Assignment 2

Starstruck! Space Flight Management System

Brandon Lacey, c3368438
Brock Brinkworth, c3331952
Genis George, c3350081
Thomas Maguire, c3380879
Zackary Jeans, c3353113

Friday 0800-1000

Room ES209

Brandon, Brock, Genis, Thomas, Zackary.

Introduction

This report documents and details further follow-up from the previous report which extracted the requirements for StarStruck's online management system, and will expand further upon this system.

The following content contained within this report will delve further and continue the findings of the previous report, this involves further expanision to the Class Diagram, refinement of the Business Rules, mapping out the Sequence of events between objects within the system in correlation with time, Prototyping User Interfaces (UI) screens, discussing deployment strategies of the system, and finally, a review of the reports findings, and a recommendation of next steps for StarStruck.

This report is intended for audiences under employment of StarStruck, in addition to individuals, groups, and organisations who have an interest in beyond-earth travel. This report involves thorough documentation of complex systems and how objects within said system interact with each other and external entities, as such it is recommended to read the first report before delving into this one, as the first report provides the appropriate context and background for a complete understanding of the contents of this report.

The objective of this report is to further enable the executive of StarStruck to modernise their business through development of a theoretical online management system, built upon the foundation established by the previous reports findings. In addition, this report also aims to further enable the documentation of the design and development of space-faring systems such that the gap between what was thought mere fiction and reality is closed ever so slighty. As such, each section of the report has been carefully broken down such that each contains a subset of easily digestible information that builds on the section previous, in such a way to further the understanding of the reader while also advancing the content of the report towards it's objectives.

Business Rules

THE FOLLOWING BUSINESS RULES ARE A REQUIREMENT IN ALL CIRCUMSTANCES AND MUST BE FOLLOWED:

Scenario Rules

Business Rule	System Mapping	Organisational Mapping
Each flight has a default capacity limit of 7 people.	This functional requirement is enforced through the class diagram multiplicity relation between the passenger and booking classes.	
If a customer or group would like their trip to be exclusive, then there will be an additional fee.		If the customer wishes to book an exclusive flight, the manager will do it for them.
If the booking is a moon trip, then food will be required.	This functional requirement is enforced through the dependency relationship between the MoonTour class and the FoodOrder classes.	
If a customer books a flight, then they can purchase booking insurance.	This functional requirement is enforced through the class diagram multiplicity relation between the passenger and insurance classes.	
If the customer would like luxury food or drink for a non-moon trip, then there will be an additional fee.	This functional requirement is enforced through aggregation between Booking and Payment class	
If the food requested is not suitable for space flight, then it will be rejected.	This functional requirement is enforced through the dependency relationship between FoodOrderController and FoodOrder classes.	
If local weather is not satisfactory, then a flight will not be scheduled.	This functional requirement is enforced through the dependency relationship between the BusinessManagerTerminal and FlightBookingSafetyCheck.	
If a flight is not safe to be approved, it can be rescheduled, declined or cancelled.	This functional requirement is enforced through the dependency relationship between the BusinessManagerTerminal and FlightScheduleController.	
If mission control availability is not satisfactory, then a flight will not be scheduled.		This non-functional requirement is enforced through a member of the administration staff, or the business manager will do it for them.
If planetary bodies are in a certain location, then a flight will not be scheduled.	This functional requirement is enforced through checkFlightPathSafety() method of the FlightBookingSafetyCheck class in the class and sequence diagrams.	
If a problem is detected of a certain severity, then the launch may be aborted.	This functional requirement is enforced through the dependency between AstronautSpacecraftTerminal and SpaceCraftController.	
If a customer's flight is aborted and the customer didn't purchase		This non-functional requirement is enforced though a member of the admin staff / business manager, who

booking insurance, then the customer will not be reimbursed.		would handle reimbursements if a flight is aborted.
If a customer's flight is aborted and the customer purchased booking insurance, then the customer will be reimbursed either partially or fully, dependent on their level of insurance.		This non-functional requirement is enforced though a member of the admin staff / business manager, who would handle reimbursements if a flight is aborted.
If the flight abort system is used, then it will be recorded in an audit log.	This functional requirement is enforced through Space Flight Sequence diagram as an alternative fragment.	
If the flight abort system is used, then this will be recorded in the relevant flight log.	This functional requirement is enforced through Space Flight Sequence diagram as an alternative fragment.	
If a customer returns to earth, they will receive a medical check.	This functional requirement is enforced through the composition relationship between MedicalCheckReport and Passenger	
The details of the customers return airport are specified in the flight booking.	This functional requirement is enforced through the class diagram as the Booking class contains details of the customers return airport.	
If the override is used, then it recorded in the audit log.	This functional requirement is enforced through the Space Flight Sequence diagram as an alternative fragment.	
If the override is used, then it is recorded in the individual flight log.	This functional requirement is enforced through the dependency between the LogController and MissionControl classes.	
The amount of fuel in each stage of the rocket, the flight crew on board are strictly tracked on each flight.	This functional requirement is enforced through the Space Flight Sequence diagram and is reflected through the Space Flight User Interface Prototype.	
Everything and everyone entering the spacecraft (including fuel) must be weighed in order to determine the spacecrafts exact weight, which must be recorded by mission control.	This functional requirement is enforced though the Mission Control controller located in the class diagram, which calculates the weight, stores this data in the system, and transfer this data to mission control.	In terms of physically weighing entities for the flight, this will be enforced non-functionality via staff member, most likely security staff weighing each passenger and their items before the Flight. Additionally, other entities such as objects & liquids will be weighed as well.
If a client wishes to cancel the trip, there will be a cancellation fee.		This non-functional requirement is enforced though a member of the admin staff informing the customer that they will be charged a fee if they cancel their trip.
If a client tries to make a cash payment it will be refused.	This functional requirement is enforced though inheritance between types of payment and PaymentType, as cash is not an existing class in the class diagram, it cannot be accepted.	Non-functionally this requirement can also be enforced by a member of the admin staff who can directly refuse a cash payment if offered.
If a payment is not cleared, then the booking cannot be confirmed.	This functional requirement is enforced composition relationship between Payment and Booking classes. And additionally, is enforced though the Business Manager sequence diagram.	Non-functionally, this requirement is enforced though the business manager ensuring that a flight's payment has been cleared when reviewing said flight.
In cases of group bookings, a 10% discount can be given per person in the group.		This non-functional requirement is enforced through the business manager who will calculate this discount and minus it from each

		customer's total flight cost at the time of confirming the customer's booking.
Acceptable payment methods are Bank cheque, direct deposit, approved cryptocurrencies, and credit cards. Cash is strictly not permitted.	This functional requirement is enforced though the inheritance relationship between Payment and each PaymentType class, those being DirectDeposit, BankCheque, CreditCard, and CryptoCurrency. Other forms of payment are not allowed or accepted.	
If a flight is booked in advance, then the cost will be reduced the more leading time given. This will be calculated at the time of booking.		This non-functional requirement is enforced through the business manager who will calculate / take this calculation and minus it from the customer's total flight cost at the time of confirming the customer's booking.
Crew information such as qualifications and their expiration dates, training completed, and medical records are all tracked by the business for safety purposes.	This functional requirement is enforced through composition between StaffQualifications and Staff classes.	
If a crew member or customer are deemed medically unfit by the medical staff, then they will be refused entry onto the flight.	This functional requirement is enforced through dependency between MedicalCheckController and MedicalCheckReport.	
If a crew member or customer are deemed medically unfit by the medical staff, then it will be recorded in a log.	This functional requirement is enforced through dependency between MedicalCheckController and MedicalCheckReport.	
If a customer books a flight, then they will have a medical check before and after the flight.	This functional requirement is enforced through dependency between MedicalCheckController and MedicalCheckReport.	
If the medical staff detect a medical problem, then the customers GP will be notified, and this medical problem will be recorded in the booking.	Functionally, this requirement is enforced though the medical staff member updating the customer's medical record after the test. Which is enforced though the class diagram and medical check sequence diagram.	Non-functionally this requirement will be fulfilled and enforced by a member of the medical staff who will contact the customers GP if said customer fails their medical test.
If a customer is found to be breaking the rules of the trip, then their privileges on the flight will be restricted.		This non-functional requirement is enforced though crew members of the spacecraft ensuring each passenger follow the rules and are subsequently punished if they do not.

Work, Health and Safety Rules

Business Rule	System Mapping	Organisational Mapping
If a document that is required to be included in application is not written in English, then a translation of the documentation must be provided.		This non-functional requirement is enforced though a member of the admin staff reviewing all document, and ensuring each document is either in English, or is accompanied by a translation.
The application for grant of permit must be written and in English.		This non-functional requirement is enforced though a member of the admin staff who duty it is to ensure this application is in English.
In order to ensure a safe working environment, all equipment on the spacecraft must be thoroughly inspected, and must operate without defects, otherwise the flight cannot go ahead.		This non-functional requirement is enforced though frequent and reoccurring maintenance of each spacecraft before and after conducting a flight.
Before entering the spacecraft, all customers must be briefed on the risks of partaking on the flight.		This non-functional requirement is enforced though member/ members of the admin staff and checking the customer read the risk documents and ensuring that customers have accepted the terms and conditions.
Before being able to be a member of the flight, all crew must be thoroughly briefed on appropriate Work Health and Safety conduct.		This non-functional requirement is enforced though checking the whole crew read the appropriate Work Health and Safety conduct documents and ensuring that crew have accepted the terms and conditions.
Each member of the flight crew must be fully trained in operation of the spacecraft before being the flight can proceed.		This non-functional requirement is enforced though by checking if the crew is trained in operation of the spacecraft with relevant and appropriate training.
On board the spacecraft, appropriate medical equipment must be present, such as defibrillators, first aid kits and fire extinguishers. Additionally, all crew members must be trained appropriately in		This non-functional requirement is enforced though member/ members of the medical staff. The admission staff checks that the whole crew has relevant and appropriate training to operate medical equipment.

order to operate said medical equipment.		
On board the spacecraft, there must be enough oxygen bags for both the customers and crew in cases of emergences, or failure in on-board systems.		This non-functional requirement is enforced though member/ members of the crew to double check before the flight travel and replace damaged oxygen bags
Before members of the crew or customers step foot onto the spacecraft, the spacecraft must be thoroughly inspected for safety risks. All risks need to be identified, such as that these risks can either be eliminated or minimised as much as possible.		This non-functional requirement is enforced though member/ members of the crew by following risk management documentation in order to avoid/minimise the risks.
Failure to comply to appropriate Work, Health and Safety regulations by customers or crew may result in said individual being barred from entry to the spacecraft. Additionally, if the violation is severe, the customer may be banned from further flights, or the crewmember may be terminated.		This non-functional requirement is enforced though member/ members of the crew to monitor the customers as well as their co-workers to see if they fail to comply to appropriate Work, Health and Safety regulations. Camera footage may be used to view the incident.

Ethical, Privacy and Security Rules

Business Rule	System Mapping	Organisational Mapping
A customer's information such as payment, medical and personal information must be kept confidential and stored appropriately and securely.		This non-functional requirement is enforced though encryption of all document's which are digitally stored. Of those documents which are physically stored, they must be locked in a secure location in which access to restricted to the business manager or senior admin staff.
All crew members private information such as medical records must be securely, and safety stored.		This non-functional requirement is enforced though encryption of all document's which are digitally stored. Of those documents which are physically stored, they must be locked in a secure location in which access to restricted to the business manager or senior admin staff.
All crew members private information such as medical records must be kept private.		This non-functional requirement is enforced though encryption of all document's which are digitally stored. Of those documents which are physically stored, they must be locked in a secure location in which access to restricted to the business manager or senior admin staff.
If a customer books a flight, then they must agree to the flight terms and conditions agreement before being granted entry onto the spacecraft.		This requirement is enforced through a member / member of the admin staff double-checking and ensuring that customers have accepted the terms and conditions before allowing the security staff to let said customer onto their flight.
A security check must be performed on all passengers (crew and customers) prior to the launch date, any illegal items must be confiscated.		This non-functional requirement is enforced through mandatory searches of each passenger before entering the spacecraft.
If any passengers (crew and customers) are caught with illegal contraband, they will be barred from flying.		This non-functional requirement is enforced through mandatory searches of each passenger before entering the spacecraft, in which the passenger will then be added to a no-fly list by a member of the admin staff.
The on-board computer system must be thoroughly tested for bugs and		This non-functional requirement is enforced though the scheduled testing of each

vulnerabilities, and subsequently patched before a flight can be commenced.		spacecraft before and after launch, this also includes testing the software utilised in each spacecraft.
All systems of the spacecraft must be thoroughly tested to ensure full operational capacity before a flight can be commenced.		This non-functional requirement is enforced though the scheduled testing of each spacecraft before and after launch, this also includes testing the software utilised in each spacecraft.
Sensitive systems and data may only be accessed by staff authorized to access said systems and data.	This functional requirement is enforced through role-based-authentication via Auth tokens, wherein each staff member must login and be granted an Auth token.	
Staff members may not disclose proprietary information regarding the business and its functions to any outside entity.		This non-functional requirement is enforced through StarStruck's employment contracts which each and all employees sign upon accepting an offer of employement at StarStruck. Violation of which will be enforced through legal action.

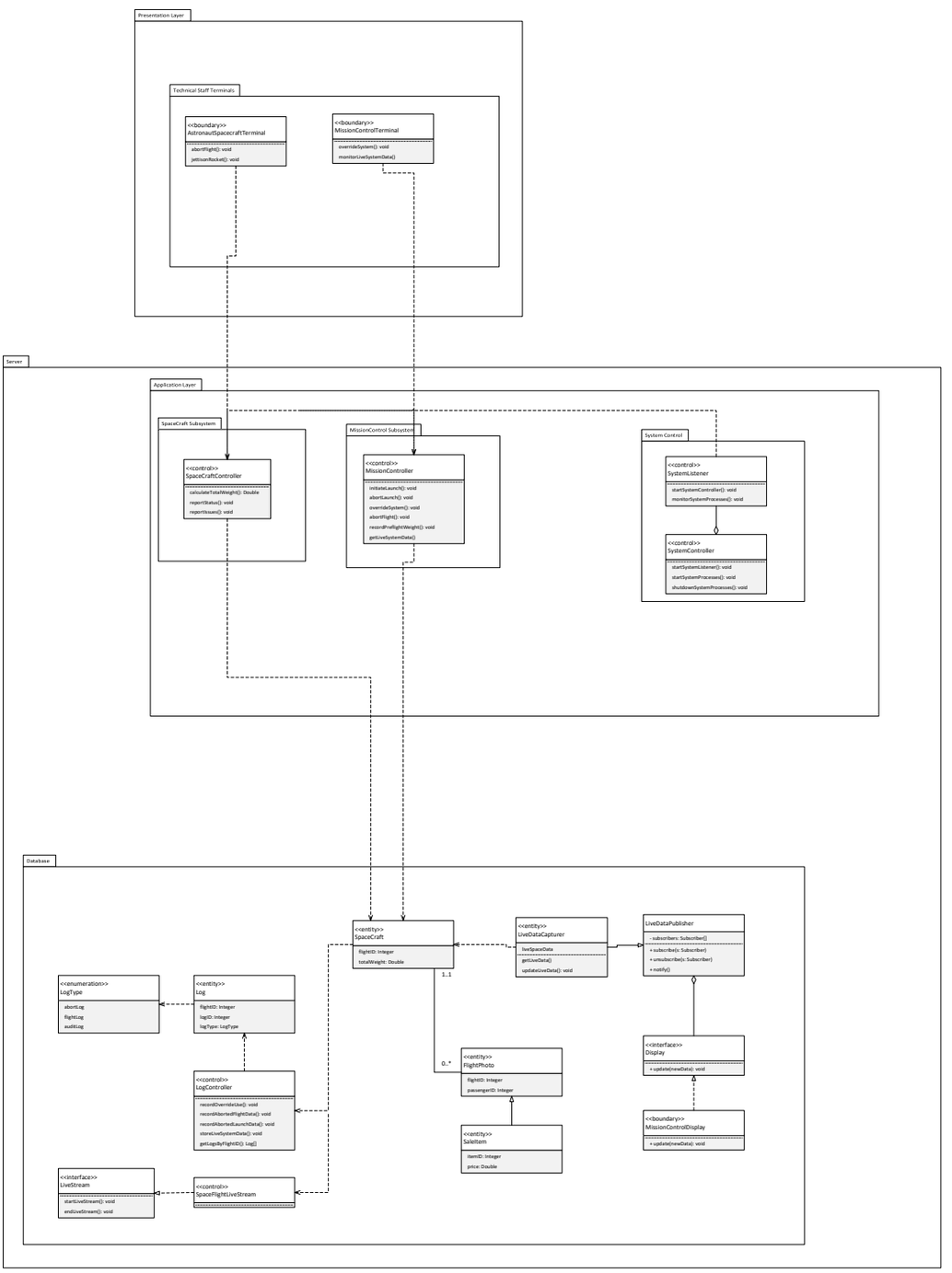
Evidence of Research Rules

Business Rule	System Mapping	Organisational Mapping
Application of license of Launches and returns must be granted from authorities before flights are permitted.		This non-functional requirement is enforced by the Business manager who as part of their duty must ensure the appropriate licenses have been provided to StarStruck or it's appropriate employee's by the necessary government entities.
The following activities need approval from Australian government or authorities: • Launch of a flying object from Australia • Return of a flying object from Australia • Launch of a flying object overseas • Return of a flying object overseas • Operating a launch facility in Australia.		This non-functional requirement is enforced by the Business manager who as part of their duty must ensure that StarStruck has permission from the required government bodies to conduct it's space tourism business, including the launch and return of spacecraft.
The holder of the license must not allow the launch facility to be used for unpermitted launches on the premises.		This non-functional requirement is enforced through ensuring that each and every launch of a spacecraft has gone through the proper procedure.
If requested the license holder must give a launch safety officer access to their launch facilities to assess the premises.		This non-functional requirement is enforced through the business manager whose duty also partakes to ensuring StarStruck follows all legal requirements and requests.
Plans and records of the launch facility must be updated and able to be given to authorities, these plans and records must include: • Launch facility management plan • Launch facility environmental plan • Launch facility design and engineering plan for the whole facility • Launch facility emergency and evacuation plan		This non-functional requirement is enforced by the Business manager and senior admin staff who as part of their duty, are required to collect, maintain, store, and provide government entities this information as required by law.

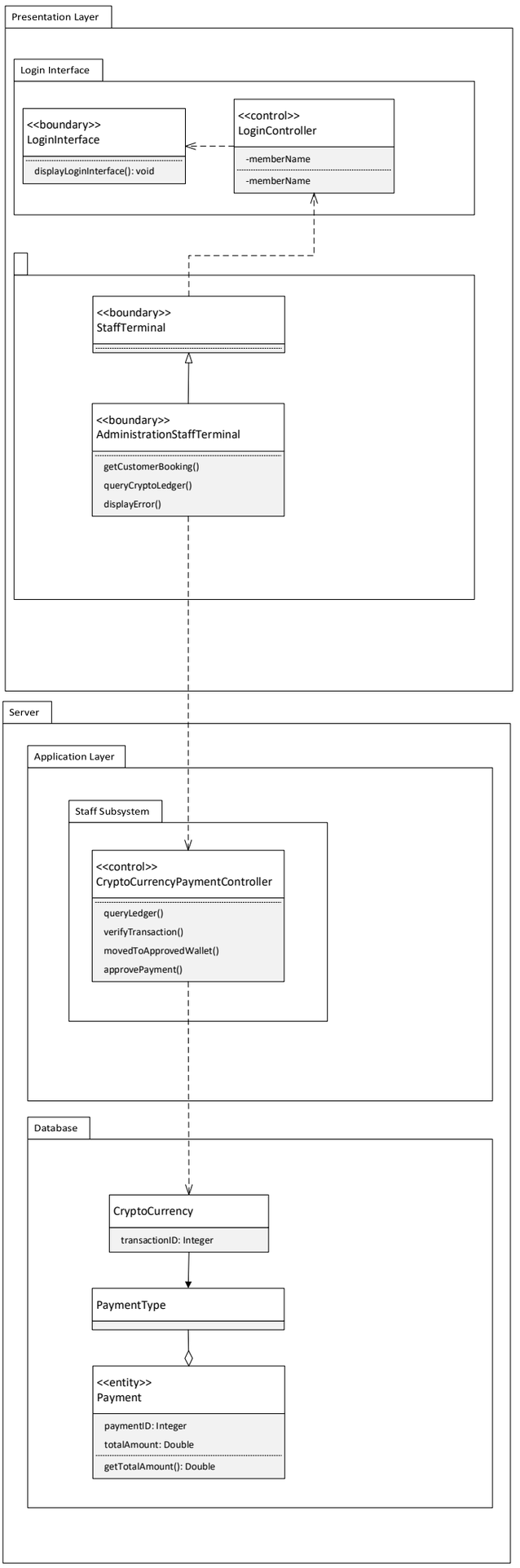
• Launch facility technology and security plan		
The holder of the license must notify Authorities before changes to the following: • The Organizational structure • The identity of individuals with responsibilities and roles in management		This non-functional requirement is enforced ofthe Business manager and executive of StarStruck who as part of their duty, are required to inform selected government departments of any of these changes.
In case of accident, authorities must be notified and included in investigation to find responsible entity.		This non-functional requirement will be fulfilled by the Business manager, who takes responsibility for ensuring the safety of the flight path, as well as each space craft before and after flight.
If a vehicle does not meet launch safety requirements it may be restricted from flying in the vicinity of significantly populated areas.		This non-functional requirement is enforced through frequent and reoccurring maintenance and testing of each space craft before and after their flight.
A designated area for controlled impact of return must be allocated to returning flights and must have a grace period between returns.	This functional requirement is enforced through the class diagram as each Booking contains the landing location designated on the flight's return. Additionally, the Business manager ensures all return locations have the correct grace period between flights as calculated through the FlightBookingSafetyCheck class in the class diagram.	
Risks and hazard analysis methodology must be able to be accessed by authorities to analyses hazards frequently.		Any and risk and hazard analyses are the responsibility of the business manager, who will fulfill this duty as required by law.

Class Diagram & Subsystems

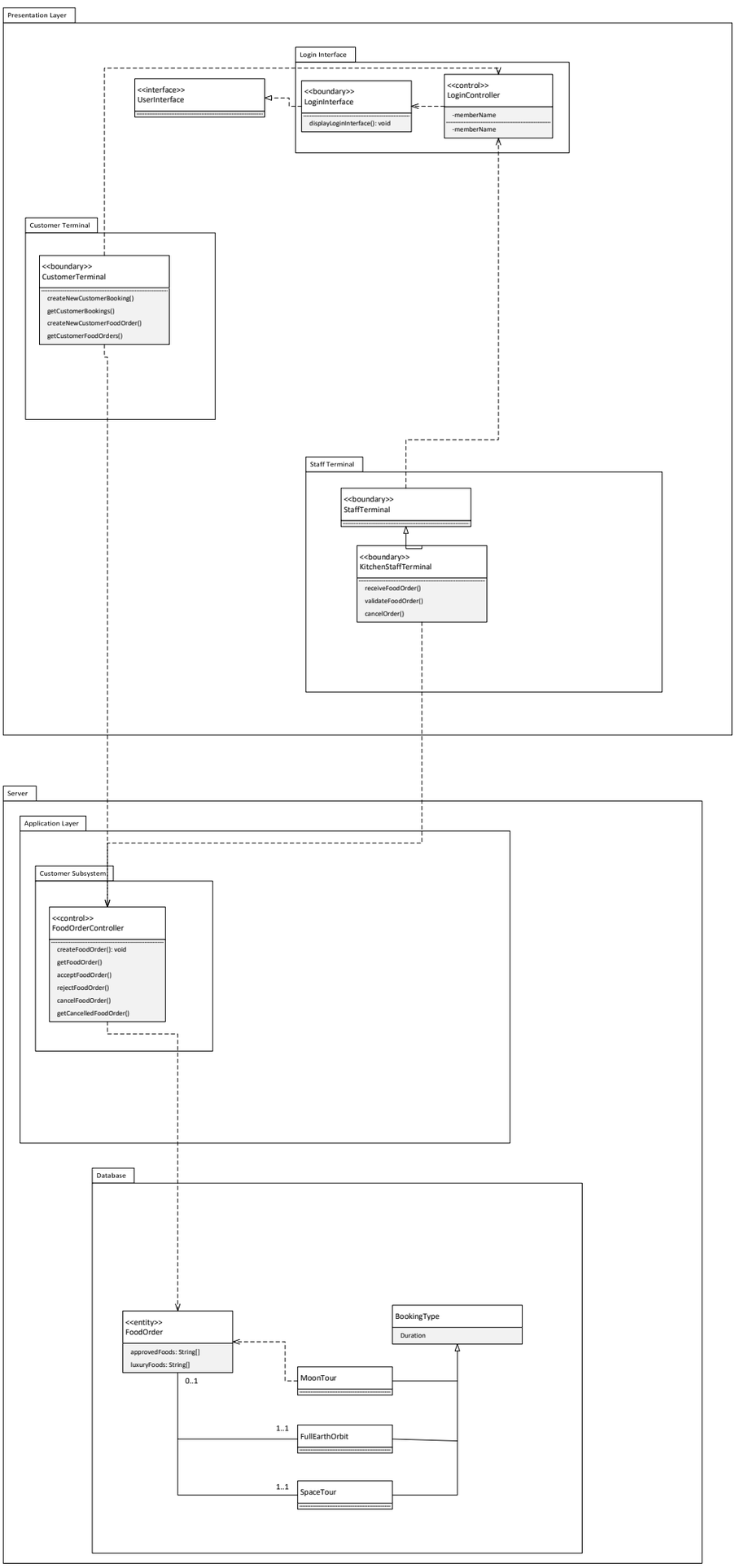
SPACE FLIGHT SUBSYSTEM



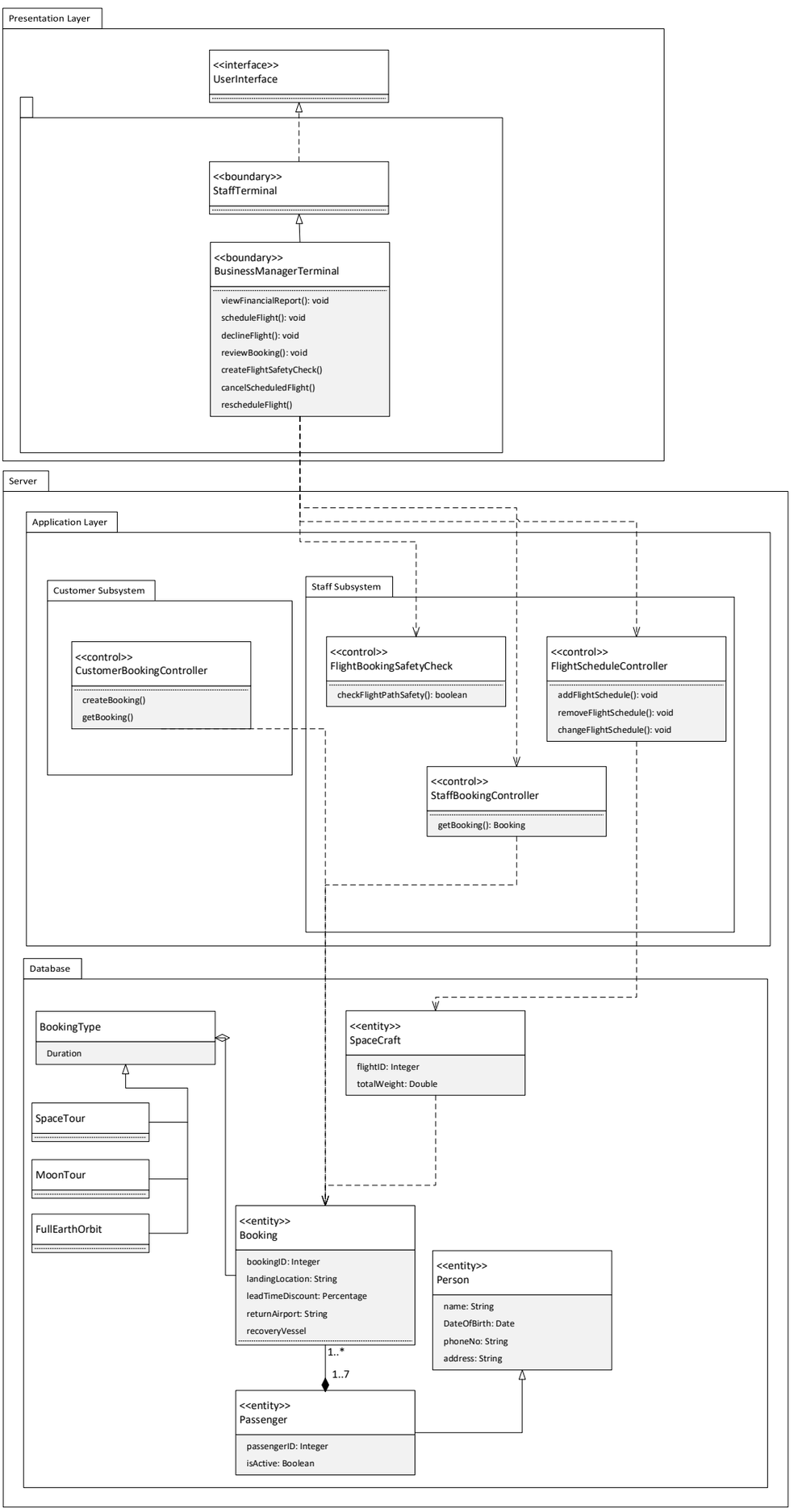
CRYPTO PAYMENT SUBSYSTEM



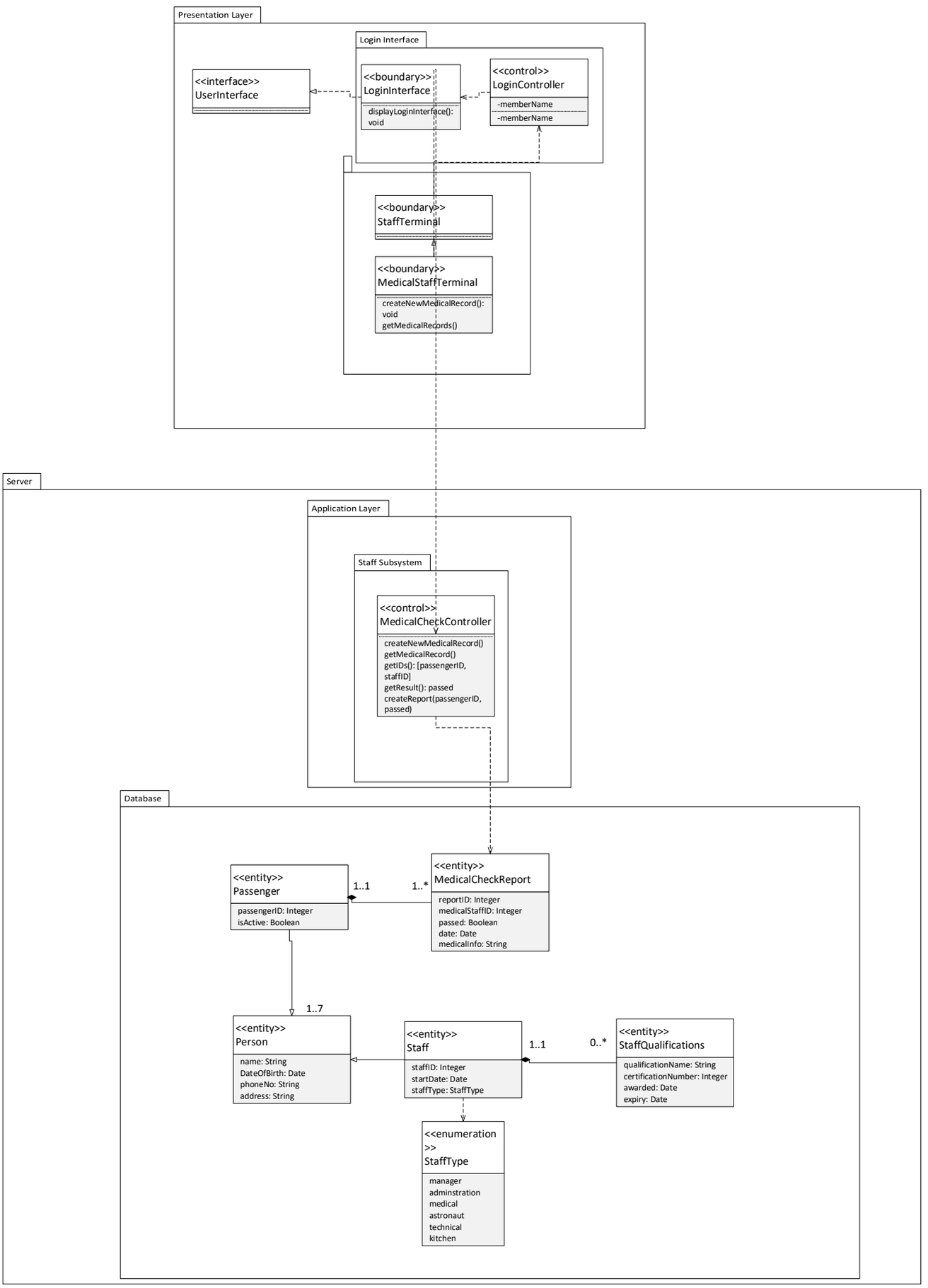
FOOD ORDER SUBSYSTEM



FLIGHT BOOKING SUBSYSTEM



MEDICAL CHECK SUBSYSTEM



Presentation Layer

Login Interface

<<interface>>
IUserInterface

<<boundary>>
LoginInterface
- displayLoginInterface(): void

<<control>>
LoginController
- memberName
- memberName

Customer Terminal

<<boundary>>
CustomerTerminal
- createNewCustomerBooking()
- getCustomerBooking()
- createNewCustomerFoodOrder()
- getCustomerFoodOrders()

<<boundary>>
AstronautSpacecraftTerminal
- abortFlight(): void
- jettisonRocket(): void

<<boundary>>
MissionControlTerminal
- overrideSystem(): void
- monitorLiveSystemData()

<<boundary>>
StaffTerminal

<<boundary>>
BusinessManagerTerminal
- viewFinancialReport(): void
- scheduleFlight(): void
- declineFlight(): void
- reviewBooking(): void
- createFlightSafetyCheck()
- cancelScheduledFlight()
- rescheduleFlight()

<<boundary>>
MedicalStaffTerminal
- createNewMedicalRecord(): void
- getMedicalRecords()

<<boundary>>
AdministrationStaffTerminal
- getCustomerBooking()
- queryCryptoLedger()
- displayError()

<<boundary>>
KitchenStaffTerminal
- receiveFoodOrder()
- validateFoodOrder()
- cancelOrder()

Server

Application Layer

Customer Subsystem

<<control>>
FoodOrderController
- createFoodOrder(): void
- getFoodOrder()
- acceptFoodOrder()
- rejectFoodOrder()
- cancelFoodOrder()
- getCancelledFoodOrder()

<<control>>
CustomerBookingController
- createBooking()
- getBooking()

SpaceCraft Subsystem

<<control>>
SpaceCraftController
- calculateTotalWeight(): Double
- reportStatus(): void
- rejectIssues(): void

MissionControl Subsystem

<<control>>
MissionController
- initiateLaunch(): void
- abortLaunch(): void
- overrideSystem(): void
- abortFlight(): void
- resendFlightWeight(): void
- getLiveSystemData()

Staff Subsystem

<<control>>
FlightBookingSafetyCheck
- checkFlightPathSafety(): boolean

<<control>>
FlightScheduleController
- addFlightSchedule(): void
- removeFlightSchedule(): void
- changeFlightSchedule(): void

<<control>>
StaffBookingController
- getBooking(): Booking

<<control>>
FinancialReportController
- getFinancialReport(): FinancialReport

<<control>>
MedicalCheckController
- createNewMedicalRecord()
- getMedicalRecord()
- getPassengerID(): passengerID
- getResult(): passed
- createReport(passengerID, passed)

<<control>>
CryptoCurrencyPaymentController
- queryLedger()
- verifyTransaction()
- moveETAAgreementWallet()
- approvePayment()

Authentication

<<controller>>
AuthProvider
- generateAuthToken(): AuthToken

<<entity>>
AuthToken
- header
- payload
- signature

System Control

<<control>>
SystemListener
- startSystemController(): void
- monitorSystemProcesses(): void

<<control>>
SystemController
- startSystemListener(): void
- startSystemProcesses(): void
- shutdownSystemProcesses(): void

Database

<<entity>>
SpaceCraft
- flightID: Integer
- totalWeight: Double

<<entity>>
LiveDataCapturer
- liveSpaceData
- getLiveData()
- updateLiveData(): void

LiveDataPublisher
- subscribers: Subscriber[]
+ subscribe(s: Subscriber)
+ unsubscribe(s: Subscriber)
+ notify()

<<interface>>
Display
+ update(newData): void

<<boundary>>
MissionControlDisplay
+ update(newData): void

<<enumeration>>
LogType
- alertLog
- flightLog
- auditLog

<<entity>>
Log
- flightID: Integer
- logID: Integer
- logType: LogType

<<control>>
LogController
- recordOverrideLog(): void
- recordAbortedFlightData(): void
- recordAbortedLaunchData(): void
- storeLiveSystemData(): void
- getLogsByFlightID(): Log[]

<<entity>>
FlightPhoto
- flightID: Integer
- passengerID: Integer

<<entity>>
SaleItem
- itemID: Integer
- price: Double

<<interface>>
LiveStream
- startLiveStream(): void
- endLiveStream(): void

<<control>>
SpaceFlightLiveStream

<<entity>>
FoodOrder
- approvedFoods: String[]
- luxuryFoods: String[]

BookingType
- Duration

<<entity>>
Booking
- bookingID: Integer
- landingLocation: String
- leafTimeDiscount: Percentage
- returnAirport: String
- recoveryVessel

<<entity>>
Payment
- paymentID: Integer
- totalAmount: Double
- getTotalAmount(): Double

<<entity>>
FinancialReport
- reportID: Integer
- date: Date
- profitStatement: String
- cashflowStatement: String

MoonTour

FullEarthOrbit

SpaceTour

PaymentType

<<entity>>
Passenger
- passengerID: Integer
- isActive: Boolean

<<entity>>
MedicalCheckReport
- reportID: Integer
- medicalStaffID: Integer
- passed: Boolean
- date: Date
- medicalInfo: String

DirectDeposit
- accountID: Integer
- bankBSB: String

BankCheque
- chequeName: String
- bankBSB: String

CreditCard
- cardNumber: String
- cardType: String
- expiryDate: Date

CryptoCurrency
- transactionID: Integer

<<enumeration>>
InsuranceType
- nowDate
- partialRefund
- fullRefund

<<entity>>
Insurance
- policyID: Integer
- flightID: Integer
- type: InsuranceType

<<enumeration>>
CargoType
- passenger
- crew
- fuel
- food
- drink
- consumable
- non-consumable

<<entity>>
CargoRecord
- recordID: Integer
- date: Date
- type: CargoType

<<entity>>
Person
- name: String
- DateOfBirth: Date
- phoneNo: String
- address: String

<<entity>>
Staff
- staffID: Integer
- startDate: Date
- staffType: StaffType

<<entity>>
StaffQualifications
- qualificationName: String
- certificationNumber: Integer
- awarded: Date
- expiry: Date

<<enumeration>>
StaffType
- manager
- administration
- medical
- astronaut
- technical
- kitchen

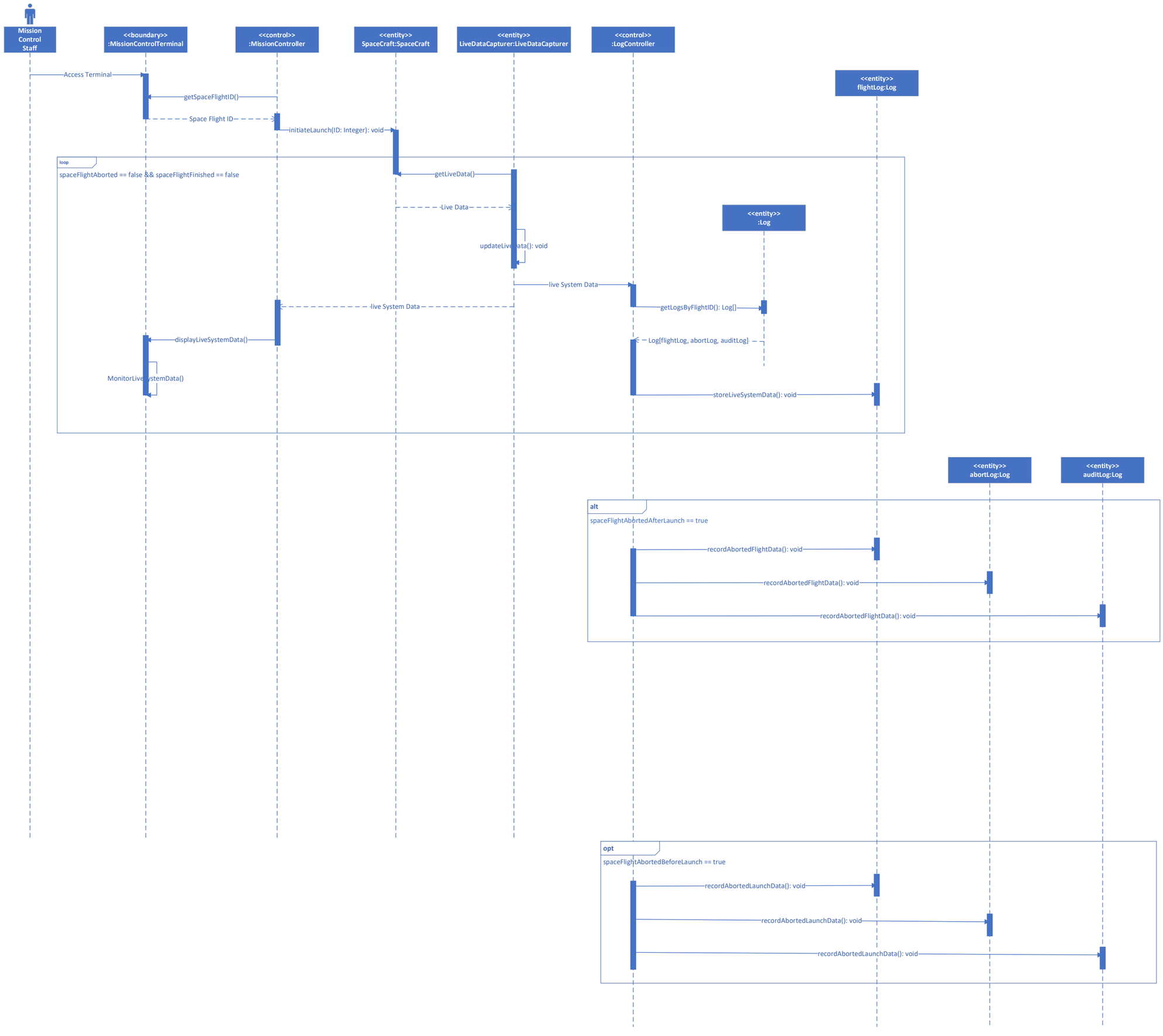
THIS PAGE IS INTENTIALLY LEFT BLANK

Sequence Diagrams

The following section of the report contains the Sequence diagrams of the Space Flight, Medical check, Crypto Payment, Booking, and Food Order subsystem, in relation to the class diagram and use case description (previous report).

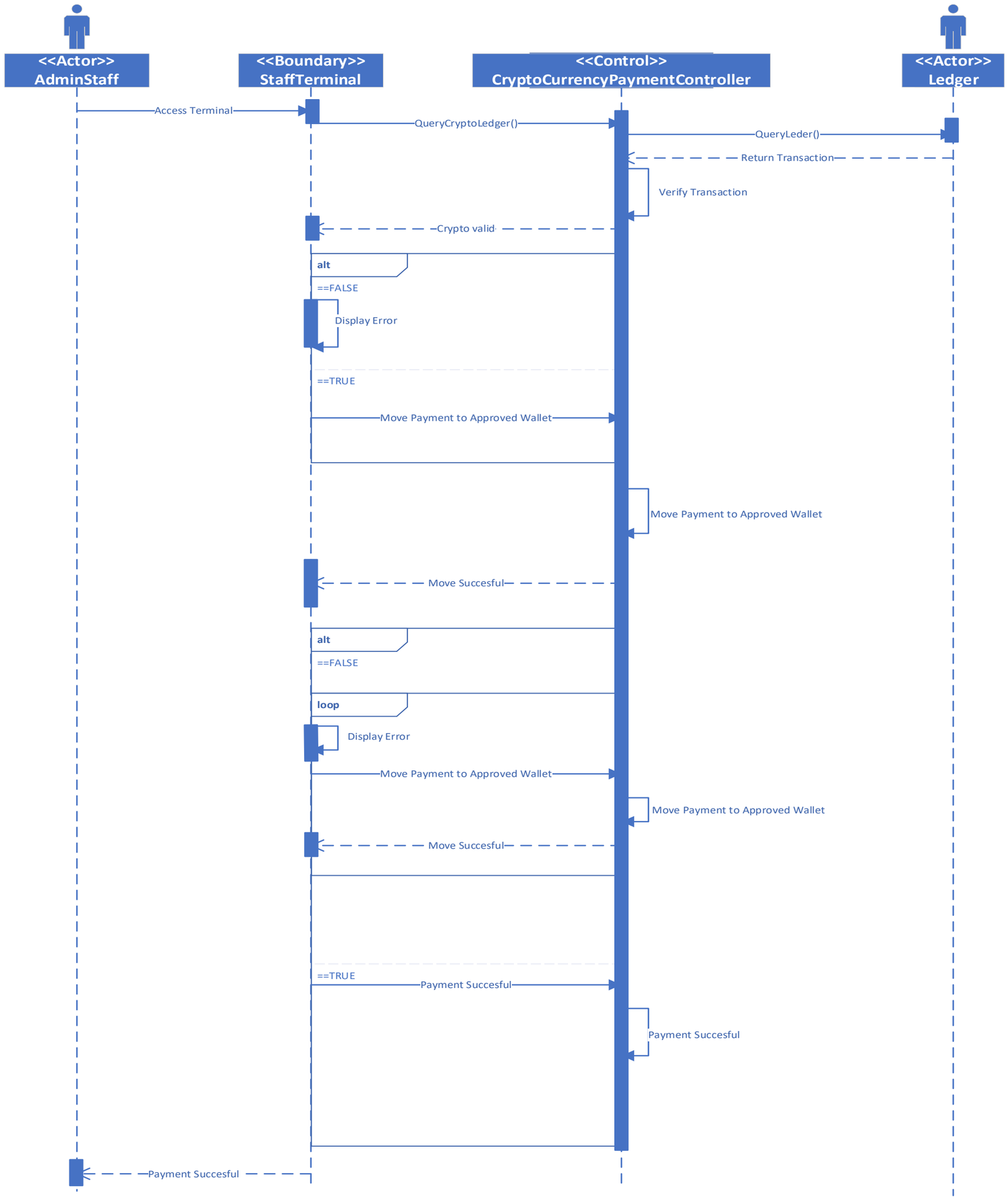
SPACE FLIGHT SEQUENCE DIAGRAM

The following sequence diagram follows mission control and the process of launching and monitoring a space flight. This includes a loop for continuously monitoring the live data which is referred to, as well as updating the logs in real time to respond to any incidents and if the flight has been aborted or not.



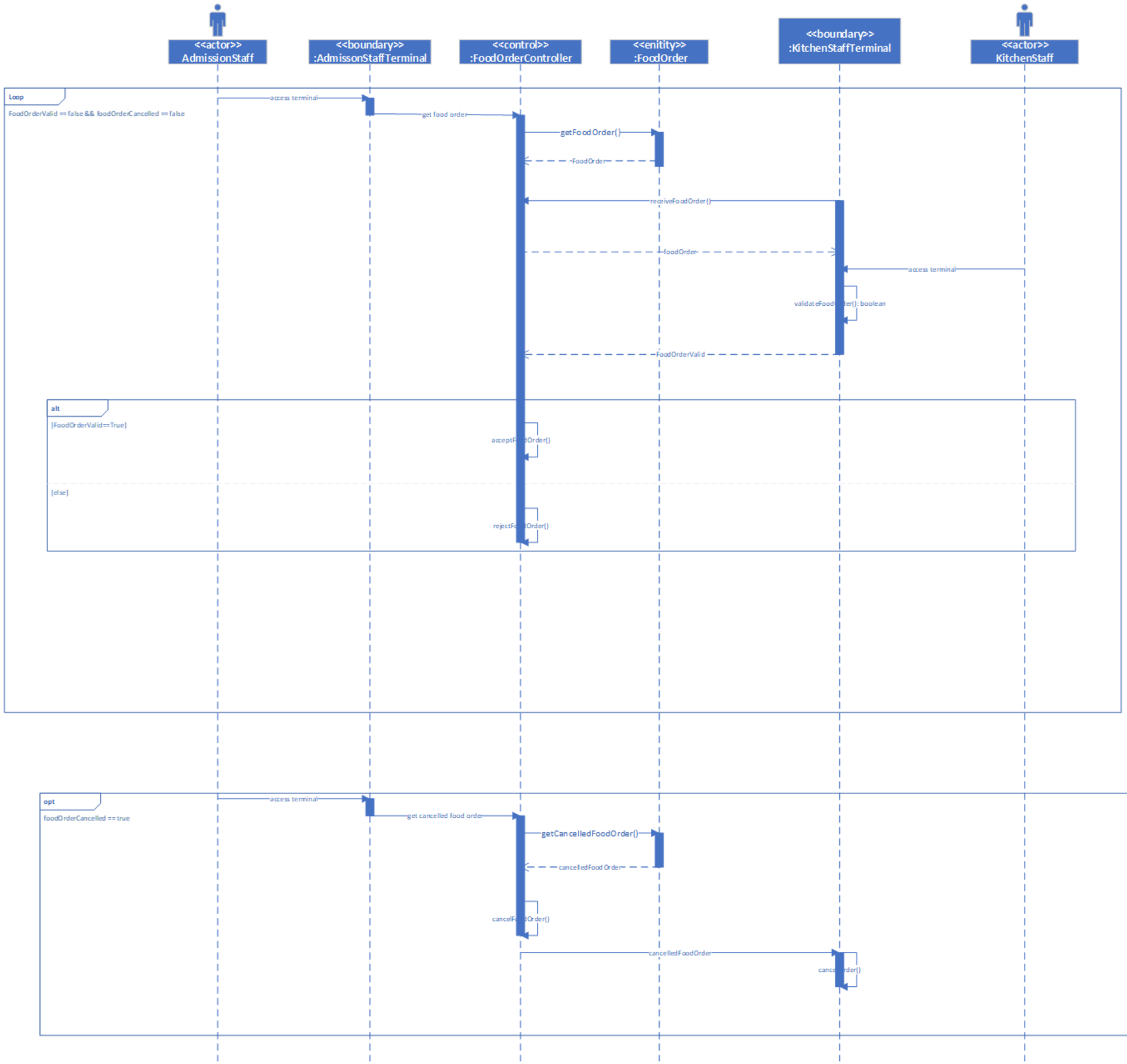
CRYPTO PAYMENT SEQUENCE DIAGRAM

The following sequence diagrams follows the process for a member of the administration staff confirming a flight booking payment, in which the payment type is of Cryptocurrency. First the staff member verifies that the transaction is valid (correct amount and in the correct type of currency e.g no dogecoin), then after this has been confirmed the staff member will move the currency into another secure wallet, which is designated for storing confirmed payments. After this the staff member will approve the whole payment process.



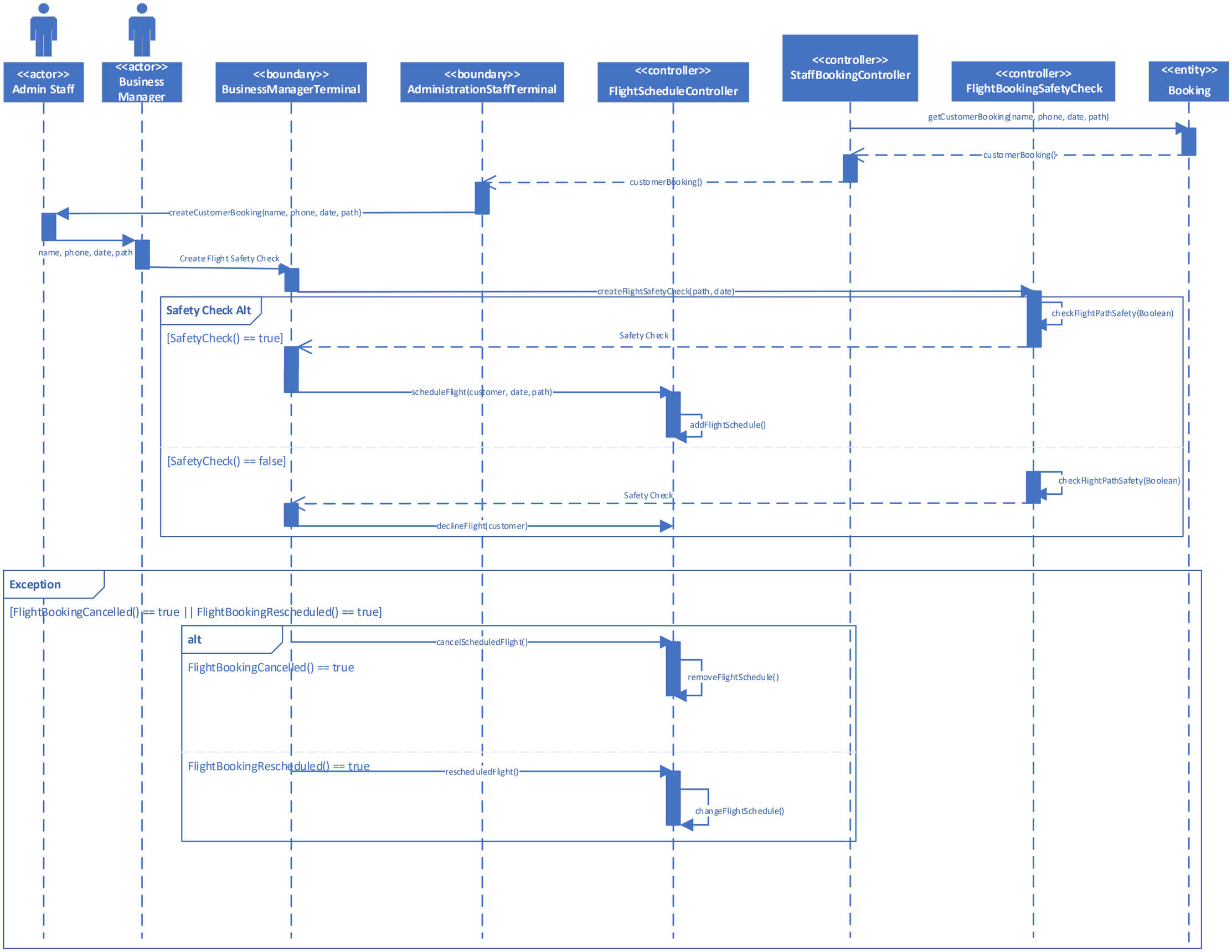
FOOD ORDER SEQUENCE DIAGRAM

The food order sequence diagram involves the administration staff receiving, and then sending a customer created food order to the kitchen staff, who either approve or deny said order. The diagram will continue to loop until the food order has been approved.



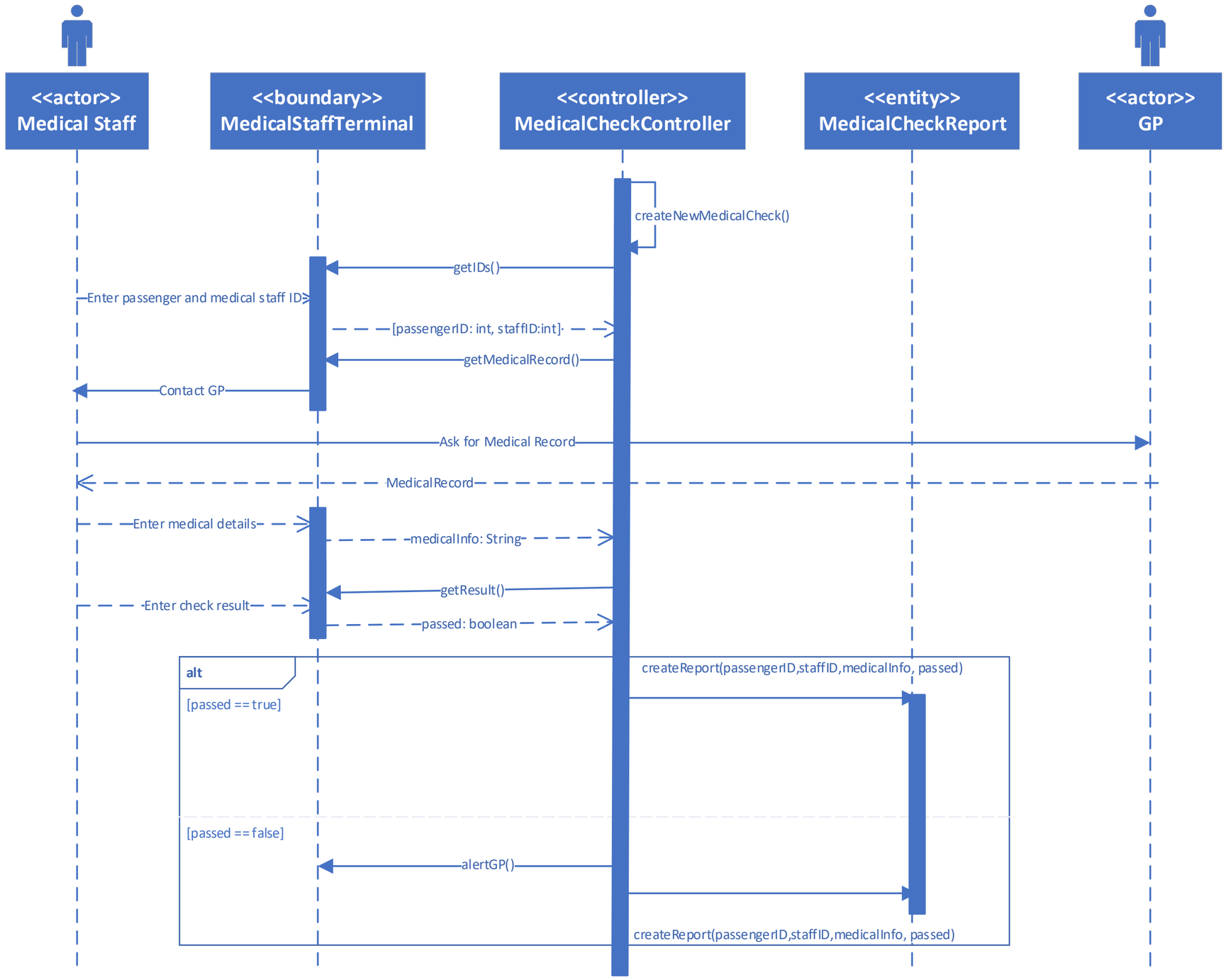
FLIGHT BOOKING SEQUENCE DIAGRAM

The following sequence diagram involves the business manager receiving a customer booking from a member of the administration staff. The business manager, through interaction via the boundary, will check if the booking is allowed to proceed (if the path is safe on a specific date). Also featured are alterative fragments for if a flight is cancelled and/or rescheduled.



MEDICAL CHECK SEQUENCE DIAGRAM

The below sequence diagram follows the sequence of a medical staff member entering the details of a customer's medical check. Options are present if the medical staff member needs to contact the passenger's GP if the passenger has failed the medical check. Ultimately, a medical report is created for the customer and stored in the system.

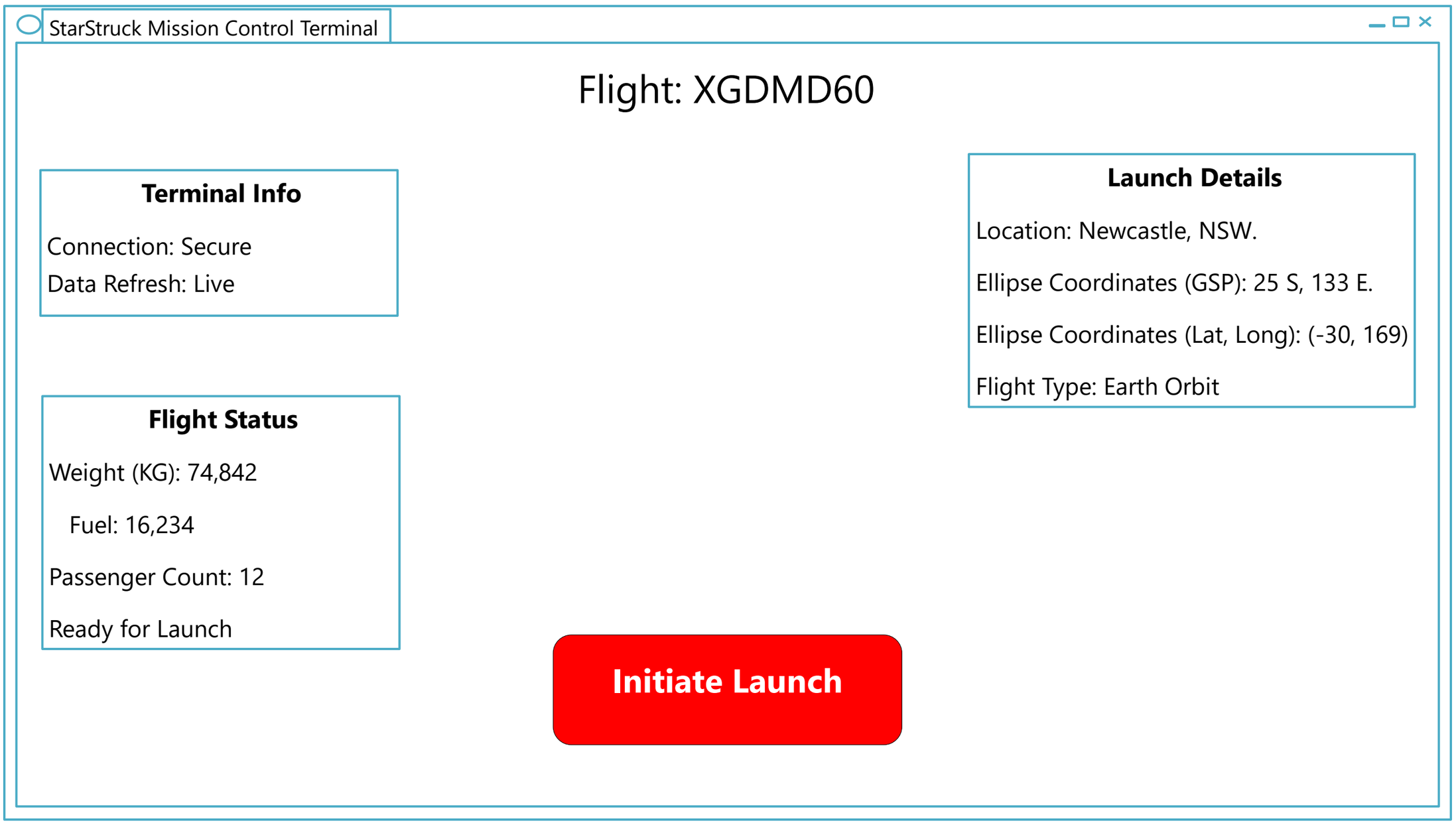


User Interfaces (UI)

SPACE FLIGHT USER INTERFACE

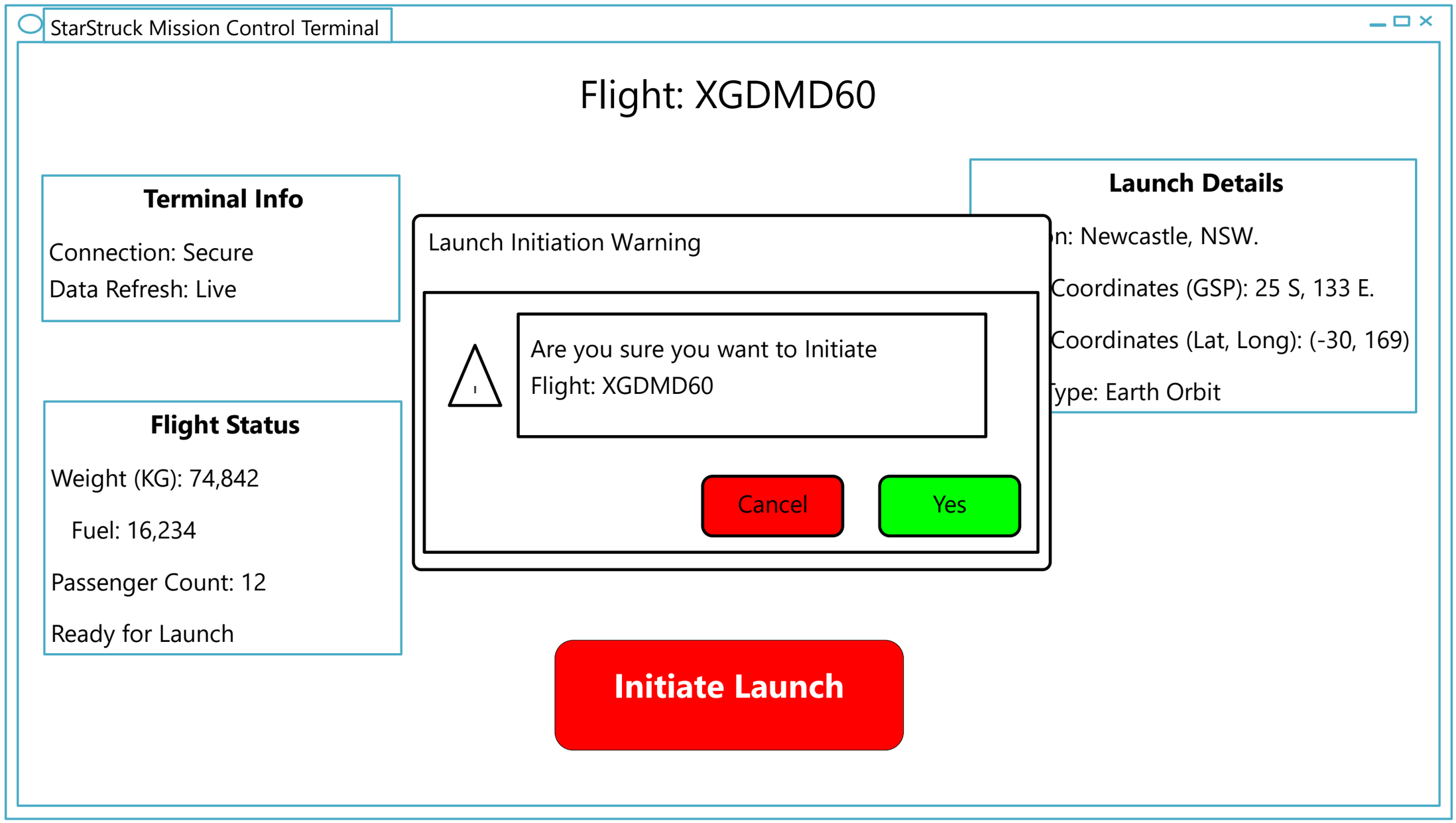
Screen #1 Description:

The following User Interface screen is the first initial screen which the staff in mission control interface with. The data is pulled from the server and continuously update, as indicated via the Data Refresh field of the Terminal Info Box. The screen contains the ID of the current Flight selected, as well as the flight's launch details, including the flight's weight. The most pressing feature of the screen is the 'big red button' – which enables the staff in mission control to begin the countdown process to launch the selected spacecraft.



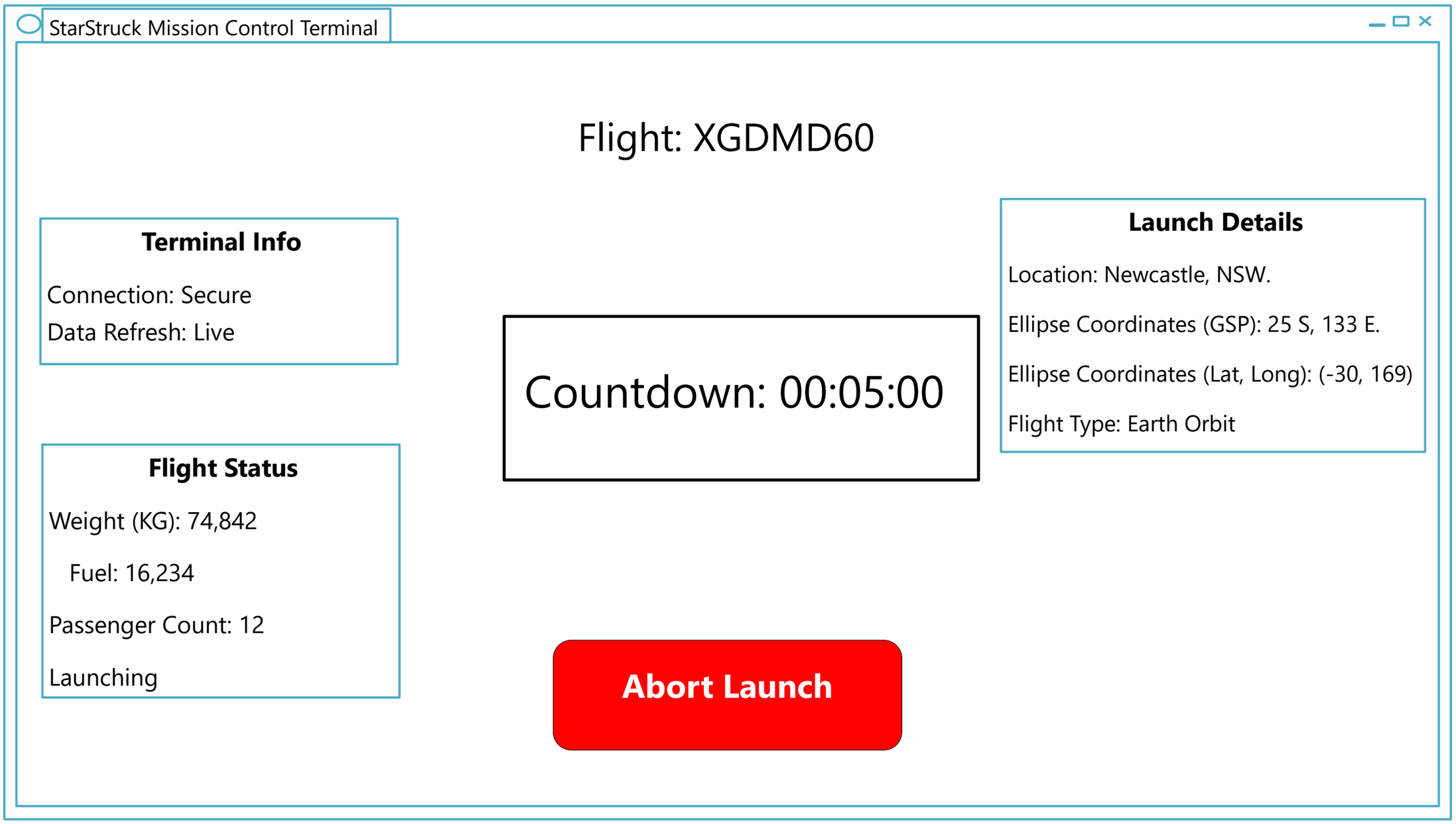
Screen #2 Description:

The following User Interface screen is the second screen which the mission control team view, and is navigated too via clicking the ‘Initiate Launch’ button, where a dialog box will pop up, and ask the mission control staff member to confirm their choice, or cancel their action.



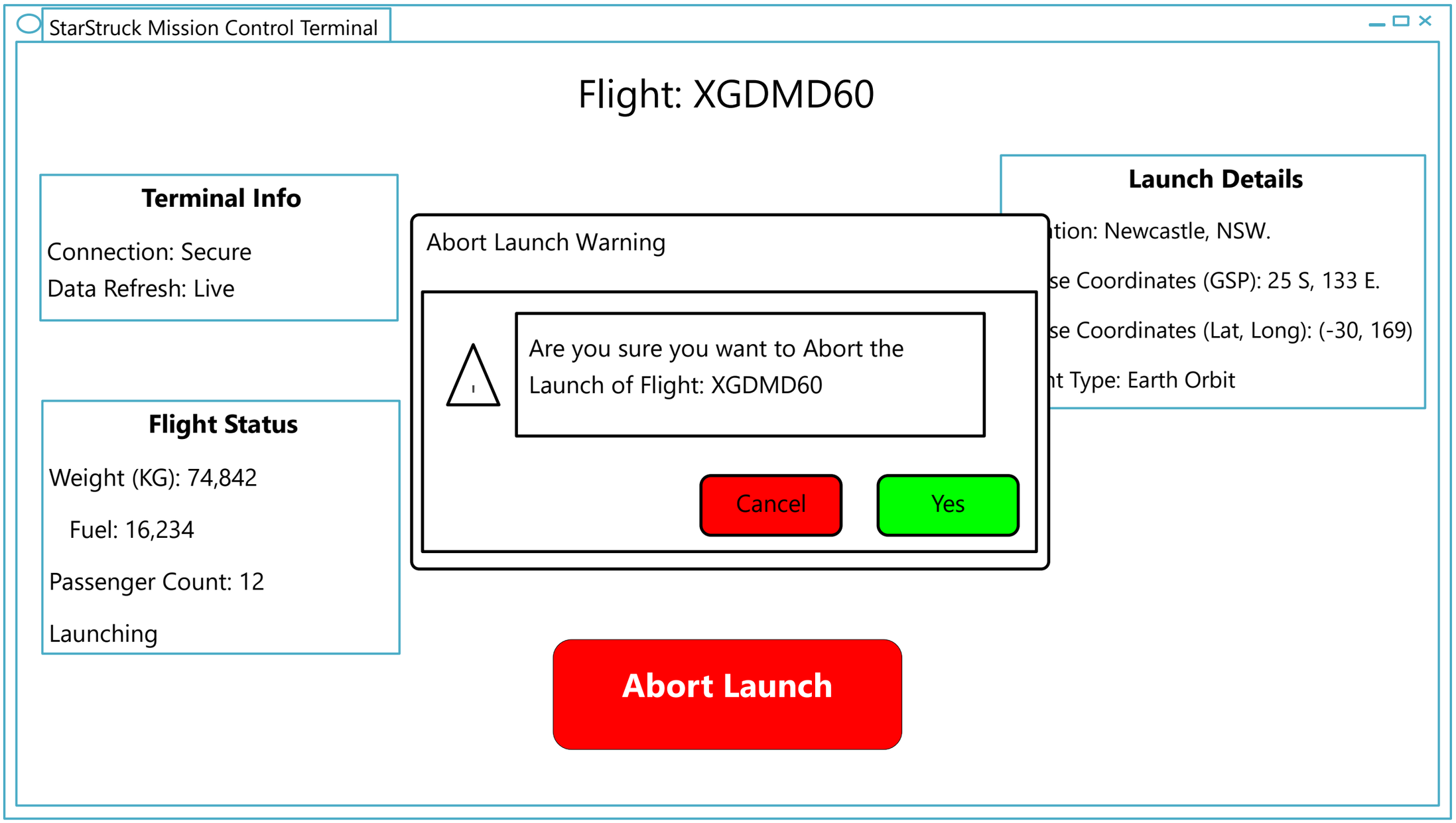
Screen #3 Description:

The following User Interface screen is the third screen, in which the countdown process has officially begin. This is reflected via the launching status in the flight status box, as well as the countdown box in the middle of the screen, in which a countdown to 5 minutes is in progress. This screen is reached via mission control confirming that they want to initiate the launch process. The most important feature of this screen is the Abort Launch button, which enables mission control to abort the launch of the spacecraft.



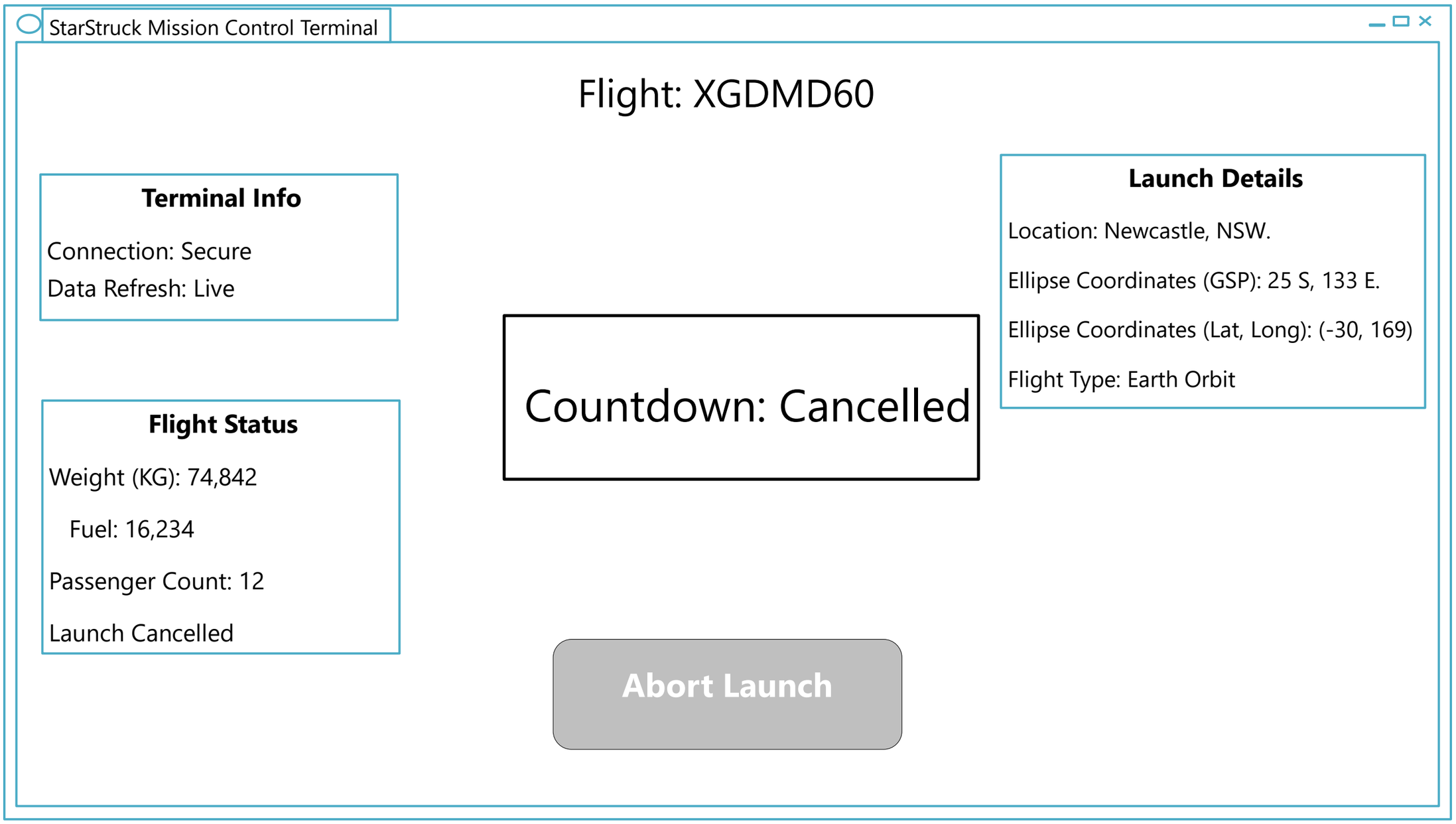
Screen #4 Description:

The following User Interface screen is the fourth screen, where in countdown to the launch has begun, however mission control is pressed the Abort Launch button. The difference between this screen and the screen 3 is that a dialog box is opened up and is prompting mission control to confirm their choice to abort the spacecraft's flight.



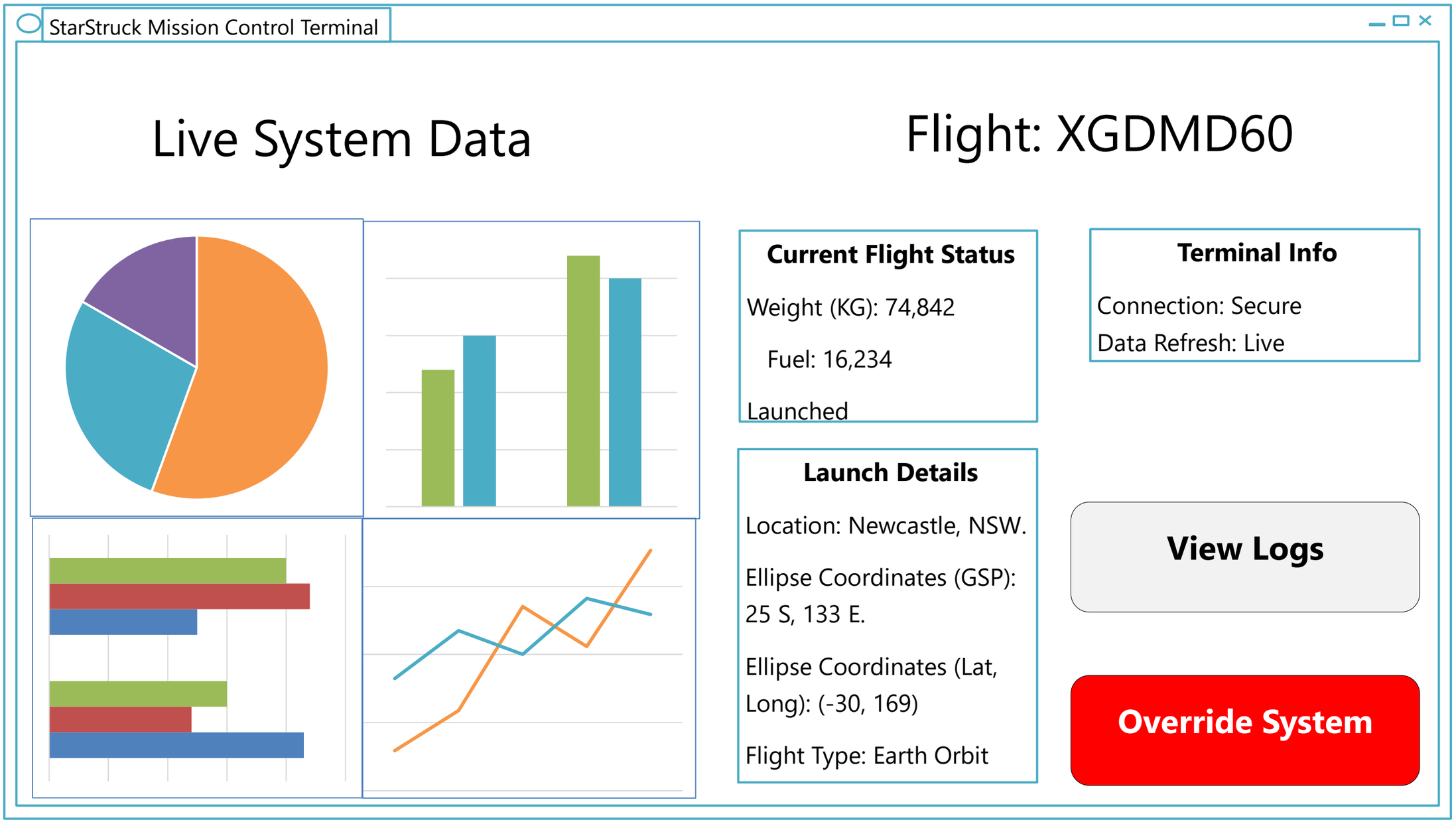
Screen #5 Description:

The following User Interface screen is result of mission control confirming their decision to abort the launch countdown, this is reflected via the abort launch button being greyed out.



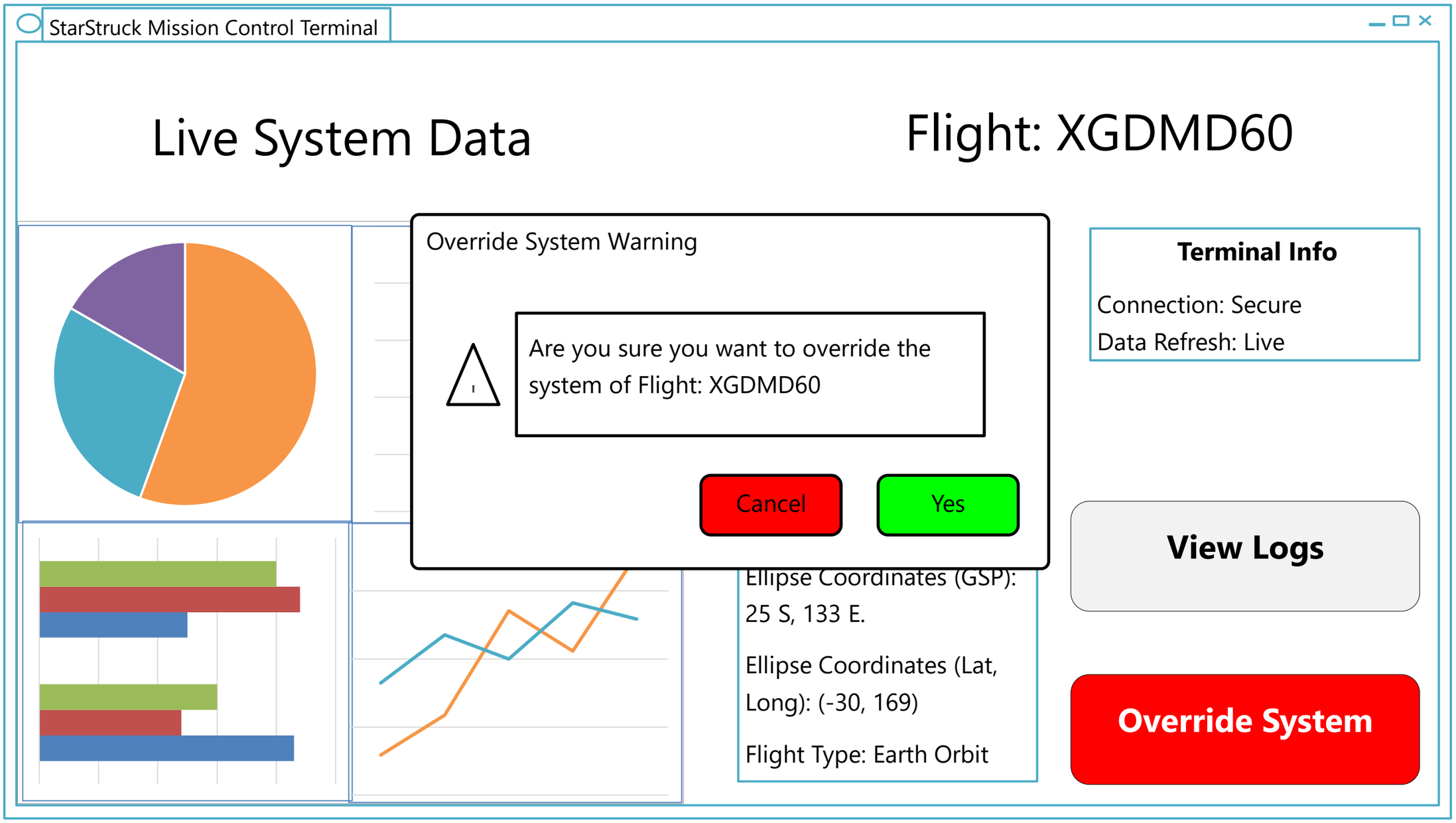
Screen #6 Description:

The following User Interface screen is a result of the countdown being completed, and the spacecraft successfully launching. It contains multiple new elements, and reflects the live data constantly being sent from the spacecraft through the four diagrams, which update in real time. In the bottom right of the screen there is two buttons, a button to override the spacecraft's system and abort the flight / trip, and another button to view the current flight logs.



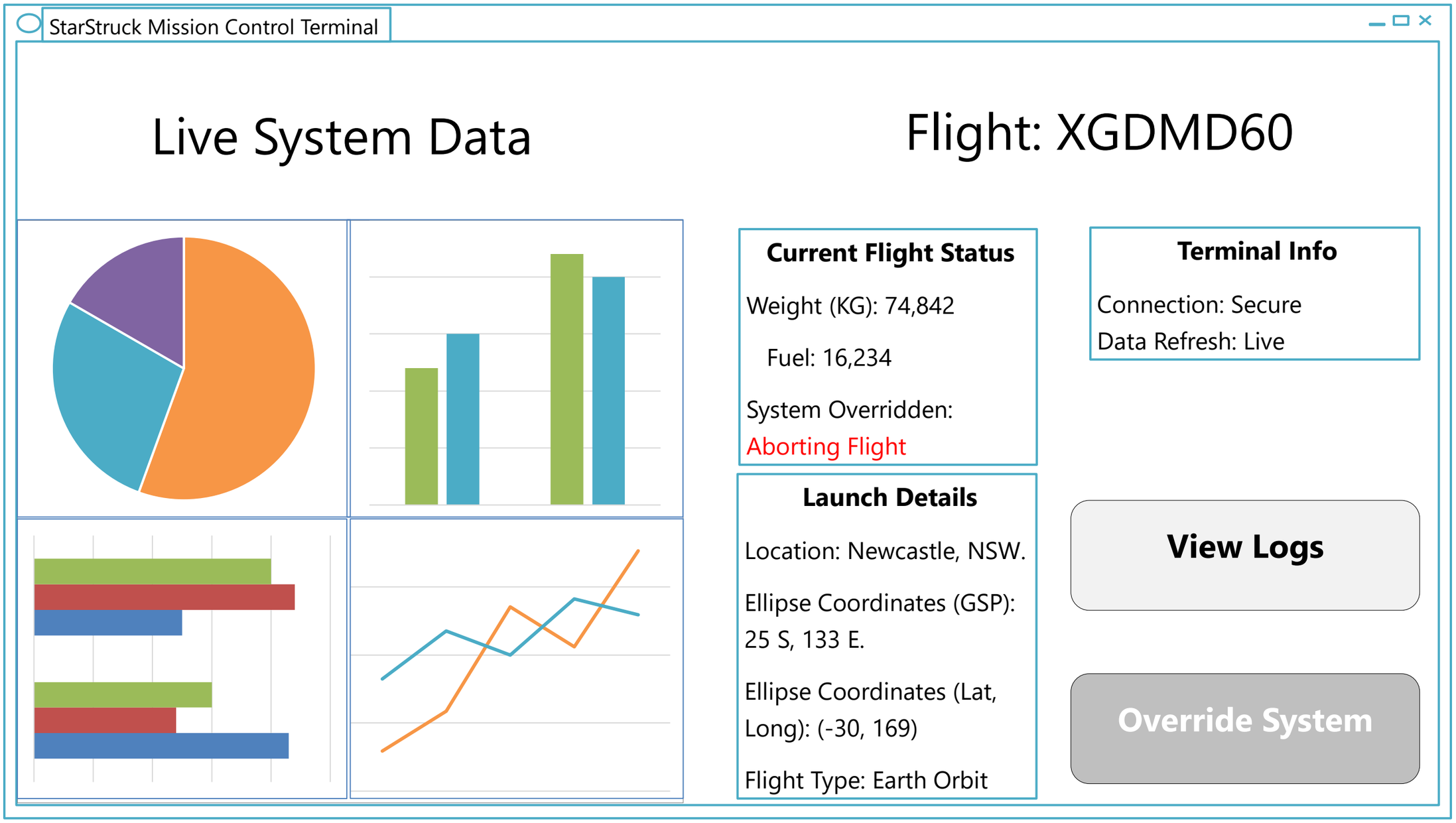
Screen #7 Description:

The following User Interface screen contains the exact same data and information which is being rendered from the last screen, however this screen reflects the result of mission control clicking the override system button. Where a dialog box appears and prompts mission control to confirm their choice.



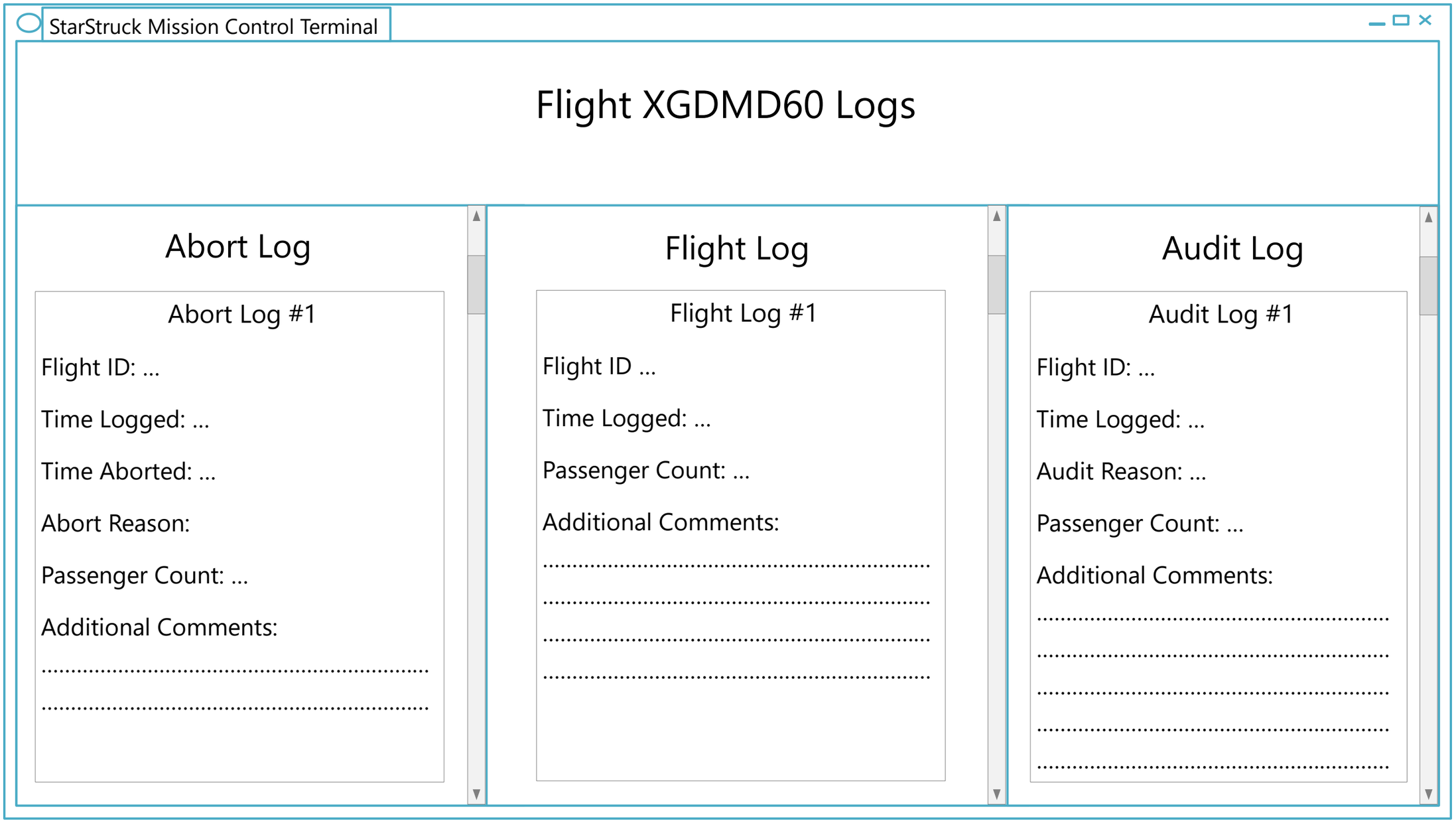
Screen #8 Description:

The following User Interface screen is the result of mission control electing to abort the spacecraft's trip / flight. Mission control will still receive live data until the spacecraft has landed in at it's Ellipse coordinates.



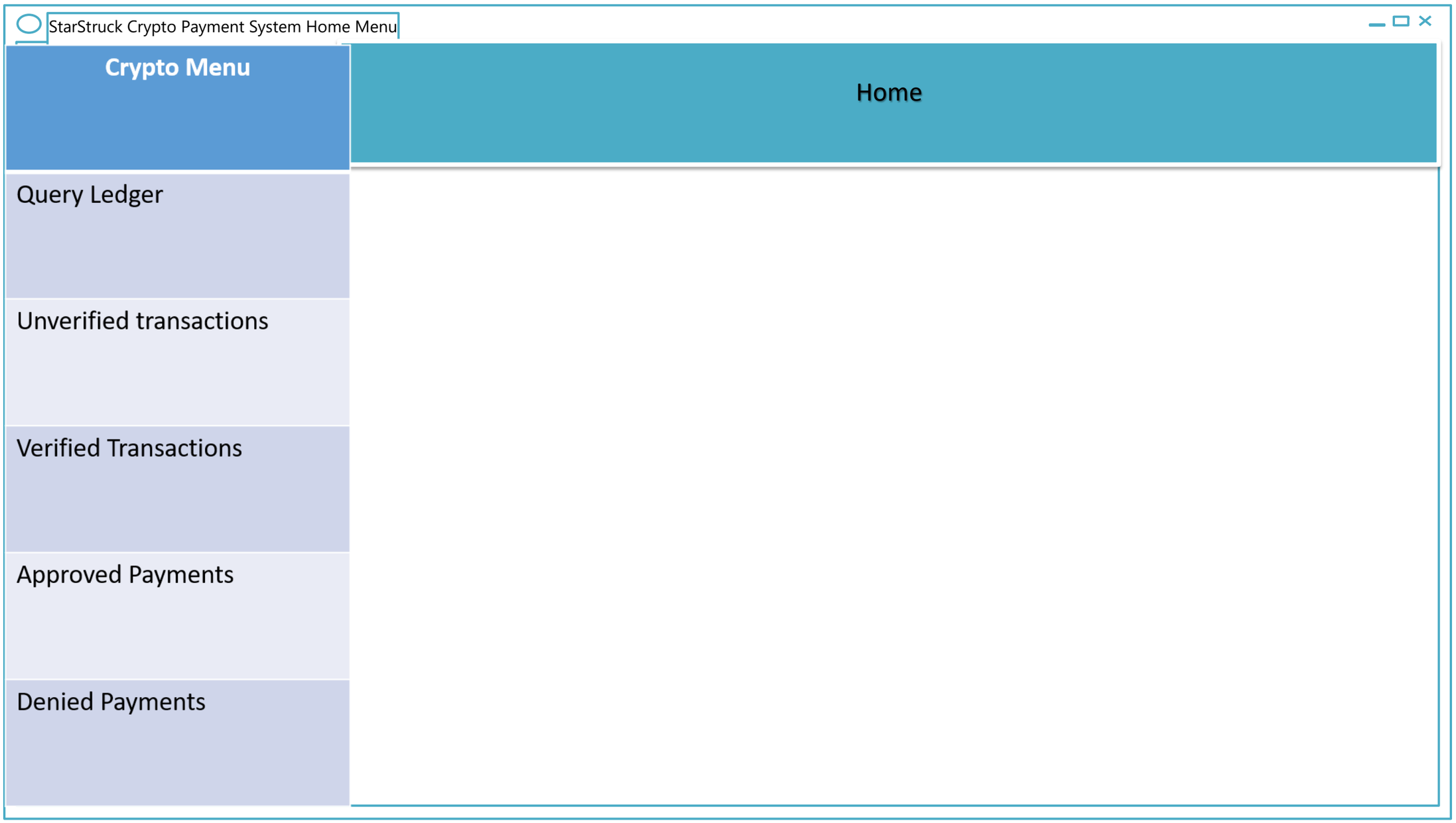
Screen #9 Description:

The following User Interface screen is the last screen which is involved in the mission control terminal, and is the result of mission control clicking the button to view the Flight's logs. This screen opens up above the prior screen, and can be closed by clicking the X icon in the top right of the screen.

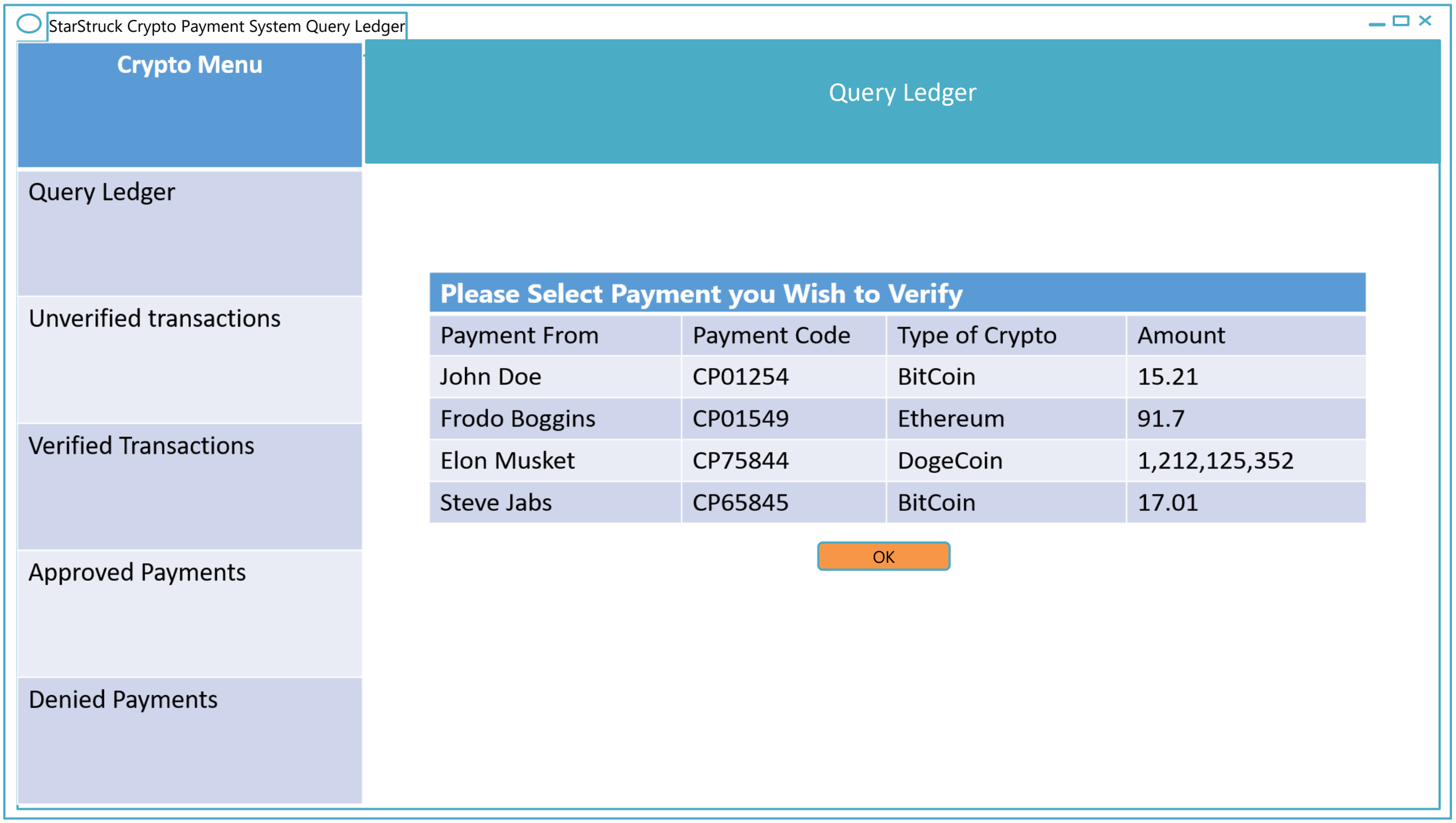


CRYPTO PAYMENT USER INTERFACE

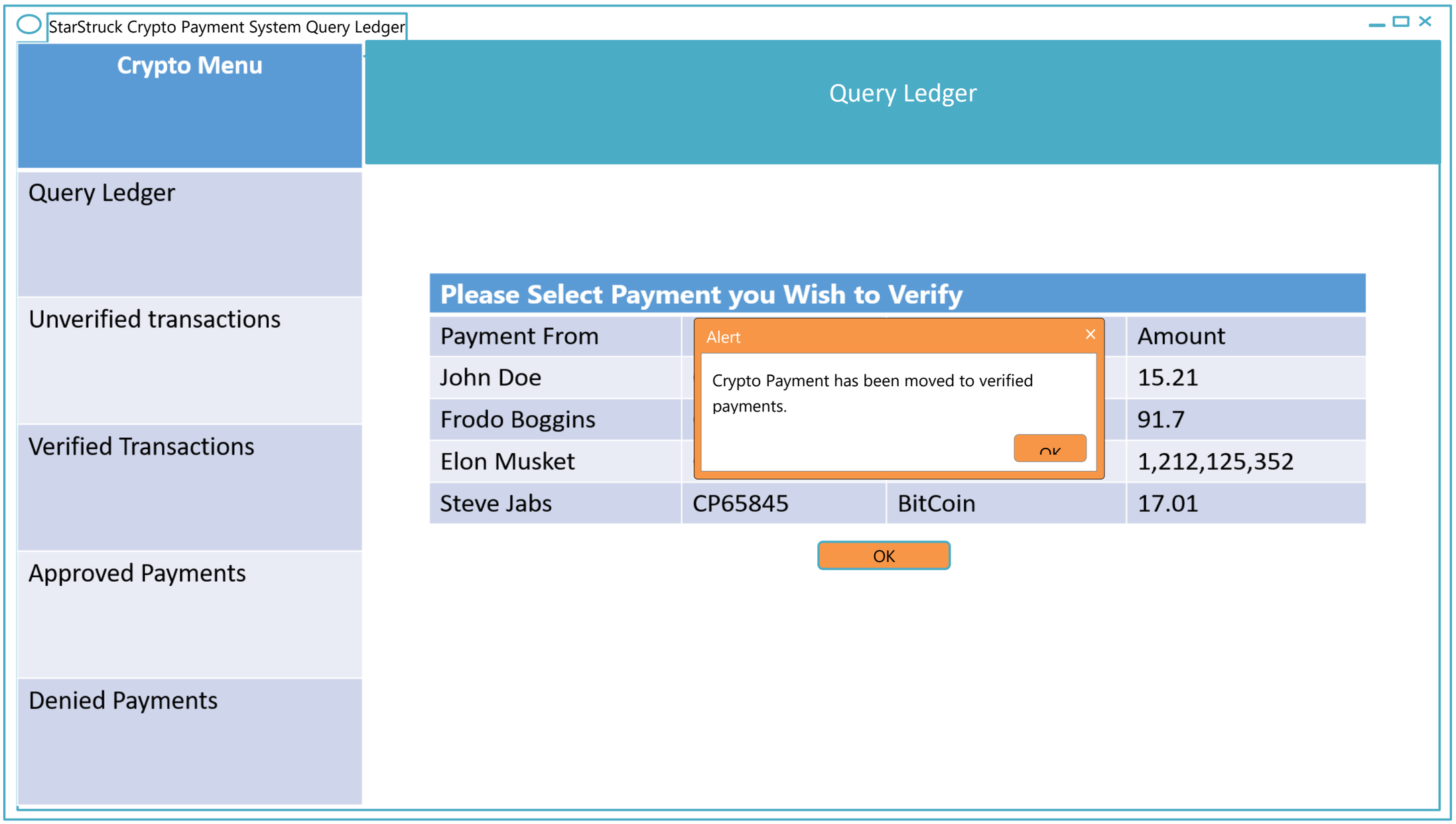
This is the home screen for the Administration Staff when dealing with crypto currency payments.



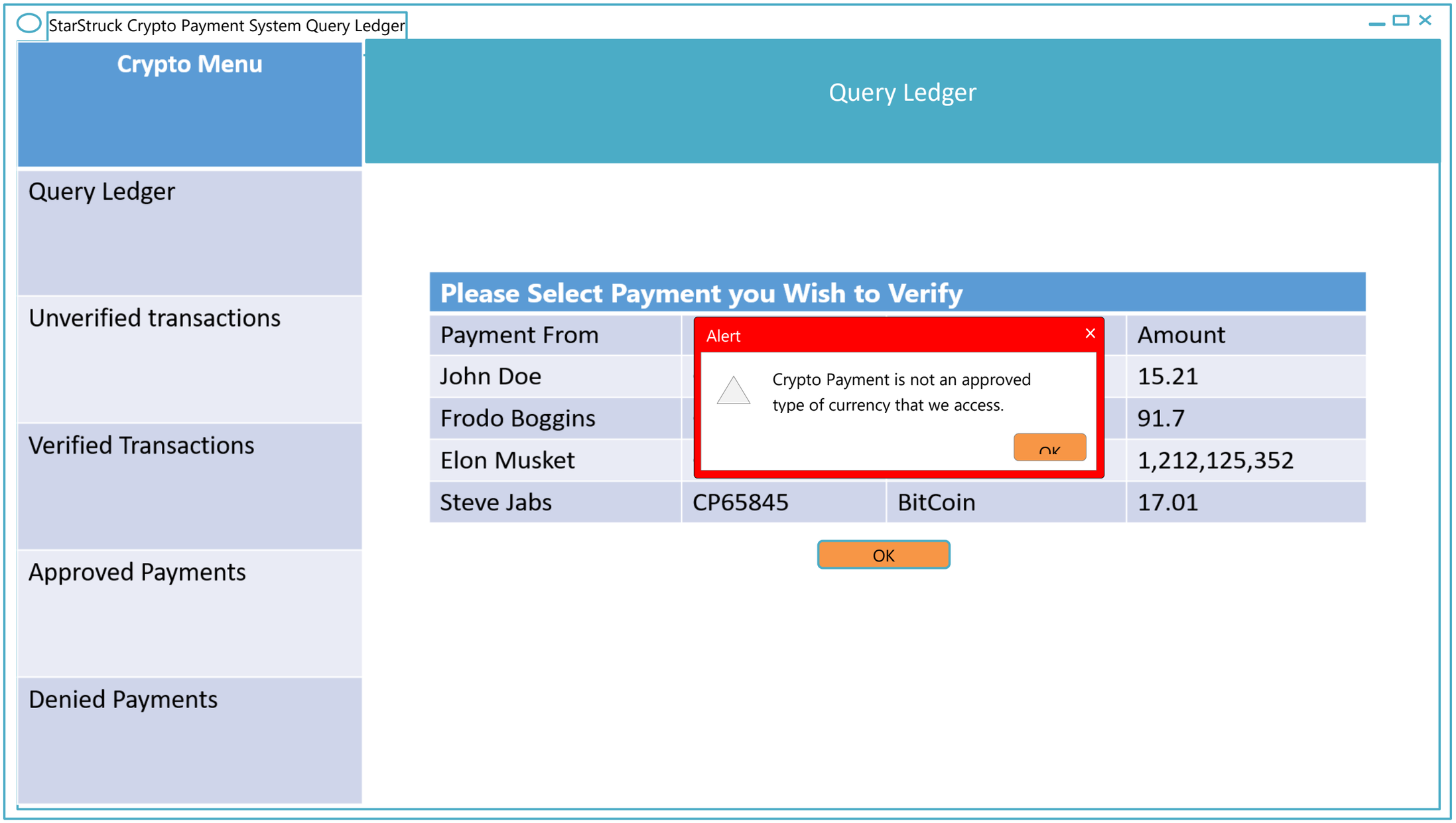
This Is the Query ledger tab, the admin staff from here can see all crypto payments inside the ledger and try to verify if they are a correct form of payment.



This Alert prompt will pop up if the Crypto payment is a correct form of payment, and it will moved to the Verified Transactions Page.



This Alert prompt will pop up if the Crypto payment is an incorrect form of payment, and it will be moved into the Denied Payments Page.



This Page is For Administration Staff to Approve Verified Payments.

StarStruck Crypto Payment System Verified Payments

Crypto Menu

- Query Ledger
- Unverified transactions
- Verified Transactions
- Approved Payments
- Denied Payments

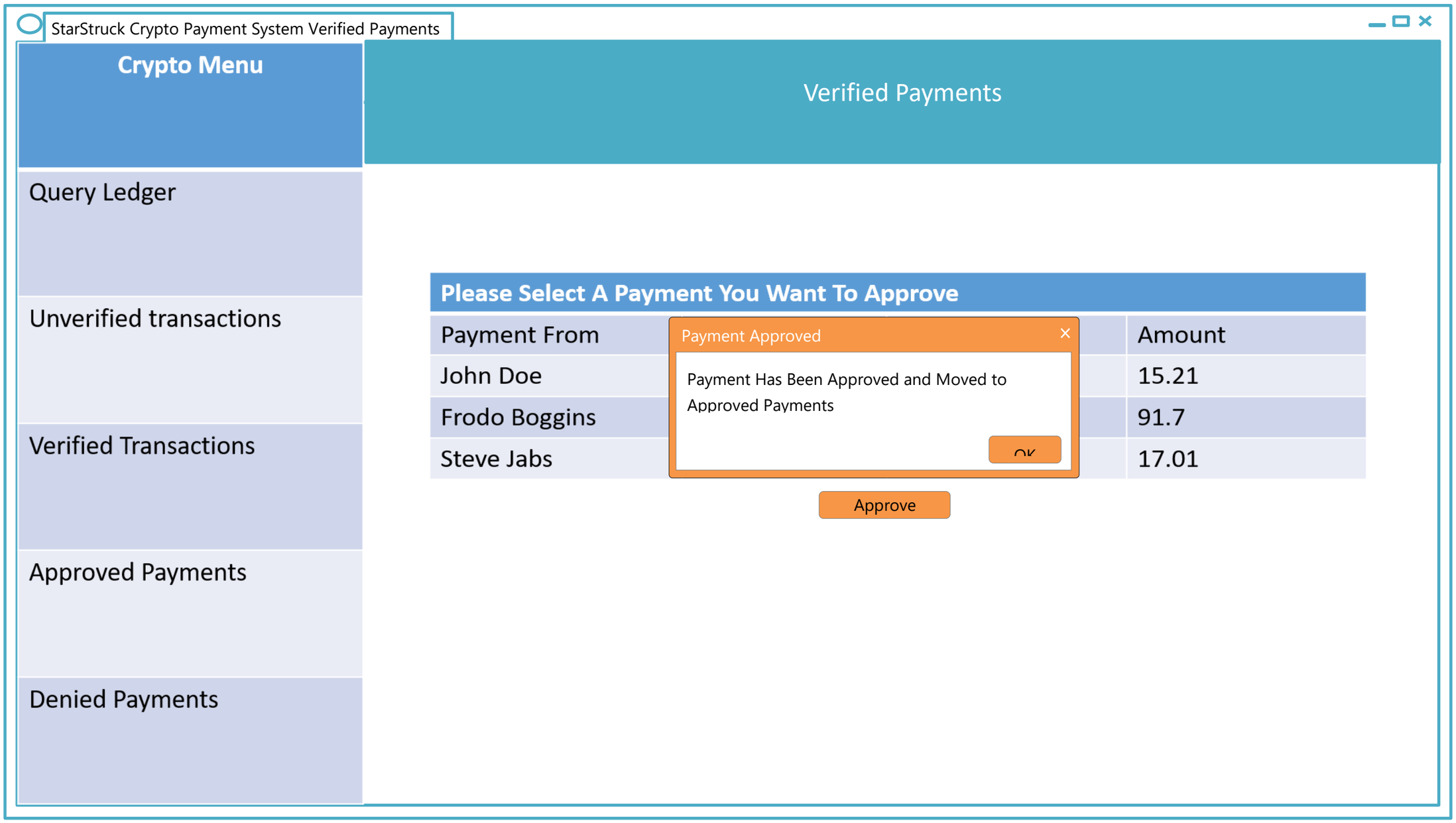
Verified Payments

Please Select A Payment You Want To Approve

Payment From	Payment Code	Type of Crypto	Amount
John Doe	CP01254	BitCoin	15.21
Frodo Boggins	CP01549	Ethereum	91.7
Steve Jabs	CP65845	BitCoin	17.01

Approve

This Alert Prompt will pop up when a payment has been approved and it will move the payment into the Approved Payments Page.



In this Page the Admin Staff Can View All Approved Payments.

StarStruck Crypto Payment System Approved Payments

Crypto Menu

- Query Ledger
- Unverified transactions
- Verified Transactions
- Approved Payments
- Denied Payments

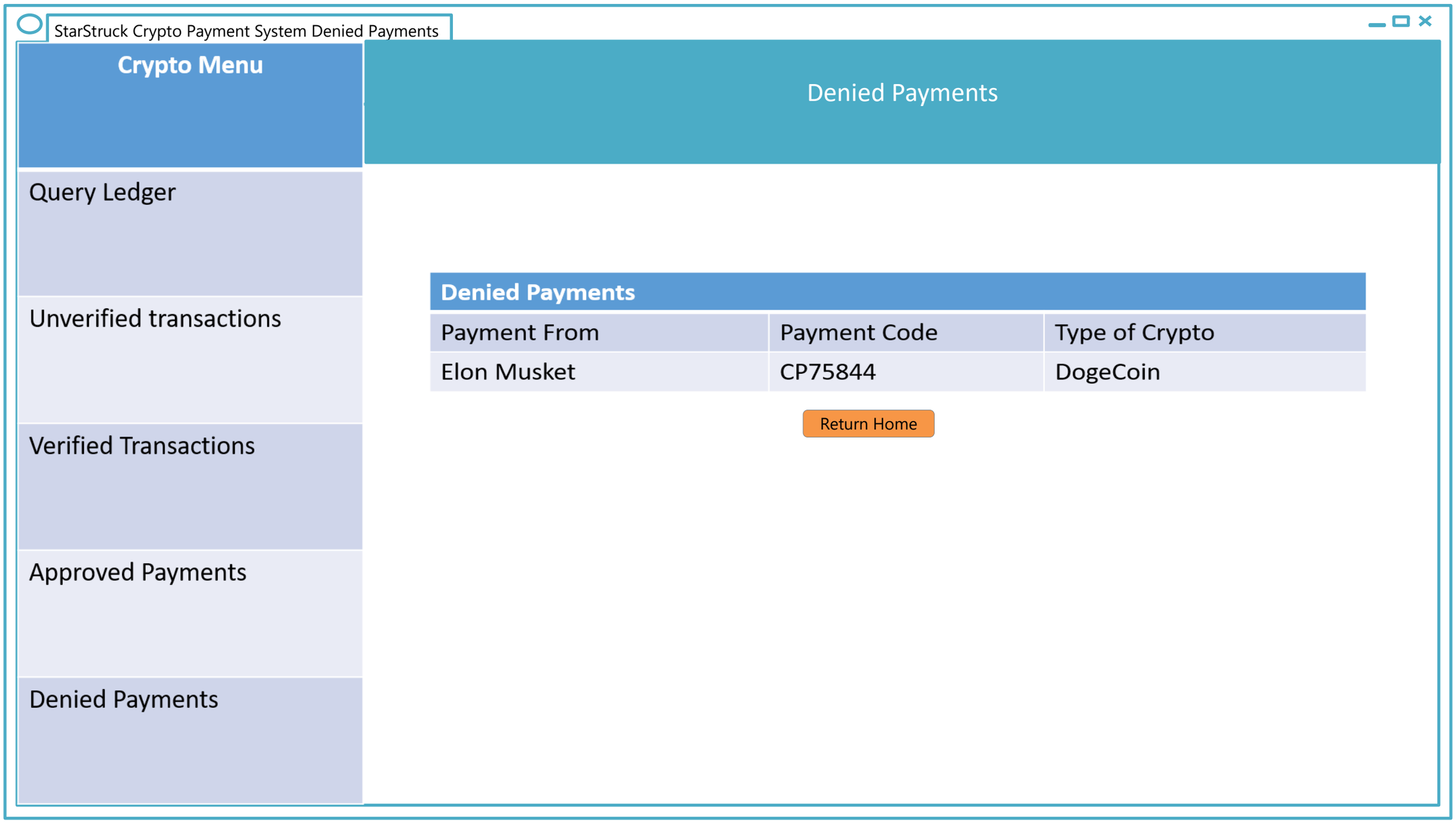
Approved Payments

List Of Approved Payments

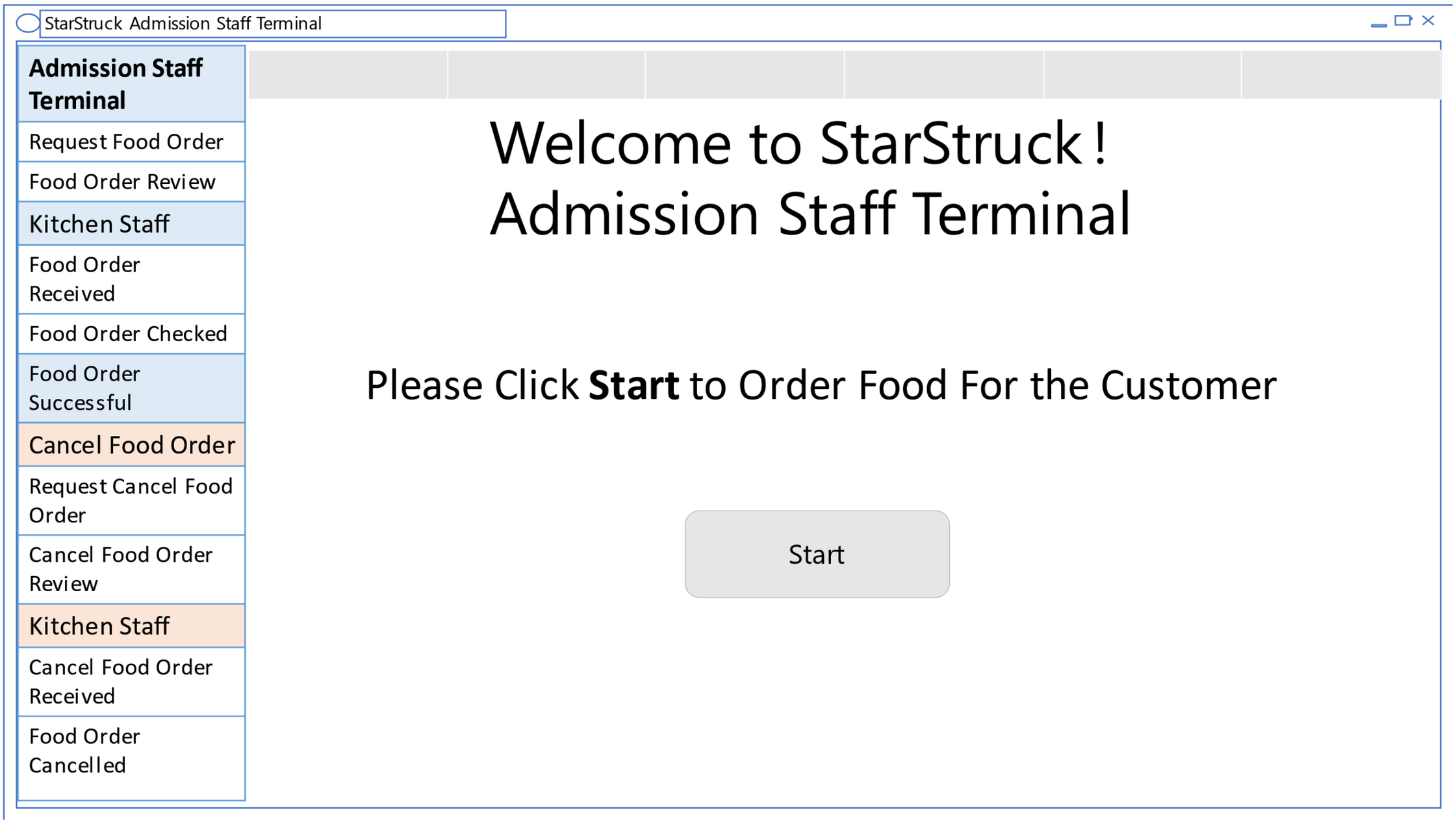
Payment From	Payment Code	Type of Crypto	Amount	Date Approved
John Doe	CP01254	BitCoin	15.21	02/12/2021
Frodo Boggins	CP01549	Ethereum	91.7	31/04/2022
Steve Jabs	CP65845	BitCoin	17.01	19/06/2022

Return Home

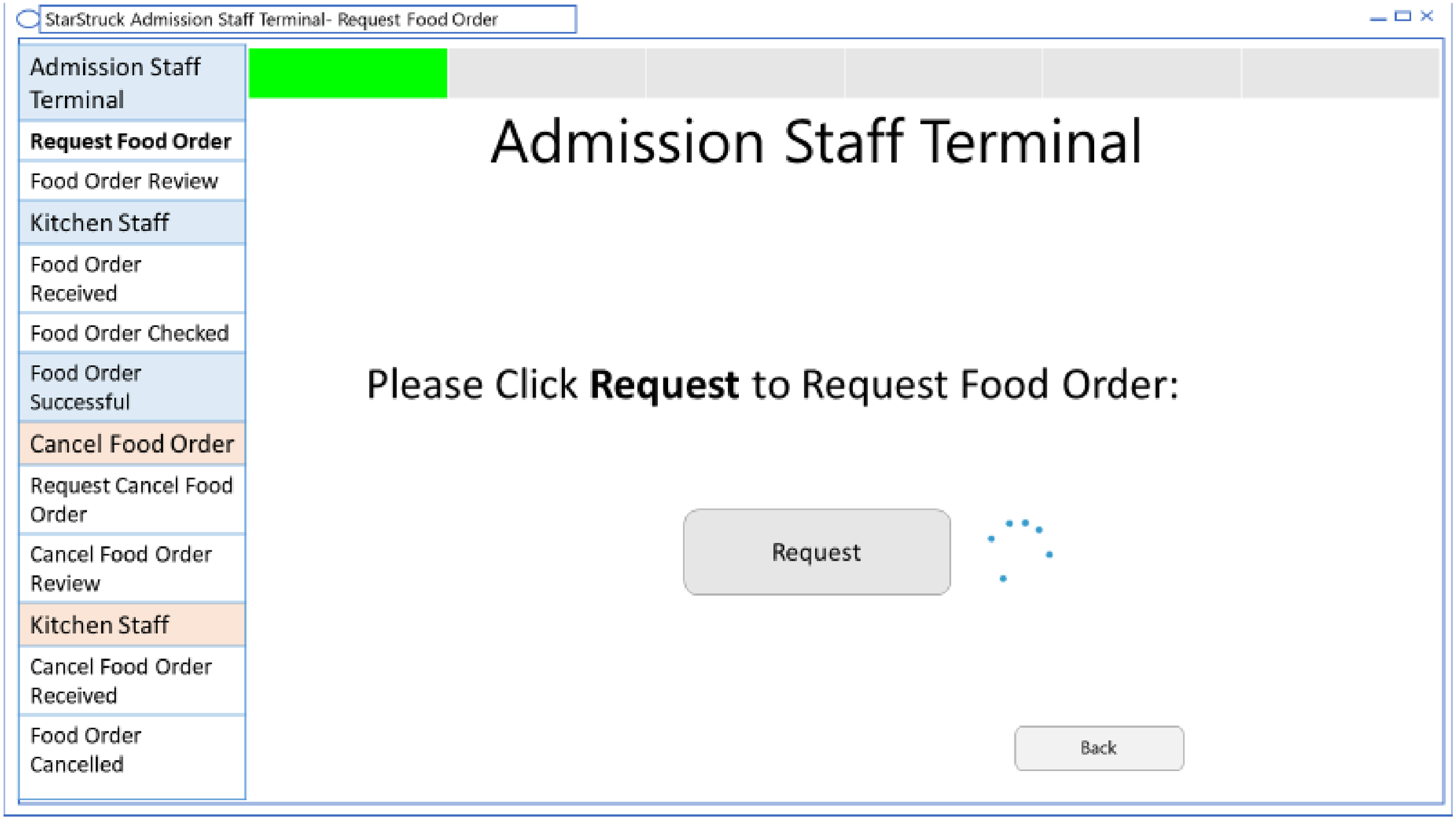
this Page the Admin Staff Can View All Denied Payments.



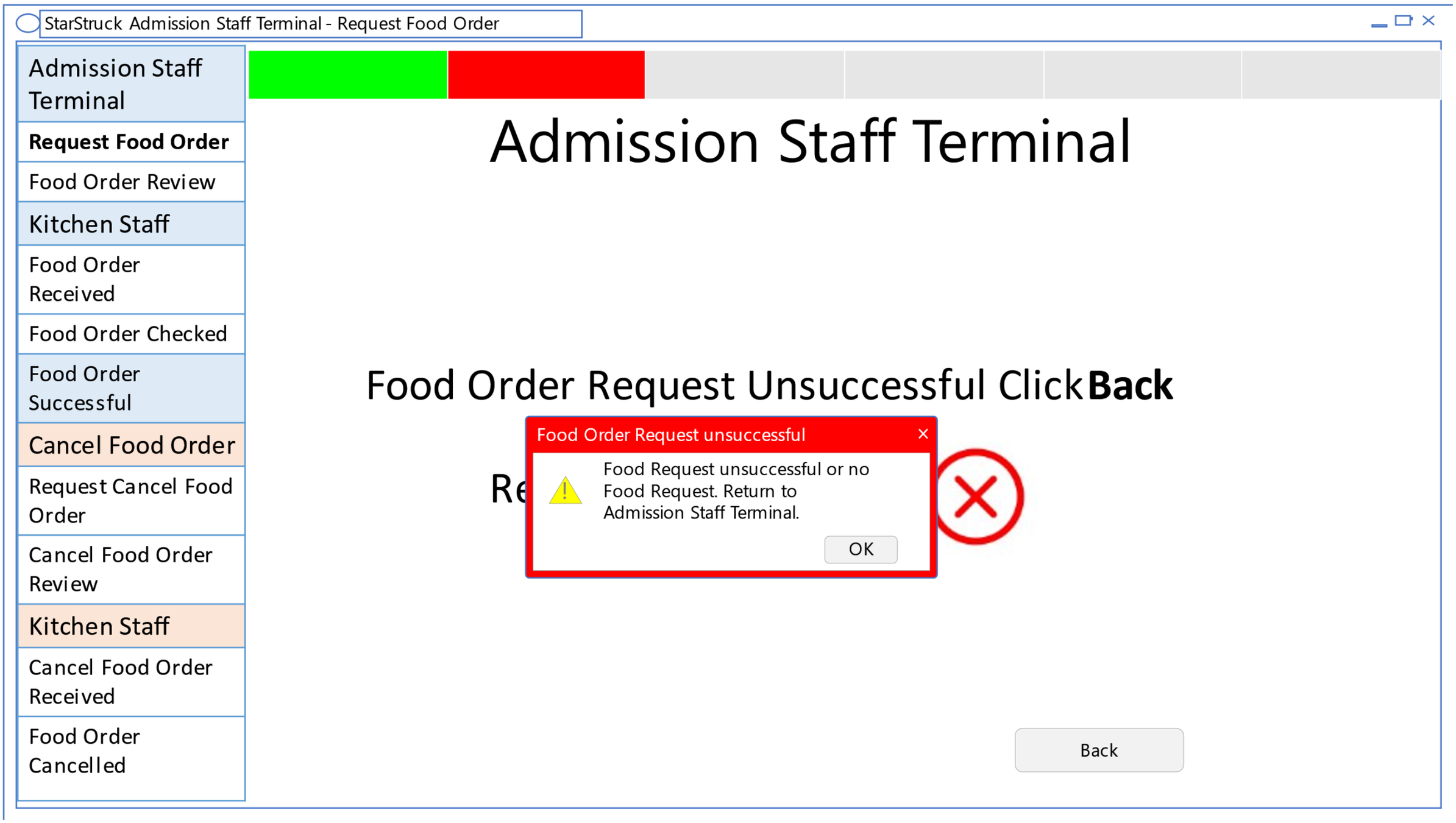
FOOD ORDER USER INTERFACE



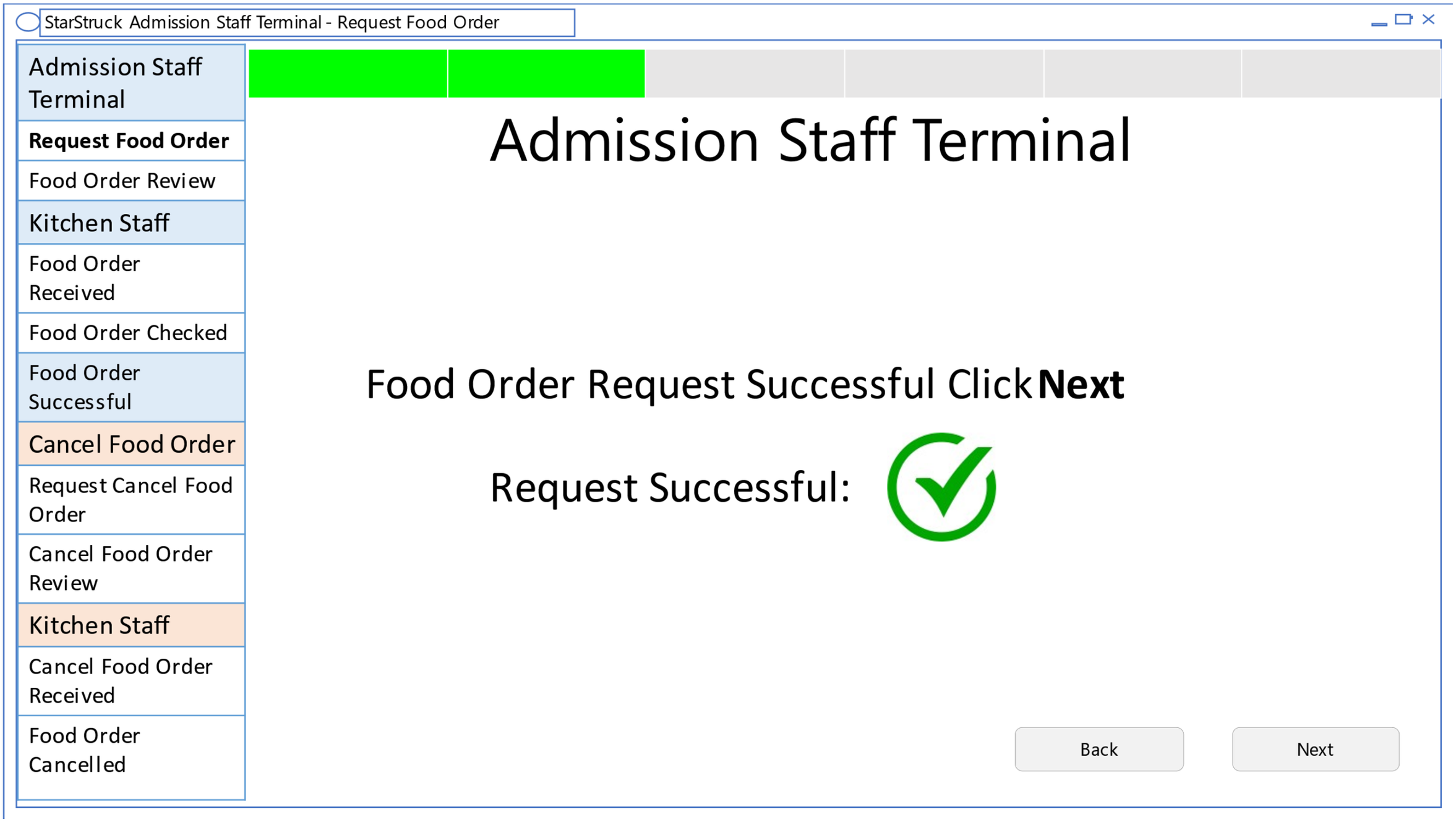
This is the starting page for Food Order. This is only accessed by Admission staff. The user is directed to click start to order food for the Customer. There is a progress bar on top for the user to know when task will be complete.



Once the user clicks start the user is send to this page. In this the user needs to Request food order from request from the customer. This takes a few seconds represented by the loading sign. A green light is lit up when user has progressed.



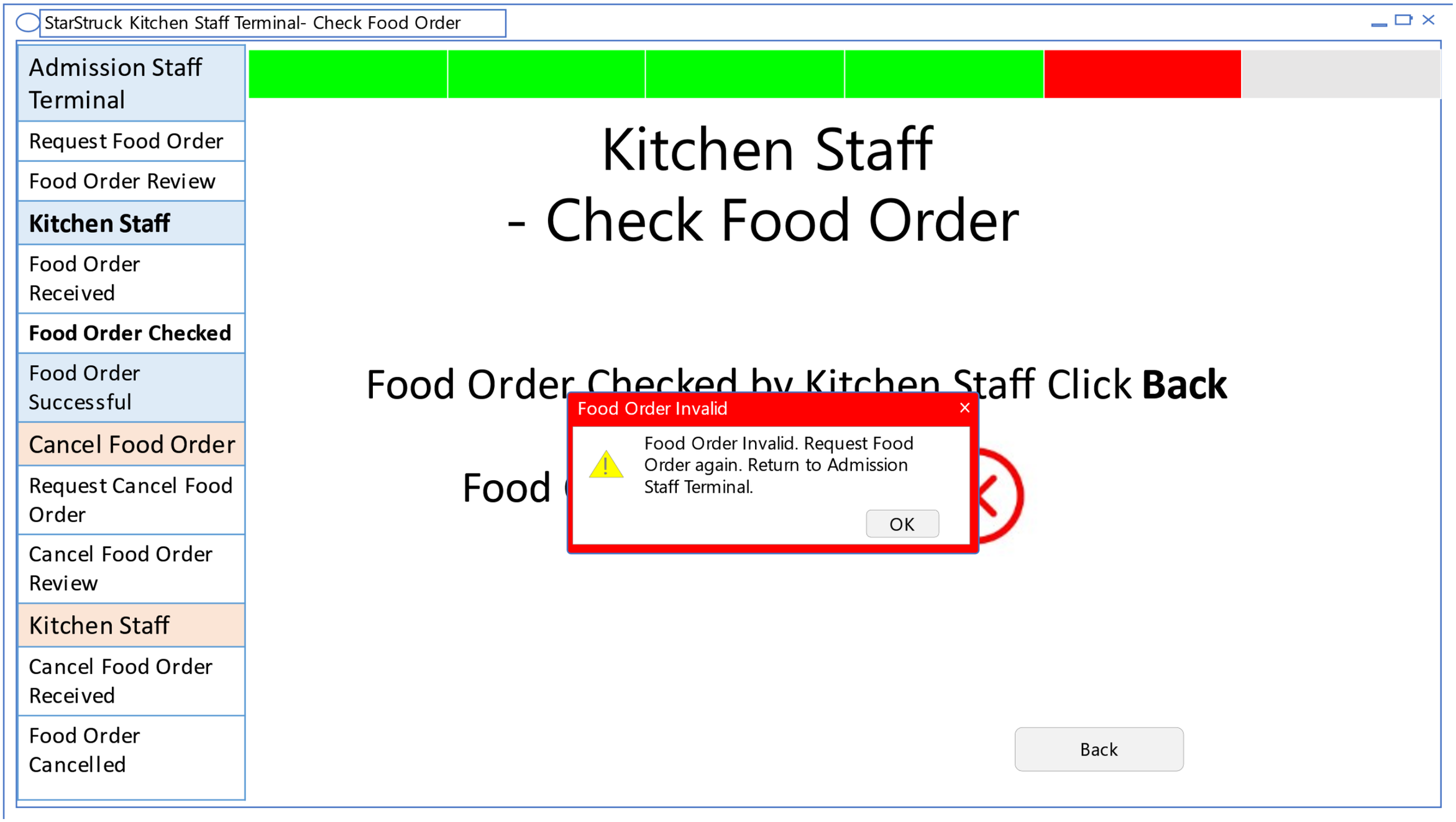
If there are no food requests from the customer. A pop-up window appears to let the user know that food order request is unsuccessful. By clicking “ok”, the user returns to the Admission Staff Terminal.



If the food order request is successful, this will see this page and is directed to click next to continue.

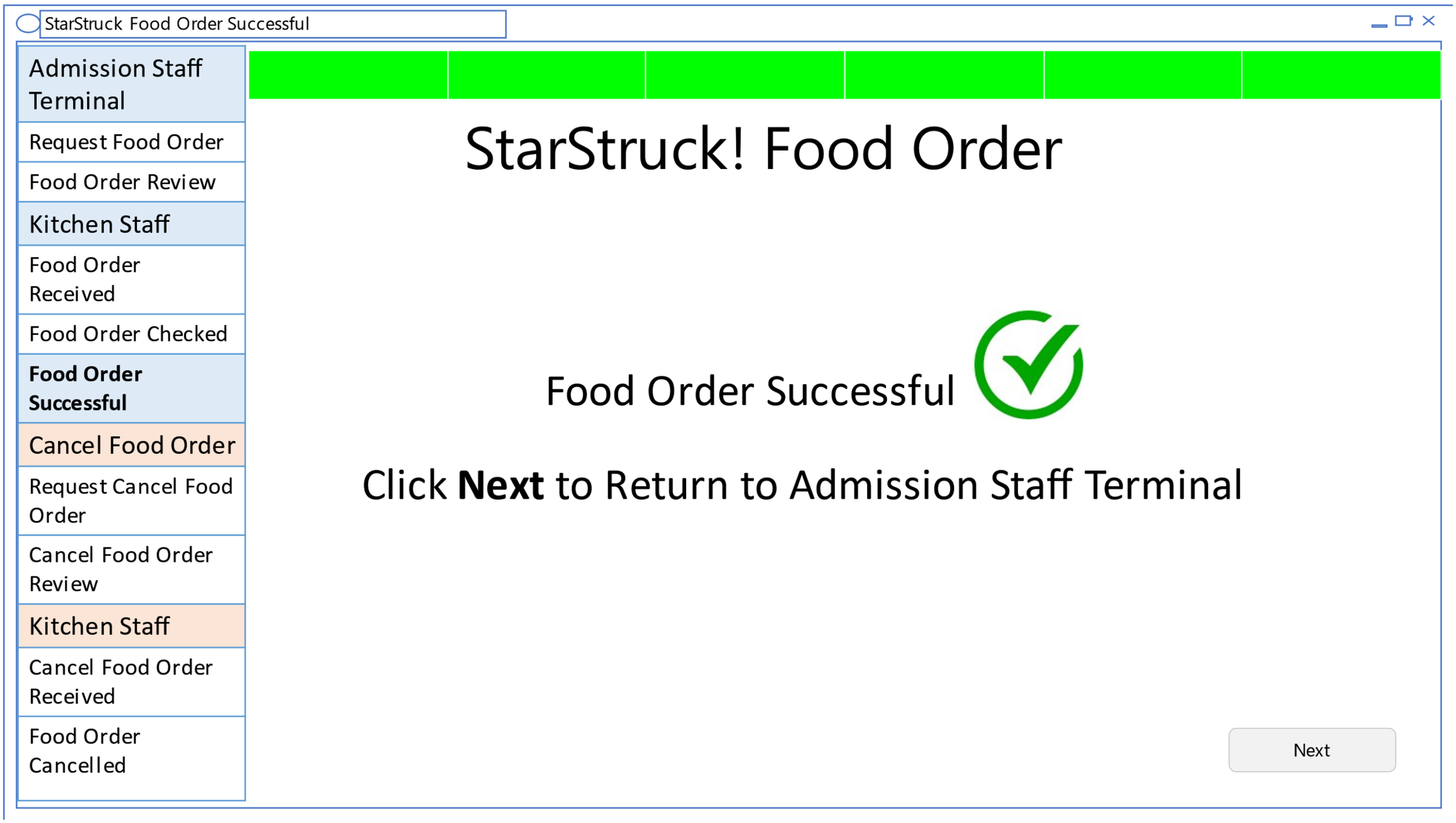
This shows the overview of the food order obtained from the request. The user is then directed to click next.

The Food Order is received by the Kitchen staff. And the user is directed to click next.

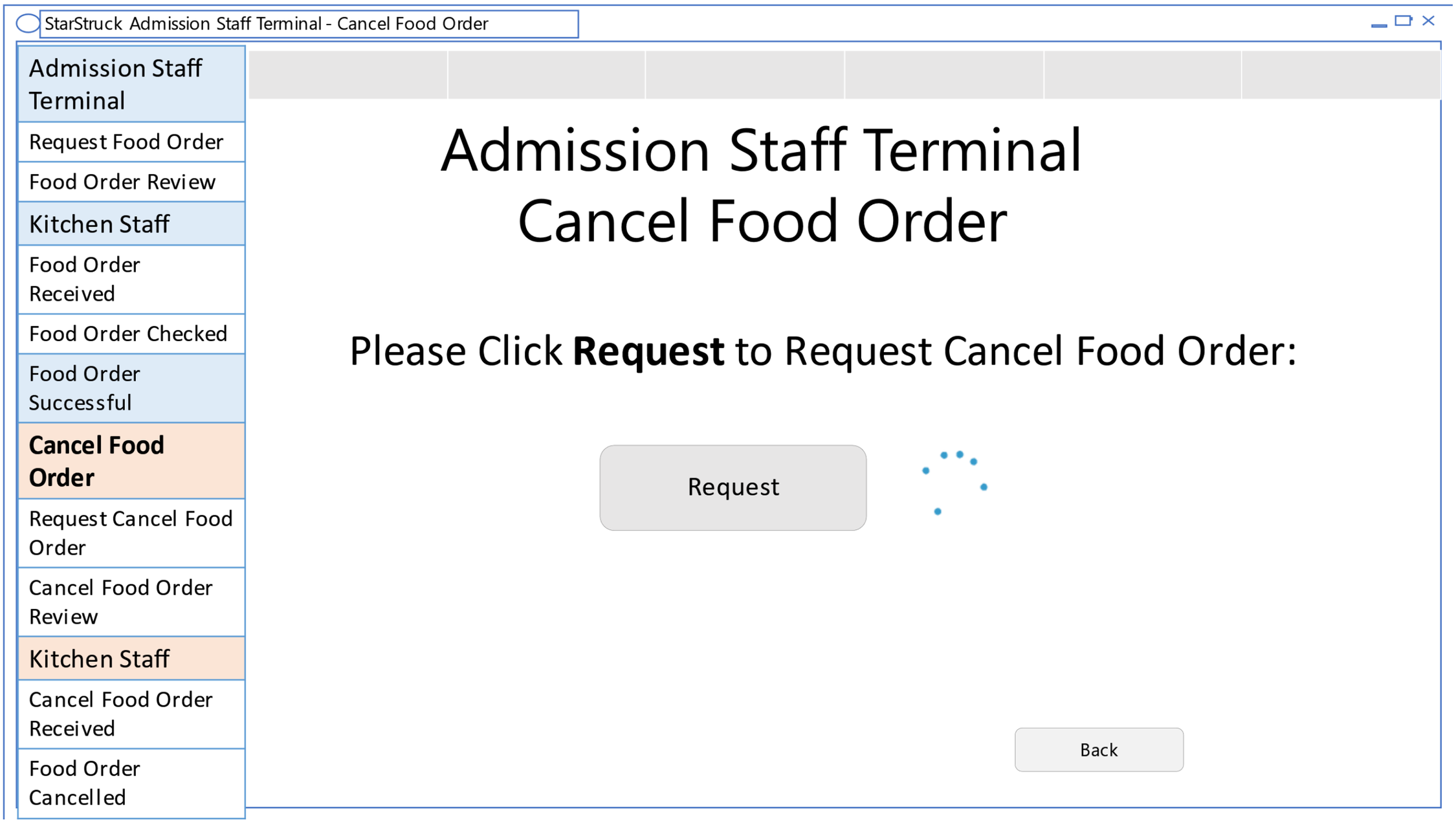


This shows that Food order is checked by Kitchen staff. This shows that Food order is unsuccessful and pop-up window directs the user to return Admission staff Terminal and start again.

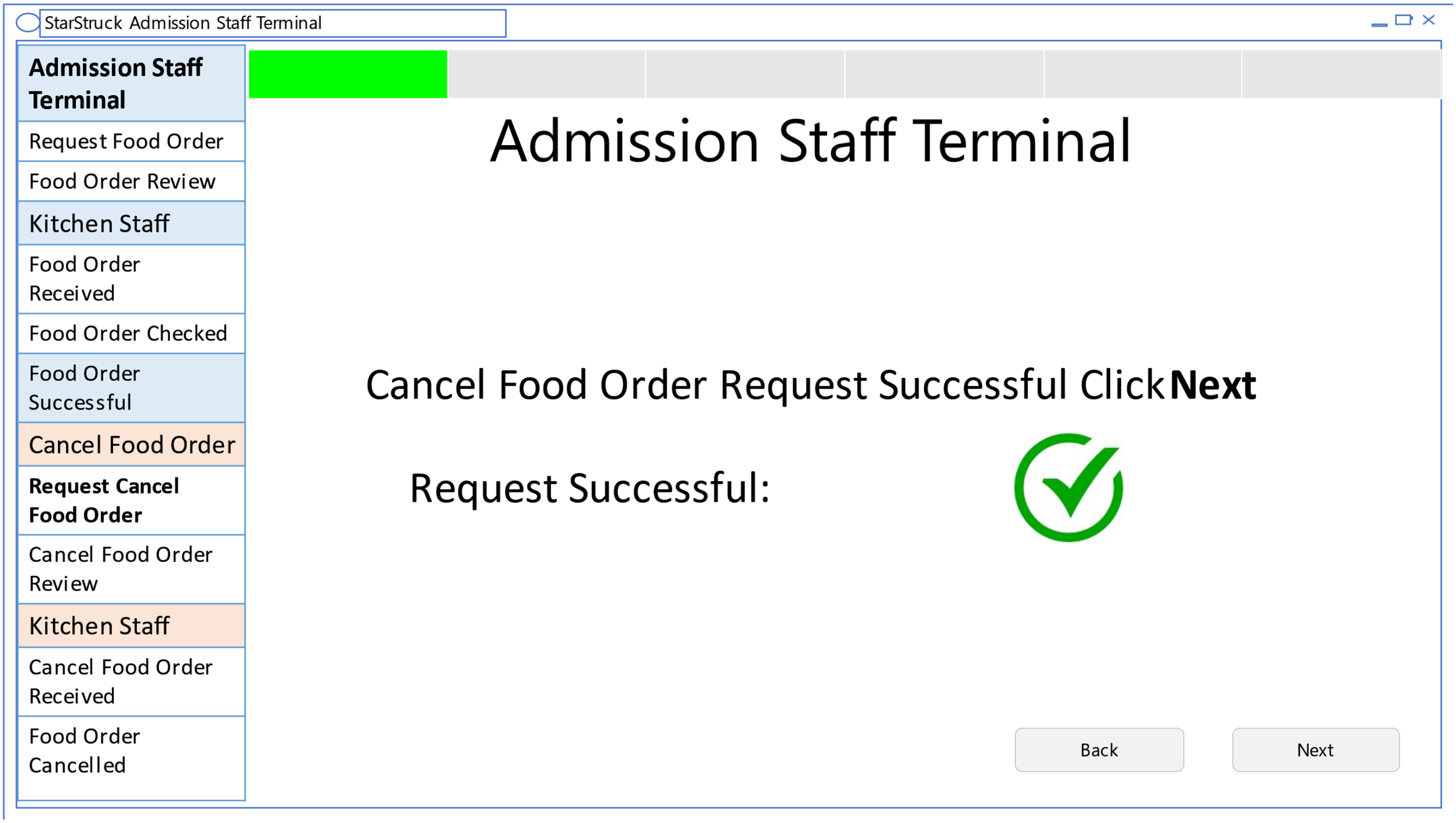
This page shows that the kitchen staff approved the food or is valid.



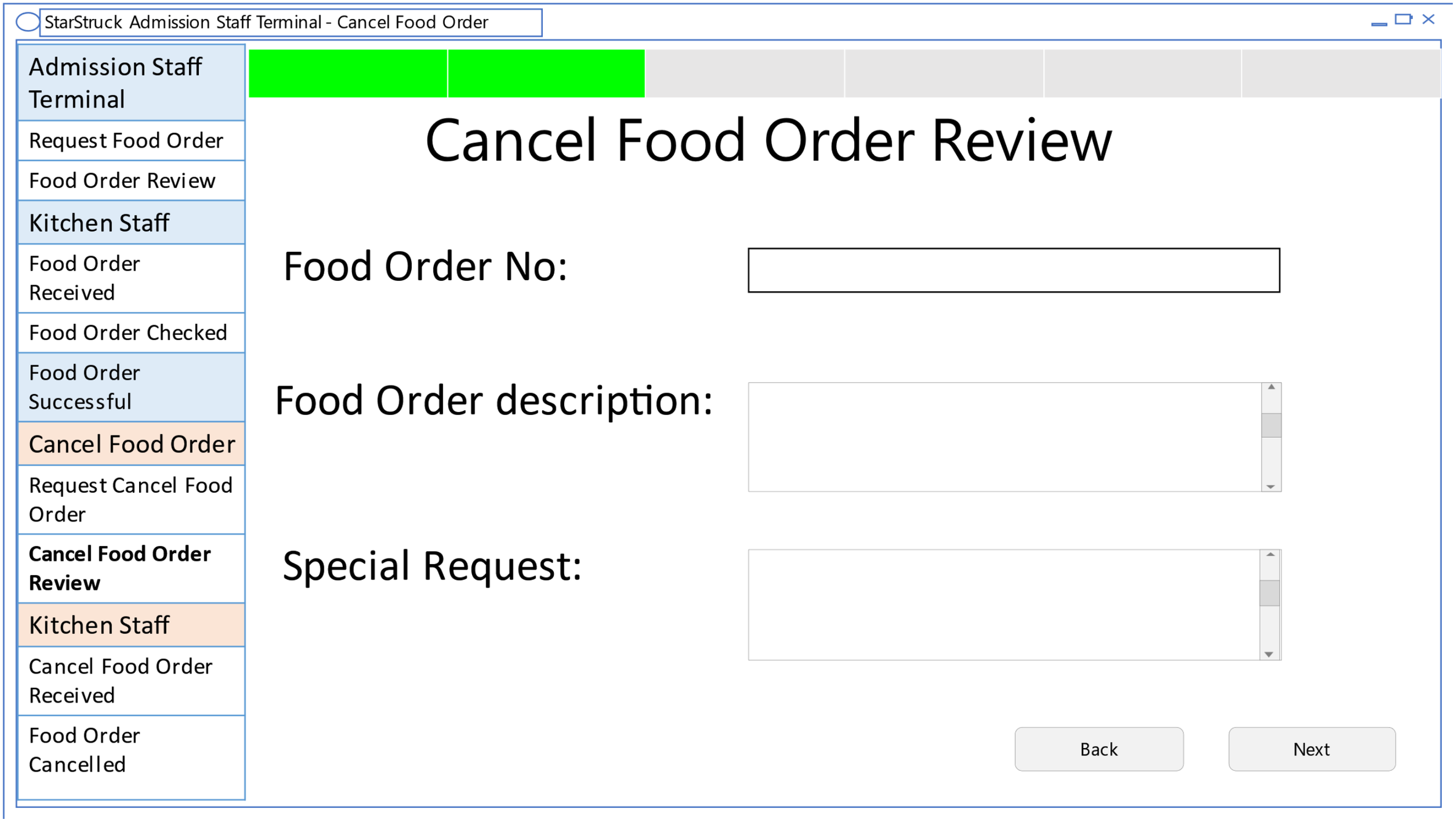
Final page showing that the food order is successful, and the user needs to click next to return back to Admission staff terminal.



In this the user needs to Request cancel food order. This takes a few seconds represented by the loading sign.



This shows that the cancelled food order is received, and the user is directed to click next.



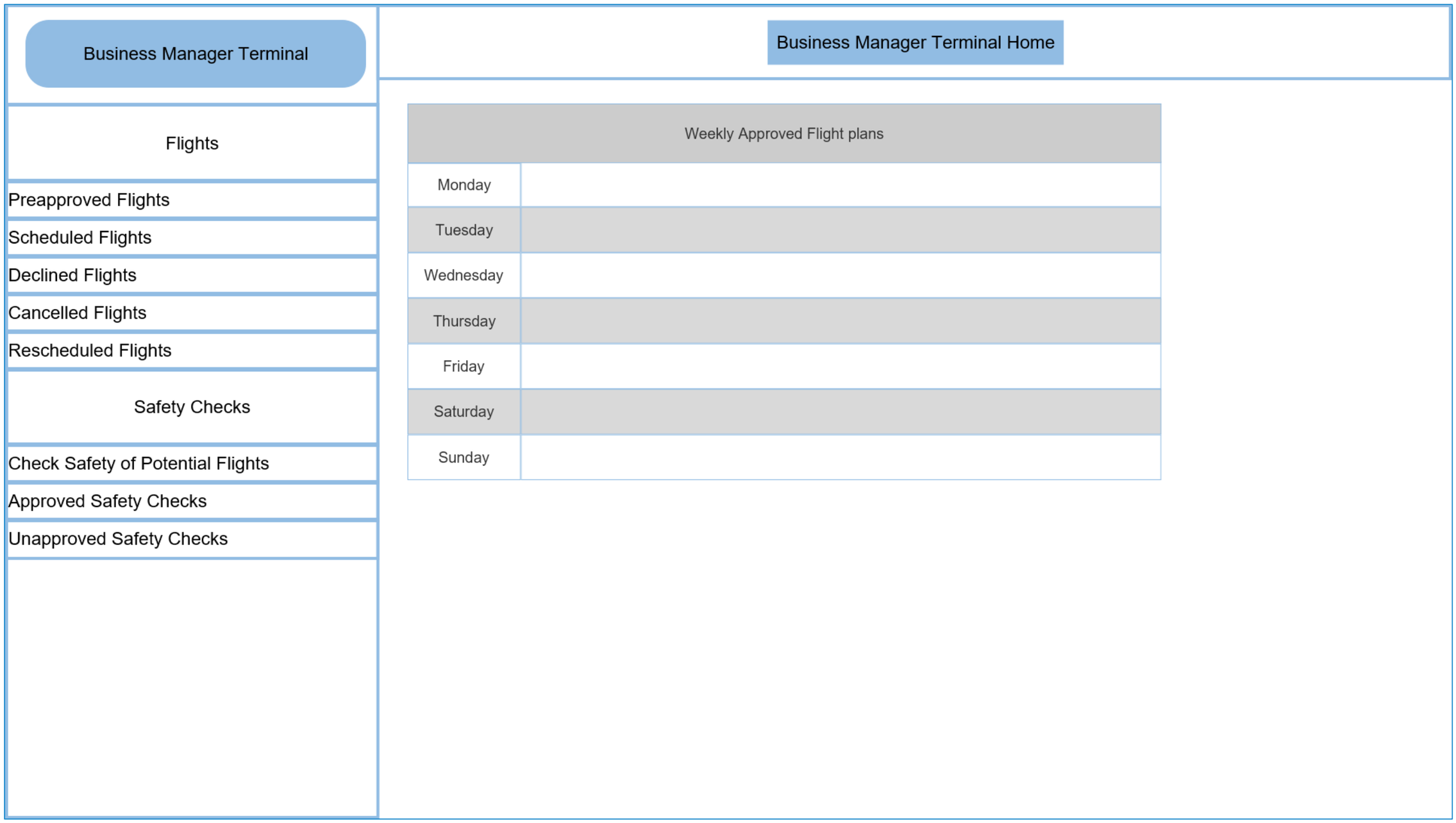
This shows the cancelled food order.

This show the cancelled food order is received by kitchen staff.

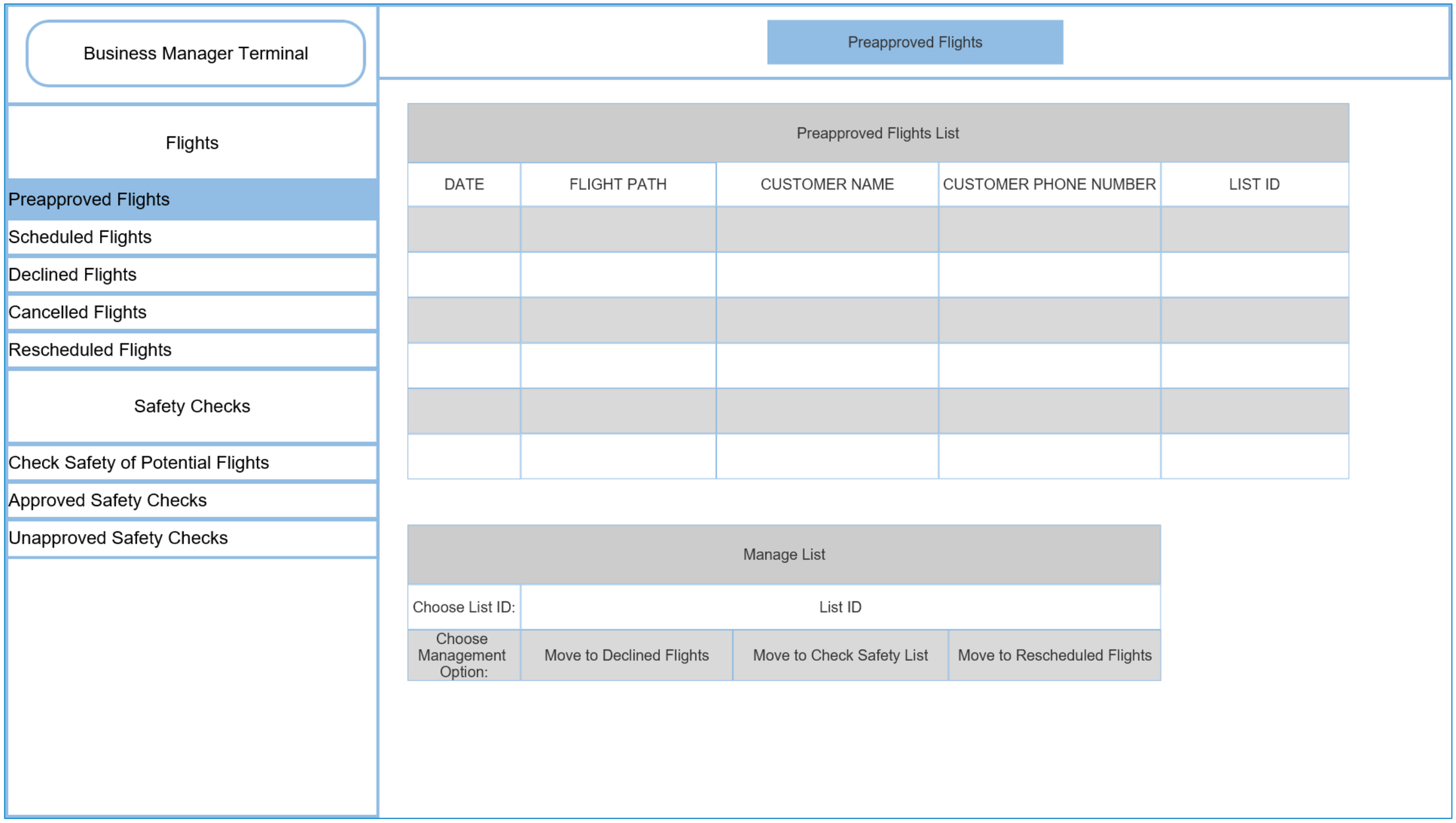
This shows the food order is cancelled by kitchen staff.

This shows that food order is cancelled. Clicking next send the user back to the Admission Staff Terminal.

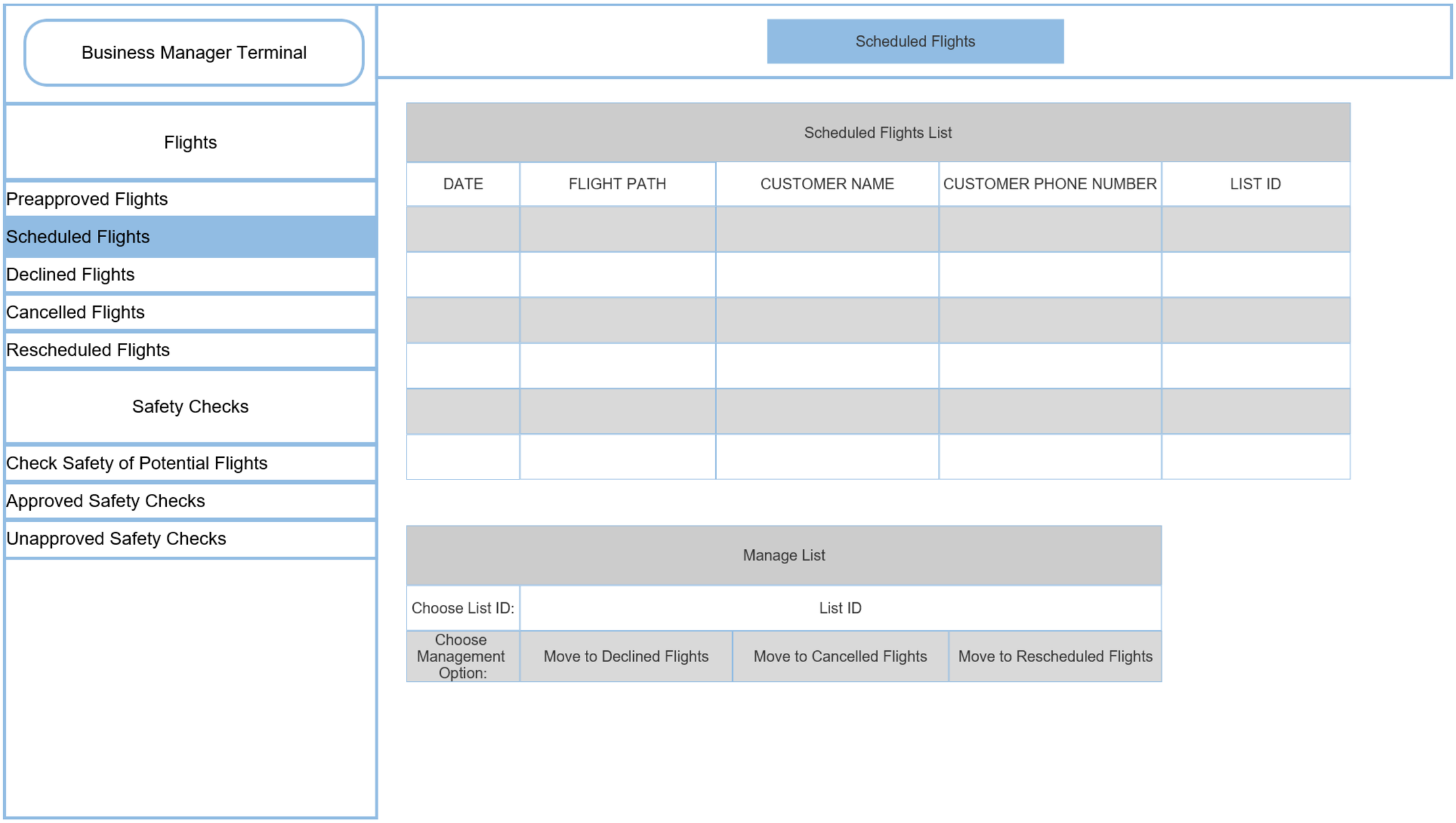
BUSINESS MANAGER TERMINAL FOR FLIGHT BOOKING USER INTERFACE



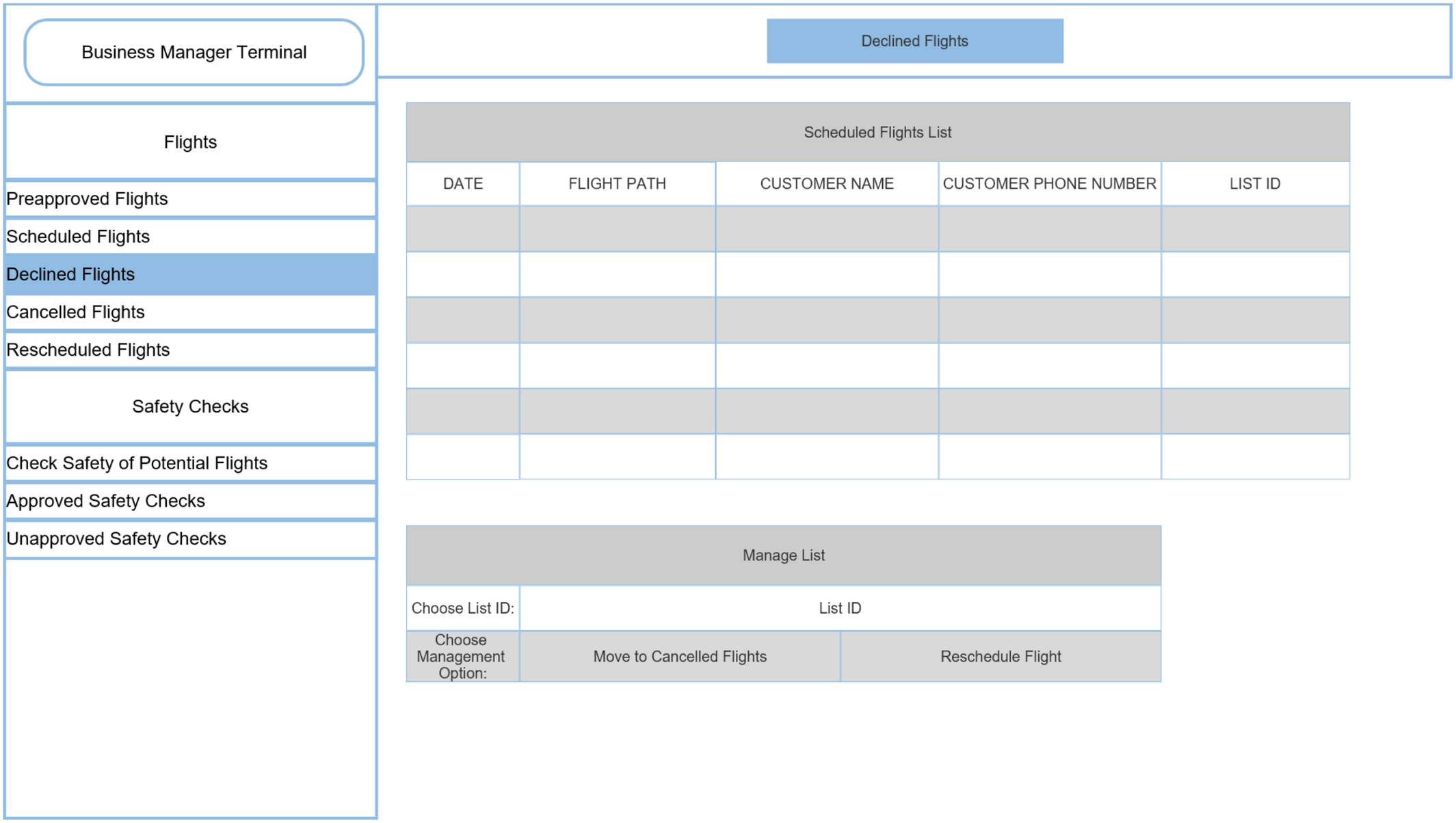
This is the home page of the Business Manager Terminal. This page has a side navigation bar on the left and allows the user to navigate to other pages. The middle of this user interface page shows the Weekly Approved Flight plans to the user. If the user wishes to manage these flight plans, they must navigate to the appropriate page.



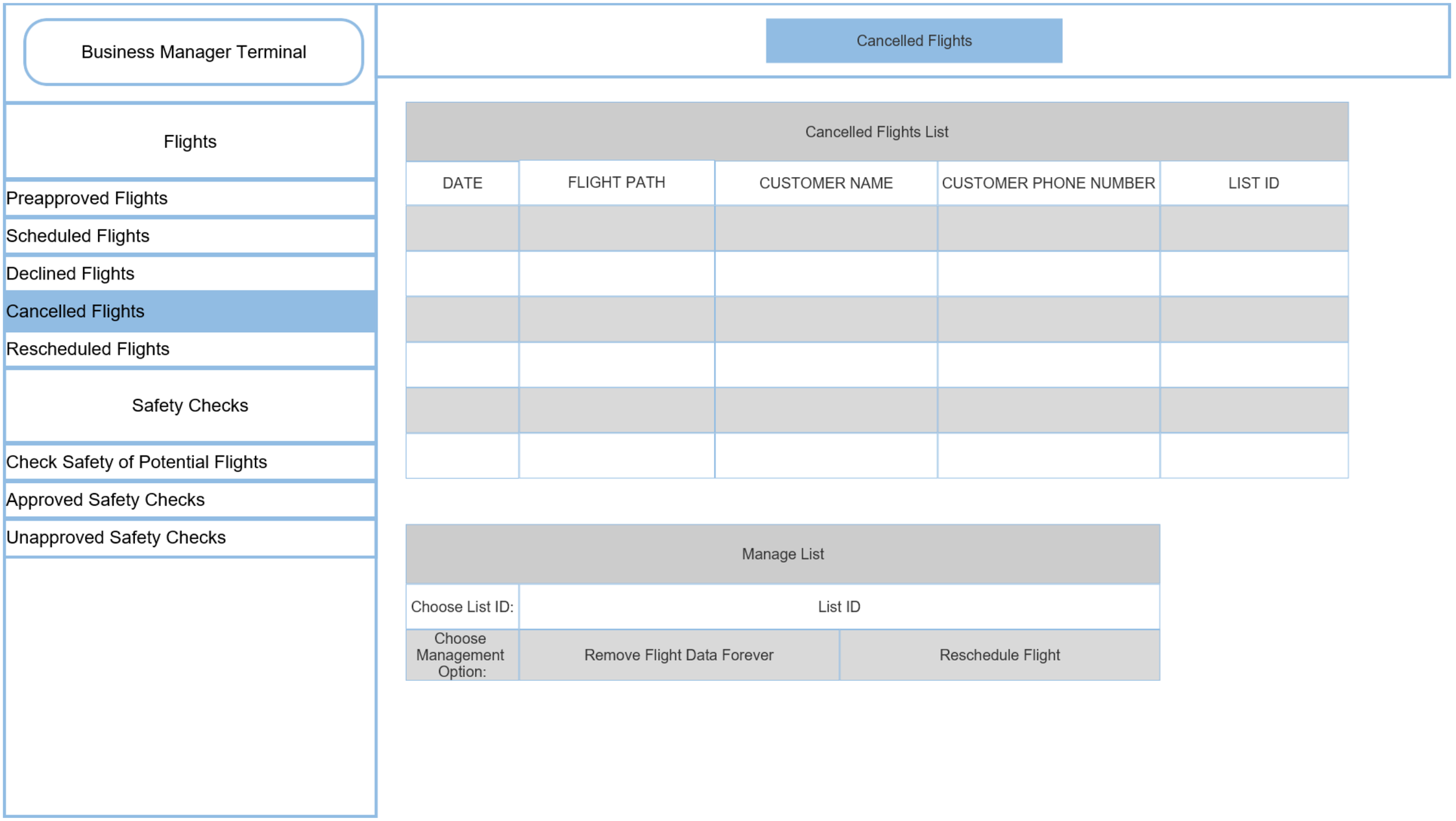
This is the Preapproved Flights user interface page. This page allows the user to see the flights waiting approval. The user can also manage this list, by moving flights to declined flights, by moving flights to check safety list, or by moving flights to rescheduled flights.



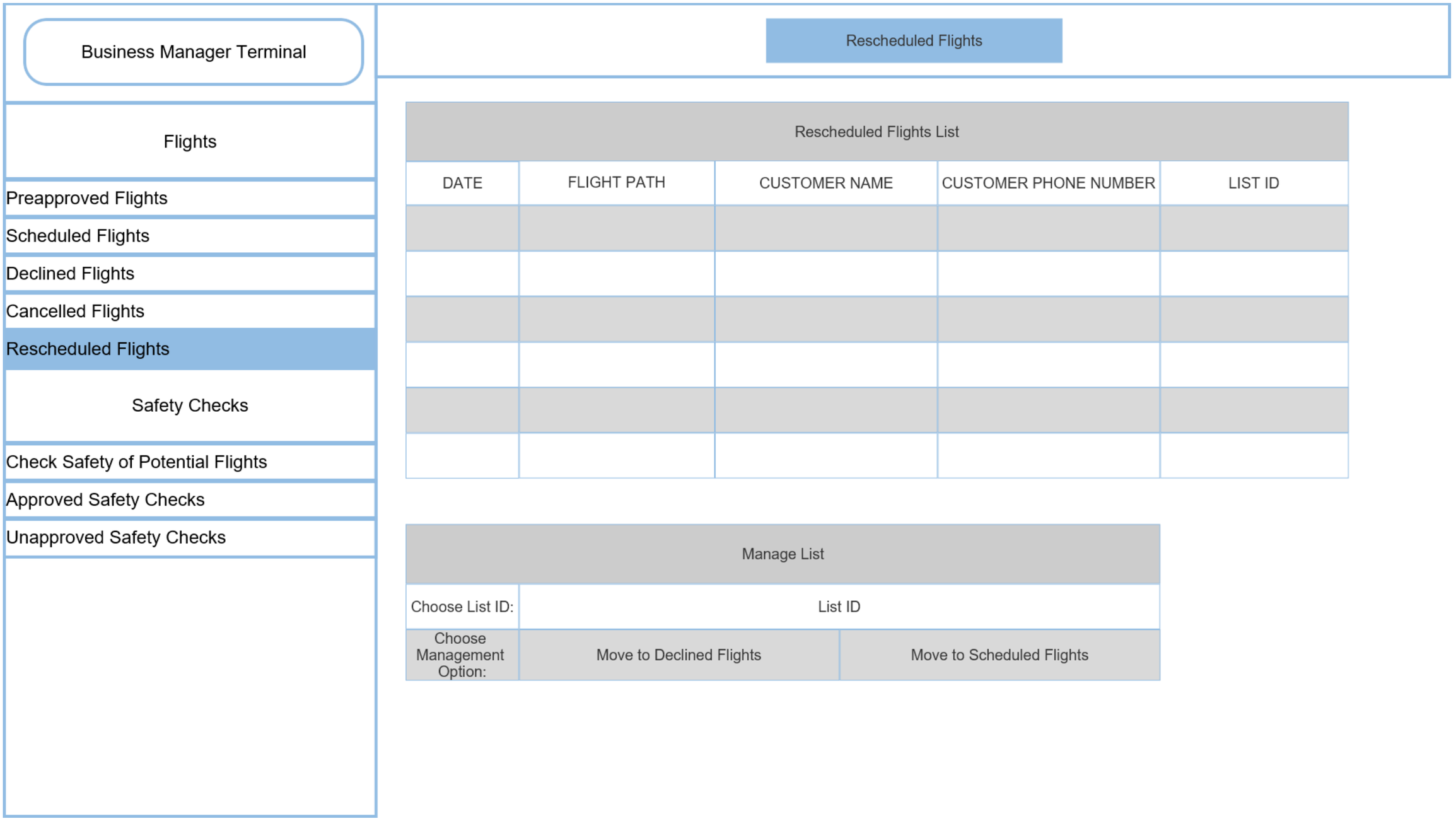
This page is the Scheduled flights list page showing the scheduled flights after approval of the safety check is given. These flights can be declined, cancelled or rescheduled by the user.



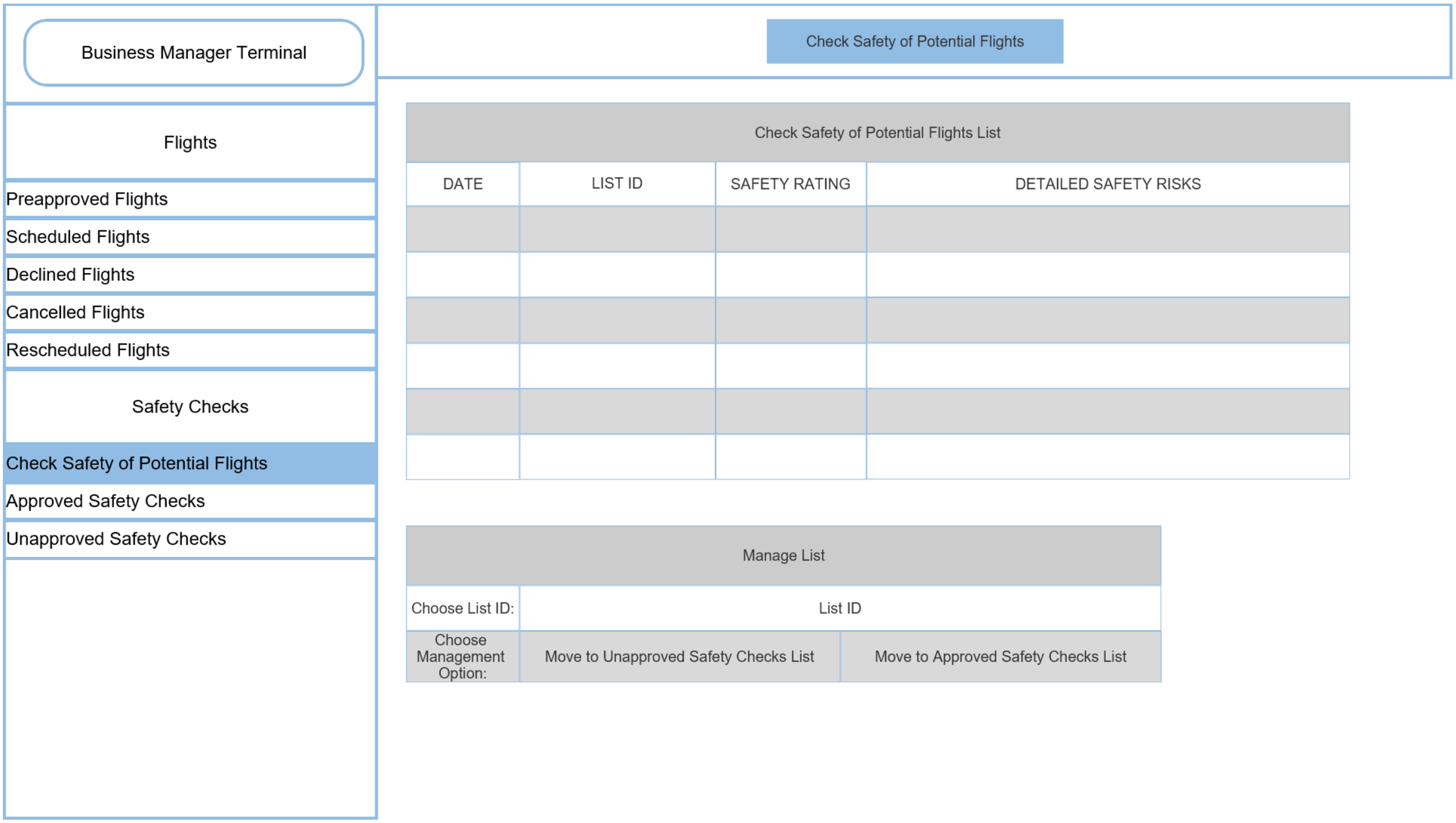
This is the Declined flights list which shows all flights that have been declined. The user can manage these flights by moving them to the cancelled flights or rescheduling the flight.



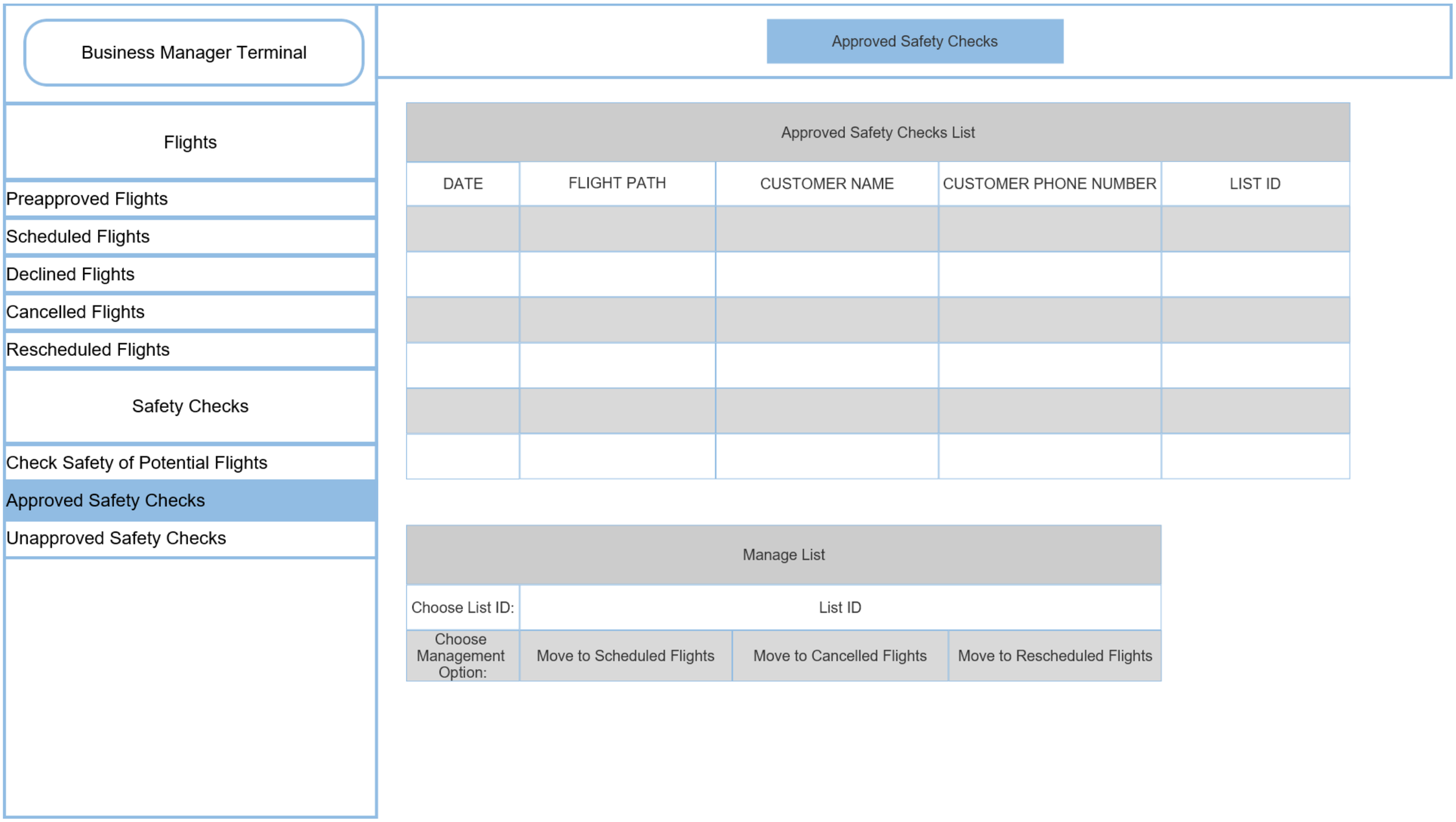
This is the Cancelled flights list which shows all flights that have been cancelled. The user can manage these flights by rescheduling the flight or deleting the data of that flight forever.



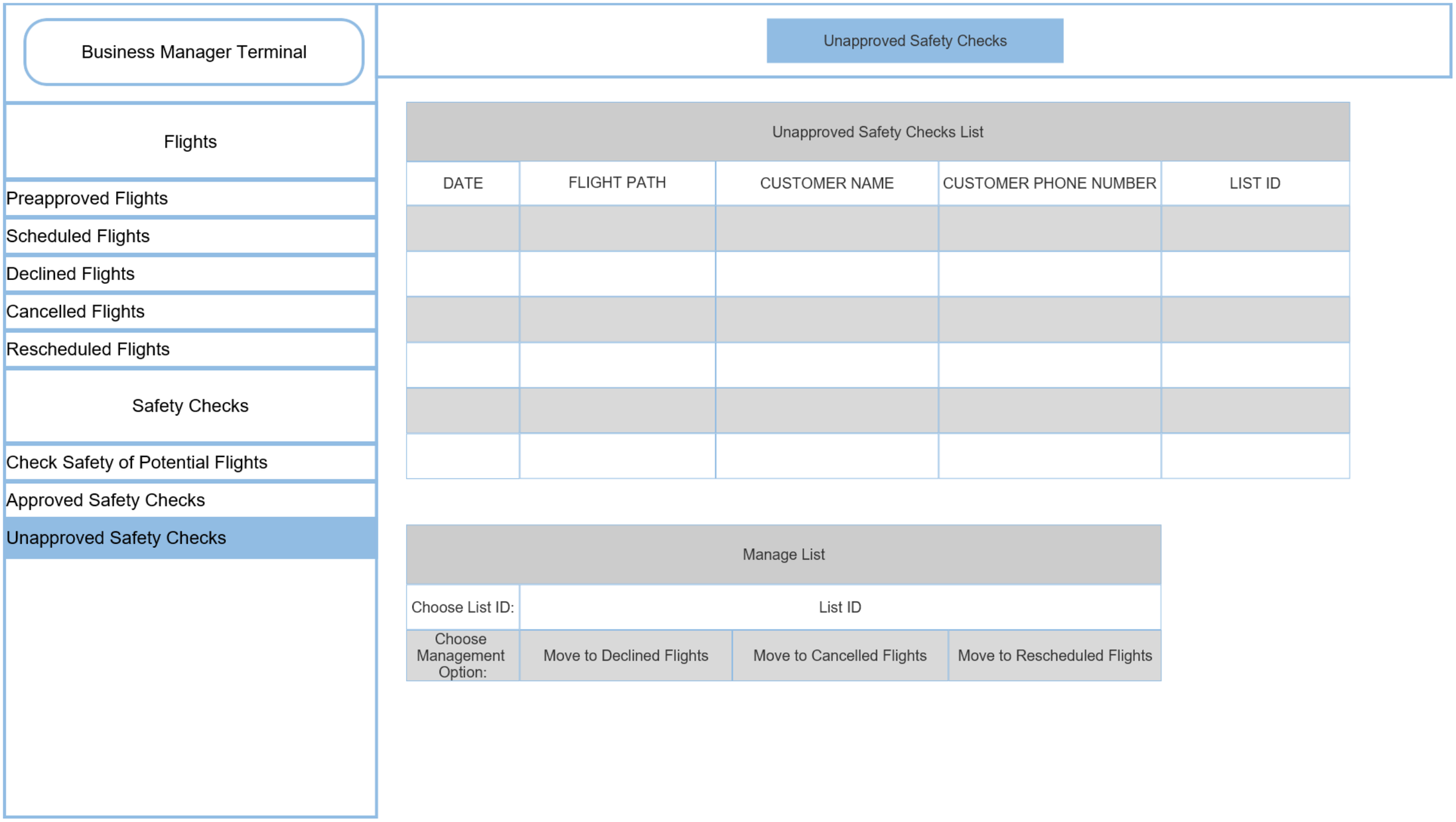
This is the Rescheduled flights list which shows all flights that have been rescheduled. The user can manage these flights by moving them to the declined flights or scheduled flights.



This is the Check safety of potential flights list which shows all flights that have been checked for safety before allowing the flight to be approved by a user. The user can manage these flights by moving them to Approved safety check flights or the unapproved safety check flights.

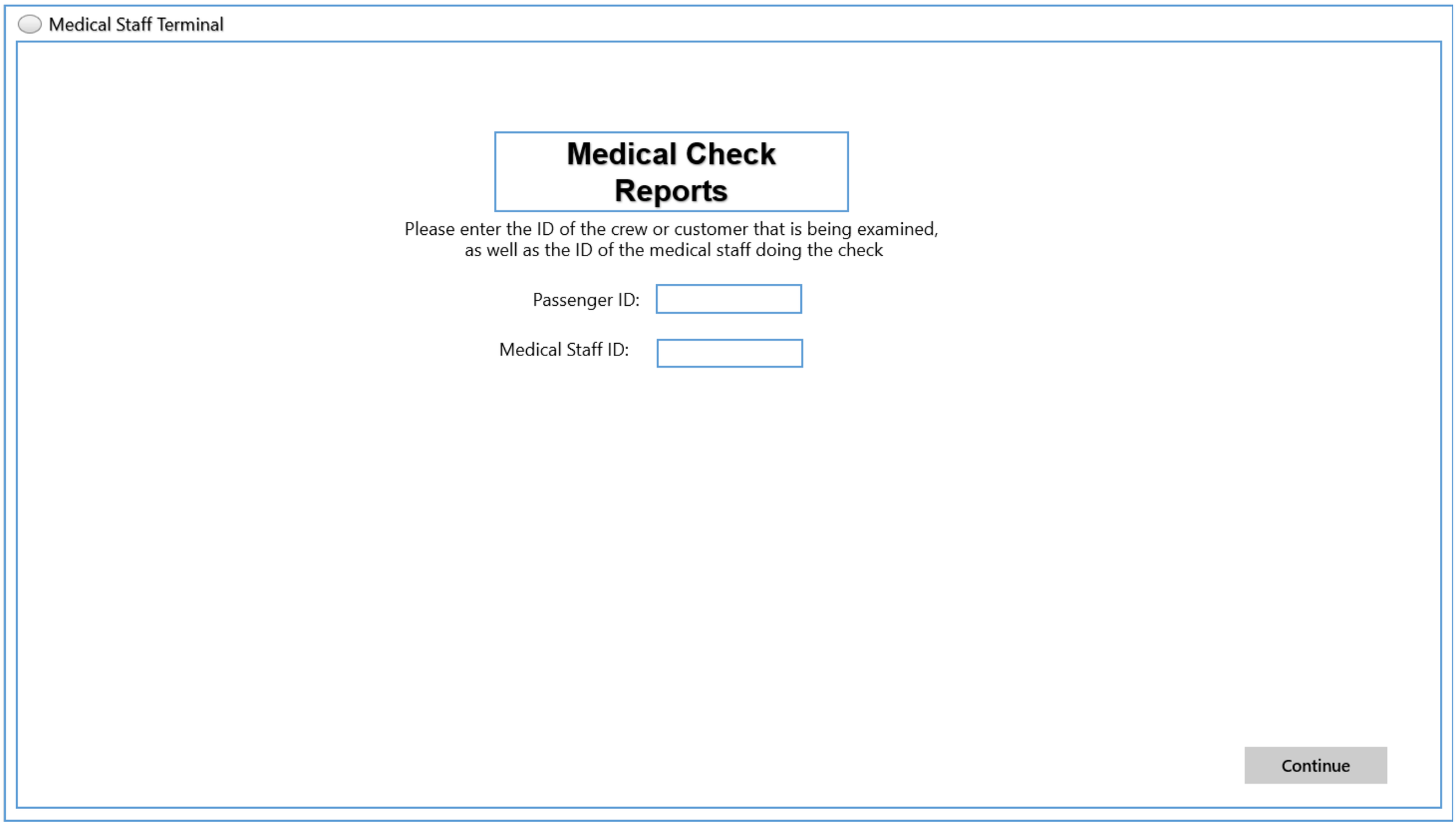


This is the approved safety check flights list which shows all flights that have been approved by the user after safety was checked. The user can manage these flights by moving them to the scheduled flights, cancelled flights or rescheduling the flight.

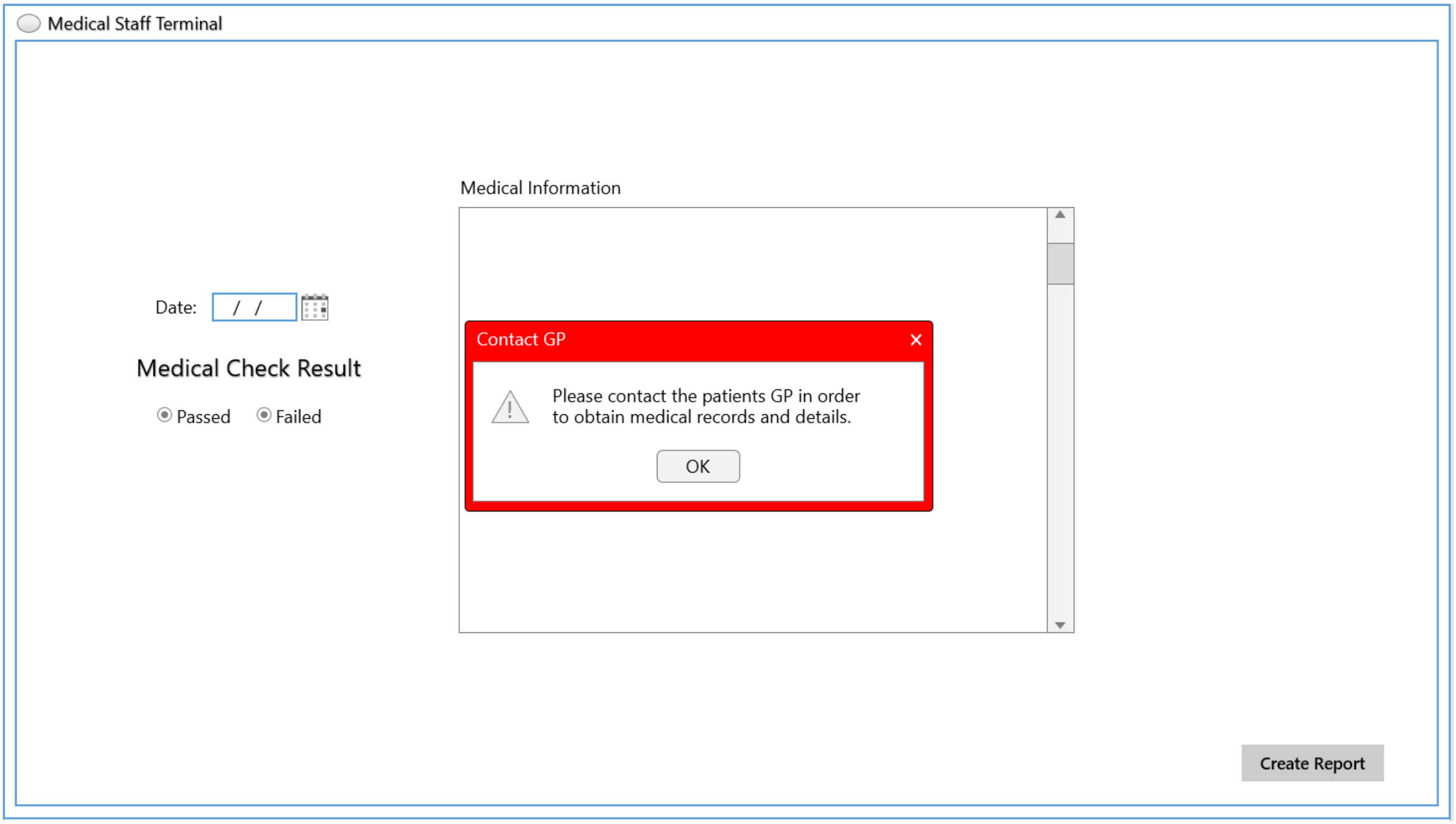


This is the unapproved safety check flights list which shows all flights that have been disapproved by the user after safety was checked. The user can manage these flights by moving them to the declined flights, cancelled flights or rescheduling the flight.

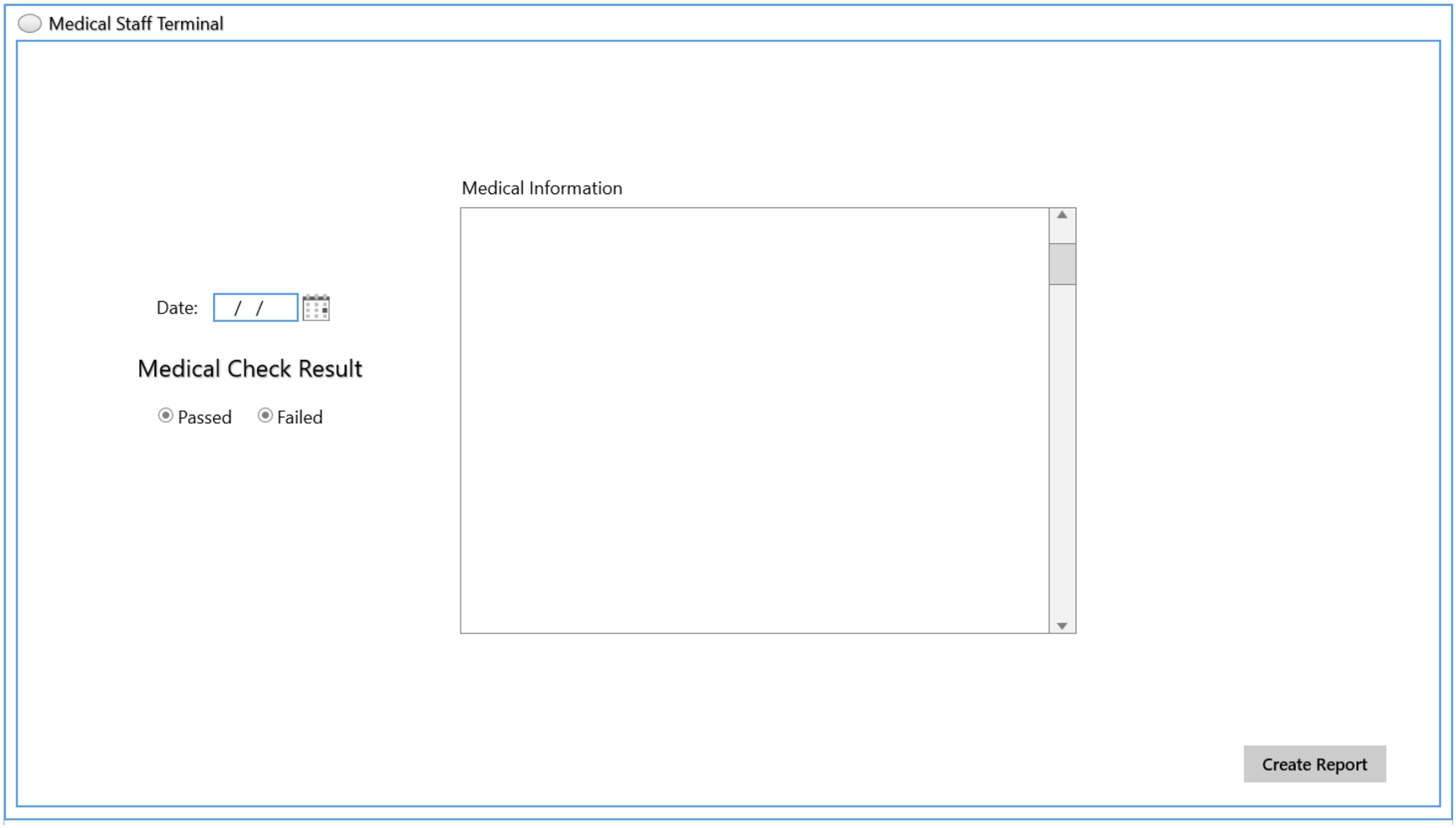
MEDICAL CHECK USER INTERFACE



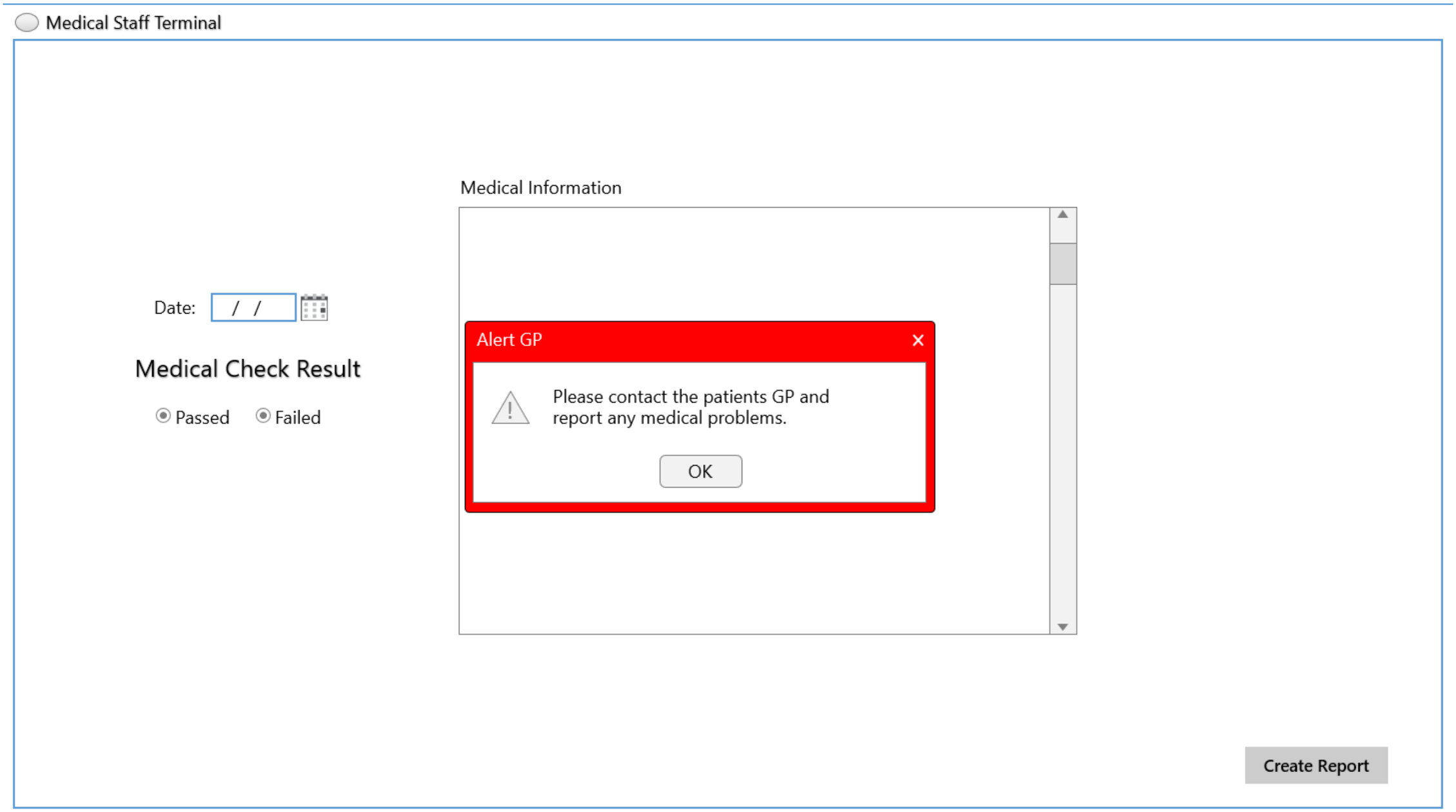
This is the initial screen within the terminal, where the medical staff may enter the ID of the passenger and the medical staff in order to continue to the creation of the medical report.



In this page the medical staff is being alerted to contact the GP so that they may acquire any details or records necessary for the medical check. It appears as the report creation page is opened.



This is the report creation page, it is used during and after the check, to enter the date of the check, and medical details they may need to be recorded, and the result of the check. Once the medical check is completed the report is created with the “Create Report” button.



Once the report is created if the check has failed for any reason, the medical staff is alerted to contact the passengers GP of any medical issues the passenger may be undergoing.

Deployment

The following section details the proposed deployment of StarStruck's online management system, and has been carefully crafted via thorough research and meticulous planning, as such that various issues have been accounted for, and each deployment approach has been thoroughly evaulauted in terms of their strengths and weaknesses.

Deployment Approach

While in traditional discussion of deployment approaches one might take, in consideration would be four of those approaches; Direct, Parallel, Phased, and Pilot. However, upon results from clear research, the team developing this report has decided to immediately disavow the Direct deployment strategy from consideration.

This is due to the fact that the nature of Direct deployment approach is incompatible with the mission-critical system being designed and developed, and as such will not be considered.

This leaves three options available, Parallel, Phased and Pilot. The following paragraphs outline each option and their advantages and disadvantages in relation to the approach we have choosen, in order to form a justification of this choice.

The Decision

While all three approaches are all valid and applicable, the most suitable deployment approach identified from this report is the Parallel approach. In comparsion with the other two approaches, utilising a parallel a approach would enable the software to thorough tested while ensuring that if any issues arise, the options to revert to the old system as a backup is still present. In addition, the transition to the new system would be smooth and nature, slowly enabling the old system to be replaced as the new system is tested and validated. This way, any critical issues which might arise from the new software can easily be mitigated via the old system still being operational.

While the Parallel approach is slighly complex than the Pilot approach, and it may be more costly having to operate both systems for a period of time, we believe that these are a small price to pay for being able to ensure that the online management system can be deployed smoothly without causing any major disruptions or risks to the business, as these could largely be eliminated through utilising the existing, established system already in place. Since this software is mission-critical, even the smallest disruption or risk could be catastrophic.

Addressing Issues

Now that we have established which deployment approach is ideal for the system, the following paragraphs will detail how common concerns are going to be addressed and mitigated / eliminated.

Data Migration

Data Migration from the old system to the new system will involve trickling data from the old to the new overtime. The migration will be broken down into smaller sub-migrations, where each sub-migration will be given a dedicated schedule, timeline and budget. This approach is ideal as it is commonly utilised when a Parallel deployment approach is involved.

User Training

Enduser training will be handled in incremental steps, with user's whose role would benefit the most from the new system, or whose role would be most impactful to the new system, being prioritised first for training. Subsections of departments would be trained at a time such that until the old system is completely phased out, there will be user's that still utilise the old system in case a fault or error occurs within the new one. The ultimate goal will be that by the time the old system is shut down for good, a majority of, if not all employees have received training in regards to the new system.

Mitigation of Faults

Our chosen deployment approach (Parallel) enables Mitigation of faults through taking advantage of the fact that the old system will still be operational for a period of time. This window of time allows plenty of opportunities to catch and correct any major errors and faults before the old system is removed entirely. This way, if the new system is comprised by a critical fault or risk, the old system can be utilised as a backup.

Team Management

TEAM MEETING #1

SENG2130 Systems Analysis and Design

Minutes of meeting

Team: The Acers Place: Discord Date/Time/03/05/22

In attendance

Brandon, Brock, Genis, Thomas, Zackary.

Apologies

Agenda

- **Matters arising from previous meeting**
  None at all.

- **Agenda items (as needed)**
  Discussing what architectural style we will be using.

- **Date, time and place for next meeting**
  We will be meeting again on discord on the 10th of May Saturday after our first lab back.

- **Matters for consideration at next meeting.**
  Nil.

Action sheet

Task	Responsible	Due	Notes
Create Crypto Sub System for the Class Diagram	Brandon	14/05/2022	
Create Medical Check Sub System for the Class Diagram	Zackary	14/05/2022	
Create Food Order Sub System for the Class Diagram	Genis	14/05/2022	
Create Space Flight Sequence Sub System for the Class Diagram	Thomas	14/05/2022	
Create Flight Booking Sequence Sub System for the Class Diagram	Brock	14/05/2022	

TEAM MEETING #2

SENG2130 Systems Analysis and Design

Minutes of meeting

Team: The Acers Place: Discord Date/Time: 10/05/22

In attendance

Brandon, Brock, Genis, Thomas, Zackary.

Apologies

Agenda

- **Matters arising from previous meeting**
  None at all.

- **Agenda items (as needed)**
  Reviewing our Class Diagram, Going through what needs to be done with the sequence diagrams.

- **Date, time and place for next meeting**
  We will be meeting again on discord on the 14th of May.

- **Matters for consideration at next meeting.**
  Nil.

Action sheet

Task	Responsible	Due	Notes
Create Crypto Sub System for the Class Diagram	Brandon	14/05/2022	
Create Medical Check Sub System for the Class Diagram	Zackary	14/05/2022	
Create Food Order Sub System for the Class Diagram	Genis	14/05/2022	
Create Space Flight Sequence Sub System for the Class Diagram	Thomas	14/05/2022	
Create Flight Booking Sequence Sub System for the Class Diagram	Brock	14/05/2022	

TEAM MEETING #3

SENG2130 Systems Analysis and Design

Minutes of meeting

Team: The Acers Place: Discord Date/Time: 14/05/22

In attendance

Brandon, Brock, Genis, Thomas, Zackary.

Apologies

Agenda

- **Matters arising from previous meeting**
  None at all.

- **Agenda items (as needed)**
  Discussing what needs to be done with the individual sequence diagrams.

- **Date, time and place for next meeting**
  We will be meeting again on discord on the 21st of May.

- **Matters for consideration at next meeting.**
  Nil.

Action sheet

Task	Responsible	Due	Notes
Create Crypto payment Sequence Diagram, Review Zackary's Sub System.	Brandon	25/05/2022	
Create Medical Check Sequence Diagram, Review Brandon's Sub System.	Zackary	25/05/2022	
Create Food Order Sequence Diagram, Review Brock's Sub System.	Genis	25/05/2022	
Create Space Flight Sequence Sequence Diagram, Review Genis's Sub System.	Thomas	25/05/2022	
Create Flight Booking Sequence Sequence Diagram, Review Thomas's Sub System.	Brock	25/05/2022	

TEAM MEETING #4

SENG2130 Systems Analysis and Design

Minutes of meeting

Team: The Acers Place: Discord Date/Time: 21/05/22

In attendance

Brandon, Brock, Genis, Thomas, Zackary.

Apologies

Agenda

- **Matters arising from previous meeting**
  None at all.

- **Agenda items (as needed)**
  Talking about what needs to go into the User Interface and deciding on a colour scheme for the UI.

- **Date, time and place for next meeting**
  We will be meeting again on discord on the 30th of May.

- **Matters for consideration at next meeting.**
  Nil.

Action sheet

Task	Responsible	Due	Notes
Create Crypto Payment User Interface, Review Zackary's Sequence Diagram.	Brandon	30/05/2022	
Create Crypto Payment User Interface, Review Brandon's Sequence Diagram.	Zackary	30/05/2022	
Create Crypto Payment User Interface, Review Brocks's Sequence Diagram.	Genis	30/05/2022	
Create Crypto Payment User Interface, Review Genis's Sequence Diagram.	Thomas	30/05/2022	
Create Crypto Payment User Interface, Review	Brock	30/05/2022	

Thomas's Sequence Diagram.			

TEAM MEETING #5

SENG2130 Systems Analysis and Design

Minutes of meeting

Team: The Acers Place: Discord Date/Time: 30/05/22

In attendance

Brandon, Brock, Genis, Thomas, Zackary.

Apologies

Agenda

- **Matters arising from previous meeting**
  None at all.

- **Agenda items (as needed)**
  Going through our individual work and start to move all work into a report.

- **Date, time and place for next meeting**
  This is anticipated to be our last formal meeting; we may have another closer to the due date.

- **Matters for consideration at next meeting.**
  Nil.

Action sheet

Task	Responsible	Due	Notes
Review all work completed.	Brandon	N/A	
Review all work completed.	Zackary	N/A	
Review all work completed.	Genis	N/A	
Review all work completed.	Thomas	N/A	
Review all work completed.	Brock	N/A	

TEAM MEETING #6

SENG2130 Systems Analysis and Design

Minutes of meeting

Team: The Acers Place: Discord Date/Time: 03/06/22

In attendance

Brandon, Brock, Genis, Thomas, Zackary.

Apologies

Agenda

- **Matters arising from previous meeting**
  None at all.

- **Agenda items (as needed)**
  Putting all work into the report.

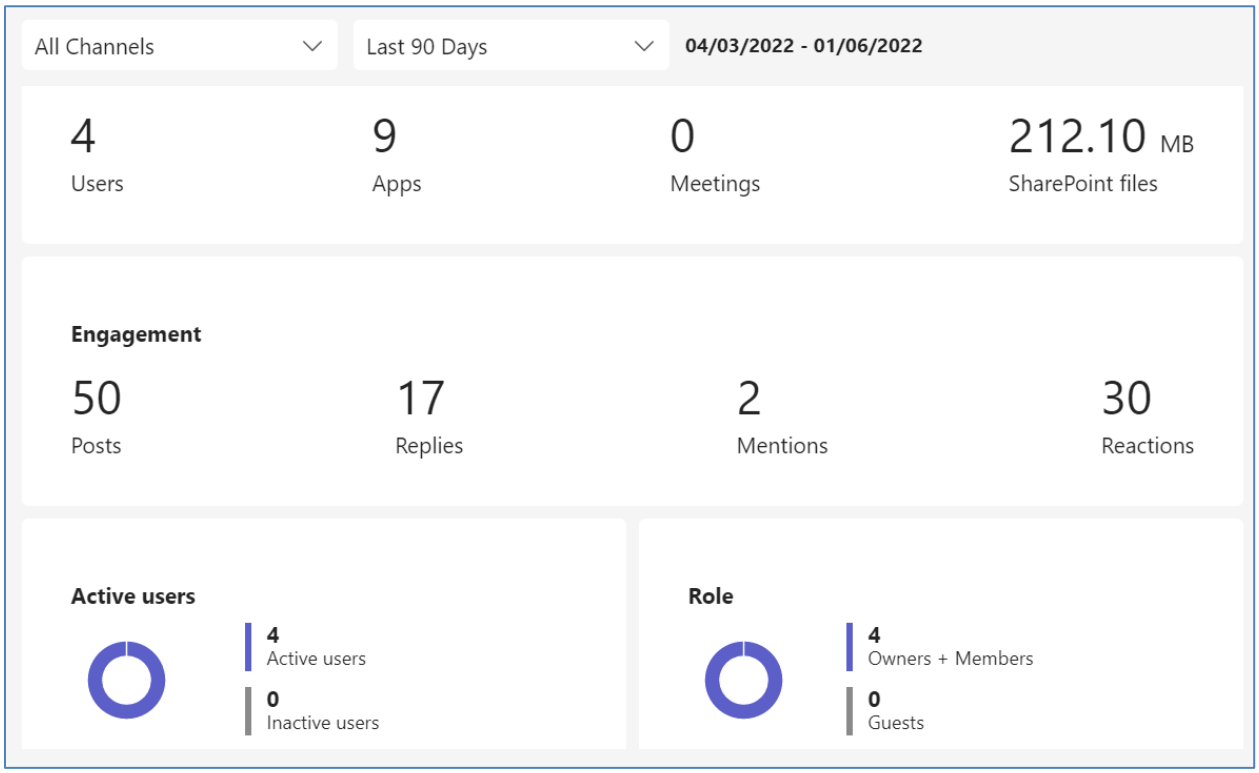
- **Date, time and place for next meeting**
  N/A Will be finalising the report today.

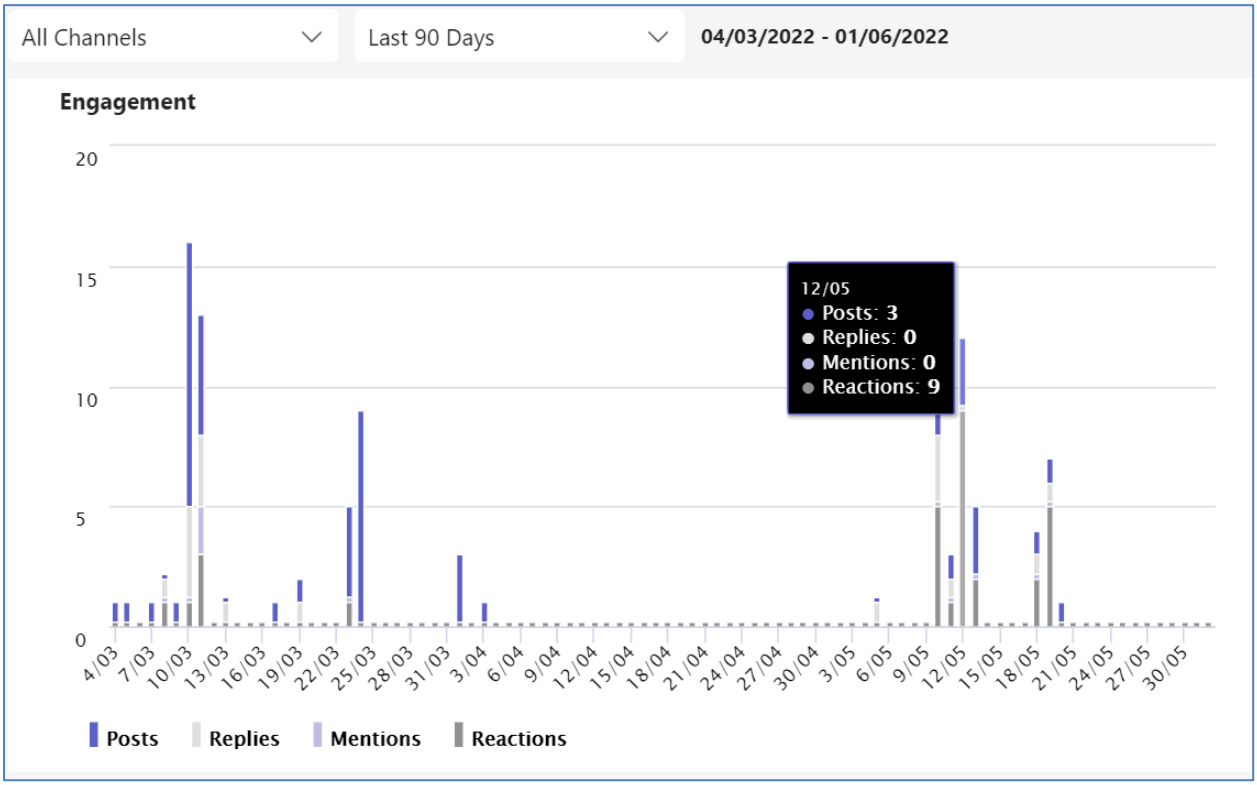
- **Matters for consideration at next meeting.**
  Nil.

Action sheet

Task	Responsible	Due	Notes
Work on Final Report.	Brandon	03/06/22	
Work on Final Report.	Zackary	03/06/22	
Work on Final Report.	Genis	03/06/22	
Work on Final Report.	Thomas	03/06/22	
Work on Final Report.	Brock	03/06/22	

MS TEAMS ANALYTICS





Conclusion

Ultimately, this report has completed a detailed and thorough analysis of StarStruck's theoretrical online system. The foundations for which originated in the first report, have been expanded upon, itching ever closer from theory to reality. Below is a brief summary of the reports findings, what was completed, what wasn't, and our recommendations for StarStruck's next steps.

What was completed

Ultimately, this report completed all of our objectives, specific subsystems of system were expanded upon into new forms, specifically via class diagram's and sequence diagrams. These two additions enable greater flexibility, but also ensuring the system meets it's necessary functional requirements. From these diagrams came Prototype User Interfaces for each subsystem, in which multiple screens were designed with user experience and interaction in mind.

Recommendations

Finally, our recommendations for StarStruck. Using the findings from this report and the last, we believe StarStruck is in a prime position to greenlight an implementation of this modern online management system, which would enable StarStruck's as a business to operate with greater efficiency, while also paving the way for other space faring enterprises. We at least hope that the executive of StarStruck will take the results of this report to heart, and understand the many benefits of upgrading, and ultimately improving their management system.

References

Space (Launches and Returns) (High Power Rocket) Rules 2019. (2019). Legislation.gov.au; Attorney-General's Department. https://www.legislation.gov.au/Details/F2019L01119/Explanatory%20Statement/Text

FLIGHT SAFETY CODE. (n.d.). https://www.industry.gov.au/sites/default/files/2019-08/space-flight-safety-code.pdf

Industry. (n.d.). *Space (Launches and Returns) (General) Rules 2019*. **Error! Hyperlink reference not valid.**. https://www.legislation.gov.au/Details/F2019L01118

An Introduction to Database Migration Strategy and Best Practices
Strim.com
https://www.striim.com/blog/an-introduction-to-database-migration-strategy-and-best-practices

Data Migration Guide: Strategy Success & Best Practice (July 2021)
Varonis
https://www.varonis.com/blog/data-migration