Deliver Secure and Fast (DSF) Requirements Specification

[redacted]

Contents

1. Introduction (Skip this! i.e., you do not need to write for this section)

[Briefly introduce the document, i.e., what is the purpose and contents of this document. Also briefly introduce the system whose requirements are specified in the document]

2. System Requirements (Maximum 2 pages)

Stakeholders:

- **Customers Individuals**: They want to send packages to their family, their friends or to individuals met through online services (Blocket, Marketplace, Vinted) to sell a product they already own.
- **Delivery staff**: They want to do their job as flawlessly as possible with the least stress possible and have the least logistical problems on their way. They want to finish on time and have no problem with the deliveries if they are hired.
- **Google Maps Location service** (API mapping/addressing system): It wants to provide a reliable GPS system service that helps users to find addresses of individuals and businesses precisely.

Functional requirements:

FR1: The system should create, print and manifest order with ease displaying sending address and receiving address.
FR2: The system should alert the customer for a delivery attempt by means decided by the user some time before the delivery..
FR3: The system should keep the package in case of non-delivery in order to plan a future delivery or send the package back.
FR4: The system should provide an individual account for individuals with user authentication using password and email.
FR5: The system should provide a business account for businesses with authentication using password and email.
FR6: The system should be accessible everywhere at any time on different existing devices commonly used.
FR7: The system should fetch and update information about the parcels as often as possible during the process of the delivery using sensors and other devices.
FR8: The system should make possible the schedule of a collection by user while ordering the shipment of a parcel.
FR9: The system should have the capacity to find any address in Sweden.
FR10: The system should inform the customer in case of non-delivery by any means decided by the user and the date and time of the delivery.

Non-Functional requirements:

FRN1: The system should be capable of generating labels complying with international laws/standards.
FRN2: The parcels should only be scanned by expert staff that undergone an official security check at the collection point
FRN3: The system should keep a package that has not been picked up for 7 days before sending it back to the sender.
FRN4: The delivery of packages can happen during normal weekdays, evenings and saturdays.

Checklist-based requirement analysis:

According to our experience, here is the check-list we elaborated:
Previous requirement : What previous requirements are needed to apply this requirement?
Additional hardware : Does this requirement include investing in additional hardware?
Requirement example : Can you find examples of this requirement being implemented in another system?
Absence of requirement : Would the main purpose of the system still be the same if this requirement was not present? What change would it make if this requirement was missing?
Design Method: What design method/technique/tools can be thought of before the implementation of this requirement?
Business goal: Does this requirement respect the business objectives of the company? If yes, please explain which ones.
Future requirement: Does this requirement implies other obvious requirements? If yes, which ones?
Requirement testability: Is this requirement testable? If yes, in what ways can the engineer test this requirement?

We made the analysis by creating an excel sheet where questions were listed and by treating all the questions requirement by requirement. The sheet is present in the appendix. During the analysis, we ran into several problems such as:

- Requirements being too vague and needing to be redefined
- Questions were limiting the thought process by their simplicity. The lack of experience attenuates its objective.
- It was complicated to prevent concrete problems that way.

Systemic Validation:

We listed all the requirements we had and submitted them to a series of questions. We graded them on a scale from 0 to 5. If the score was important, the requirement would be classified as implementable, otherwise not. Here is the list of questions :

- Do the competitors already have this requirement?
- Will it affect the business value by not having this requirement?
- Is this requirement a bonus or a necessity?
- Is this requirement expensive to implement and how?
- Will this requirement provide a competitive advantage? “Yes”, “No”, “Maybe”
- Is this requirement explicit enough to not have to discuss it more with the client?
- Is there some substitute way of achieving this requirement?

After the analysis and having all the scores, we addressed some color codes to each grade in order to differentiate the implementable requirements to the ones that needed further thoughts. We figured out that 2 requirements needed to be implemented while others should be discussed further: FR7 and FR8.
An important problem we ran into was that the validation did not gather several stakeholders to address the problem.

3. System Interfaces (Specify Maximum two interfaces for each of the subsections below – One and half page approximately)

3.1. User Interfaces

3.1.1. Look & Feel

Colors: The colors that will be used are Electric blue (#0C48F1), White (#F4F4F4), Blue pale (#6F8DDE), Grey-Blue (#7885AB), Grey (#303645) and Light Dark(#1A2033). The emphasis on the color blue will give the user the feeling of efficiency.

Style: The style of the design will be "flat". It will have an emphasis on minimalism. It is aimed to provide a feeling of efficiency and ease to use.

3.1.2. Layout and Navigation Requirements

Major Screen Areas:

1. **Homepage:** This will contain the following section
   a. Login/Registration Section
   b. Parcel Sending
   c. Parcel Tracking
   d. Popular services
   e. Last articles (SEO articles to attract traffic)
2. **Parcel Sending Section**
   a. Parcel Details Form
   b. Delivery Options
   c. Payment Details and Confirmation

3.1.3. Consistency

Navigation Controls - by maintaining consistent navigation controls such as menus and buttons across all sections of the interface, ensuring predictability and ease of use. Using standardized terminology and language for elements to ease understanding and keep it consistent throughout the system.

Screen Layout - using a consistent layout, placement and structuring will improve usability and promote familiarity. By also standardizing the entering and presentation of data fields and controls to enhance user familiarity and efficiency.

3.1.4. User Personalization & Customization Requirements

User Profiles - allow and enable users to create and manage profiles with personalized settings, such a s contact information, language preferences and account type. Users should be able to update and modify their profiles as needed.

Customizable Content - allow users to customize displayed content. This may include adjusting settings related to the layout or presentation of information based on their needs and preferences

3.2. Interfaces to External Systems or Devices

3.2.1. Software Interfaces

Google Maps Integration - Integrate Google Maps API for address validation, route optimization and location tracking within the DSF interface. The data should adhere to the format specified by the Google Maps API. Should utilize HTTP protocols for data transmission.

Data Visualization Cloud Services - Provide integration with a data visualization cloud service for business customers to view parcel information and analytics, example Google Data Studio. Common formats include JSON, CSV or XML.

3.2.2. Hardware Interfaces

Scanning of parcels in house: DSF would use some scanner Honeywell CK65 for its operations. The scanner is connected to the local network by 1) communicating through wifi with its charging station 2) It's charging station being connected to the LAN. It sends information about: the serial number of the parcel, the date and the location in a JSON format.

Scanning of parcels during delivery: Delivery staff have an Opticon H-33 that allows them to update the status of the parcels during delivery. The hardware communicates with software by sending status, date, location and parcel number directly to the app, with staff access, using JSON formatting.

3.2.3. Communications Interfaces

Wi-Fi, Bluetooth and LAN: Scanners will communicate on the network using Wi-Fi with their own charging station. Their charging station will communicate to the software using LAN. Complementary devices, such as Datalogic hand scanner, will use bluetooth to communicate with a station connected to Wi-Fi.

4G and 5G: Delivery staff, during their work will use mobile phones to update the status of the parcels. They will communicate through 4G or 5G, that will reach the ISP and send information to the DSF servers.

4. Business Rules (Maximum 3 – No more than 1 page)

4.1. User Account Setup (ID: 1)

Conditions:

- Customers must set up an account, business or individual to be able to use the DSF services.

- All customer types must provide email addresses for their account creation.

- During account setup, the user must choose whether they want a monthly or a yearly subscription plan.

Actions:

- Allow customers to create accounts based on their business or individual status.

- Collect and validate email addresses during the account creation process.

- Provide options for monthly or yearly subscription plans during the account creation process.

4.2. Delivery Notification and Redelivery (ID: 2)

Conditions:

- If DSF is unable to deliver a parcel or if the customer is unavailable, and after three delivery attempts, delivery has not been made.

- If a customer requests redelivery or a different arrangement for delivery.

Actions:

- Leave a delivery notification for the customer with instructions on how to collect the parcel or rearrange the delivery.

- Hold the undelivered parcel for up to seven days before returning it.

4.3. Subscription Fee Calculation (ID: 3)

Conditions:

- Based on the total volume of parcels sent, calculate the unit parcel rate and annual subscription fee.

Actions:

- Adjust the unit parcel rate and annual subscription fee accordingly.

5. System Constraints (Maximum 3 – Approximately Half Page)

Constraints:

1. **Cloud Based**
   The system must be deployed on a cloud infrastructure such as Amazon Web Services (AWs) or Google Cloud Platform (GCP).
   This constraint ensures that the system must be designed and developed to leverage cloud based services and resources.
   Adopting a cloud based model offers several advantages such as scalability, flexibility and accessibility.

2. **Primary Language**
   The primary programming language for the system development must be Java. This constraint dictates that the majority of the system's codebase should be written in the Java programming language. Using Java offers platform independence, extensive libraries and frameworks, and scalability.

3. **Database**
   The system must utilize MySQL as the database management system for data storage and management. Thus ensuring structured data storing, enforcing data integrity and supporting complex queries.
   MySQL is a widely adopted database management system, open source, known for its performance, scalability and reliability. It integrates well with popular development frameworks and tools.

6. Use-Cases

6.1 Use-Case: Individual sending a traceable parcel

6.1.1 Brief Description

An individual customer wishes to send a package using DSF services, it goes through the process of sending it.

6.1.2 Actor Brief Descriptions

The actor is an individual that aims to send a package. He/she is used to sending packages (as the user is ready to send at least 10 during the year) and aims to have a look on where its package is.

6.1.3 Preconditions

The individual needs to have an individual account at DFS that stipulates he can send a package. He must have his personal information a way to pay.

6.1.4 Basic Flow of Events

1. The use case begins when the individual decides to send a packet.
2. The person goes on the website of DSF and connects to its account.
3. The individual chooses between an international parcel or a local (Swedish) parcel.
4. After a time requesting different APIs to recognize the address through the search, the system displays different possible choices:
   - Pick up points: if chosen, a google map integration appears with all the drop-off points near the specified address.
   - The specified address.
5. The individual chooses between different time frames. From the cheapest and longest to the fastest and more expensive.
6. The customer decides the level of tracing. The more tracing, the more expensive.
7. The customer chooses an insurance type in case the product is broken or stolen. The user needs to evaluate the value of the good.
8. The customer indicates if the product is fragile.
9. The customer writes the weight of the package. There is an alternative choice for a package of different size (such a bike for instance). The user has a table of price per weight in case he can’t correctly evaluate the weight of the package.
10. The customer notifies when and where he wants the package to be picked up. Depending on the service paid, several choices are offered to him such as someone coming, to leave somewhere, or need to be present at a certain time at his address.

11. The customer pays the amount displayed for the service by entering his credit card information or by using Swish.

6.1.5 Alternative Flows

1 - *The customer does not have an account and wants to create one*:
If in step <2> of the basic flow the individual does not have an account, then

a. The individual needs to create an account and enter all of its coordinates such as email, mobile phone, address and payment.

b. The use case resumes at step <2>

2 - *The customer does not have an account and does not want to create one:*
If in step <2> of the basic flow the individual does not have an account, then

a. A button "Continue without an account is shown"
b. The use case resumes at step <3>
c. The user is asked to enter all of his information later at step 10 or 11.

6.1.6 Post-conditions

The client needs to connect to the DSF app to track the package.

6.2 Use-Case: Business sending a traceable parcel

6.2.1 Brief Description

This use case describes the process for businesses like H&M, Ikea, or Amazon to send traceable parcels through the DFS system. The businesses will have access to various delivery options, including autonomous drone delivery, instant home delivery within 1-2 days and delivery to nearby collection points.

6.2.1 Actor Brief Descriptions

Business Customer/Client: Represents the company which wants to send parcels using the DFS system.
DFS Delivery Personnel: The delivery staff, including drones, responsible for picking up parcels from businesses.
DFS System: The digital platform that manages parcel delivery operations.

6.2.3 Preconditions

The business customer must have a registered account with DSF ensuring that they are able to send packages.
The business customer must meet the minimum annual parcel delivery requirement.
The parcel should be packaged according to DFS standards.

6.2.4 Basic Flow of Events

1. *Log In & Selection of Delivery Options :* The business customer accesses the DFS system and logs into their account. Navigates to the parcel sending section and chooses the option to send a parcel.

2. *Parcel Details :* The business customer provides parcel details such as dimensions, weight and contents ensuring compliance with international customs requirements.
3. *Parcel Delivery Options* : Navigates to the delivery option section where the business customer selects their preferred delivery method from a range of options, including autonomous drone delivery and instant home delivery.
4. *Confirmations and Payment :* The business customer reviews the order details and proceeds to make the payment for the delivery service.
5. *Parcel Pickup and Processing :* DFS delivery personnel receive notification of the parcel pickup request. They visit the business premises at the scheduled time to collect the parcel.
6. *Delivery to Recipient :* Depending on the selected delivery method, the parcel is dispatched for delivery via the chosen mode of transportation.

6.2.5 Alternatives Flows

1 - The customer does not have an account:
If in step 1 of the basic flow the business customer does not have an account, they need to create an account and provide all details accordingly such as email, phone number, address and payment details. After creation, the customer must start again from step 1 and log in.

2 - The chosen delivery is not possible:
If the selected delivery method in step 3, such as autonomous drone delivery, is not feasible due to weather conditions or regulatory restrictions, the DFS system automatically offers alternative delivery options such that the parcel reaches its destination on time. After a delivery option has been selected it continues to step 4.

6.2.6 Post-conditions

The business customer receives confirmation of the successful pickup and dispatch of their parcel, with real-time tracking available through the DSF platform.
The recipient receives the parcel within the specified time, with live updates provided by the DSF platform and tracking system.

6.3 Use-Case: DSF staff returning an undelivered parcel

6.3.1 Brief Description

This use case describes what happens when a customer does not pick up its parcel at a pick-up point (here a gas station) and the package is sent back to the sender.

6.3.2 Actors Brief Descriptions

DSF delivery person: He’s the person in charge of delivering and picking up the package again.
Gas station employee: The person in charge of scanning the article at the pickup point. This allows it to send notification to both the sender and the receiver and update the package’s status.
Sender: The person who has sent the package.

Receiver: The person in charge of receiving the package.

6.3.3 Preconditions

A package was sent, the fees were paid and the package arrived at the pick up point. The employee from the pick up point scanned the article. The package stayed for 7 days and no one came despite messages and email being sent.

6.3.4 Basic Flow of Events

1. The employee from the gas station scans the package on the seventh day.
2. The scanner indicates that this package has been present for 7 days without being picked up and needs to go back to DSF services.
3. A DSF delivery person arrives at the pick up point for the delivery of the day. (If this is a working day)
4. The DSF delivery person scans the package and places it in the truck.
5. The DSF delivery person brings back the package to the warehouse. There, it is scanned and placed in the queue to be sent back to the customer.
6. The package is sent back to the original sender's warehouse closest to the sender's home. The rapidity of the process depends on the initial plan bought by the sender.
7. A notification is sent to the sender that a delivery will be made. (Or he needs to pick up at a pick up point)
8. The package is delivered the next day.

6.2.5 Alternatives Flows

a. At step 2, the customer comes to pick up the package before the delivery person comes. Use case cancels and the delivery person needs to get updated. Do not resume.

b. At step 3, the delivery staff receives a message during a day with no delivery. The delivery staff will take the package the next working day. Resumes at step 3.

c. At step 7, the receiver complains (through the receiver or not) that he did not receive the package's notification. A ticket is opened. A new delivery is planned. A confirmation of information is established. New use case to create. Do not resume.

6.2.6 Post-conditions

The operation is successful if the sender has taken back its package. If this is not the case, a special requirement for lost package handling needs to be evaluated.

6.2.7. Special Requirements

Both the sender and the receiver need to be informed of their packet's status through email and/or sms once the decision of sending it back has been taken.

Appendix – Time Report

Date	Member	Activity	Time (hours)
2/2/2024	André	Business Rules	2
3/2/2024	Benoit	System Requirement	3
5/2/2024	Benoit	System interface	1.5
6/2/2024	Benoit	Use case 1	2
6/2/2024	André	System Constraints	2
7/2/2024	Benoit	Use case 3	2
7/2/2024	André	Use Case 2	2.5
8/2/2024	André	System Interface	3
8/2/2024	Benoit	Use case	2
11/2/2024	Benoit	Fixes	1
12/2/2024	Benoit	Last details	3.5
12/2/2024	André	Last details	1

Appendix - Checklist Based Requirement Analysis

Checklist-based requirement analysis					
	FR1	**FR2**	**FR3**	**FR4**	**FR5**
What previous requirements are needed to apply this one?	Adaptive system to international standards for delivery. Knowledge of cost for sending across the world	The system needs to be able to trace the supposed delivery	Needs a set of policies defining what is a delivery that did not finished	The capacity to have an interface that is nuanced	The capacity to have an interface that is nuanced
Does this requirement include investing in additional hardware/equipment?	Printer being able to print a lap for any country	Not necessarily if the delivery person has a way to communicate the update of the delivery	This requirement request the traceability of package and more warehouse space	Other saas have those kind of implementation/plans	Other saas have those kind of implementation/plans
Can you find examples of this requirement being implemented in another system?	Yes at some other delivery companies	Possible to get the same results using other delivery services	Competitions	No because it would restrict the service only to business customers and then deprive company from market segments	No because it would restrict the service only to individual customers and then deprive company from market segments
Would the main purpose of the system still be the same if this requirement was not	The scope of the system would be reduced and	The main purpose would still be the same but	This would lower the quality of service	Yes it would but the scope of the	Yes it would but the scope of the

present? What change would it make if this requirement was missing?	the attractivity for the service as well	the quality of the customer experience would be reduced	drastically	business would arguably be reduced	business would drastically be reduced
What design method/technique/tools can be thought of before the implementation of this requirement?	Database of difference between postal system, 3 party software implementing this functionality	3rd party message sender	Inventory management and	UML can prevent the design of such a requirement by implementing inheritance in the user interface	UML can prevent the design of such a requirement by implementing inheritance in the user interface
Does this requirement respect the business objectives of the company? If yes, please explain which one.	"Sending parcel internationally"	Yes as it ensures a reliable service	Yes by insuring a good quality of service expected by the customer	Yes because the business aims to serve both individuals and companies	Yes because the business aims to serve both individuals and companies
Does this requirement imply other obvious requirements? If yes, which ones?	This requirements implies to be able to track packet delivery in the world by either working with other platform or doing that independently	This requirement does not imply obvious requirement	The requirement of tracking the deliveries, time a package has been kept	This requirement implies to have different scope on the entire system	This requirement implies to have different scope on the entire system
Is this requirement testable? If yes, in what ways can the engineer test this	Possibility to test the outcome according to every country	Yes it is possible to automatically test this requirement	This requirement is testable by getting a package	This requirement is testable by a session, not	This requirement is testable by a session,

requirement?	the system wants to deliver to		status	by automatic tests	not by automatic tests

FR6	FR7	FR8	FR9	FR10
The UI should be adapted for different devices	The system should be able to get receive updates from different sources of updates for the packages	There should be a page or an app to access this service	The system needs to have access to a database of addresses in Sweden	There need to be an effective way to trace delivery and notes those missed
This requirement necessities more servers	The system will have to invest in sensors as well as scanners. The system will have to invest in satellite communication .	Not necessarily, but on additional software capabilities	This can eventually include precise GPS, smartphone app	No but the ability to communicate with the customer
The competition has apps and accessible websites	Update by scanner is already implemented but not by sensors.	Some competitors may have this functionality, not seen yet	Google maps and GPS are good examples	Yes through mobile phone messages or email sending
Not really as it wouldn't be aligned with what other competitors do	The main purpose of the system would still be the same. The capacity to get information on the parcel, and then comfort the customer would be lower	The main purpose would still be the same. It would make the deliveries more effective maybe	No because a delivery system is expected to be precise and to deliver at any location. The company would lost its credibility	The main purpose would be the same but the customer would have less available information hence less satisfaction

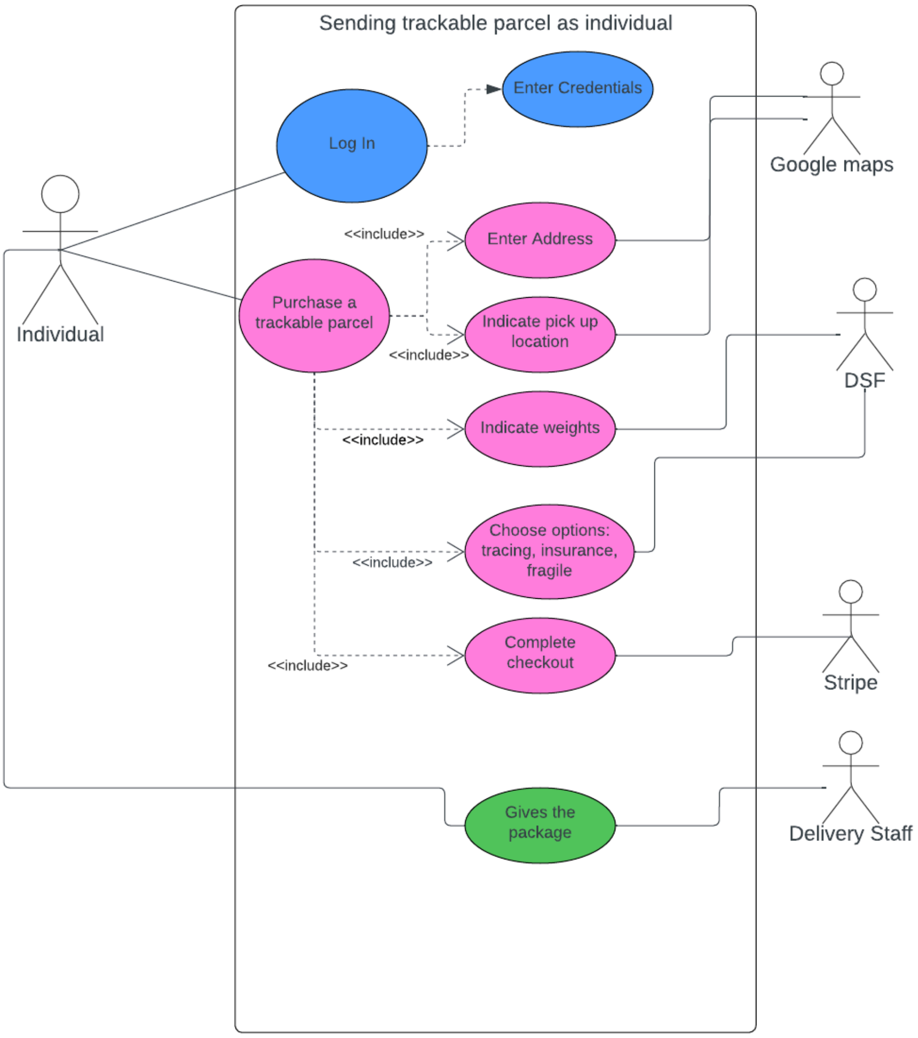
Responsive design, response automation using AI for customer service	Distance communication , IoT devices	A capacity to change the deliveries itineraries according to the planned schedules	Google maps and GPS technology	Tools for communicating with the customer
Yes by insuring a good quality of service expected by the customer	This requirement provide a more reliable service	Yes because it provides a more comfortable service and tackles a common problems of the deliveries (not being present during the delivery)	Yes as it provides more reliability on the system hence more satisfaction	Yes as it increases reliability
Implies the capacity to reponds to customer service 24/7 and to maintain the servers running 24/7	This requirement will constraint the system to deal with an increased amount of information hence implies the capacity to treat this information	This requirement implies to schedule deliveries around people's available time. This can be a big logistic constraint for drivers. It can have impact of effectiveness of deliveries.	Integration of 3rd party software inside the system + being dependent on this software	It implied to communicate with the customer
This requirement is testable by testing the quality of service at every moment of the day	This is testable, the engineer can try fetching information from package at any time	Engineer can test reschedule of delivery round according to constraints	Yes by finding addresses maybe?	Yes by sending notification/messages/email when a delivery is made

Appendix - Systemic Validation

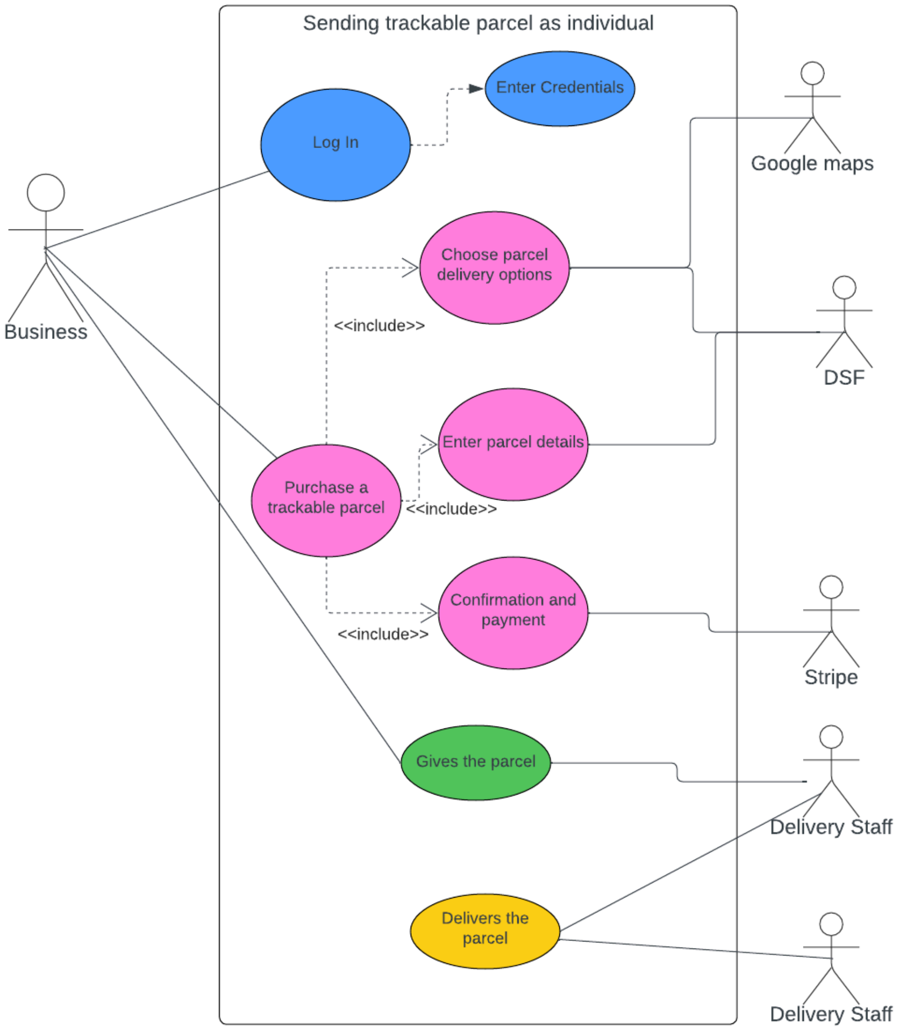
Systemic Validation										
	FR	*FR*	*FR*	*FR*	*FR*	*FR*	*FR*	*FR*	*FR*	*FR10*

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	
Does the competition already have this requirement? (0 no, 5 yes)	4	5	5	4	5	4	2	2	4	5
Will it affect the business to not have this requirement? (0 no, 5 yes)	5	5	5	2	5	5	1	2	4	5
Is this requirement a bonus or a necessity? (0 bonus, 5 necessity)	4	5	5	3	5	5	2	3	4	5
Is this requirement expensive to implement? (0 expensive, 5 not expensive)	3	4	5	3	5	3	2	1	4	4
Will this requirement provide a competitive advantage? (0 no, 5 yes)	2	1	0	3	5	4	3	4	3	3
Is this requirement explicit enough to not have to discuss it more with the client? (0 no, 5 yes)	4	1	4	3	5	3	2	2	5	4
Is there some substitute way of achieving this requirement? (0 yes, 5 no)	4	3	4	5	5	2	2	2	3	4
Score	26	24	28	23	35	26	14	16	27	30

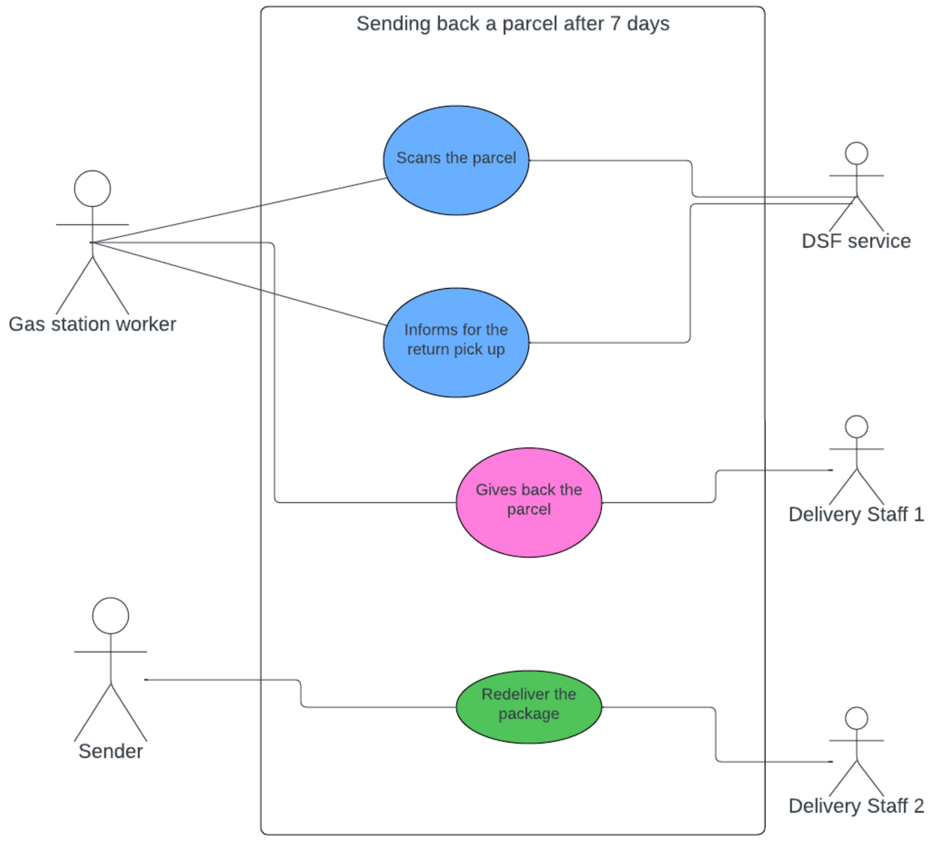
Appendix - Use Case 1



Appendix - Use Case 2



Appendix - Use Case 3



Assumptions

- DSF can be used the same way a service like PostNord is used to send packages as it is targeted for individuals.
- We assumed that delivery staff was staff employed by DSF (in comparison with contractors that can be hired per delivery).
- We assumed the stakeholders' opinions on the systemic validation
- We assumed the color of the application
- We assumed the style of the application according to our own tastes
- We assumed that the staff would use several types of hardware such as a Honeywell scanner (link) and an Opticon scanner (link). Those assumptions came from our own experience working in warehouse environments.
- We assumed that the communication between hardware devices and the software would be JSON.
- We assumed the hardware devices would use Bluetooth, Wifi, 4G and 5G to communicate.
- We assumed that the primary language the software would use is Java.
- We assumed that the software would set up a MySQL database as the primary database.
- We assumed the sales' process for the basic flow of events for use case 1.
- We assumed that there would be an option for packages with a special form (such as Bike, snowboard for instance).
- We assumed it was possible to create an account without taking into consideration the 10 packages limit.
- We assumed that the pick up point for use case 3 was a gas station and that the employee of the gas station would be the one updating the status of the package directly to DSF. We also assumed that they would scan the package every day. Even the one from previous days.
- We assume payment could be made through Stripe in the use case diagram one.
- We assumed that DSF would have some SEO articles on its webpage.
- We assumed that the DSF experience would be similar to what we already knew with services such as Amazon or AliExpress.
- We assumed that it was possible to pay by Swish for an individual.