Systems Engineering — Triad FOXTROT

Food Delivery using Drones in Congested Cities

ORD – Deliverable 4

Avci, Ayça	s4505972
Adelekan, Iyanu	s4890019
Badescu, Andrei	s5011124
Menschikov, Dmitrii	s5125626
Amiri, Zamir	s2780542
Baltatzis, Athanasios	s4076206
Joshy, Rahul	s4421833
van der Meide, Jonah	s3242390
de Groot, Robbin	s3376508

rijksuniversiteit groningen

CONTENTS

1 HIGH-CONCEPT

1.1 SYSTEM INTRODUCTION

This document is a part of a series of systems design documents containing requirements and architectures of an aerial drone delivery system in Mumbai, that should replace a regular human-involved delivery as a more accessible and cheap approach. Firstly we provide the main use case of the system, its benefits and then we present stakeholders and the goals of our system.

1.1.1 *Aerial drone delivery concept*

Drones have been used as an alternative to the couriers since 2014 and are used now in such countries as the US, Australia, Finland [1], Canada [2], etc. Drones don't require a driver and automatically deliver packages even to hardly accessible areas. The drones are capable of delivering packages weighing up to 3 kg [3], so it is possible for them to carry several packages at a time.

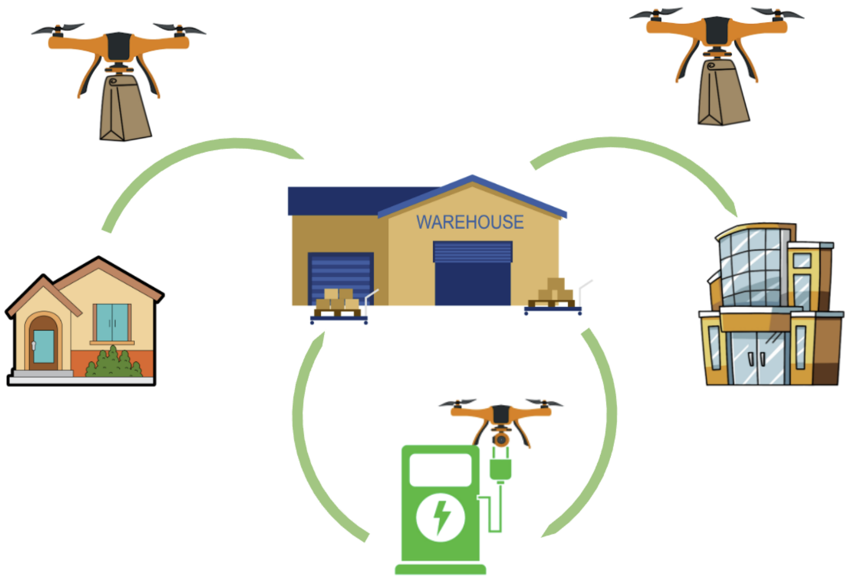

Figure 1: Sketch of the concept of drone food delivery.

In such system the customer makes an order via phone or on-site, then the package is being dispatched using a special package designed to be hung to a drone. Such drones use a VTOL system: Vertical Take-Off and Landing. Next, the drone navigates to the destination point via GPS, cameras and sound sensors. The food package is being handed over to the customer and then the drone either delivers the rest of the packages or heads to the base. There the drone is being inspected and prepared for the next delivery.

Nevertheless, such approach is still not fully investigated in terms of system design and such gaps as, for example, protection from drone stealing, strong winds and flying objects (especially other drones) avoiding still need to be solved and described. This series of design documents will analyse the design aspects that are necessary for the drone food delivery system in Mumbai.

1.1.2 *Motivation – Facilitate meal delivery at any place, any time*

From early 90's, it was a common sight in India to see tiffin whalas (delivery boys) moving around on their bicycles loaded with multiple tiffin boxes (hot home-cooked cargo) on the chaotic streets of Mumbai. They were responsible for delivering piping hot home-cooked lunches to more than 200,000 busy office workers. Most of these workers lives 50 kilometres or more from their workplace which involes a long commute on a packed train. There is certainly not time for the cook of the house to prepare a full meal before they leave home. So the lunch-filled tiffin boxes are picked up later in the morning, colour-coded and transported to the station, where they are collected by the tiffin wallahs, whose mission is to deliver each box to its corresponding workplace still hot from the pan – and to return the empty tiffin to the home before the end of the working day [4].

(a) An example of a tiffin box that is traditionally used in Mumbai. [source]

(b) A group of tiffin whalas on delivery. [source]

Figure 2: Examples of a tiffin box and the whalas delivering them.

This practise is still continued in Mumbai but with the rapid increase in vehicles, the mobility of these workers is becoming restricted day by day making it less punctual and unreliable. Labour cost being cheap was one of the main attraction but with rising prices in daily goods has seen a demand of increased wages by the workers. A need for alternate system is required to combat the existing problems.
Aside from the traffic congestion, and increase of labour costs, recent event's such as the coronavirus crisis has show us the importance of automated delivery systems that do not require contact between multiple people. Contact between people is especially hard to avoid in large cities such as Mumbai, which currently has 21 million inhabitants and is estimated to have more than 27 million inhabitants by the end of this decade. An automated delivery system would greatly help with crisis's such as the COVID-pandemic in the future.

An automated drone delivery system is also preferred in some ways by the customers over the in-person deliveries. First of all as the drones will not make use of the road-system of the country, deliveries can be made significantly faster. Furthermore a customer could also send a meal to another persons address with this system, allowing parents to deliver meals to their children if required.

We should also not forget that many restaurants went out of business during the COVID pandemic. Some restaurants attempted to stop the financial bleeding by delivering customer orders. But not all restaurants had the means and the connections required to do so. Hence a drone delivery system would have been a great resource to have for these types of scenarios.

Another use case, for a well developed automatic drone delivery system, would be to transport cargo to islands. Countries such as Indonesia, that consists out of more then thirteen thousand islands, can greatly benefit from this system as they currently have to rely on small boats which are greatly dependent on the weather.

But perhaps the most important use-cases are the ones that have not been invented yet. Aerial transport of goods (through) air-planes have existed for many decades and was one of the first use-cases for air planes was to transport mail, because of its weight. As planes became more sturdy and could lift heavier load, the commercial flight sector was opened. Perhaps something similar is also possible with drones.

1.1.3 *Benefits of the system to be*

Since Mumbai is a highly populated city, which 21 million people live approximately, traffic congestion is a huge problem that needs to be tackled. Lack of parking areas and ongoing constructions are reason of this problem. We wanted to avoid dealing and adding to that problem while making food deliveries. Using drones for food delivery purposes is a faster, reliable and punctual option to avoid ongoing traffic. This approach also allows work-from-home concept as well since drones are needed to be navigated remotely instead of hiring food delivery people and ultimately removing the human operators as well after gathering enough data about the topology of city in order to enable fully autonomous deliveries. Usually food delivery comes with the cost of training the delivery people, providing proper uniforms for them, using vehicles such as motorbike and car that consumes fuel, providing navigation devices, suitable healthcare policy for each delivery person and so on. Mentioned approach reduces the total cost of workforce. It is also a environment-friendly option for the food delivery by preventing air pollution since drones do not consume non-renewable energy sources such as fuel or gas. It decreases the amount of traffic accidents arising from road/highway, involving fatal or major injuries for the delivery person. There is also a cultural aspect in Mumbai that people tend to prepare fresh/homemade lunch for the loved ones to deliver to their work places. Our approach is an alternative option for those people by allowing them to deliver food in an effortless and direct way thus contributing to keeping local traditions alive.

1.2 PROJECT MISSION

The primary scenario for the food delivery service using drones is to empower customers to seamlessly deliver food items to any location in Mumbai within minutes. Delivering food effectively across a large and highly populated city is a non-trivial problem that involves many interconnected processes and considerations.

This system is a pilot project whose design has been outsourced to system design team FOXTROT by DoorDash Incorporated to aid its expansion into Asia starting from India's city, Mumbai. DoorDash is an American company that operates an online food ordering and delivery platform. As of this moment, DoorDash operates solely in the united states but has plans for expansion into Asia.

Based on our discussions with DoorDash, we intend for the drone delivery service to cater to day to day users and businesses - specifically restaurants. Both user categories require speedy, safe, reliable, and cost-effective deliveries. The service has the following purpose and goal.

Purpose: Facilitate the fast, and efficient delivery of food from individuals and restaurants to target recipients across the city of Mumbai with drones.

Goal : To develop a fully functional food delivery system that is capable of reliably servicing the entire populace of Mumbai.

1.3 composition of the system-to-be

1.3.1 *Stakeholders*

Stakeholders who do interact with the system to be

The main stakeholder is the owner of the start-up company that creates the whole system and oversees all the parts of the system. The next three stakeholders are interconnected in the system and have to cooperate with each other to obtain the main goal of the system, which is to have a sustainable business delivering food via drones. The first two departments(drone operations and drone security) are within the company. While the restaurants are outside of the company.

Customers

Our customers are the contractors at DoorDash. They determine the performance requirements of the system, as well as the input and output limitations.

Consumers

The Restaurant owners and individuals who prepares food for their beloved ones are considered as the customers, and are the end users of our system. The implementation of the drone delivery project in Mumbai, would demand some major alternations of the existing procedures followed by restaurant owners in food delivery and order receipt. For instance, they should adopt the know-how of using this technology and try to cover the costs occurred. Furthermore, the restaurant owners have to communicate with the drone operations department about the implementation of pick-up points and charging stations. Similarly, the individuals might be wary of the new technology as they have no prior experience to such an application and ease of accessibility should be prioritised.

Stakeholders who do not interact with the system to be

Labour unions

Labour unions may oppose to an implementation of food delivery by drones, since a respectful amount of delivery employees might end up losing their occupation. Therefore, it is advised to take into account a stakeholder like that, in order to prevent obstacles that my affect the progress of the project.

Food delivery companies

Established food delivery companies such as Uber, Swiggy etc., will oppose the new system as they have already invested lot of resources for the already established delivering system (use of bikes). A collaboration with these companies is one way of avoiding conflicts.

Regulatory entities

Both the government and on a smaller level the municipalities where the drones will operate are stakeholders. They may impose new legislation for the benefit of both the regulatory entities and the companies (problem owner). Furthermore, these regulatory entities should implement security constraints, since both the drones and the general population might be vulnerable to hostile activity.

1.3.2 *System goals*

We will briefly outline the necessary characteristics of the system after deployment. These characteristics are concrete goals that are seen as make-or-break requirements for the system to operate. Thus, these characteristics need to be measured during testing and deployment.

At a glance, the objectives of our system:

1. To facilitate the rapid delivery of food to customers in Mumbai.
2. To provide reliable delivery services to local restaurants.
3. To provide users of the service with online and offline channels for placing and managing delivery requests.
4. To be robust and scalable enough to serve the entire Mumbai region.

These key objectives can then be translated into the following goals.

The system needs to be **accessible** for all its users. To schedule a delivery, one may consider an over-the-internet system to schedule a delivery. However, in the city of Mumbai, only 55.8% of the population has an internet connection [5]. We thus require that our system meets an accessibility quota of at least 99%. Therefore, we should consider alternatives that reach a larger fraction of our (potential) users.

With respect to online access, the system will provide mobile and web client applications to enable the submission of delivery requests, tracking of delivery statuses, and many more activities. Offline service support will be provided via a USSD channel and physical stores.

From a user point of view, the system needs to be **reliable** and **punctual**. If a delivery gets scheduled, the system needs to ensure that the delivery is completed at the expected time within some margin of error; no earlier and at most 10 minutes later than expected. This is less than the accepted standard deviation in delivery time in the United States is 13 minutes [6]. As well as that, the system needs to complete the delivery without damaging the package contents and maintaining the conditions of the food inside. That is, warm food should not be cooled, fresh food should not get stale, the humidity needs to be preserved, etc. Such a failure to deliver should occur at most in 1% deliveries to combat food waste. According to the United Nations, about 5% of food in food services is wasted [7]. Our system should undercut this. If the user wishes or reliability and punctuality are not met, the system should be able to provide resolution or compensation for this failure, and dispose of the wasted food.

The use of the system will likely vary over time. Demand will be lower at night than during the day, for example. Regardless of the time and place, the system needs to be **available** to users. A scheduled delivery should always be executed. The system needs to maintain a sufficient amount of drones to cover all usages of the system, as well as ensure that a pickup or delivery can be completed within the target area of Mumbai.

1.4 DOCUMENT CONTENTS

This document is the Operational/Stakeholders Requirements Document (ORD) of our system-to-be. It contains the stakeholder requirements elicitation, the interpretation into system requirements, a description of the operational concept, and a model of the system context. The purpose of this document is to communicate or findings and progress with our stakeholders, and thus the content is not necessarily final[1]. The transformation of system requirements to the functional architecture (FA), and those to the physical architecture (PA) are modelled in IDEF0 in the System Requirements Document (SysRD). For completeness, some of the contents of the SysRD are repeated in this ORD (see chapter 4 and chapter 5). Note that the definitions of the FA and PA in the SysRD are leading in case of discrepancies.

[1] For the deadline of deliverable D4, we assume the content to be final.

2 OPERATIONAL CONCEPT

2.1 OVERVIEW OF SCENARIOS

In developing a drone delivery system it is crucial to define its primary operational scenarios. Each scenario describes a way that a particular stakeholder will use the system. Operational concept scenarios also define how the system will interact with relevant inputs and outputs and other systems.

In the following sections, we firstly list key operational concept scenarios of our system, then describe them in detail. In the list, we employed a numbering scheme that uses a coded 'S' followed by a sequential numbering. The numbering in itself represents no hierarchy of the scenarios and no priority of the scenarios. It is solely a numbering scheme.

List of operational concept scenarios:

- S.001 - Drone monitoring
- S.002 - Drone inspection
- S.003 - Customer-Customer delivery processing
- S.004 - Restaurant-Customer delivery processing
- S.005 - Emergency fire control
- S.006 - Emergency landing and recovery protocols
- S.007 - System security
- S.008 - Policy compliance
- S.009 - Drone identification
- S.010 - Damage mitigation
- S.011 - Analytics
- S.012 - Feedback

S.001 - Drone monitoring
The system consists of warehouses that are fully equipped with tools necessary for the management and maintenance of drones. Highly trained staff a on-site ready to maintain the drones at a moments notice. An inventory of all drones is kept with an internal application. The application tracks the statuses of all drones (e.g idle, and delivering), their locations, battery level, and rate of power consumption. Information is streamed from the drones to the application via IoT technology. The drone operations staff monitor the drones and take action immediately its needed.

S.002 - Drone inspection
The operations team inspects drones daily to ensure that they are in a fully functional condition.

Considering our system will maintain an initial fleet of hundreds of drones, it will be tedious to inspect all of them solely manually. Instead, a data driven approach is taken, and crucial data about the state of the drones is constantly sent by the drones to the system's monitoring application. Algorithms then analyse the data and make decisions on which drones need to be checked.

Besides the automated approach to drone inspections, manual inspections of the drones will before and after every flight. Components like the propellers, outer shell, and landing gear will be checked. Based on the results of an inspection, a drone repair may follow. After a repair a drone will be returned back to the fleet.

S.003 - Customer-Customer delivery processing
Warehouses, and kiosks - collectively referred to as service points - are strategically located around the city of Mumbai to handle delivery processing. The service points will have trained personnel to handle delivery requests. A customer that chooses to visit a service point to get his food delivered is catered to by the onsite staff. Details of his food package such as its width, height, weight, and temperature are recorded. If the customer wants a scheduled delivery, the time at which the customer wants the package to be delivered is also recorded and the package is stored. Otherwise, the package is sent for delivery immediately.

S.004 - Restaurant-Customer delivery processing
The system offers features to facilitate restaurant to customer deliveries such as permitting restaurants to create delivery requests, realtime tracking of the status and location of packages, and requesting more drones during peak service periods. The drone delivery service bundles these features in a web client application which restaurants will be granted access to.

Drone landing stations will be installed in the premises of restaurants the stations will be located away from obstacles that can obstruct the landing and take-off of drones. In addition, the stations will have charging facilities for the drones. Drone compatible carrier packages will be located on premises too. A select group of restaurant staff will be trained on how to properly attach packages to the drones for delivery. When a package is mounted to a drone, the drone will immediately commence delivery to the target destination.

S.005 - Emergency fire control
In the highly unlikely case of a drone catching fire, automatic fire extinguishers, and smoke detectors will be placed at every drone station. When smoke is detected, the fire extinguisher will be triggered to extinguish the flames, and staff within the vicinity will be notified immediately.

Furthermore, all drones will be equipped with heat sensors which will send a realtime stream of temperature data to the service's backend. If a drone reaches an abnormal temperature, the drone operation staff will be notified immediately to take necessary action.

S.006 - Emergency landing and recovery protocols
Although the loss of aircraft control is generally an infrequent event, a flyaway may happen as a result of inference with the aircraft. To cater for such a situation, all drones in the fleet are equipped with an Attitude (ATTI) flight mode that disables GPS positioning and other visual positioning systems to allow for manual control of the drone to be regained by the drone operations department. We also will equip all drones with a Return to Home (RTH) mode that can be used in the unexpected situation that control of the drone cannot be regained after enabling ATTI. If both the forementioned fail, the operations team will note the drone's battery life, height, speed, and heading, after which they will inform the police and Air Traffic Control (ATC) of the situation. In all situations, an incident will be logged in an internal incident tracking tool and a post-mortem document will be created once the emergency has been handled.

S.007 - System security
It is essential to the integrity of the system to have real time data regarding the position, flight path, battery level, altitude, and other important data of each drone in the system. This implies rigorous security practices, a well designed and secure network in order to handle all the incoming data necessary for analysis and other big data processing and at least one employee responsible for maintaining each component of the system updated to the latest firmware, installing the latest security patches, checking network traffic and constantly monitoring the network activities.

S.008 - Policy compliance
As this field is still in it's infancy, state and government laws are still being developed, changed and implemented regularly. Therefore there must be someone within the company who is appointed with following the legal developments and regulations regarding aerial vehicles and if necessary report them or submit a proposal of needed changes accordingly, in order to keep the company compliant with regulatory entities.

S.009 - Drone identification
In order to maintain a healthy logistics chain and the integrity and security of the drone fleet, a system of identification needs to be implemented. Each drone must be registered into the system alongside it's distinct manufacturer identification numbers and it must be assigned an unique ID and registered into the system in order to be recognised from other drones, to deter any potential malicious intents and to simplify any other processes in the case of issues that might arise.

S.010 - Damage mitigation
In order to prevent possible liabilities, the security department should be prepared to handle any type of unpleasant incidents e.g. accidents, failures, disruptions, etc. . Therefore, it is the manager's responsibility to understand all types of implications that might occur and to prepare accordingly in order to properly manage and contain any situation that might arise as fast and efficient as possible in order to minimise the extent to which the company might be liable for.

S.011 - Analytics
Consumers are the backbone of this business and the data that is generated through their interaction with the system, is of the upmost importance in order to understand how the product is used, when are the times that the traffic is the most intense and what parts of the infrastructure need expanding and what metrics to take into consideration when thinking of adding new features.

2.2 EXTERNAL AND CONTEXT SYSTEMS

We give the functional architecture definition of the system as a A0 function in Figure 1. This gives us a basis for decomposition of our system. We then define the functions that form the context of our system. We obtain this primarily by reverse engineering existing delivery service systems.

Modelling the context gives us the the hierarchy of functions and systems as seen in Figure 2.

We then model the context as seen in Figure 4, Figure 6, and Figure 8.

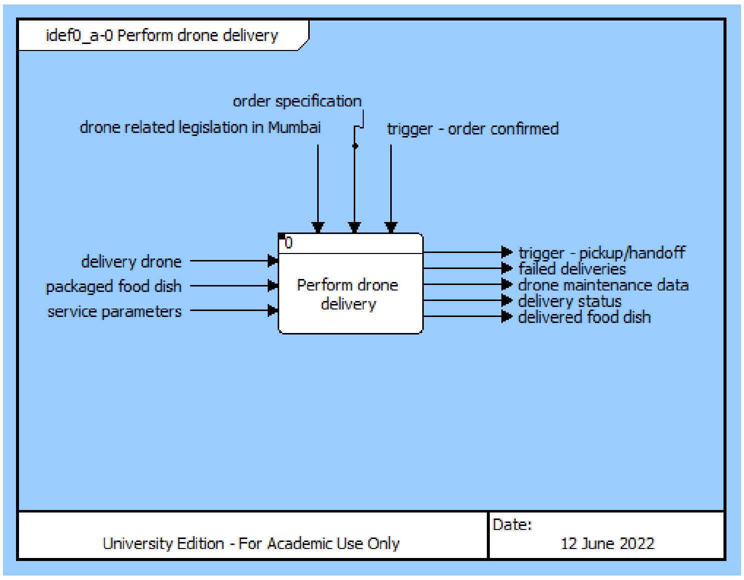

Figure 1: The A-0 diagram of level 0.

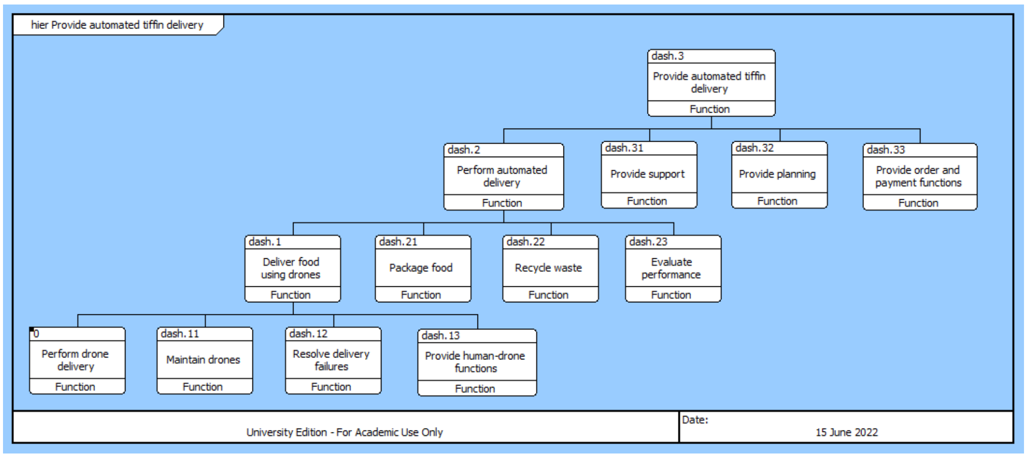

Figure 2: Hierarchy diagram of -3 to 0.

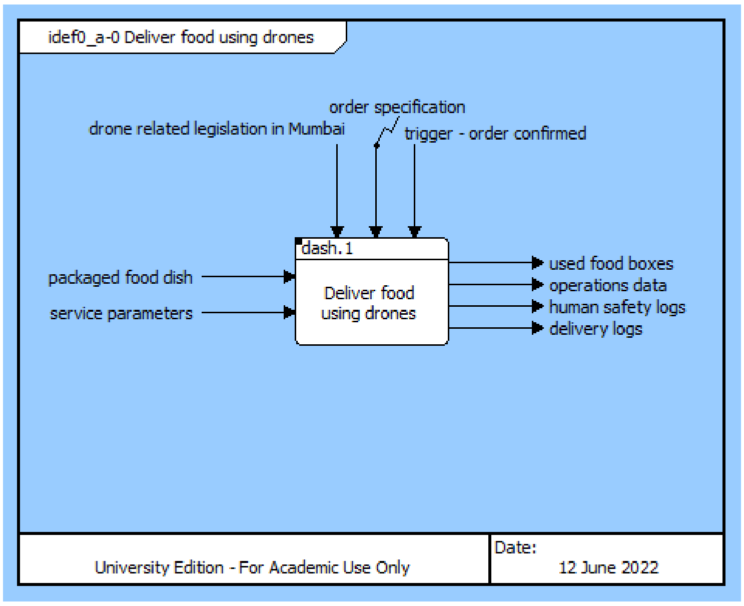

Figure 3: The A-0 diagram of function -1.

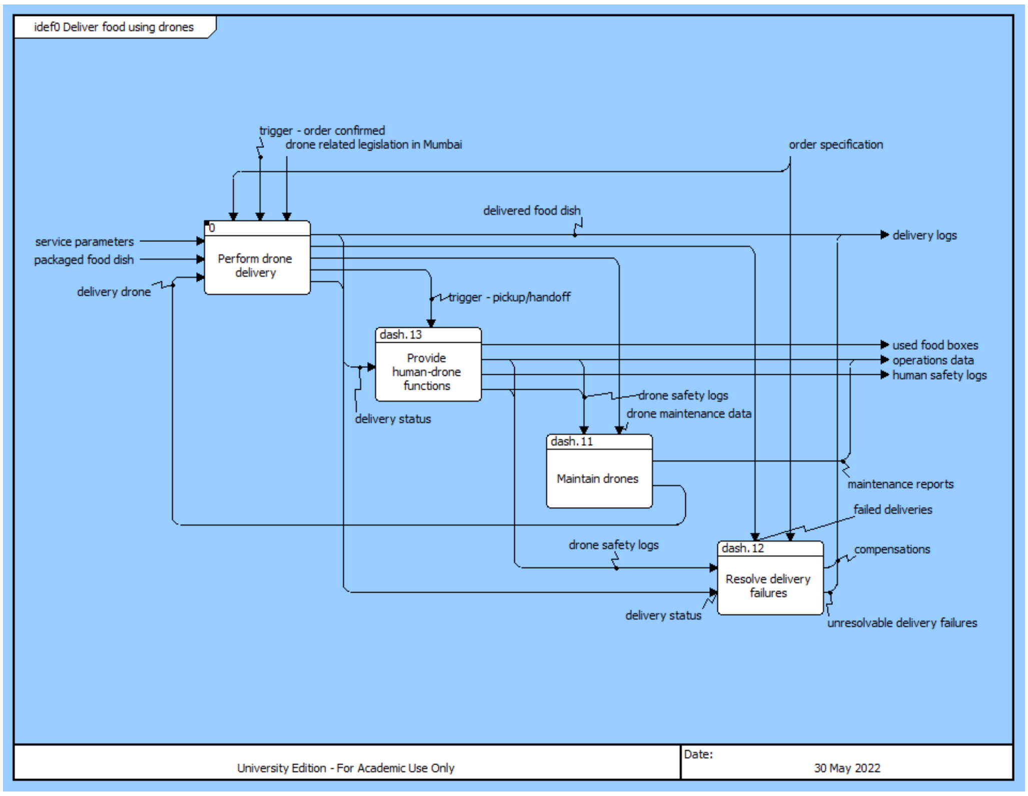

Figure 4: The IDEF0 diagram of the decomposition of function -1.

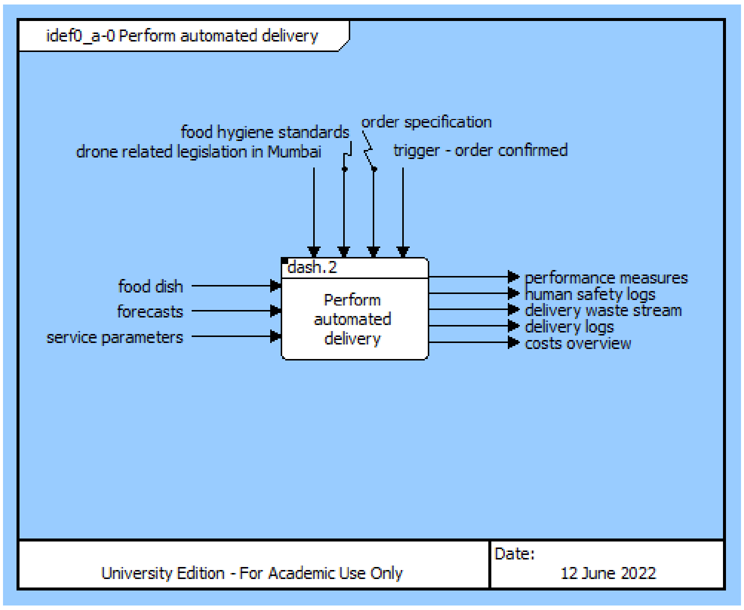

Figure 5: The A-0 diagram of function -2.

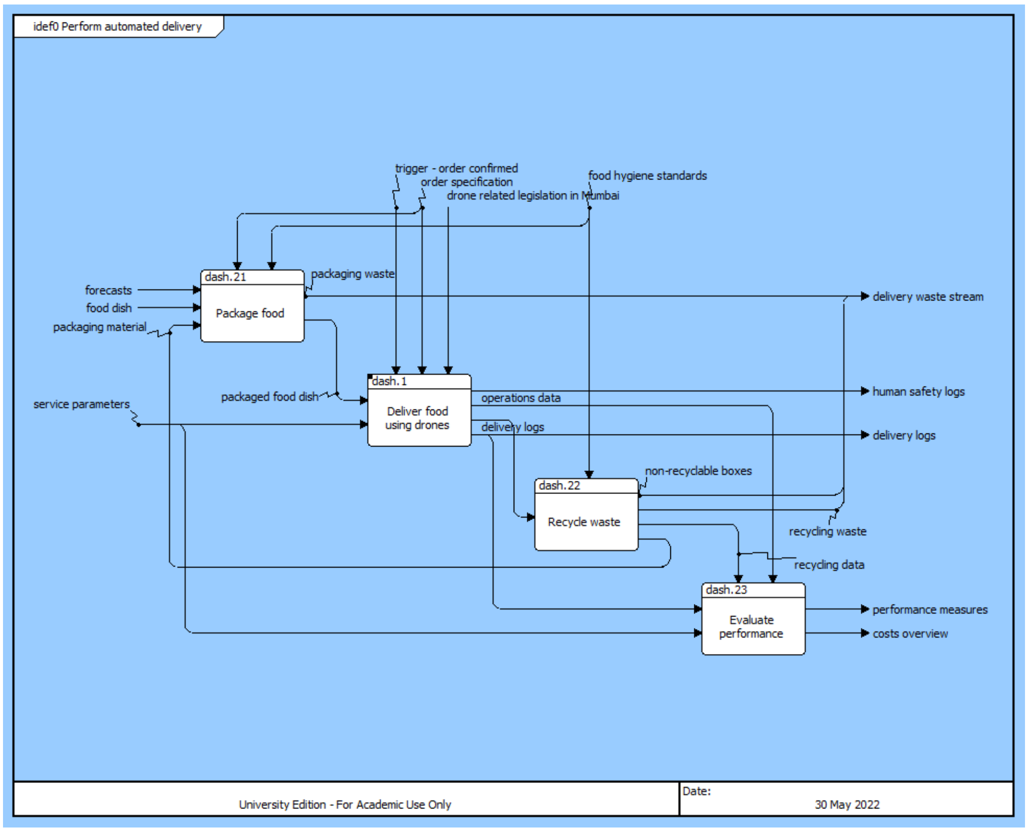

Figure 6: The IDEF0 diagram of the decomposition of function -2.

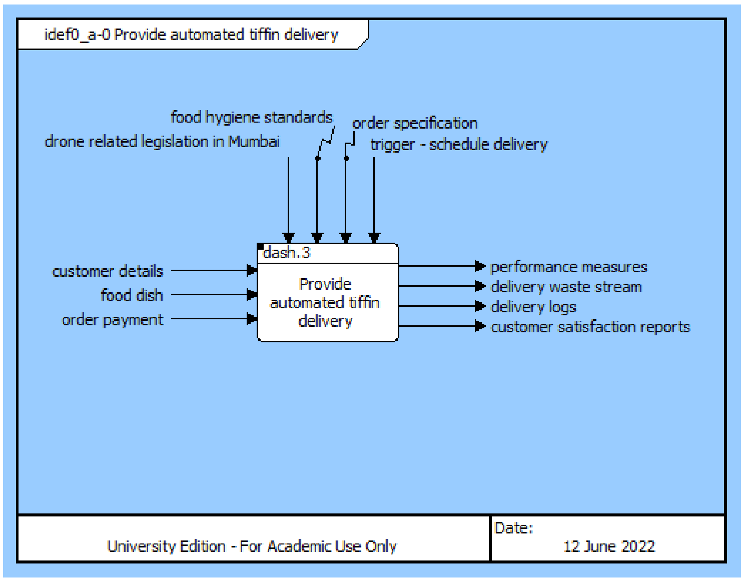

Figure 7: The A-0 diagram of function -3.

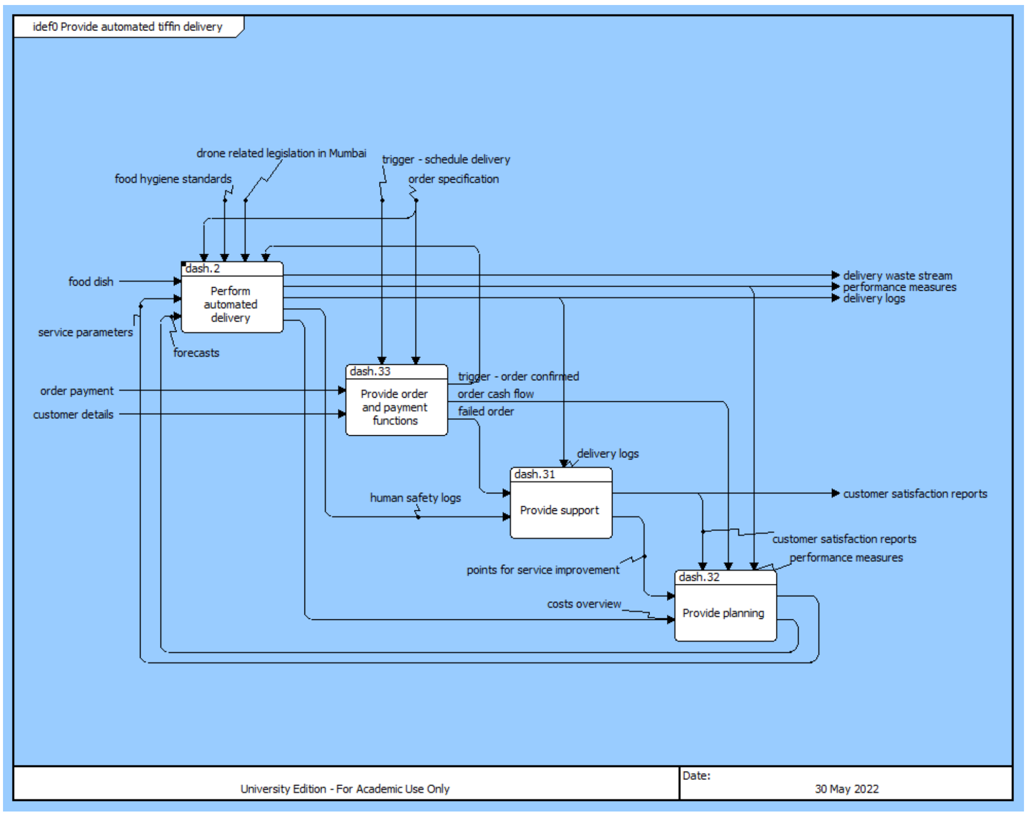

Figure 8: The IDEF0 diagram of the decomposition of function -3.

2.3 OBJECTIVE HIERARCHY OF THE DRONE FOOD DELIVERY SYSTEM

The objective hierarchy captures the requirements (objectives) of the system's stakeholders, as it may be depicted from the following diagram. Our system intends to established food delivery by drones in Mumbai; a densely populated area of India. The basic components of the system are: (a) the Social Impact (0.1), (b) the System Costs (0.2), (c) the System Reliability (0.3), and (d) the System Security (0.4).

Social Impact [0.1]
The Social Impact of the system is of less importance, but still a notable component of the system. It consists of the external and internal re-investments (from profits), the jobs lost, and the jobs created. All these have to be taken under consideration, since regulatory entities, food delivery companies, and labor unions are the mostly considered stakeholders for this part, along with consumers, restaurants, which come with less importance at this stage.

System Costs [0.2]
Since the project attempts to implement such a novel procedure, the System Costs should not be taken as the major component of the system, but definitely needs to be revised well. The cost components consist of the investment costs (investors, company), and the third-party costs (users and restaurants, with delivery costs, maintaining, investing, and operating as the next level components). These are considered of less importance at this stage, compared the maintenance and operational costs of the system itself, which need to be well realised for the system to flourish. Maintenance is divided to destroyed and serviced drones. The former concerns the drones that are totally destroyed or not worth to be fixed (buy new), while the latter accounts for the drones that may be fixed (operational manager).

Security [0.3]
Second highest in terms of importance and attention comes the System Security. The weight given for this aspect, reflects the goal of the system to achieve functional operations in terms of serving. Consequently, it is divided into network security and software & application security. Since mobile and wireless network being the two interface used for the system, data shouldn't be breached and hence should be monitored using protocols. Similarly, the software and application should be safeguarded upon two distinctive units, one being specific to domain and other related to privacy architecture.

Safety [0.4]
The most important component of the system is the Safety of the system. The system does not want to make any discount with the safety and the focus should be the highest of all. In more detail, the sub-components of this major component are the drone nest safety and the safety in flight and during transportation. The most important component under safety in flight and during transportation is the landing/take-off and the transportation safety which care about safe routes and hostile activity.The unsuccessful flights discriminate the lost drones and the drones that delivered the order in a wrong location. Stakeholders who care the most for this component are the investors, the consumers, the restaurants, the regulatory entities and the drone security manager

The complexity of the system is shown by accounting for these components and their sub-components. Having all these determined and weighted would benefit the project in terms of needs and requirement identification.

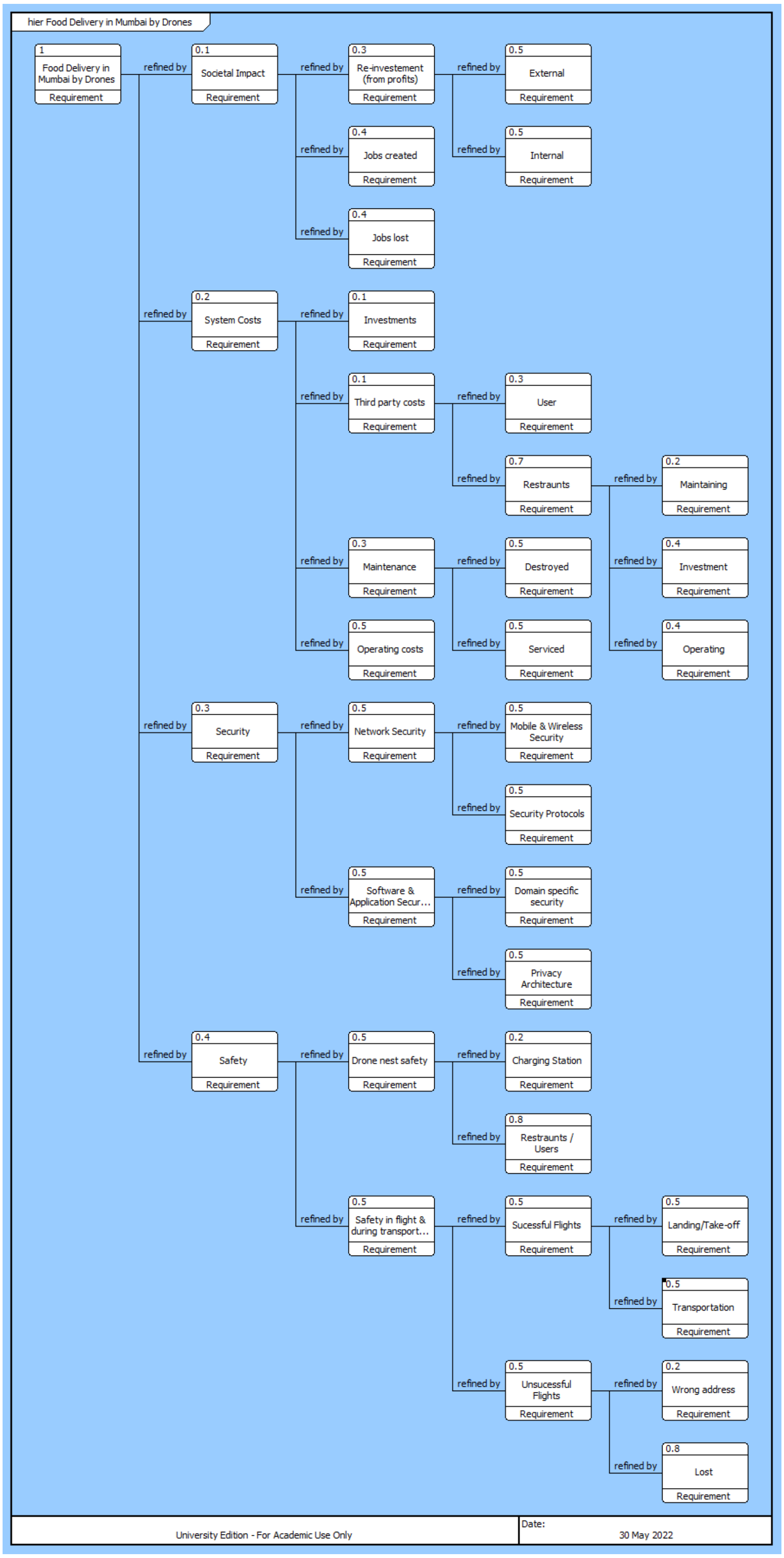

Figure 9: Objective Hierarchy diagram

3 REQUIREMENTS

3.1 STAKEHOLDER REQUIREMENTS

This section discusses various requirements placed by the stakeholders. Each subsection is visualised from their perspective and what they expect to achieve from the system. The 3.1.1 and 3.1.2 are inter-related. Failure in accomplishing the requirements set by one would eventually make the other a major bottle neck in the system. The requirements shown are the result of a dialogue between the design team and stakeholders in order to sharpen the requirements. Shown is the most recent version of each set of requirements. The order of requirements does not denote importance.

3.1.1 *Customer Requirements*

CsR-1 The system shall deliver foods in a faster way than the existing delivery system.
CsR-2 The system shall provide a more efficient alternative for meal delivery. Efficient means a delivery time reduced by at least 25% and sendoff/pickup locations that are on average 2 times closer than competing systems.
CsR-3 The system shall remain affordable and competitive against other delivery companies.
CsR-4 The system shall anticipate the preparation time of food.
CsR-5 The system shall arrive on time for the pick-up in order to reduce the lead time.
CsR-6 The system shall accommodate food boxes of acceptable dimensions and weight. The set of acceptable food boxes depends on external availability at the time of deployment.*
CsR-6 The system shall take measures to prevent/reduce security and safety hazards. This should reduce the amount of failures due to security and safety related issues to under 1%.

* if there is no suitable or affordable option for food boxes at deployment, the system has a no-go for deployment.

3.1.2 *Consumer Requirements*

CR-1 The drone delivery system shall be able to deliver in various locations, not only to particular addresses.
CR-2 The food shall not lose more than 20% of it's initial temperature during the delivery.
CR-3 The customer shall be able to send food to someone else.
CR-4 The delivery system shall be accessible through mobile phones.
CR-5 The contents being sent shall not be damaged, scrambled, or otherwise modified during delivery.

3.1.3 *Government Requirements*

GR-1 The system shall not interfere with the other flying devices.
GR-2 The system shall not breach and invade no fly zones.
GR-3 The system shall comply with the governmental rules and regulations.
GR-4 The system shall avoid collision with other drones, also those outside of the system.
GR-5 The system shall avoid cordoned area when assigning routes.
GR-6 The system shall not produce pollution during it's operation.
GR-7 The system shall not pry and collect information of any means.

3.2 SYSTEM REQUIREMENTS

System requirements are derived from the stakeholder requirements, and form an isomorphic tree with the functional architecture to be developed. As such, each requirement defines the behaviour of one function. Requirements can thus also be decomposed and refined hierarchically. The system requirements have been split in two types of requirements namely:

- Function requirements **R/FR**
- Technical Non-functional requirements **TNFR**

Functional requirements are requirements which adhere to some form of an action. While the technical non-functional requirements define properties of the system-to-be. TNFR are used to steer the definitions of functions and the assignment of components later on.

3.2.1 *Functional Requirements*

R.0 The system shall automatically deliver Tiffin
R.1 The system shall be secure
R.2 The system shall be safe
R.3 The system shall monitor its own behaviour
R.4 The system shall transport food
R.11 The system shall detect possible risks based on previous and real-time data
R.12 The system shall implement safety constraints
R.13 The system shall evaluate degree of the risks **R.21** The system shall initiate the drone recovery process in case of any malfunction
R.22 The system shall measure, identify and store any possible risks
R.23 The system shall evaluate safety data while in flight
R.31 The system shall generate reliable and consistent security data
R.32 The system shall perform safety check before data compilation
R.33 The system shall carry out timely maintenance
R.34 The system shall perform journey mapping
R.41 The drones shall communicate with air-traffic control
R.42 The drones shall perform take-off and landing actions
R.43 The drones shall perform in-flight functions
R.44 The drones shall load and unload food
R.111 The system shall recognise undesirable behaviour and deviations
R.112 The system shall highlight, classify and store any identified key external factors
R.113 The system shall structure previous logs and related data
R.121 The system shall perform precaution measures
R.122 The system shall simulate security scenarios
R.123 The system shall update drone architecture
R.124 The system shall adhere to drone technical requirements

R.131 The system shall compose security protocols
R.132 The system shall analyse current environmental conditions
R.133 The system shall predict future risks
R.134 The system shall highlight actual risks and their degree
R.211 The drone shall initiate the ATTI drone recovery protocol in case of malfunction
R.212 The drone shall initiate the RTH drone recovery protocol in case of malfunction
R.213 The system shall notify authorities in case of failure of the previous recovery process
R.214 The system shall recover the drone in case of malfunction
R.221 The system shall continuously monitor error conditions
R.222 The system shall identify risks based on previously analysed data and metrics
R.223 The system shall measure the risk severity of identified errors and risks
R.231 The system shall extract the safety state
R.232 The system shall perform structural and timing checks
R.233 The system shall aggregate all safety related data
R.311 The system shall screen the incoming data
R.312 The system shall perform fault detection
R.313 The system shall produce operational and security data
R.321 The system shall have a constant flow of system data
R.322 The system shall adhere to technical and government regulations
R.323 The system shall generate safety data
R.331 The system shall evaluate the battery of the drone
R.332 The system shall ensure that structure is operational at all time
R.333 The system shall inspect the overall condition of the drone
R.341 The system shall allocate drone
R.342 The system shall chart flight plan
R.343 The system shall visualise journey to check for possible disturbance
R.344 The system shall initiate journey
R.411 The drones shall plan a route
R.412 The drones shall perform risk assessment to threats and safety issues
R.413 The drones shall assess drone charge planning
R.414 The drones shall shall communicate with the rest of the system
R.421 The drones shall clear the landing area
R.422 The drones shall indicate take-off or landing status
R.423 The drones shall clear surroundings for take-off
R.431 The drones shall provide data about the physical structure of themselves
R.432 The drones shall control their position and altitude
R.433 The drones shall provide in-flight data
R.434 The drones shall provide emergency protocol
R.441 The drones shall allow for food to be added and removed from the dish from the food compartment
R.442 The drones shall measure the weight of the dish added to the food compartment
R.443 The drones shall open/close the food compartment

3.2.2 *Technical Non-functional requirements*

TNFR-1 All drones shall go through a maintenance check on a monthly basis.
TNFR-2 The system requires a box cleaning facility.
TNFR-3 The system shall own storage warehouses.
TNFR-4 The meals loaded on a drone shall be loaded in an insulated box.
TNFR-5 All personal data stored shall adhere to privacy rules of the country.
TNFR-6 The system shall consider the weather conditions when assigning routes.

4 FUNCTIONAL ARCHITECTURE

4.1 DECOMPOSITION OF FUNCTION 0

Now that we have established the context systems of our system at A0, we aim to decompose it into relevant subfunctions. Recall the I/O/C of function 0 in Figure 1. We propose an initial 3 decomposition alternatives following different stakeholder perspectives. These perspectives may be used according to the prioritisation of the requirements in chapter 3, specifically the objective hierarchy in Figure 9.

4.1.1 *Finance priority*

Our first alternative may be derived from requirements prioritising financial support of the system. In Figure 1, we show the hierarchy diagram of the functions that decompose "Perform drone delivery", with the respective interaction diagram in Figure 2.

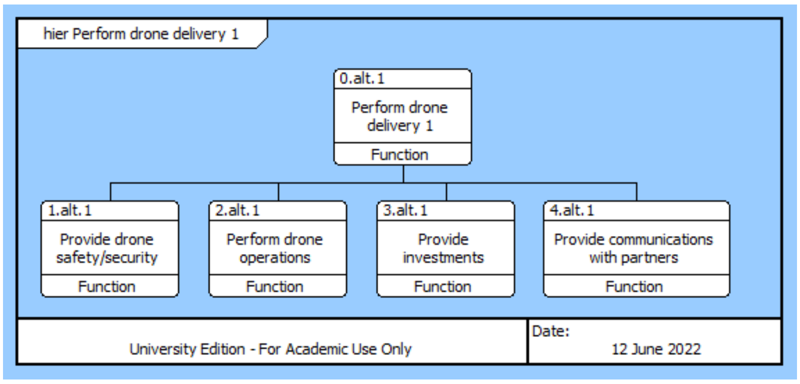

Figure 1: Hierarchy diagram of the decomposition of function 0 (alternative 1).

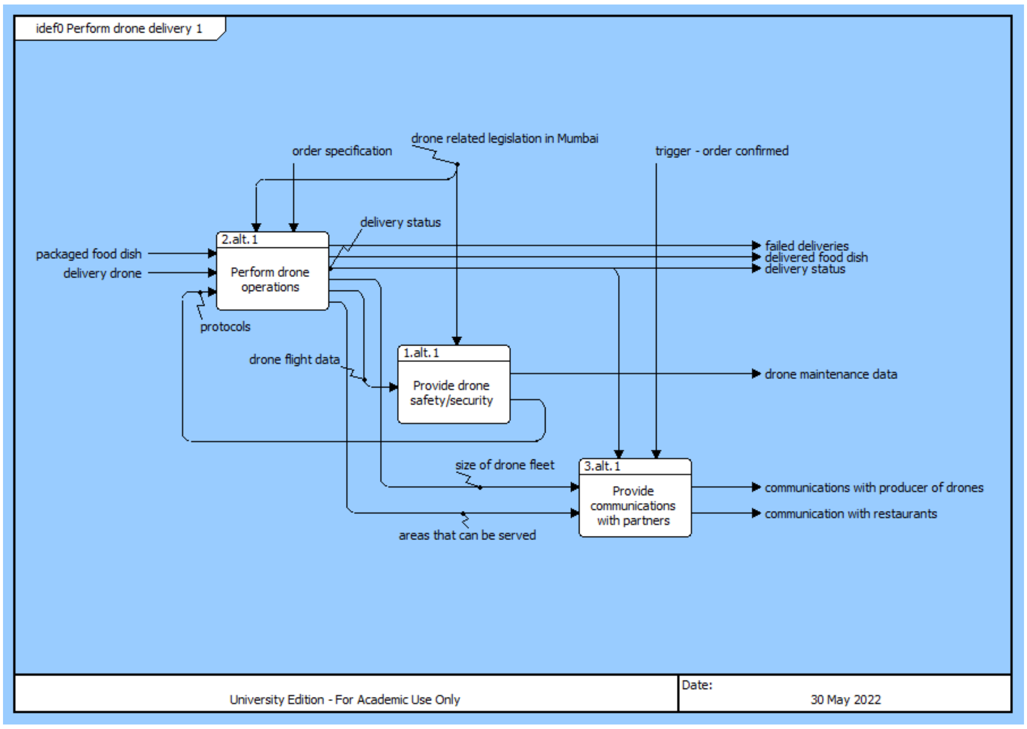

Figure 2: IDEF0 diagram of the decomposition of function 0 (alternative 1).

4.1.2 *Safety and security priority*

Our second alternative may be derived from requirements prioritising safety around the transport drones, and the security of the drones themselves. The focus lies on detecting and handling potential issues in those categories. In Figure 3, we show the hierarchy diagram of the functions that decompose "Perform drone delivery", with the respective interaction diagram in Figure 4.

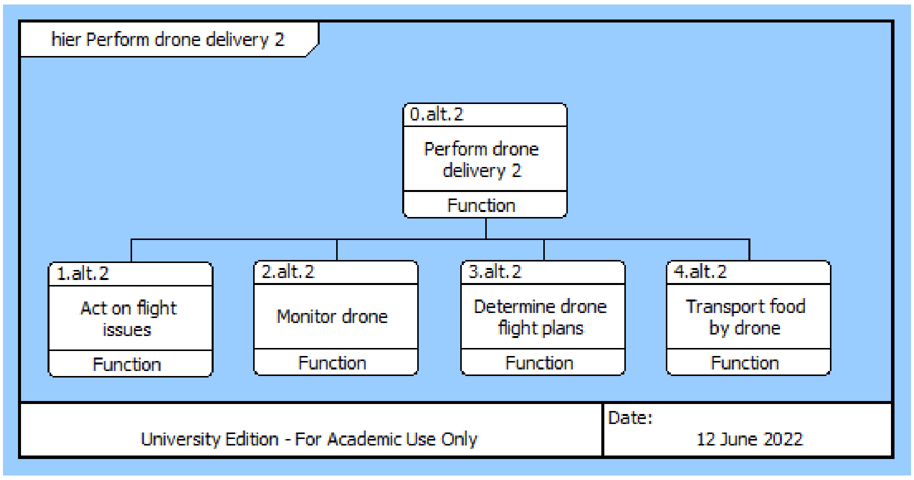

Figure 3: Hierarchy diagram of the decomposition of function 0 (alternative 2).

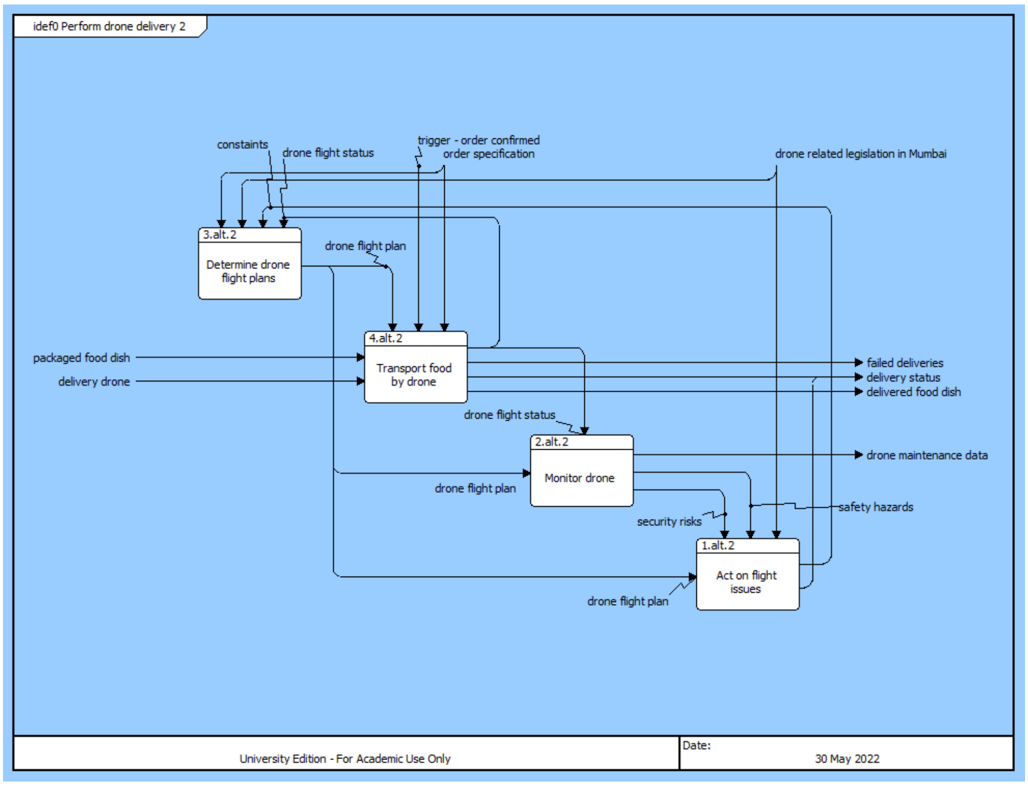

Figure 4: IDEF0 diagram of the decomposition of function 0 (alternative 2).

4.1.3 *Logistical organisation priority*

Our third alternative may be derived from requirements prioritising logistical soundness. In Figure 5, we show the hierarchy diagram of the functions that decompose "Perform drone delivery", with the respective interaction diagram in Figure 6.

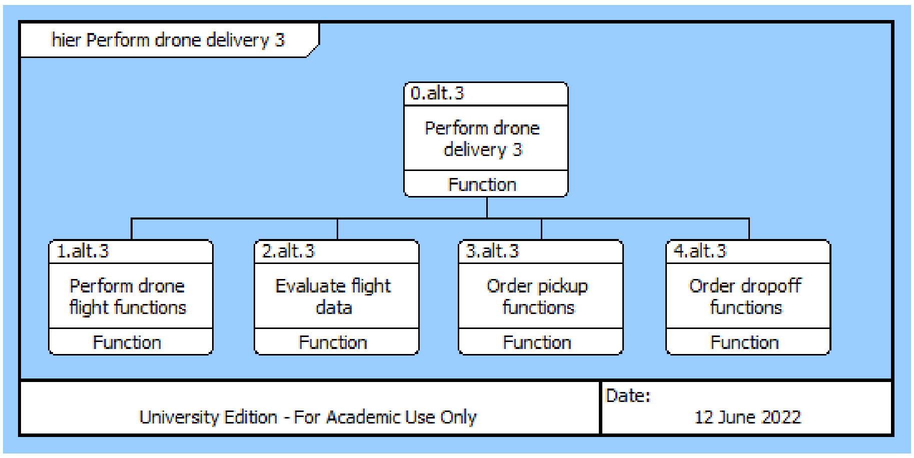

Figure 5: Hierarchy diagram of the decomposition of function 0 (alternative 2).

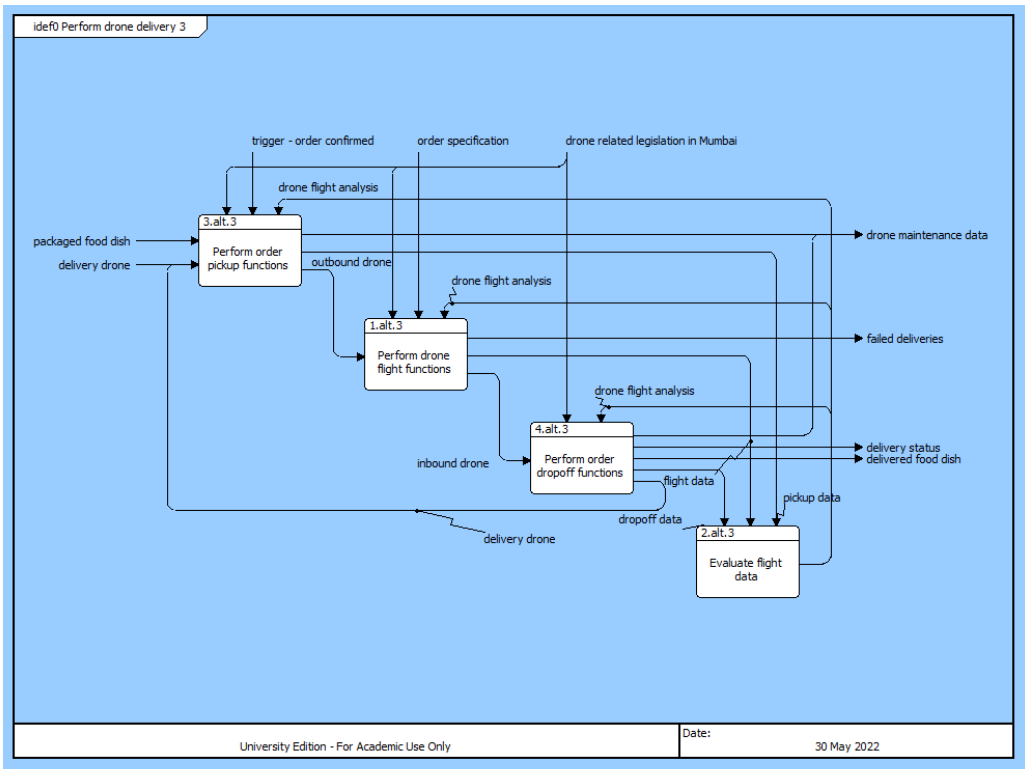

Figure 6: IDEF0 diagram of the decomposition of function 0 (alternative 3).

4.2 SELECTION OF THE MOST SUITABLE ALTERNATIVE

Each alternative has a different focus and it is necessary to identify the alternative which encompasses necessary operations and be compatible with the requirements laid out by the stakeholders. The alternative 2 which prioritise safety and security (4.1.2) is apt for the proposed system. This address flight issues, drone monitoring, determination of flight plans and efficient way of transportation of food.

Ensuring safety in all perspectives (air traffic, property and user) is of highest priority particularly for drones operating in densely populated areas. It has to be integrated efficiently to be on par with other aerial devices complying to the rules and regulations set by the government. The whole system can collapse if there is a lapse in safety. If the drone looses its control and cause property damage or threat to life, this can lead to serious consequences which might result in complete shut down of the company. The stint cargo bike which was used to carry children had to shut down it's operation because of the tragic accident that took place. The system experienced a sudden failure in one of the safety components which eventually resulted in loss of 4 young children as it collided with an incoming train [8]. This is why safety should be given the most importance as huge investment done for the development of the company becomes meaningless if it leads to such a scenario.

4.3 DECOMPOSITION OF HIGH LEVEL FUNCTIONS

We will now decompose the high level functions in our system into more detailed subdivisions. These functions will touch on design choices for our system. Functions 1 and 2 have been decomposed in 3 different ways, from different perspectives. Each time, the most suitable (combination of) alternative(s) is chosen to decompose further. This way, we end up with a 3-level decomposition of function 0. We show them in order of importance.

4.3.1 *Decomposition of function 1*

This function is in charge of consolidating security inputs from previous and current processes in the system, analysing the inputs, assessing the current security state of the system with respect to the inputs, and eliminating all security risks that are identified to facilitate a rapid recovery in the system's security state.

Based on the responsibilities of function 1, we developed three alternative decompositions for it. The functional decompositions all fulfil its core responsibilities but have different approaches to doing so. Consequentially, they prioritise 3 different perspectives.

Alternative 1

The first alternative that we developed prioritizes detecting environmental threats, evaluating the degree of risk and eliminates them by taking several actions. Additionally, it is an automated procedure that leverages real-time data from the environment to assess the security state of the drone. It determines the degree of risks and eliminate escalated security risks. "Eliminate environmental threat" function uses specialized software and tools to achieve its goal.

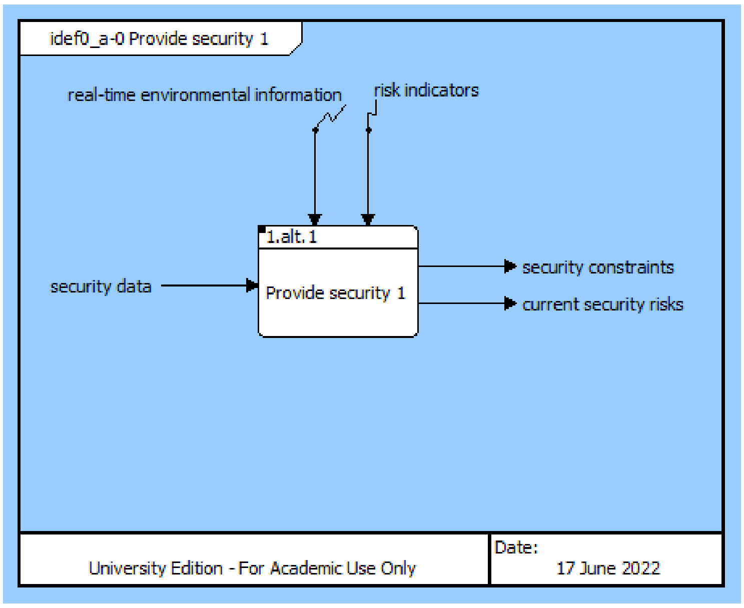

Figure 7: A0 diagram of the decomposition of function 1, alternative 1

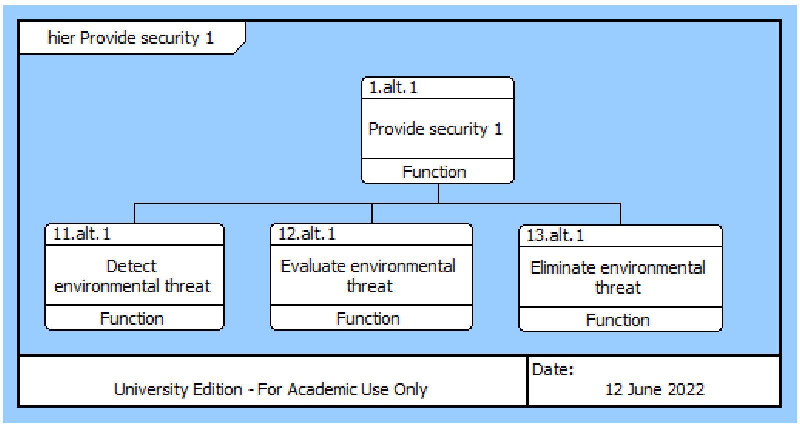

Figure 8: Hierarchy diagram of the decomposition of function 1, alternative 1

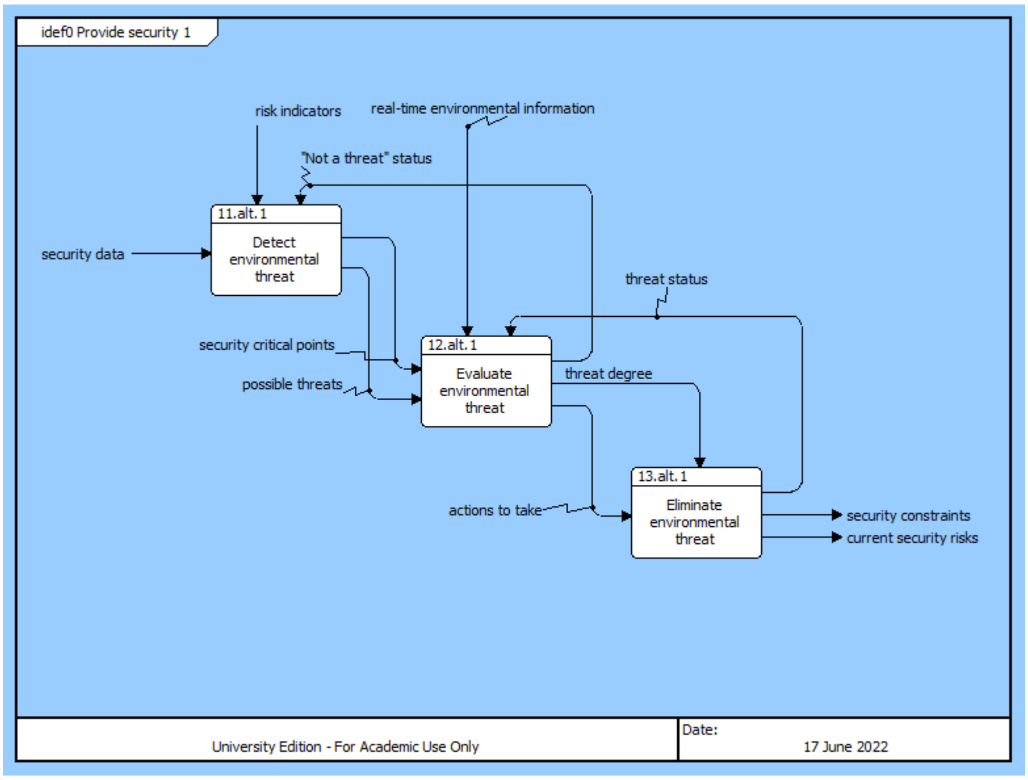

Figure 9: IDEF0 diagram of the decomposition of function 1, alternative 1

Alternative 2

This alternative detects possible risks such as real-time happenings in drone path considering real-time drone status information. It evaluates the degree of the risks and applies a security protocols for each cases. Additionally, this alternative does not necessarily employ automation in eliminating security risks. These mentioned protocols can be either predetermined for specific security risk scenarios or can be determined by the internal staff and security risk officers to isolate and control security risks. In situations where a security risk is too difficult to control by internal staff alone, local security authorities may be contacted for assistance.

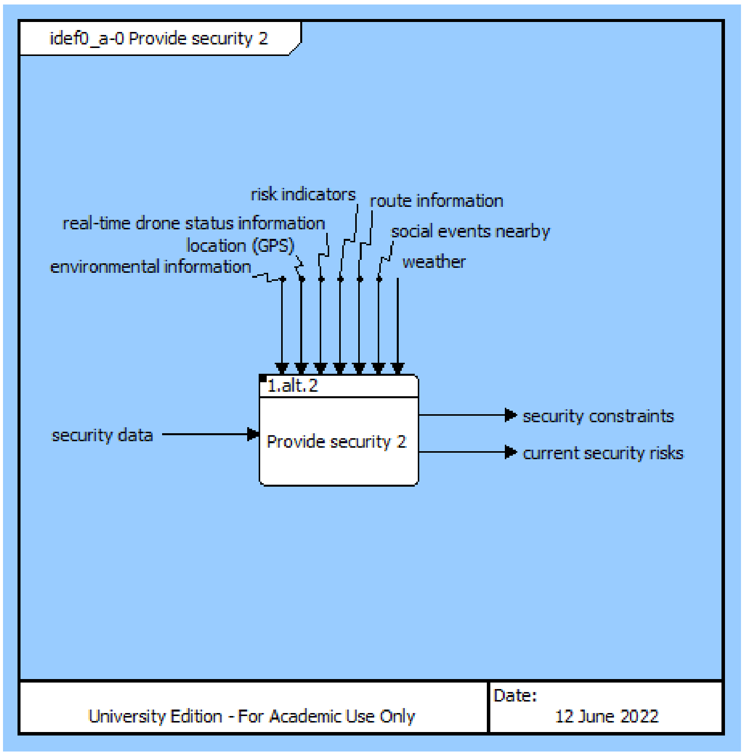

Figure 10: A0 diagram of the decomposition of function 1, alternative 2

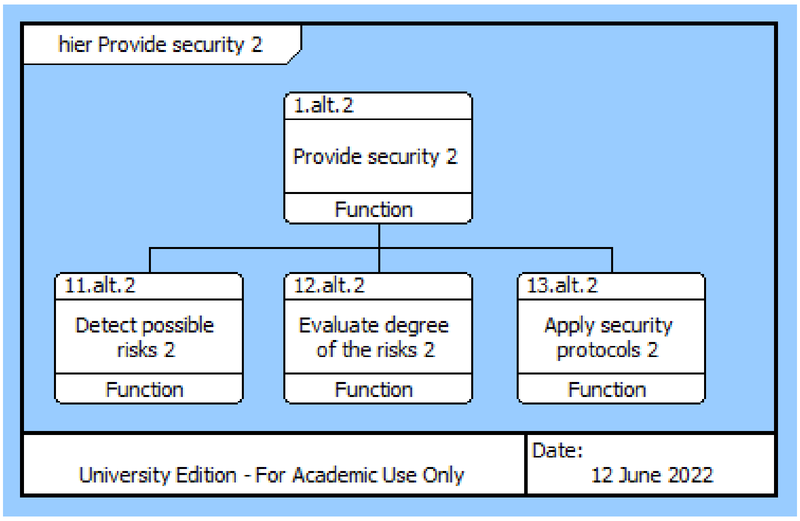

Figure 11: Hierarchy diagram of the decomposition of function 1, alternative 2

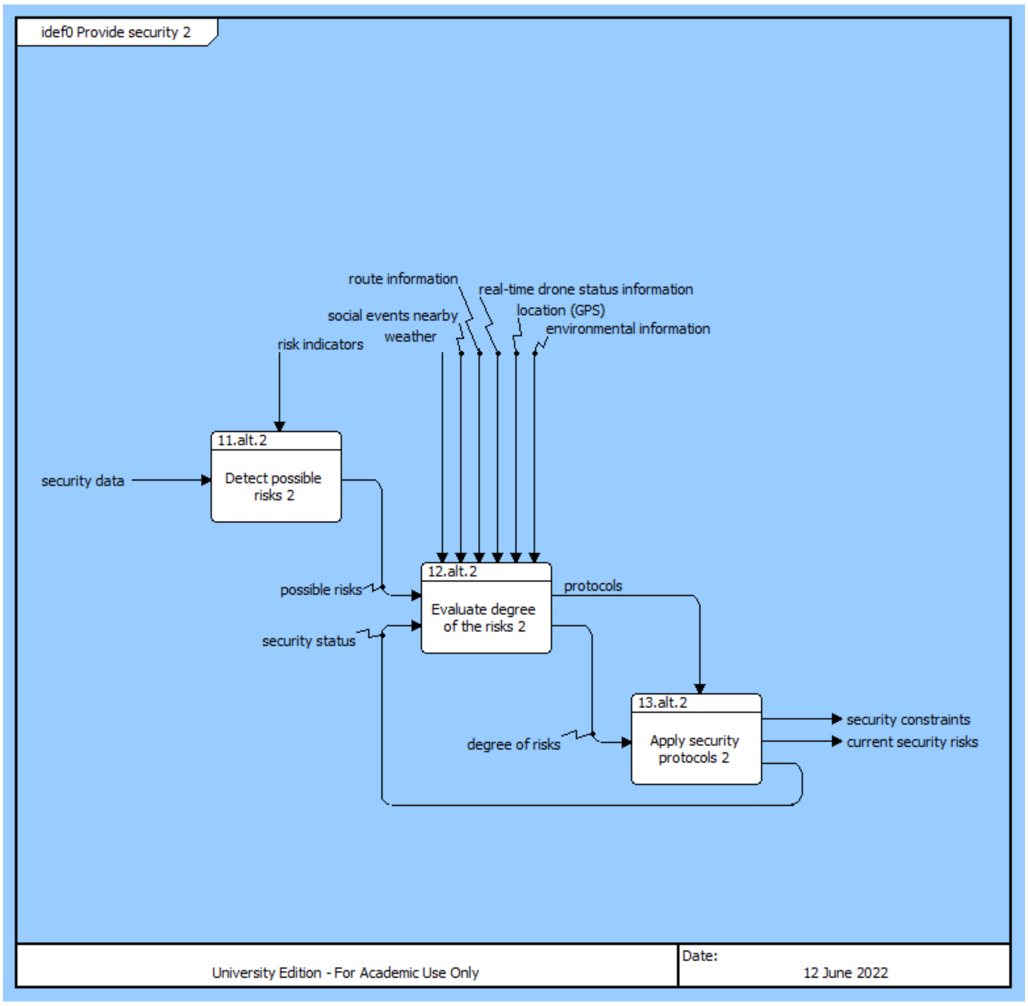

Figure 12: IDEF0 diagram of the decomposition of function 1, alternative 2

Alternative 3

This alternative primarily focuses on determining the best possible secure route to take for the drone by evaluating current air traffic on the routes, any structural/constructive factors such as electric cables and security critical points. This alternative also takes real-time security state changes on the possible routes by leveraging real-time environmental information, and rearranges the best possible secure route accordingly.

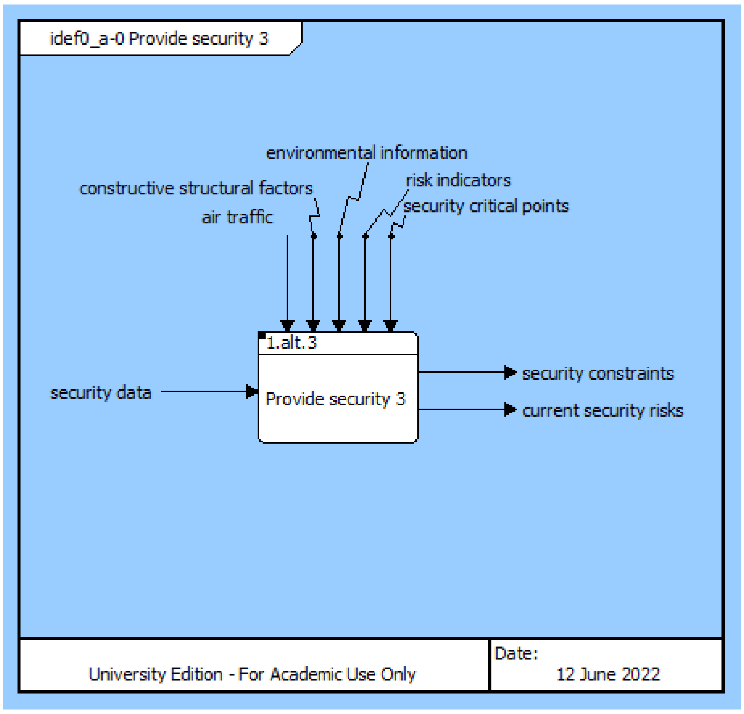

Figure 13: A0 diagram of the decomposition of function 1, alternative 3

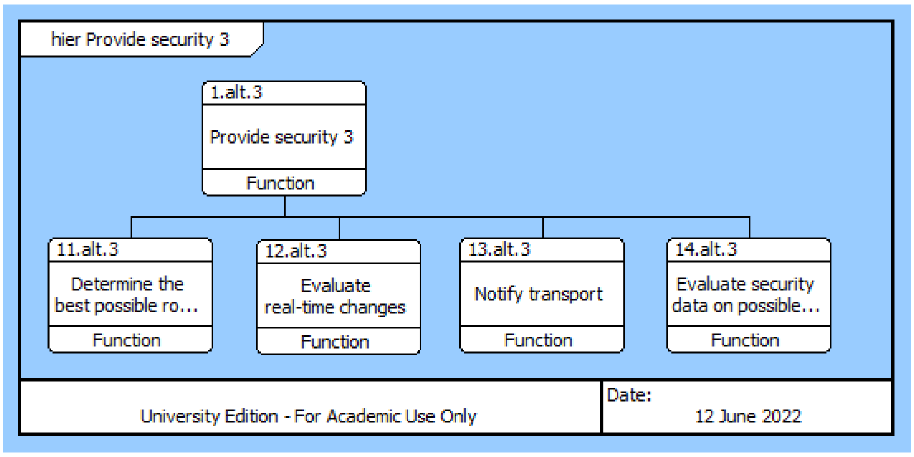

Figure 14: Hierarchy diagram of the decomposition of function 1, alternative 3

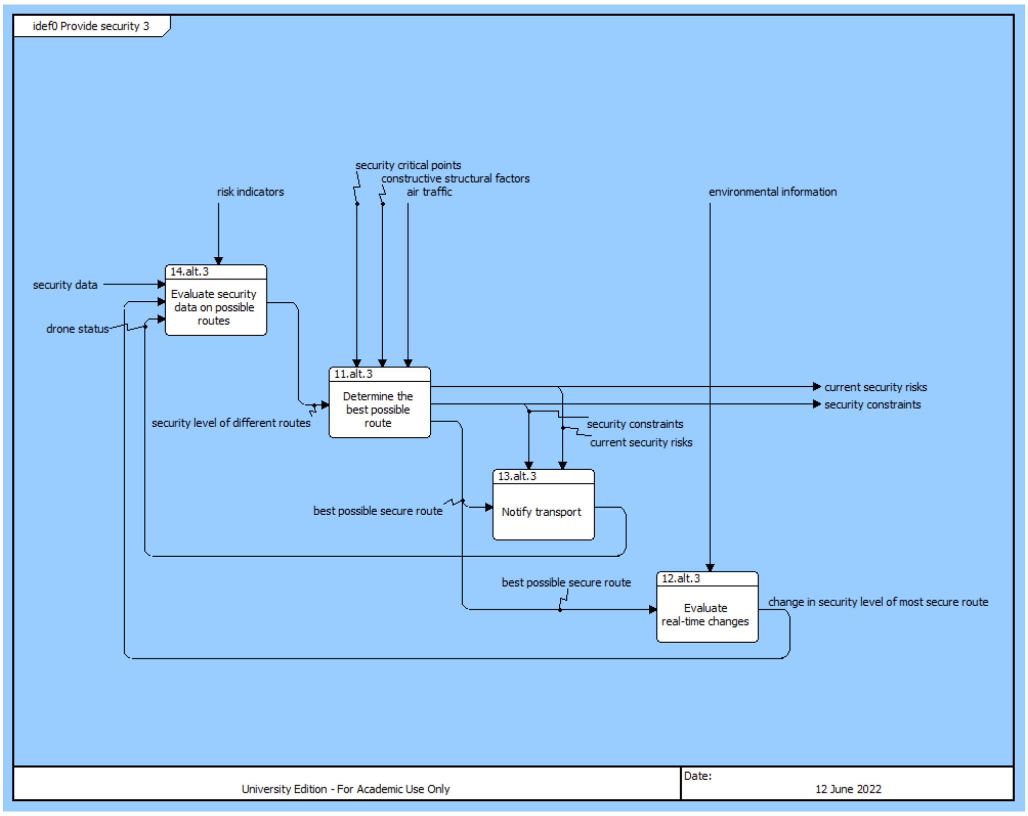

Figure 15: IDEF0 diagram of the decomposition of function 1, alternative 3

Selection of the most suitable alternative

Considering the aforementioned advantages, our team selected alternative 2 as the decomposition of choice for function 1. Security is an utmost concern in our system, hence we believe that even automated procedures saves cost of acquiring resources meaning that workforce, we might need internal staff, security risk officers and local security authorities to determine protocols such that isolates and controls security risks in situations where a security risk is too difficult to be controlled by automated protocols only. For our purposes, alternative 2 is the most suitable choice.

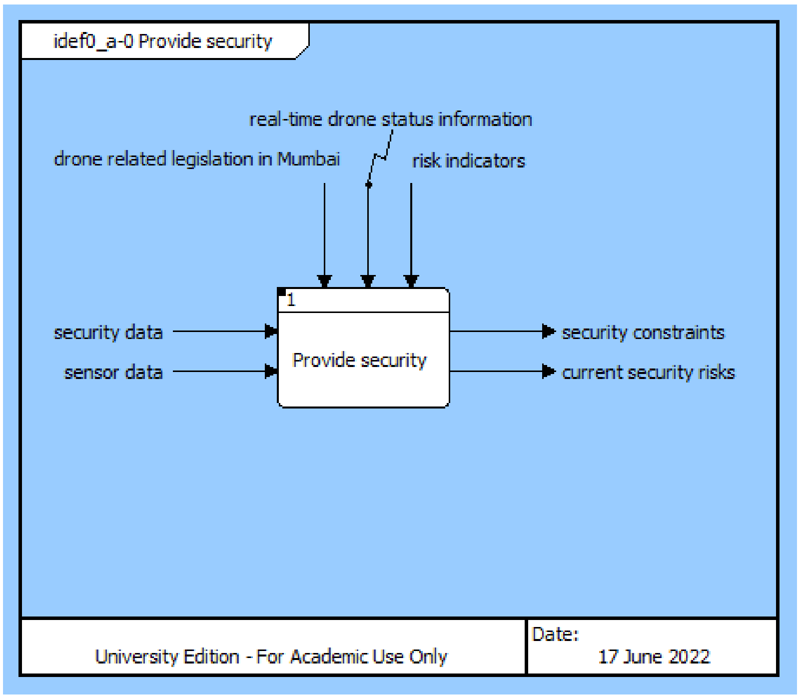

Figure 16: A0 diagram of the decomposition of function 1, chosen alternative.

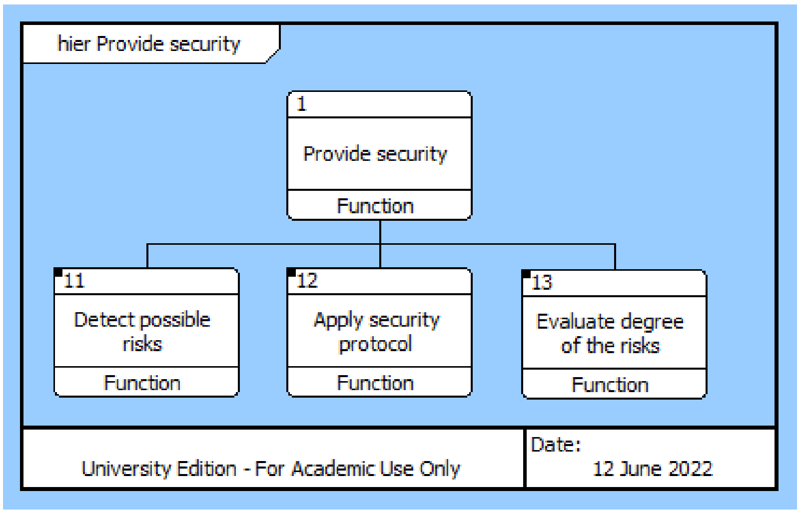

Figure 17: Hierarchy diagram of the decomposition of function 1, chosen alternative.

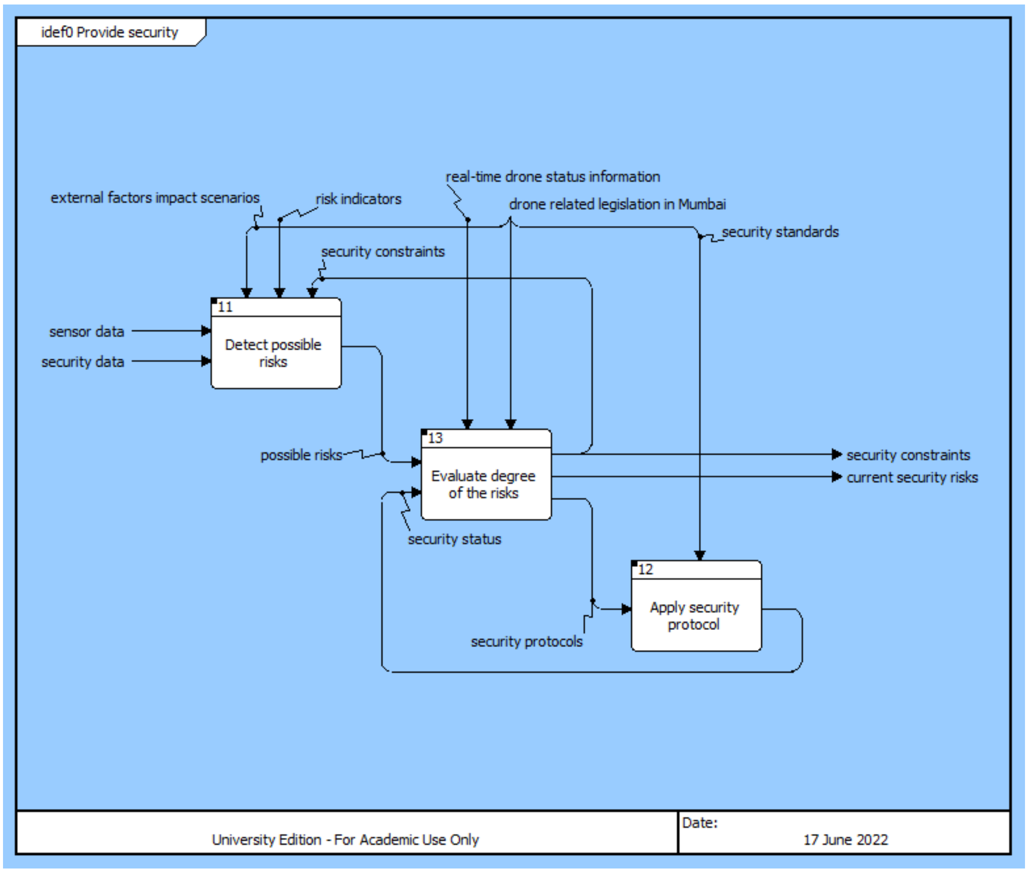

Figure 18: IDEF0 diagram of the decomposition of function 1, chosen alternative.

Decomposition of lower level functions

To detect possible security risks in our system, we define a feedback loop on detecting issues, utilising previously detected risks as an indicator. Combining the previously occurred risks with current data, we obtain a set of currently ongoing risks that can be further investigated.

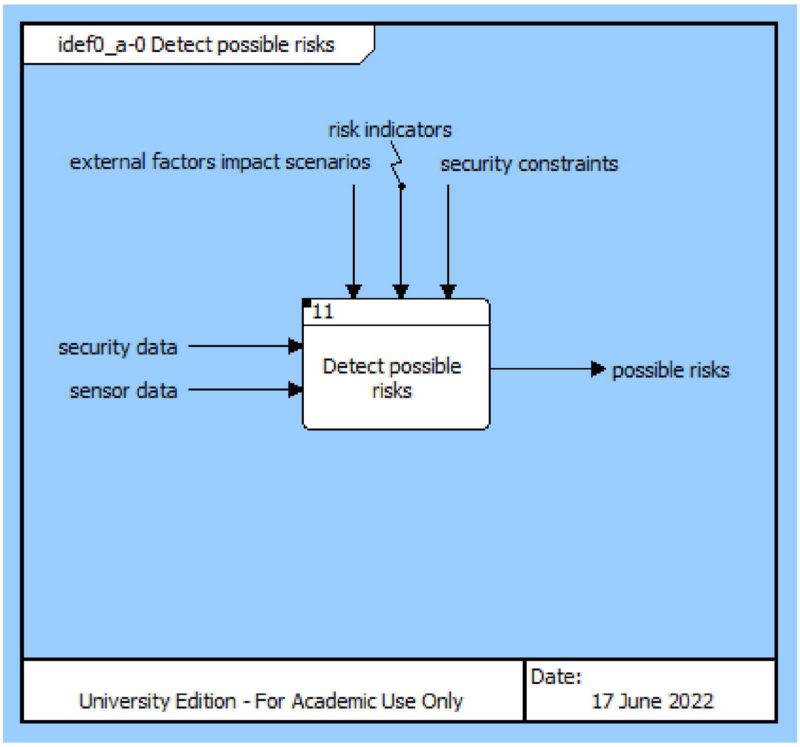

Figure 19: A0 diagram of the decomposition of function 11, chosen alternative.

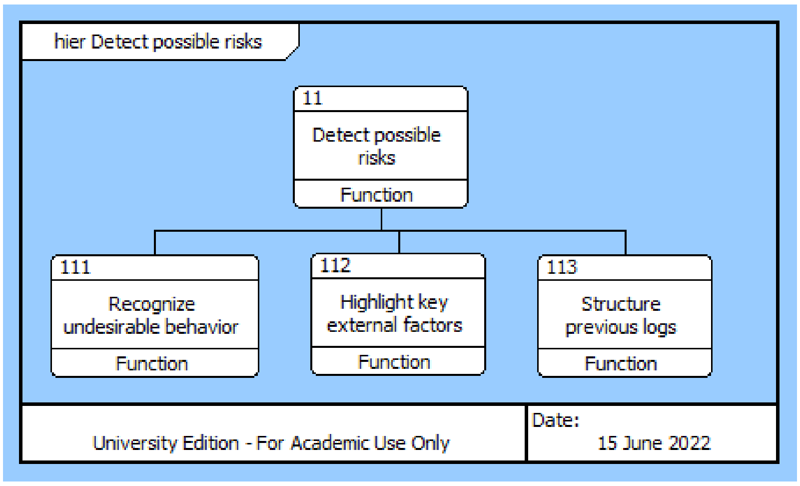

Figure 20: Hierarchy diagram of the decomposition of function 11.

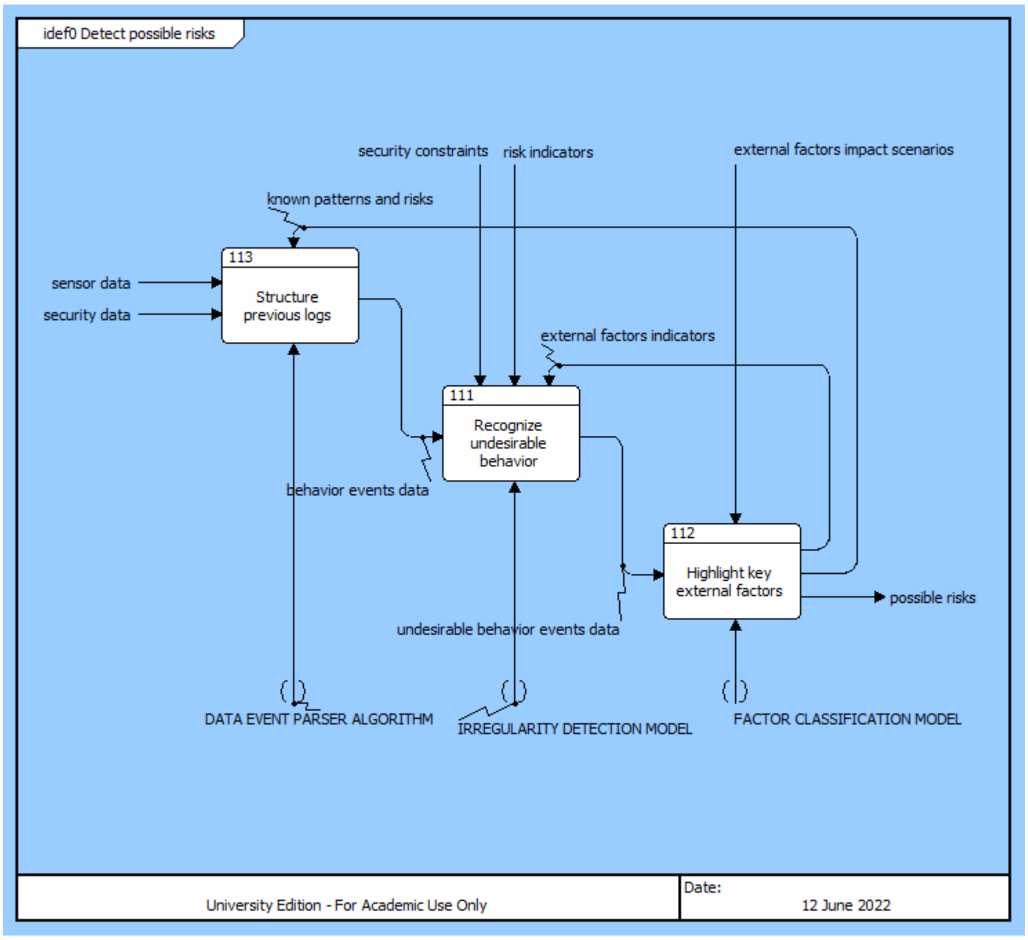

Figure 21: IDEF0 diagram of the decomposition of function 11.

Apply security function is responsible from implementing scenarios programmatically, such that drones avoid risks, updates security scenarios, and implement mentioned updates to drone architecture accordingly by evaluating compliance of them.

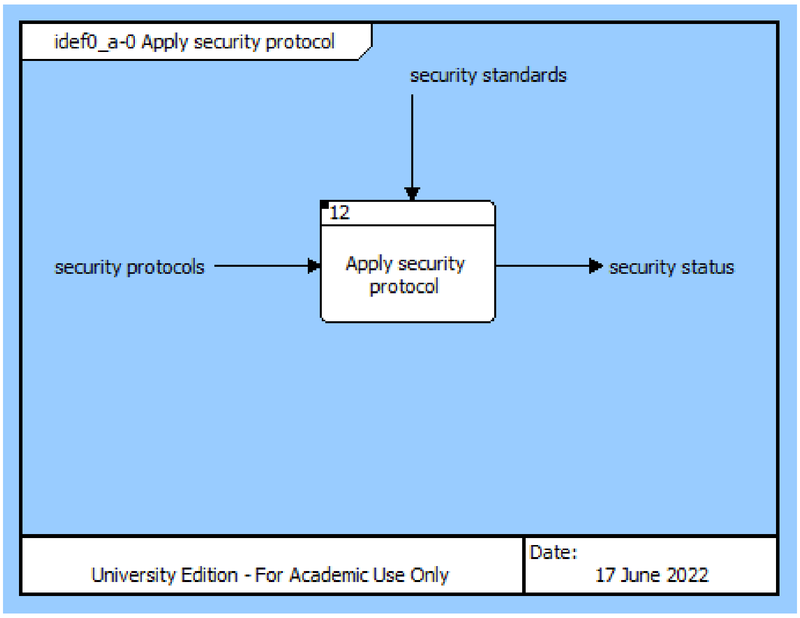

Figure 22: A0 diagram of the decomposition of function 12, chosen alternative.

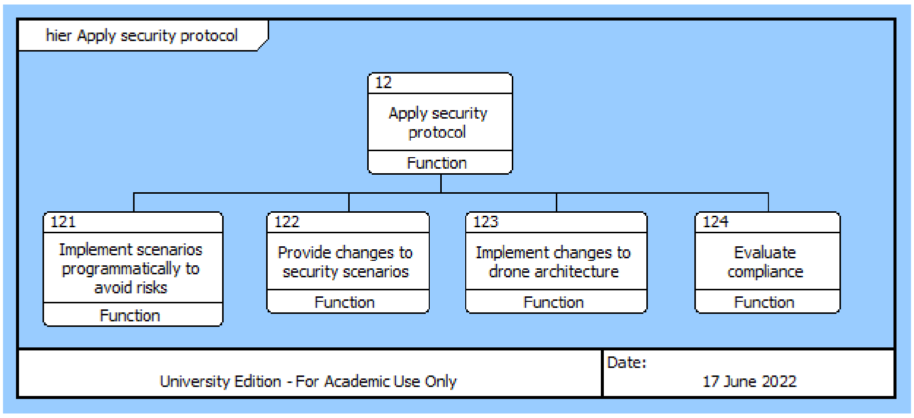

Figure 23: Hierarchy diagram of the decomposition of function 12.

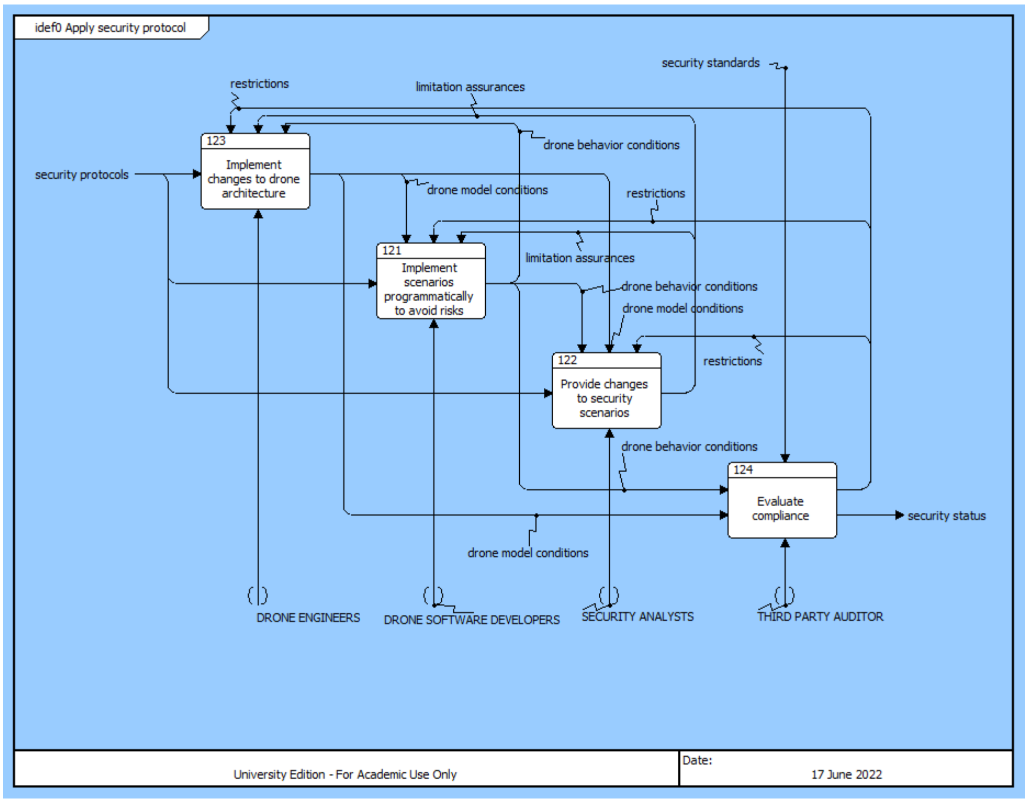

Figure 24: IDEF0 diagram of the decomposition of function 12.

Lastly, we evaluate the degree of the earlier detected risks. We put emphasis on environmental conditions. Moreover, we want to satisfy requirement CsR-6, and as such implement a system that predicts future risks based on the degree of risks to update the security constraints followed by the drones.

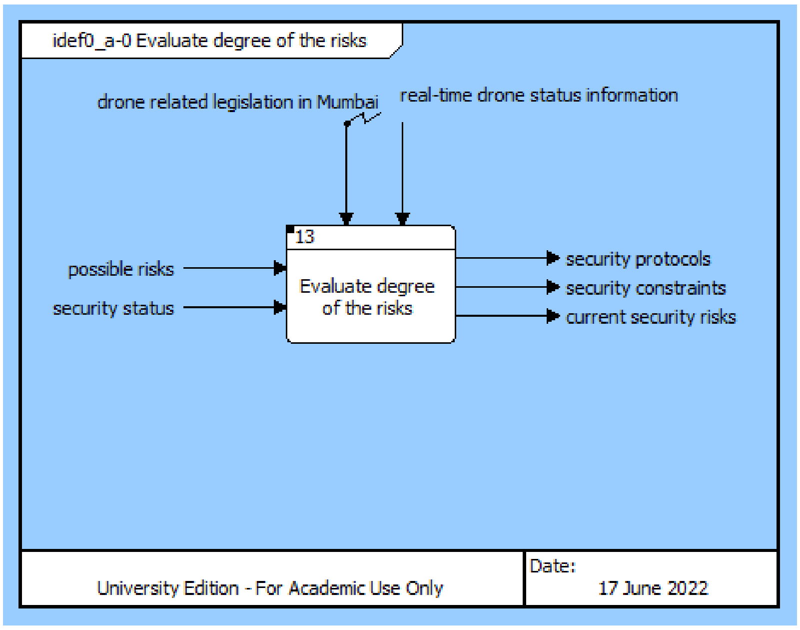

Figure 25: A0 diagram of the decomposition of function 13, chosen alternative.

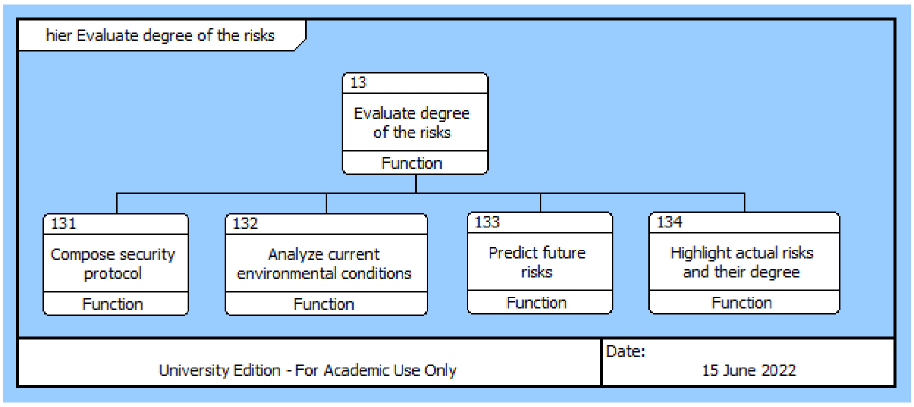

Figure 26: Hierarchy diagram of the decomposition of function 13.

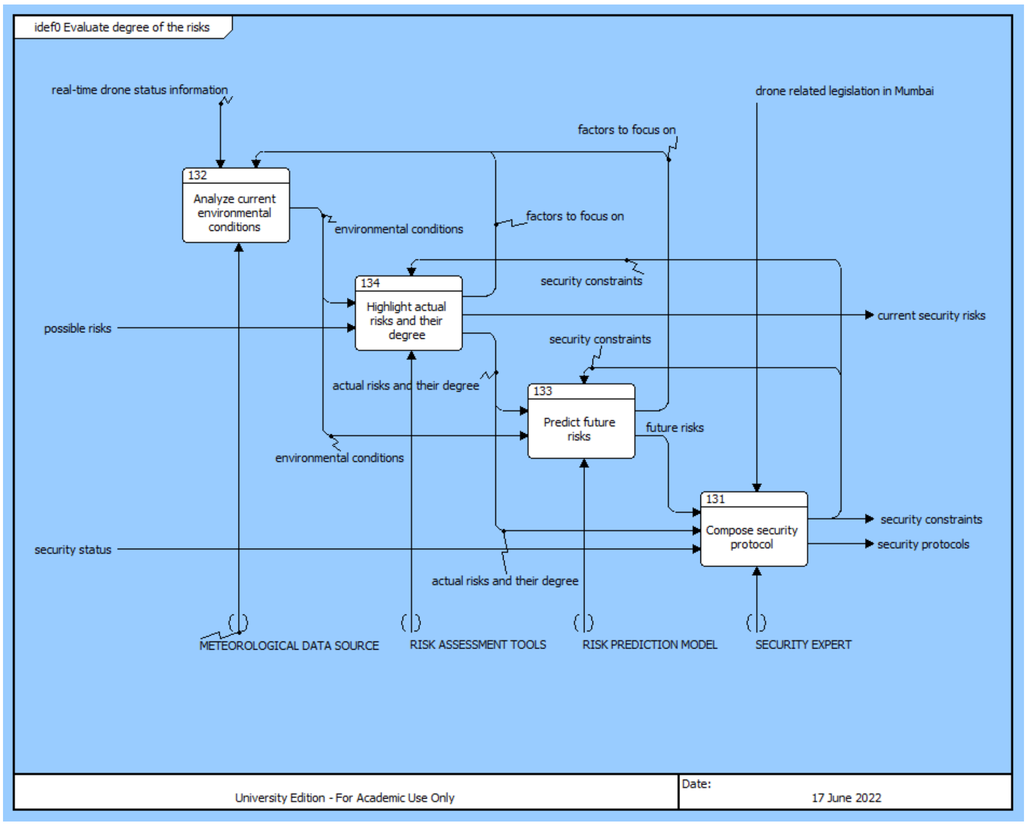

Figure 27: IDEF0 diagram of the decomposition of function 13.

4.3.2 *Decomposition of function 2*

This function is in charge of consolidating safety inputs from previous and current processes in the system, analysing the inputs, assessing the current safety state of the system with respect to the inputs, and eliminating all safety risks that are identified to facilitate a rapid recovery in the system's safety state.

Based on the responsibilities of function 2, we developed three alternative decompositions for it. The functional decompositions all fulfil its core responsibilities but have different approaches to doing so. Consequentially, they prioritise 3 different perspectives.

Alternative 1

The first alternative that we developed prioritizes the efficiency of the safety provision process by minimizing the number of sub processes that are needed to address and recover from internal safety risks. Furthermore, it fully embraces automation and leverages realtime streams of information to assess the safety state of the done, determine the level of risks and recover from escalated safety risks.

The cost of achieving safety is not a primary concern of this alternative and as such, it uses specialized tools and software to achieve its 'Perform Drone Recovery' procedure.

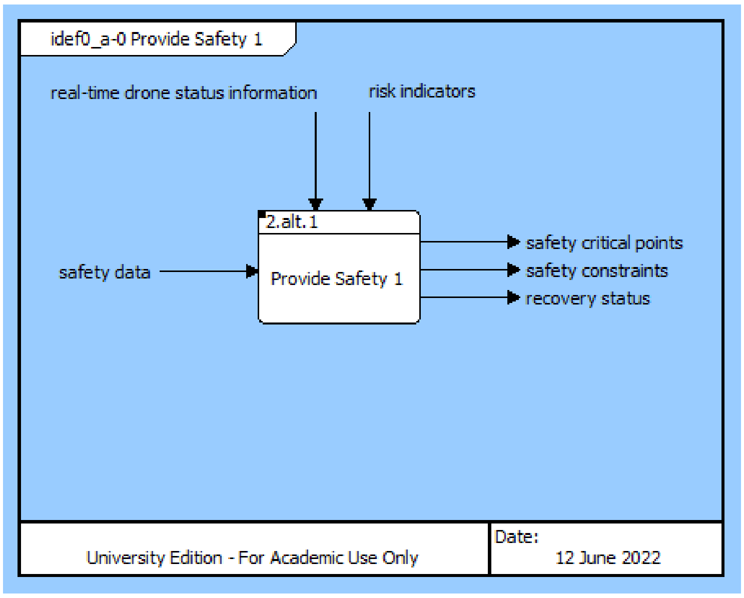

Figure 28: A0 diagram of the decomposition of function 2, alternative 1

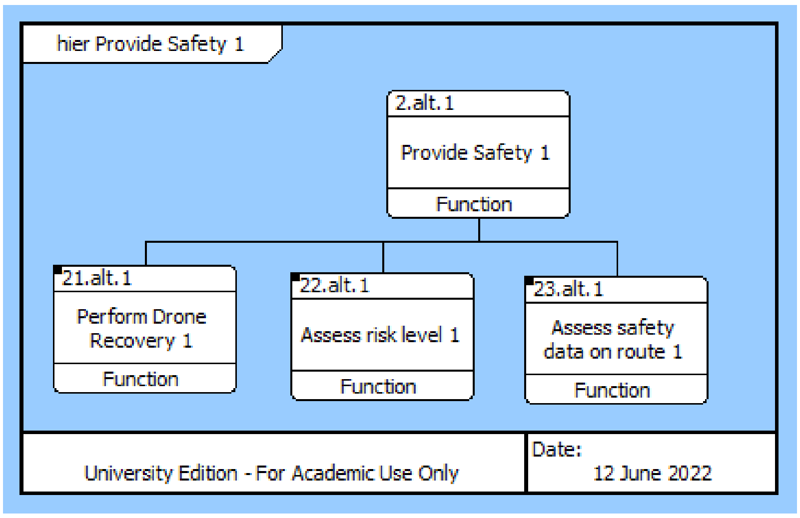

Figure 29: Hierarchy diagram of the decomposition of function 2, alternative 1

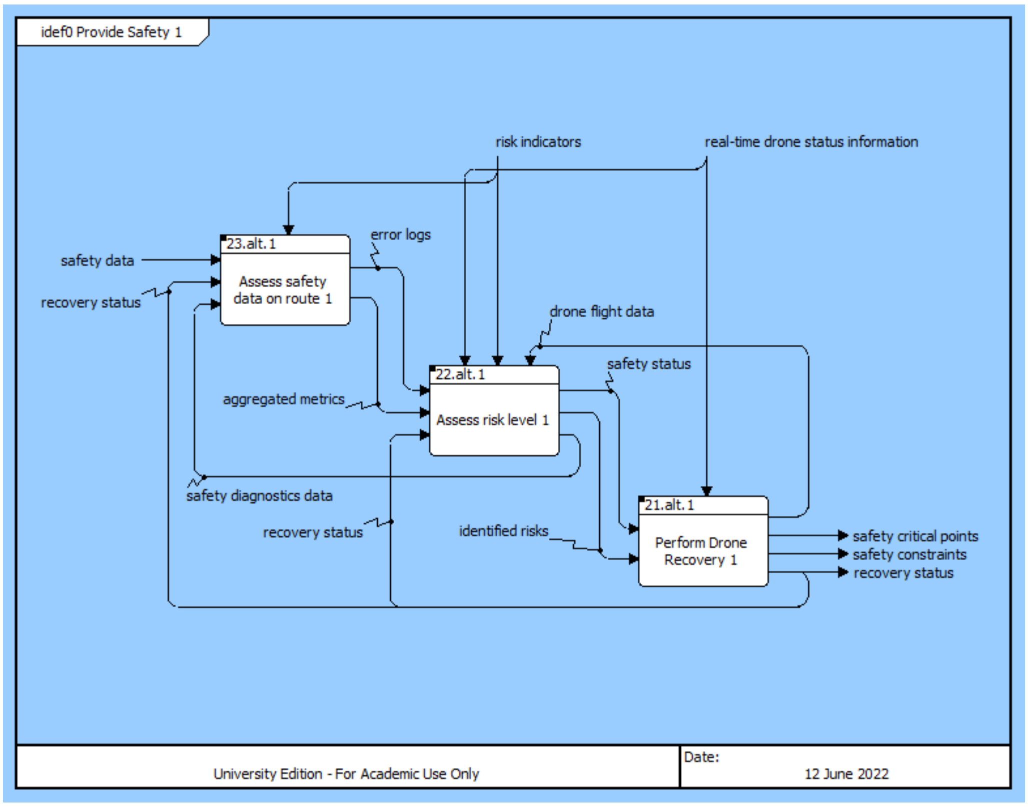

Figure 30: IDEF0 diagram of the decomposition of function 2, alternative 1

Alternative 2

This alternative prioritizes the cost to providing safety over the efficiency of how the safety is provided. This is immediately evident by the fact that it requires 4 sub-procedures compared to the 3 that the first alternative needed. In addition, this alternative does not employ much automation in eliminating safety risks. Instead, it relies on the effort of internal staff and risk officers to isolate and control safety risks. In situations where a safety risk is too difficult to control by internal staff alone, local safety authorities such as fire services may be contacted for assistance.

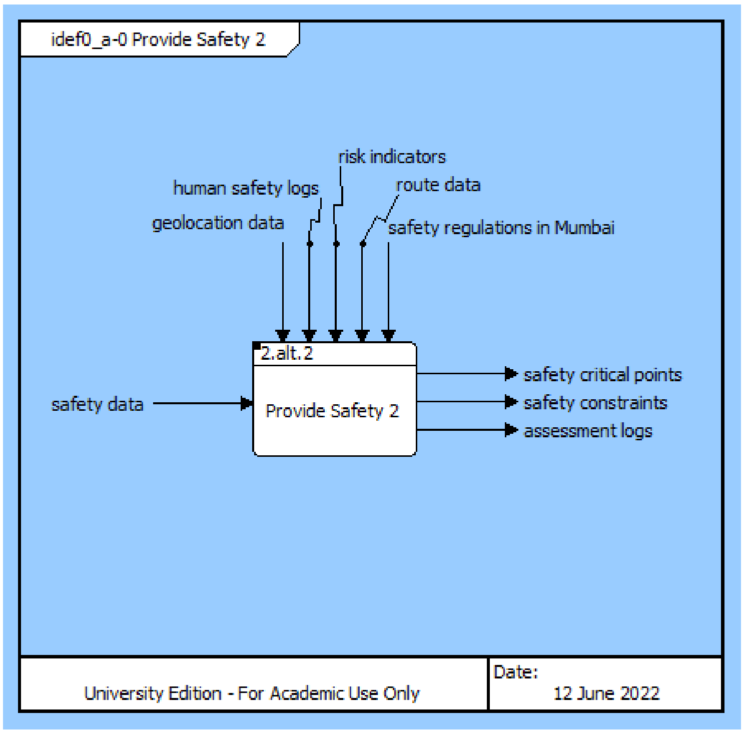

Figure 31: A0 diagram of the decomposition of function 2, alternative 2

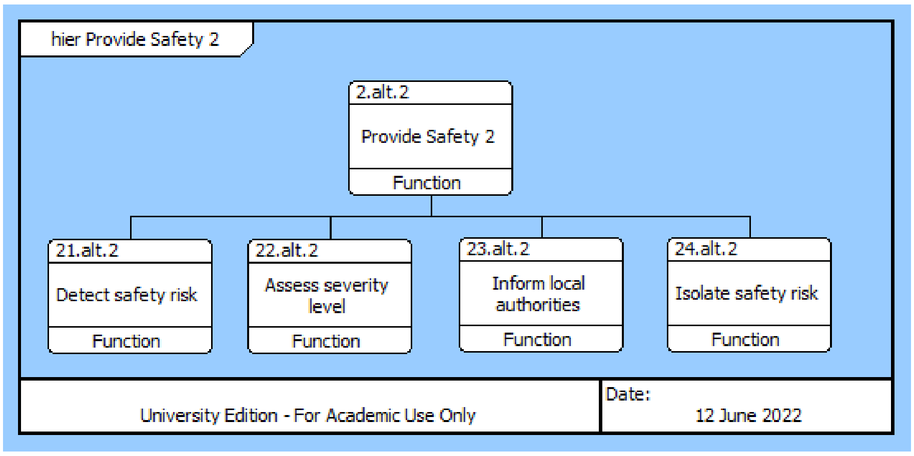

Figure 32: Hierarchy diagram of the decomposition of function 2, alternative 2

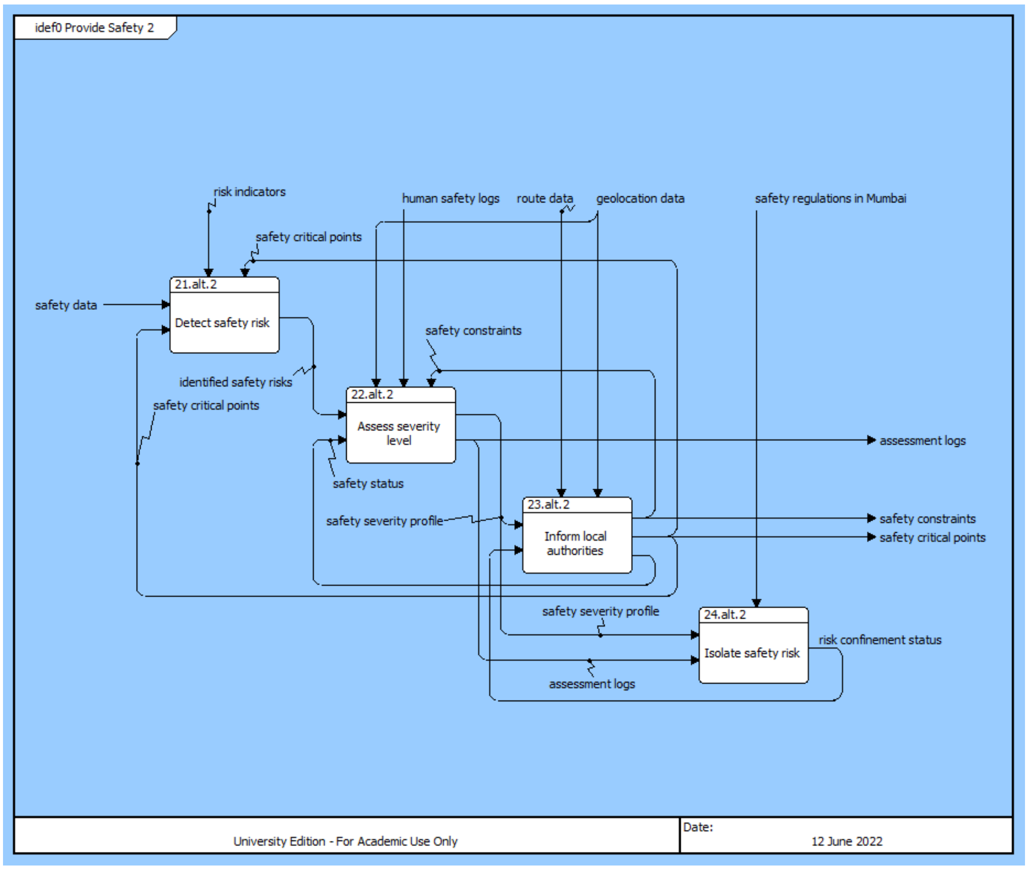

Figure 33: IDEF0 diagram of the decomposition of function 2, alternative 2

Alternative 3

Alternative 3 is a compromise between the two previous alternatives. It uses automation while remaining cost conscious. In other words it attempts to minimize cost while maximizing the efficiency of the process. It is worth noting that it does not embrace automation as much as alternative 1 does.

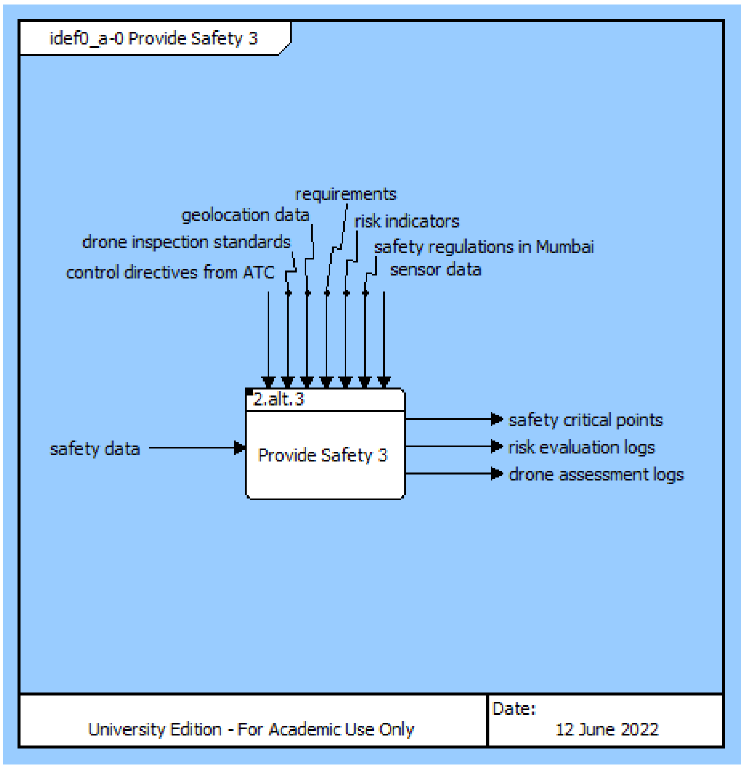

Figure 34: A0 diagram of the decomposition of function 2, alternative 3

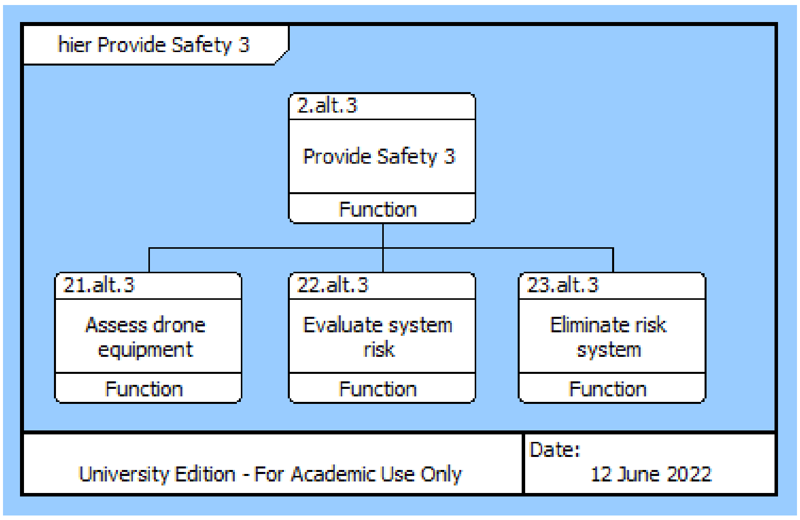

Figure 35: Hierarchy diagram of the decomposition of function 2, alternative 3

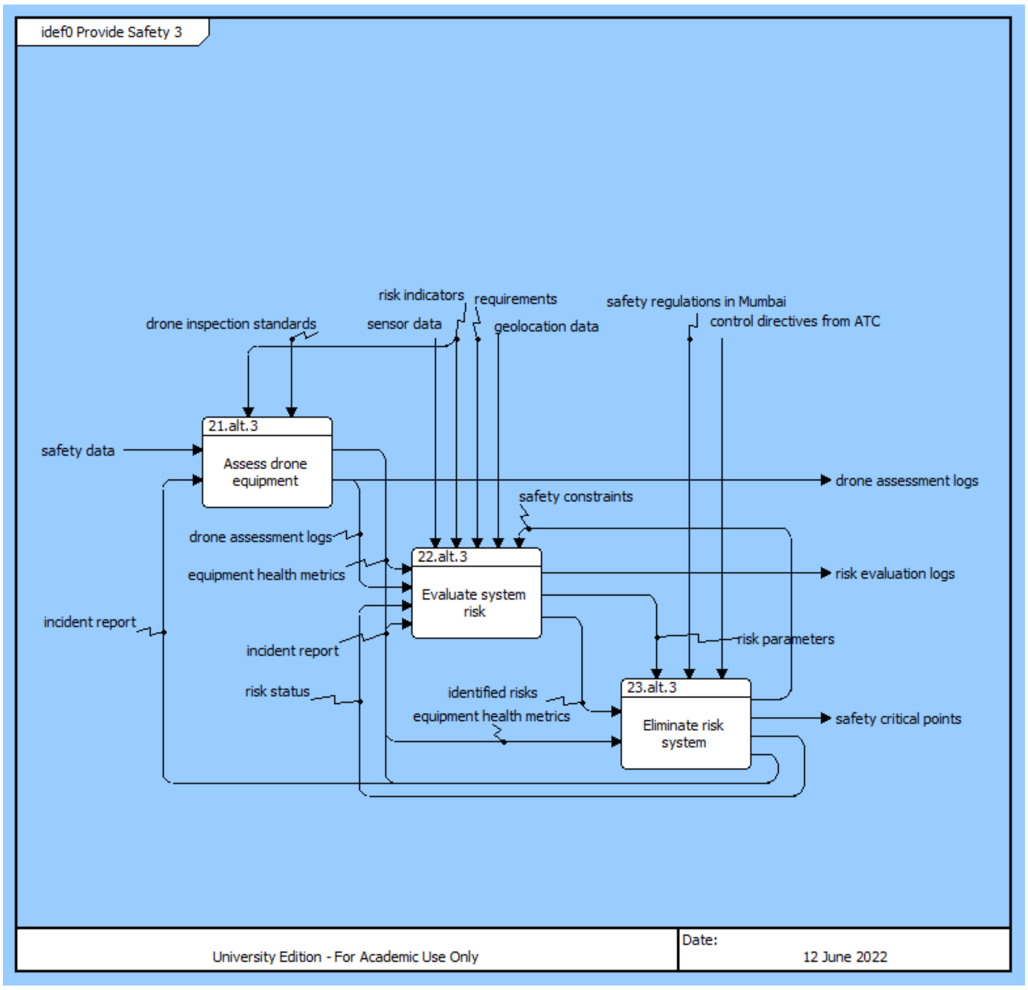

Figure 36: IDEF0 diagram of the decomposition of function 2, alternative 3

Selection of the most suitable alternative

With the fore-mentioned advantages and disadvantages in mind our team selected alternative 1 as the decomposition of choice for function 2. Safety is an utmost concern in our system and we believe that prioritizing the efficiency of the provision of safety far with automation far outweighs the cost of acquiring, configuring, and deploying the resources that are needed to achieve said efficiency.

For our purposes, alternative 1 is the most suitable choice. Furthermore, spending more on efficiently providing safety at the beginning will lead to cost savings in the long run for many reasons such as the reduction of the rate of damage and injury in our service points, warehouses, and charging stations which will directly lead to a decrease in expenses on repairs, medical treatment of employees, and litigation.

Decomposition of lower level functions

The 'Perform Drone Recovery' procedure is the key component of the chosen alternative this being due to the myriad of possible risks and legal complications that could arise after losing control over a drone.

The function is a well thought, 4 step process, that takes into account multiple indicators, real-time data feeds and historical factors. It also makes use of multiple software pieces, developed in-house that further simplify the flow of the process and as a last resort there is a specialised team of people who have all the necessary tools and training in order to facilitate the recovery of drones.

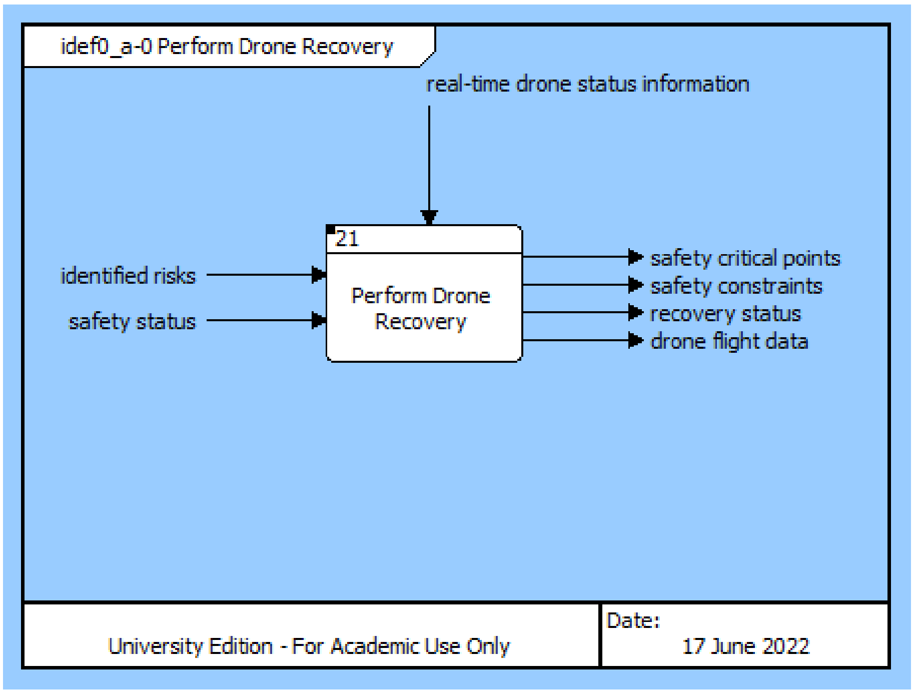

Figure 37: A0 diagram of the decomposition of function 21.

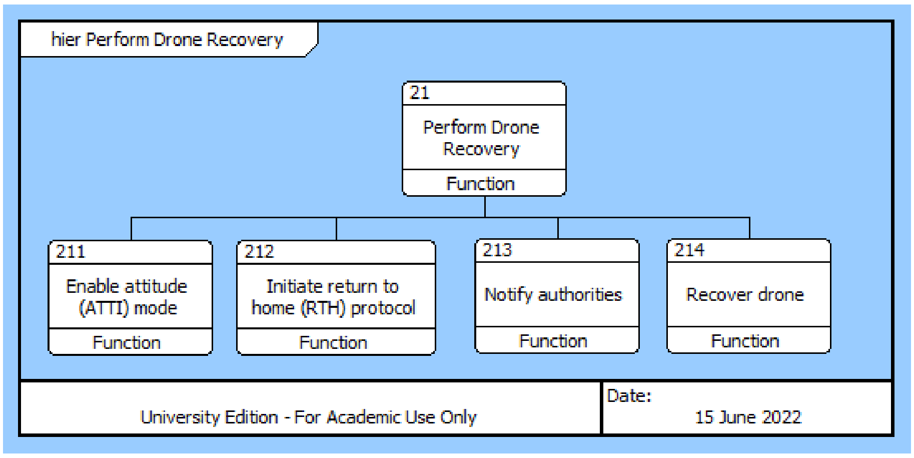

Figure 38: Hierarchy diagram of the decomposition of function 21.

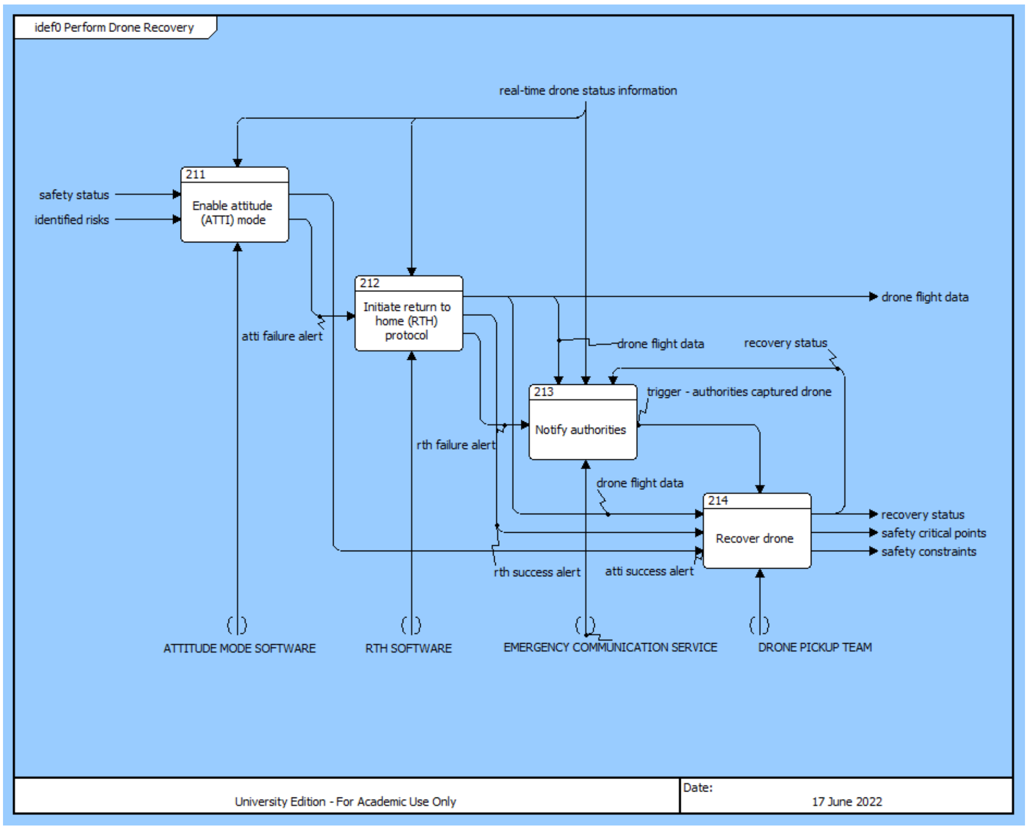

Figure 39: IDEF0 diagram of the decomposition of function 21.

The second most important process is responsible for assessing the risk level that the drone is currently exposed to. This provides us with a clear understanding of the internal and external factors that contributed or could contribute to a possible malfunction. The process also takes into consideration plenty other data sources provided by the other processes in the function and historical factors in order to provide a quick and accurate assessment. The whole process consists of 3 subdivisions, each programmed to do the job as efficient and swift as possible using special analysis and risk identifications software and human supervision.

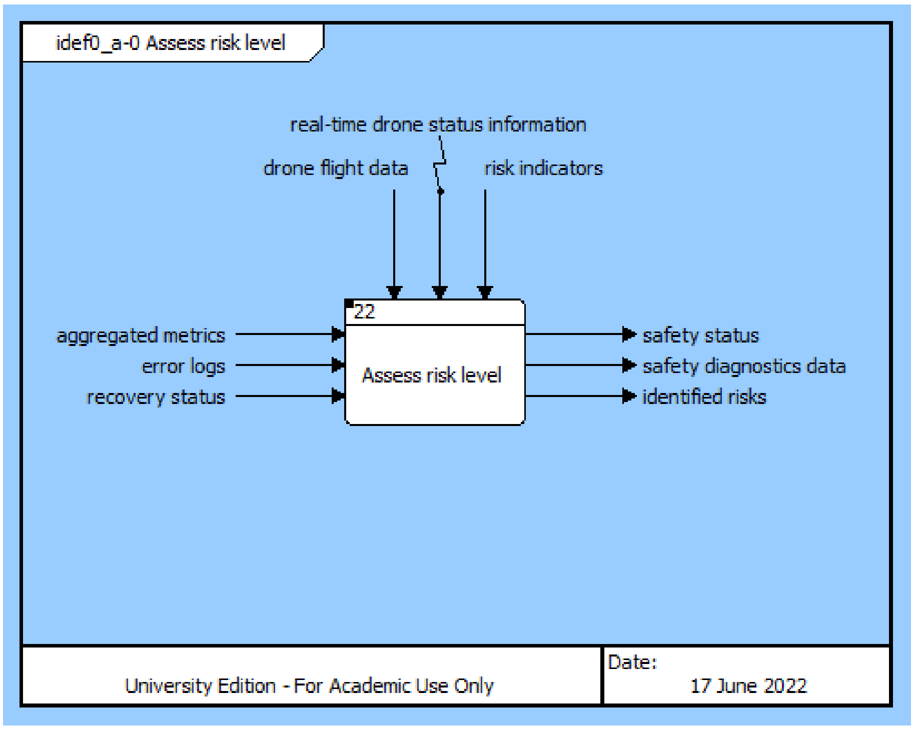

Figure 40: A0 diagram of the decomposition of function 22.

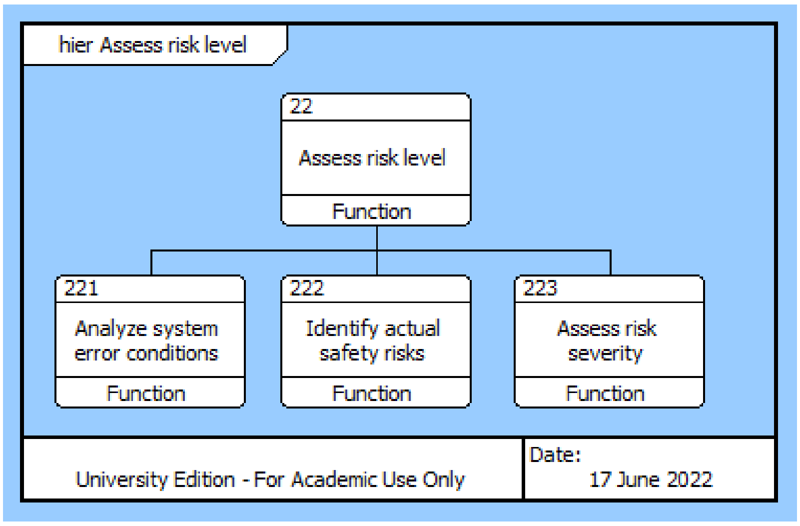

Figure 41: Hierarchy diagram of the decomposition of function 22.

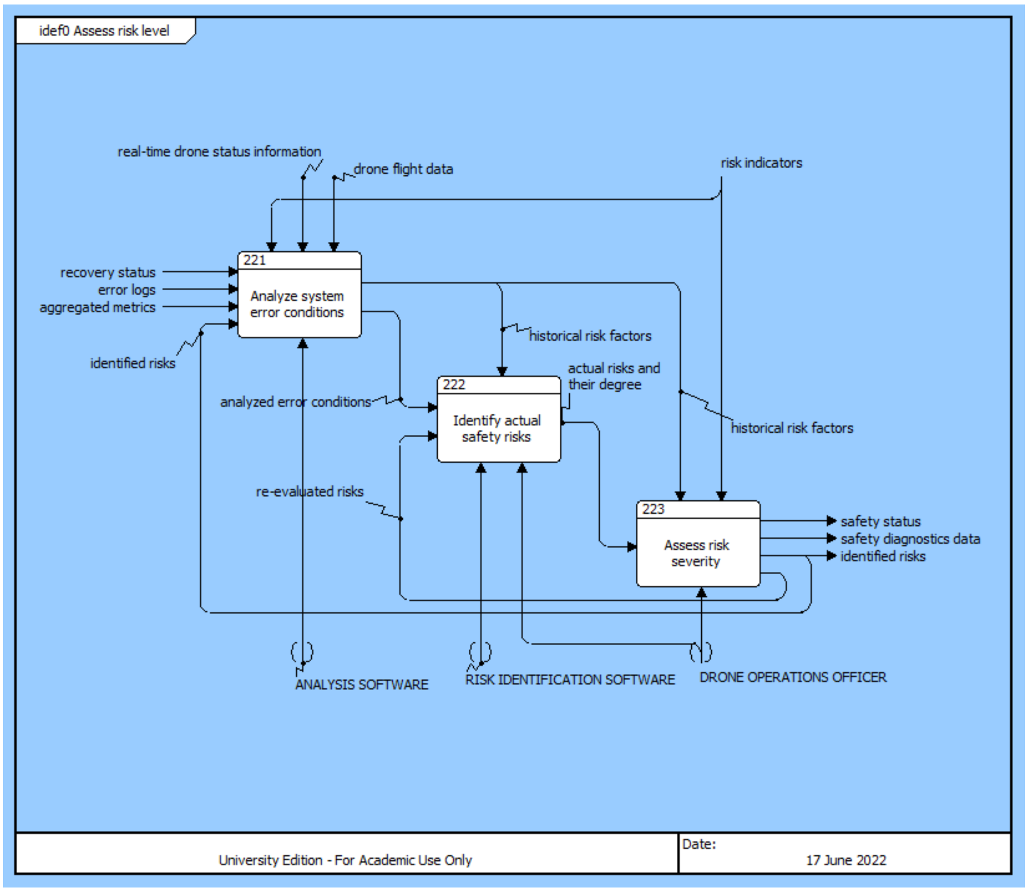

Figure 42: IDEF0 diagram of the decomposition of function 22.

The third and final process is responsible for calculating the safety of the drone while it is on the go. Monitoring and knowing the current integrity status of each drone is crucial to keeping operations smooth and detecting any faults in time, therefore a lot of data, from multiple relevant sources is being analysed by specialised software. The hardware is also routinely checked by the equipment specialist which generates a report whose data is analysed by the data transformation software together with the data gathered by the state parsing software.

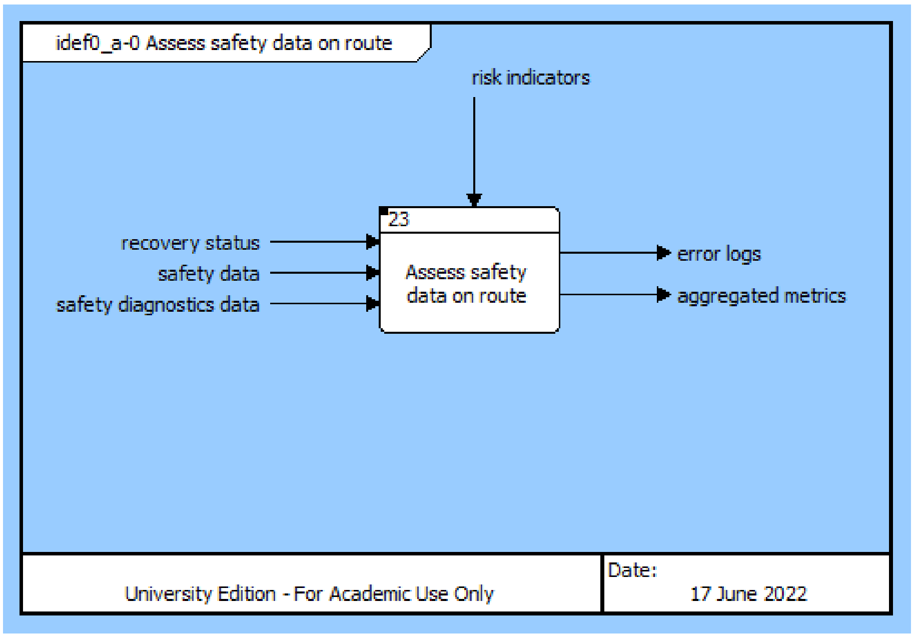

Figure 43: A0 diagram of the decomposition of function 23.

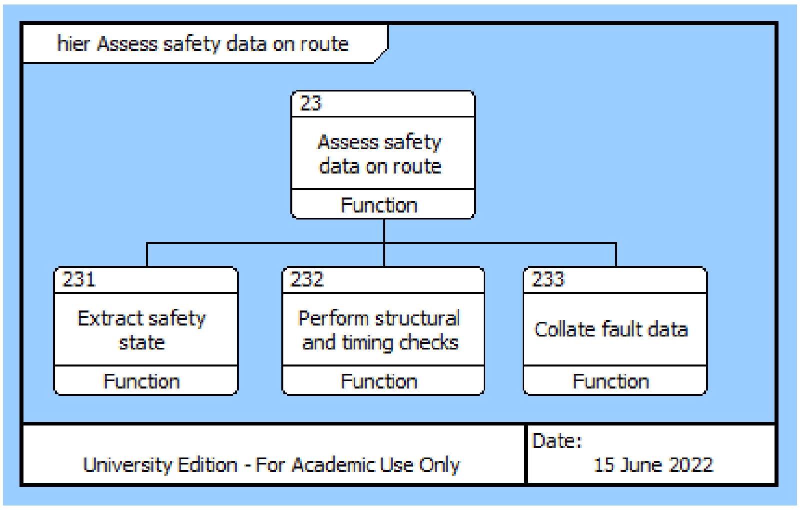

Figure 44: Hierarchy diagram of the decomposition of function 23.

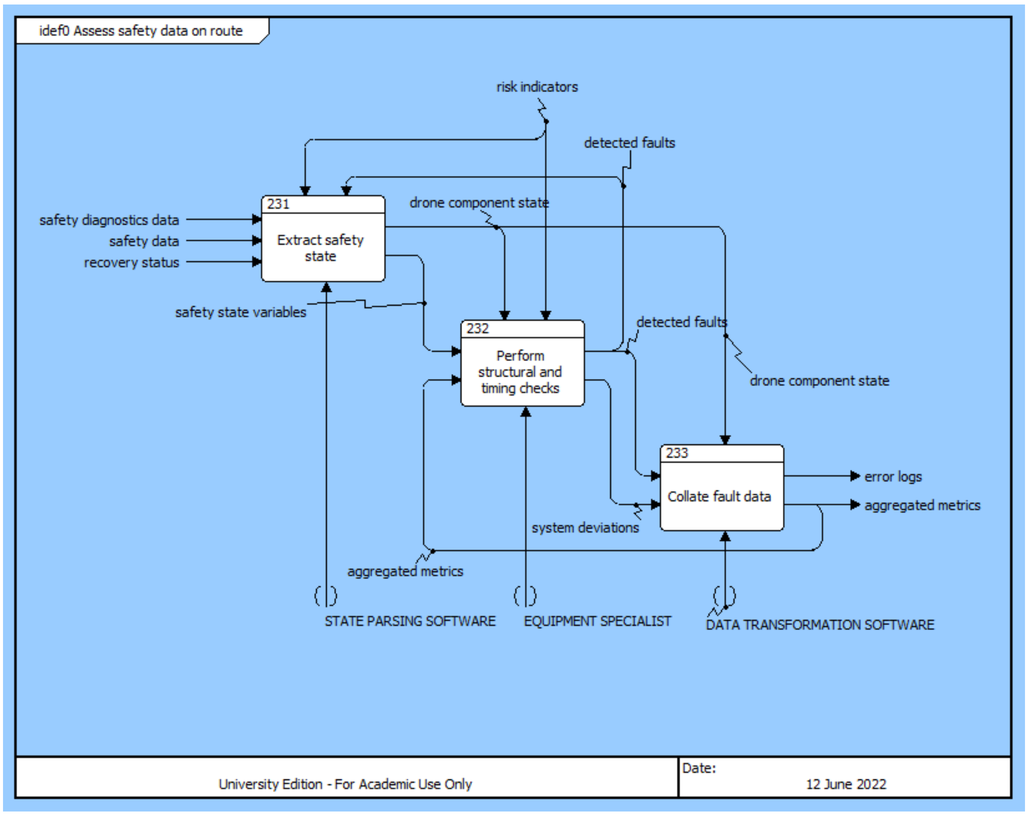

Figure 45: IDEF0 diagram of the decomposition of function 23.

4.3.3 *Decomposition of function 3*

The monitor function analyses the behaviour and generate data which is important for smooth operation of the system. It is segregated into 4 main component which deals with security, safety , maintenance and journey. Security data and safety data makes the system robust. They are interlinked and is extremely important to work hand in hand. A lapse in safety data could compromise the entire system and hence it is important to update them with the norms and additional requirements required by the function. The drone maintenance has to comply with the standards and this can be only initiated once the approval is granted by the safety officer. The final component gives a real time study and performs a simulation to understand the behaviour and risk involved for the possible journey and the once it leaves the monitor function, an approved drone with compliance to safety and security is obtained and is ready for the final function.

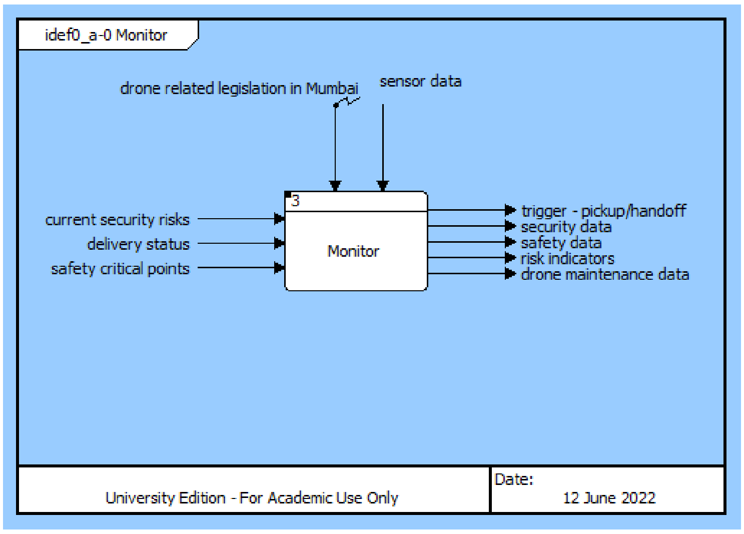

Figure 46: A0 diagram of the decomposition of function 3.

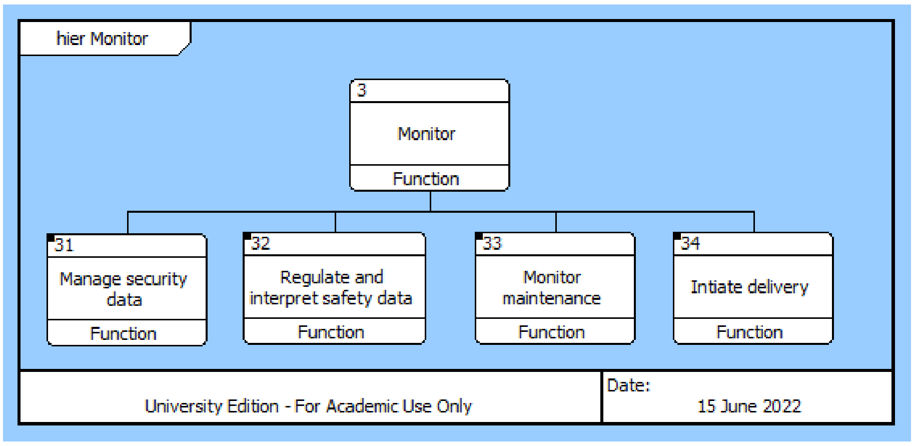

Figure 47: Hierarchy diagram of the decomposition of function 3.

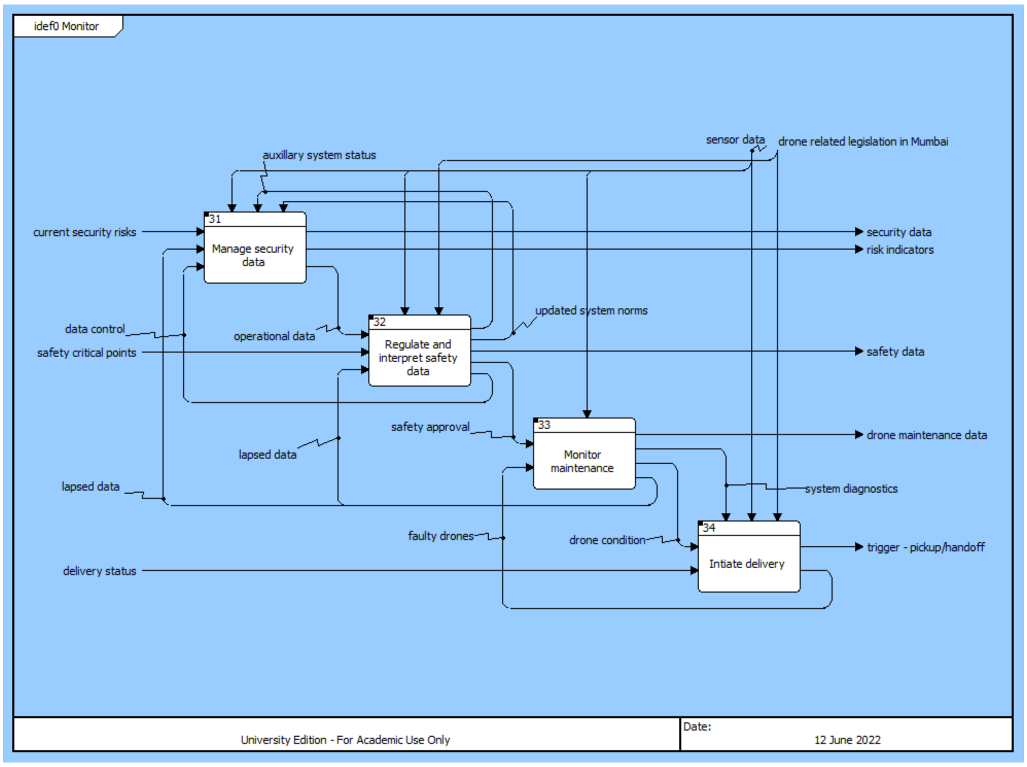

Figure 48: IDEF0 diagram of the decomposition of function 3.

Decomposition of lower level functions

It is essential for system to only accept data which are essential for it's operation. Security data should be handled with care and this is carried out in this particular function. The data moves through three channels in order to generate security data and in addition, identifies risk indicators. To prevent deviation and malfunction, the system is updated frequently to meet the norms set by the stakeholder. All sorts of data is fed into the system and it has to be screened before it moves for fault detection. Minor faults are fed back to the system and additional parameters are established to make the data processing more efficient. Security and operational data is generated once it gets compiled after thorough checks and clean up.

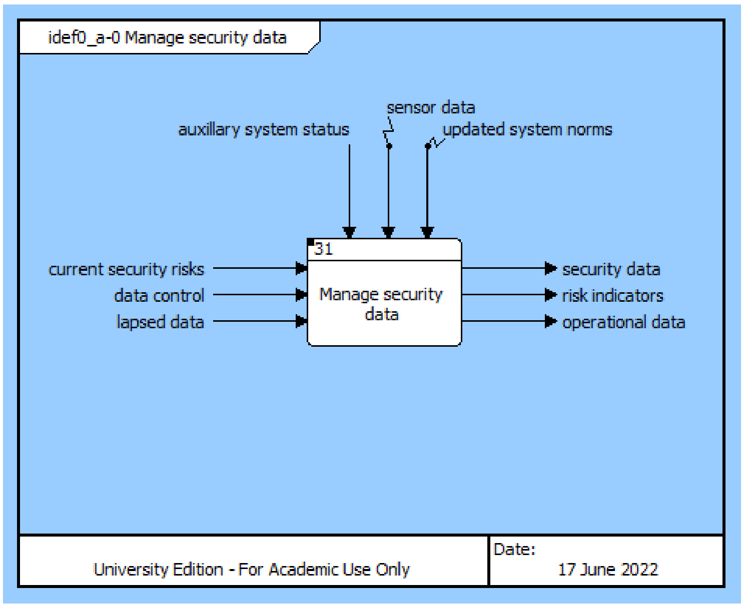

Figure 49: A0 diagram of the decomposition of function 31.

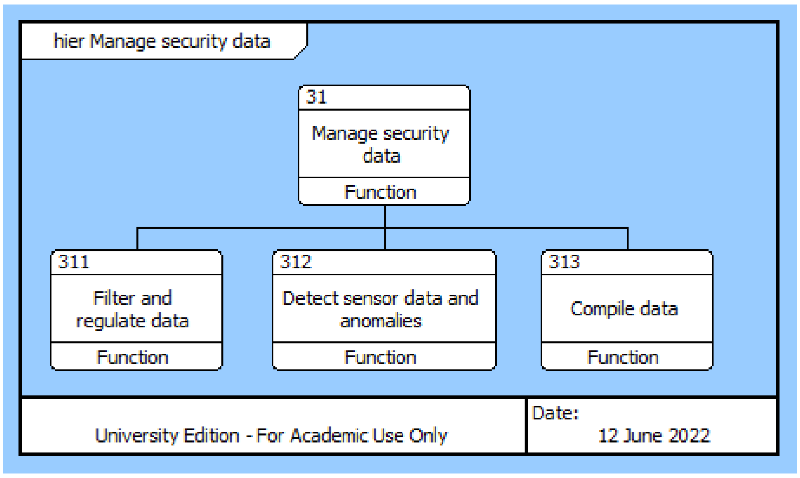

Figure 50: Hierarchy diagram of the decomposition of function 31.

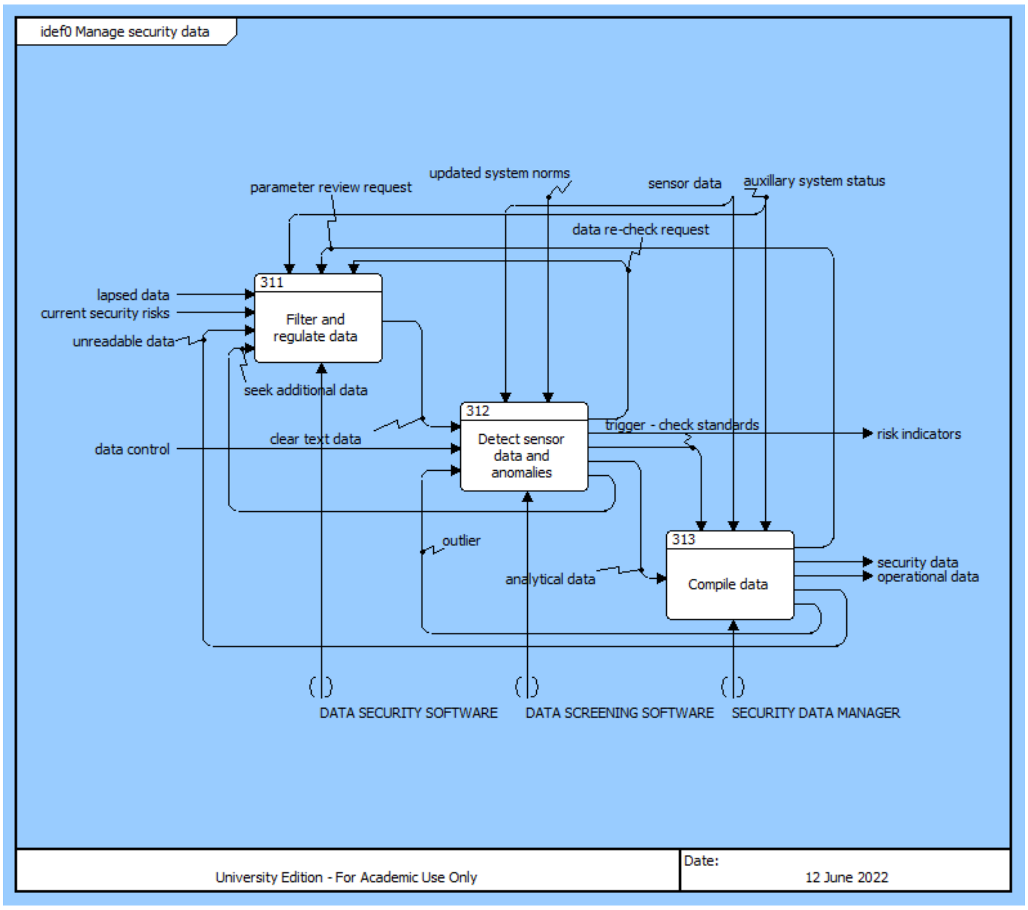

Figure 51: IDEF0 diagram of the decomposition of function 31

Safety data contains valuable information and instructs the system how to function efficiently. It is important to generate data which is reliable and consistent. The data should not violate any of the constraints set by the stakeholders and hence undergoes quality assurance check to ensure that the data complies with the safety critical points. By constant generation of quality data and updated protocols, safety assurance can be optimised as the system utilises the most recent data. The final component approves the data and informs the security function with the updated norms and provide additional requirements for improved performance. Once the safety approval is established, a back up data is generated and is then stored in the initial channel so that it could be available on request.

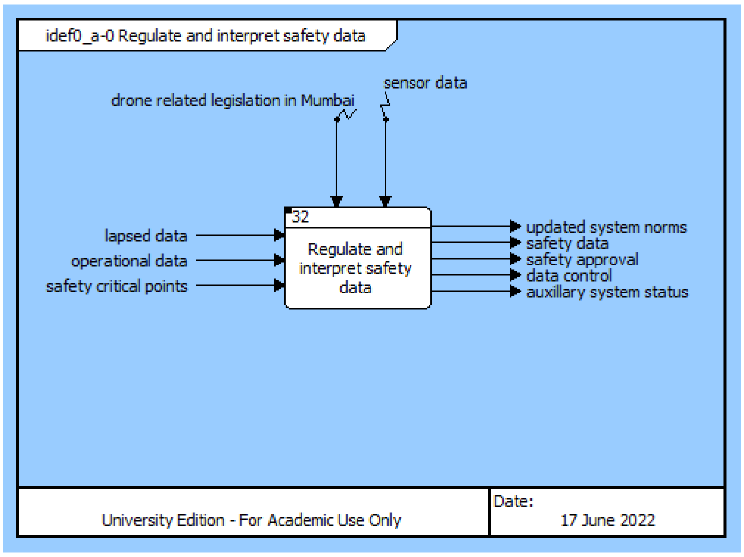

Figure 52: A0 diagram of the decomposition of function 32.

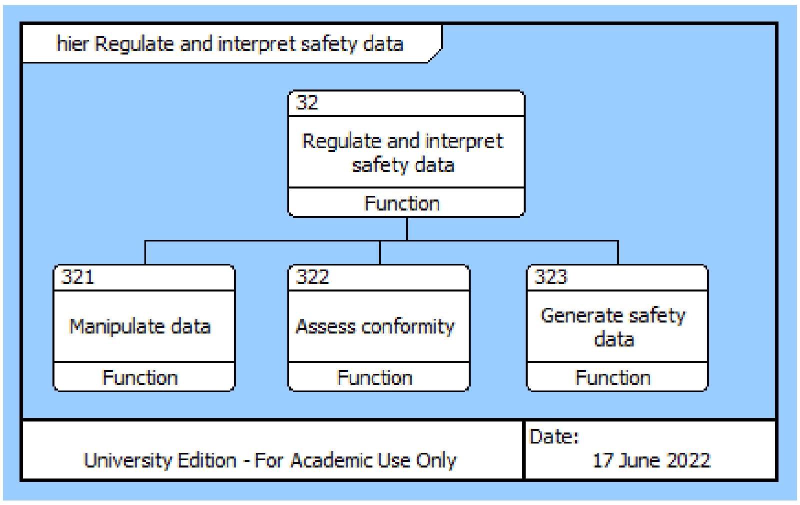

Figure 53: Hierarchy diagram of the decomposition of function 32.

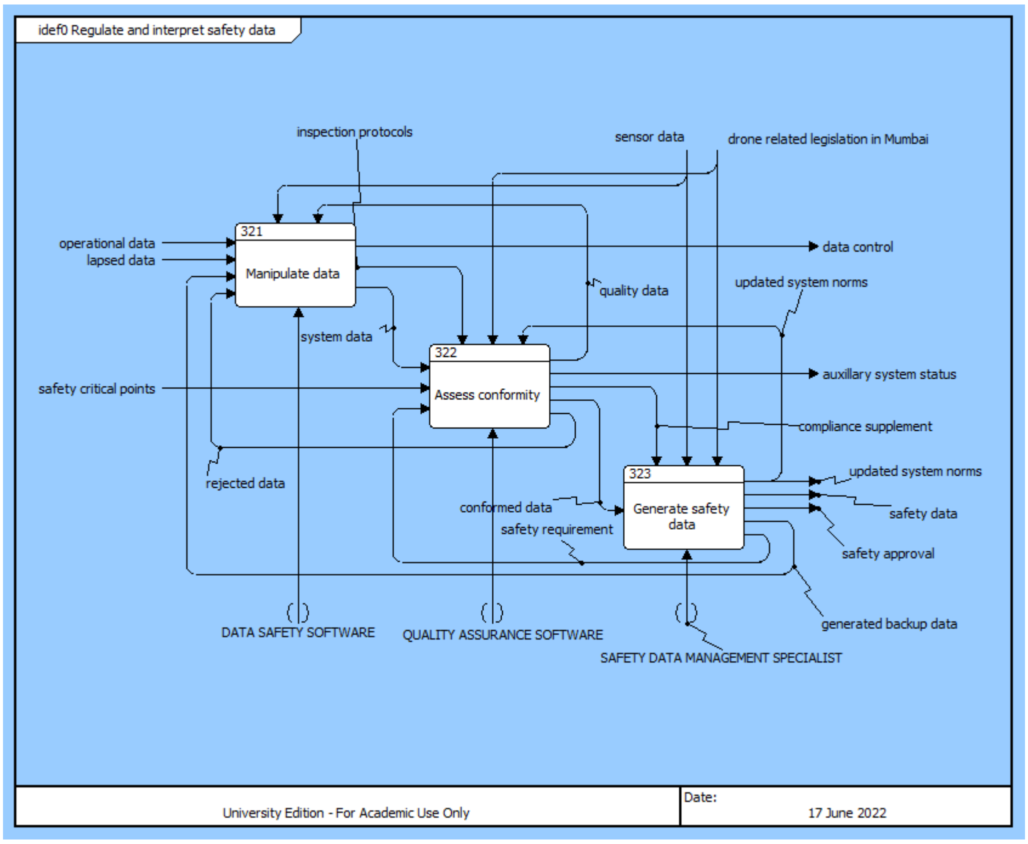

Figure 54: IDEF0 diagram of the decomposition of function 32

Monitor maintenance will provide the system with the ability to inspect battery and structure of drones. Eventually the condition of the drone will be evaluated to determine whether the drone is in a condition to operate or not. This will satisfy the system's need to evaluate whether a specific drone is reliable to execute an assignment or not. Drone condition will determine whether a drone is capable to be selected and assigned a flight number and journey or not. Stakeholders will be satisfied in case inadequate drones are never selected and therefore the system's efficiency and service level will not be affected. Creating a system which will provide valid information regarding the drone's condition is considered crucial, since the operating drones do have a substantial part in the system's performance.

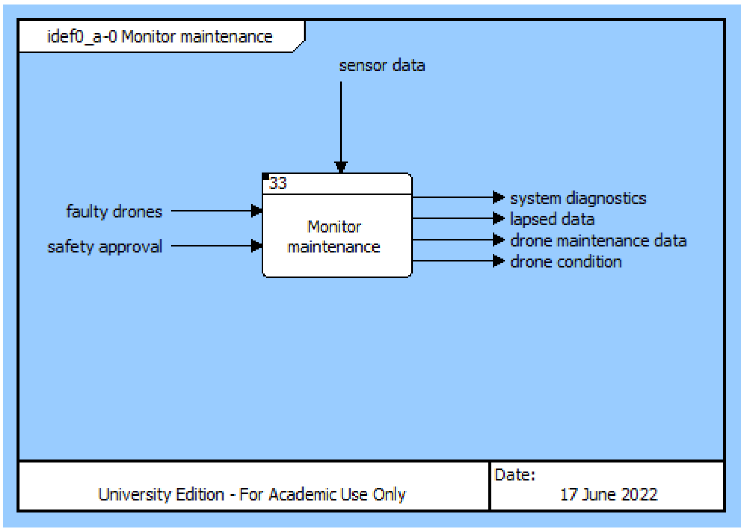

Figure 55: A0 diagram of the decomposition of function 33.

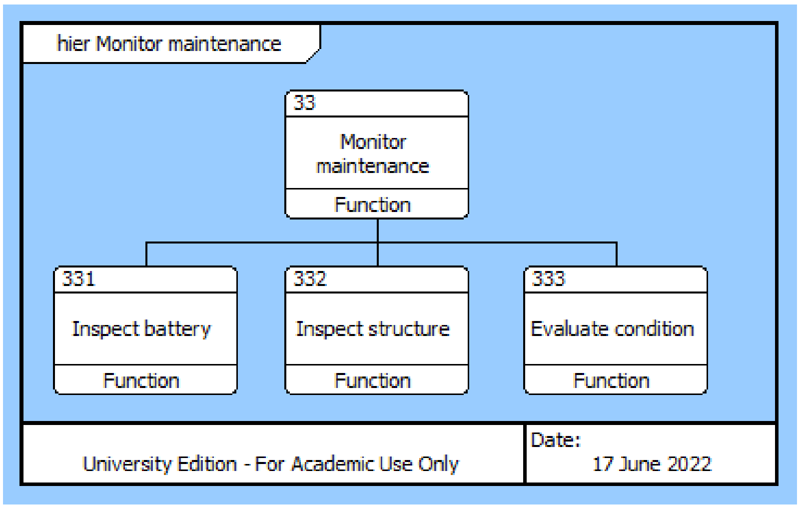

Figure 56: Hierarchy diagram of the decomposition of function 33.

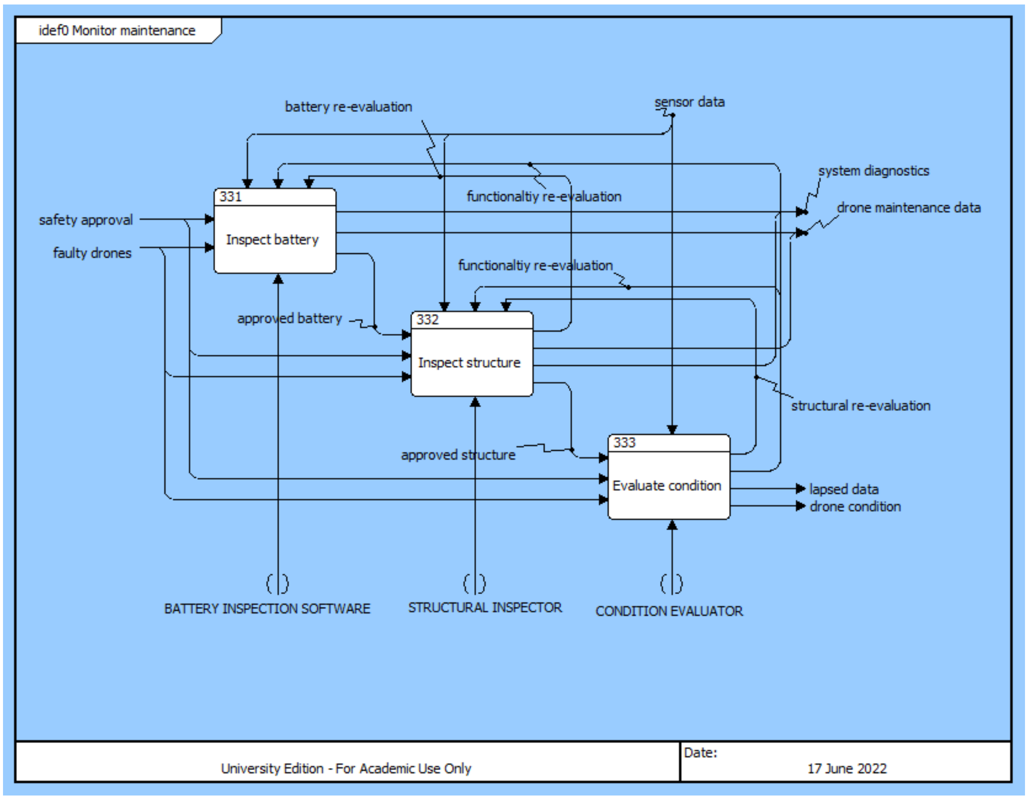

Figure 57: IDEF0 diagram of the decomposition of function 33

After having valid information regarding the drone's condition, the systems will be enabled to select a drone and assign a flight number for each assignment (delivery). Depending on the specific task the drone needs to carry out, the journey of the drone should be defined and simulated, in order to provide crucial information both for the system and the customer. This information will define the approximate duration of the processes (preparing food, delivery pick-up, commuting to customer, and delivery drop-off). By doing so, this information should be evaluated afterwards on how realistic it was. This will enhance the system's performance and identify malfunctions. Stakeholders should be satisfied when a system operates in a transparent way. The end-user of the system (customer) should also be satisfied the most. This should be achieved by providing trustworthy information regarding the duration of the delivery (from the starting point of ordering until the ending point of receiving an order). All in all, calculating the approximate time of the journey after the journey is defined and simulated, means that the customer and the system should be provided with information which will be evaluated after initiating the journey (and completing it).

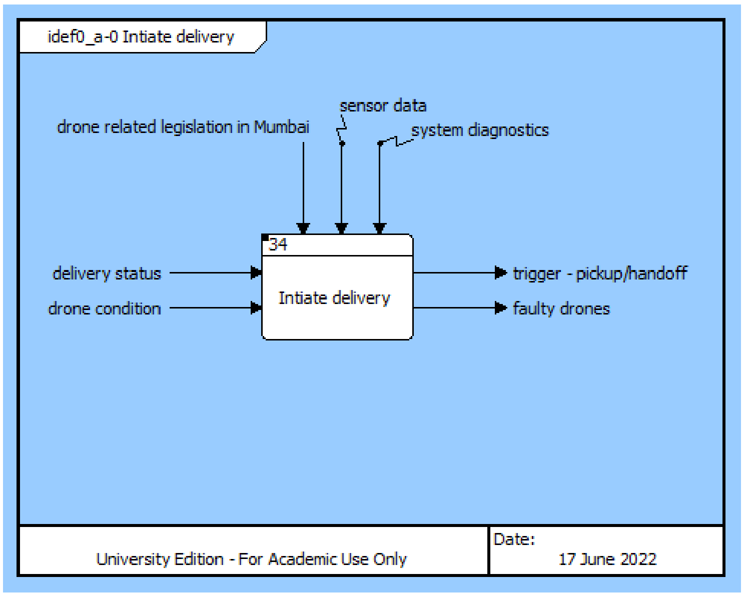

Figure 58: A0 diagram of the decomposition of function 34.

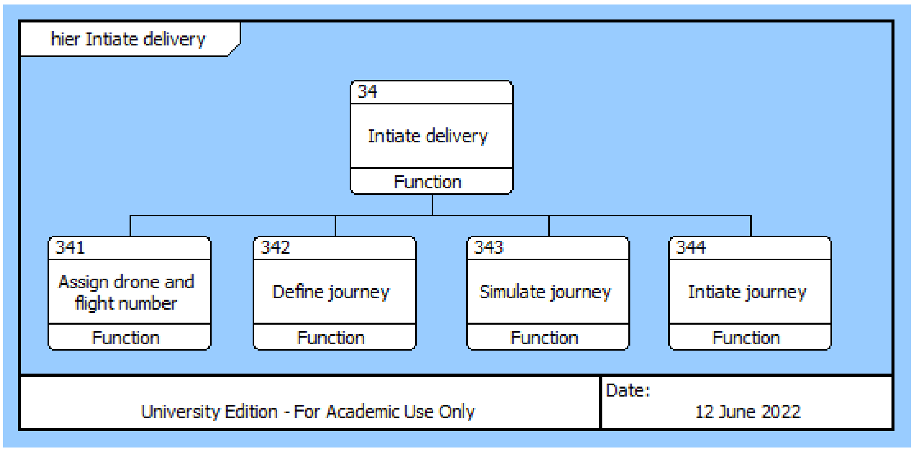

Figure 59: Hierarchy diagram of the decomposition of function 34.

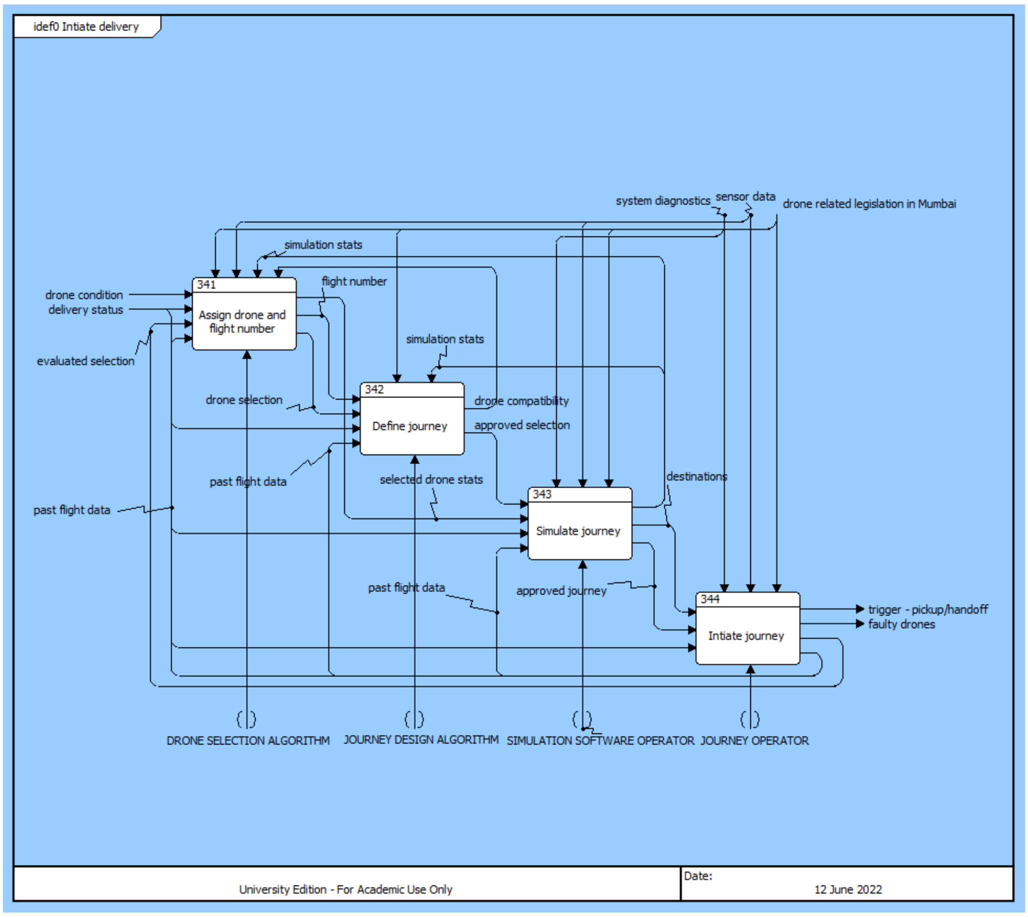

Figure 60: IDEF0 diagram of the decomposition of function 34

4.3.4 *Decomposition of function 4*

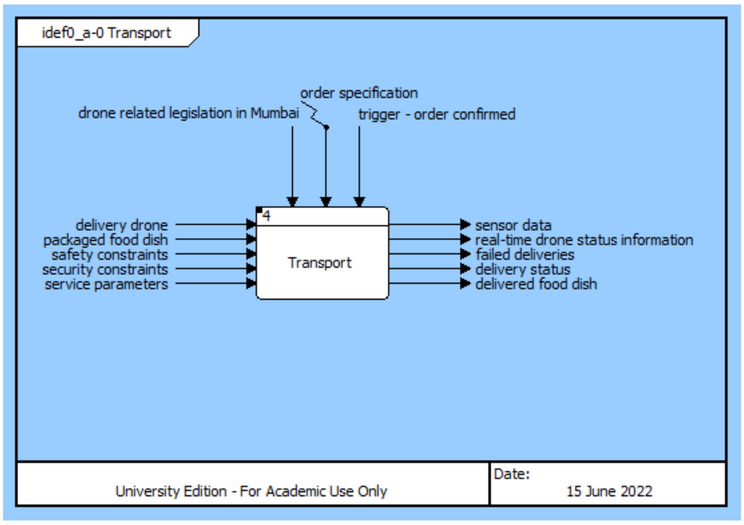

Figure 61: A0 diagram of the decomposition of function 4.

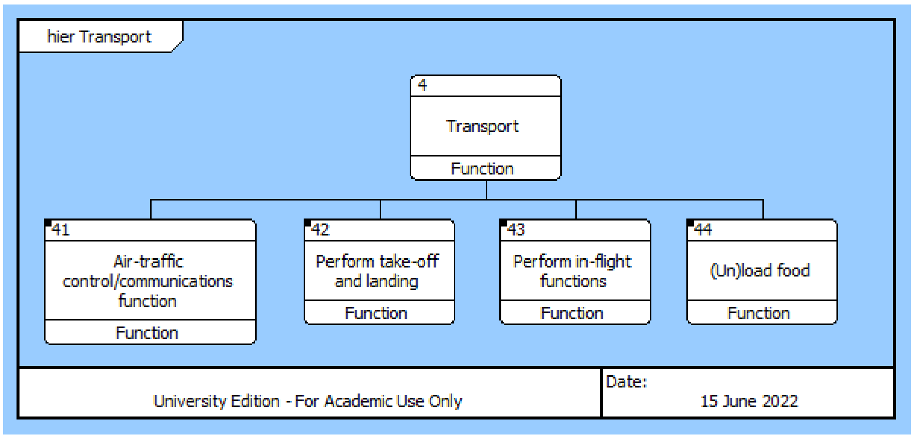

Figure 62: Hierarchy diagram of the decomposition of function 4.

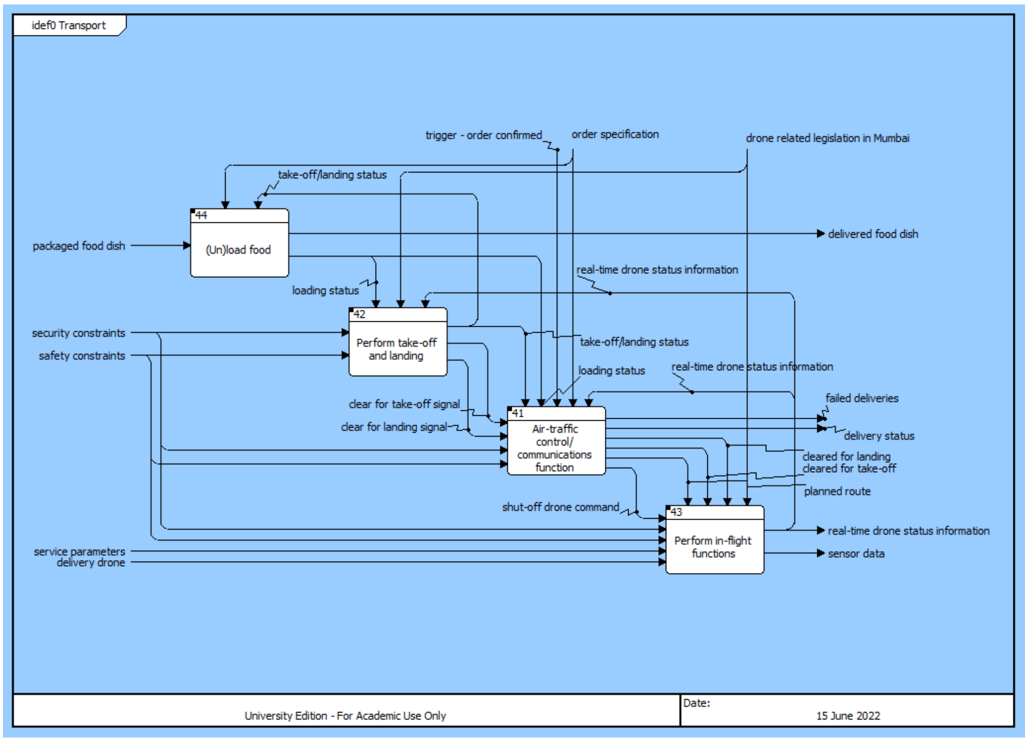

Figure 63: IDEF0 diagram of the decomposition of function 4.

Decomposition of lower level functions

The transport function has been decomposed into four functions that were derived from the use case of the drone transporting food to a customer. During this exercise, it became clear that communicating with the air traffic control was the most important sub-function. While in figure 54 the focus is set on the drone side of the communication. It is worth noting that a human is listening on the other side of the communication system. This would mean that the drone flight data must be transformed in a digestible manner for the person listening. In our case, we make use of a middle man between the drone and the air traffic controller which is our own Transport communications and operations department. Another idea implemented is to send new orders, to drones that are charging in relatively close proximity to the destination. This is why their order specification is used as a control for assessing drone charge planning.

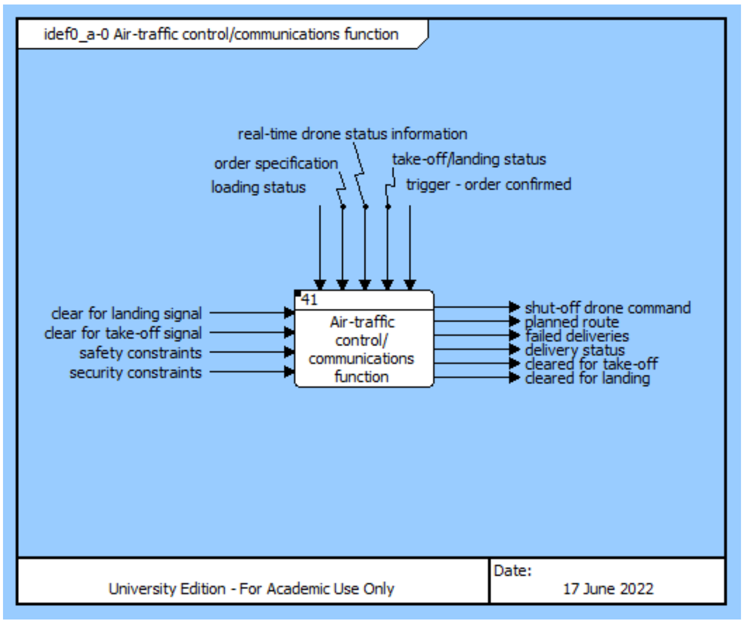

Figure 64: A0 diagram of the decomposition of function 41.

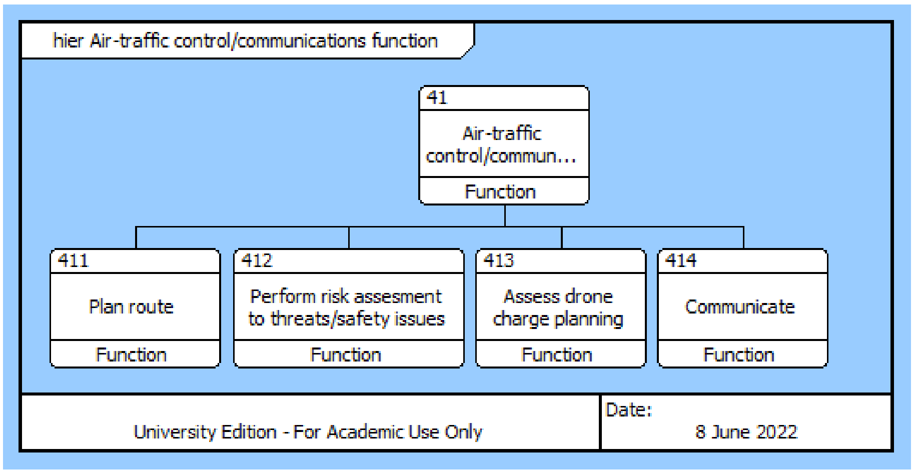

Figure 65: Hierarchy diagram of the decomposition of function 41.

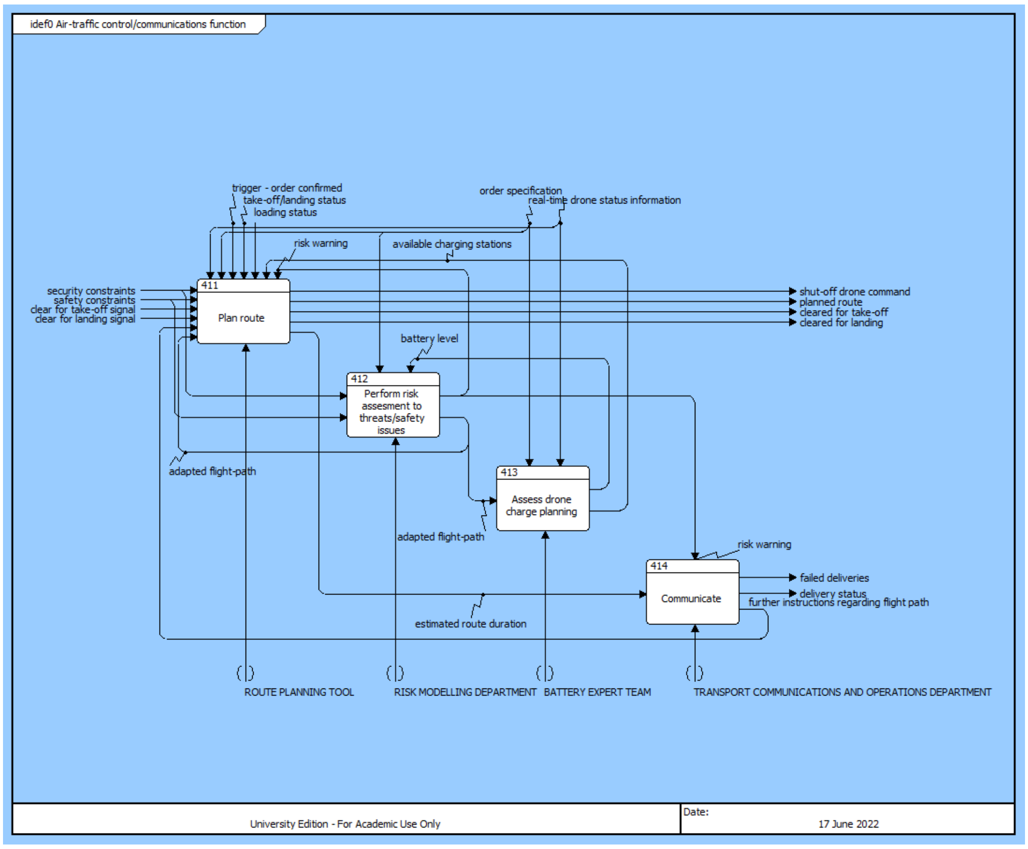

Figure 66: IDEF0 diagram of the decomposition of function 41.

Figure 56 is very straight forward. However there is one aspect that could use a clarification. The idea is that no drone is allowed to just land or take off once the correct signal was given. As the most important aspect of the system was security and safety, it is important to think of the people who might be close to a landing area. This is why the functions 422 was added to the system. This functions will make use of flashing lights and sound alarms to warn the people around the drone. This explains the physical component attached to the functions. Further more, our team realised that for the drone to land in a particular spot, it has to have clearance to do so. This is why the drone legislation of Mumbai are used as controller by 421.

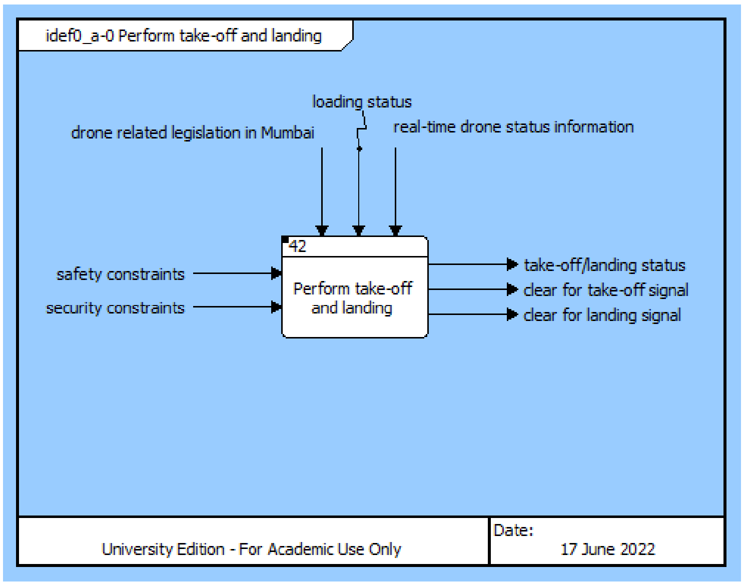

Figure 67: A0 diagram of the decomposition of function 42.

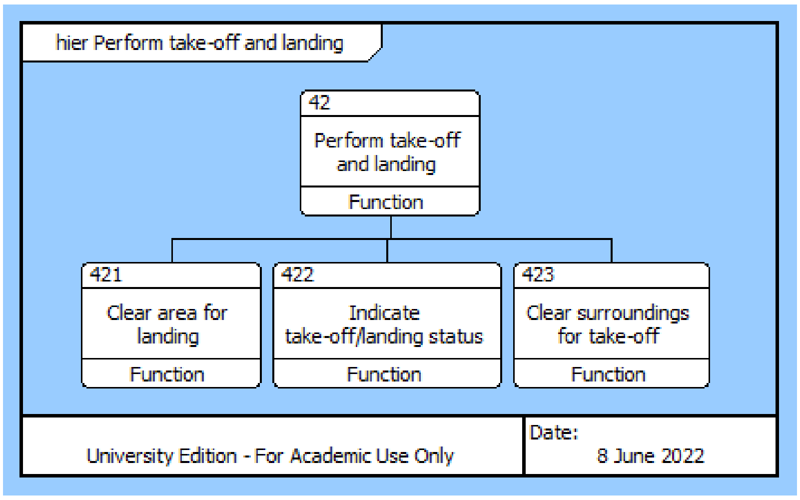

Figure 68: Hierarchy diagram of the decomposition of function 42.

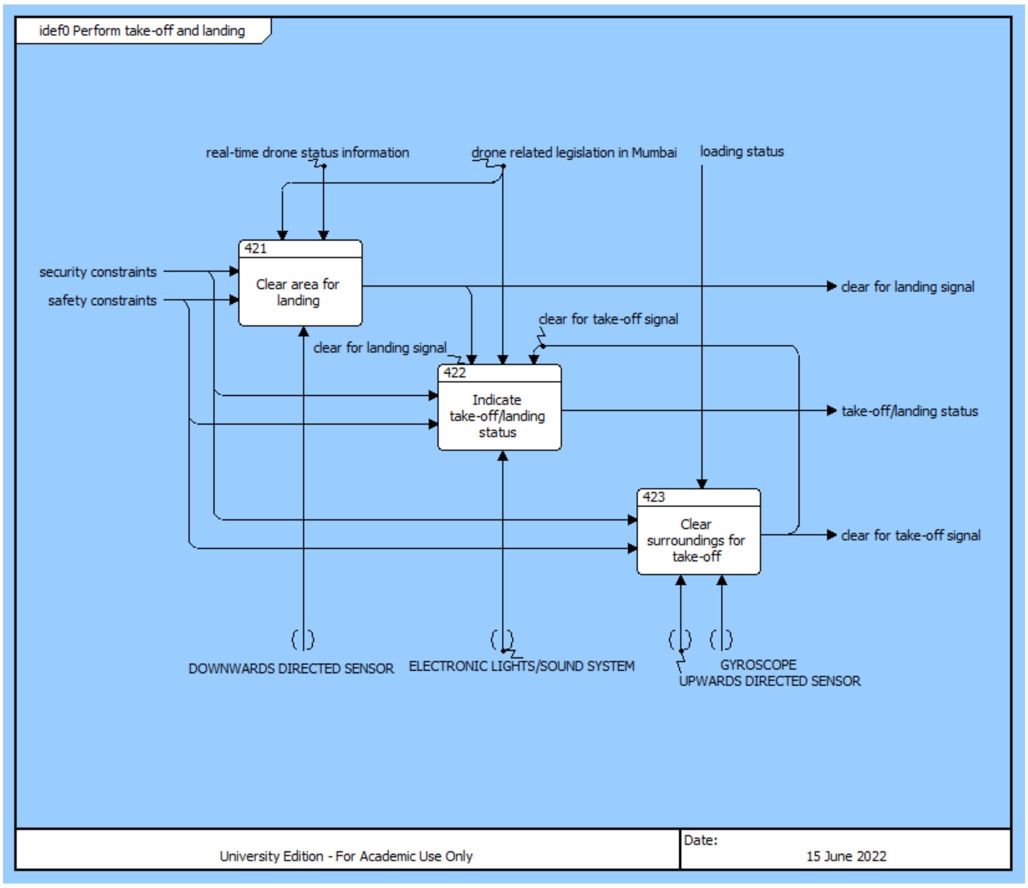

Figure 69: IDEF0 diagram of the decomposition of function 42.

In figure 58 we distinguish between the actual drone flying and the base drone model that was delivered to the company. The function 431 uses the drone model to deliver required data about the drone and its physical architecture for the system to control the drone correctly. The system, therefore, remains flexible for the type/model of drone used. The physical bare-bones drone used is called drone bases in this diagram. However, the actual drone that will be transporting dishes will be the drone bases with additional gadgets attached to it, which will allow the drone to satisfy all other functions that the drone has. This complete version of the drone is what we refer to as the drone.

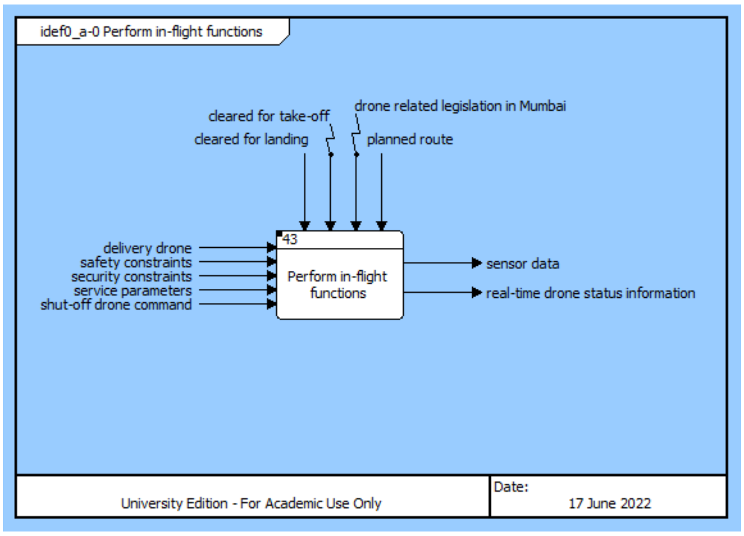

Figure 70: A0 diagram of the decomposition of function 43.

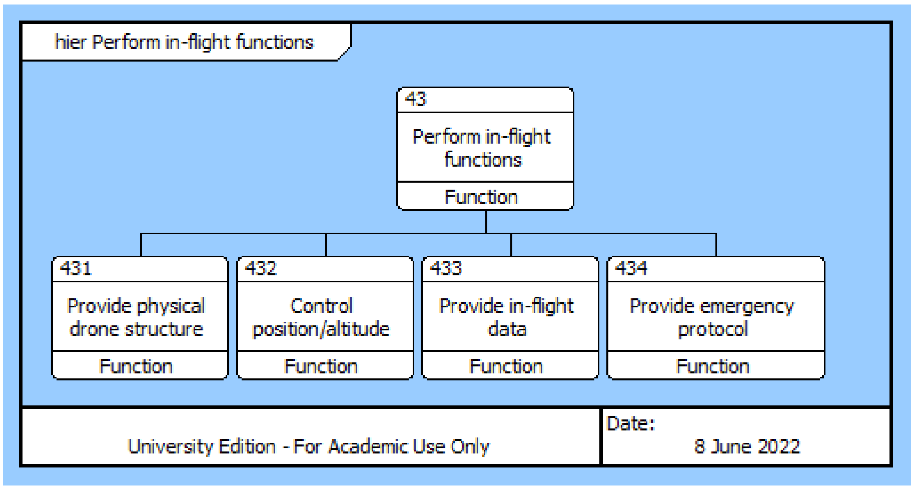

Figure 71: Hierarchy diagram of the decomposition of function 43.

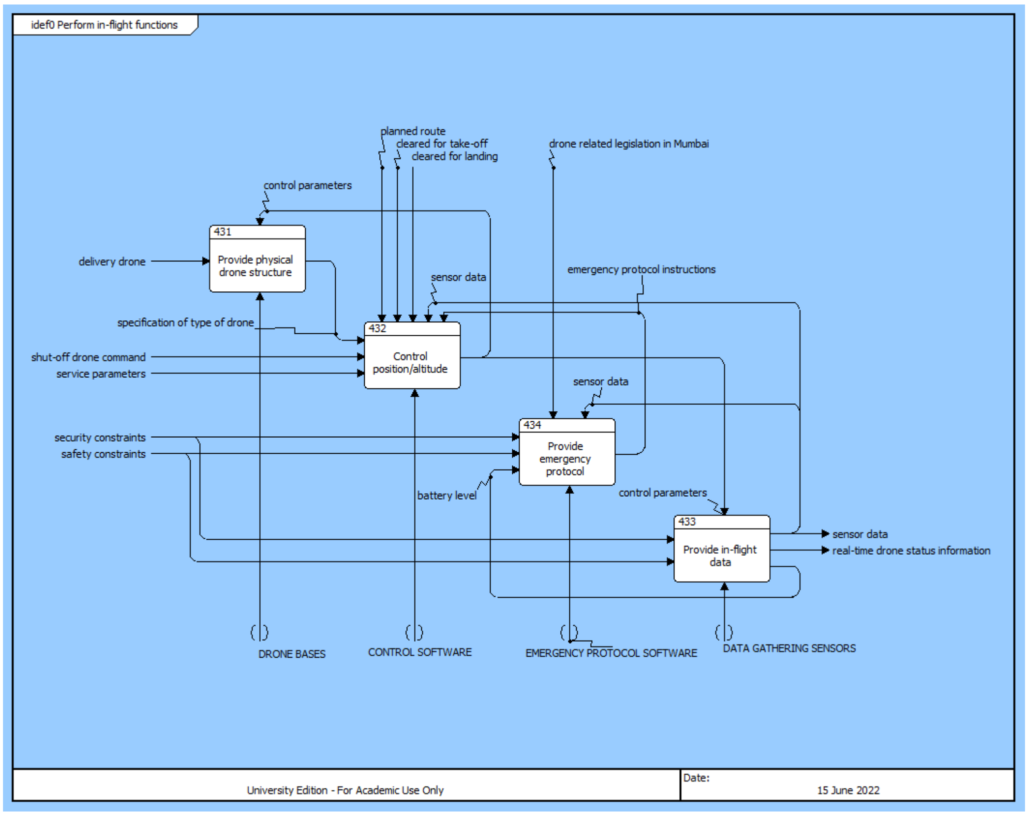

Figure 72: IDEF0 diagram of the decomposition of function 43.

When designing the diagram in figure 60, our team considered three different ways to deliver the dish. One real-world example is to deliver the food by dropping it in front of the homes, in the same way as blood is dropped by mini air planes in Africa. However since we are working with food, this idea was rejected. The next idea is to gently place the food at the door. However this system also has the problem of deciding where to place the food and there is a chance to have the dish stolen by someone nearby. Therefore the final solution was used which is to land the drone near the delivery location and have to consumer open the compartment box himself. This way all of the above problems are negated. Further more one can add an authentication system, which will check if the person taking the dish is the actual customer. This will allow the drone to immediately notify the system that the delivery was successful. Thus the consumer is added here to the system as part of the function.

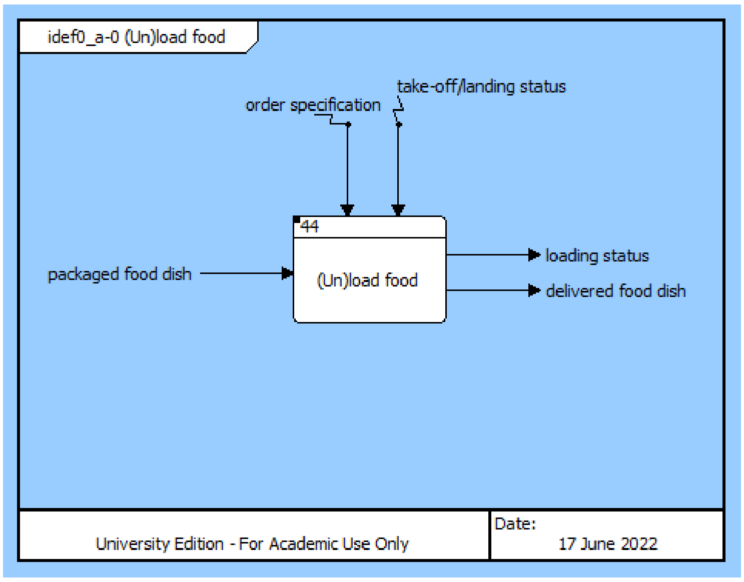

Figure 73: A0 diagram of the decomposition of function 44.

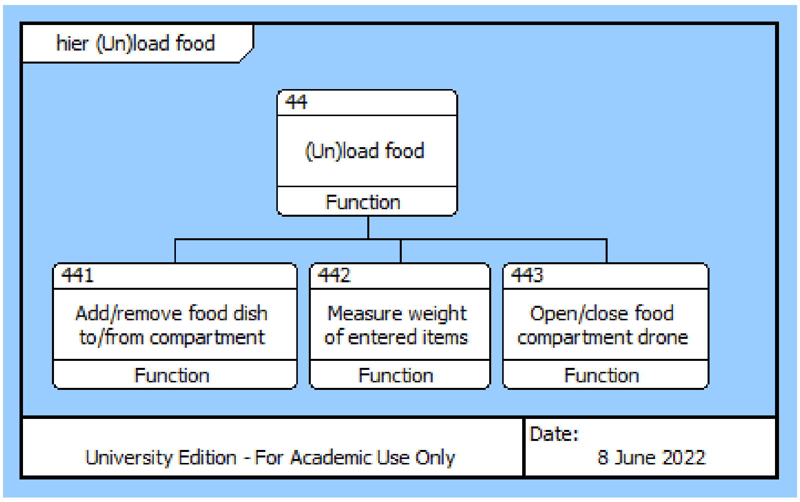

Figure 74: Hierarchy diagram of the decomposition of function 44.

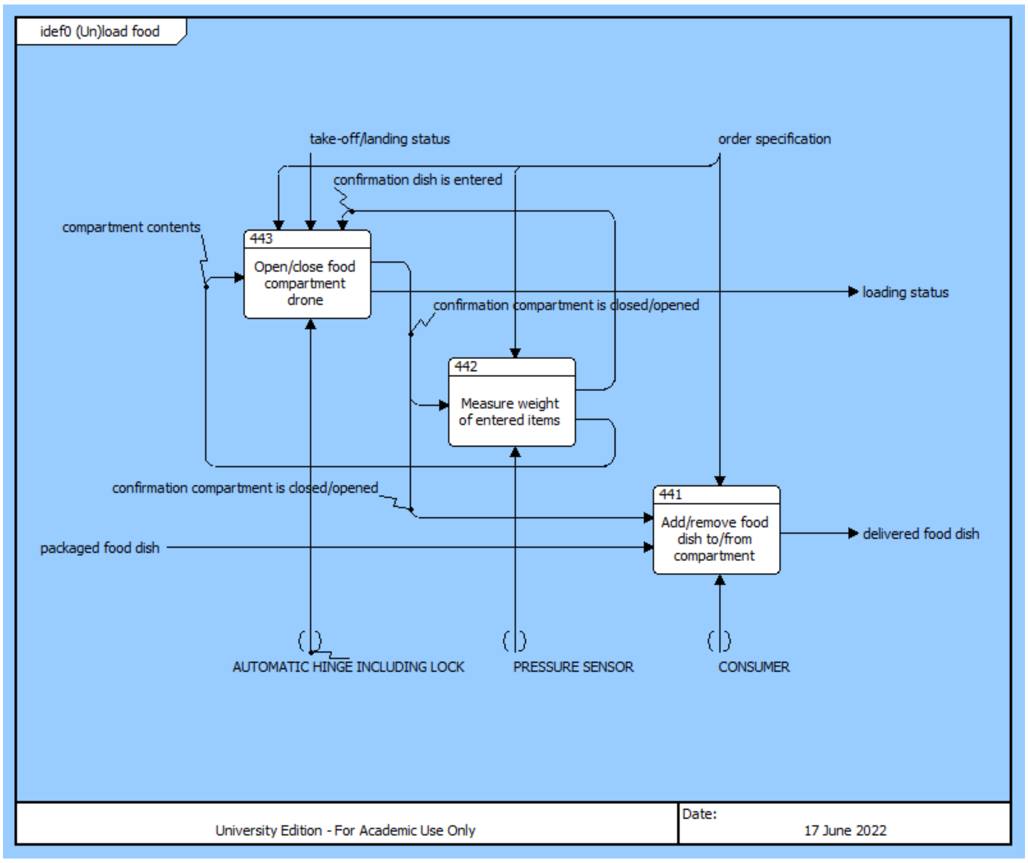

Figure 75: IDEF0 diagram of the decomposition of function 44.

5 PHYSICAL ARCHITECTURE

From our functional architecture, we build a physical architecture. This is achieved by grouping the leaf functions of our FA into logical modules and subsystems. The allocation of leaf functions to components can be seen in the IDEF0 diagrams of our FA (chapter 4) as well as the allocation table, Table 1, for reference.

No.	Function	Component	Description
111	Recognise undesirable behaviour	Irregularity Detection Model	A model will detect if a drone behaviour event is not a regular behaviour.
112	Highlight key external factors	Factor classification model	A model will classify external factors which led to undesirable behaviour (e.g., rain or strong wind).
113	Structure previous logs	Data event parser algorithm	An algorithm will parse raw data and structure into behavior events that are general for different drone flights.
121	Implement scenarios programmatically to avoid risks	Drone Software Developers	Developers will implement new scenarios into a drone's code that should help drone to avoid risks that were detected previously.
122	Provide changes to security scenarios	Security analysts	An analyst will apply changes to drone logistics and it's terms of use (e.g., maximal weight to carry).
123	Implement changes to drone architecture	Drone Engineers	Engineers will fortify or make other changes to drone architecture (e.g., make it lighter or add new sensors).
124	Evaluate compliance	Third Party Auditor	An auditor will verify whether governmental (or international) security standards are met or not.
131	Compose security protocol	Security Expert	An expert will collect risks and describe which measures should be taken for their avoidance.
132	Analyse current environmental conditions	Meteorological Data Source	Current weather conditions to detect possible risks from will be taken from a meteorological data source.
133	Predict future risks	Risk Prediction Model	A model will predict possible future risks according to current risks and weather conditions.
134	Highlight actual risks and their degree	Risk Assessment Tools	Tools will take possible risks, specify the ones that could be caused by current environmental conditions and measure degree of these risks according to level of this condition (e.g., wind/rain strength).
211	Enable attitude (ATTI) mode	Attitude Mode Software	Attitude mode software is used to enable a human to take control of a drone that malfunctions in-flight by disabling the autonomous flight software of and assigning full flight control to a human operator.
212	Initiate return to home (RTH) protocol	RTH Software	Return-to-home (RTH) software that autonomously navigates a malfunctioning drone that is in-flight to a safe location for retrieval.

213	Notify authorities	Emergency communication service	The system will try and inform the authorities of a lost drone in case all recovery protocols failed
214	Recover drone	Drone Pick-up Team	The team in charge of picking up a retrieved drone.
221	Analyze system error condition	Analysis Software	Software to automatically detect errors and anomalous activities in drone data logs.
222	Identify actual safety risks	Risk Indication Software	The system will make use of a risk identification software to process error conditions and also by taking into consideration historical factors with the purpose of identifying safety risks
223	Assess risk severity	Drone Operations Officer	The drone operations officer measures the severity of safety risks by evaluating data provided by the risk identification software.
231	Extract safety state	State Parsing Software	The software parses raw safety data and diagnostic information to determine the current safety state of the system.
232	Perform structural and timing checks	Equipment Specialist	The equipment specialist is in charge of performing checks to detect any deviations from the acceptable safety state of the system.
233	Collate fault data	Data Transformation Software	The systems ingests, and aggregates, error data and transforms it into meaningful metrics and logs for later use.
311	Filter and regulate data	Data Security Software	The software gathers various information data and merges to one clear text file
312	Detect sensor data and anomalies	Data Screening Software	The software detects data and checks for faults and error
313	Compile data	Security Data Manager	The manager manipulates the data and performs final compilation
321	Manipulate data	Data Safety Software	The software streamline the data processing and establishes safety protocols
322	Assess conformity	Quality Assurance Software	The software checks to see if it complies with predefined standards and regulations to achieve data of desired quality
323	Generate safety data	Safety Data Management Specialist	The specialist certifies his/her approval and generates final (safety) data
331	Inspect battery	Battery Inspection Software	The drone's battery will be inspected by a software and will indicate its condition and the time for the next recharging
332	Inspect structure	Structural Inspector	A structural inspector will inspect the drones to identify possible structural failures.
333	Evaluate condition	Condition Evaluator	A condition evaluator will determine if the drone is appropriate to be assigned an assignment.
341	Assign drone and flight number	Drone Selection Algorithm	Since the drone is evaluated as appropriate it may be assigned with a flight number for the next delivery.
342	Define journey	Journey Design Algorithm	An algorithm will define the journey that the assigned drone, with the assigned number will follow.
343	Simulate journey	Simulation Software Operator	An operator will simulate the journey for of the assigned drones and the defined journey.
344	Initiate journey	Journey Operator	The journey operator could initiate the delivery after having the simulation step processed.
411	Plan route	Route Planning Tool	A tool which finds the optimal route for each flight
412	Perform risk assessment to threats/safety issues	Risk Modelling Department	This department is in charge of modelling the influence of measured risks on flights

413	Assess drone charge planning	Battery Expert Team	Performs calculations regarding the charge needed for certain flight paths, includes alternatives paths
414	Communicate	Transport Communications and Operations Department	Provides communication between the transport department and the drone in case adaptations are needed and communicates with other parts of the system, makes final decisions on problems in the transport section
421	Clear area for landing	Downwards Directed Sensor	A sensor assessing whether the area underneath the drone is clear for landing
422	Indicate take-off/landing status	Electronic Light/-Sound System	An electronic system including lights and speakers to alert surroundings that the drone is landing
423	Clear surroundings for take-off	Gyroscope and Upwards Directed Sensor	A sensor assessing whether during take-off there are no objects above the drone
431	Provide physical drone structure	Drone Basis	A pre-manufactured drone including standard components (i.e. motors, propellers, standard sensors), this drone including the components specifically needed for this system is supplied
432	Control position/altitude	Control Software	Software to control the drone is made to order, to conform to the demands of the Transport department
433	Provide in-flight data	Data Gathering Sensors	Data gathering sensors are used to gather data, the type of sensors needed is determined by the data gathering department
434	Provide emergency protocol	Emergency Protocol Software	Specifically designed software downloaded on the drone with a protocol in case there is no communication with the operations and communications department
441	Add/remove food dish to/from compartment	Consumer	The consumer adds their food dish to the drone or when receiving removes the dish from the drone
442	Measure weight of entered items	Pressure Sensor	A pressure sensor is used to measure the weight of the items to make sure loading is done correctly and the weight does not exceed the limits
443	Open/close food compartment drone	Automatic Hinge Including Lock	The automatic hinge + lock enables the food compartment of the drone to be opened and closed by the drone itself

Table 1: The allocation table of components to the leaf function, including a description of the purpose and functioning of the component.

The components are further integrated into modules and subsystems that together form the complete physical system. This assembly of subsystems can be found in our SysRD.

6 GENERALIZABILITY OF THE SYSTEM

Although the system as we design it is tailor made for a single purpose, some components may be generalised in order to support a similar design. This is done by decomposing high level functions differently, reinterpreting items, assigning other components to leaf functions, or integrating the components differently. Doing so allows future design scientists to build upon the knowledge embedded in our design, by for example taking a high level decomposition for the design of a new system. In this chapter, we will list the various ways in which we expect that our system can be generalised.

Firstly, we note our focus on the geographical region of Mumbai in this design. In the design process, this allows us to focus down on Tiffin delivery and assume the Indian and Mumbai legislations. For the most part, the actual details of these do not matter for the decomposition of our FA. This means that the leaf functions are only affected by the legislation. Think about compliance to security standards such as a maximum velocity or altitude, as seen in Figure 24. Thus, we can generalise the legislation according to the region of deployment.

Secondly, the entire first level decomposition (functions 1-4) can be reused for automated delivery systems other than Drone based. In fact, except for function 4, even all second level decomposition's are generalizable in such a way. This allows competing systems to be developed according to the same standard of security and safety, provided they reused and improve upon our high level decomposition. Simulation and deployment of such a design will highlight the flaws of the high level design, yielding a more robust architecture for future implementations.

Lastly, the packaging and transport method of the food can be generalised. The system hinges on a context function that assumes recyclable, food-grade boxes to be available to transport food in. This, however, may change in the future or depending on the deployment context. However, the function 0 is independent of the packaging method used, and only requires the food to be loadable and unloadable. If, for instance, another method ought to be used which uses bio-degradables instead of recycling, the system does not need to be altered. These boxes can be safely disposed and degraded without an intermediate recycling step. The context changes, yet the system remains identical.

Not the entire system is generalisable, however. This is because our design was developed to serve a purpose, so naturally certain design decisions had to be made. These decisions are seen in, for example, the highly software driven design. Many functions such as those seen in Figure 27, Figure 42, and Figure 60 all rely heavily on software with a human in the loop. In our approach, using a more manual approach, on by contrast, a fully automated approach is difficult do realise with our design. A more general design would allow us to assign either software or a human as a component to each function.

7 CONCLUSIONS

Our design team was tasked by DoorDash to design a system to automate the Tiffin delivery boys often seen in Indian cities. We specifically focus on the Mumbai geographical area. Alongside other design teams, we opt for a drone based delivery. As a high concept, this design would use modern developments to replace human delivery boys with aerial quadcopters that carry the means from door to door. Our initial research showed that such a system has great benefits in terms of reduced delivery times, reduced congestion on public infrastructure, and an improved area of service. Our mission as a design team is thus to develop a fully functional food delivery system that is capable of reliably servicing the entire populace of Mumbai.

We firstly identify our major stakeholder groups, Customers, Consumers, Labour unions, Food delivery companies, and regulatory entities. These stakeholders provide our stakeholder requirements and system goals. We identify 4 high-level system goals, accessibility, reliability, punctuality, and availability. We additionally provide an operational concept in the form of 11 scenarios. We use these to communicate our interpretation of the requirements to our customers (of the design). With the requirements, system goals and the operation concept in hand, we then construct the system requirements that form the basis for our functional architecture.

To provide a framework for our functional architecture and to scope the system, we develop a model of the context. This allows us to define our assumptions and interpretations of requirements in the IDEF0 language, as well as highlight the systems that our system will interact with, and how. The context definition also provides us with input, output, and control requirements of our system. We revised the context twice, with the final version being shown in our SysRD.

To properly prioritise our functional architecture, we first define an elaborate objectives hierarchy. The objectives hierarchy denotes the importance of certain aspects for our system. The weights of the objectives hierarchy is fully derived from the stakeholder and system requirements. Again, we use these to communicate with our customer to verify our interpretation and open the dialogue before starting design. This step caused a few revisions of earlier steps, and gradually sharpened the system requirements.

After finalising a first version of the system requirements, we start designing the system according to specification. We make sure to offer ample room to revise designs, especially at the higher level. This gives our designers a chance to beat selection bias and approach the design from a different point of view. We show 3 alternative designs for some of the functions, and clearly elicit what we base our final selection on. Our design has a focus on preventing security and safety hazards. This stems from our stakeholders' firm requirement that malicious human activity should impact the system as little as possible.
When subdividing our functions, we often discovered I/O/Cs that we had not considered before. As an effect, other functions had to adapt of be fully revised to include the new found I/O/Cs. As well as that, sometimes a function gave an output we could not assign as input or control for other functions, nor the entire system. Such items were deemed irrelevant and triggered a redesign of the originating function. Together, many occurrences of these observations led to an often redesigned functional architecture.

We then proceeded to defining a physical architecture. We start by assigning components to the leaf functions. To determine which functions are leaves, we question whether we can assign a component – be that software, a human, or a mechanical device – to perform the function in deployment. If so, we stop decomposing and assign the component as a mechanism. These leaf components form the groundwork for our functional architecture. We then logically and hierarchically group there components in a tree structure. The root of this tree is our physical system. Our high-level subsystems contain departments and the consumer. This physical architecture is thus designed with an eye on deployment, in terms that are understandable to our stakeholders.

This completes the design of the system. A cross-section of the model allows us to trace the system requirements down to the physical components realising that requirement.

We see that our system offers some level of generalisability. A generalisable system would allow future designers to use and build upon the knowledge of automated Tiffin delivery presented in this document. In our design, especially the higher levels do not depend on the deployment setting of the system. Lower level functions, however, contain design decisions that makes is difficult to generalise to other solutions. Work needs to be done to improve the design, which is primarily done by more testing and solving issues found during deployment.

8 WORKLOAD DIVISION

Section	Contributor(s)
section 1.1	Dmitrii
subsection 1.1.1	Dmitrii
subsection 1.1.2	Athanasios, Jonah, Rahul & Zamir
subsection 1.1.3	Ayça & Andrei
subsection 1.3.1	Athanasios, Jonah, Rahul & Zamir
subsection 1.3.2	Iyanu & Robbin

Table 1: Workload distribution for Deliverable D1.

Section	Contributor(s)
chapter 2	Iyanu & Andrei
section 2.2	Jonah & Robbin
section 2.3	Ayça, Athanasios, Rahul & Zamir
chapter 3	Ayça, Athanasios, Rahul & Zamir
chapter 4	Jonah & Robbin

Table 2: Workload distribution for Deliverable D2.

Section	Contributor(s)
subsection 4.3.1	Ayça & Dmitrii
subsection 4.3.2	Iyanu & Andrei
subsection 4.3.3	Athanasios & Rahul
subsection 4.3.4	Jonah & Zamir
chapter 4 upgrade	Robbin
Integration, documentation, merging, planning	Robbin

Table 3: Workload distribution for Deliverable D3.

Section	Contributor(s)
Functional Architecture revision	All
Physical Architecture component integration	Jonah, Athanasios & Dmitrii
Requirements revision	Rahul, Zamir & Andrei
Presentation brainstorm	All
Presentation slides and contents developing	Iyanu, Ayça & Robbin
CORE merging, integration	Robbin
ORD content revision	Robbin

Table 4: Workload distribution for Deliverable D4.

BIBLIOGRAPHY

[1] wing.com. *Wing*. Sourced on: 29/04/2022. URL: https://wing.com/.

[2] dronedeliverycanada.com. *Drone Delivery Canada | Reimagining the way you deliver®*. Sourced on: 29/04/2022. URL: https://dronedeliverycanada.com/.

[3] Industry News. *How Much Weight Can a Delivery Drone Carry?* Sourced on: 29/04/2022. URL: https://www.dronedek.com/news/how-much-weight-can-a-delivery-drone-carry/.

[4] *Time for tiffin: the history of India's lunch in a box*. https://www.theguardian.com/lifeandstyle/2014/aug/17/tiffin-the-history-of-indias-lunch-in-a-box-mumbai.

[5] statista.com. *Number of internet users in Indian metro cities 2019*. Sourced on: 27/04/2022. URL: https://www.statista.com/statistics/1115168/india-number-of-internet-users-in-metro-cities.

[6] US Foods. "New study shows what consumers crave in a food delivery service". In: (). Sourced on: 29/04/2022. URL: https://www.usfoods.com/our-services/business-trends/2019-food-delivery-statistics.html.

[7] United Nations. "Stop Food Loss and waste, for the people, for the planet". In: *End Food Waste Day* (). Sourced on: 27/04/2022. URL: https://www.un.org/en/observances/end-food-waste-day.

[8] *Driver not to blame in Stint accident that killed four kids*. https://nltimes.nl/2019/07/04/driver-blame-stint-accident-killed-four-kids.