



QUICK COMMUTE

TEAM 6

Eamon Glackin | Kaitlin Heneghan | Alejandro Porcel | Kavitha Vishwanathan

APP OVERVIEW

Simple Platform

- Tells you what track your train is on for the following lines:
 - LIRR
 - MTA Metro North
 - NJ Transit
- Knows your favorite route and time
 - Great for commuters
- Clean interface to quickly find other trains outside your regular commute at any time

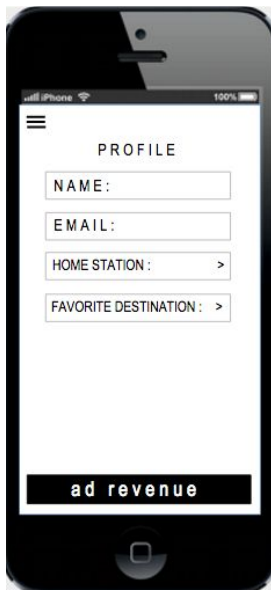


HOW IT WORKS

STEP 1

LOGIN

Create an account with favorite home station and destination station for quicker results



STEP 2

FIND YOUR TRAIN

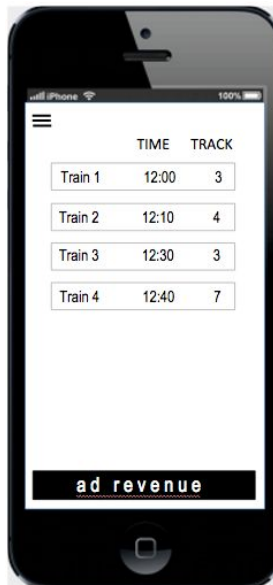
Either use saved preferences or enter new route and press the big red button



STEP 3

TRACKS

App will generate a list of all of the times for your departure and the designated track



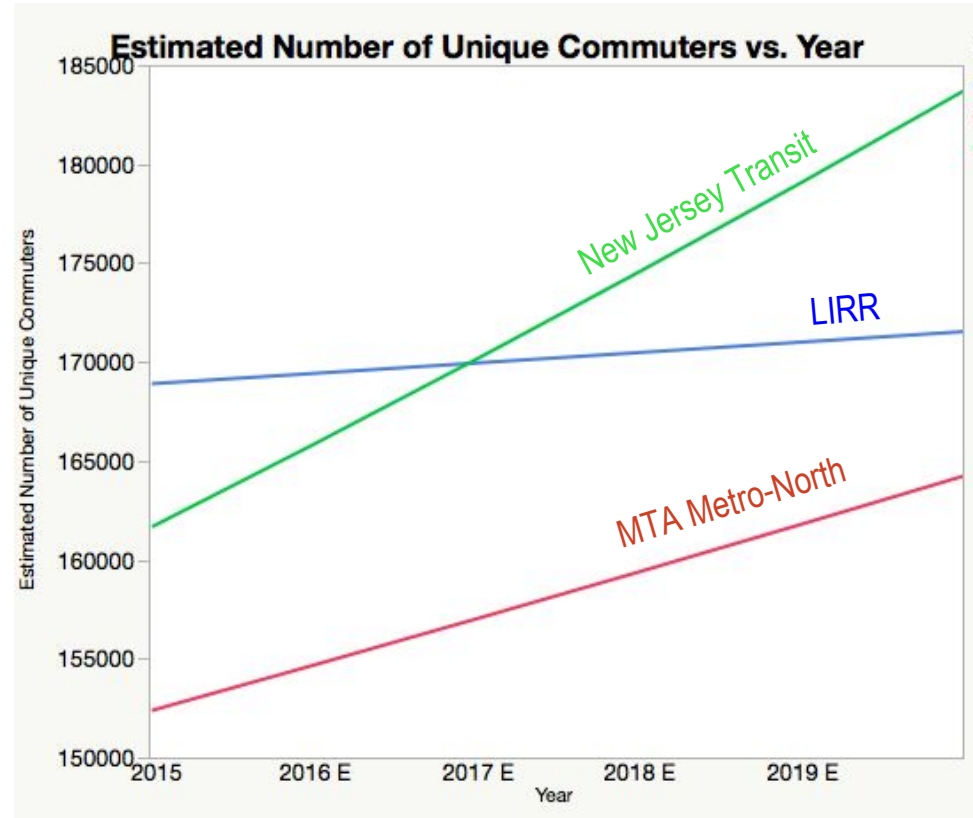
STEP 4

SOCIAL

Share your status, route, and ETA with your social media followers



MARKET



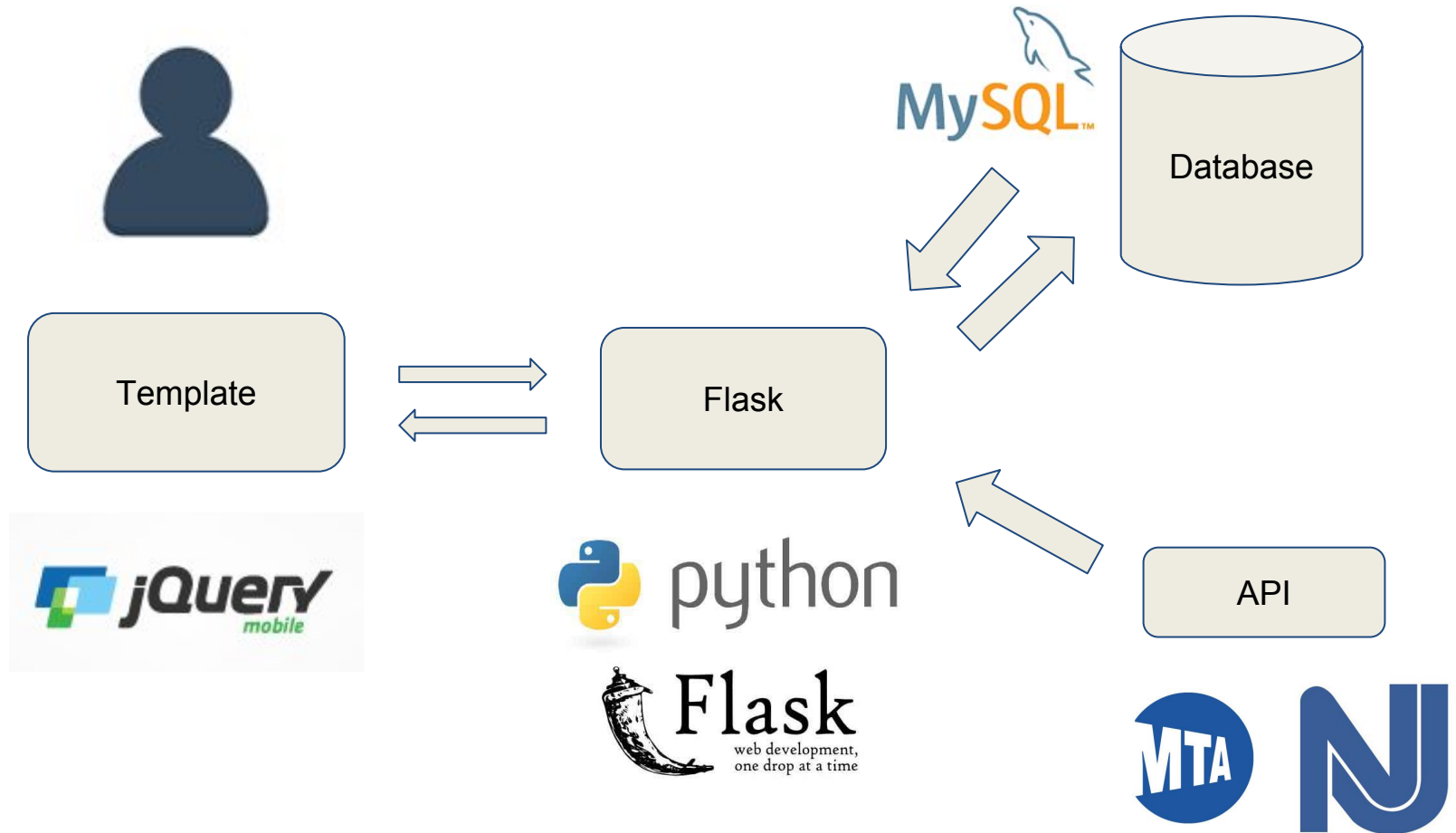
- Total number of daily commuters on our three targeted trains is **483,000**.
- Growing at an average rate of **1.5%** annually.
- Expected number in 2020 is **519,547**

REVENUE MODEL

CUSTOMIZED PAID ADS	
Expected App Downloads, 2017	150,000
# of people who download, but don't use	135,000
Revenue from non-converts (at \$3 eCPM)	\$3,443
# of people who download, and use	15,000
Revenue from converts (at \$3 eCPM)	\$36,000
Annual Revenue, 2017	\$39,443

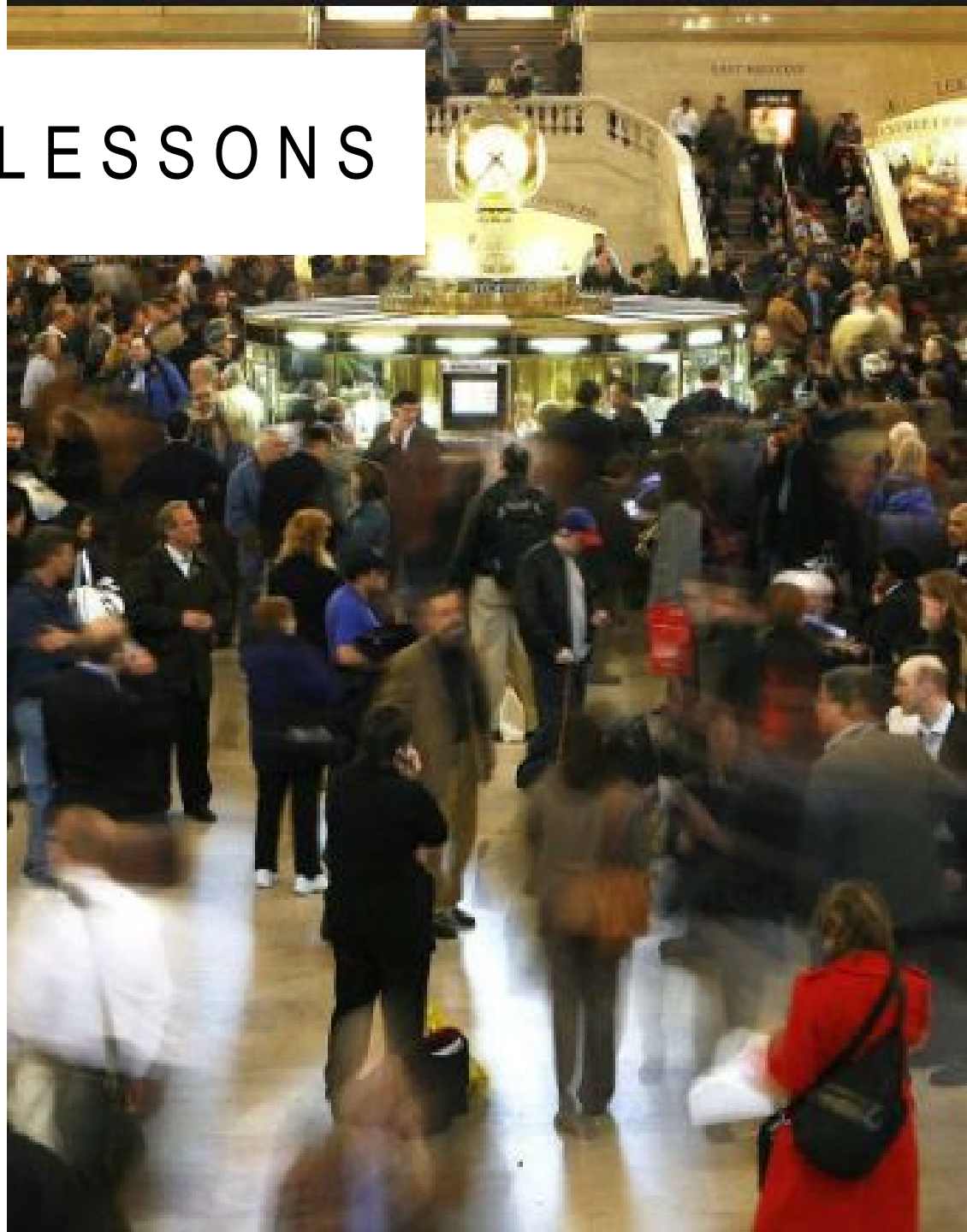


SYSTEM ARCHITECTURE



PROBLEMS & LESSONS

- Getting real time api data from LIRR and NJT
- Could not find real time api from Metro North
- IOS requirements - creating a provisioning profile



NEXT STEPS

- Expand to NY Metro North
- Offer SMS updates
- Find advertisers and build ad serving capability



APPENDIX

TEAM 6

CODE

```
In [3]: def get_trains(Service, FROM, TO, HOUR, MIN):
    count = 0
    today = datetime.date.today()
    YEAR = today.year
    MONTH = today.month
    DAY = today.day
    trains = []
    reqtime = datetime.datetime(YEAR, MONTH, DAY, int(HOUR), int(MIN))
    if Service == 'LIRR':
        urlData = ('https://traintime.lirr.org/api/TrainTime?api_key=%3CYOUR_KEY%3E&startsta={0}&endsta={1}&year={2}&mon
        webUrl = urllib2.urlopen(urlData)
        data = webUrl.read()
        jsonData = json.loads(data)
        Trips = jsonData["TRIPS"]
        for trip in Trips:
            legs = trip["LEGS"]
            tripTime = datetime.datetime.strptime(trip["ROUTE_DATE"]+legs[0]["DEPART_TIME"], "%Y%m%d%H%M")
            if tripTime >= reqtime:
                out_trip = {}
                out_trip["TRAIN_ID"] = legs[0]["TRAIN_ID"]
                out_trip["ETA"] = legs[0]["DEPART_TIME"]
                if legs[0]["TRACK"] == None:
                    out_trip["TRACK"] = "-"
                else:
                    out_trip["TRACK"] = legs[0]["TRACK"]
                trains.append(out_trip)
        output = {}
        output["TRAINS"] = trains
        json_output = json.dumps(output)
    elif Service == 'NJT':
        FROMName, TOName = Get_StationName(FROM, TO)
        username = 'aporcel'
        #password = 'S2PC4VhL3JgE7W'
        password = 'OTBhV18N3O66ZL'
        request = urllib2.Request("http://traidata.njtransit.com:8092/NJTTrainData.asmx/getTrainScheduleJSON?username=
        result = urllib2.urlopen(request)
        xmlstr = result.read()
```

CODE

```
In [2]: def Get_StationName(StationID1,StationID2):
urlData = "http://traindata.njtransit.com:8092/NJTTrainData.asmx/getStationListXML?username=aporcel&password=OTBhV1l"
webUrl = urllib2.urlopen(urlData)
data = webUrl.read()
root = ET.fromstring(data)
NJStations = []
for child in root:
    #print child.tag, child.text
    row = {}
    for grandchild in child:
        row[grandchild.tag] = grandchild.text
    NJStations.append(row)
NJTdf = pd.DataFrame(NJStations)
return NJTdf['STATIONNAME'][NJTdf[NJTdf['STATION_2CHAR'] == StationID1].index[0]],NJTdf['STATIONNAME'][NJTdf[NJTdf[
```

CODE

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:password@localhost:3306/test'
db = SQLAlchemy(app)

lm = LoginManager()
lm.init_app(app)
lm.login_view = 'login'

@lm.user_loader
def load_user(id):
    return User.query.get(int(id))

class User(db.Model):
    __tablename__ = "UserProfile"
    id = db.Column('UserId', db.Integer, primary_key=True)
    email = db.Column('email', db.String(45), unique=True)
    firstname = db.Column('FirstName', db.String(45))
    lastname = db.Column('LastName', db.String(45))
    password = db.Column('Password', db.String(128))
    registration_time = db.Column('RegistrationTime', db.DateTime)

    def __init__(self, email, firstname, lastname, password):
        self.email = email
        self.set_password(password)
        self.firstname = firstname
        self.lastname = lastname
        self.registered_time = datetime.utcnow()

    def set_password(self, password):
        self.password = generate_password_hash(password)

    def check_password(self, password):
        return check_password_hash(self.password, password)

    def is_authenticated(self):
        return True

    def is_active(self):
        return True

    def is_anonymous(self):
        return False

    def get_id(self):
        try:
            return unicode(self.id) # python 2
        except NameError:
            return str(self.id) # python 3
```

CODE

```
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'GET':
        return render_template('register.html')

    firstname = request.form['firstname']
    lastname = request.form['lastname']
    email = request.form['email']
    password = request.form['password']

    user = User(email,firstname,lastname,password)
    db.session.add(user)
    db.session.commit()

    flash('User registered successfully!!')
    return redirect(url_for('login'))

@app.route('/login', methods=['GET','POST'])
def login():
    if request.method == 'GET':
        return render_template('login.html')

    email = request.form['email']
    password = request.form['password']

    registered_user = User.query.filter_by(email=email).first()
    #registered_user = User.query.filter_by(email=email).first()
    flash(registered_user.check_password(registered_user.password))

    if registered_user.check_password(password):
        login_user(registered_user)
        flash('Logged in successfully')
        return ( '{"%s":"success"}'%email)
    else:
        flash('Email or password is invalid', 'error')
        return ( '{"%s":"failed"}' %email)

@app.route('/',methods=['GET','POST'])
def landing():
    if request.method == 'GET':
        return render_template('commuterAppTemplate.html')
```


CODE

```

<!DOCTYPE html>
<html>

<head>

<meta charset="utf-8">
<title>jQuery Mobile Web App</title>
<link href="http://code.jquery.com/mobile/1.0a3/jquery.mobile-1.0a3.min.css" rel="stylesheet" type="text/css"/>
<link href="static/commuter.css" rel="stylesheet" type="text/css">
<script src="http://code.jquery.com/jquery-1.5.min.js" type="text/javascript"></script>
<script src="http://code.jquery.com/mobile/1.0a3/jquery.mobile-1.0a3.min.js" type="text/javascript"></script>
<script src="static/script.js" type="text/javascript"></script>

</head>

<body>

<div data-role="page" id="page">
  <div data-role="header">
    <h1 style="color: white;" <style="font-family:arial narrow;">L O A D I N G &nbsp;P A G E <br> needs logo </h1>
  </div>

  <div data-role="content">
    <ul data-role="listview">
      <li><a href="#page2">F I N D &nbsp;M Y &nbsp;T R A I N</a></li>
      <li><a href="#page3">T R A I N &nbsp;I N F O</a></li>
      <li><a href="#page4">P R O F I L E</a></li>
    </ul>
  </div>

  <div data-role="footer">
    <h4>$ $ &nbsp;&nbsp;A D &nbsp;&nbsp;R E V E N U E &nbsp;&nbsp;$ $</h4>
  </div>
</div>

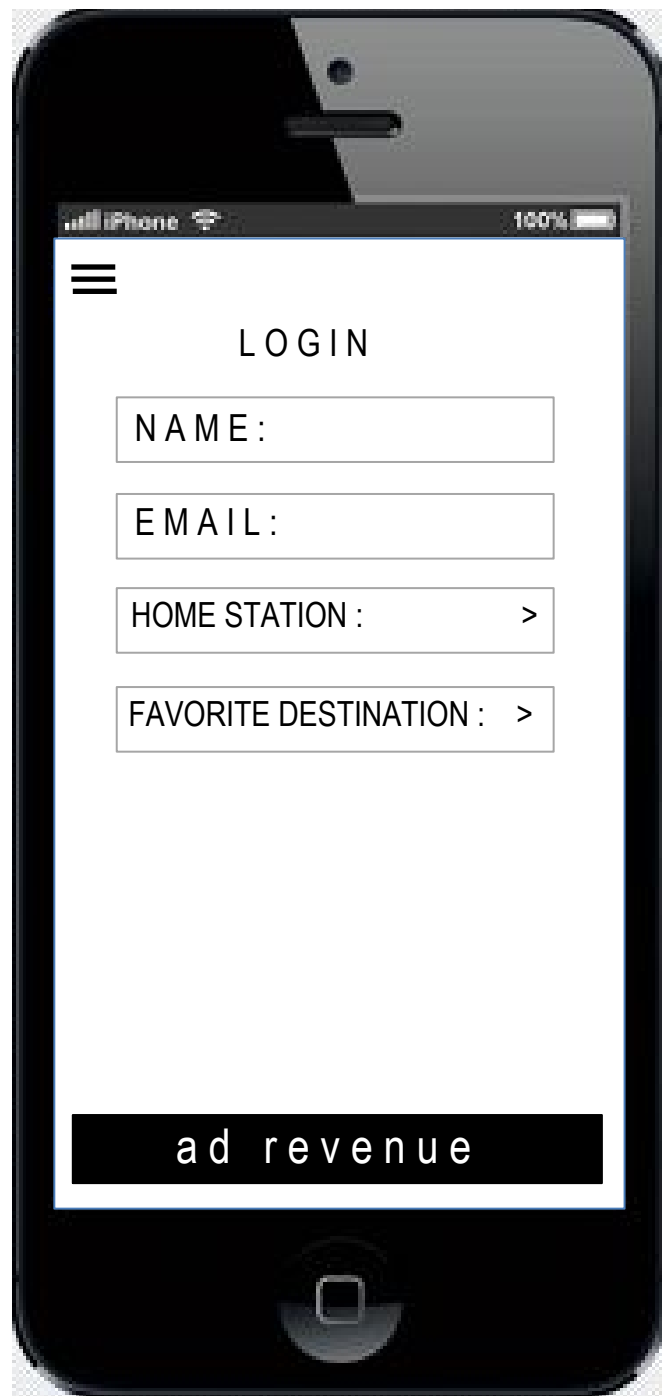
<div data-role="page" id="page2">
  <div data-role="header">
    <h1 style="font-family:arial narrow;">F I N D &nbsp;M Y &nbsp;T R A I N</h1>
  </div>

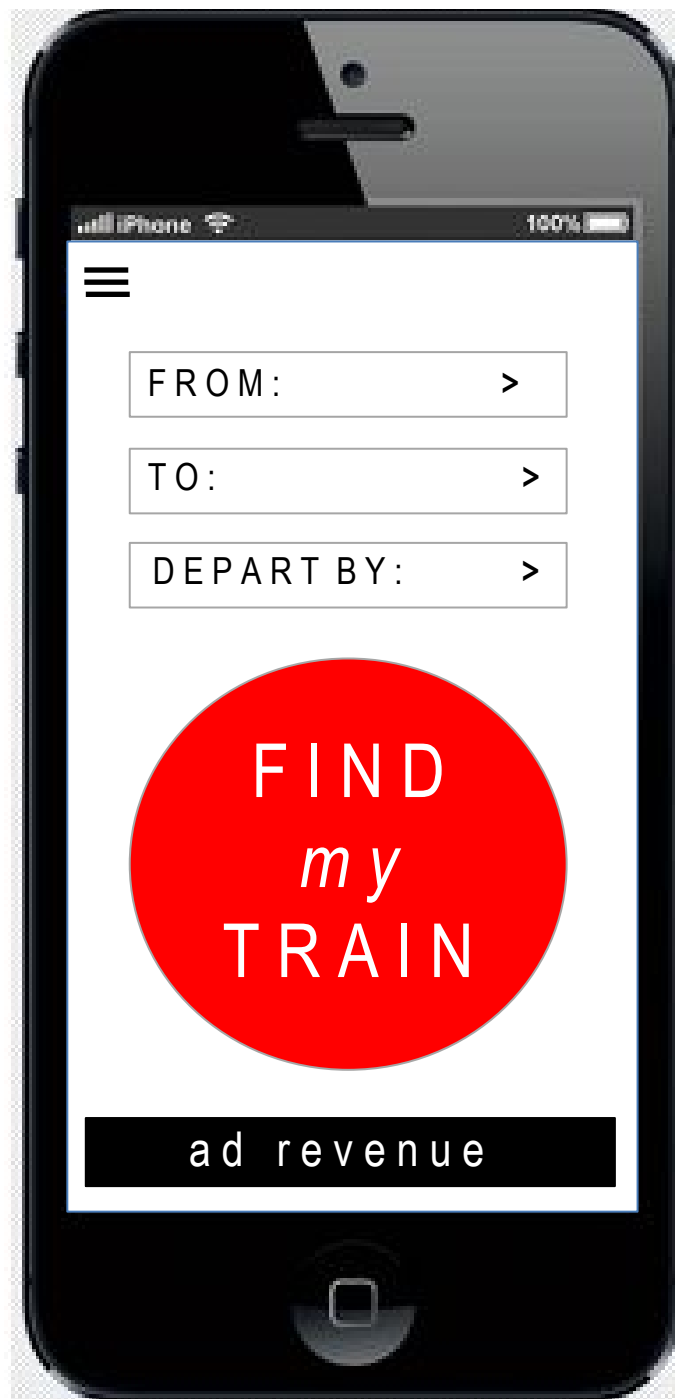
```

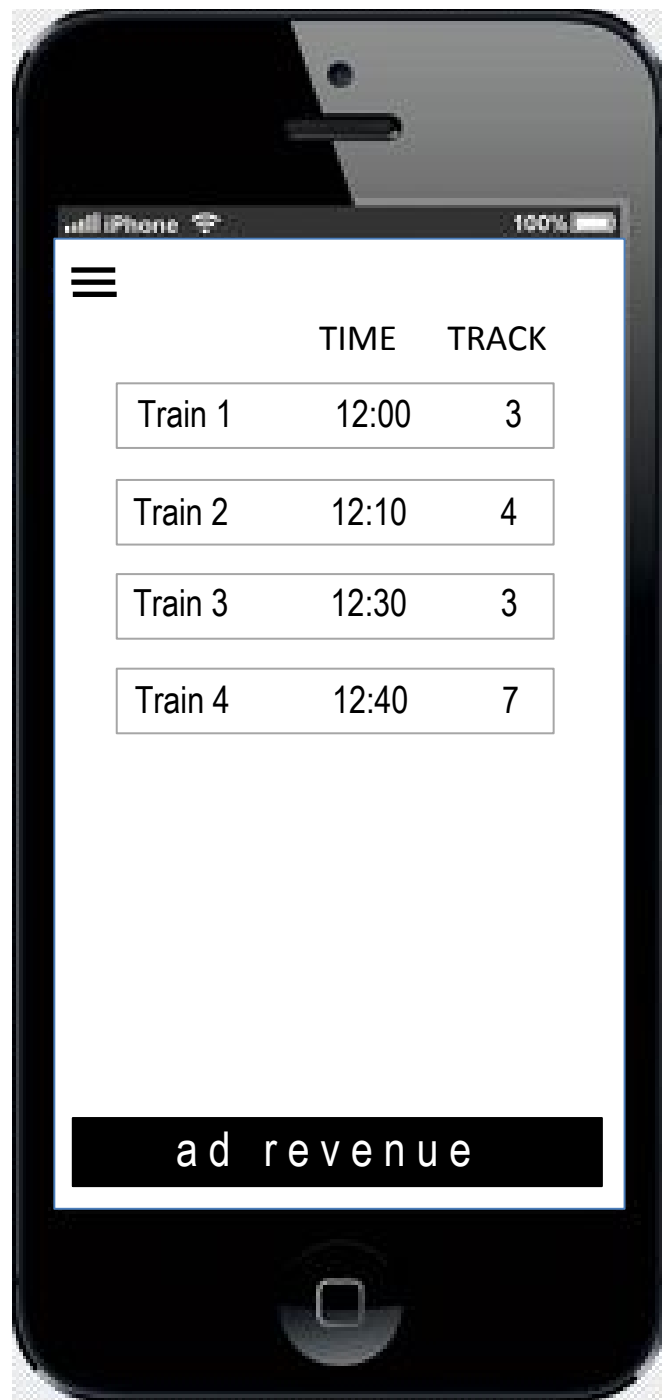
SCREENSHOTS

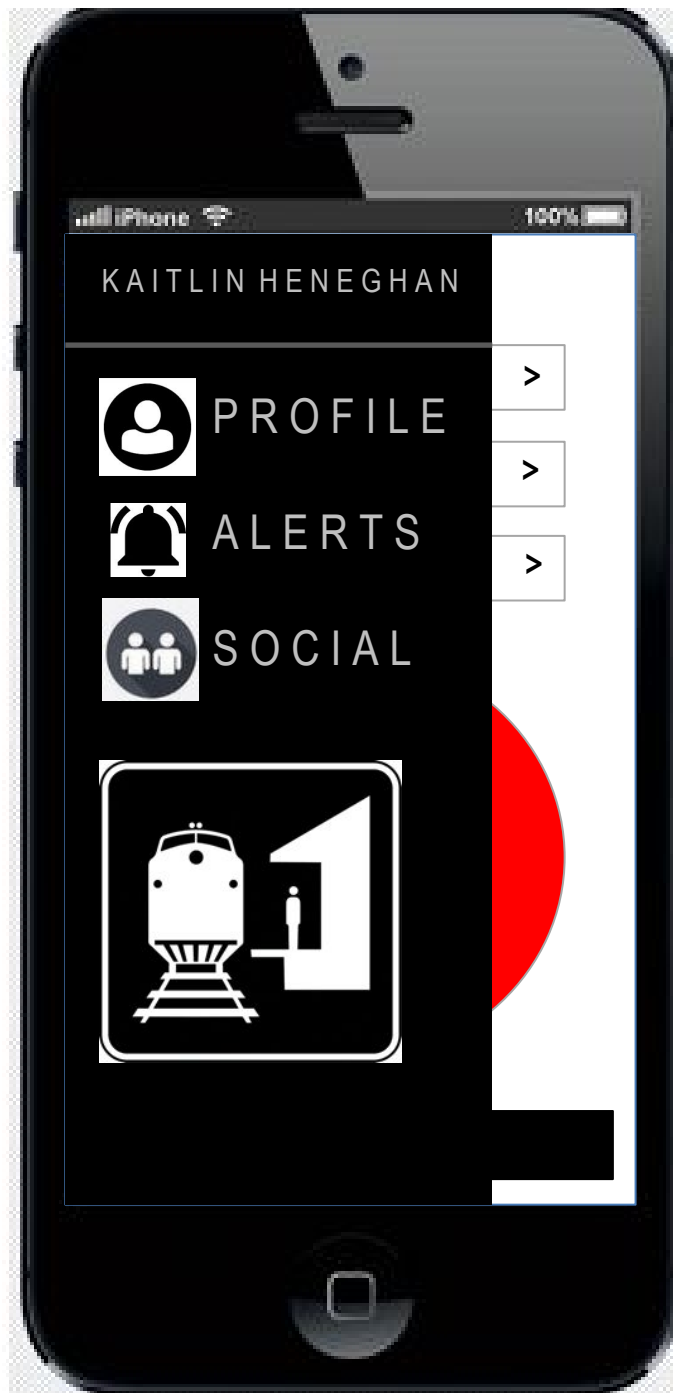
TEAM 6











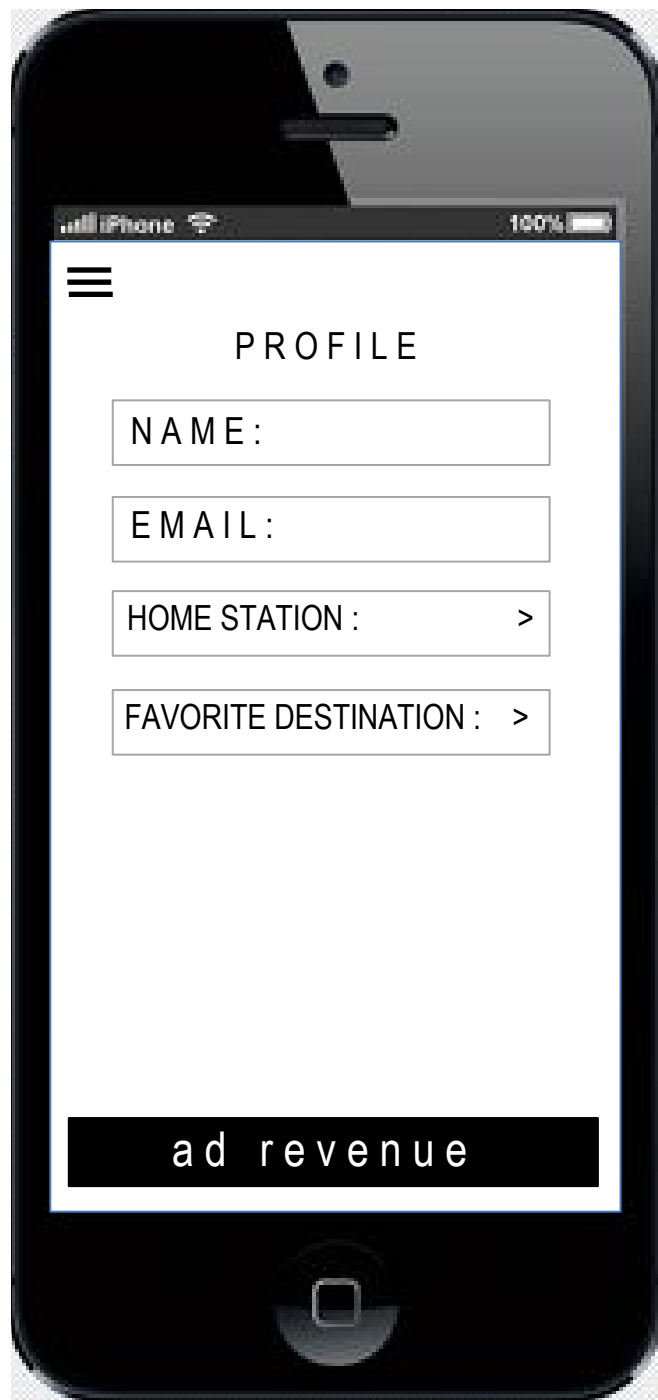


TABLE DESIGNS

