REYKJAVÍK UNIVERSITY

Semester Project Course 1

Design Brief

Andrea H. Reykjalín Jónsdóttir
andrear23@ru.is
Anton Elvar Þórðarson
antont22@ru.is
Hera Brá Tómasdóttir
hera23@ru.is
Marvin Sigrúnarson Jónasson
marvin23@ru.is
Viggó Ýmir Hafliðason
viggoh21@ru.is

Teacher:
Gylfi Þór Guðmundsson
December 15, 2023

Contents

1 Introduction

After Chuck Norris' unambiguously godly career in media, he set his sights on the transportation industry with his new company, NaN.(New Awesome Norris) Air.
Our programming group known colloquially as "Group 40", was blessed when, since Chuck Norris doesn't ask things of people, his assistant rang us up to write this program in Mr. Norris' stead for his new "skybreaking" company.

Adhering to the previously constructed narrative, we started our brainstorming session and decided on a three-tiered encapsulating design philosophy which we will explain in section 3. We chose this design philosophy because of its scalability and modularity.

In this project we have finished all of the A requirements and have done some of the B requirements. In the requirements list we have marked the requirements that we didn't finish with a red color. Because we have finished all of the A requirements, it ensures the basic functionality of our program works as intended and is ready for use by the NaN air employees. We struggled with many different things and overcame many challenges over the whole of the development process.

Figure 1: Our boss and ruler, Chuck Norris.

2 Use Cases

Here after are tables describing what our system has to be able to do.
These use cases below try to cover a wide range of functionalities that are essential for the NaN-Air booking system.

Name	Register employee
Number	1
Priority	A
Precondition	Know how to read
Description(base flow)	The Crew manager wants to register a new employee, he presses "2. Employees", then he presses "2. Register new employee". He then selects what type of employee he would like to register and inputs the correct information about the employee; name, phone number (home phone optional), SSID, Email, Rank, License, Employee ID, address. The Crew manager inputs “save” to save changes.
Alternative flow	The user accidentally pressed "1." instead of "2." and got into voyages. The user has to press "b. Back" to get back to then select the right option.
Post condition	User has registered a new employee that is written into the software database.
Source (Requirements)	1. 2.
Actors	Crew Manager
Authors	Marvin

Figure 2: Slide 1: Register employee

Name	Register new voyage
Number	2
Priority	A
Precondition	None
Description(base flow)	The flight manager selects “1.” for voyages in the main menu. He selects “2.” register new voyage the user enters the UUID and then enter the destination then the departure time and departure date then arrival time and then arrival date then they should enter the crew (captain, copilot, flight service manager and flight attendant) if they leave it empty then you can man the voyage later
Alternative flow	• The selected date/dates are invalid
Post condition	New Voyage Registered.
Source (Requirements)	3. 13. 20. 22. 31
Actors	The Flight Manager
Authors	Marvin

Figure 3: Slide 2: Register new voyage

Name	Register captain for a registered voyage
Number	3
Priority	A
Precondition	A voyage is registered with a selected airplane but without a crew.
Description(base flow)	The Crew Manager selects “1.” for voyages in the main menu. He selects “5.” to choose staff from there the user can select 1. Add Captain 2. Add Co-Pilot 3. Add Head of Service 4. Add Flight Attendant From there the user selects 1 to register a captain for a voyage, then it prints out a list of all captains, and there the user inputs the name of the captain that we want to assign, then the program gives us a success message
Alternative flow	• A voyage has not been registered • No pilots nor crew members are registered
Post condition	Successfully set captain {captain name} to voyage {voyage id}
Source (Requirements)	4. 18. 30. 32. 33. 34.
Actors	The Crew Manager
Authors	Marvin

Figure 4: Slide 3: Man registered voyage

Name	Copy existing voyage
Number	4
Priority	A
Precondition	None
Description(base flow)	The user “1.” to enter voyages in the main menu. User selects “3.” to copy existing voyage. From there the user has to enter a Voyage ID and from there the user has to enter the new date that they want to copy the voyage to. And then the program prints a success message and a list of all the voyages
Alternative flow	• No voyage has been registered to copy (see use case 2: register a new voyage)
Post condition	Voyage copied.
Source (Requirements)	5. 6. 14.
Actors	The Flight Manager
Authors	Hera

Figure 5: Slide 4: Copy existing voyage

Name	Update employee information (update Flight attendant)
Number	5
Priority	A
Precondition	An employe is registered
Description(base flow)	The user selects “2.” for employees in the main menu. He selects “3.” to update employee and from there the user selects if they want to update pilot or update flight attendant from there the user enters the SSID of the employee that they want to update, from there they enter all of the new info. If the user does not want to change some information then they should leave it empty as, we convey this with a message before. After entering the new information the program gives us a success message and we go back to the manage employee menu.
Alternative flow	No employees have been registered (see. Use case 1: register employee)
Post condition	Employee has been updated!
Source (Requirements)	9. 10. 11.
Actors	Crew Manager
Authors	Marvin

Figure 6: Slide 5: Change employee information

Name	Display all pilots
Number	6
Priority	A
Description(base flow)	The Flight manager wants to see a list of all pilots that are registered at the company. So he selects "2." for employees and then "1." for Display Employees from there the user can select "1." For search by: and then the user can enter the search filter in this case the search filter is Pilot and then it prints a list of all pilots
Alternative flow	The Flight manager wants to see a list of all pilots that are registered at the company. So he selects "2." for employees and then "1." for Display Employees from there the user can select "2." For sort by: and then can select sort by pilots
Post condition	A list of all pilots is displayed.
Source (Requirements)	7. 8.
Actors	Crew manager
Authors	Marvin

Figure 7: Display all pilots

Name	Register Aircraft
Number	7
Priority	B
Precondition	An unregistered aircraft needs registration.
Description(base flow)	The Flight manager wants to register a new aircraft. He presses "4." for Aircrafts, then he selects "2." Register Aircraft. From there he proceeds to enter the airplane insignia in the format (XX-XXX) and from there the user selects the airplane type 1. BAE146 2. FokkerF28 3. FokkerF100
Alternative flow	The manager presses "1." in the Aircrafts menu and goes into display airplanes He has to go back and select the right option.
Post condition	An aircraft has been registered into the system
Source (Requirements)	23.
Actors	Flight Manager
Authors	Anton Elvar Þórðarson

Figure 8: Register aircraft

Name	Get list of all employees not working on a selected date
Number	8
Priority	A
Precondition	An employee and shift is registered
Description(base flow)	The user selects “2.” for employees in the main menu. He selects “4.” for the shift plan, that prints out the whole shift plan and from there the user can select to search by if an employee is or isn't working on a specific date, in this case the user selects "2." For not working on specific day and then the user enters the day they want to search for and then the program prints a list of the employees and their shifts that aren't working on that day.
Alternative flow	• No employees are registered • All employees are working on the selected date(if this happens it will be an empty list)
Post condition	List of all employees not working on selected date has been displayed
Source (Requirements)	16. 18.
Actors	Crew manager
Authors	Marvin

Figure 9: Get list of all employees not working on a selected date

Name	Register pilot license
Number	9
Priority	B
Precondition	Know how to read
Description(base flow)	The Crew manager wants to register a pilot license, so he selects "2." Employees in the starting menu. Then he presses "5." to register new employee. He is then given the option between pilots, flight attendant and heads of service, he pick pilots. In pilots he needs to fill in the information that is required for the pilot. In that information is his license the manager fills it out.
Alternative flow	Instead of going in to Register New employee, the manager goes to "2." Edit pilots (in Air Employees) and changes the license from there.
Post condition	A Pilot license has been registered
Source (Requirements)	23.
Actors	Crew Manager
Authors	Anton Elvar Þórðarson

Figure 10: Register pilot license

Name	Print a summary that shows all the employee's work trips in a certain week
Number	10
Priority	A
Precondition	None
Description(base flow)	The crew manager wants to print out a certain week that a specific employee is working a voyage he selects "1." For manage voyages and from there he selects "7." For Check Voyages an Employee is Working. Then the user has to input the name of the employee and the start date to search for and then the program prints a list of the voyages that the employee is working.
Alternative flow	The user goes to employees tab instead of voyages and doesn't find where to check voyages an employee is working
Post condition	A list of the voyages that the employee is working in a specific week
Source (Requirements)	17.
Actors	Crew Manager
Authors	Marvin

Figure 11: Print summary that shows employees voyages in a certain week

Name	Register a new destination
Number	11
Priority	A
Precondition	None
Description(base flow)	The user selects “3.” for destinations in the main menu. He then selects “2.” to register a new destination. He enters the destination name, country, airport, flight time, distance from Iceland, the name for the destinations contact and the emergency phone number. He inputs ”save” to save changes.
Alternative flow	• The destination already exists
Post condition	A new destination has been registered
Source (Requirements)	12.
Actors	The Flight Manager
Authors	Marvin

Figure 12: Register a new destination

Name	Display All Voyages
Number	12
Priority	A
Description(base flow)	The Flight manager wants to see all voyages, so he selects "1." Voyages in the starting menu. Then he presses "1." to display voyages. Then the program displays a list of all voyages.
Alternative flow	
Post condition	User is showed a list of the voyages they wanted to see.
Source (Requirements)	33.
Actors	Flight Manager
Authors	Marvin

Figure 13: Display all voyages

Name	Change emergency contacts phone number for given destination
Number	13
Priority	B
Precondition	A destination exists and has an emergency contact
Description(base flow)	The user selects “3.” for Manage destination in the main menu. He selects “4.” for update destination, then the user enters the name of the destination that they would like to update, then They are prompted to enter the new updated contact name and contact phone number.
Alternative flow	• No destination has been registered (see use case 6: register a new destination)
Post condition	The emergency number for the destination has been updated
Source (Requirements)	26. 27.
Actors	User
Authors	Marvin

Figure 14: Change emergency contact’s phone number for a given destination

Name	Display pilots by plane type
Number	14
Priority	B
Precondition	There needs to at least be a single registered Pilot
Description(base flow)	The Flight manager wants to see a list of all the pilots that have a license for a specific plane. He sits down at the computer and opens our booking software. He presses "2. Manage Employees", from there he selects "1. Display Employees" and there the program displays all employees and from there the user selects "1. Search by:" and there the user can enter the license that they want to search for and see all pilots that have that license.
Alternative flow	The company has not registered what license the pilot has so no pilot shows up for any plane type
Post condition	Shows a list of all pilots that have a license for a specific plane
Source (Requirements)	19. 25.
Actors	Flight Manager
Authors	Marvin

Figure 15: Display pilots by plane type

3 Class Diagram

When writing our class diagram, we adhered to the three-tiered encapsulation design we decided on. This means that all classes are only able to communicate "down" tier layers through wrapper classes, the only exception to this is when the data layer functions create and write files/modules.

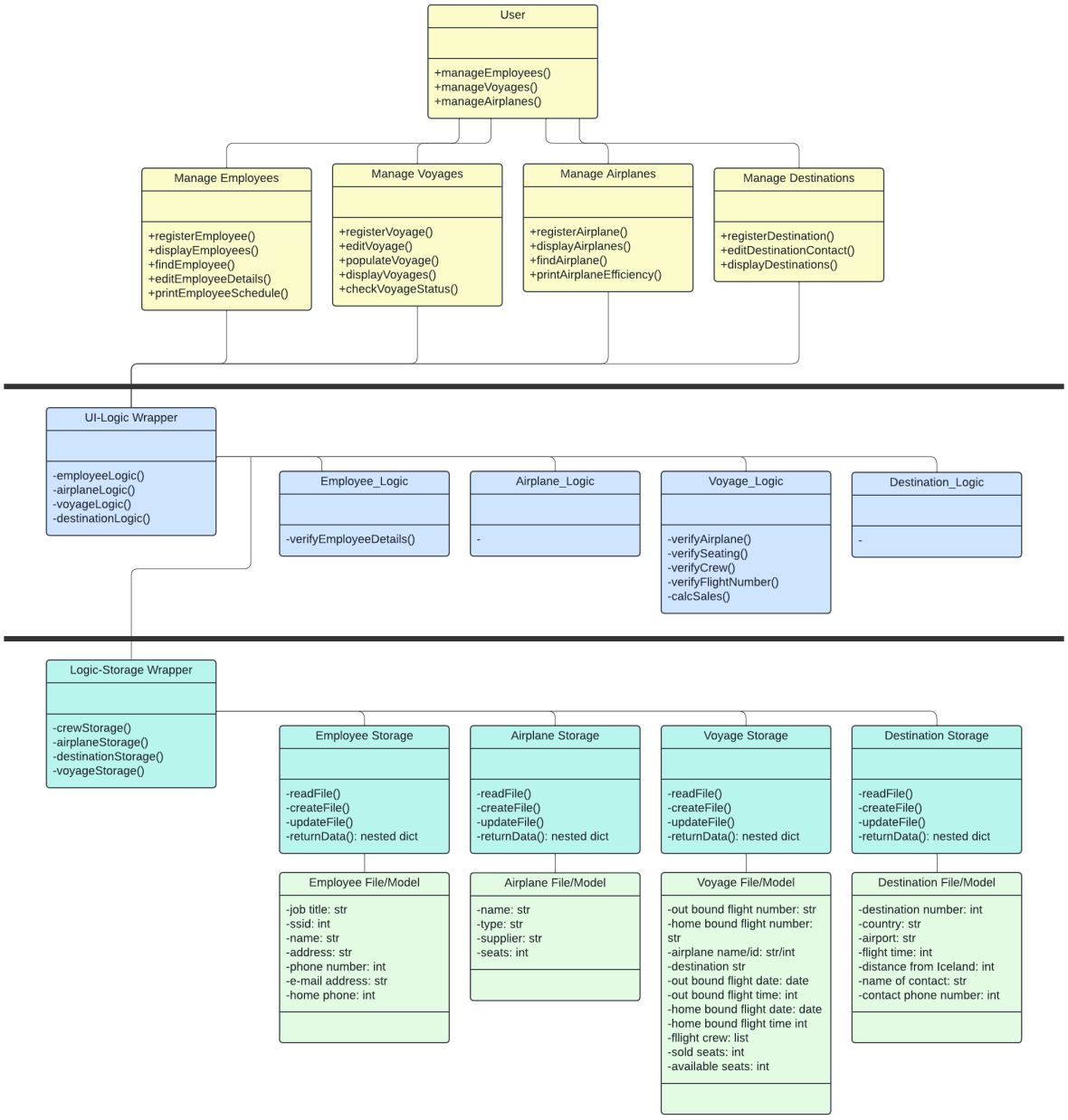

Figure 16: Three-layer communication class diagram

3.1 UI Layer

Since the user interface layer will be created with a text-based UI in mind, it should be programmed to read the user's inputs and call the Logic wrapper class with the corresponding command.
In our class diagram, the user interface layer consists of 1 main class that calls 5 other classes: Manage Employees, Manage Flights, Register Flights, Manage Airplanes, and Manage Destinations.

The Manage Employees class was created to house all the users' employee management commands. That would include commands such as: register employee, display employees, update employee details, show the shift plan and filter employees .

In the Manage Voyages class, users can display a voyage, register a new voyage, copy existing voyage, make recurring voyage, choose staff, check voyage status and check voyages an employee is working.
Manage Airplanes lets users register Airplanes currently owned by NaN-Air and display all airplanes owned by NaN-Air
The final class, Manage Destinations, allows a user to display all destinations, register a new destination, edit a certain destination's emergency contact information, find a destination.

3.2 Logic Wrapper

This "wrapper" is meant to call the logic layer's classes for the UI layer. This causes both layers to be detached from one another, meaning that each layer can exist individually from the other. This is the bread and butter of three-layered encapsulation programming design. Essentially this allows programmers to patchwork their software with new layers, swapping them out for the previous iteration without affecting other layers at all.

3.3 Logic Layer

This will most likely be the largest layer as it will house all of the program's logical functions.
That means that any calculations or verifications happen in this layer.
We expect to have the logic functions verify: register airplane, update pilot, and most other data inputs from the user.
Here, each logic class or function is meant for a certain task which it completes when called upon by the UI-Logic wrapper.

3.4 Data Wrapper

This wrapper's functionality is almost the same as the previous ones, except it communicates between the logic layer and data layer, returning the previously registered data to the logic layer for computation and taking data from the logic layer to write into files/models.

3.5 Data Layer

This layer is the only one that has anything to do with data, its function is to write data that is inserted into the program into files or other data storing modules, read said data, and send it on its way to the logic layer where it will be used to verify inserted data.
Each class/function may structure its data differently, but since no inter-communication will be happening in this layer, it won't have any effect on the software's logistical function.

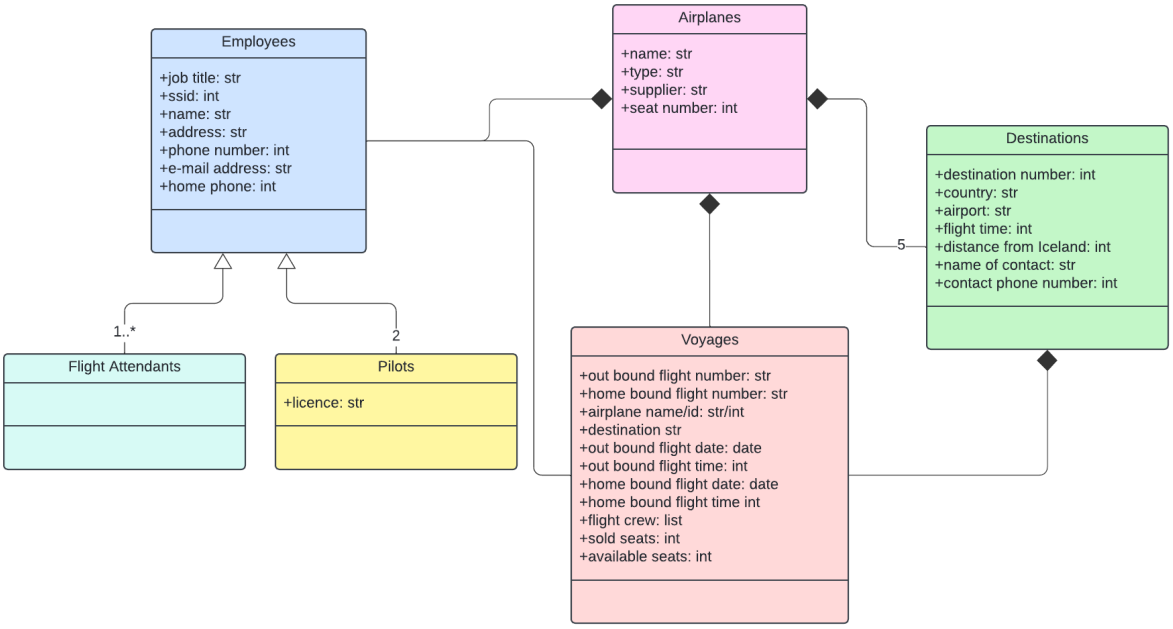

Figure 17: Internal Class Functionality Diagram

4 User Group Analysis

We have two user groups that would be using our system, the groups are crew manager (mannauðsstjóri) and flight manager (skipulagsstjóri).

Both individuals are 20 to 65 years old and have at least a college education and moderate computer skills.
They both work for about 8-10 hours each day, they seem very similar but what differentiates them is that the flight manager focuses on managing and controlling the booking system while the crew manager focuses on overseeing the pilots and flight attendance.

NAME of group	**Flight manager (Skipulagsstjóri)**	**Crew manager (Mannauðsstjóri)**
WHO background	**Age:** 20 - 65	**Age:** 20 - 65
	Gender: All genders	**Gender: All genders**
	Education: at least college education	**Education: at least college education**
	Abilities/Disabilities: The ability to know how to type and plan.	**Abilities/Disabilities: The ability to know how to man jobs.**
	Computer skills: Moderate	**Computer skills: Moderate**
	Number: At least 2	**Number: at least 1**
WHY main goals	To manage and control the system that orginizes the booking system. To have a complete control over all the bookings	To manage the pilots and flight attendence for the flights. Who will work on certain shifts and who is not on a shift. Overall role is a shift-manager.
WHAT equipment	Computer Phone White board Tablets	Computer Phone White board Tablets
WHERE environment	In the Office / At workplace	In the Office / At workplace
WHEN usage of system	**How often: Every day**	**How often: Every day**
	For how long each time: Cirka 8 - 10 hours	**For how long each time:** Cirka 8 - 10 hours
	Skills: computer skills and managment skills	**Skills:** computer skills and managment skills
HOW Important	Very Important	Very Important

Figure 18: User Group Analysis

5 UI Design and Happy Paths

This link directs to a Stately website where you can press the simulate in the top right corner of the website to show the path of each menu. Link to wireframe

this link is to the docs with the wireframe and happy path Link to docs

6 Requirements List

6.1 Functional Requirements

Table 1: Functional Requirements Table

#	Requirement/Description	User Group	Priority	Additional Information
1.	User can register a flight attendant	Flight Manager	A	Employee's SSID, Name, Home Address, Telephone Number, E-mail Address, Home Telephone(Optional), Rank, License
2.	User can register a pilot	Flight Manager	A	Employee's SSID, Name, Home Address, Telephone Number, E-mail Address, Home Telephone(Optional), Rank, License
3.	User can register a voyage	Flight Manager	A	Destination Date of departure from Iceland, Time of departure from Iceland, Date of departure to Iceland, Time of departure to Iceland, Airplane
4.	User can add staffing information for a registered voyage	Crew Manager	A	At least two pilots one of which is designated as the captain, At least one flight attendant, Employees are forbidden to work more than one shift per day, Both pilots have to have the required license to fly the plane
5.	User must be able to copy a work trip registration for the same destination and at the same time over multiple days	Crew & Flight Manager	A	
6.	User must be able to copy the registration of a voyage for the same destination and at the same time as an event that occurs at regular intervals	Crew & Flight Manager	A	
7.	The system should be able to display a list of all pilots	Crew & Flight Manager	A	
8.	The system should be able to display all flight attendants	Crew & Flight Manager	A	
9.	The system should be able to display a list of all employees	Crew & Flight Manager	A	
10.	The system should be able to display information about a specific employee	Crew & Flight Manager	A	Employee SSID, Name, Home Address, Telephone Number, E-mail Address, Home Telephone(optional), Rank, License, Employee ID
11.	User should be able to change information about an employee	Crew Manager	A	User should not be able to change employees' SSID or Name
12.	User can register a destination	Crew & Flight Manager	A	Country, Airport, Flight Time, Distance from Iceland, Name for Emergency Contact, Phone Number of Emergency Contact

13.	The system should be able to display a list of all destinations	Crew & Flight Manager	A	
14.	The system should be able to display a list of all employees who are not working on a specific day	Crew & Flight Manager	A	
15.	The system should be able to display a list of all employees who are working a specific day	Crew & Flight Manager	A	
16.	The system should be able to display a printable summary that shows all voyages of an employee for a specific week	Crew & Flight Manager	A	
17.	The system should be able to display a list of all voyages for a selected date/week	Crew & Flight Manager	B	List should also show whether each voyage is fully manned or not
18.	The system should be able to keep track of which pilots are allowed to fly which aircraft	Crew & Flight Manager	B	
19.	The system should be able to display a list of all pilots that have a license for a specific plane type	Crew & Flight Manager	B	
20.	User should be able to select what airplane is used for voyage	Crew & Flight Manager	B	
21.	User should be able to enter the number of sold seats for each flight	Crew & Flight Manager	B	
22.	The system should be able to assign a flight number to each flight	Crew & Flight Manager	B	Each flight number starts with "NA" and is followed by the 2-digit number assigned to the flight destination, followed by another 2-digit number that is even or odd, dependant upon if the flight is flying to or from Iceland
23.	User should be able to register a plane	Crew & Flight Manager	B	Name, Type, Manufacturer, Number of passenger seats
24.	User should be able to register what license a pilot has for what plane	Crew & Flight Manager	B	
25.	The system should be able to display a list of all pilots by plane/license type	Crew & Flight Manager	B	
26.	User can change the name of a contact that is registered to a destination	Crew & Flight Manager	B	
27.	User can change the emergency number that is registered to a destination's contact	Crew & Flight Manager	B	

28.	The system should be able to display a list of all aircrafts owned by the airline and show their status on a specific date and time.	Crew & Flight Manager	B	
29.	The system should be able to display a list of all employees who are not working a specific date	Crew & Flight Manager	B	
30.	The system should be able to display a list of all airplanes that not in use for a specific date	Crew & Flight Manager	B	
31.	The system should be able to display a list of all pilots who are not working specific date and have the required license	Crew & Flight Manager	B	
32.	The system should be able to display a list of all unmanned voyages	Crew & Flight Manager	B	
33.	The system should be able to show the status of voyages	Crew & Flight Manager	B	
34.	The system should be able to keep track of which employee is working each day	Crew & Flight Manager	B	
35.	The system should be able to keep track of what voyages are manned and unmanned	Crew & Flight Manager	B	

Here above the requirements that we marked red are the ones that we didn't finish

6.2 Requirements Justifications

6.3 Usability Requirements

The Basics	Each and every user of this software has to: be able to read, be able to write, know how to operate a keyboard, be able to operate a keyboard, have access to this software.
Effective to use	Our software has to be good at doing what it was made for.
Efficient to use	What does the software do to help the user complete tasks? Our software has to help the user complete their tasks as efficiently and easily as possible.
Safe to use	Our software needs to protect the user from making grievous mistakes that would harm the company and its data or the user. It should also hinder the user so as to make it more difficult for those kinds of situations to occur.
Having good utility	Our software should allow the user to easily complete all of their differing tasks in an easy manner.
Easy to learn	Our software should be intuitive and transparent in its usage, and thus, easy to learn.
Easy to remember how to use	Our software should be set up in such a way as to facilitate transparent usage, meaning that the user should not be able to mistake one option for another. In so doing we hope to reduce the complexity of our software and increase the user's proficiency in its usage faster.

Table 2: Usability Requirements Table

6.4 User Experience Requirements

We want our software to be available and easy to use for as many people as possible. Keeping in mind that this software will start as a text-based interface, it is paramount that the UI's text is easy to read and understand.

Our user experience goals are to evoke feelings of autonomy, confidence and completion.

7 Future plans

How we would finish implementing the requirements that are left and what we would like to fix in the future

7.1 how we would finish requirements

all the requirements that we didn't finish we marked red in the requirements list, here we will list them and how we would finish implementing them.

- 14. The system should be able to display a list of all voyages for a selected date/week: We would have to add another option in the voyages menu that says "list of all voyages for a date/week" and there would have to be a sub-menu where we select if we want to sort by a specific date or if we want to sort by week.There we can use similar code as before when we are sorting by captains, copilots and heads of service in the employee menu.
- 20. User should be able to select what airplane is used for voyage: To do this we would have to connect voyageData and planeData so that when we are registering a new voyage we can select a plane that we can get from planeData
- 21. User should be able to enter the number of sold seats for each flight: Right now we don't have anything called flight, only voyage. So the first step would be to add the function to create

something called flight. From there we can then generate a flight number for each registered flight.

- 22. The system should be able to assign a flight number to each flight: Similar to 21, we would need a function called flight, and when registering a flight then the user should enter the number of seats sold.
- 26. User can change the name of a contact that is registered to a destination: !!!!!Right now we almost have this, please change before turning in !!!!!!
- 27. User can change the name of a emergency number that is registered to a destination's contact: !!!!!Right now we almost have this, please change before turning in !!!!!!
- 28. The system should be able to display a list of all aircraft's owned by the airline and show their status on a specific date and time: We can currently display all aircraft's owned by the airline but we cant show their status on a specific date and time as of right now. To finish this we could add option in manage Airplanes to display all airplanes status. and there we would have to take in the input of the user of what date they want to search for and from there it should display all airplanes and their status for that specific date.
- 30. the system should be able to display a list of all airplanes that are not in use for a specific date: Similar to 28 but this time we would have to filter by the planes that are not in use, otherwise should be quite similar to 28
- 31. The system should be able to display a list of all pilots who are not working specific date and have the required license: we could add another option in the sub-menu in employees when we select filter employees and there we would add another option named filter pilot by date and license. There we would have to take in the date the user wants to sort by and also the name or NID of the pilot they want to search by.

8 Final words

The past two weeks we have been working on programming the NaN-Air booking system, there were many challenges along the way but with our relentless enthusiasm we overcame many challenges to make this beautiful system that we stand behind with pride.
In the NaN-Air booking system, we tried to embrace the innovation and efficiency of our legendary leader, Chuck Norris, and implemented those traits as much as possible.
The next step is to finish the requirements that we listed our future plans. We are very gratefully to our boss and thank him for giving us this awesome assignment, we learned so much and are very happy with our personal progress as programmers and also as a team, working together. The next step is to start coding the system with our selected design philosophy, but since our gracious boss is taking the weekend off, so are we.