PizzaDronz Requirements Document

Matthew Davidson

January 27, 2023

1 Functional Requirements

ID	Requirement	Priority	Comments
F1	The system shall generate a series of paths that don't enter "no fly zones"	high	
F2	The system shall correctly validate customer's card details	high	Card number, expiry date and CVV
F3	The system shall generate a series of paths that total less than 2000 moves	high	
F4	The system shall generate three files containing the paths taken for the day given when executing the program	high	one containing the orders placed/delivered, one with the paths in a `.geojson` format and one with the paths in a `.json` format
F5	The system shall always exit after generating a flight path for a given day	high	
F6	The visibility graph shall always be correctly generated	high	part of the pathfinding subsystem
F7	The card verification subsystem shall correctly validate expiry dates	high	
F8	The card verification subsystem shall correctly validate card numbers	high	just checking length, not performing the luhn algorithm
F9	The card verification subsystem shall correctly validate a card's CVV number	high	just checking length
⋮		⋮	⋮

2 Qualitative Requirements

ID	Requirement	Priority	Comments
Q1	The system shall generate the necessary paths in under 2 minutes	high	
Q2	The system shall generate a valid path if a valid path exists	medium	
Q3	The system shall be available 95% of the time	medium	
Q4	The system shall use no more than 4GB of system memory	low	
⋮		⋮	⋮

3 Test Approach for chosen attributes

ID	Test Approach
F1	Take the generated flight path and ensure that none of the moves generated cross the edge of a no fly zone
F2	This shall be verified by testing *F7*, *F8* and *F9*
F3	Take the output file and count the number of moves taken. This should always be under 2000
F4	Check that three unique files are created after running the program, all with different contents also
F5	Verify that the program terminates after passing in valid input data
F6	Hard to verify for all instances, potential approach is to just create a few test cases (including some edge cases) and check that these are handled correctly
F7	check that the card expiry date is not before the current date
F8	Check that the length of the card number is exactly 16 digits long
F9	Verify that the CVV is exactly 3 digits long
Q1	Test the system on various inputs and ensure that the runtime **never** exceeds 2 minutes
Q2	Find edge cases and test these as an input, making sure that the paths produced are valid
Q3	Measure the uptime of the system and verify that it is up 95% of the time
Q4	Measure the amount of system memory used by the system and check that it never goes above 4GB