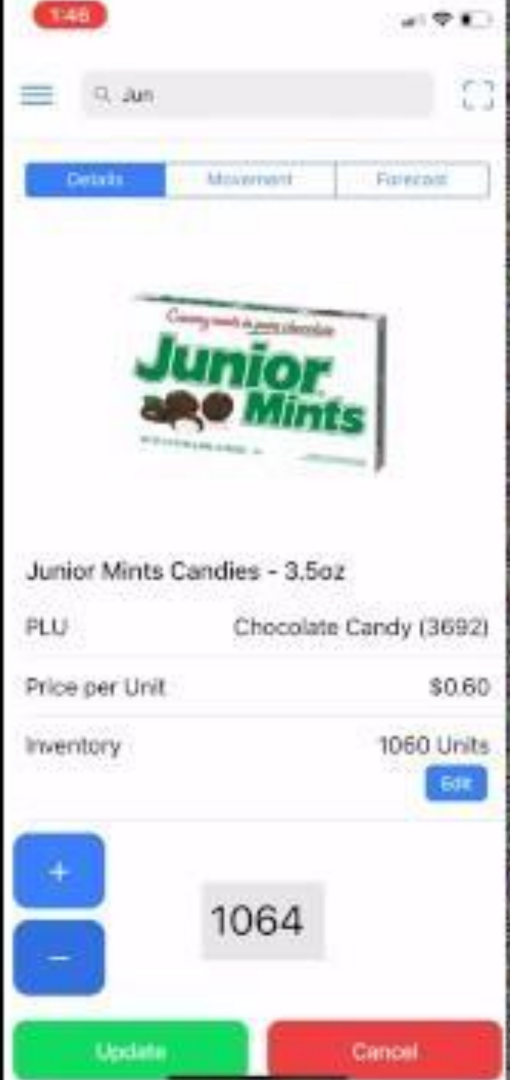
Username: bgreen58

User type: Employee

Password: ••••••••••

Confirm Password: ••••••••••

Username is already taken

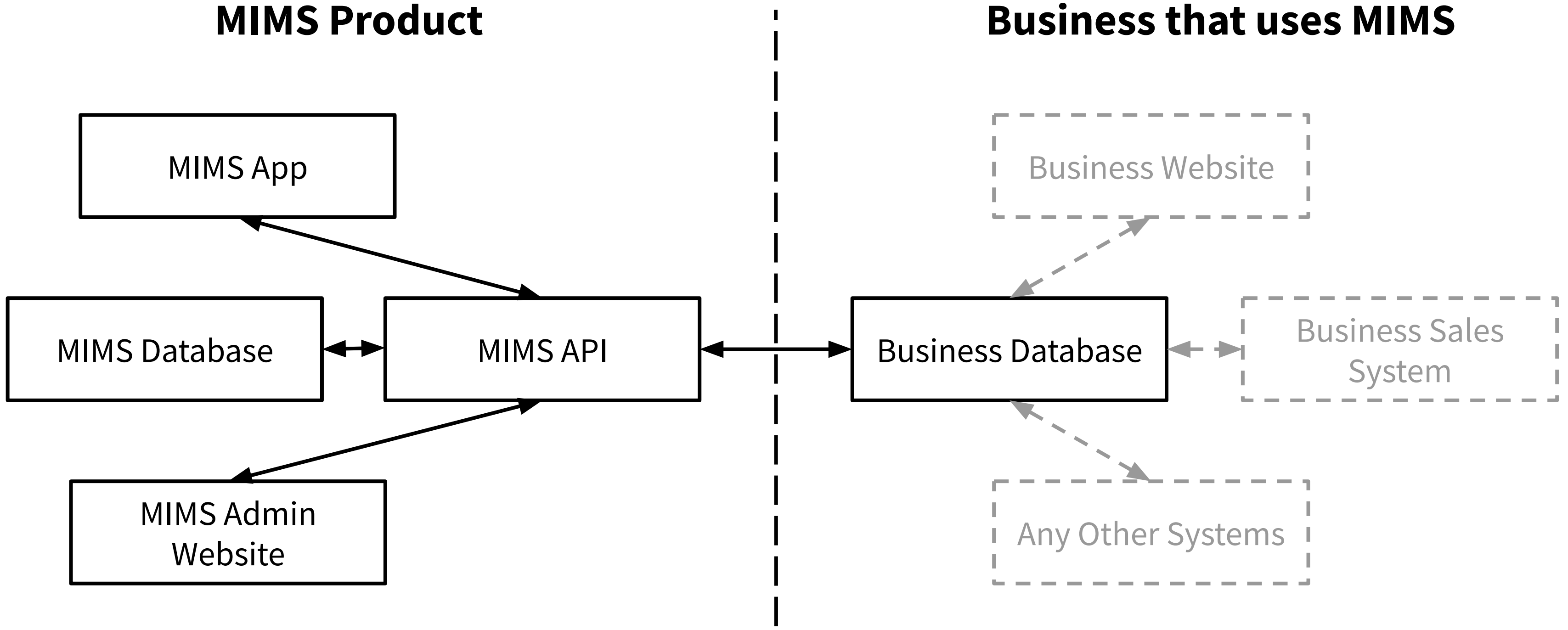

1:46

Jun

Junior Mints Candies - 3.5oz

PLU	Chocolate Candy (3692)
Price per Unit	$0.60
Inventory	1060 Units

Project Architecture



MIMS App

- Built with Ionic 4, a hybrid mobile app framework
- Single code base, can be built for all mobile platforms + web

Benefits

- One code base, many platforms
- App can be accessed from a web browser
- Open source plugins (barcode scanner, visualizations)

ionic

MIMS Admin

- Built with Angular and Bootstrap
- Allowed for rapid development of features

Benefits

- Similar setup as our app because Ionic is built on top of Angular
- Easy and quick to get good looking and functional features with Bootstrap

MIMS API

- Built with Python Flask
- Uses many Python modules like SQLAlchemy, Json Web Tokens (JWT), and PassLib

Benefits

- Flask allowed for rapid development of endpoints
- Open source Python modules gave us access to a lot of high quality functionality with little development time for us

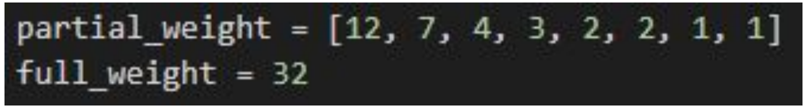
Flask
web development,
one drop at a time

Simulated Business Database

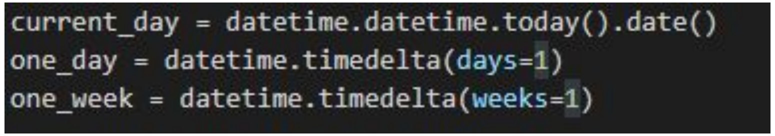
- Simulates sales going a year back
- Sales only happen during store hours
- More sales during busy times of day
- 3 different types of shoppers
- May order new items everyday
- Changes prices at the beginning every week

Forecast Algorithm

- Exponentially weighted algorithm
  - Makes sure more recent dates are taken more heavily than older dates.
  - Helps take seasonal items into play

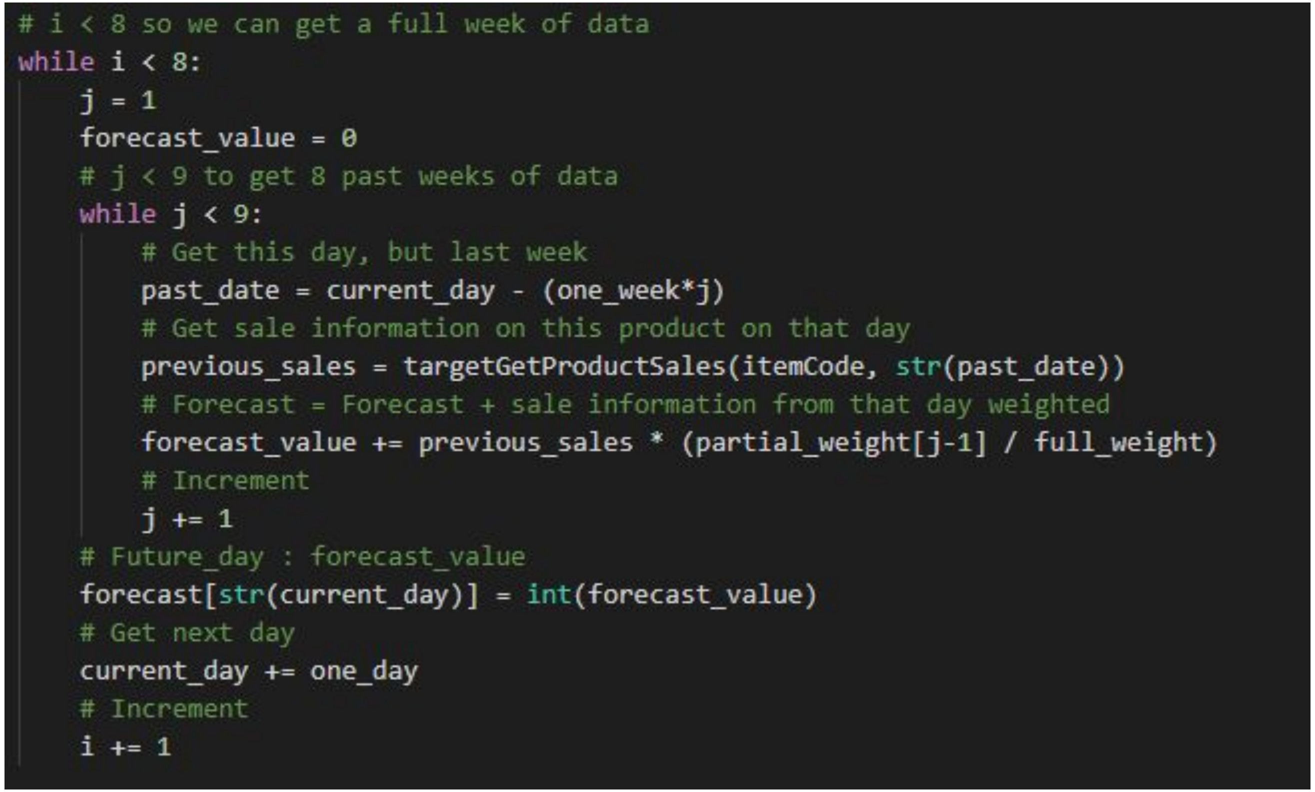


```
partial_weight = [12, 7, 4, 3, 2, 2, 1, 1]
full_weight = 32
```



```
current_day = datetime.datetime.today().date()
one_day = datetime.timedelta(days=1)
one_week = datetime.timedelta(weeks=1)
```

Forecast Algorithm



```python
# i < 8 so we can get a full week of data
while i < 8:
    j = 1
    forecast_value = 0
    # j < 9 to get 8 past weeks of data
    while j < 9:
        # Get this day, but last week
        past_date = current_day - (one_week*j)
        # Get sale information on this product on that day
        previous_sales = targetGetProductSales(itemCode, str(past_date))
        # Forecast = Forecast + sale information from that day weighted
        forecast_value += previous_sales * (partial_weight[j-1] / full_weight)
        # Increment
        j += 1
    # Future_day : forecast_value
    forecast[str(current_day)] = int(forecast_value)
    # Get next day
    current_day += one_day
    # Increment
    i += 1
```

Questions or comments?