

GoBird

Justin Linwood, Lieu Phung, Chaughn Robin, Cole Snyder

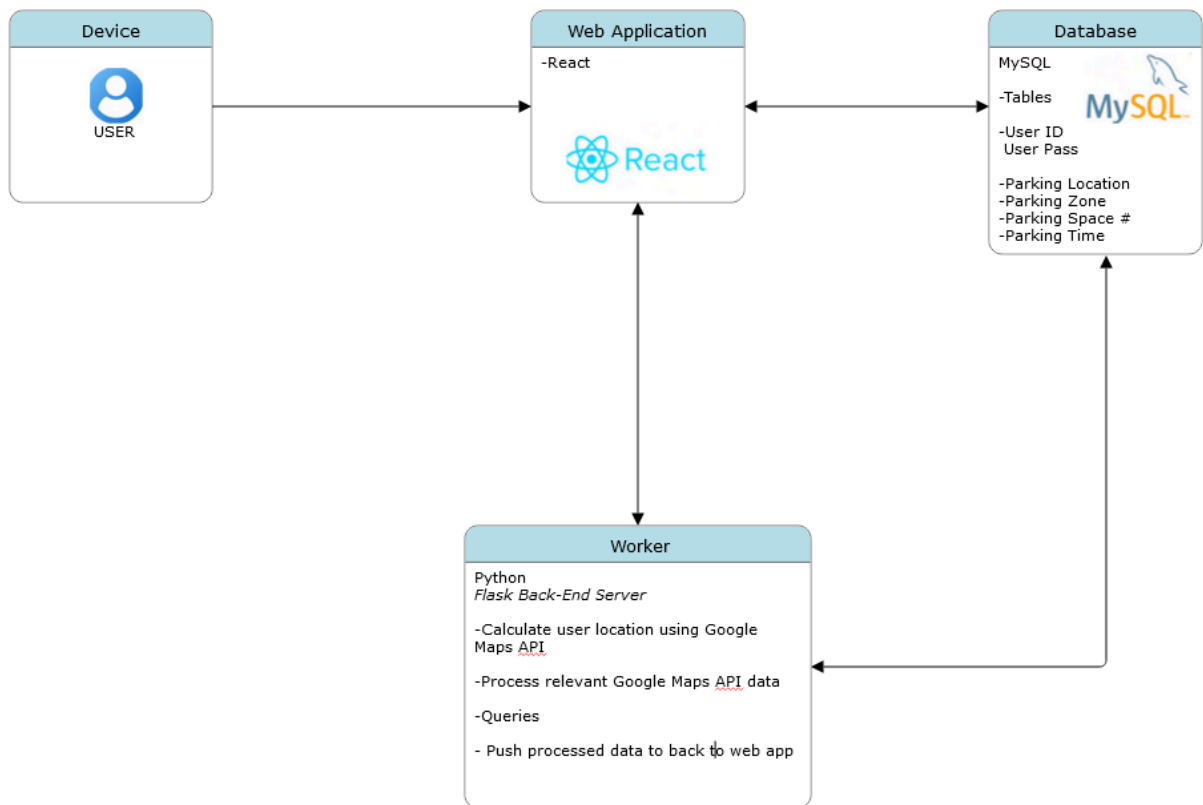
CHAPTER 1:

Purpose: The goal of our project is to create an attendance tracker for cars parked in West Chester University spaces, GoBird.

Navigate parking at West Chester University with ease using our app tailored to the WCU area, simplifying your campus commute and ensuring you're always steps away from where you need to be. GoBird is a parking app that will allow students and faculty at West Chester University to find open parking spots with ease, reducing the stress of searching for parking.

Key Features:

- Real Time Updates - GoBird will deliver real-time parking availability updates to the user by incorporating Google Maps API, which makes use of real-time refreshing, as well as data crowdsourced from users. The user will have a clear image of what areas have availability and what areas do not while using familiar mapping to get there.
- Geolocation - Using Google Maps API, GoBird will use geolocation to display relevant information as well as collect information from users.
- Google Maps API - The app will use Google Maps API to have a clean, familiar interface that is very widely used to make use of customized interactive maps with Google's large amounts of data.
- User-Friendly Interface - A simple interface utilizing a map service most users are familiar with will make using the app an easy process and minimize the cost of educating users.

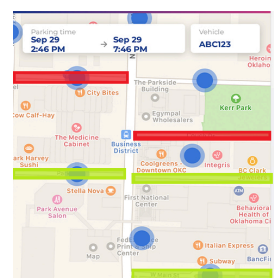


CHAPTER 2:

Implementation:

- React - React will be used to build the front-end user interface. It will handle user interactions such as logging in/out and parking detail information. It will also fetch data from the Python backend using API calls. The information that has been fetched will be processed and integrated with the Google Maps API to display the information on the user's screen. It will also display relevant user information, parking times, and available locations.
- MySQL - MySQL will be used to store the user's username/password, parking location, parking space number, parking zone, and parking time information.
- Python - Python is the primary language for the back-end. This allows us to build logic for processing user information and Google Maps API. It also handles data needed to push/pull from MySQL.
- Google Maps API - Google Maps API will be used for geolocation of the user, customized maps for interaction and display, as well as location data which will be stored in MySQL.

The overall idea is the user will have a service presented to them similar in look to FlowBird or Google Maps. The map presented to the user will



display available and unavailable spots in zones overlaid on the map similar to FlowBird's zones.

INTERMEDIATE MILESTONES (CHAPTER 3):

We have worked on creating the front end for our app (GoBird) by using React. We have started making this by creating a new App.js file and creating a base template to build the WebUI. At the moment it is just a page with simple headers, paragraphs, lists, and buttons.

```
1 import React from 'react';
2 import './globals.css';
3
4 function App() {
5   return (
6     <div className="App">
7       <header className="Header">
8         <h1>My Website</h1>
9         <nav>
10           <ul>
11             <li><a href="#">Home</a></li>
12             <li><a href="#">About</a></li>
13             <li><a href="#">Contact</a></li>
14           </ul>
15         </nav>
16       </header>
17       <main className="main">
18         <section className="hero">
19           <h2>Welcome to GoBird</h2>
20           <p>GoBird is a cutting-edge mobile application designed to simplify and streamline the parking process around West Chester University. Our app provides real-time information on available and taken parking spaces, ensuring that you can spend less time searching for a spot and more time focusing on what matters to you.</p>
21           <a href="#">Learn More</a>
22         </section>
23         <section className="content">
24           <h3>Our Services</h3>
25           <ul>
26             <li><p>Our team's goal is to simplify parking on the West Chester University campus for students, faculty, and visitors by developing a web application that directs you to available parking spaces. GoBird is an attendance tracker for cars parked in West Chester University spaces; it will deliver real-time parking availability updates to the user in an easy-to-use interface. The user will have a clear image of what areas have availability, what areas do not, while using familiar mapping to get there.</p>
27             <li><p>As users enter a parking area, they will be logged in our system as they leave. They will be logged out using data downloaded from each user. We can offer near real-time updates to each user on where they can or cannot park while also displaying this information on the Google Maps interface the users are familiar with.</p>
28           </ul>
29         </section>
30       </main>
31       <footer className="footer">
32         <p>©2024 My Website</p>
33       </footer>
34     </div>
35   );
36 }
37
38 export default App;
```

There is also a file called index.js which initializes our React application. index.js imports all of our necessary libraries, defines our root element, and renders the main app within the root element. We have created our Dockerfile as well as our .yaml files for our WebUI as well. We plan on implementing the Google Maps API with our WebUI to show the user's current location. We also plan on having an overlay over the API display to show the parking space locations. This is one of our current struggles. We are able to display our map on our website but the overlay still needs a lot of work, also we have other technical issues like hiding our API key in our public GitHub repository. In addition, we plan on using maps to count and create templates to mark out the parking spots rather than having someone go down each street and parking lot to count all of the parking spots and map it out.

Our database Container has been created using MySQL in Docker. We have our basic dockerfile created but we still have to add columns for [Lieu can you add that information here?]

We have worked on creating the back end for our app by using Python. Our Dockerfile builds a Python image and installs necessary dependencies (mysql-connector, Flask) listed in the requirements.txt file. It also exposes ports for the other containers to communicate with. Currently, we are deciding whether to use Flask to communicate with the React container or whether we will just use the HTTP libraries built-in python. Flask may be more user-friendly and have more features, but depending on our use cases, HTTP may be sufficient. We will have more input into which method to use as we build the React container. Choosing to use Flask may require some API in React, but looks like the correct choice. It will allow us to more easily route and process HTTP requests from React, push HTTP responses to React, process data and insert into mySQL, handle json data, authentication (if necessary), and error handling. Dealing with all of these issues ourselves would be much more time costly and less feasible.

Our python worker currently can communicate with other containers and do basic functions of the Google Maps API like getting locations, converting them to meaningful, storable data, and storing that data in the mySQL container. It also builds a Flask back-end server for communicating with the React front-end. In Flask, we deal with functions that correspond with specific jobs / functionalities – called endpoints. These endpoints route the requests from the React front-end. As we continue, we will build code for endpoints used by Flask. These will have logic for handling several different tasks, such as submitting forms, updating information, or asking the user for more information.

```

1  from flask import Flask, request, jsonify
2  import mysql.connector
3  import requests
4
5  # Initialize Flask app
6  app = Flask(__name__)
7
8  # Configure MySQL connection ***** MUST BE UPDATED!!!!
9  mysql_host = "mysql"
10 mysql_user = "your_mysql_user"
11 mysql_password = "your_mysql_password"
12 mysql_database = "your_mysql_database"
13
14 # Connect to MySQL database
15 db = mysql.connector.connect(
16     host=mysql_host,
17     user=mysql_user,
18     password=mysql_password,
19     database=mysql_database
20 )
21
22 # Cursor for executing SQL queries
23 cursor = db.cursor()
24
25 # Google Maps API key (replace 'your_api_key' with your actual API key)
26 GMapKey = "your_api_key"
27
28 # Google Maps API base URL for Geocoding
29 geocoding_url = f"https://maps.googleapis.com/maps/api/geocode/json?key={GMapKey}"
30
31 # Endpoint for processing data
32 @app.route('/process', methods=['POST'])
33 def process_data():
34     # Get data from request
35     data = request.json
36
37     # Example: Convert address to geographic coordinates using Google Maps Geocoding API
38     address = data.get('address')
39     if address:
40         # Send request to Google Maps Geocoding API
41         params = {'address': address}
42         response = requests.get(geocoding_url, params=params)
43         result = response.json()
44
45         # Extract latitude and longitude from API response
46         if 'results' in result and result['results']:
47             location = result['results'][0]['geometry']['location']
48             latitude = location['lat']
49             longitude = location['lng']
50
51             # Example: Insert data into MySQL along with geographic coordinates
52             sql = """
53             INSERT INTO your_table_name (column1, column2, latitude, longitude) VALUES (%s, %s, %s, %s)
54             """
55             val = (data['value1'], data['value2'], latitude, longitude)
56             cursor.execute(sql, val)
57             db.commit()
58             return jsonify({'message': 'Data inserted successfully'}), 200
59         except Exception as e:
60             return jsonify({'error': str(e)}), 500
61     else:
62         return jsonify({'error': 'Failed to geocode address'}), 500
63     else:
64         return jsonify({'error': 'Address not provided'}), 400
65
66 if __name__ == '__main__':
67     app.run(host='0.0.0.0', port=8080)

```

The main struggles right now are building functionality within each respective container and then making those functionalities work together. Endpoints/functionality need to be constructed within the python code for properly handling requests, processing data, storing properly in mySQL and sending requests back to the front-end. Also incorporating whatever Google Maps API functions we wish to use. Lastly, the matter of properly running these things in Kubernetes / proper construction of yaml files (nodePorts, environment variables, etc.).

Our development process begins within our GitHub repository. Within our repository, we have set up multiple branches for each aspect of our design structure, for example we have a branch for web ui which is separate from our SQL database. When we begin to build the Docker experiments we have files that will use Kubernetes to create and manage our Docker containers. Once Kubernetes has completed their tasks the

Docker containers should be able to communicate with one another to complete the goal of our app.

Justin Linwood

Warminster, PA 18974
215-915-0111 | justin.linwood@verizon.net

Education

West Chester University, West Chester PA

Bachelor of Science – Major Computer Science, Minor Mathematics
Graduation Date: May 2024
GPA: 3.368

Technical Skills

Programming Languages

- Strong understanding and comfort coding complex projects in Java.
- Understanding of C and C++
- Understanding of Python and how to use it working close to hardware

Development Tools

- IntelliJ, JGrasp, Eclipse, Linux, Windows, GitHub, CloudLab, Jenkins

Relevant Coursework

Computer Science courses

- Cybersecurity · Cybersecurity II · Program Concepts and Paradigms · Datacom/Networking I · Software Engineering, Software Security, Artificial Intelligence, Intro to Cloud, Modern Malware
- · Calculus I - III · Linear Algebra, Discrete Mathematics, Statistics, Differential Equations

Certifications

Security+, CompTIA, January 2024
Computer Security, West Chester University, May 2024

Experience

Tony's Place, 4/17 – present

- Bartender
 - Duties included: taking orders, preparing drinks, providing high quality customer service in a high paced environment

References

Jason Howard – Manager - Tony's Place
Jayhoward77@gmail.com | 267-516-0765

John Sedlacsik – Teacher - Hatboro-Horsham School District
jsedlacs@hatboro-horsham.org | 267-475-9402

Lieu Phung

484-425-4683 | lieunphung@gmail.com

EDUCATION

West Chester University

Bachelor of Science in Computer Science

West Chester, PA

Aug. 2023 – May 2025

- Related coursework:
 - Computer Science I
 - Computer Science II
 - Computer Science III
 - Computer Security & Ethics

Lehigh Carbon Community College

Associate of Science in Computer Science

Schnecksville, PA

Aug. 2020 – May 2023

- GPA of 3.82

EXTRACURRICULAR

Programming Contest | *Java*

September 2023

- Participated in West Chester University's programming contest to improve coding skills
- Solved problems correctly in a timely manner

Computer Science Club |

Aug. 2023 – Present

WORK HISTORY

Recreation Attendant

West Chester University

August 2023 – Present

West Chester, PA

- Ensure patrons have valid identification
- Monitoring patrons to ensure a safe environment
- Clean strength and conditioning equipment
- Properly check-out/in equipment
- Act as a first responder in the event of an emergency

Stocking and Unloading Associate

Walmart

Aug. 2020 – Aug. 2023

Whitehall, PA

- Provide customers with any assistance needed in the store
- Unload general merchandise trucks according to each department
- Stock merchandise in its proper location

TECHNICAL SKILLS

Languages: Java

Developer Tools: Visual Studio Code

Chaughn Robin

chaughnr@outlook.com :: {484} 723-9366

EDUCATION

West Chester University of Pennsylvania, West Chester PA
Bachelor of Science in Computer Science, December 2024
GPA: 3.7/4.0 | Dean's List
Computer Science Club, Cyber Security Club

RELATED COURSEWORK:

- *Data Structures and Algorithms*
- *Computer Security and Ethics*
- *Computer Systems*
- *Digital Image Processing*

CURRENT COURSEWORK:

- *Database Management Systems*
- *Programming Language Concepts/Paradigms*
- *Introduction to Cloud Computing*

WORK HISTORY

West Chester University of Pennsylvania, West Chester, PA
Computer Technician **January 2024 – Current**

- Analyzed and processed information from students
- Collaborating with students to facilitate a variety of introductory programming courses
- Accommodated and instructed students in different levels of Computer Science curriculum with diverse concepts

PROJECTS

LibGDX (Java Game-Development Framework) **January 2024 - Current**
Word-Based iPhone/Android App
Developer

- Programmed a unique “hangman-like” word game based on acronyms and a large library of possible words using Java
- Created icons and artwork for the game using Paint.net
- Utilized LibGDX's portability features to create releases for PC, Android, and iPhone
- Produced several in-game tracking tools to measure player metrics and data for playtesting and further improvement / monetization using Java

Sidescroller Role-Playing Shoot 'Em Up PC Game **August 2023 - Current**
Developer

- Constructed several in-game systems for tracking player, player aim, different forms of movement, hitboxes, events and input based on player as well as A.I. characters using Java
- Created fully rigged 3D and 2D models and animations using Blender and Spine
- Recorded audio effects and voice lines using Audacity

RELATED SKILLS

Languages
Java, C++, Python, C#, Haskell, Ruby, SQL

Tools
Eclipse, MS Visual Studio, Blender, Spine, Access, SPSS, MySQL

Operating Systems
Windows, Linux (Kali & Tails), macOS

Skills

- Conceptualizing and designing efficient algorithms for implementation
- Understanding and implementing appropriate data structures
- PC Building and Hardware

Cole Snyder

Phone: 484-467-8451
Email: 20snyderc@gmail.com

Education

West Chester University, West Chester, PA

Exp. Grad. Dec 2024.

- Bachelor of Computer Science
- **Current GPA: 3.431/4** (Deans list)

Experience

Coding

Started August 2020

- Partook in Computer Science classes at West Chester University.
 - CSC 141 Computer Sci I (Java).
 - CSC 142 Computer Sci II (Java).
 - CSC 240 Computer Sci III (Java).
 - CSC 231 Computer Systems (C & Linux).
 - CSC 241 Data Structures and Algorithms (Java).
 - CSC 301 Computer Security & Ethics.
- Completed a code academy course that covered Java.
- Completed a code academy course that covered HTML.
- Experience with object orientated programming (Java).
- Won the 2023 CSTA Hackathon at WCU.
- Attended WCPC Coding challenges.

Freelance I.T.

Started April 2019

- Successfully completed computer repair and maintenance.
- Executed computers construction.
- Delegated software instillation, management, and set up.
- Administered peripheral repair.

Skills

- Meeting Deadlines.
- Managing Projects.
- Identifying, Analyzing, and creatively solving problems.
- Multi-tasking.
- Time Management.
- Accepting responsibility.
- Ability to work in a team and collaborate effectively.
- Proficient in Microsoft Word and Excel.
- Proficient in Google Workspace applications.
- Proficient and Java.
- Adapting and working with problems with new problems as they arise.