Software Requirements

This document contains the software requirements for the ILP Pizza Dronz project. It covers the stakeholders for the system, the functional and non-functional requirements and the measurable quality attributes the project possesses.

Stakeholders

Listed below are some of the most important Stakeholders of Pizza Dronz, and some of the requirements that they may expect the system to meet.

The customers

The customers of the system are the students and professors in Appleton tower. As they are the ones who will be ordering the pizzas, they are perhaps the most key stakeholder. The system should be accessible to these customers at any time, it should be easy to use, process their orders in an accurate and timely fashion and should ensure any private information the customer gives is secure.

The restaurants

The restaurants are responsible for providing the pizzas for the customers. The clients rely on them being able to use the system without any issues and they rely on the system for additional profit. The system should allow the restaurants to easily update their menu on the app. It should make sure the restaurants receive orders as soon as they are placed, and it should work with them to ensure that the drones arrive and leave at correct times and with the right orders.

The drone suppliers

The drone suppliers are vital stakeholders in a drone delivery service, and the drones provided should be able to deliver the pizzas around Edinburgh in normal conditions. The system will need to integrate with these drones, and give them appropriate flight paths to follow. The drone suppliers will need to maintain the drones in case of any mechanical faults.

The future software developers

Future software developers will be taking over the project and maintaining it, so the condition of the code is important to consider for these stakeholders. The system should be built to be scalable and maintainable in future.

The public

The drones will be flying over many parts of Edinburgh so residents of the city and the general public are stakeholders. The system will need to make sure that the drones do not endanger the public, and also consider their privacy.

The government

The government and the council will have restrictions on how and when drones can be used around Edinburgh and so act as another stakeholder. The system should adhere to all laws imposed on drones and adapt accordingly should those rules ever change.

Functional requirements

Below some of the functional requirements of the Pizza Dronz project are specified, many of which are listed in the ILP coursework specification document. A functional requirement specifies a function that the system must be capable of performing.

- The system must not allow for the drone's flight path to enter any no fly zones.
- The system should be able to represent Restaurant locations and Appleton tower and longitude and latitude coordinates.
- The drone should be considered to be close to a location if it is within 0.00015 degrees to it.
- Each move that the drone makes should be of distance 0.00015 degrees and it can move in one of the 16 tertiary compass directions.
- The drone should start and end every day at Appleton tower.
- The drone should "hover" whenever it has arrived at a restaurant or at Appleton.
- The drone should deliver one order at a time, with an order consisting of a maximum of 4 pizzas.
- After collecting the pizzas from outside the Central Area, the drone must return to the Central Area using the shortest distance possible.
- Once the drone has re-entered the Central Area, it must not leave this area again until all of the pizza has been delivered to the drop off point.
- The system must receive dynamic data from the REST-service to avoid this data from being hardcoded.
- Rest data will be fetched from the REST API every time the system is started.

- The system must record the drones flightpaths for each day in three different JSON-format files.

- The system must ensure that the orders coming in are valid and if they are not, they will not be carried out. The system will go through each order when it is placed and check if all aspects of the order are valid such as an acceptable order date and of the restaurant is available on the day of the order. If the order is invalid, it will indicate this with an error message and this order will not be placed.

Non-functional requirements

Below some of the non-functional requirements of the Pizza Dronz project are specified, some of which are listed in the ILP coursework specification document. A non-functional requirement describes not exactly what the software will do, but how the software will do it.

- The system should also aim to route the drone above buildings and away from populated areas when possible, as this would reduce the chance of harm if the drone malfunctions.

- The drone should avoid infringing on the publics privacy or making the public feel unsafe about it. Making sure the drones do not have cameras and adequately relaying this to the public will help.

- The system should be coded in a way that is easy to read and to maintain. This is important as the system will be passed on to separate software developers.

- The system will need to have a large enough database to deal with the orders and should be tested with a large amount of data thoroughly before hand.

- The system should aim to plan and plot the flightpath of the drone in as short a time as possible, at most 60 seconds, as any longer than this will increase the time the drone is docked, and further back up the orders.

- Only the necessary information given to the system through the orders should be passed on to the restaurants, and all other information must be securely stored. This can be ensured by encrypting the information received and not storing any information that is not necessary to store.

- The payment method used to place an order should be secure, using a trusted third party such as PayPal can help ensure this.

- The interface should be easy and intuitive for a customer to use.

Measurable quality attributes

Below are the measurable quality attributes that were considered during the development of the project. Some attributes can be deemed more important than others when time and cost are taken into consideration.

Availability

The availability of a system is measured by considering how often the system can be accessed. The Pizza Dronz system should be ideally available for the customers to places orders at any time and the drones should be able to deliver these orders when the restaurants are open and within any legal drone flying restrictions.

Efficiency

The efficiency of a system concerns how quickly it takes to perform the different functions it may have. In our case, we will try and reduce the time of the routing algorithms as much as possible, making it take at maximum 60 seconds to compute a day of orders.

Security

The security of a system is measured by how well the system can prevent any type of attack, unauthorised changes or manipulation of the data. Our system should make sure that any data that the customer has provided is kept secure and it should have adequate protection to feasible attacks.

Maintainability / Scalability

The maintainability / scalability of a system concerns how easily the software can be changed or fixed if there is an error in the code. As the Pizza Dronz project is being handed on to a separate team, it is important to make the code as maintainable as possible, to avoid future issues.

Reliability

The reliability of a system measures the chance of an issue or failure over a long period of time. The more reliable we can make the delivery system, the less time and money we will have to spend to fix the problem.