Software Requirements Specification

for

Medicine Delivery Drone

Version 1.0 approved

Prepared by Zakiyyah Harris, Chelsea Davis, Nikitha Malladi, and Shaniyah Jones

CS420/520 Software Engineering

2.22.2024

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

Through this assignment, our team can create a program that correlates with a drone to simulate medication dropoff. The product will be used by pharmacies and delivered to clients, which will be useful within the pharmaceutical business scope.

1.2 Product Scope

Currently in modern medicine and technology, there are a few choices to take when obtaining prescribed medication, which are picking up medicine, getting medicine through postal mail, or having a delivery driver drop it off. What does one do when they have nobody to take them to the pharmacy, can't wait too long for mailed medication, and doesn't want to risk their privacy for medication?A pharmaceutical delivery drone is the solution to these issues. While this could save a lot of money for pharmacy companies, a pharmaceutical delivery drone could positively change the expected time patients get their medication,and reduce the need for mailed medications and pharmaceutical delivery drivers.

1.3 References

https://learn.org/articles/How_Do_I_Become_a_Pharmaceutical_Delivery_Driver.html#:~:text=A%20pharmaceutical%20delivery%20driver%20delivers,pharmacy%20drivers%2C%20or%20pharmacy%20couriers.

https://www.proxet.com/blog/pharmacy-delivery-problems-and-solutions-you-can-implement-for-your-business

2. Overall Description

2.1 Product Perspective

It is a follow up concept of delivering pharmaceuticals, while creating the original idea of using drones to deliver the medications.

2.2 Product Functions

The following are functions that will be present in the product delivered in class:
• ***Flight Management:***

- Drone has ability to safety launch, head to inputted location, and return
- (As stated in Deliverable #1, an optional function) Ensures secure and successful operations during flight which includes preventing
hazards and detections of accidents.

• ***Observing and Reports***

- Message log to report when the drone is taking off, delivering an item, % of route traveled.
- Error messages will appear when a error occurs: such as the drone being unconnected, or the drone seemingly being stuck.
-(Optional function): Whenever a delivery has fully completed, it is added to a general database of completed deliveries that can be accessed by the Manager.

2.3 User Classes and Characteristics

The following user classes are:

- Operator:
  -an employee of the store
  -uses the program for daily deliveries a customer has notified them to be delivered by drone, and thus uses it the most.
- Store Manager:
  -only one with access to the Log History, to send that information to their superiors regarding frequency of drone usage and dealing with customers who want to confirm if their delivery was successful. Sending information about the Log is done manually through paperwork.

2.4 Operating Environment

- *A common computer that has the ability to run Java files and executables should be able to run this program. The most common operating system used for pharmaceutical retail is Windows, so we will be prioritizing that it runs successfully in that OS. This program only works with the Trello Drone, so it must be compatible with the drone's API in order to move, route, and utilize its video camera.*

2.5 Design and Implementation Constraints

Since the most apt application of the drone will most likely be in retail pharmaceuticals, the program's UI must be simple enough for even a teenager working part-time to operate the product. The most important constraint is the drone itself, the drone's movement, video, flight time, and ability to reach its destination will all be determined by the drone's hardware and software capabilities.

2.6 User Documentation

A training video and FAQ of the program will be provided for the clients to be trained to use. It is expected that any hardware concerns about the drone itself should fall back on the documentation/help provided with the drone's manufacturing company.

2.7 Assumptions and Dependencies

Most notably, the drone itself does not have GPS capabilities, making routing difficult. Inputted "Addresses" will be less likely to be actual addresses and more likely to be a selection of steps to be flown to. (Ex: Fly X distance at Y angle or Z height from your current point) With this in mind, there is a certain assumption that the drone must always start at the exact same position for accurate delivery. We are dependent on the Tello Java API for movement capabilities.

3. External Interface Requirements

3.1 User Interfaces

The user interface is made from Java GUI, and will include one screen with an input box, two buttons, and a message box. The buttons include a "deliver" button and a "return" button. The input box allows a user to enter an address's coordinates and press deliver to send the drone to the given coordinates, and the return button allows the drone to return to the home coordinates (which will not be entered by the user), and doubles as canceling the trip to return "home". All messages (such as error messages, take off messages, returning messages, and percentages) will appear in the message box section. In a real life situation, a login screen would be necessary along with a showing the user's ID or name at the bottom of the screen to track accountability in mistakes of delivering medicine such as typing a coordinate incorrectly.

3.2 Hardware Interfaces

This program is developed to only be compatible with the Tello Drone and uses specific commands from the Tello Drone's API to connect and move the drone. We will also be using the Tello Drone's video camera for navigation.

3.3 Software Interfaces

This product is developed entirely in Java, and thus works on any computer's operating system so long as they have a Java Development Kit installed on their computer to execute the program.

3.4 Communications Interfaces

While not present in the product we are returning in class, if this product was made by a real team, employees (needed for access to the product itself for accountability reasons) profile information would most likely be stored in a private database owned by the client, the pharmaceutical company that requested this product. In the product that we are delivering in class, the name and address to send to is manually inputted by the employee using the product, and is not stored. The Log of past deliveries should be stored locally, and after the Store Manager sends a record of week's deliveries(through hardcopy paperwork), the Log should be cleared out.

4. System Features

4.1 Flight Routing

Description: Given a certain address, the drone must be able to launch, fly to that given location, land so that the medicine may be recovered by the customer, and then return to the store. [Priority: High]

- Benefit: 10 - Main feature of program
- Risk: 7 - No GPS software so accuracy might prove dubious even with the assistance of a video camera and sensors to guide it during flight. Flight time is also largely dependent on the drone itself.
- Penalty: 10 - Lack of this feature means that the delivery cannot happen.
- Cost: 8 - Investing in better drones or additional hardware equipment to be used on the current drone may result in a total overhaul of the program in post-maintenance.

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

The user, an employee, will set the drone in its starting location. They activate the program on a computer, input the address via coordinates, and hit the "Deliver" button. When that occurs, the drone connects to the computer through Wi-Fi, and takes flight to the address. Once reached the address and successfully landed to deliver the medicine, the employee hits the return button. The return button automatically inputs the store's location as the address to fly to, and the drone returns. A message log is on the bottom screen to report when the drone is taking off, delivering an item, % of route traveled. and error messages will appear when a error occurs: such as the drone being unconnected, or the drone seemingly being stuck.

4.1.3 Functional Requirements

In case of failure to complete a requirement on any given delivery, the fault will be reported to the user in the message log. In case the message log is not working, the program will refuse to launch a delivery, and a separate error box will pop up.

REQ-1: Ability to connect to drone through Wi-Fi
REQ-2: Ability to detect its own initial starting point
REQ-3: Ability to fly the distance between its starting point and an inputted address
REQ-4: Ability to land safely
REQ-5: Ability for messages concerning the drone's movement to appear on-screen.
REQ-6 Ability to grab current coords and return back to its initial starting using the method that REQ 3 represents.
takeoff() (detect if drone is off a certain charge, detect if drone is properly connected and start hovering)

initalizeStarting() set starting point
delivery(coordinates from the UI) (move from StartingPoint to Coords in parameter)
land() (stop moving, hover, then descend)

4.2 Log History

Description: Once a successful delivery has been concluded, where the drone has made the full round trip back to its Launch Zone, the customer's name, location of delivery, and medicine delivered should be stored in a log locally for future reference for the client to refer to, such as in case of refund fraud. The Store Manager, after filing away hardcopy records of said logs at closing time, should wipe the log history for the next day. [Priority: High]

- Benefit: 8 -Secondary feature of program, helps explicitly keep track of successful delivery and implicitly know when deliveries were unsuccessful.
- Risk: 7 - Storing customer information locally may have privacy risks.
- Penalty: 10 - Lack of this feature means that neither employees or customers have a record of the delivery, which may prove problematic if the medicine was stolen by a 3rd party, like in the case of regular package delivery.
- Cost: 3 - Little cost due to how small of a database a day's worth of logs would be, and that long-term storage is not an issue.

4.1.2 Stimulus/Response Sequences

After the drone returns, the message log should notify the employee operating the computer that a log was recorded of this delivery. At closing, the manager hits the Log button, verifies their status as a manager, and records the logs using pen and a hardcopy "Drone Delivery Record: (Date)" page that the company personally prints. After that is done, the manager hits the Clear buttons, verifies themself again, and the log is wiped.

4.1.3 Functional Requirements

If the log is not properly sent to the Log Storage, a pop-up error occurs. An employee should notify the Store Manager, and they will manually write the log in the Drone Delivery Record, and the employee operating at the time will put their signature alongside their record, as proof of existence.

REQ-1: Ability to store customer information (name, address, medicine)
REQ-2: Ability to be only accessed by a manager by entering their managerID
REQ-3: Ability to wipe the log only after re-verifying they are a manager.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

For performance requirements, the time between the drone taking off when the "Deliver" or "Return" button is hit should be as quick as possible, within 2 minutes, with an emphasis on "Return" being fast in case of emergency. Messages that pop up in the Message Box should be near immediate to the event they are responding to, within 5 seconds.

5.2 Safety Requirements

There are several safety concerns with the program, such as it getting lost due to not setting it in its proper launch zone when taking off, dropping the medicine, landing improperly and receiving damages, or getting struck or hit by obstacles enroute to delivery. It is expected for employees to decline or delay deliveries via drone if weather conditions are too severe for the drone to operate safely. They are expected to always place the drone exactly in its launch zone before hitting delivery. The "Return" button should also be able to be pressed not only at the end of a flight, but enroute as well.

5.3 Security Requirements

The Log feature saves the name, place of delivery, and medicine type for a given successful delivery. As the addresses of our client's customers are sensitive information, only the Store Manager has access to the Log. The Store Manager has to input their employee ID in order to access the Log, and it is a responsibility of their to wipe the Log daily as a security measure.

5.4 Software Quality Attributes

As this program should be used by employees, usability should be a must. Employees who are serving as Operators should be able to grasp the product's mechanics and be able to operate it smoothly by a single person. The correctness and reliability of routing is critically important, given that customers receiving important medicine will be delayed or not happen if the drone gets lost.

5.5 Business Rules

As stated earlier, only Managers should have access to the Log.

6. Other Requirements

Appendix A: Glossary

Address: not a literal address, but coordinates or a series of steps to be taken to a certain location in relation to the drone's launch zone.
Place of delivery: refers to the real-world location that the "address" is pointing to

Launch Zone: where the drone takes off. The drone should always be in the Launch Zone before starting the program
Client: A pharmaceutical retail store.
Customers: refers to the client's customers who want their medicine to be delivered by drone.

Appendix B: Analysis Models

A rough of what the intended UI looks like during deliveries: