PizzaDronz Project Proposal

Stressed students spending countless late evenings in the confines of Appleton Tower working away on important projects often crave a quick, easy, and tasty meal during their break. Currently their options are going out or bringing a packed lunch which are cumbersome and dull respectively. But what if there was another option? A pizza delivery service delivering pizza straight to the rooftop of Appleton Tower from their favourite pizza restaurants. This project proposal will layout the functional and non-functional requirements, foreseen challenges, a potential project plan and some rudimentary UML diagrams to exhibit the feasibility of the project.

Requirements

The following statements are based on the Project Specification (Glienecke, 2023)

Functional Requirements

- Account creation
  - Users must be able to create accounts and login securely
- Order handling and validation
  - Project must be able to take users' orders, validate them
- Drone flight path calculation
  - Project must be able to take a drone's current position, a restaurant's position and calculate the most optimal flight path between the three, accounting for restrictions (expanded on in non-functional requirements)
- Customer support
  - Project must be able to handle customer complaints in case of lost items, or other issues

Non-functional requirements

- Compliance
  - Drone routes must strictly comply with no-fly zones to avoid suspicions of breeching privacy and injuring students upon a drone malfunction
- Robust
  - Project must be able to handle dynamic world data. i.e. changes in available restaurants, menu's and no-fly zone zoning
- Scalability
  - Project should be able to handle a large amount of orders
- Reliability
  - Project must be available during Appleton Tower opening hours (currently 8:00 – 22:00, subject to change)
- Efficiency
  - Project must calculate the most efficient routes as only one drone is available to make deliveries
- Security

- Project must be up to date with the latest security procedures to ensure safe transactions and prevent data breeches

Foreseen Challenges

Weather conditions

As the drone will be operating in Edinburgh, we must account for Edinburgh's temperamental weather patterns. The city regularly gets sudden onslaughts of rain without much warning and is often very windy (Weather Spark, 2023), both of which could affect the drone's ability to function and navigate effectively. To survive Edinburgh's torrents of rain, the drone must be fully waterproof, to combat the extreme wind is more challenging. Studies often suggest using the Euler method to perform numerical analysis to predict the trajectory of drones under different kinds of wind conditions (Gugan, Gopi, and Anwar Haque, 2023). This of course would entail the usage of real-time wind data in our drone navigation algorithm. Furthermore, it must be ensured that stormy weather conditions do not interfere with communications between the central server and the drone.

Unpredictable collisions

The abundance of food being consumed by students all around the central area, has created a very seagull friendly environment causing them to frequently harass students. Our hot pizza carrying drone may be no different, especially as seagulls are known to be aggressive towards drones as they see them as a threat. (Leslie, 2021) The drone must be fitted with propeller guards to avoid damaging the drone (and bird) upon collision, as well as we must ensure it flies at a high enough altitude to minimise the risk of seagull collisions. The pathfinding algorithm must also be up to date with the drones current position and updates it's trajectory in real time to adjust for any unpredicted collisions.

Project Plan (Critical Path)

Here is a high-level pathway for developing the project.

1. Requirements Analysis

Define and document the specific requirements for the optimal path-finding algorithm.
Identify input data (e.g., delivery locations, obstacles, drone capabilities).
Determine the desired output (e.g., a path or route for the drone).

2. Research and Feasibility Study

Research existing path-finding algorithms and technologies.
Conduct a feasibility study to assess the technical and resource requirements.

3. Algorithm Design

Design the core path-finding algorithm, considering factors like obstacles, drone speed, altitude, and safety.
Determine the data structures and data representation needed for the algorithm.

4. Prototype Development

Develop a prototype of the path-finding algorithm to test its feasibility and functionality.
Implement basic functionalities for path calculation.

5. Data Collection and Integration

Set up data collection mechanisms, which may include GPS data, drone capabilities, and maps.
Integrate the algorithm with data sources and implement data pre-processing.

6. Path Visualization

Develop a visualization component to display the calculated optimal path.
Implement a user interface for input and display of the path.

7. Testing and Debugging

Test the algorithm with different scenarios and edge cases.
Identify and fix bugs, errors, and performance issues.

8. Optimization

Optimize the algorithm for efficiency, considering factors like runtime and memory usage.
Fine-tune the algorithm to improve the quality of the optimal path.

9. Integration with Drone Control System

Integrate the path-finding algorithm with the drone's control system.
Implement communication protocols for transmitting the path to the drone.

10. Safety Measures

Implement safety features to ensure the drone avoids collisions and adheres to regulations.
Perform risk assessments and safety testing.

11. User Testing and Feedback

Conduct user testing with drone operators and users to gather feedback.
Make necessary adjustments based on user input.

12. Documentation

Create comprehensive documentation for the algorithm, including user manuals and developer guides.

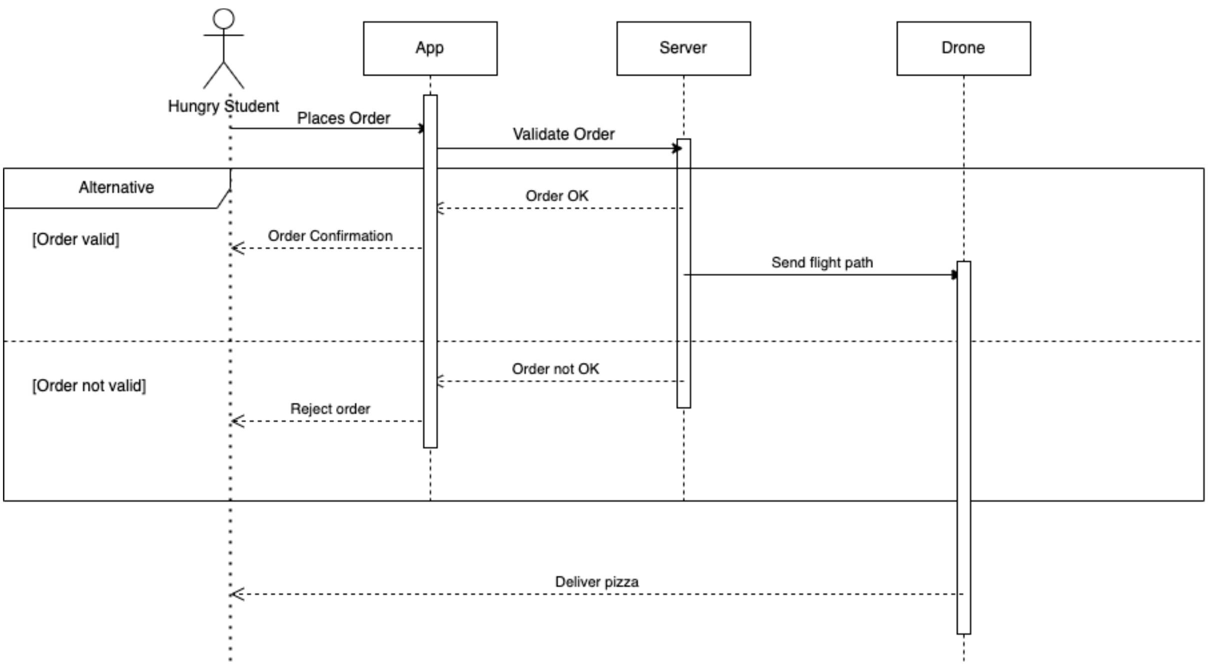
13. Deployment

Deploy the algorithm and software on the drone delivery platform.
Ensure compatibility with various drones and operational environments.

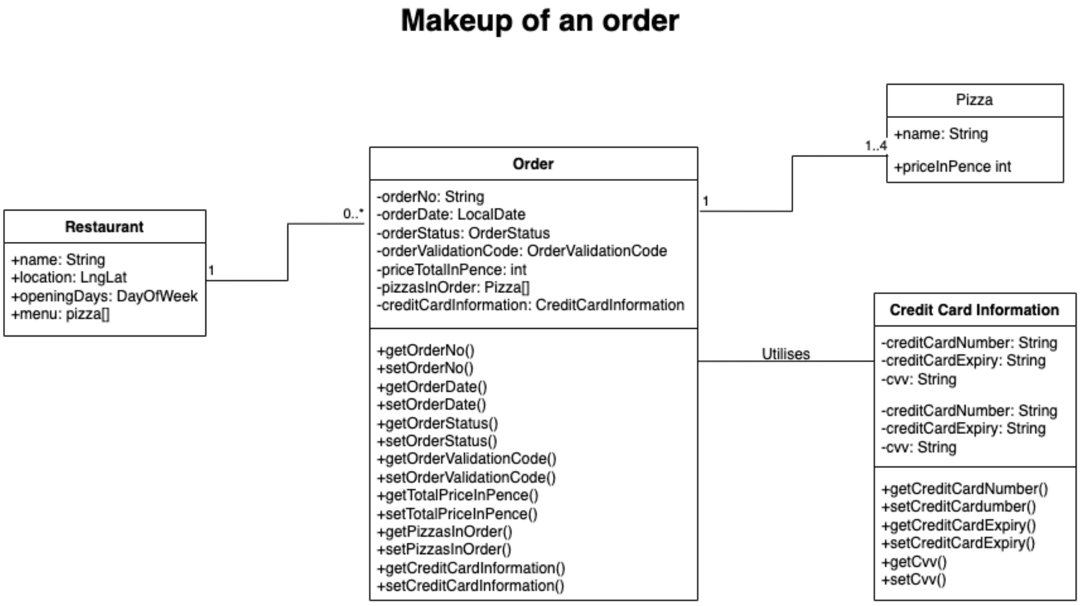
UML Diagrams

Here are some UML diagrams to illustrate the functionalities of the service. The sequence diagram illustrates the how the service would operate at a high level, from user to the drone and back to the user, whereas the simple class diagram illustrates the makeup of an order.

Sequence Diagram



Class Diagram



References

Micheal Glienecke (2023) Informatics Large Practical Course Specification
https://www.learn.ed.ac.uk/ultra/courses/_112434_1/outline/edit/document/_9457283_1?courseId=_112434_1&view=content

WeatherSpark.com (2023) Climate and Average Weather Year Round in Edinburgh
https://weatherspark.com/y/38026/Average-Weather-in-Edinburgh-United-Kingdom-Year-Round#:~:text=The%20windier%20part%20of%20the,of%2015.6%20miles%20per%20hour

Gugan, Gopi, and Anwar Haque (2023) "Path Planning for Autonomous Drones: Challenges and Future Directions" https://doi.org/10.3390/drones7030169

James Leslie (2021) HOW TO STOP BIRDS FROM ATTACKING YOUR DRONE
https://dronesurveyservices.com/prevent-birds-attacking-your-drone/