



Department of Applied Informatics

Systems Development and Design

Group 13

Giannakovitis Antonios

Lambrianidis Dimitrios

Siatras Nikolaos

Sofianopoulos Ioannis-Christos

Konstantinos Psillas

May 2023

FINAL SUMMARY

Staff Summary

The work concerns the development of an information system for a hypothetical car repair shop which seeks to increase its turnover. The participants carried out a series of research and analysis in order to understand the needs of the workshop, proposed solutions and designed an information system that will improve its operation.

Workshop owners by providing specific requirements and instructions invite participants to manage their own systems development team, make critical decisions and work hard to meet the requirements of the owners and the entire business.

In their effort to achieve their goal they separate the development of the project in Preparation, Analysis, Design and Implementation.

In the Preparation stage participants should define the system goals and perform a series of analyzes and studies to understand the system and user requirements. During the Analysis stage the participants examine the needs and requirements of the workshop and create a complete map of the system. In the Design stage they plan the system and define its characteristics and its functions. Finally, in the Implementation stage, the participants plan what they planned.

To create the diagrams, participants will use special design programs, which will help them visualize user requirements and make better decisions. Even the participants will benefit significantly from organizing and managing a team through flexible software methods and their management skills will be greatly increased.

A great benefit in the area of code organization and sharing will be the Github platform, which allows easy access to the code by all team members. An important addition to the code will be the development of a graphical interface with an object-oriented approach and also a graphical interface using the JavaFX library.

Finally, for the asphalting and writing of the code, the participants will use the Eclipse development environment which will be the dominant tool during the implementation.

Table of Contents

Staff Summary	ii
Table of Contents.....	iii
Introduction.....	iv
PREPARATION	1
2.1 System Application	2
2.2 Feasibility Study	4
ANALYSIS.....	11
3.1 Business Modeling.....	12
3.2 Collection of Claims.....	16
3.3 Epics and User Stories.....	20
3.4 Use Case Diagram.....	26
3.5 Verbal Descriptions.....	27
3.6 Domain Model.....	51
PLANNING.....	52
4.1 Verbal Descriptions & Sequence Diagrams	53
4.2 Class Diagram.....	121
4.3 Object Diagrams.....	122
IMPLEMENTATION.....	124
Code.....	125
CONCLUSIONS.....	128
Requirements Analysis & Understanding.....	129
Design	129
Programming & Object Orientation.....	129
APPENDIX A.....	130
Team Organization and Roles.....	131
Timesheets	133
APPENDIX B	137
Team Evaluation.....	138
APPENDIX C.....	140
Bibliography	141

Introduction

In the field of software development, in order to properly develop an information system, a series of stages and processes are required. Through the case study of the "CarOps" workshop, we can see how the knowledge and application of these steps has a major role in the success of an information system.

Initially the idea for an information system should go through the preparation stage, which includes system application, feasibility analysis and project management. The term "system request" refers to a formal document that describes the desired results of the project as specified by the user who submitted the request. Feasibility analysis assesses the requirements and constraints of the business and determines whether the information system is suitable to meet those requirements.

Then, once the above stages are approved and the project management is completed, the transition to the analysis stage is made. Included here are three steps by which the current system is analyzed and room for improvement is identified. In particular, the development team is asked to develop an analysis strategy, collect requirements and integrate them into a document called a system proposal.

After passing the analysis, the requirements of the business are understood in order to build the Use Case Diagram correctly.

As we move toward the final stage of code development, we place special emphasis on implementing sequence diagrams, classes, and objects. These diagrams illustrate how the system works during interactions and reveal the structure of the data used in the information system.

Their goal is to help understand the functionality of the system and at the same time make it easier to write the code in the future stages. The design process is critical to successful system development, as the diagrams created during this phase form the foundation upon which developers will build the desired system.

Caution! The PDF document has links and buttons for easy navigation. Links are not active within the Google PDF viewer
Drive and Classroom



PREPARATION

- 2.1 System Application
- 2.2 Feasibility Study

2.1 System Application

Business Necessity

The system to be developed has as its basic need the improvement and automation of the company's business processes, which currently remain informal and unorganized.

For this reason, the system will contribute decisively to the improvement and management of relations with customers as well as to optimal coordination during the diagnosis and repair of vehicles.

Finally, the system will significantly help in making better decisions providing a series of reports on the set of business processes.

Business Requirements

In order to meet the modern operational needs of a company, it is important that the systems provide a wide range of capabilities, always optimized for the company.

The system provides this wealth of features and adequately covers all user requirements. Customers now have the possibility to make transactions with the VISA payment system and have the receipts of their transactions in their e-mail. Engineers work more efficiently and are more flexible thanks to scheduling appointments.

Finally, owners can, with their own eyes, see financial success coming, through Income-Expense Reports!

Business Value

The system contributes significantly to improving the functionality of the car workshop, facilitating the organization of vehicles to be repaired and the management of customers.

In particular, the system streamlines the customer arrival process, effectively addressing the crowding problem and enhancing the cooperation of engineers during repair. In addition, it increases customer satisfaction and attracts new customers to the business. The system also helps reduce excess inventory and improve management's supply chain management, thereby leading to cost savings and increased revenue.

The implementation of this system creates a competitive advantage for the business, making it more efficient and profitable.

Special Issues and Limitations

The system enhances users' security and privacy without storing sensitive financial information such as customers' credit card information. The system is also designed with the best specifications so that it is easy to maintain and upgrade it in the future.

Project Principal

The need to develop this particular system was expressed by the owners of the CarOps workshop as they were the first to see the expected benefits they would reap.

2.2 Feasibility Study

2.2.1 Technical Feasibility

Technical Requirements Survey

- Development of a new and small-sized information system
- Familiarization of the team with the necessary development technologies
- Non-existing technological structure
- Company members are not familiar with similar information systems. They will use this technology for the first time.
- A track record of integrating similar information systems is appreciated
low technical risk

Technical Feasibility Conclusions

After completing the technical requirements research, it was found that the development team can implement the specific IT project for the CarOps company. Furthermore, according to the results of the research, the information system that will be developed is characterized by low risk and can be completed easily in a certain time that will be agreed upon by the company.

In addition, the development team is familiar with the technologies it will be based on information system. He has extensive experience in the JAVA programming language, Object Oriented thinking, and web technologies.

However, the lack of technological equipment and the inexperience of the company's employees in using corresponding systems require the installation of a new system from scratch as well as the training of employees on it. Training should be key as it can greatly increase the performance and functionality of the business.

Factors Impact System		Analysis	Conclusion
Size		The system is not for a large enterprise	It will be completed faster and cost less. The business will be able to bear it more easily
Familiarity with the technology		Business members do not know the requested technology	Additional training is required to the employees who will use it and to the management of the company
Team Experience		The development team has a lot of experience	No additional development team training required, shorter project completion
Existing Technological Infrastructure		The company does not have any pre-existing technological infrastructure	Requires installation of equipment from scratch
Similar Cases in the Market		Developing systems of this size is a common occurrence	This is not an innovation. The risk is low

Table 1 – Impact of Technical Factors - Technical Feasibility

2.2.2 Corporate Feasibility

Corporate Claims Survey

- Refusal/Difficulty of employees to adopt the system in their work. Their job duties are simple and the system makes things complicated when they don't know how to handle it properly
- Coordination and better organization in arranging appointments and repairs
- With the specialization that the system will bring, there is a reduction in time execution of the tasks
- Benefit to the owners of the business by using its analysis tools
business progress
- The objectives of the Information System go hand in hand with the objectives of the business and together they create a competitive advantage

Corporate Feasibility Conclusions

Upon completion of the research, it is found that the information system that will be developed for the CarOps company aligns with its goals and enhances the business functions

More specifically, the system contributes decisively to the improvement of relations business and customer and maximizes the profit of both members by providing faster and more specialized service.

However, to be successful a system must also be supported by its users. The process of integrating the system, in the majority of cases, is subject to a negative attitude from some user groups, however in the case of CarOps, the employees through a complete and understandable training will be able to take full advantage of it and recognize benefits from the use

of.

Finally, the promotion and extensive presentation of the opportunity of the information system by the owners is important.

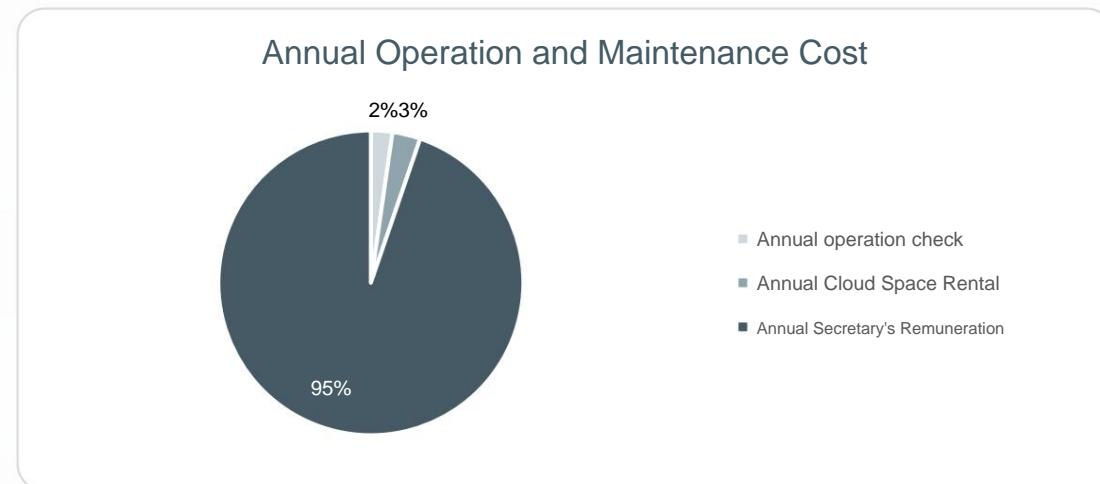
Employees	Stand	Role	Education	Impact
Administration	Super fighter	View Reports	Small	Creating a Competitive Advantage
				Development costs and maintenance costs
Secretariat		Customer Relationship Management	Great	New job position
				Education and familiarization with the system
Specialized Engineers	Slightly Opposite	Record data for repairs and spare parts	Great	Better organization and error avoidance
				Training and familiarization with the system
Simple ones Engineers	Neutral		Small	Better organization and normalization of the repair process
				Training and familiarization with the system

Table 2 - Employee Attitude - Corporate Purposefulness

2.2.3 Economic Feasibility

Cost Estimate

Company Fee	€40,300
Cost of Hardware and Software	€9,400
Total development costs	€49,700
Remuneration of Training Officers	€1,100
Total training costs	€1,100
Annual operation check	€245
Annual Cloud Space Rental	€320
Annual Secretary's Remuneration	€10,200
Annual operation & maintenance cost	€12,365
Total Development Cost	50. 800€
Annual cost	€12,365



Anticipated Financial Benefits

With the installation of the information system in the car workshop CarOps provides significant financial benefits for the business. Indicative:

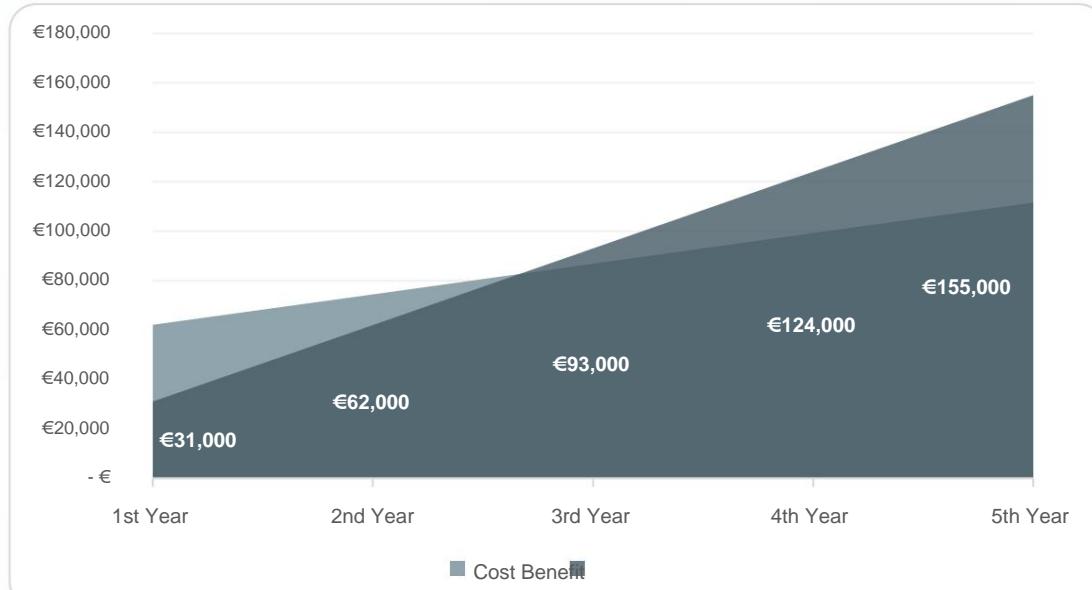
- More efficient management of parts and spare parts inventory. A reduction in inventory costs is expected by at least €3,000 per year.
- Reduction of customer waiting time, with improved and more accurate estimation of vehicle repair time. An increase in customers of at least 30% is expected with a potential increase in annual turnover of €24,500.
- Better prices and organization. By analyzing monthly reports owners can plan better and buy inventory earlier and at better prices. They are even able to make agreements with suppliers for even more favorable prices. Profit of at least €3,500 per year.

Cost-Benefit Analysis

Cost-benefit analysis is a method used to evaluate the economic advantages and offsets of the costs of investing in the system to be developed.

From the following figures, taking into account that the system has a lifespan of 5 years, it follows that the benefits at the end of the five years exceed the costs by €43,475.

Total Development Cost	€49,700
Annual cost	€12,365
Annual Benefit	€31,000
Total development cost + Cost 5 years	€111,525
Benefits 5 years	€155,000





Years	Result	Cost	Avail
1st Year	€31,065	€62,065	€31,000
2nd Year	€12,430	€74,430	€62,000
3rd Year	€6,205	€86,795	€93,000
4th Year	€24,840	€99,160	€124,000
5th Year	€43,475	€111,525	€155,000

Calculation of Financial Sizes

Finally, to evaluate the financial feasibility of the system, the Net Present Value (NPV) figures were calculated as well as the company's results at the end of each year.

Based on the data below, the NPV is calculated at 31,025 euros. The result of the NPV calculation is positive, so the investment in the system seems to be profitable and beneficial.



ANALYSIS

- 2.3 Business Modeling
- 2.3 How to Gather Requirements
- 2.4 Epics and User Stories
- 2.5 Use Case Diagram
- 2.6 Verbal Descriptions
- 2.7 Domain Model

3.1 Business Modeling

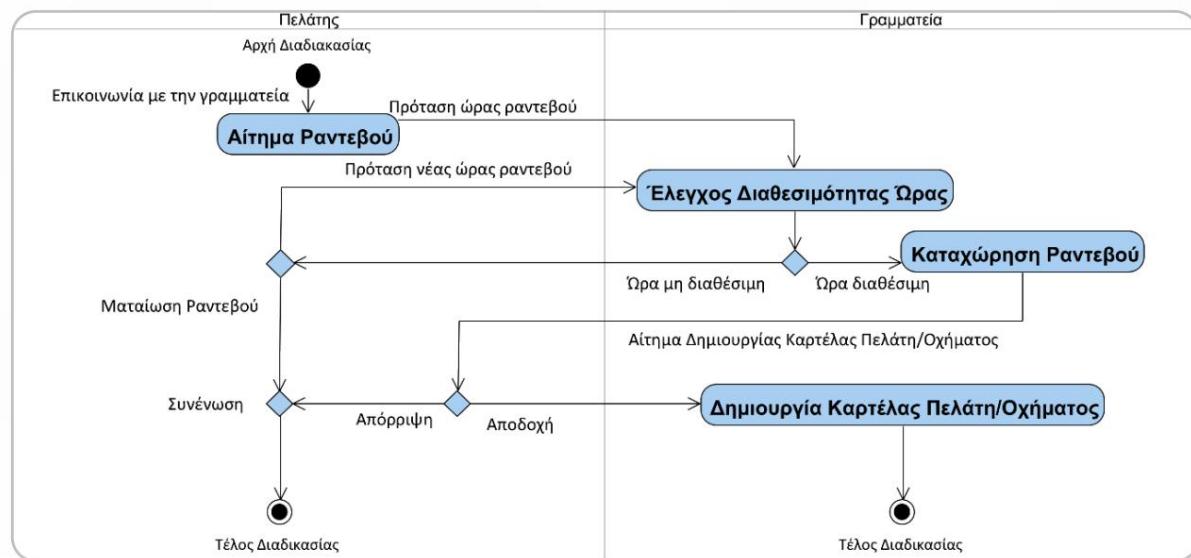
3.1.1 Process Lists

AS-IS procedures	Code
Vehicle Diagnosis and Costing	D1
Vehicle Repair	D2
Vehicle Pickup	D3

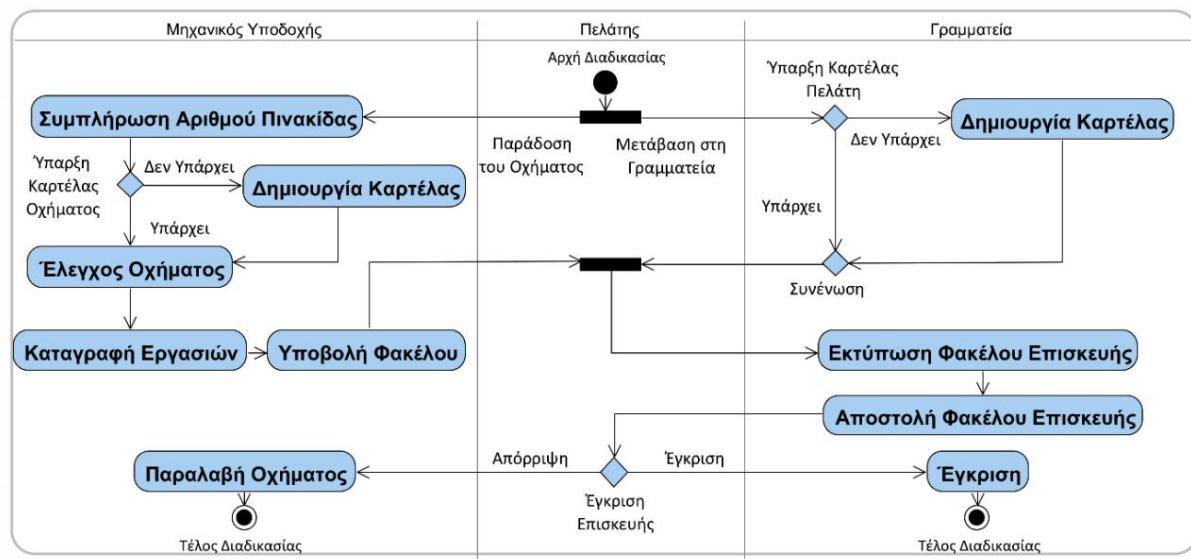
TO-BE procedures	Code	Change
Appointment Scheduling	DR	BRR
Vehicle Diagnosis	DD	BPI
Vehicle Repair	NOT	BPI
Vehicle Pickup	DP	BPA
Total		BPI

3.1.2 Activity Charts

Appointment Scheduling Process



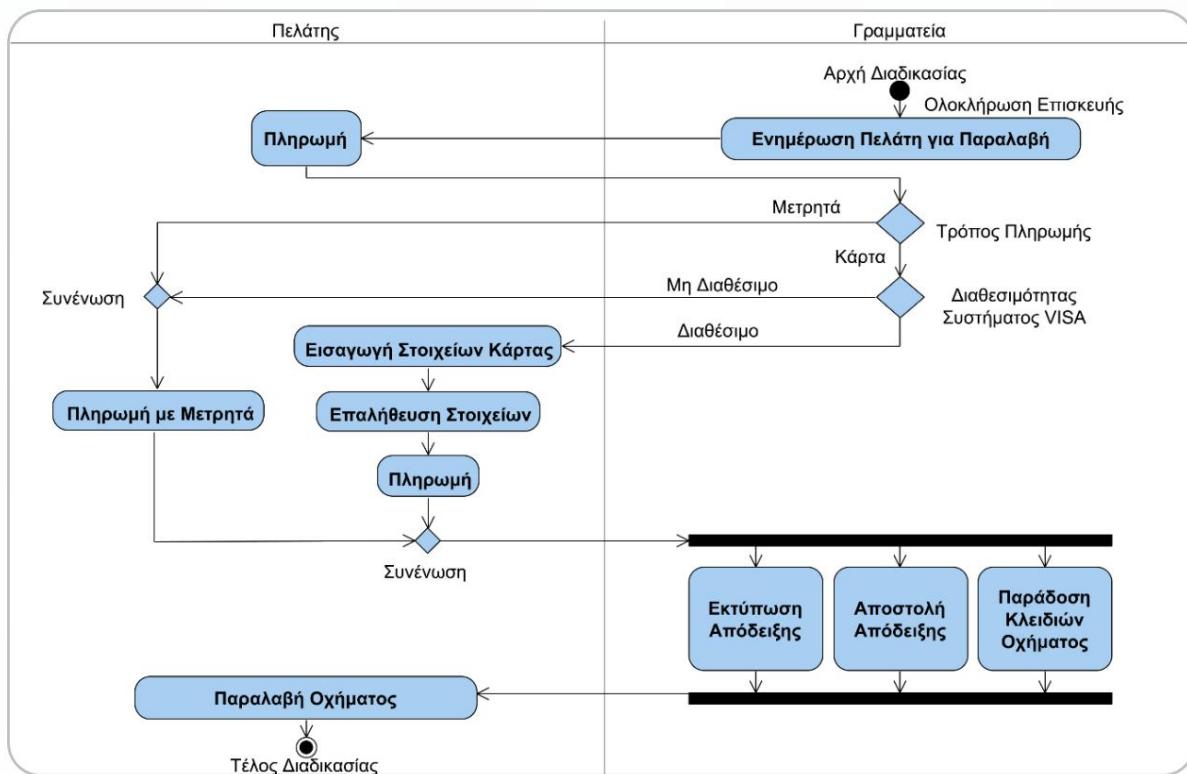
Vehicle Diagnostic Procedure



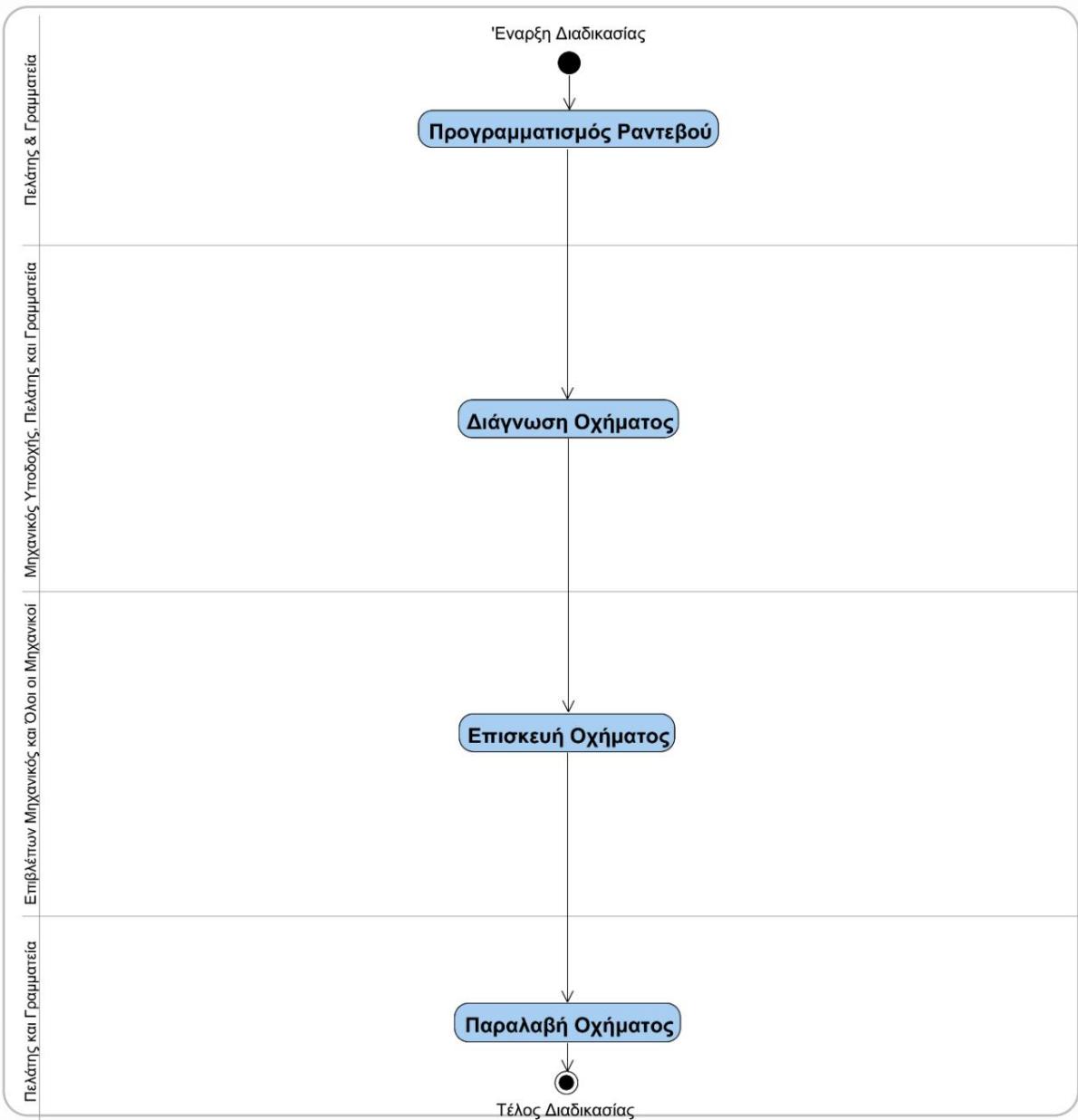
Vehicle Repair Procedure



Receipt Procedure



Overall the Procedures



3.2 Collection of Claims

Method of Collection

The development team largely gathered user requirements through interviews as this technique allows participants to provide detailed information about their needs and the features they would like to see on the system.

In addition, the team through questionnaires collected useful information from customers as this technique allows collecting requirements from a large number of people in a short period of time. The requirements collected and combined with the requirements collected through the interviews were used to compile a detailed list of software system requirements.

This list of requirements defines the functions and features that the system is expected to provide and will be used as a guide for software development.

It is divided into functional requirements and non-functional requirements. Functional requirements refer to the functions the software is expected to provide and describe the capabilities of the system. While non-functional requirements refer to constraints that must be met by the software.

Writing a detailed list of requirements is important because it helps ensure that the system being developed will meet all the needs and expectations of the users.

Functional Requirements

L1. Organization and Customer Service

1. The Secretariat will be able to create a new customer record and a new record vehicle with new customer details – **High Demand**
2. The Secretariat will be able to modify existing customer records and vehicles – **High Demand**
3. The Secretariat will be able to search for clients, view their details and associates them with vehicles or vice versa – **Average Requirement**
4. The Secretariat will be able to print the client's repair file – **Medium Requirement**
5. The Secretariat will be able to inform the system of the client's approval to proceed to repair – **Medium Requirement**
6. The Secretariat will be able to send the repair file electronically customer mail – **Low Req**

L2. Appointment Management

1. The Secretariat will be able to schedule appointments for a selected period space with unique identifiers – **High Requirement**
2. The Secretariat will be able to modify already scheduled appointments – **High Requirement**
3. The Secretariat will be able to see the already scheduled appointments in a weekly view – **Medium Requirement**

L3. Payment process

1. The Customer will be able to transact with the VISA payment system – **Average Requirement**
2. The Secretariat will be able to send payment receipts electronically of the customer – **Average Requirement**

L4. Vehicle Diagnostic Procedure

1. Reception Engineer will be able to submit repair file – **High Requirement**
2. The Reception Engineer will be able to schedule repairs and break them down in tasks – **Average Requirement**

L5. Project Coordination and Management

1. The Supervising Engineer will have the ability to manage and schedules some of the pending repairs – **High Demand**
2. The Supervising Engineer will be able to delegate to other engineers – **High Requirement**
3. All engineers will be able to mark a job as complete once the turnaround time and parts used details are filled in – **Average Requirement**
4. Supervising Engineer will be able to view pending repairs and their tasks – **Medium Requirement**
5. The Supervising engineer will be able to mark a repair as complete if all the work is done and he fills in all the types of spare parts that were used in the context of the specific repair – **Average Requirement**

L6. System Automations

1. Automatic generation of reports for the list of customers, vehicles, repairs, spare parts and Income-Expenses - **High Demand**
2. Automatic Repair Costing – **Average Requirement**
3. Automatic update of the number of used spare parts from the database after completion of repairs – **Average Requirement**
4. Automatic archiving of repair files in the database - **Low Requirement**

L7. Management of Stored Data

1. The Director will be able to see the Monthly Reports that he produces Information System – **High Demand**
2. The Manager will be able to view the repair history – **High Demand**
3. The Director and the Superintending Engineer will be able to modify the table with repair work – **Medium Requirement**
4. Manager and Superintending Engineer will be able to modify the table of spare parts (Part Name, Cost and Quantity) – **Average Requirement**

L8. Manage Login Accounts

1. Owners will be able to be notified when one of the accounts is locked and you will give them the option to unlock it – **Medium Requirement**
2. Accounts for Users will be automatically created by the system and will include their email and VAT number as Login Details – **Average Requirement**
3. Users will be able to change their password in the system – **Low Requirement**

Non-Functional Requirements

ML1. Operational Requirements

1. The system will cooperate with the VISA system for transactions – **High Requirement**
2. The system will be able to process and export files in .docx and .pdf format – **Medium Requirement**
3. The system is installed on Windows OS – **Medium Requirement**

ML2. Security Requirements

1. Owners will have access to reports as well as company data – **High Demand**
2. Each user of the system will have a separate type of login account to limit their access to the features assigned to them – **High Demand**
3. Rules will be followed for the creation of login codes in the system security so as to ensure account protection and reduce the likelihood of a breach – **Medium Requirement**
4. The system will maintain bi-monthly copies of the data of -**Low Requirement**

ML3. Performance Requirements

1. The system should be available and operational whenever the business needs – **High Demand**
2. The system to respond within 3 seconds after a user action – **Medium Requirement**
3. The system will support parallel usage without much increase in its response delay – **Low Demand**

ML4. Cultural and Political Requirements

1. The system will operate and provide information in the Greek Language – **Average Requirement**
2. The system is based on principles and operates within the legal framework of the General Data Protection Regulation – **Medium Requirement**

3.3 Epics and User Stories

Product Backlog

Epic ID	Description	MoSCoW	Ideal Days	Sprint
P1	He wants to transact with the VISA payment system	SH	2	1
P2	He wants payment receipts sent to his email	SH	2	1
M1	He wants to be informed about the tasks assigned to him	SH	2	3
M2	He wants to enter the number and types of spare parts used into the system	MH	3	2
M3	He wants to pass to the system statistical data related to the repair of vehicles	MH	3	2
WITH 1	He wants to outsource repairs to mechanics	MH	3	3
WITH 2	He wants to change the parts information and repair ready lists	SH	3	2
MY 1	He wants to register the details of the imported vehicle in the system	SH	3	1
MY 2	He wants to register the results of the check in the system	SH	3	3
MY 3	He wants the repair cost to be displayed automatically	CH	4	3

Epic ID	Description		MoSCoW Ideal Days Sprint		
JURMO	D1	He wants customer reports, vehicles and repair items sent to him every 1st of the month	MH	4	2
	D2	He wants to be sent the monthly income of the business every 1st of the month	MH	3	2
	D3	He wants to change the parts information and repair ready lists	SH	3	2
	C1	He wants to enter customer details into the system	SH	2	1
	C2	He wants to manage client appointments	MH	3	1
	C3	He wants to print the repair file	CH	2	3
	C4	He wants to validate the start of a vehicle repair	MH	2	3

Detailed User Stories

Secretariat

- C1 - As Secretariat I want to be able to create a form for each client with the their necessary information, the customer record.

He wants to enter customer details into the system

- C2 - As the Secretariat, I want to be shown a screen with the monthly scheduled appointments for better organization and reduction of overcrowding.

He wants to manage client appointments

- C3 - As the Secretariat I want to print the repair file to provide it to the customer.

He wants to print the repair file

- C4 - As the Secretariat, I want to inform the system about the customer's decision to stop or continue the vehicle repair process.

He wants to validate the start of a vehicle repair

Customer

- Q1 - As a Customer, I want to be able to enter my card details into the VISA Information System to complete the payment and pick up my vehicle.

He wants to transact with the VISA payment system

- P2 - As a Customer, regardless of the payment method I choose, I want the receipt to be sent to me by email so that I can be informed about the transactions my.

He wants his transaction receipts sent to his email

Owner

- D1 - As an Owner I want the system to automatically generate monthly reports with customers, vehicles, repair items and spare parts stored in the system so that I am informed about the state of the business.

He wants customer reports, vehicles and repair items sent to him every 1st of the month

- D2 - As an Owner I want to have access to the monthly income of the business to monitor its efficiency.

He wants to be sent the monthly income of the business every 1st of the month

- D3 - As the Owner, I want to update the list of spare parts and the list of standard repair work in the system for better organization and correct repair.

**He wants to change the details of the spare parts and their ready lists
repairs**

Supervising engineer

1. ME1 - As a Supervising Engineer I want to clearly assign the appropriate tasks to the engineers of the corresponding experience to avoid mistakes and misinterpretations.

He wants to outsource repairs to mechanics

- ME2 - As a Supervising Engineer I want to update the list of spare parts and the list of standard repair work in the system for better organization and more correct repair.

**He wants to change the details of the spare parts and their ready lists
repairs**

Reception Engineer • MY1

- As a Reception Engineer I want to register his details in the system imported vehicle, to create the vehicle record.

He wants to register the details of the imported vehicle in the system

- MY2 - As a Reception Engineer I want to register the repair work in the system to create the repair file.

He wants to register the results of the check in the system

- MY3 - As a Reception Engineer I want based on the repair work I have register to have the repair cost automatically calculated.

He wants the repair cost to be displayed automatically

All Engineers

- M1 - As an Engineer I want to see in the system the tasks assigned to me for to process them successfully without problems.

He wants to be informed about the tasks assigned to him

2. M2 - As an Engineer I want to update the system on the number and types of repair parts used so that the repair is automatically costed and the spare parts inventory is replenished.

He wants to enter the number and types of spare parts used into the system

- M3 - As an Engineer I want to inform the system about the duration of its completion work assigned to me.

He wants to pass to the system statistical data related to the repair of vehicles

Backlog Snapshots in the Jira Environment

Projects / CarOps System

Backlog

Sprint 1 1 Apr – 13 Apr (5 issues)

- CROS-7 Θέλει να κάνει συναλλαγές με το σύστημα πληρωμών της VISA ΠΕΛΑΤΗΣ TO DO
- CROS-8 Θέλει οι αποδείξεις πληρωμών να του αποστέλλονται στο ηλεκ... ΠΕΛΑΤΗΣ TO DO
- CROS-9 Θέλει να κατοχυρώνει στο σύστημα τα στοιχεία το... ΜΗΧΑΝΙΚΟΣ ΥΠΟΔΟΧΗΣ TO DO
- CROS-10 Θέλει να κατοχυρώνει στο σύστημα τα στοιχεία των πελατ... ΓΡΑΜΜΑΤΕΙΑ TO DO
- CROS-11 Θέλει να διαχειρίζεται τα ραντεβού των πελατών ΓΡΑΜΜΑΤΕΙΑ TO DO

+ Create issue

Sprint 2 13 Apr – 29 Apr (6 issues)

- CROS-12 Θέλει να περνάει στο σύστημα το πλήθος και τα είδη των α... ΜΗΧΑΝΙΚΟΣ TO DO
- CROS-13 Θέλει να περνάει στο σύστημα στατιστικά δεδομένα σχετικ... ΜΗΧΑΝΙΚΟΣ TO DO
- CROS-14 Θέλει να αλλάζει τα στοιχεία των ανταλλακτικώ... ΕΠΙΒΛΕΠΩΝ ΜΗΧΑΝΙΚΟΣ TO DO
- CROS-15 Θέλει κάθε 1η του μήνα να του στέλνονται αναφορές πελα... ΙΔΙΟΚΤΗΤΗΣ TO DO
- CROS-16 Θέλει κάθε 1η του μήνα να του στέλνονται τα μηνιαία έσοδα... ΙΔΙΟΚΤΗΤΗΣ TO DO
- CROS-17 Θέλει να αλλάζει τα στοιχεία των ανταλλακτικών και τις έτ... ΙΔΙΟΚΤΗΤΗΣ TO DO

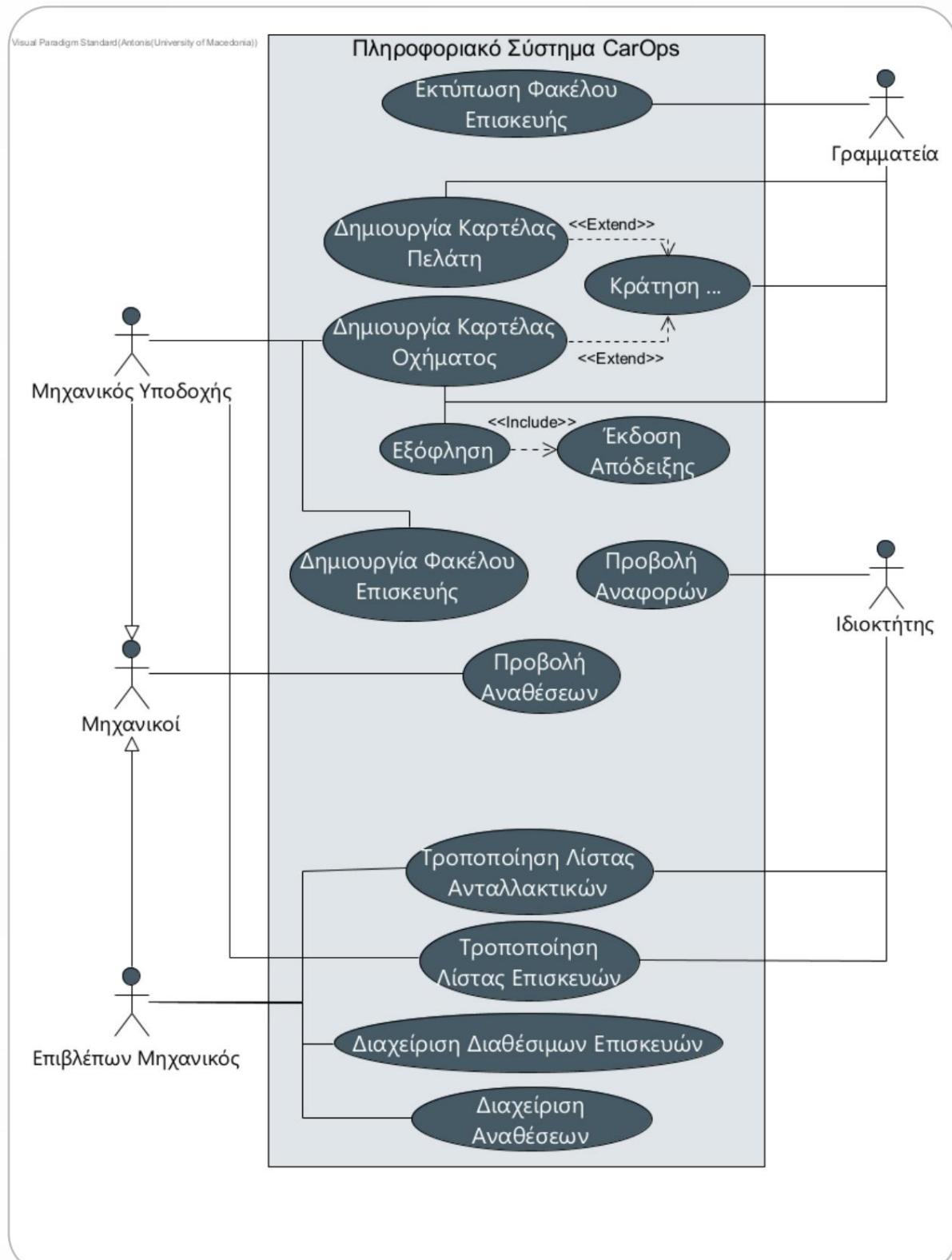
+ Create issue

Sprint 3 29 Apr – 16 May (6 issues)

- CROS-18 Θέλει να ενημερώνεται για τις εργασίες που του έχουν ανα... ΜΗΧΑΝΙΚΟΣ TO DO
- CROS-19 Θέλει να αναθέτει επισκευές σε μηχανικούς ΕΠΙΒΛΕΠΩΝ ΜΗΧΑΝΙΚΟΣ TO DO
- CROS-20 Θέλει να κατοχυρώνει στο σύστημα τα αποτελέσ... ΜΗΧΑΝΙΚΟΣ ΥΠΟΔΟΧΗΣ TO DO
- CROS-21 Θέλει να εμφανίζεται το κόστος επισκευής αυτόμ... ΜΗΧΑΝΙΚΟΣ ΥΠΟΔΟΧΗΣ TO DO
- CROS-22 Θέλει να εκτυπώνει τον φάκελο επισκευής ΓΡΑΜΜΑΤΕΙΑ TO DO
- CROS-23 Θέλει να επικυρώνει την έναρξη επισκευής ενός οχήματος ΓΡΑΜΜΑΤΕΙΑ TO DO

+ Create issue

3.4 Use Case Diagram



3.5 Verbal Descriptions

Verbal descriptions were a key aspect of system analysis and design. They provided a powerful method for describing system functions, interfaces, and requirements.

Thanks to these, the team was able to develop a detailed picture of the system's behavior and characteristics, while capturing its structure and the relationships between its various components.

In addition, verbal descriptions proved essential to the successful development and implementation of the system, ensuring accuracy, understanding and efficiency of information exchange between all stakeholders.



1. "Print Repair Folder" Basic Flow

1. The secretariat selects "View Repair File List" (Figure 1)
2. The system displays a list of repair folders
3. The secretariat selects a desired repair file and selects "Print" (Fig 2)
4. The system displays the repair report (Figure 3)

Σύστημα CarOps Αρχική Οθόνη Σύνδεση ως Γραμματεία

Λίστα Επιλογών

Πελάτες **Οχήματα** **Λίστα Φακέλων Επισκευής** **Ραντεβού**

Σύστημα CarOps Λίστα Φακέλων Επισκευής

Κωδικός Φακέλου	Κόστος σε Ευρώ	Ημερομηνία Ραντεβού	Προβολή	Εκτύπωση
1	50	12/4/2023 13:30	[Προβολή]	[Εκτύπωση]
2	35	12/4/2023 15:30	[Προβολή]	[Εκτύπωση]
3	30	13/4/2023 11:00	[Προβολή]	[Εκτύπωση]
4	70	13/4/2023 17:00	[Προβολή]	[Εκτύπωση]
5	120	14/4/2023 16:00	[Προβολή]	[Εκτύπωση]

Επιστροφή

Σύστημα CarOps Κωδικός Φακέλου: 1

Η εκτύπωση έγινε με επιτυχία !

Ονοματεπώνυμο : Γιάννης Παπαδόπουλος Κωδικός Φακέλου : 2
 Πινακίδα Οχήματος: ZXI-6581 Κωδικός Ραντεβού: 2
 Εκτιμώμενος Χρόνος : 5 ώρες Συνολικό Κόστος Επισκευής: 60 ευρώ

Λίστα Εργασιών

Όνομα Εργασίας	Κόστος Εργασίας
1 Ευθυγράμμιση Τροχών	20
2 Αλλαγή Λάστιχον	25
3 Αλλαγή Λαδιών	15

Επιστροφή **Κατάσταση Επισκευής : Αναμονή Απόφασης** [Ενημέρωση]

Comments

The 'Repair Folder List' screen is visible to the secretariat and the reception engineer



2. Basic Flow "Create Customer Record"

1. The secretariat selects "Customers" (Figure 1)
2. The system displays a list of customer records
3. The secretariat selects "Create New Tab" (Figure 2)
4. The system displays a data filling form for the customer 5. The secretary fills in the data and presses "Finish" (Figure 3)
6. The system displays in the secretariat "Successful Registration of Customer Card"

Alternative Flow

- 3.a The secretariat selects the desired tab and clicks "Modify" (Fig 4)
- 3.b The system displays the details of the selected customer
- 3.c The secretariat modifies the information it wishes and selects "Completion" (Figure 5)
- 3.d The system displays in the secretariat "Customer Card Processing Successful"

Σύστημα CarOps Αρχική Οθόνη

Σύνδεση ως Γραμματεί

Λίστα Επιλογών

Πελάτες **Οχήματα** Λίστα Φακέλων Επισκευή Ραντεβού

Σύστημα CarOps Πελάτες

Δημιουργία Νέας Καρτέλας

Πλάνης Παπαδόπουλος	Τροποποίηση	Διαγραφή
Μανώλης Κόντογλου	Τροποποίηση	Διαγραφή
Ιωάννης Αντωνίου	Τροποποίηση	Διαγραφή
Γιώργος Ντριανταφύλλου	Τροποποίηση	Διαγραφή
Βαγγέλης Χατζηγεωργίου	Τροποποίηση	Διαγραφή
Νίκος Ιωαννίδης	Τροποποίηση	Διαγραφή
Αντώνης Λιόλιος	Τροποποίηση	Διαγραφή
Μάκης Κοτσάμπασης	Τροποποίηση	Διαγραφή

Σύστημα CarOps Νέα Καρτέλα Πελάτη - □ X

Όνομα	Επίθετο
Σταθερό Τηλέφωνο	
Διεύθυνση	
Email	

Σύστημα CarOps Πελάτες - □ X

<input type="button" value="search"/>	Δημιουργία Νέας Καρτέλας
Γιάννης Παπαδόπουλος	<input type="button" value="Τροποποίηση"/> Διαγραφή
Μανώλης Κόντογλου	<input type="button" value="Τροποποίηση"/> Διαγραφή
Ιωάννης Αντωνίου	<input type="button" value="Τροποποίηση"/> Διαγραφή
Γιώργος Ντριανταφύλλου	<input type="button" value="Τροποποίηση"/> Διαγραφή
Βαγγέλης Χατζηγεωργίου	<input type="button" value="Τροποποίηση"/> Διαγραφή
Νίκος Ιωαννίδης	<input type="button" value="Τροποποίηση"/> Διαγραφή
Αντώνης Λιόλιος	<input type="button" value="Τροποποίηση"/> Διαγραφή
Μάκης Κοτσάμπασης	<input type="button" value="Τροποποίηση"/> Διαγραφή

Σύστημα CarOps Τροποποίηση Καρτέλας Πελάτη - □ X

Πιάννης	Παπαδόπουλος
2310000000	
Βενιζέλου 21	
john@gmail.com	



3. Basic "Book Appointment" Flow

1. The secretariat selects "Appointment" (Figure 1)
2. The system displays the Appointment screen
3. The secretariat selects "New Appointment" (Figure 2)
4. The system asks the secretariat for the booking details
5. The secretariat fills in the reservation details and selects "Save" (Figure 3)
6. The system displays in the secretariat "Successful Appointment Registration"

Alternative Streams

- 3.1.a The secretariat selects "Change" (Figure 4)
 3.1.b The system asks the secretariat for the new booking details (Figure 5)
 3.2.a The secretariat selects "Cancel Appointment" (Figure 6)
 3.2.b The system displays informational message and deletes the appointment from the program (Figure 7)

Σύστημα CarOps			Αρχική Οθόνη	- □ ×
Σύνδεση ως Γραμματεία				
Λίστα Επιλογών				
<input type="button" value="Πελάτες"/> <input type="button" value="Οχήματα"/> <input type="button" value="Λίστα Φυκέλων Επισκευής"/> <input type="button" value="Ραντεβού"/>				
Σύστημα CarOps			Ραντεβού	- □ ×
<u>Όνοματεπώνυμο</u>	<u>Πινακίδα</u>	<u>Ημερομηνία</u>	<input type="button" value="Αναζήτηση"/>	
Γιάννης Παπαδόπουλος	ZXI-6581	12/4/2023 13:30	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Μανώλης Κόντογλου	KZN-1013	12/4/2023 15:30	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Ιωάννης Αντωνίου	TKN-3554	13/4/2023 11:00	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Γιώργος Ντριανταφύλλου	INY-9876	13/4/2023 17:00	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Βαγγέλης Χατζηγεωργίου	KZN-1098	14/4/2023 16:00	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
<input type="button" value="Επιστροφή"/> <input type="button" value="Νέο Ραντεβού"/>				
Σύστημα CarOps			Ραντεβού	- □ ×
<u>Όνοματεπώνυμο</u>	<u>Πινακίδα</u>	<u>Ημερομηνία</u>	<input type="button" value="Αναζήτηση"/>	
Γιάννης Παπαδόπουλος	ZXI-6581	12/4/2023 13:30	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Μανώλης Κόντογλου	KZN-1013	12/4/2023 15:30	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Ιωάννης Αντωνίου	TKN-3554	13/4/2023 11:00	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Γιώργος Ντριανταφύλλου	INY-9876	13/4/2023 17:00	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Βαγγέλης Χατζηγεωργίου	KZN-1098	14/4/2023 16:00	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
<input type="button" value="Επιστροφή"/> <input type="button" value="Νέο Ραντεβού"/>				

Σύστημα CarOps		Τροποποίηση Ραντεβού		- □ ×
Κωδικός Ραντεβού : 2				
Όνοματεπώνυμο:	Γιώργος Ιακωβίδης			
Πινακίδα Οχήματος:	ANK-1243			
Ημερομηνία:	12/4/2023	11:30		
		<input type="button" value="Ακύρωση"/>	<input type="button" value="Αποθήκευση"/>	
Σύστημα CarOps		Δημιουργία Νέου Ραντεβού		- □ ×
Όνοματεπώνυμο:	Γιώργος Ιακωβίδης			
Πινακίδα Οχήματος:	ANK-1243			
Ημερομηνία:	10/4/2023			
Ώρα	10:30			
		<input type="button" value="Ακύρωση"/>	<input type="button" value="Αποθήκευση"/>	
Σύστημα CarOps		Ραντεβού		- □ ×
Όνοματεπώνυμο	Πινακίδα	Ημερομηνία	<input type="button" value="Αναζήτηση"/>	
Γιάννης Παπαδόπουλος	ZXI-6581	12/4/2023 13:30	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Μανώλης Κόντογλου	TKN-4040	10/4/2023 15:30	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Ιωάννης Αντωνίου		Επιβεβαίωση Διαγραφής 00	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Γιώργος Ντριανταφύλλου	Aκύρωση	Επιβεβαίωση 00	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Βαγγέλης Χατζηγεωργίου	KZN-1098	14/4/2023 16:00	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
		<input type="button" value="Επιστροφή"/>	<input type="button" value="Νέο Ραντεβού"/>	
Σύστημα CarOps		Ραντεβού		- □ ×
Όνοματεπώνυμο	Πινακίδα	Ημερομηνία	<input type="button" value="Αναζήτηση"/>	
Μανώλης Κόντογλου	KZN-1013	12/4/2023 15:30	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Ιωάννης Αντωνίου	TKN-3554	13/4/2023 11:00	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Γιώργος Ντριανταφύλλου	INY-9876	13/4/2023 17:00	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Βαγγέλης Χατζηγεωργίου	KZN-1098	14/4/2023 16:00	<input type="button" value="Αλλαγή"/>	<input type="button" value="Διαγραφή"/>
Το Ραντεβού διαγράφηκε με επιτυχία !				
		<input type="button" value="Επιστροφή"/>	<input type="button" value="Νέο Ραντεβού"/>	

Comments

The appointment screen contains a calendar in weekly format as well as the options "New Appointment", "Modify Appointment" and "Cancel Appointment". This screen is only available in the secretariat.



4. Basic Flow "Create Vehicle Tab"

1. The secretariat selects "Show Vehicle List" (Figure 1)
2. The system displays a list of vehicle tabs
3. The secretariat selects "Create New Tab" (Figure 2)
4. The system displays a vehicle details form
5. The secretary fills in the vehicle details and clicks "Finish"
- (Figure 3)
6. The system displays in the secretariat "Successful Tab Registration vehicle"

The figure consists of three vertically stacked screenshots of the CarOps application:

- Screenshot 1 (Top):** Shows the main menu with tabs for Πελάτες, Οχήματα, Λίστα Φακέλων Επισκευή, and Ραντεβού. The Οχήματα tab is selected.
- Screenshot 2 (Middle):** Shows a list of vehicles under the heading Δημιουργία Νέας Καρτέλας. The table data is as follows:

ΚΩΔΙΚΟΣ	ΤΡΟΠΟΠΟΙΗΣΗ	ΔΙΑΓΡΑΦΗ
ZXI-1581	Τροποποίηση	Διαγραφή
KZN-1113	Τροποποίηση	Διαγραφή
TKN-3554	Τροποποίηση	Διαγραφή
INY-9876	Τροποποίηση	Διαγραφή
KZN-9810	Τροποποίηση	Διαγραφή
TZN-2345	Τροποποίηση	Διαγραφή
HMN-1026	Τροποποίηση	Διαγραφή
ZXN-5492	Τροποποίηση	Διαγραφή

- Screenshot 3 (Bottom):** Shows a new vehicle registration form titled Νέα Καρτέλα. It contains fields for Πινακίδα, Μάρκα, Μοντέλο, and Ετος Κατασκευής. At the bottom are buttons for Ακύρωση and Ολοκλήρωση.

Alternative Flow

- 3.e The secretariat selects the desired tab and clicks "Modify" (Fig 4)
- 3.f The system displays the details of the selected vehicle
- 3.g The secretariat modifies the information it wishes and selects "Completion" (Figure 5)
- 3.h The system displays in the secretariat "Successful Card Processing"

Σύστημα CarOps Οχήματα		
<input type="button" value="search"/>	Δημιουργία Νέας Καρτέλας	- □ X
ZXI-1581	Τροποποίηση	Διαγραφή
KZN-1113	Τροποποίηση	Διαγραφή
TKN-3554	Τροποποίηση	Διαγραφή
INY-9876	Τροποποίηση	Διαγραφή
KZN-9810	Τροποποίηση	Διαγραφή
TZN-2345	Τροποποίηση	Διαγραφή
HMN-1026	Τροποποίηση	Διαγραφή
ZXN-5492	Τροποποίηση	Διαγραφή

Σύστημα CarOps Νέα Καρτέλα		- □ X
ZXI-1581	BMW	
iX1 xDrive30		
2022		
	<input type="button" value="Ακύρωση"/>	<input type="button" value="Ολοκλήρωση"/>

vehicle"



5. "Create Repair Folder" Basic Flow

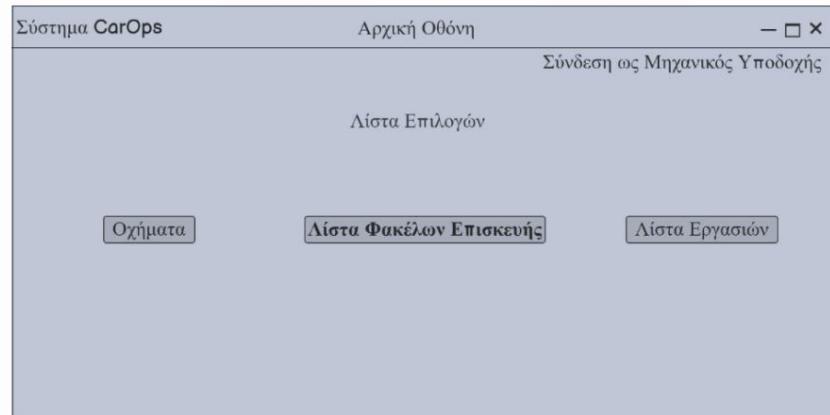
1. The host engineer selects "View Repair Folder List" (Figure 1)
2. The system displays a list of repair folders
3. The host engineer selects "Create New Repair File" (Figure 2)
4. The system prompts the receptionist to select which appointment he wants to create a new folder for (Figure 3)
5. The system displays on the Screen the types of the vehicle when it is not registered for the Reception Engineer to select
6. Reception Engineer selects vehicle type (Figure 4)
7. The system displays a corresponding data form to be filled suitable for each type of vehicle 8. The Reception Engineer fills in the data and selects <>Registration>> (Figure 5)
9. System displays registration confirmation message 10. Reception Engineer confirms registration
11. The system displays a create task list screen
12. The Reception Engineer adds the tasks he wants and chooses <>Submission Folder Overview>> (Figure 6)
13. The system displays an overview of the repair folder
14. The Reception Engineer selects Submit (Figure 7)
15. The system displays an informational message and returns to the home screen

Alternative Flow

- 5.1.1 The vehicle is already registered and the system displays the Vehicle details 5.1.2 Reception Engineer selects Continue (Figure 8)
5.1.3 Flow continues from step 11

Comments

The 'Repair Folder List' screen is visible to the secretariat and the reception engineer. The Reception Engineer creates repair files based on the corresponding scheduled appointments. When the Reception Engineer adds one manually (Figure 6) then it is automatically saved in the list of saved jobs.



Σύστημα CarOps Λίστα Φακέλων Επισκευής			
Κωδικός Φακέλου	Κόστος σε Ευρώ	Ημερομηνία Ραντεβού	Προβολή
1	50	12/4/2023 13:30	[Προβολή]
2	35	12/4/2023 15:30	[Προβολή]
3	30	13/4/2023 11:00	[Προβολή]
4	70	13/4/2023 17:00	[Προβολή]
5	120	14/4/2023 16:00	[Προβολή]

[Επιστροφή] [Δημιουργία νέου Φακέλου]

Σύστημα CarOps Δημιουργία Φακέλου για Ραντεβού

Επιλέξτε για ποιο ραντεβού θέλετε να δημιουργήσετε φάκελο επισκευής

Όνοματεπώνυμο	Πινακίδα	Ημερομηνία	
Ιωάννης Αντωνίου	TKN-3554	13/4/2023 11:00	<input type="button" value="Δημιουργία"/>
Γιώργος Ντριανταφύλλου	INY-9876	13/4/2023 17:00	<input type="button" value="Δημιουργία"/>
Βαγγέλης Χατζηγεωργίου	KZN-1098	14/4/2023 16:00	<input type="button" value="Δημιουργία"/>

Σύστημα CarOps Δημιουργία Φακέλου για Ραντεβού

Το όχημα με Πινακίδα TKN-3554 δεν είναι καταχωρημένο στο σύστημα

Επιλέξτε τον τύπο του οχήματος

Σύστημα CarOps Συμπλήρωση Στοιχείων Οχήματος

Συμπληρώστε τα Στοιχεία του Οχήματος

Μάρκα:	<input type="text" value="Toyota"/>
Μοντέλο :	<input type="text" value="Yaris"/>
Έτος Κατασκευής:	<input type="text" value="2011"/>

Σύστημα CarOps Συμπλήρωση Στοιχείων Οχήματος

Συμπληρώστε τα Στοιχεία του Οχήματος

Καταχώριση οχήματος και συνέχεια διαδικασίας;

OXI	NAI
-----	-----

Σύστημα CarOps Δημιουργία Λίστας Εργασιών Διάγνωσης

Προσθήκη Αποθηκευμένης Εργασίας		Λίστα Εργασιών	Χειροκίνητη Προσθήκη
<input type="checkbox"/> Αναζήτηση 1 Αλλαγή Λυχνία 2 Αντικατάσταση Αναρτήσεων 3 Αντικατάσταση Φρένων 4 Επισκευή Τουρμπίνας 5 Καθαρισμοί Μπεκ 6 Ευθυγράμμιση Τροχών 7 Αλλαγή Λάστιχων	1 Ευθυγράμμιση Τροχών × 2 Αλλαγή Λάστιχων × 3 Αλλαγή Λαδιών ×	Όνομα Εργασίας : Αλλαγή Λαδιών Εκτιμώμενο Κόστος : 15 Προσθήκη	
		Εκτιμώμενος Χρόνος Εισαγωγή: 5 ώρες Ακύρωση Επισκόπηση Φακέλου Υποβολής	

Σύστημα CarOps Επισκόπηση και Υποβολή Φακέλου

Όνοματεπώνυμο : Ιωάννης Αντωνίου Πινακίδα Οχήματος: TKN-3554 Εκτιμώμενος Χρόνος : 5 ώρες	Κωδικός Φακέλου : 3 Κωδικός Ραντεβού: 3 Συνολικό Κόστος Επισκευής: 60 ευρώ
Λίστα Εργασιών	
Όνομα Εργασίας 1 Ευθυγράμμιση Τροχών 2 Αλλαγή Λάστιχων 3 Αλλαγή Λαδιών	Κόστος Εργασίας 20 25 15
Κατάσταση Επισκευής : Αναμονή Απόφασης Επιστροφή Υποβολή	

Σύστημα CarOps Εμφάνιση Στοιχείων Οχήματος

Το όχημα με Πινακίδα TKN-3554 εντοπίστηκε στο σύστημα

Στοιχεία Οχήματος:

Μάρκα:	Toyota
Μοντέλο :	Yaris
Έτος Κατασκευής:	2011
Τύπος:	Αυτοκίνητο

Επιστροφή **Συνέχεια**

Σύστημα CarOps Εμφάνιση Στοιχείων Οχήματος

Το όχημα με Πινακίδα TKN-3554 εντοπίστηκε στο σύστημα

Στοιχεία Οχήματος:

Συνέχεια διαδικασίας ;	
OXI	NAI
Tύπος:	Αυτοκίνητο

Επιστροφή **Συνέχεια**



6. Basic Flow "View Reports"

1. The owner selects "Reports" (Figure 1) 2.

The system prompts the owner to select the type of report 3. The owner selects the type of report he wants (Figure 2) 4. The system displays a list of reports of this type 5. The user selects the desired report (Figure 3) 6. The system displays the desired report

Comments

The owner has a choice between Income and Expense Reports and reports spare parts. The 'Reports' option is not visible to other users.

Σύστημα CarOps	Αρχική Οθόνη	- □ X
		Σύνδεση ως Ιδιοκτήτης
	Αναφορές	

Σύστημα CarOps	Αναφορές	- □ X
		Σύνδεση ως Ιδιοκτήτης
	Αναφορές Εσόδων/Εξόδων	Αναφορές Ανταλλακτικών

Σύστημα CarOps	Αναφορές Εσόδων-Εξόδων	- □ X
<input type="button" value="search"/> <input type="button" value="Εσόδων"/>	<input type="button" value="Εσόδων"/>	
<input type="button" value="Αναφορά Ιανουαρίου 2023"/> <input type="button" value="Εκτύπωση"/>	<input type="button" value="Αναφορά Ιανουαρίου 2023"/> <input type="button" value="Εκτύπωση"/>	
<input type="button" value="Αναφορά Δεκεμβρίου 2022"/> <input type="button" value="Εκτύπωση"/>	<input type="button" value="Αναφορά Δεκεμβρίου 2022"/> <input type="button" value="Εκτύπωση"/>	
<input type="button" value="Αναφορά Νοεμβρίου 2022"/> <input type="button" value="Εκτύπωση"/>	<input type="button" value="Αναφορά Νοεμβρίου 2022"/> <input type="button" value="Εκτύπωση"/>	



7. "Modify Parts List" Basic Flow

1. Supervising engineer selects “Spare parts” (Figure 1)
2. System asks supervising engineer to select part type
3. Supervising engineer selects appropriate type (Figure 2)
4. System asks supervising engineer to select spare part
5. The supervising engineer selects the desired spare part (Figure 3)
6. The system displays the spare part page
7. The supervising engineer selects “Change Inventory” (Figure 4)
8. The system asks the user for the new quantity
9. The user fills in the new quantity and presses “Save” (Figure 5)



Σύστημα CarOps **Επιλογή ανταλλακτικού**

Παρακαλώ επιλέξτε ανταλλακτικό για το σύστημα φρένων

Τακάκια φρένου	Αναζήτηση
Αισθητήρας φθοράς	
Δισκόπλακα	
Τέλασμα πιτσίλισματος	
Φρένα τομπάνου	
Σιαγόνες	
Ταμπούρο	
Δαγκάνα φρένων	
Πιστόνι	
Χειρόφρενο	

Σύστημα CarOps **Τακάκια φρένου**

TAKAKIA ΦΡΕΝΟΥ

Τιμή: 25€
Απόθεμα: 50 τεμάχια [Τροποποίηση Αποθέματος]
Ημερομηνία τελευταίας αγοράς: 08/04/202

Παρακαλώ εισάγεται την νέα ποσότητα:

45	
Ακύρωση	Αποθήκευση

Comments

The part page displays various information such as what is currently in stock, date of last purchase, who is the supplier of the part as well as the 'Change Stock' option. The Parts option is available to both the supervising engineer and the owner.



8. "Modify Task List" Basic Flow

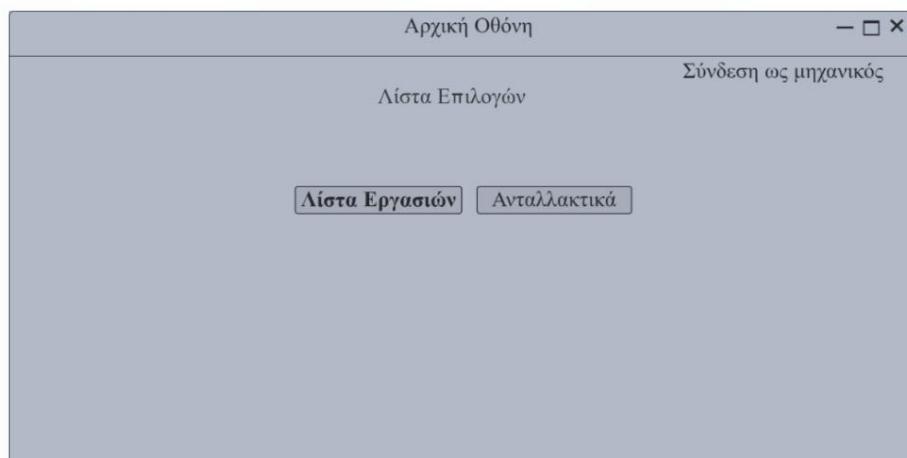
1. Reception Engineer selects "Task List" (Figure 1)
2. The system shows the Reception Engineer a list of saved jobs and the cost of each
3. The Reception Engineer selects "Add new task" (Figure 2)
4. The system prompts the Receptionist to enter the name and cost of new work
5. The Reception Engineer enters the details of the new job and selects "Save" (Figure 3)
6. The system displays an informational message and returns to the updated screen task list screen

Alternative Flow

- 3.1.1 The Reception Engineer selects to modify a job (Figure 4) 3.1.2 The system displays a screen with the details of the selected job
- 3.1.3 The Reception Engineer changes the elements he wishes and selects "Save" (Figure 5)
- 3.1.4 The system displays an informational message and returns the Reception Engineer to the screen with the updated task list

Comments

The part page displays various information such as what is currently in stock, date of last purchase, who is the supplier of the part as well as the 'Change Stock' option. The Parts option is available to both the supervising engineer and the owner.



Σύστημα CarOps		Λίστα Αποθηκευμένων Εργασιών	
Όνομα Επισκευής	Κόστος	Κωδικός Εργασίας	
KTEO	60€	A1	<input type="button" value="Επεξεργασία"/>
Αλλαγή Καθρεύτη	30€	A2	<input type="button" value="Επεξεργασία"/>
Αλλαγή Λάστιχα	40€	C1	<input type="button" value="Επεξεργασία"/>
Αλλαγή Κλειδιού	30€	C2	<input type="button" value="Επεξεργασία"/>
Αλλαγή Φανάρια	15€	D1	<input type="button" value="Επεξεργασία"/>

Σύστημα CarOps		Δημιουργία Εργασίας	
Όνομα Επισκευής	Κόστος	Αλλαγή Οροφής	
		<input type="button" value="60€"/>	

Σύστημα CarOps		Λίστα Αποθηκευμένων Εργασιών	
Όνομα Επισκευής	Κόστος	Κωδικός Εργασίας	
KTEO	60€	A1	<input type="button" value="Επεξεργασία"/>
Αλλαγή Καθρεύτη	30€	A2	<input type="button" value="Επεξεργασία"/>
Αλλαγή Λάστιχα	40€	C1	<input type="button" value="Επεξεργασία"/>
Αλλαγή Κλειδιού	30€	C2	<input type="button" value="Επεξεργασία"/>
Αλλαγή Φανάρια	15€	D1	<input type="button" value="Επεξεργασία"/>

Σύστημα CarOps		Τροποποίηση Εργασίας	
Όνομα Επισκευής	Κόστος	Αλλαγή Οροφής	
		<input type="button" value="60€"/>	



9. "View Assignments" Basic Flow

1. The Engineer selects View Assignments from the Home Screen (Figure 1)
2. The system displays a list of active assignments
3. Engineer selects completion for desired assignment (Figure 2)
4. The system displays a form to fill in the time and
spare parts
5. The Engineer clicks <<Add Parts>> (Figure 3)
6. The system displays a list of available spare parts and their parts
(Figure 4)
7. The Engineer selects a spare part from the list, types it
quantity and press <<Add>>
8. The system displays the list of spare parts added by
Engineer
9. The Engineer presses confirm completion (Figure 5)
10. The system returns the Engineer to the Assignments screen and displays an
informational message (Figure 6)

Comments

All Engineers have the ability to view the tasks assigned to them and complete them by recording the parts used and the time the task was completed

Σύστημα CarOps	Αρχική Οθόνη	— □ ×
Σύνδεση ως Μηχανικός		
Λίστα Επιλογών		
Προβολή Αναθέσεων		

Σύστημα CarOps	Ενεργές Αναθέσεις	— □ ×
Κωδικός Επισκευής	Όνομα Εργασίας	Υπεύθυνος
1	Αντικατάσταση Λυχνία	Επιβλέπων Μηχανικός Β
2	Αλλαγή Λάστιχων	Επιβλέπων Μηχανικός Α
Ολοκλήρωση		
Επιστροφή		

Σύστημα CarOps	Καταγραφή Στοιχείων Ανάθεσης	— □ ×
Όνομα Εργασίας : Αντικατάσταση Λυχνία Υπεύθυνος Μηχανικός : Επιβλέπων Μηχανικός Β		
Κωδικός Επισκευής : 1		
Εισαγωγή Χρόνου Ολοκλήρωσης : <input type="text" value="30 λεπτά"/>		
Ανταλλακτικά που Χρησιμοποιήθηκαν		
Τύπος Ανταλλακτικού	Ποσότητα	
<input type="button" value="Επιστροφή"/> <input type="button" value="Προσθήκη Ανταλλακτικού"/> <input type="button" value="Επιβεβαίωση Ολοκλήρωσης"/>		

Σύστημα CarOps	Καταγραφή Στοιχείων Ανάθεσης	— □ ×
Όνομα Εργασίας : Αντ	Λίστα Διαθέσιμων Ανταλλακτικών	τον Μηχανικός Β
Εισ	<input type="text" value="Αναζήτηση"/>	
Τύπο	1.Ανταλλακτικό Α - 4 Τεμάχια 2.Ανταλλακτικό Β -10 Τεμάχια 3.Ανταλλακτικό Γ - 6 Τεμάχια 4.Ανταλλακτικό Δ -15 Τεμάχια 5.Λυχνία Αυτοκινήτου - 5 Τεμάχια	
<input type="button" value="Επιστροφή"/> <input type="button" value="Ποσότητα : 1"/> <input type="button" value="Προσθήκη"/> <input type="button" value="Ολοκλήρωσης"/>		

Σύστημα CarOps	Καταγραφή Στοιχείων Ανάθεσης	— □ ×
Όνομα Εργασίας : Αντικατάσταση Λυχνία Υπεύθυνος Μηχανικός : Επιβλέπων Μηχανικός Β		
Κωδικός Επισκευής : 1		
Εισαγωγή Χρόνου Ολοκλήρωσης : <input type="text" value="30 λεπτά"/>		
Ανταλλακτικά που Χρησιμοποιήθηκαν		
Τύπος Ανταλλακτικού	Ποσότητα	
Λυχνία Αυτοκινήτου	1	
<input type="button" value="Επιστροφή"/> <input type="button" value="Προσθήκη Ανταλλακτικού"/> <input type="button" value="Επιβεβαίωση Ολοκλήρωσης"/>		

Σύστημα CarOps	Ενεργές Αναθέσεις	— □ ×
Κωδικός Επισκευής	Όνομα Εργασίας	Υπεύθυνος
2	Άλλαγή Λάστιχων	Επιβλέπων Μηχανικός Α
		<input type="button" value="Ολοκλήρωση"/>
Η ανάθεση ολοκληρώθηκε και αφαιρέθηκε με επιτυχία !		
<input type="button" value="Επιστροφή"/>		



10. "Manage Available Repairs" Basic Flow

1. Supervising Engineer selects Repair Management (Figure 1)
2. The system displays a screen with 2 options: "Repairs I'm Managing" and "Repairs Pending"
3. The Supervising Engineer selects "Repairs I Manage" (Figure 2)
4. The system displays the repairs for which the Supervisor is responsible
5. The Supervising Engineer selects one of the Repairs to view more information (Figure 3)
6. The system will show him information about the repair he chose
7. The Supervising Engineer chooses to change the assignment for the work he wishes (Figure 4)
8. The system displays a list of available Engineers for the Supervising Engineer to assign (Figure 5)
9. The Supervising Engineer once all work is completed will select repair completion (Figure 6)
10. The system will display a confirmation message for the parts used and return to the repair management screen

Alternative Flow

- 3.1.1 Supervising Engineer selects "Pending Repair List" (Figure 7)
- 3.1.2 The system displays a list of pending repairs
- 3.1.3 The Supervising Engineer selects a repair for more information (Figure 8)
- 3.1.4 The system displays details of the selected repair
- 3.1.5 Supervising Engineer selects Repair Management (Figure 9)
- 3.1.6 The system displays a confirmation message and returns to the screen repair management (Figure 10)

Comments The Supervising Engineer will be considered free when he is not managing any repairs and there are none on the pending repairs list. The Superintending Engineer will track repair completion progress by seeing which assignments have been completed and which parts have been used.

Σύστημα CarOps	Αρχική Οθόνη	— □ ×
Σύνδεση ως Επιβλέπων Μηχανικός		
Λίστα Επιλογών		
<input type="button" value="Διαχείριση Επισκευών"/> <input type="button" value="Λίστα Ανταλλακτικών"/> <input type="button" value="Προβολή Αναθέσεων"/>		

Σύστημα CarOps	Διαχείριση Επισκευών	— □ ×
Σύνδεση ως Επιβλέπων Μηχανικός		
Λίστα Επιλογών		
<input type="button" value="Επισκευές που Διαχειρίζομαι"/> <input type="button" value="Επισκευές σε Αναμονή"/>		

Σύστημα CarOps	Λίστα διαχειρίσιμων Επισκευών	— □ ×		
Κωδικός Επισκευής	Κωδικός Φακέλου	Εργασίες	Κατάσταση	
1	3	3	Σε εξέλιξη	<input type="button" value="Προβολή"/>
2	5	2	Καμία Ανάθεση	<input type="button" value="Προβολή"/>

Σύστημα CarOps Διαχείριση Επισκευής _1

Εργασία	Ανάθεση	Τύπος και Ποσότητα	Χρόνος	Κατάσταση
Ευθυγράμμιση Τροχών	Μηχανικός A	Σύνοψη Ανταλλακτικών : 4 Λάστιχα		
Αλλαγή Λάστιχων	Επιβλέπων Μηχανικός A	Ολοκλήρωση		
Αλλαγή Λαδιών	Μηχανικός B	Ακύρωση	Ολοκλήρωση	Ολοκληρώθηκε
				<input type="button" value="Αλλαγή"/>
				<input type="button" value="Ανάθεση"/>
				<input type="button" value="Ολοκλήρωση Επισκευής"/>
<input type="button" value="Επίστροφη"/>				

Σύστημα CarOps Διαχείριση Επισκευής _1

Εργασία	Ανάθεση	Τύπος και Ποσότητα	Χρόνος	Κατάσταση
Ευθυγράμμιση Τροχών	Μηχανικός A	-	-	Ανατέθηκε
Αλλαγή Λάστιχων	-	-	-	Δεν ανατέθηκε
Αλλαγή Λαδιών	Μηχανικός B	-	20 λεπτά	Ολοκληρώθηκε
<input type="button" value="Επίστροφη"/>		<input type="button" value="Ολοκλήρωση Επισκευής"/>		

Σύστημα CarOps Διαχείριση Επισκευής _1

Εργασία	Ανάθεση	Τύπος και Ποσότητα	Χρόνος	Κατάσταση
Ευθυγράμμιση Τροχών	Μηχανικός A	-	1 ώρα	Ολοκληρώθηκε
Αλλαγή Λάστιχων	Επιβλέπων Μηχανικός A	Λάστιχα , 4	30 λεπτά	Ολοκληρώθηκε
Αλλαγή Λαδιών	Μηχανικός B	-	20 λεπτά	Ολοκληρώθηκε
<input type="button" value="Επίστροφη"/>		<input type="button" value="Ολοκλήρωση Επισκευής"/>		

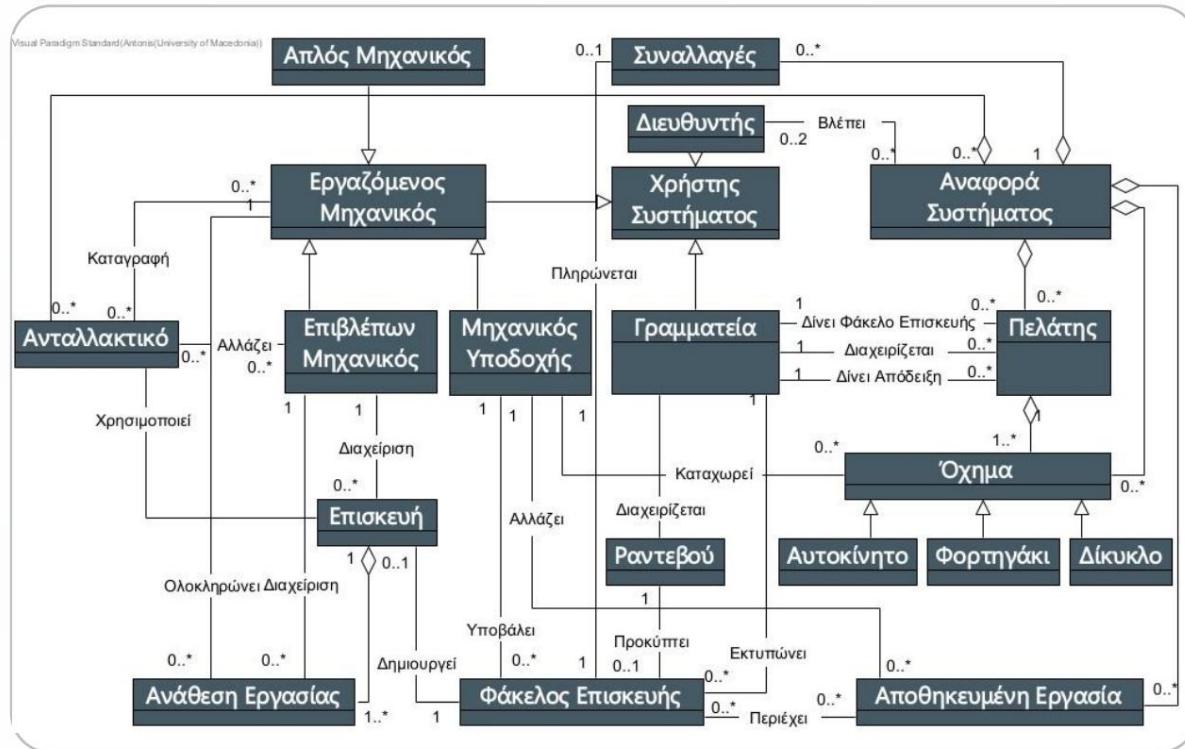
Σύστημα CarOps	Λίστα Επισκευών σε Αναμονή				— □ ×
Κωδικός Επισκευής	Κωδικός Φακέλου	Εργασίες	Κατάσταση		
3	6	3	Σε αναμονή	<input type="button" value="Προβολή"/>	
5	8	2	Σε αναμονή	<input type="button" value="Προβολή"/>	

Σύστημα CarOps	Στοιχεία Επισκευή_3	— □ ×
Κωδικός Επισκευής : 3		
Κωδικός Φακέλου Επισκευής : 6		
<u>Λίστα Εργασιών</u>		
1 Ευθυγράμμιση Τροχών 2 Αλλαγή Λάστιχων 3 Αλλαγή Λαδιών		
<input type="button" value="Επίστροφη"/>		<input type="button" value="Διαχείριση Επισκευής"/>

Σύστημα CarOps	Στοιχεία Επισκευή_3	— □ ×
Κωδικός Επισκευής : 3		
Κωδικός Φακέλου Επισκευής : 6		
Επιβεβαίωση Διαχείρισης Νέας Επισκευής		
<input type="button" value="Ακύρωση"/> <input type="button" value="Επιβεβαίωση"/>		
<input type="button" value="Επίστροφη"/>		<input type="button" value="Διαχείριση Επισκευής"/>

Σύστημα CarOps	Διαχείριση Επισκευών	— □ ×
Λίστα Επιλογών		
<input type="button" value="Επισκευές που Διαχειρίζομαι"/> <input type="button" value="Επισκευές σε Αναμονή"/>		
Η διαχείριση νέας επισκευής προστέθηκε με επιτυχία!		
<input type="button" value="Επιστροφή"/>		

3.6 Domain Model





PLANNING

- 4.1 Verbal Descriptions & Sequence Diagrams
- 4.2 Class Diagram
- 4.3 Object Diagram



4.1 Verbal Descriptions & Sequence Diagrams

Briefly

Using verbal descriptions and sequence diagrams, we can understand and precisely define the functions and interactions between system components during the design phase.

Verbal descriptions provide a linguistic representation of the processes and actions that take place between components, while sequence diagrams provide a graphical representation of the interaction between units. Through these methods, we can identify the time sequence of actions, detect potential errors and improve the design of the CarOps system.

Contents

1. Flow "Print Repair Folder" Verbal Description
2. Flow "Customer Tab" Verbal Description
3. "Book an Appointment" Stream Verbal Description
4. "Vehicle Tab" Stream Verbal Description
5. Flow "Generate Repair Folder" Verbatim Description
6. Stream "View Reports" Verbal Description
7. Flow "Modify Parts List" Verbatim Description
8. Flow "Modify Task List" Verbatim Description
9. Stream "View Assignments" Verbal Description
10. Stream "Manage Available Repairs" Verbal Description
1. Flow "Print Repair Folder" Sequence Chart
2. Flow "Customer Tab" Sequence Chart
3. Flow "Book an Appointment" Sequence Chart
4. Flow "Vehicle Tab" Sequence Diagram
5. Flow "Generate Repair Folder" Sequence Diagram
6. Flow "View Reports" Sequence Chart
7. Flow "Change Parts List" Sequence Diagram
8. Flow "Modify Task List" Sequence Diagram
9. Flow "View Assignments" Sequence Chart
10. Flow "Manage Available Repairs" Sequence Diagram



1. "Print Repair Folder" Basic Flow

1. The secretariat from its home screen presses the "View Repair Folder List" button
 - 1.1. System calls RepairsScr screen
 - 1.1.1. The repair screen calls up the repair catalog to populate the list repairs
 - 1.1.1.1. The report directory returns all repair folders
 - 1.1.1.1.1. The repair screen shows the list and a search box
2. The clerk from the repair screen taps the desired repair
 - 2.1 The system calls the repair detail screen (DetailRepairScr)
 - 2.1.1. The detail screen shows detailed information and the button "Printing"
4. Clerk from details screen press 'Print' button
 - 4.1 The system downloads the print document
 - 4.1.1. The system displays a success message
 - 4.1.1.1. View the printable document on the screen

Alternative Flow 2

3. The clerk from the repair screen searches for the desired repair and selects it
 - 3.1 The system calls the repair detail screen (DetailRepairScr)
 - 3.1.1. The detail screen shows detailed information and the button "Printing"



2. Basic "Customer Tab" Flow

1. The secretariat from its home screen presses the "Customers" button
 - 1.1.The system calls the clients screen (ClientsListScr)
 - 1.1.1 The customer screen calls the customer list to populate the customer list
 - 1.1.1.1 The customer list returns all customers
 - 1.1.1.1.T The customer screen shows a list of all customers and "Create New Tab" and "Modify" buttons
2. The secretary from the customer screen presses the "Create New Tab" button
 - 2.1. The system calls the data filling screen (VehicleForm)
 - 2.1.T The screen shows a data filling form and the "Finish" button
3. The secretary from the data filling screen enters the required fields and presses the finish button
 - 3.1 The system with these fields creates a new Customer element and displays a message
 - 3.1.1.T The message "Customer Card Creation Successful" appears on the screen

Alternative Underflow of 1

2. The secretary from the customer screen presses the "Modify" button
 - 2.1 The system calls the data filling screen (VehicleForm)
 - 2.1.T The screen shows a data filling form and the "Finish" button
3. The secretariat from the data entry screen updates the required fields and press the "Finish" button
 - 3.1 The system with these fields updates the customer record and displays a message
 - 3.1.1.T The message "Customer Card Processing Successful" appears on the screen



3. Basic "Book Appointment" Flow

1. The secretariat from its home screen presses the "Appointment" button
 - 1.1.The system calls the appointment screen (SessionScr)
 - 1.1.1. The appointment screen calls up the appointment directory to populate the list appointment
 - 1.1.1.1. The appointment list returns a list of all scheduled appointments
 - 1.1.1.1.1. The appointment screen shows the "New Appointment" button and a list of all with appointments where next to each item there are buttons "Edit" and "Delete"
2. The secretary from the appointment screen presses the "New Appointment" button
 - 2.1 The system calls the appointment information filling screen (EditSessionScr)
 - 2.1.T The screen shows a data filling form and the "Finish" button
3. The secretary from the data filling screen enters the required fields and presses the finish button
 - 3.1 The system with these fields creates a new appointment item and displays a message
 - 3.1. T The screen shows the message "Successful Appointment Creation"

1 the Alternative Undercurrent of 1

4. The secretary from the appointment screen presses the "Edit" button
 - 4.1 The system calls the appointment information filling screen (EditSessionScr)
 - 4.1.T appears a form to fill in details and the "Finish" button
5. The secretariat from the details filling screen updates the required fields and press the "Finish" button
 - 5.1 The system with these fields updates the appointment details and displays a message
 - 5.1.T The message "Successful Appointment Processing" appears on the screen

2 the Alternative Undercurrent of 1

6. The secretary from the appointment screen presses the "Delete" button
 - 6.1 The system deletes the entry of the selected appointment and displays a message
 - 6.1.T The message "Successfully Deleted Appointment" appears on the screen



4. "Vehicle Tab" Basic Flow

1. The secretariat from its home screen presses the "Vehicles" button
 - 1.1. System calls the vehicle screen (VehiclesListScr)
 - 1.1.1. The vehicle screen calls up the vehicle catalog to populate the list vehicles
 - 1.1.1.1. The vehicle list returns a list of all vehicles
 - 1.1.1.1.1. The vehicle screen shows the "New Tab" button and all the list of the vehicle tabs where next to each item there is edit button
2. The secretary from the vehicle screen presses the "New Tab" button
 - 2.1 The system calls the vehicle type selection screen (TypesScr)
 - 2.1.T The screen shows a list of "Car" "Truck" and "Bike" type
3. The secretariat from the selection screen selects the type of vehicle
 - 3.1. The system calls the data filling screen (VehicleForm) where it displays the corresponding fields depending on the type
 - 3.1.1.T. The screen shows a form to fill in details and the "Finish" button
4. The secretary from the data filling screen enters the required fields and presses the finish button
 - 4.1 The system with these fields creates a new tab item and displays a message
 - 4.1.T The message "Vehicle Card Creation Successful" appears on the screen

Alternative Underflow of 1

7. The secretary from the vehicle screen presses the "Edit" button
 - 7.1 The system calls the data filling screen (VehicleForm) where it depends on the type displays the corresponding fields
 - 7.1.1.T The screen shows a data filling form and the "Finish" button
8. The secretary from the data filling screen enters the required fields and presses the finish button
 - 8.1 The system with these fields updates the vehicle record and displays a message
 - 8.1.T The message "Vehicle Card Processing Successful" appears on the screen



5. "Create Repair Folder" Basic Flow

1. The host engineer from the home screen clicks the "Repair Folders" button
 - 1.1. System calls the repair folder screen (FolderScr)
 - 1.1.1. The called screen calls up the repair catalog to populate the list
 - 1.1.1.1. repair
 - 1.1.1.1.1. Directory returns a list of all repair folders
 - 1.1.1.1.1.1. T The "Create New" button can be seen on the repair folder screen
 - 1.1.1.1.1.2. "Folder" and a list of all repair folders so far
2. The host engineer from the repair folder presses the button "Create New folder"
 - 2.1 The system calls the vehicle data completion screen (VehicleScr)
 - 2.1. T A search bar and a "Search" button appear on the screen
3. The reception engineer from the fill in vehicle details screen searches for the vehicle with license plate number and press "Search"
 - 3.1. The system asks the vehicle list to search for the vehicle
 - 3.1.1. The directory returns a true value depending on the success of the search
 - 3.1.1.1. T The screen displays the message "Not Registered" or "Registered"
4. If the vehicle is not registered then the basic Vehicle Tab flow is called, if it is registered the host engineer selects it
 - 4.1. The system displays the details of the vehicle created or selected
 - 4.1.1. T The screen shows all the details of the vehicle and the "Next" button
6. The host engineer selects "Next"
 - 6.1 The system calls the task screen (TasksMenu)
 - 6.1.1. The task screen calls the task list to populate the task list
 - 6.1.1.1. Task list returns all ready tasks
 - 6.1.1.1.1. T On the screen appears a list of all ready tasks, a button "Input" and a "Time Estimate" field
 7. The host engineer from the tasks screen selects the tasks that will be needed and press "Import"
 - 7.1. The system with these tasks creates a new task list object and displays message
 - 7.1.1. T The screen shows a message "Successful Addition of Tasks"
 8. The host engineer from the tasks screen fills in the "Time Estimate" field
 - 8.1. The system with the list of repair files updates the time field of the repair
 - 8.1.1. The system calls the folder preview screen (FolderReviewScr)
 - 8.1.1.1. The preview of the repair file is shown on the screen



6. Basic Flow "View Reports"

1. The owner from their home screen taps the "Reports" button
 - 1.1 The system calls the options screen (ReportsSelectScr)
 - 1.1.T Options screen shows the 'Income/Expense Reports' paddles and "Parts Reports"
2. The owner from the options screen taps the "Income/Expense Reports" button
 - 2.1 The system calls the reports screen (ReportsListScr)
 - 2.1.1 The report screen calls the report catalog to populate reports
 - the list checks the
 - 2.1.1.1 reports and returns only income/expenses
 - 2.1.1.1.T A list of reports appears on the reports screen income/expenses
4. From the reports screen, the owner selects the report he wants
 - 4.1 The system calls the detailed report screen (ReportScr)
 - 4.1.T The report details are shown on the screen

Alternative Flow 2

3. From the options screen, the owner presses the "Parts Reports" button
 - 3.1 The system calls the reports screen (ReportsListScr)
 - 3.1.1 The reports screen calls the directory to populate the list reports
 - 3.1.1.1 The report directory checks the reports and returns only parts of
 - 3.1.1.1.T The screen shows the report screen with a list of spare parts references



7. "Modify Parts List" Basic Flow

1. The supervising engineer from the home screen presses the "Parts" button
 - 1.1.The system calls the parts type screen (PartsTypeScr)
 - 1.1.1. The called screen calls up the list of part types to populate her list of spare parts types
 - 1.1.1.1. Catalog returns a list of all part types
 - 1.1.1.1.T On the parts types screen, all types and the button are shown "Next"
2. The supervising engineer from the parts types screen selects the types parts and then "Next"
 - 2.1 The system calls the parts screen (PartsScr) according to the selected formula
 - 2.1.1. The parts screen calls up the parts catalog to populate the list spare parts
 - 2.1.1.1. The vehicle catalog returns the parts belonging to the specific type
 - 2.1.1.1.T The screen shows a list of requested spare parts
3. The supervising engineer from the parts screen a part
 - 3.1 System calls the parts detail screen (DetailPartScr)
 - 3.1.T. The details screen shows all the details of the part and the button "Inventory Change"
4. The supervising engineer from the detail screen taps "Modify Part"
 - 4.1 The system calls the edit quantity screen (EditQuantityScr)
 - 4.1.T. The screen shows a field to fill in and the "Save" button
5. The supervising engineer from the inventory modification screen updates the inventory and presses "Save"
 - 5.1.The system with the spare parts list updates the inventory and displays a message
 - 5.1.T A successful update message appears on the screen



8. "Modify Task List" Basic Flow

1. The host engineer from the home screen presses the "Task List" button
 - 1.1 System calls task screen (SavedTasksScr)
 - 1.1.1. The task screen calls up the task list to populate the list tasks
 - 1.1.1.1. Directory returns a list of all tasks
 - 1.1.1.1.T On the tasks screen you can see an "Add New Task" button, a list of all tasks and next to each item the button "Amendment"
2. The host engineer from a task screen clicks the "Add New Task" button
 - 2.1 The system calls the task creation screen (CreateTaskScr)
 - 2.1.T The screen shows empty fields to fill in and the "Save" button
3. The host engineer from the job creation screen fills in the fields and clicks "Save"
 - 3.1. The system through the task list creates a new task and displays a message
 - 3.1.T The message "Successful Job Creation" appears on the screen

Alternative Underflow of 1

4. The host engineer from a task screen presses the "Modify" button
 - 4.1 The system calls the edit task screen (EditTaskScr)
 - 4.1.T The screen shows the job fields and the "Save" button
5. The host engineer from the job processing screen updates the fields and clicks "Save"
 - 5.1. The system through the task list updates the new task and displays a message
 - 5.1.T The message "Job Processing Successful" appears on the screen



9. "View Assignments" Basic Flow

1. The engineer from the home screen presses the "View Assignments" button
 - 1.1. The system calls the assignments screen (AssignmentsScr)
 - 1.1.1. The assignment screen calls up the assignment list to populate the list assignments
 - 1.1.1.1. Directory returns a list of tasks assigned to engineer
 - 1.1.1.1.1. The assignments screen shows a list of assignments and next to each assignment the "Complete" button
2. If a task has been completed, the engineer from the tasks screen presses "Completion"
 - 2.1 The system calls the mission completion screen (MissionLog)
 - 2.1.1. The completion screen calls up the parts catalog to populate it list with all accessories
 - 2.1.1.1. On the completion screen, a work duration field is shown, a list of all the spare parts and the "Finish" button
3. The engineer from the task completion screen fills in the fields and presses "Finish"
 - 3.1. The system stores the repair duration
 - 3.2 The system through the job list saves the list of used ones spare parts
 - 3.3. The system calls the assignments screen and displays a message
 - 3.3.1. The assignment screen does not show the assignment and a success message is displayed completion



10. "Manage Available Repairs" Basic Flow

1. The Supervising engineer from the home screen presses the "Manage Repairs" button
 - 1.1 The system calls the selection screen (RepairSelectScr)
 - 1.1.T On the options screen, the buttons "My Repairs" and "Repairs to Stand by"
2. The Supervising Engineer from the options screen taps the "My Repairs" button
 - 2.1 The system calls the repair screen (RepairListScr)
 - 2.1.1 The repair screen calls up the repair catalog to populate the list repairs
 - 2.1.1.1 The repair list returns the repairs undertaken by the supervisor engineer
 - 2.1.1.1.T The repairs screen shows a list of repairs he has undertaken supervising engineer
3. The supervising engineer selects the desired repair from the repair screen
 - 3.1.The system calls the detailed repair screen (RepairScr)
 - 3.1.1. The repair details screen calls up the assignment list to display the list of engineers assigned to this task
 - 3.1.T The screen shows the details of the repair, a list of engineers and the 'Change Assignment', 'Complete' and 'Exit' buttons
4. The supervising engineer from the repair screen presses the "Change Assignment" button
 - 4.1 System Call Engineers Screen (EngineerListScr)
 - 4.1.1 The mechanics screen calls the mechanics list to populate the mechanics list
 - 4.1.1.1 The engineers list returns the names of all engineers
 - 4.1.1.1.T The mechanics screen shows a list of all of them engineers
5. The supervising engineer from the engineer screen selects engineers for the repair and after "End"
 - 5.1. The system updates the details of the repair in the catalog and displays a message
 - 5.1.T A successful processing message appears on the repair screen

Alternative Underflow of 1

8. The Supervising Engineer from the options screen taps the "Pending Repairs" button
 - 8.1 System calls the repair screen (RepairListScr)
 - 8.1.1 The repair screen calls the repair catalog to populate the list repairs
 - 8.1.1.1 The repair list returns repairs that have not been undertaken by anyone
 - 8.1.1.1.T The repairs screen shows a list of unassigned repairs
9. The supervising engineer selects the desired repair from the repair screen

9.1 System calls detailed repair screen (RepairScr)

9.1.T The screen shows the details of the repair and the "Take over" button

10. The supervising engineer from the repair screen presses the "Take over" button

10.1 The system updates the details of the repair in the catalog and displays a message

10.1.T A successful processing message appears on the repair screen

1 the Alternative Undercurrent of 3

6. The supervising engineer from the repair screen presses the "Finish" button

6.1 The system updates the repair as complete in the repair list 6.1.T A successful processing message appears on the repair screen

2 the Alternative Undercurrent of 3

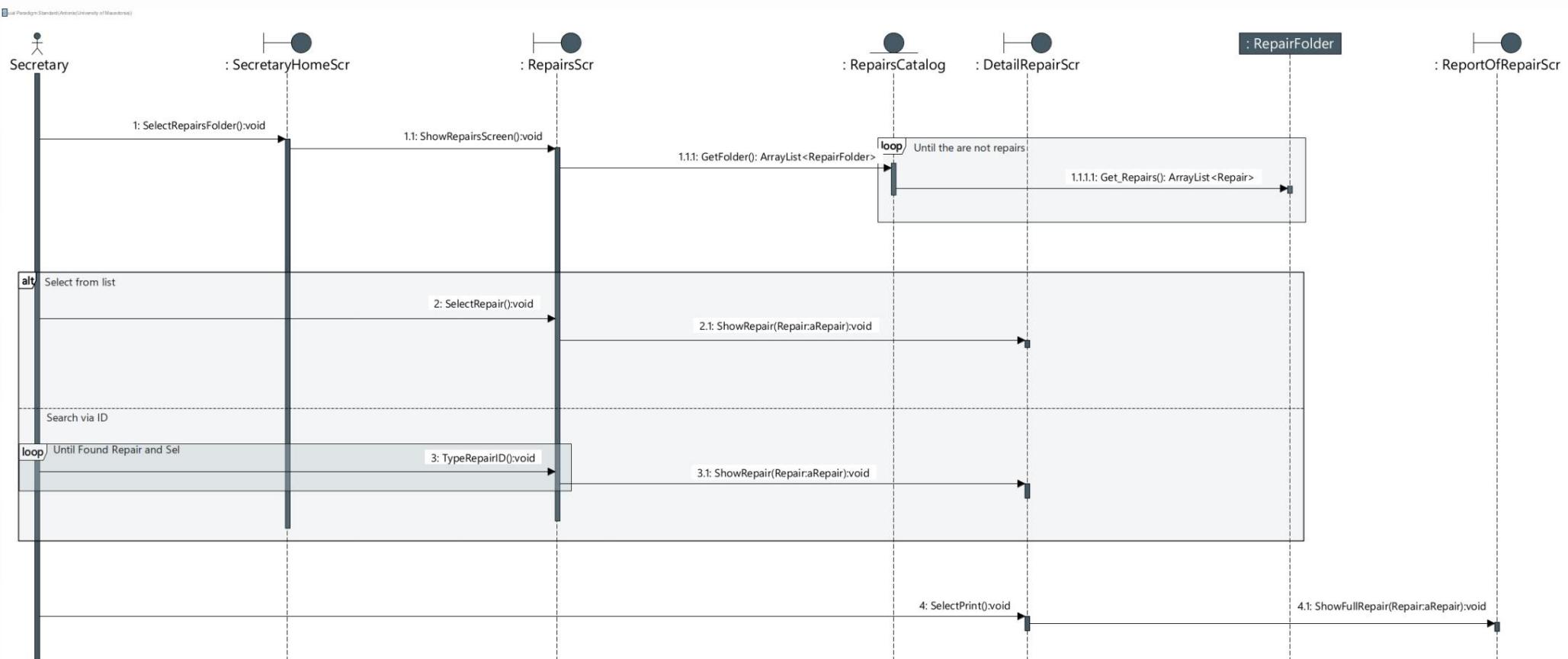
7. The supervising engineer from the repair screen presses the "Exit" button

7.1 The system updates the repair as not assigned to the repair list 7.1.T A successful processing message appears on the repair screen

1. "Print Repair Folder" Basic Flow



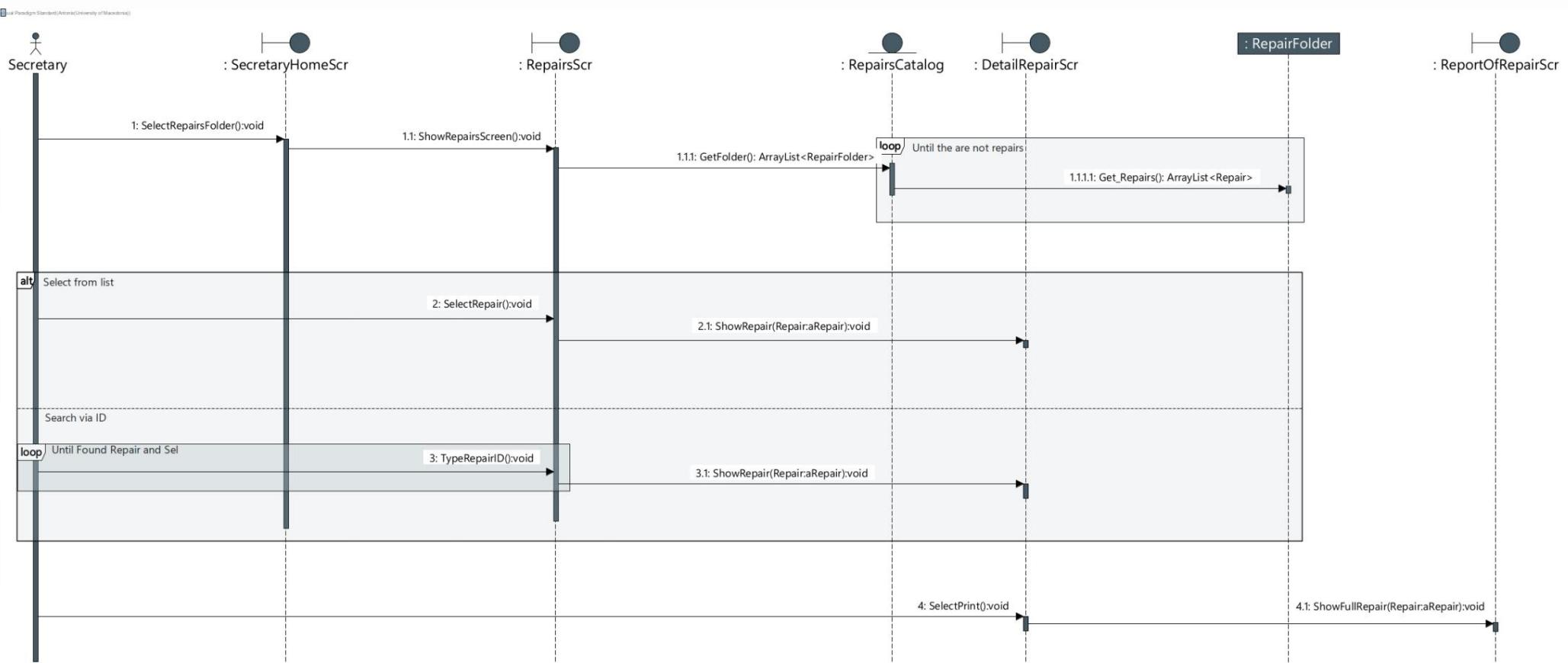
1. The secretariat from its home screen presses the "View Repair Folder List" button
 - 1.1. The system calls the repair folder screen (RepairsScr)
 - 1.1.1. The repair screen calls up the repair catalog to populate the repair list
 - 1.1.1.1. The report directory returns all repair folders
 - 1.1.1.1.T The repair screen shows the list and a search box



1. "Print Repair Folder" Basic Flow



2. The clerk from the repair screen taps the desired repair
 - 2.1 The system calls the repair detail screen (DetailRepairScr)
 - 2.1.T The detail screen shows detailed information and the "Print" button



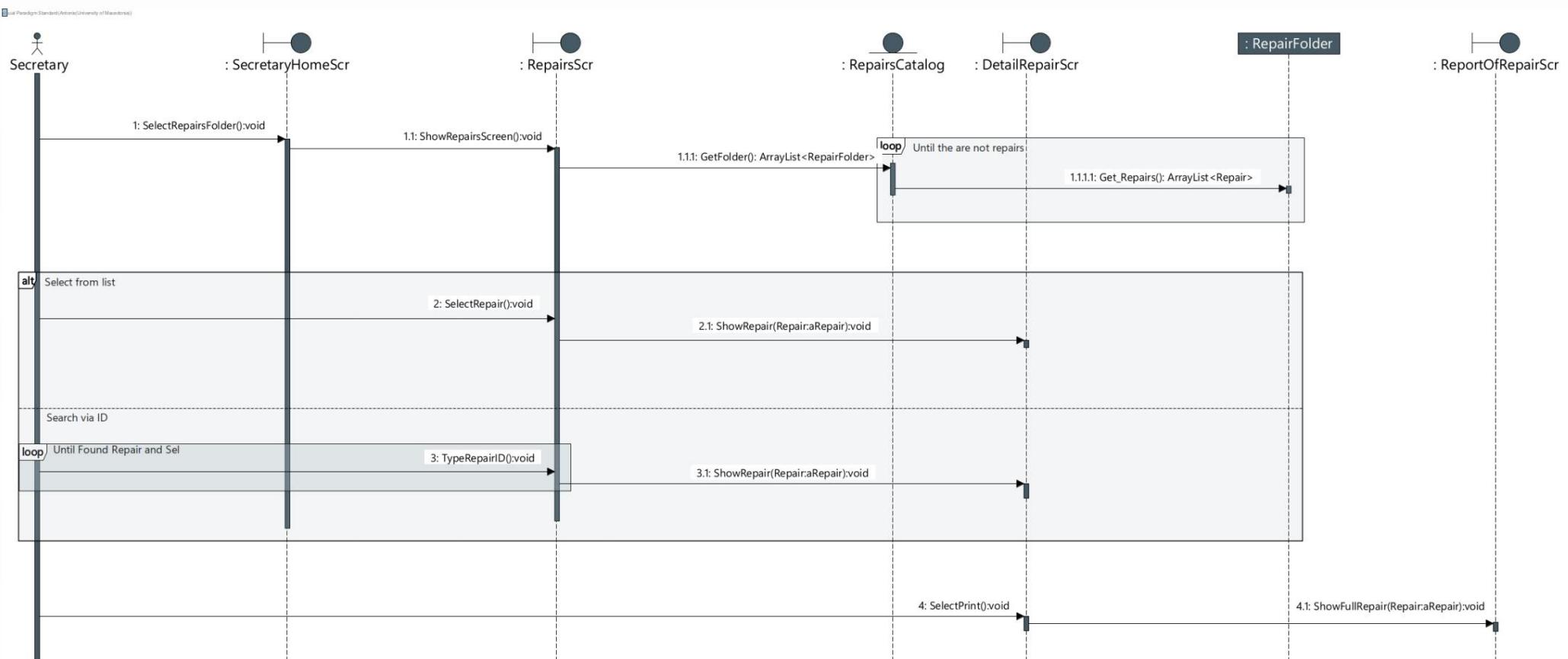
1. Basic Flow "Print Repair Folder" - Alternative Flow



3. The clerk from the repair screen searches for the desired repair and selects it

3.1 The system calls the repair detail screen (DetailRepairScr)

3.1.T The detail screen shows detailed information and the "Print" button



1. "Print Repair Folder" Basic Flow

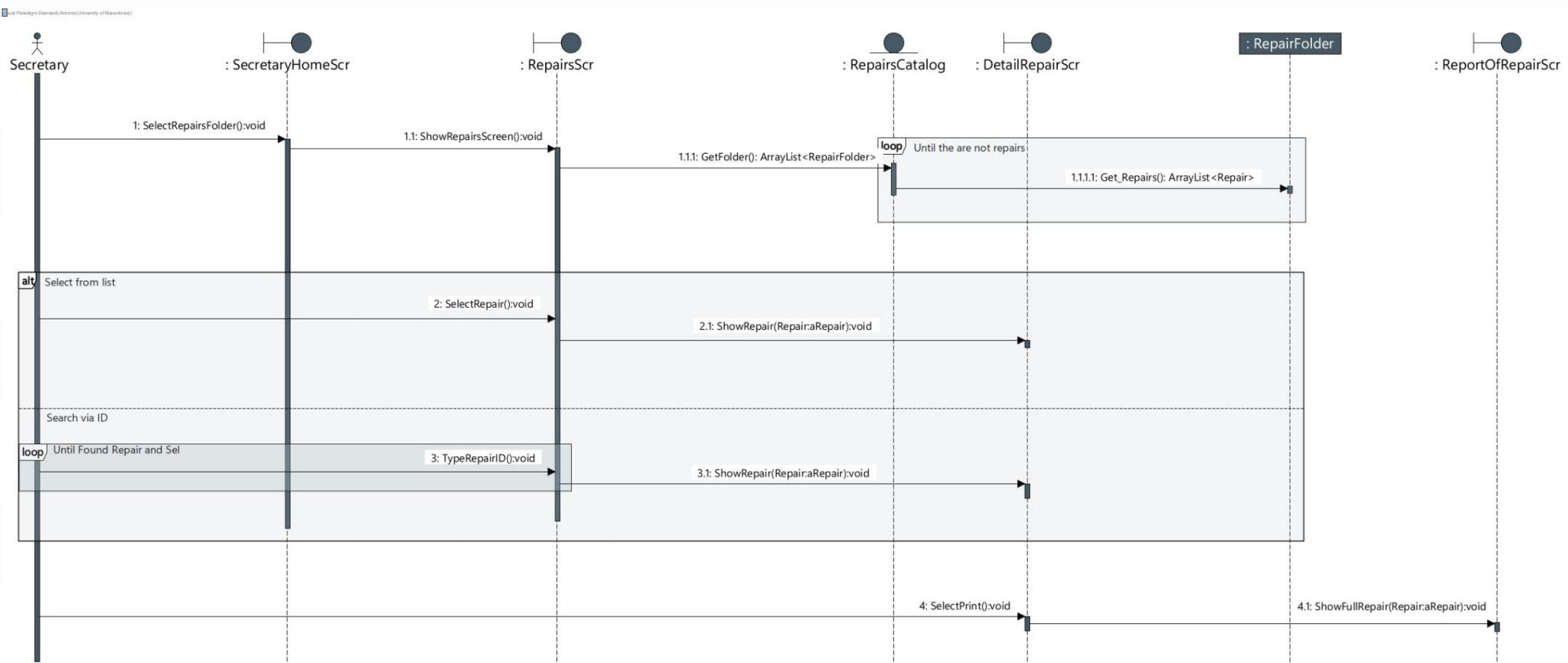


4. Clerk from details screen press 'Print' button

4.1 The system downloads the print document

4.1.1 The system displays a success message

4.1.1.T On the screen view the printable document



2. Basic "Customer Tab" Flow



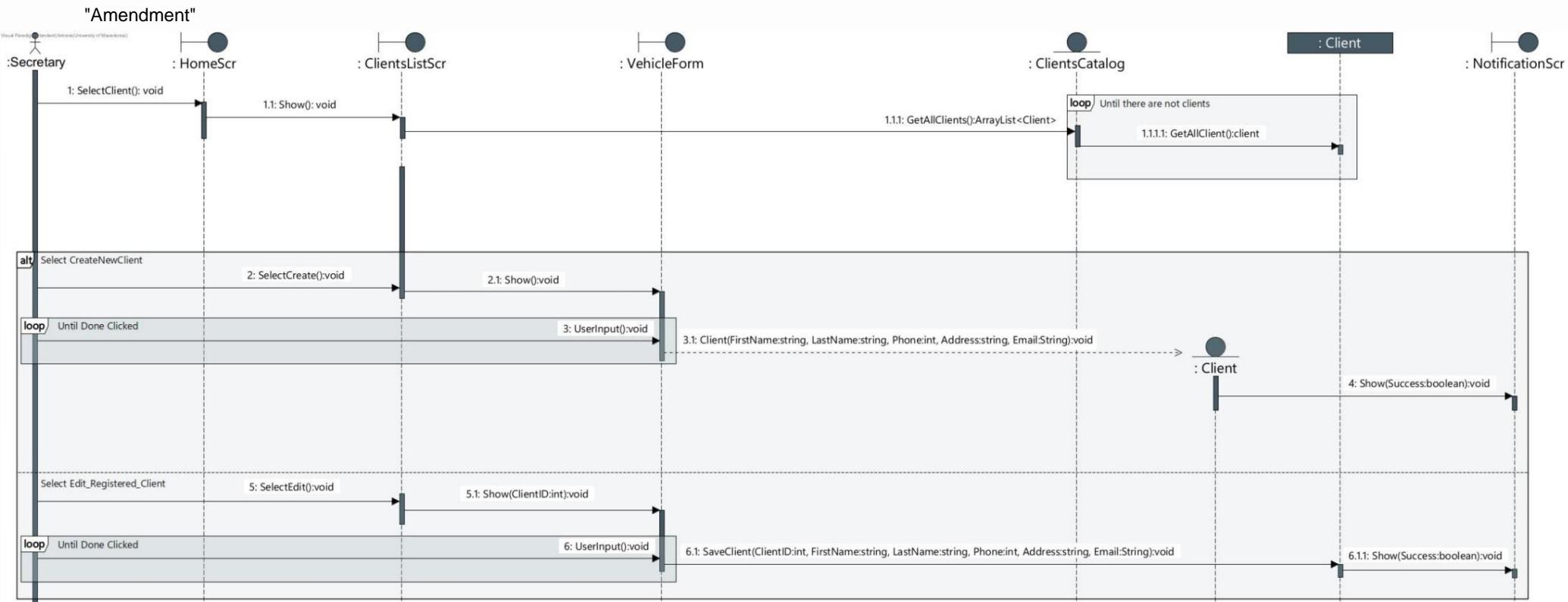
1. The secretariat from its home screen presses the "Customers" button

1.1. The system calls the clients screen (ClientsListScr)

1.1.1 The customer screen calls the customer directory to populate the customer list

1.1.1.1 The customer list returns all customers

1.1.1.1.T The customer screen shows a list of all customers and the buttons "Create New Tab" and



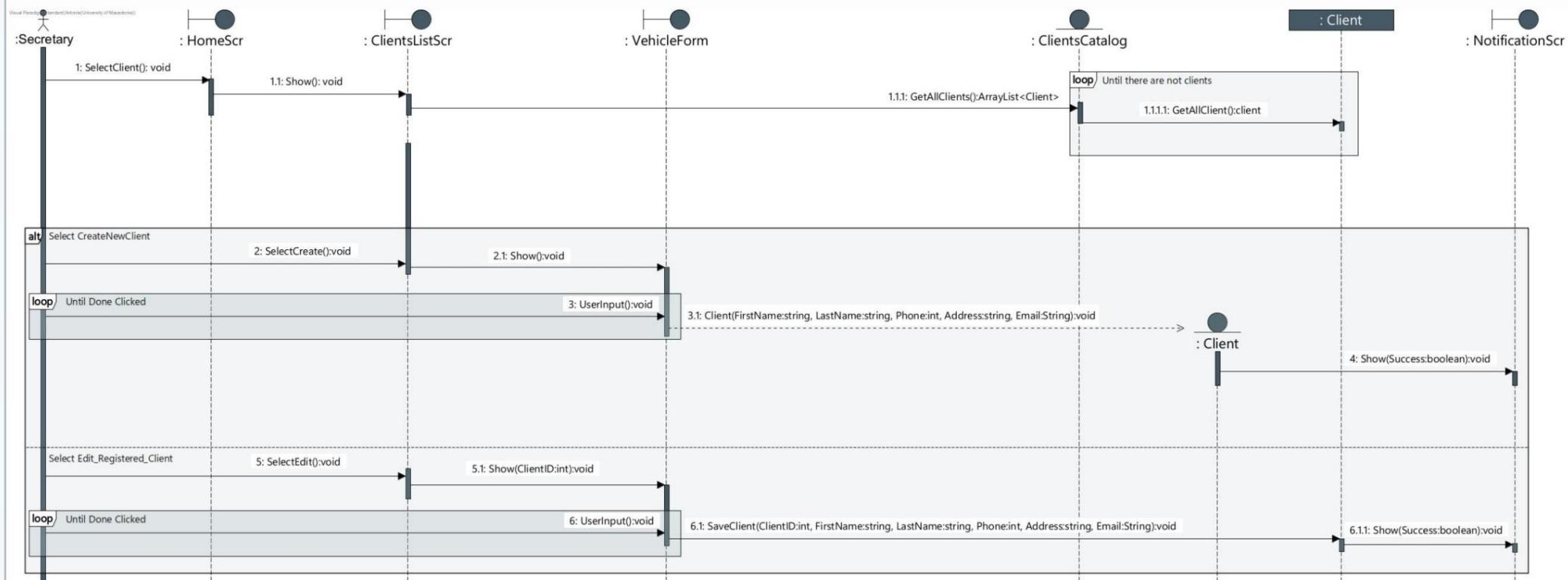
2. Basic "Customer Tab" Flow



2. The secretary from the customer screen presses the "Create New Tab" button

2.1. The system calls the data filling screen (VehicleForm)

2.1.T The screen shows a form to fill in details and the "Finish" button



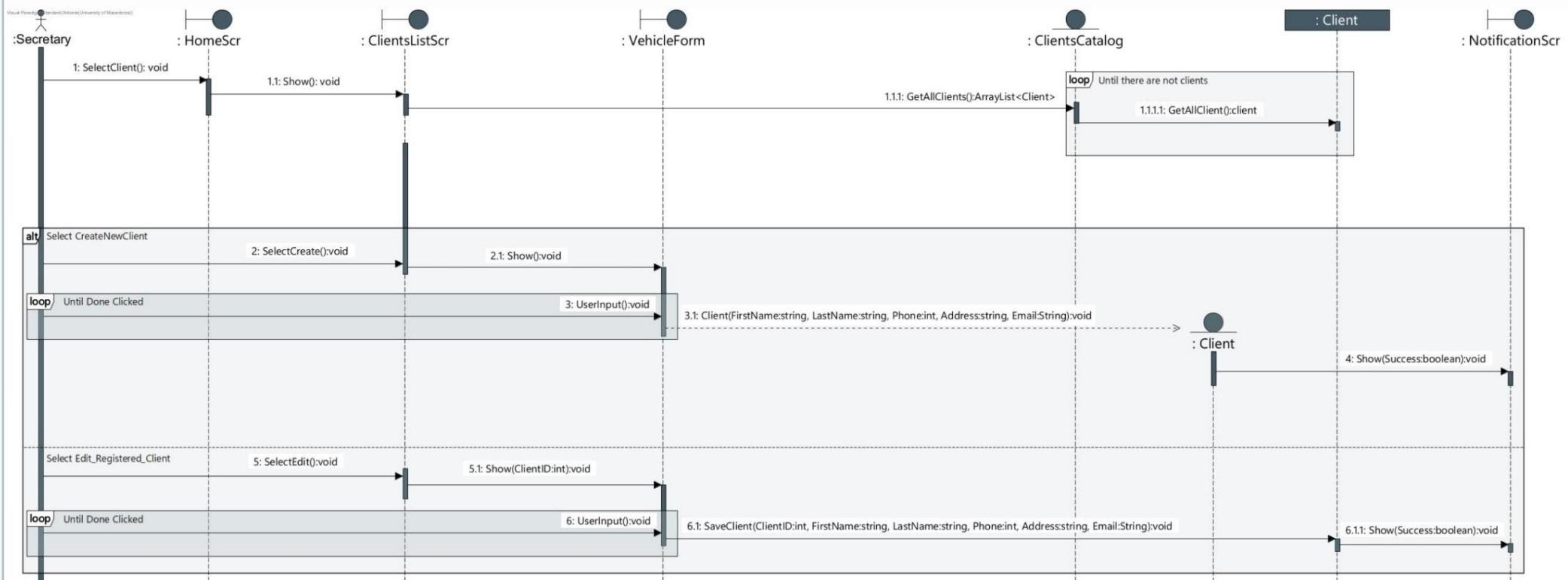
2. Basic "Customer Tab" Flow



3. The secretary from the information filling screen enters the required fields and presses the "Finish" button

3.1 The system with these fields creates a new Customer element and displays a message

3.1.1T The message "Customer Card Creation Successful" appears on the screen



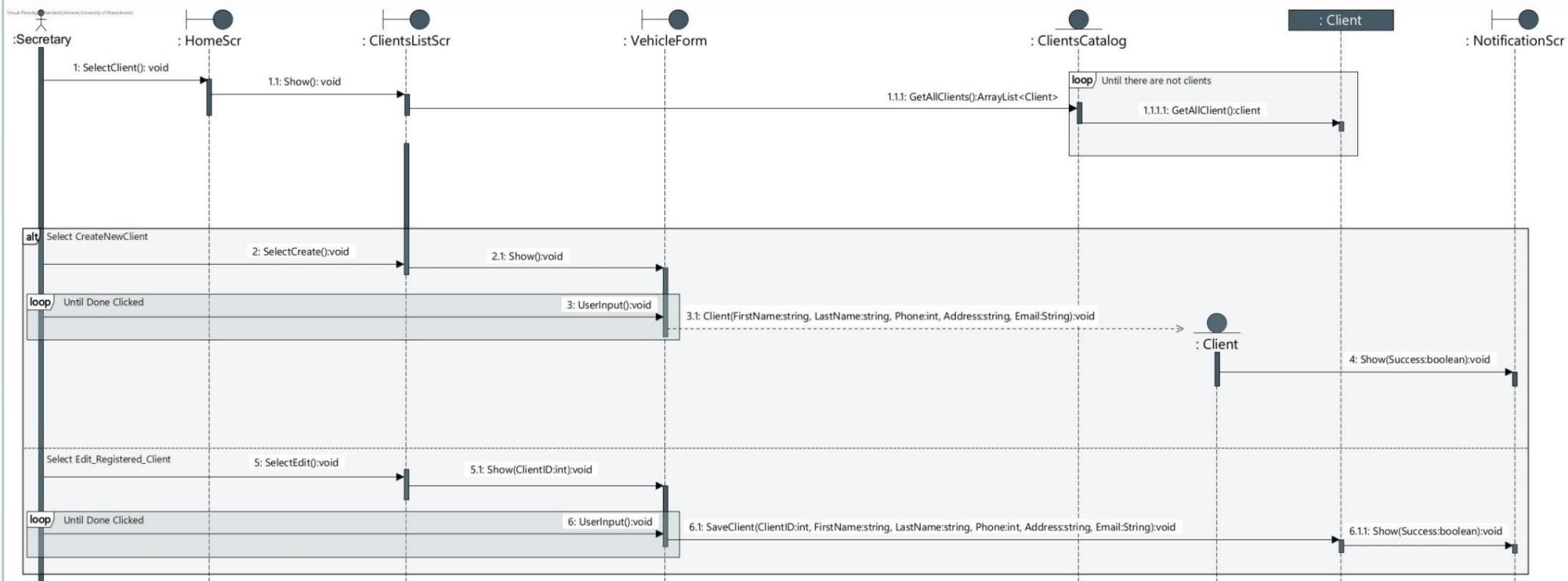
2. "Customer Tab" Basic Flow - Alternate Subflow of 1



2. The secretary from the customer screen presses the "Modify" button

2.1 The system calls the data filling screen (VehicleForm)

2.1.T The screen shows a form to fill in details and the "Finish" button



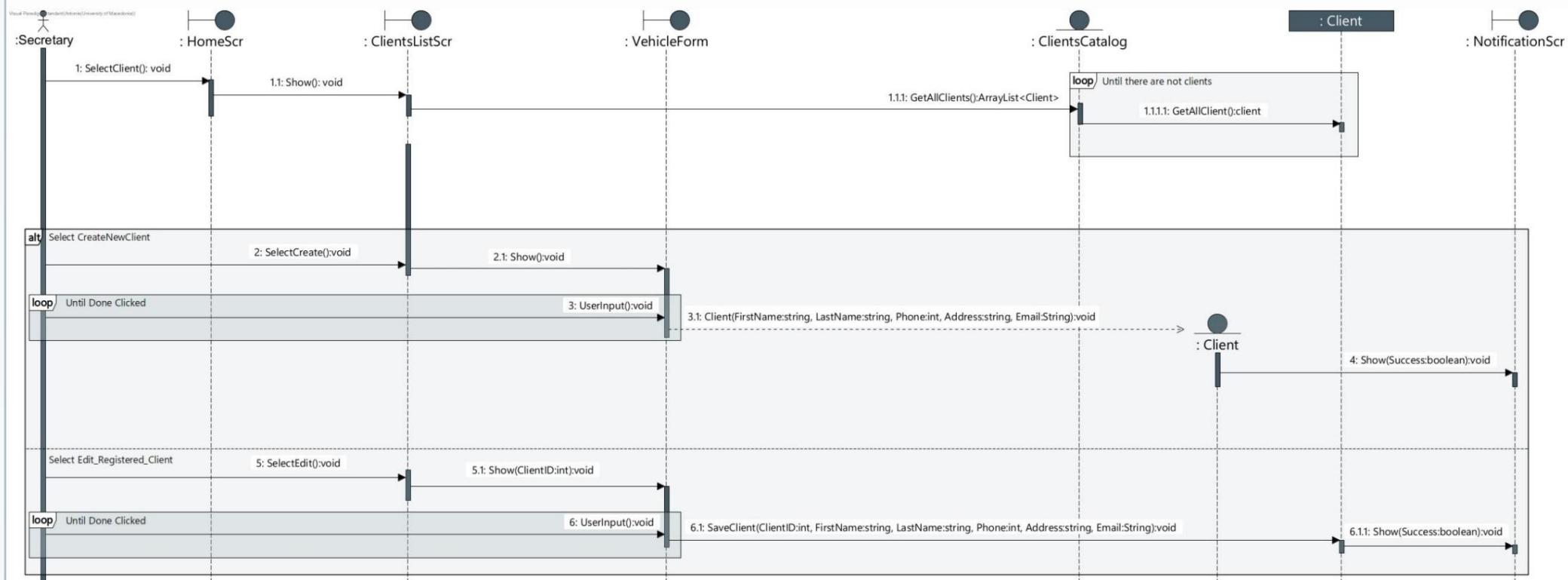
2. "Customer Tab" Basic Flow - Alternate Subflow of 1



3. The secretary from the data filling screen updates the required fields and presses the "Finish" button

3.1 The system with these fields updates the customer record and displays a message

3.1.1T The message "Customer Card Processing Successful" appears on the screen



3. Basic "Book Appointment" Flow



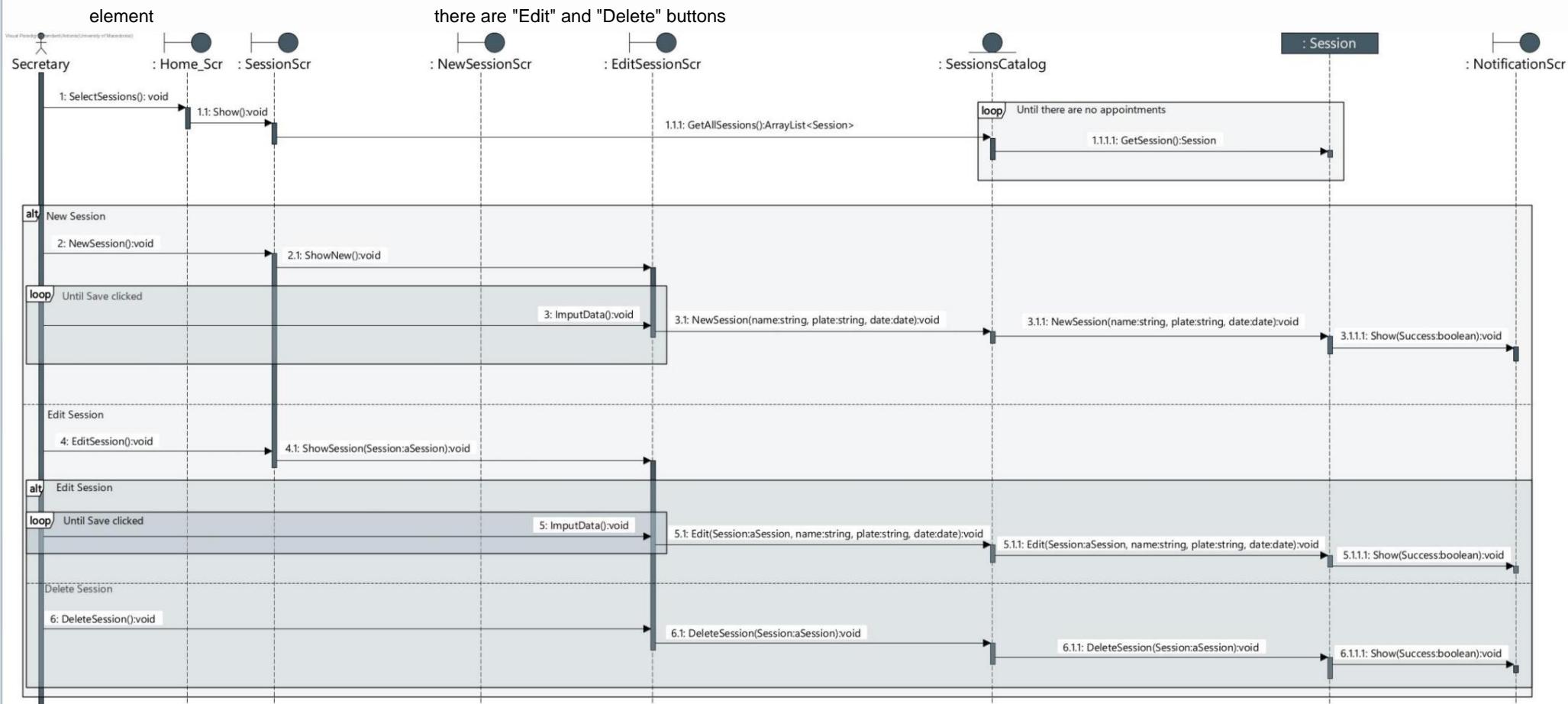
1. The secretariat from its home screen presses the "Appointment" button

1.1. The system calls the appointment screen (SessionScr)

1.1.1. The appointment screen calls the appointment directory to populate the appointment list

1.1.1.1. The appointment list returns a list of all scheduled appointments

1.1.1.1.T On the appointments screen you will see the 'New Appointment' button and a list of all appointments where next to each



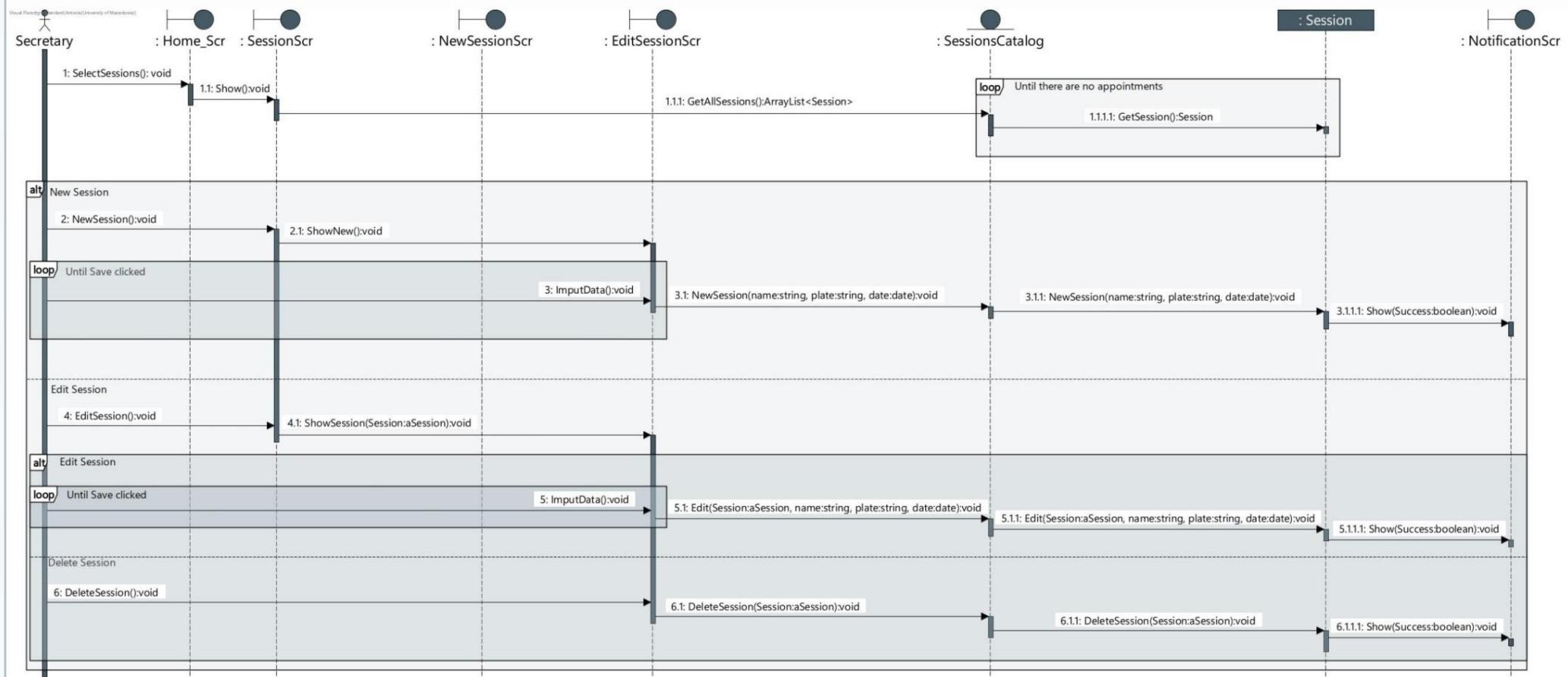
3. Basic "Book Appointment" Flow



2. The secretary from the appointment screen presses the "New Appointment" button

2.1 The system calls the appointment information filling screen (EditSessionScr)

2.1.T The screen shows a data filling form and the "Finish" button



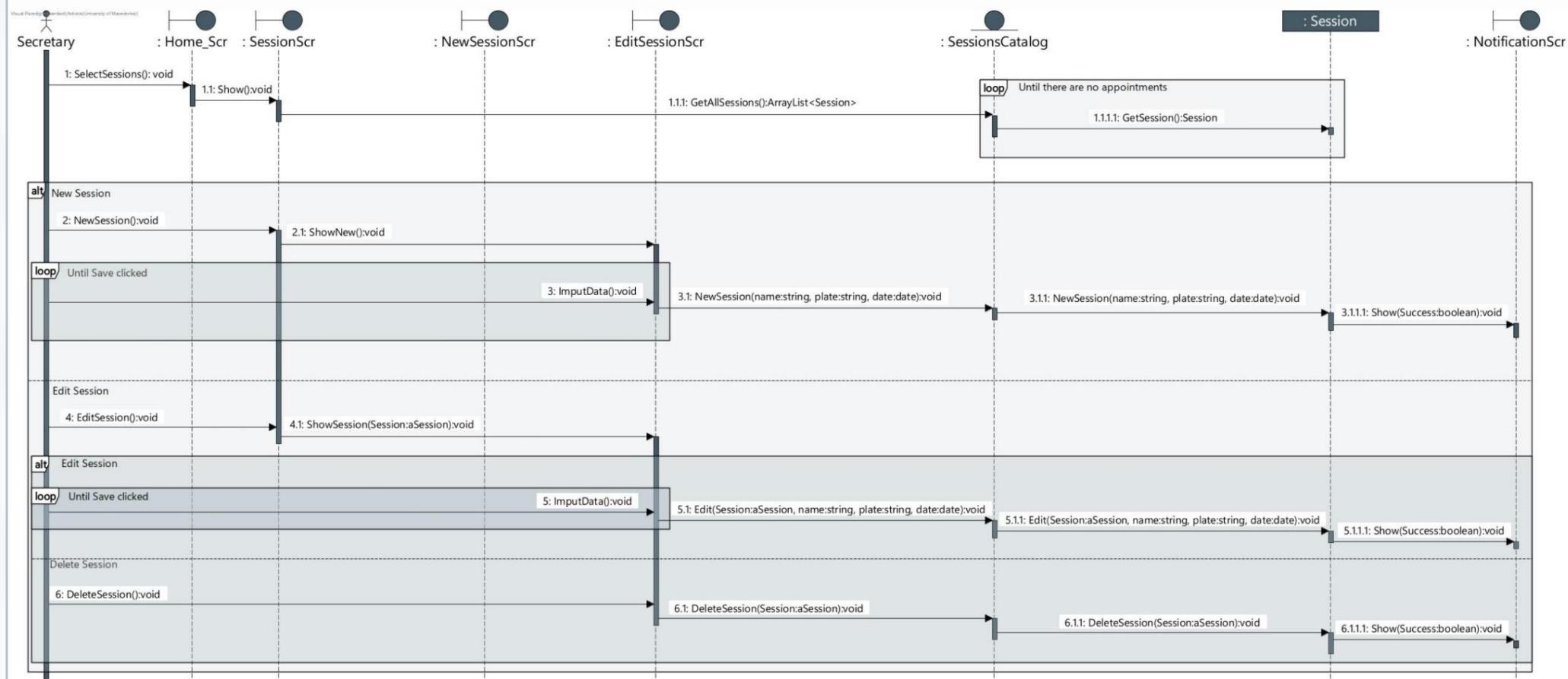
3. Basic "Book Appointment" Flow



3. The secretary from the information filling screen enters the required fields and presses the "Finish" button

3.1 The system with these fields creates a new appointment item and displays a message

3.1. T The screen shows the message "Successful Appointment Creation"



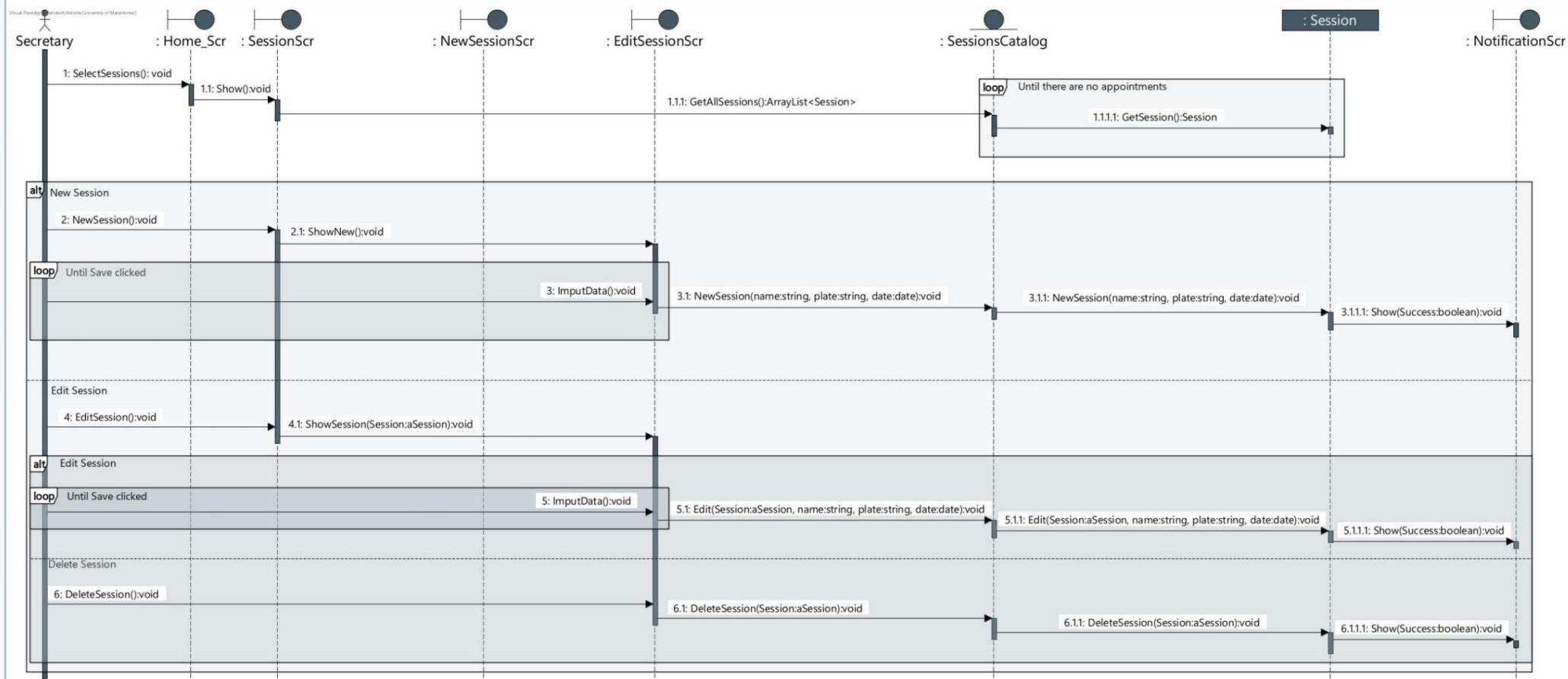
3. "Book an Appointment" Main Stream - 1 Alternate Substream of 1



4. The secretary from the appointment screen presses the "Edit" button

4.1 The system calls the appointment information filling screen (EditSessionScr)

4.1.T appears a form to fill in details and the "Finish" button



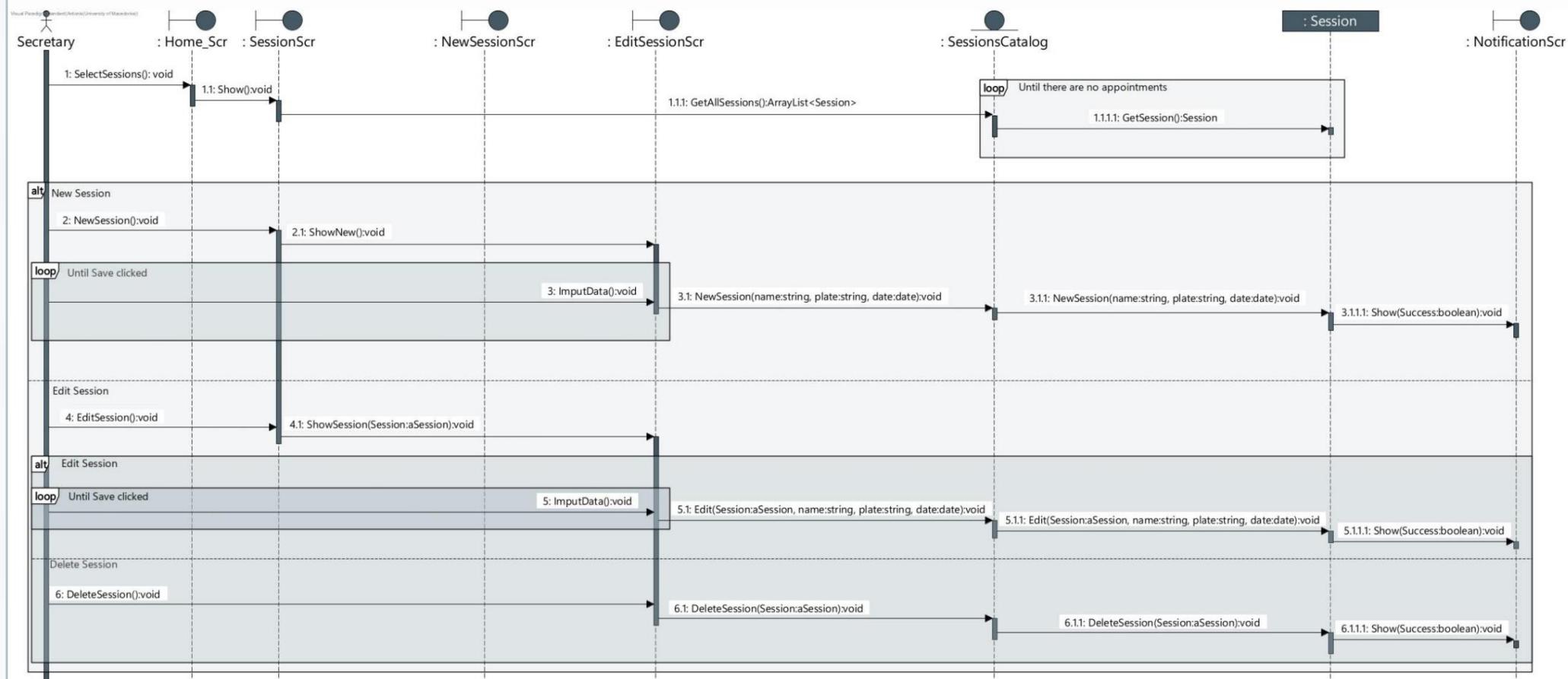
3. "Book an Appointment" Main Stream - 1 Alternate Substream of 1



5. The secretary from the data filling screen updates the required fields and presses the "Finish" button

5.1 The system with these fields updates the appointment details and displays a message

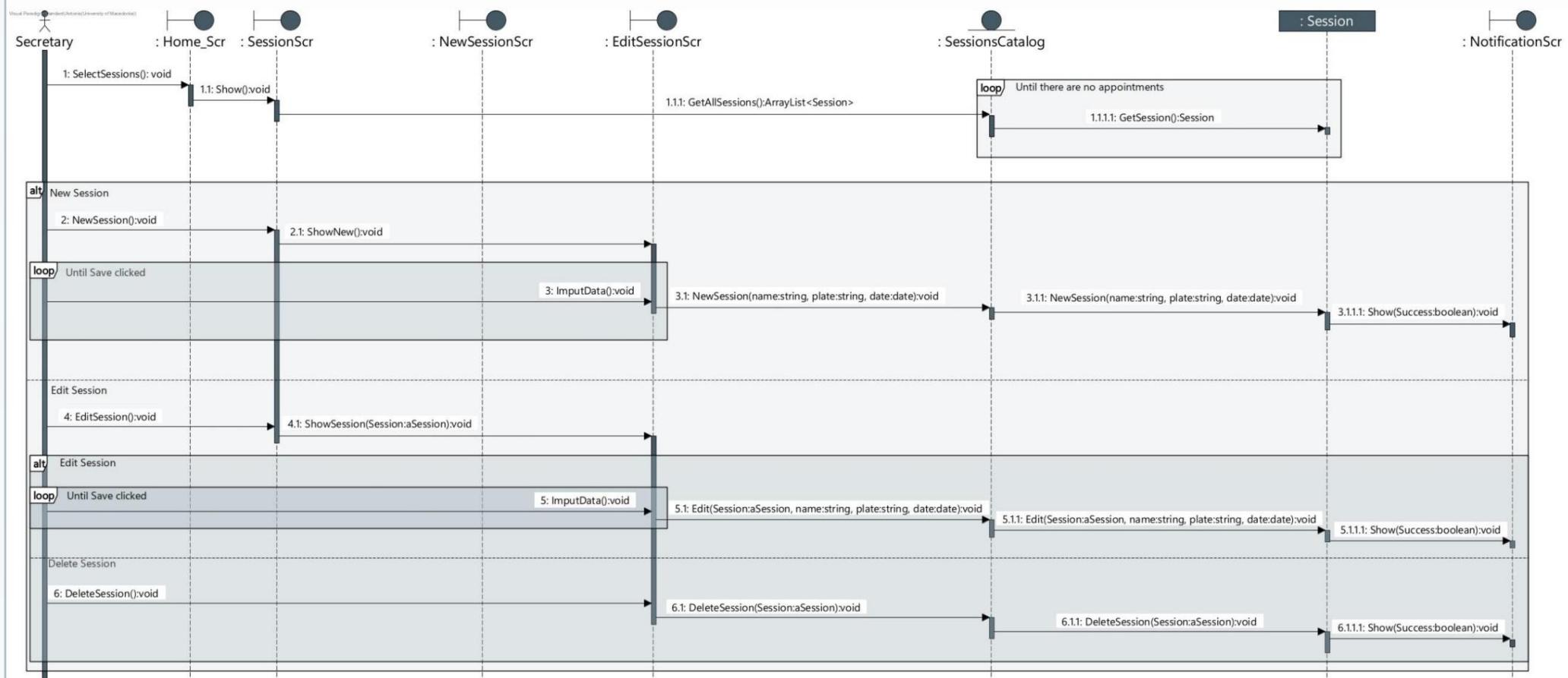
5.1.T The message 'Successfully Edit Appointment' appears on the screen



3. "Book an Appointment" Main Stream - 2nd Alternate Substream of 1



6. The secretary from the appointment screen presses the "Delete" button
- 6.1 The system deletes the entry of the selected appointment and displays a message
- 6.1.T The screen shows the message "Successfully Deleted Appointment"



4. "Vehicle Tab" Basic Flow



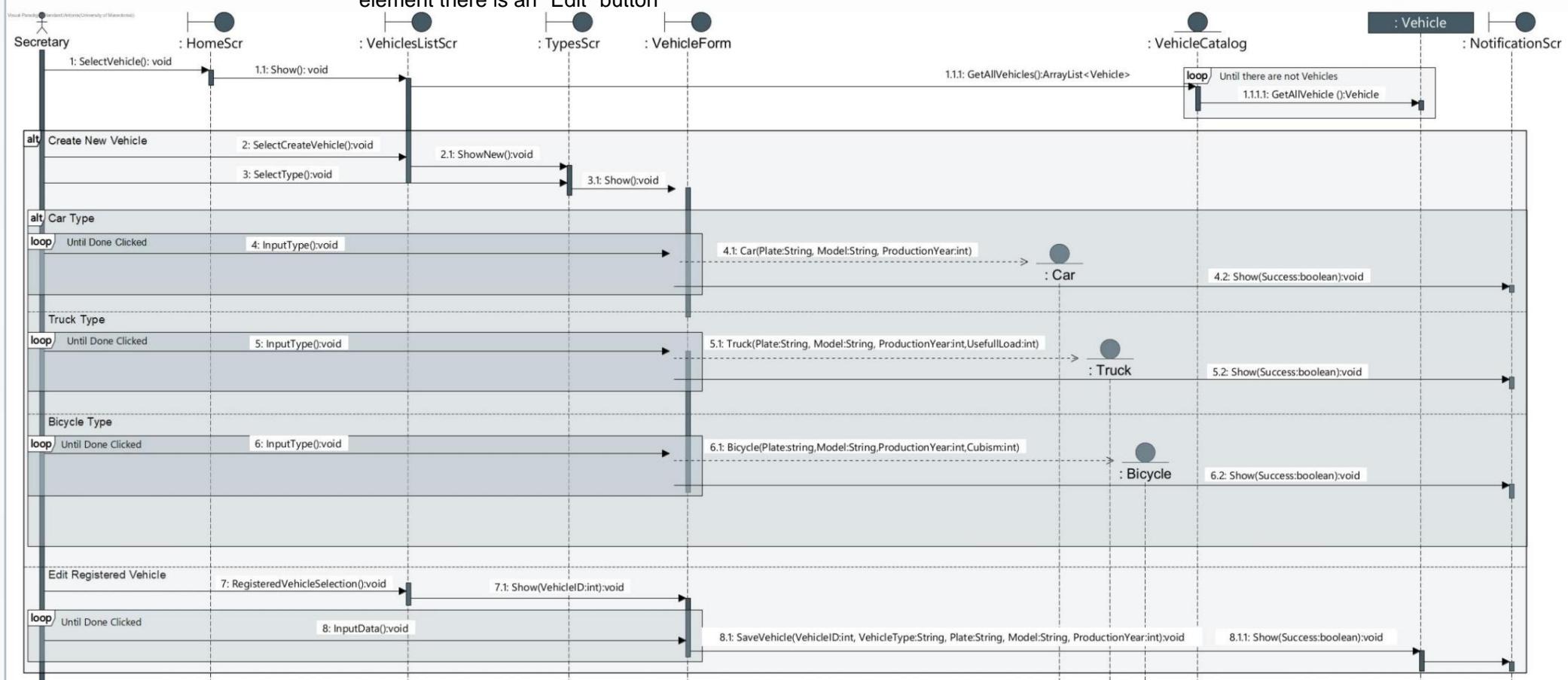
1. The secretariat from its home screen presses the "Vehicles" button

1.1. System calls vehicle screen (VehiclesListScr)

1.1.1. The vehicle screen calls the vehicle catalog to populate the vehicle list

1.1.1.1. The vehicle list returns a list of all vehicles

1.1.1.1.T The vehicle screen shows the 'New Tab' button and a list of all vehicle tabs where next to each element there is an "Edit" button



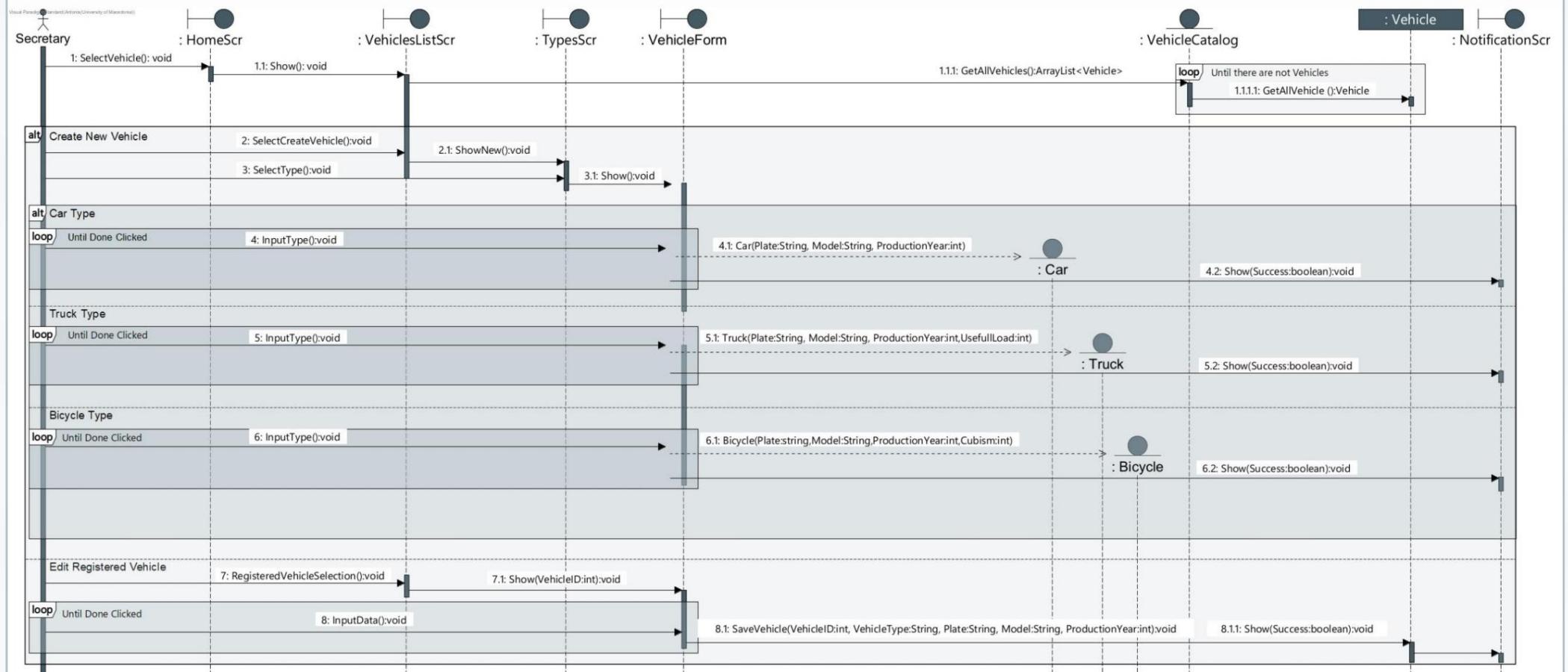
4. "Vehicle Tab" Basic Flow



2. The secretary from the vehicle screen presses the "New Tab" button

2.1 The system calls the vehicle type selection screen (TypesScr)

2.1.T The screen shows a list of "Car" "Truck" and "Bike" type



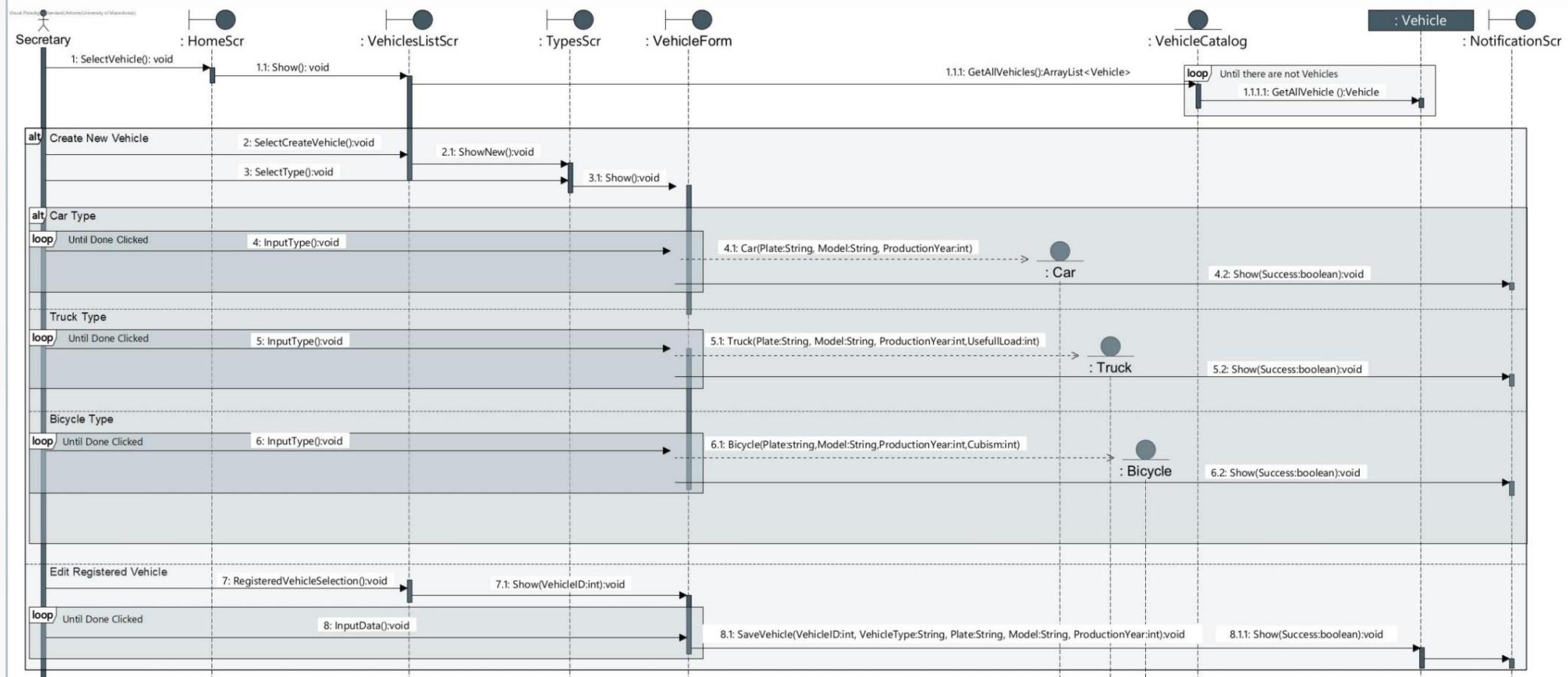
4. "Vehicle Tab" Basic Flow



3. The secretariat from the selection screen selects the type of vehicle

3.1. The system calls the data filling screen (VehicleForm) where it displays the corresponding fields depending on the type

3.1.1.T. The screen shows a form to fill in details and the "Finish" button



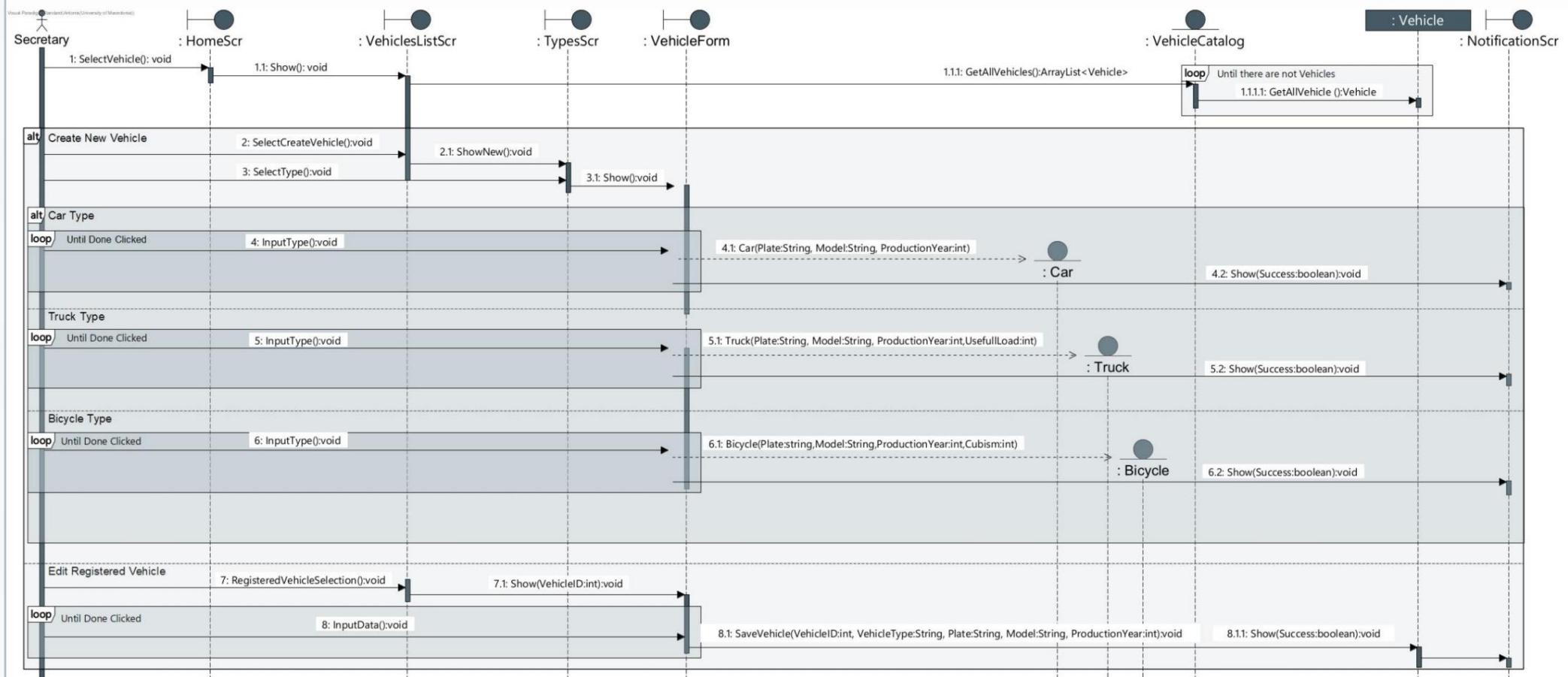
4. "Vehicle Tab" Basic Flow



4. The secretary from the information filling screen enters the required fields and presses the "Finish" button

4.1 The system with these fields creates a new tab item and displays a message

4.1.T The message "Vehicle Card Creation Successful" is shown on the screen



4. "Vehicle Tab" Basic Flow - Alternate Subflow of 1

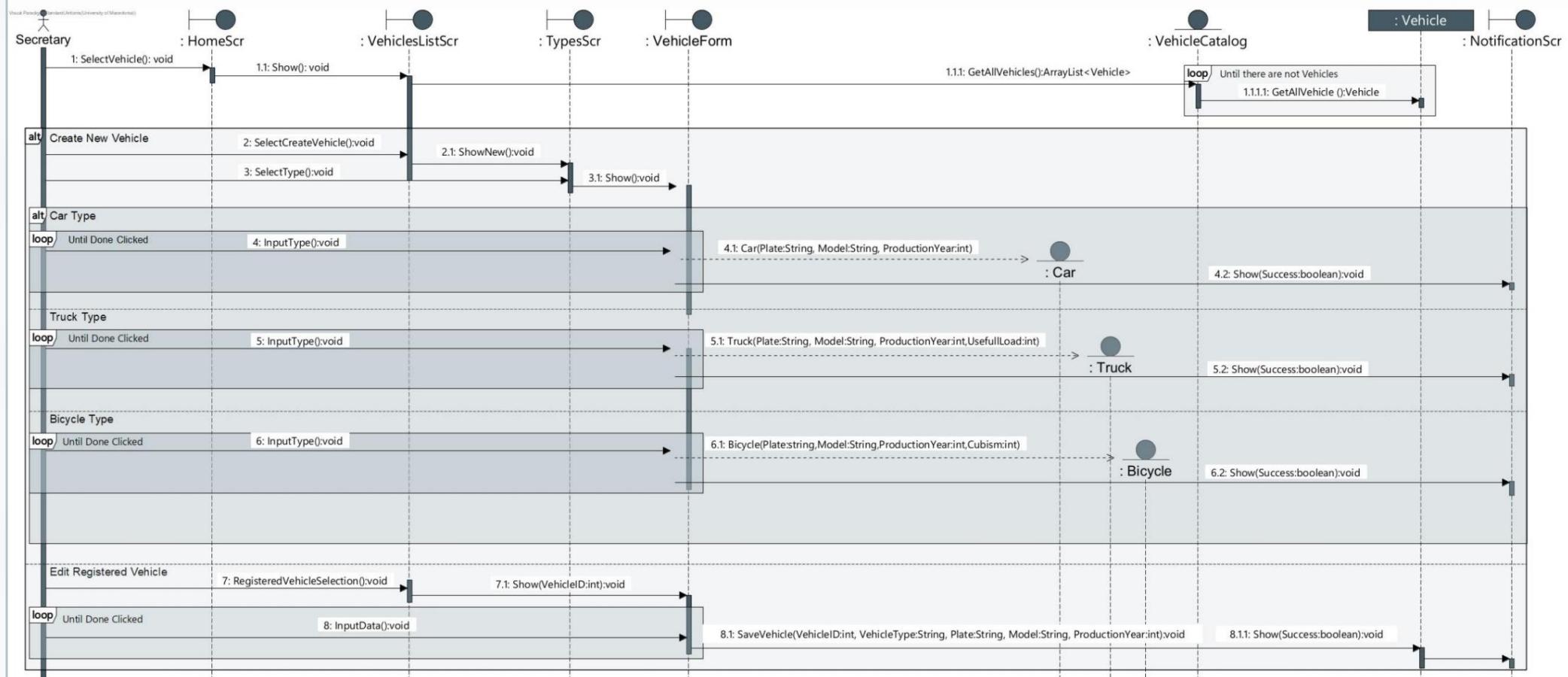


7. The secretary from the vehicle screen presses the "Edit" button

7.1 The system calls the data filling screen (VehicleForm) where it depends on the

type displays the corresponding fields

7.1.1.T The screen shows a data filling form and the "Finish" button



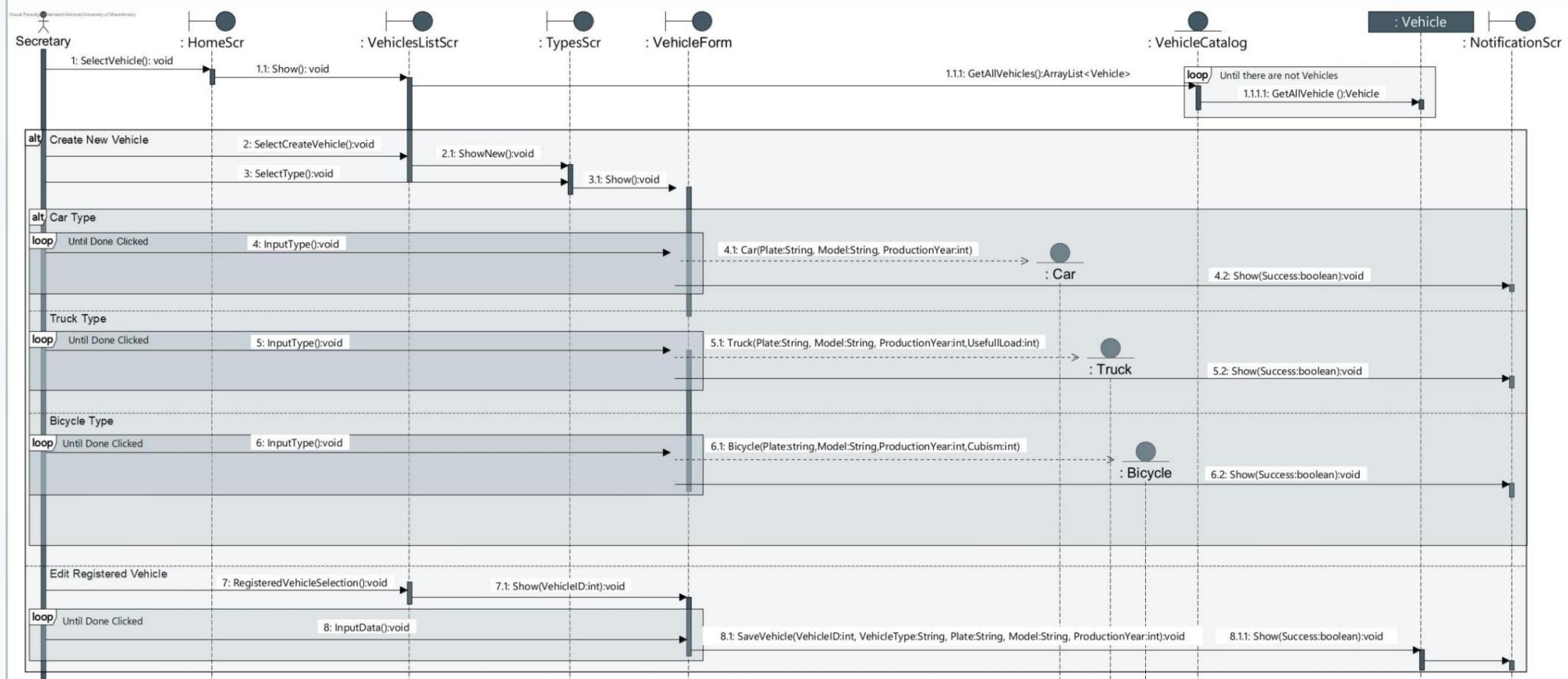
4. "Vehicle Tab" Basic Flow - Alternate Subflow of 1



8. The secretary from the information filling screen enters the required fields and presses the "Finish" button

8.1 The system with these fields updates the vehicle record and displays a message

8.1.T The message "Vehicle Card Processing Successful" is shown on the screen



5. "Create Repair Folder" Basic Flow



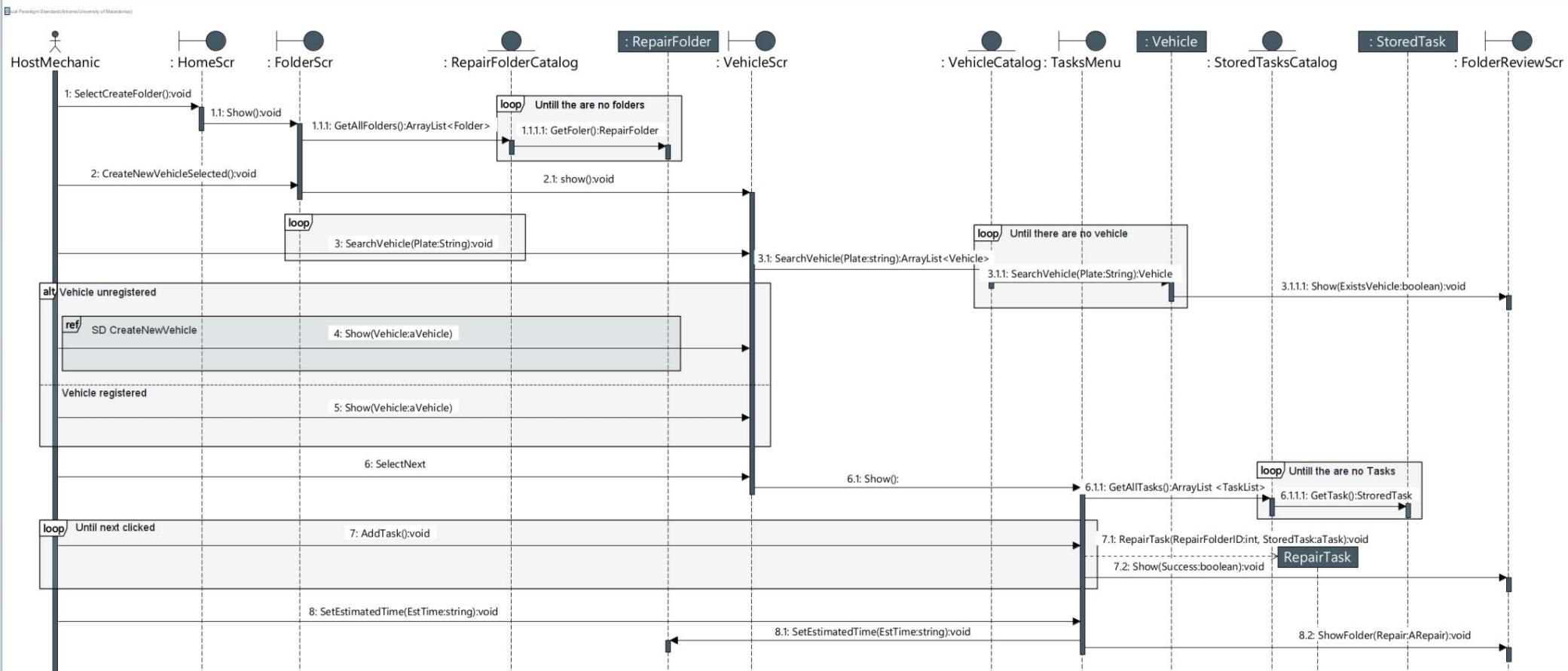
1. The host engineer from the home screen clicks the "Repair Folders" button

1.1. System calls the repair folder screen (FolderScr)

1.1.1. The called screen calls the repair catalog to populate the repair folder list

1.1.1.1. Directory returns a list of all repair folders

1.1.1.1.1. The repair folders screen shows the "Create New Folder" button and a list of all repair folders so far



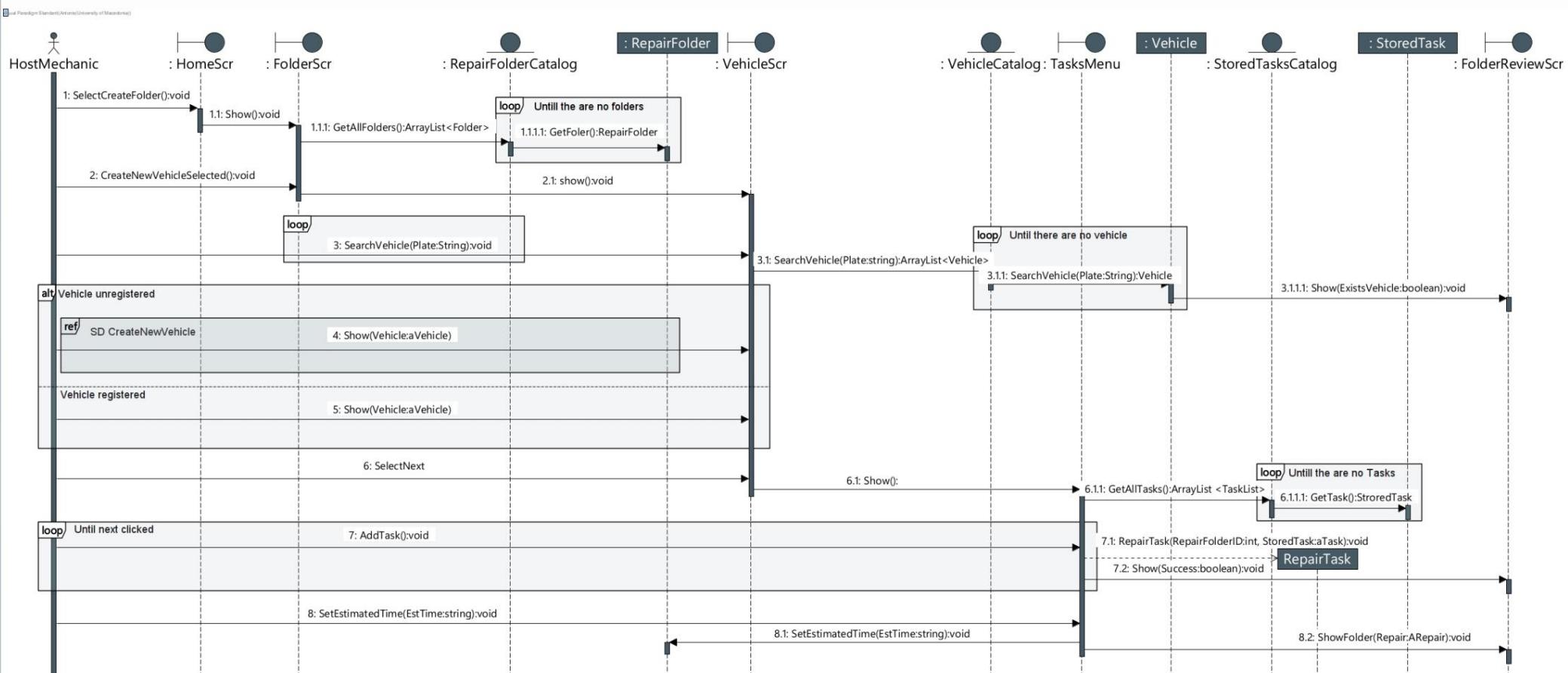
5. "Create Repair Folder" Basic Flow



2. The host engineer from the repair case clicks the "Create New Case" button

2.1 The system calls the vehicle data completion screen (VehicleScr)

2.1.T A search bar and a "Search" button appear on the screen



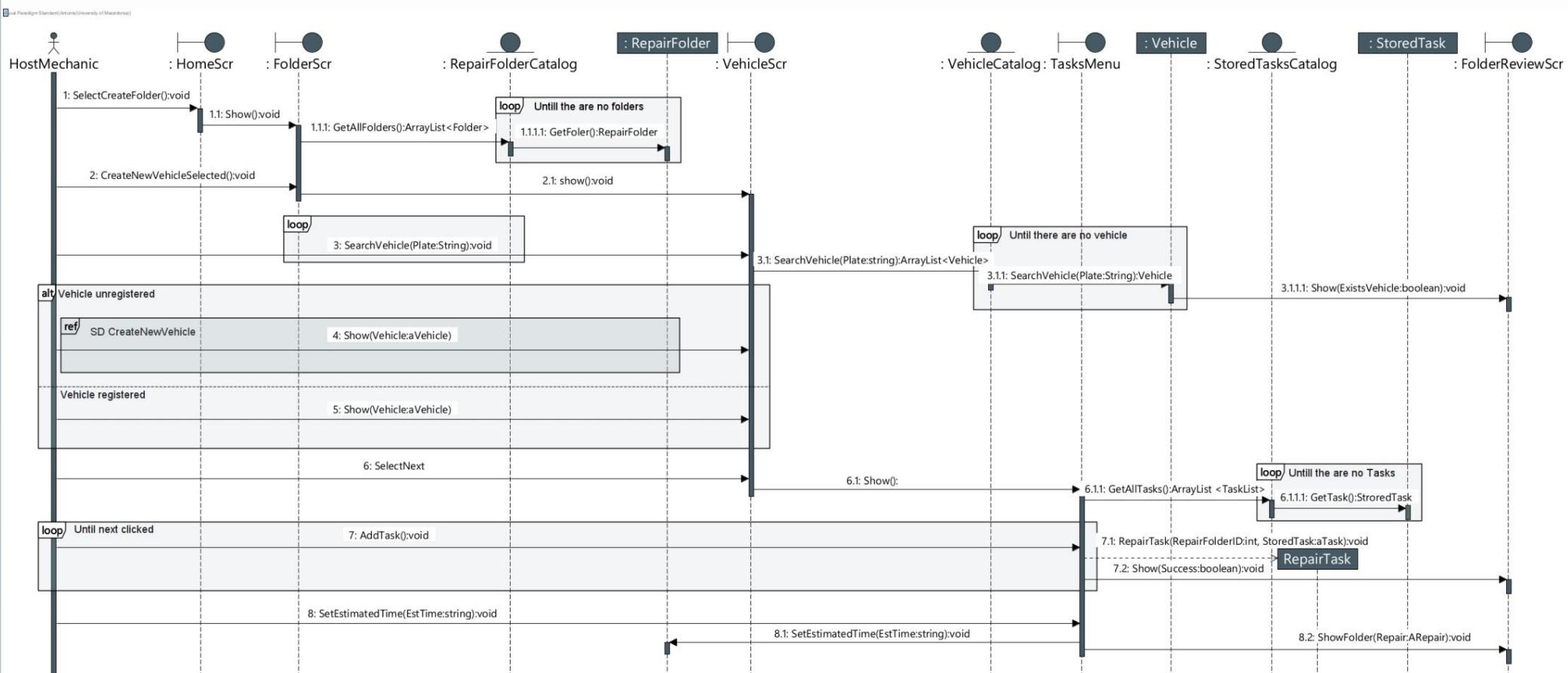
5. "Create Repair Folder" Basic Flow



3. The reception engineer from the vehicle details filling screen searches for the vehicle by the license plate number and presses "Search"

3.1. The system asks the vehicle directory to search for the vehicle

3.1.1. The directory returns a true value depending on the success of the search 3.1.1.T. The screen shows "Not Registered" or "Registered"



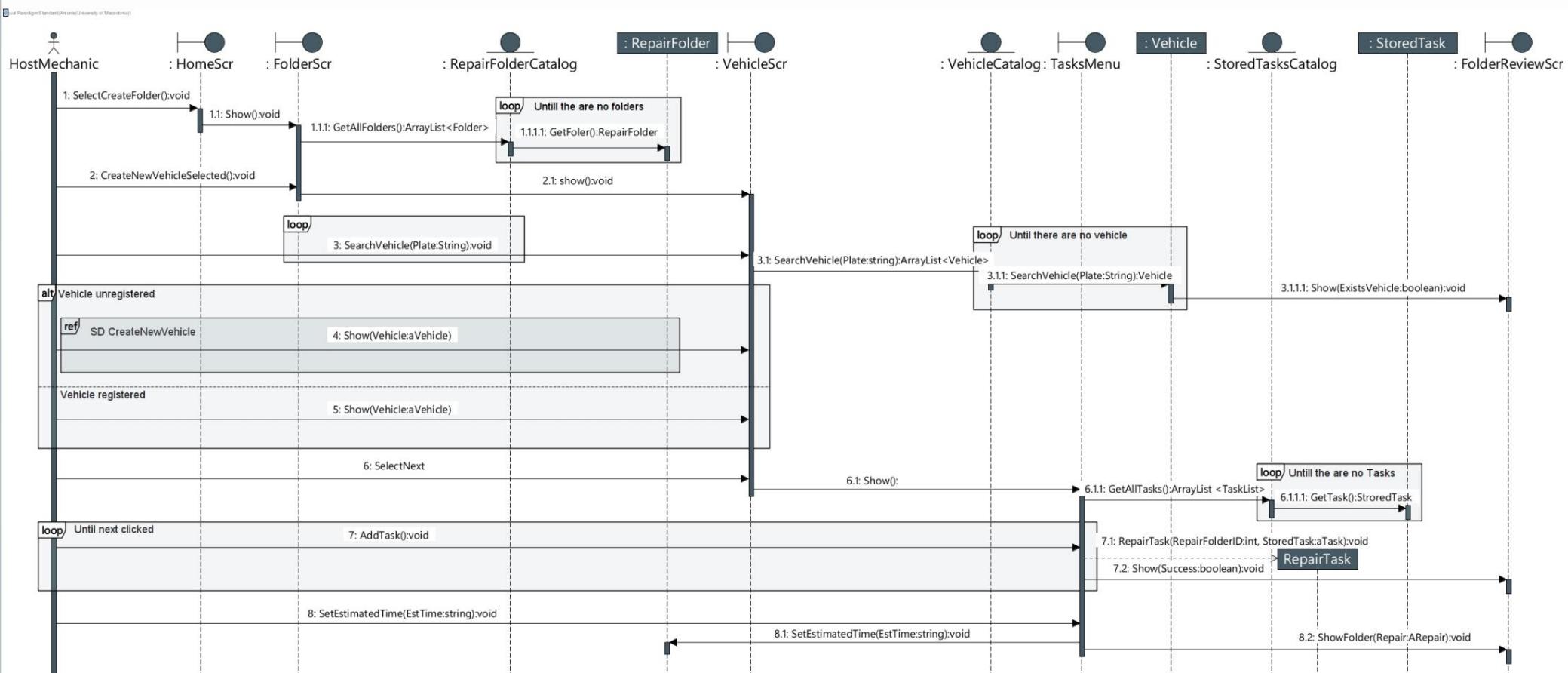
5. "Create Repair Folder" Basic Flow



4. If the vehicle is not registered then the basic Vehicle Tab flow is called, if it is then the reception engineer selects it

4.1. The system displays the details of the vehicle created or selected

4.1.T The screen shows all the vehicle details and the "Next" button



5. "Create Repair Folder" Basic Flow



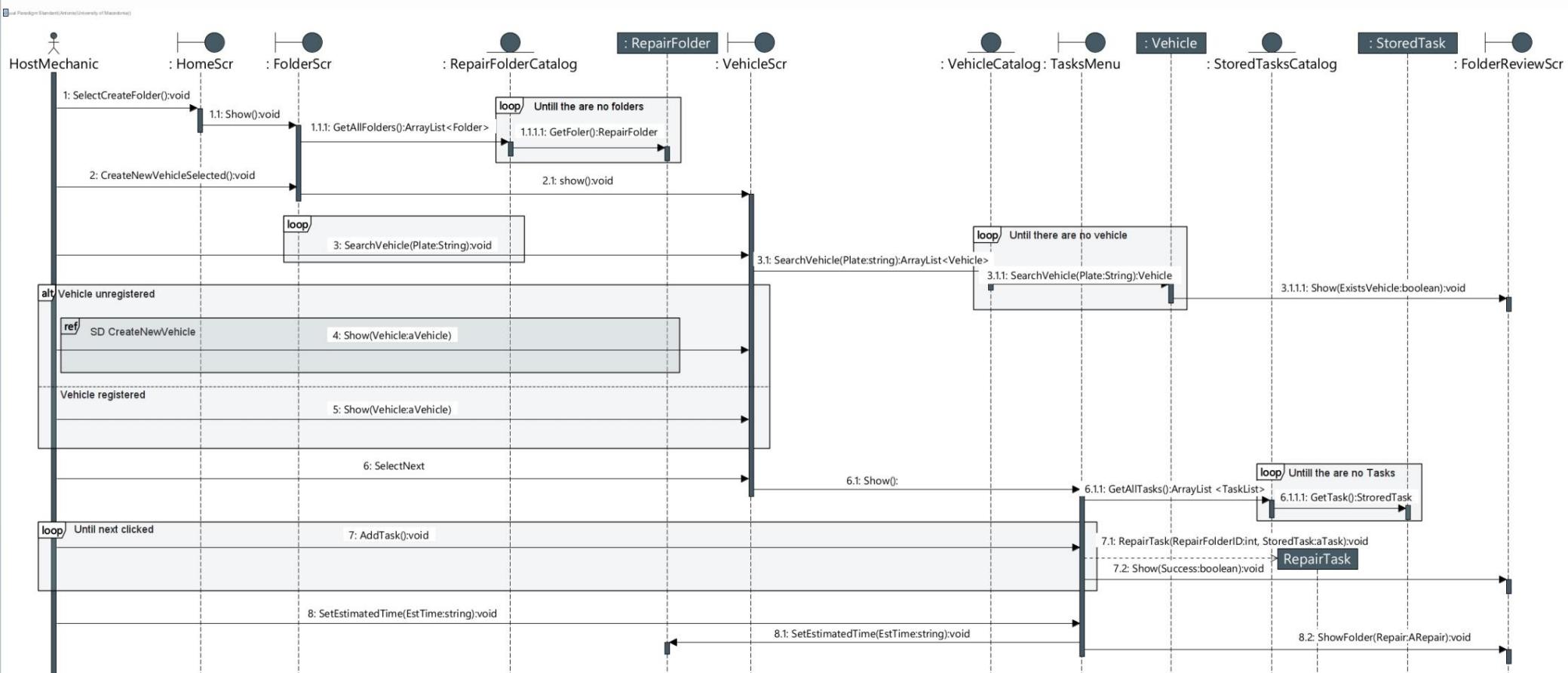
6. The host engineer selects "Next"

6.1 The system calls the task screen (TasksMenu)

6.1.1 The task screen calls the task list to populate the task list

6.1.1.1 The task list returns all ready tasks

6.1.1.1.T. The screen shows a list of all ready tasks, an "Enter" button and a "Time Estimate" field



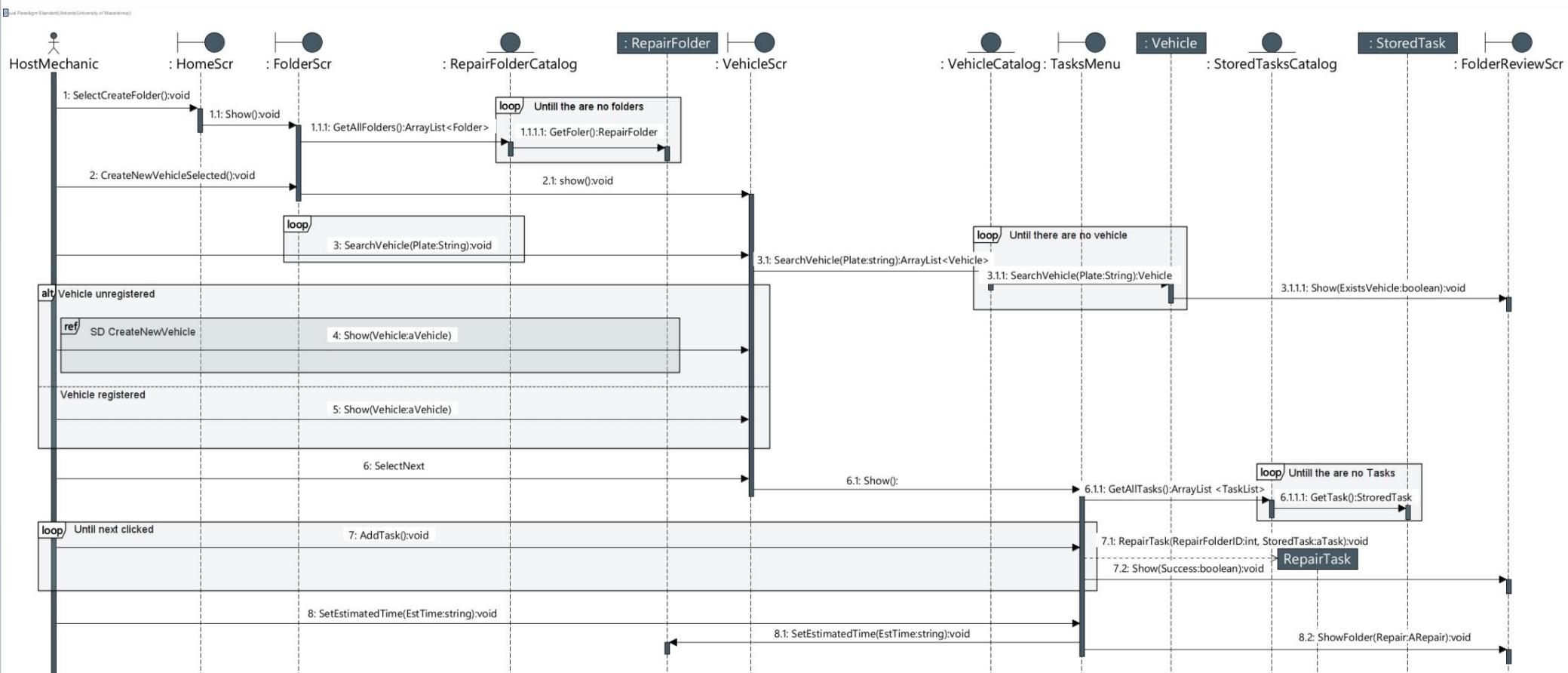
5. "Create Repair Folder" Basic Flow



7. The host engineer from the tasks screen selects the tasks that will be needed and presses "Enter"

7.1. The system with these tasks creates a new task list object and displays message

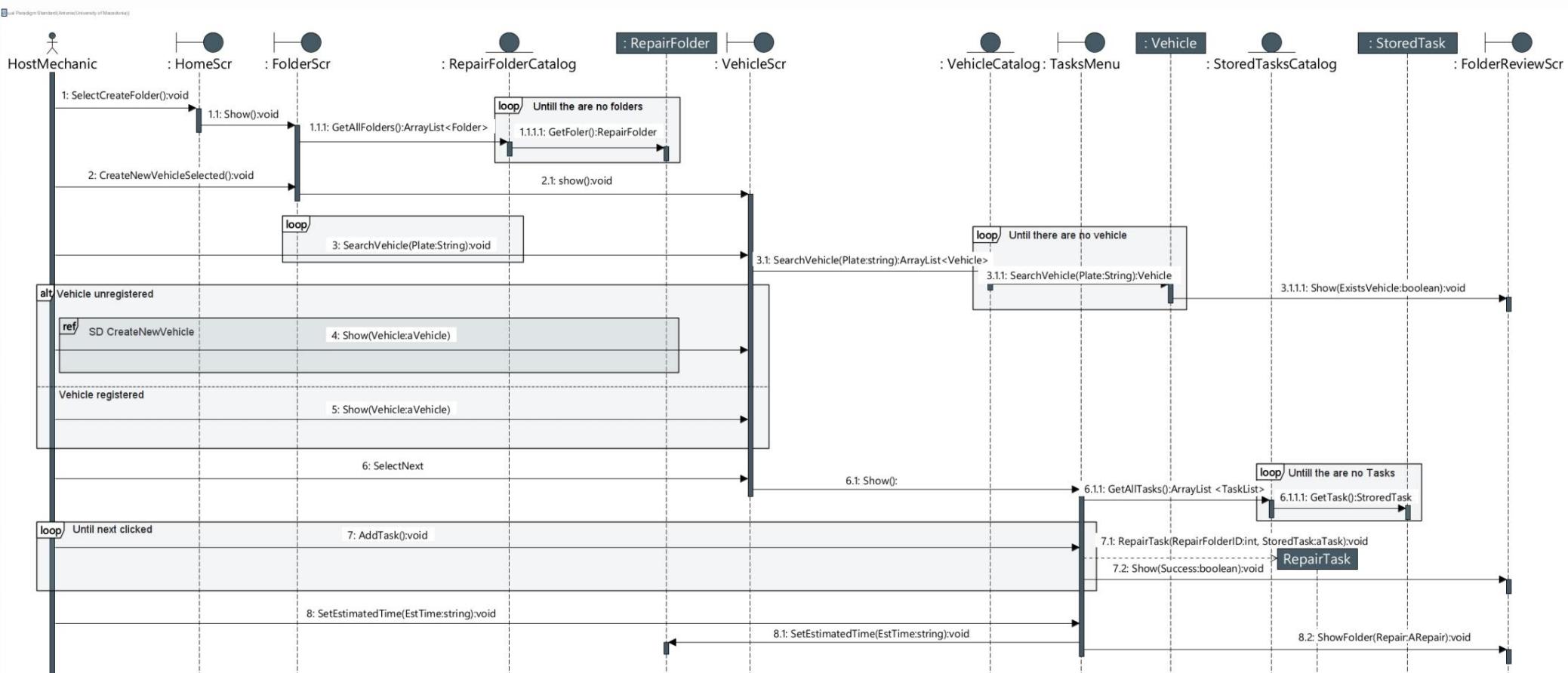
7.1.T "Task Addition Succeeded" message appears on the screen



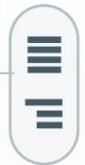
5. "Create Repair Folder" Basic Flow



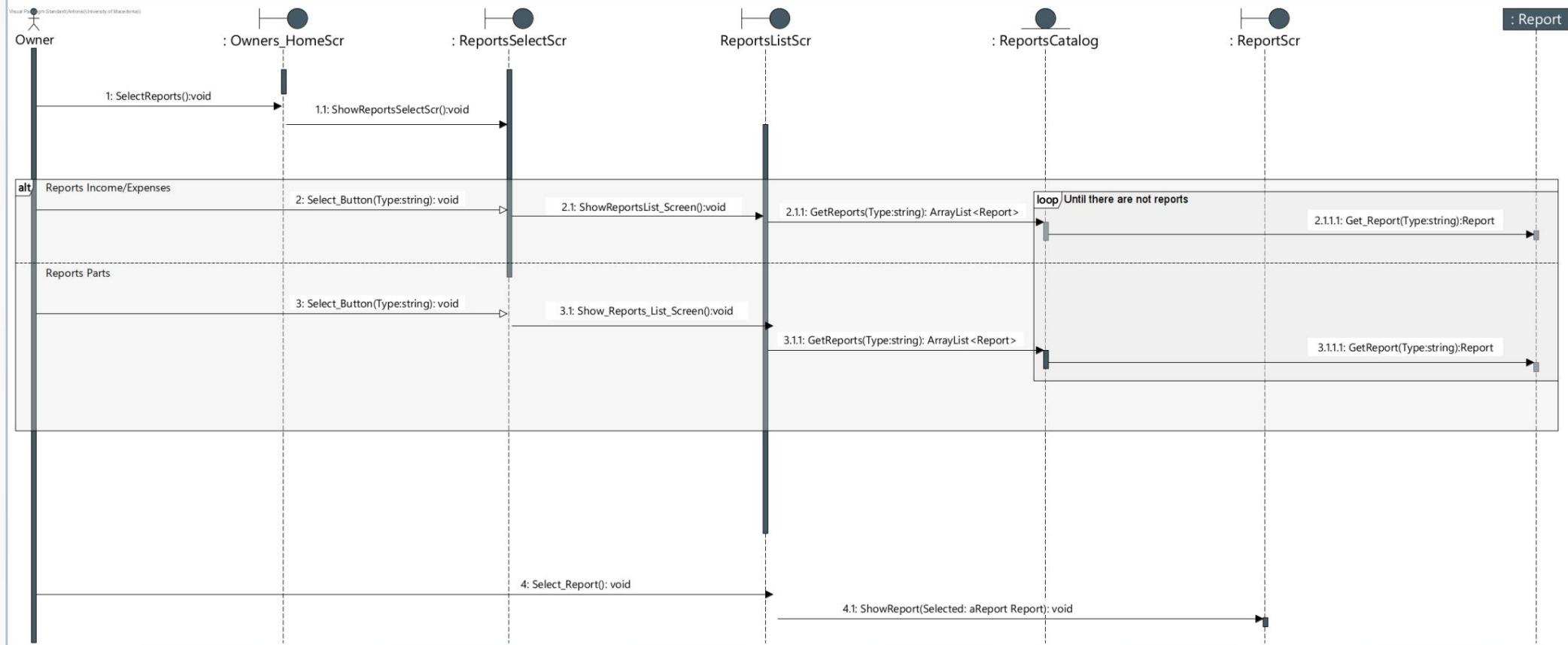
8. The host engineer from the tasks screen fills in the "Time Estimate" field
- 8.1 The system with the list of repair files updates the time field of the repair
- 8.1.1. The system calls the folder preview screen (FolderReviewScr)
- 8.1.1.T The preview of the repair folder is shown on the screen



6. Basic Flow "View Reports"



1. The owner from their home screen taps the "Reports" button
 - 1.1 The system calls the options screen (ReportsSelectScr)
 - 1.1.T The options screen shows the 'Income/Expense Reports' and 'Parts Reports' paddles



6. Basic Flow "View Reports"



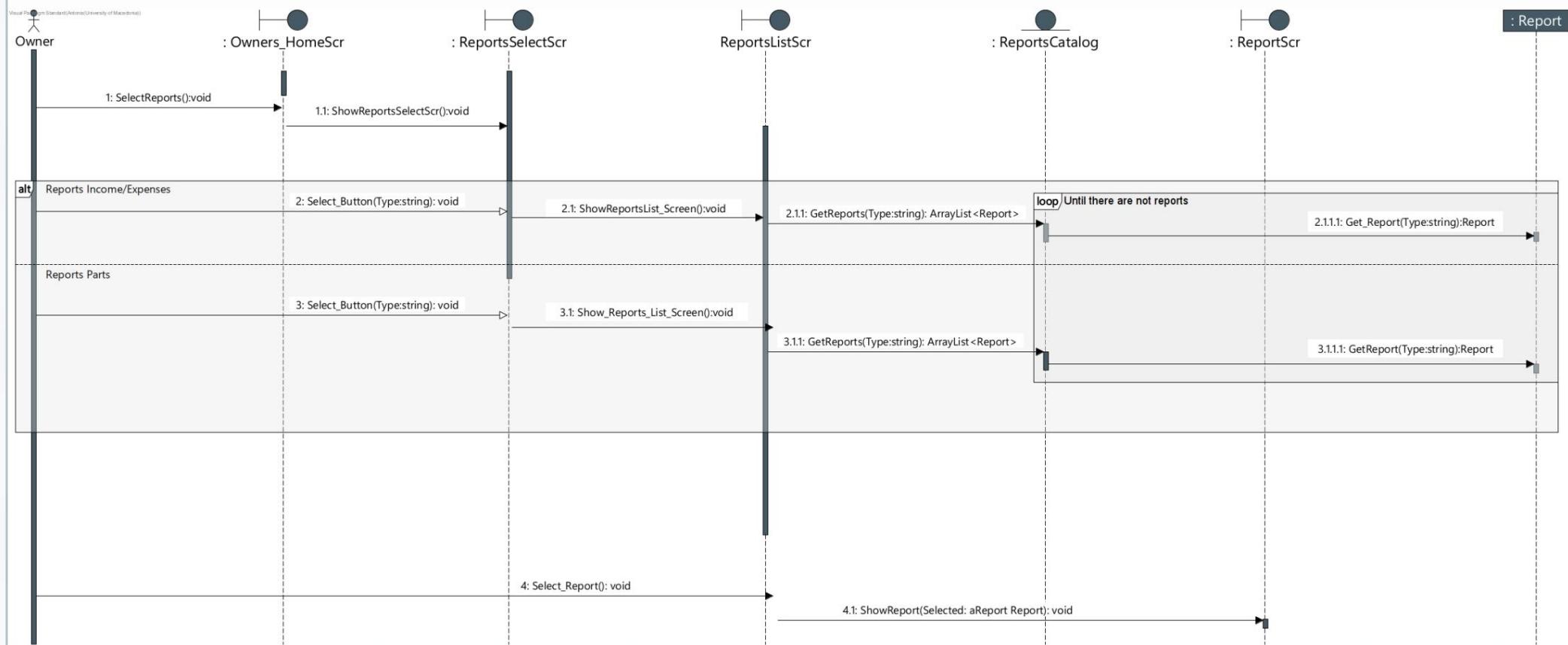
2. The owner from the options screen taps the "Income/Expense Reports" button

2.1 The system calls the reports screen (ReportsListScr)

2.1.1 The report screen calls the report catalog to populate the report list The report catalog checks the reports

2.1.1.1 and returns only the income/expense

2.1.1.1.T The reports screen shows a list of income/expense reports



6. Basic Flow "View Reports" - Alternate Flow of 2



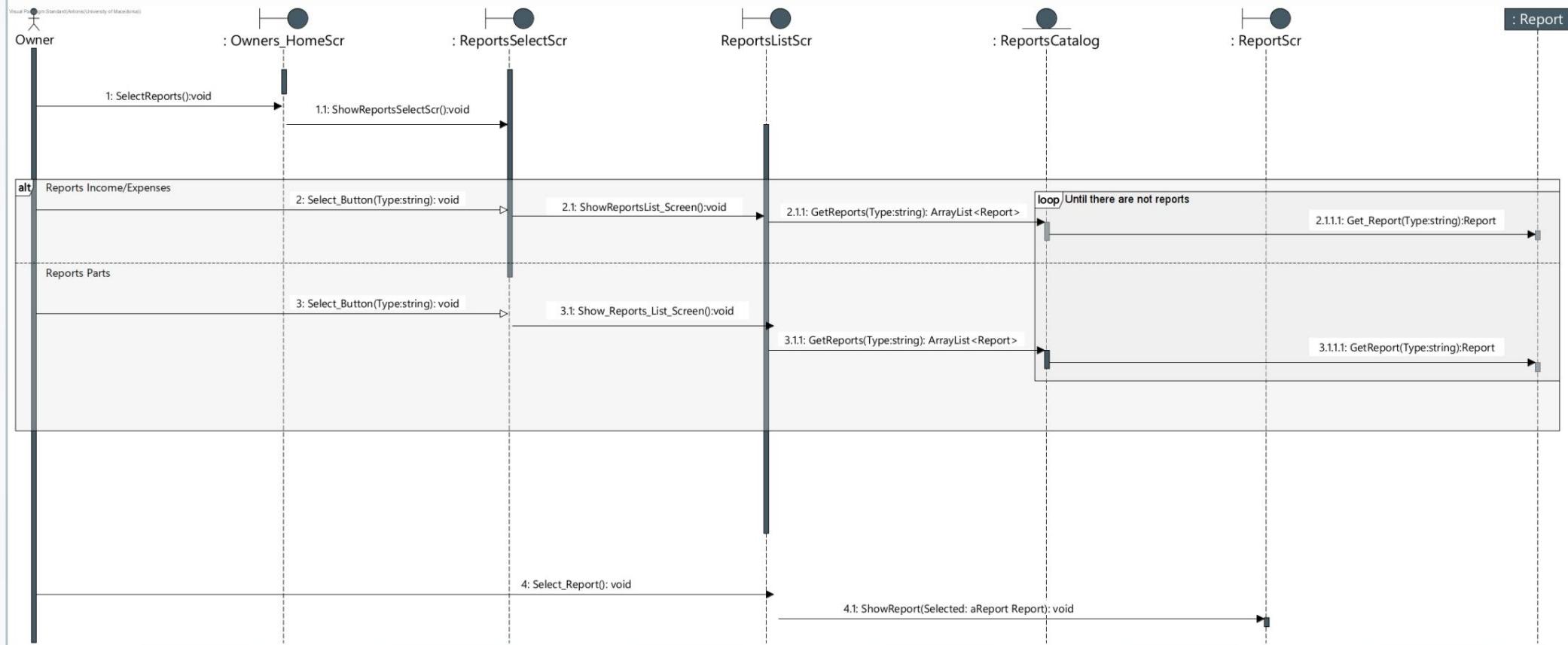
3. From the options screen, the owner presses the "Parts Reports" button

3.1 The system calls the reports screen (ReportsListScr)

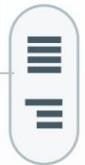
3.1.1 The reports screen calls the catalog to populate the list of reports

3.1.1.1 The report list checks the reports and returns only the parts

3.1.1.1.T The screen shows the reports screen with a list of spare parts reports



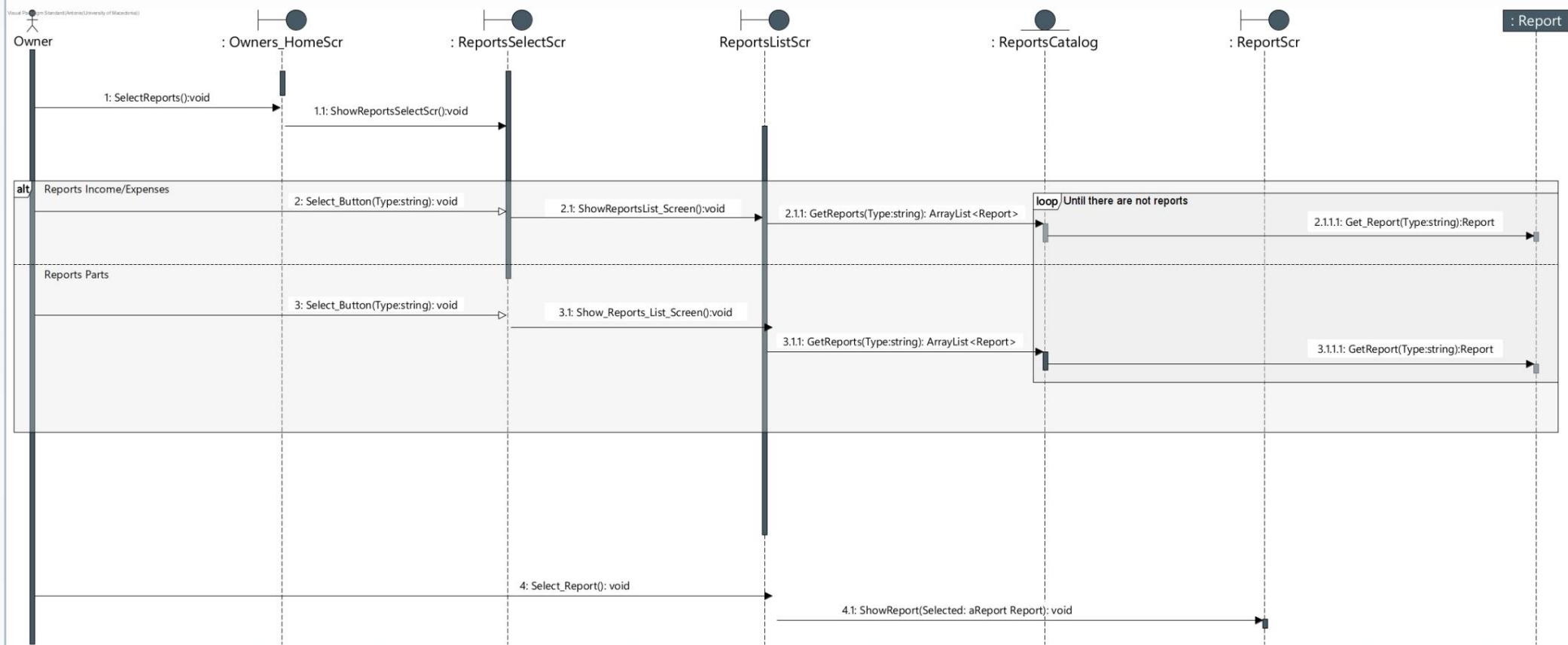
6. Basic Flow "View Reports"



4. From the reports screen, the owner selects the report he wants

4.1 The system calls the detailed report screen (ReportScr)

4.1.T The report details are shown on the screen



7. "Modify Parts List" Basic Flow



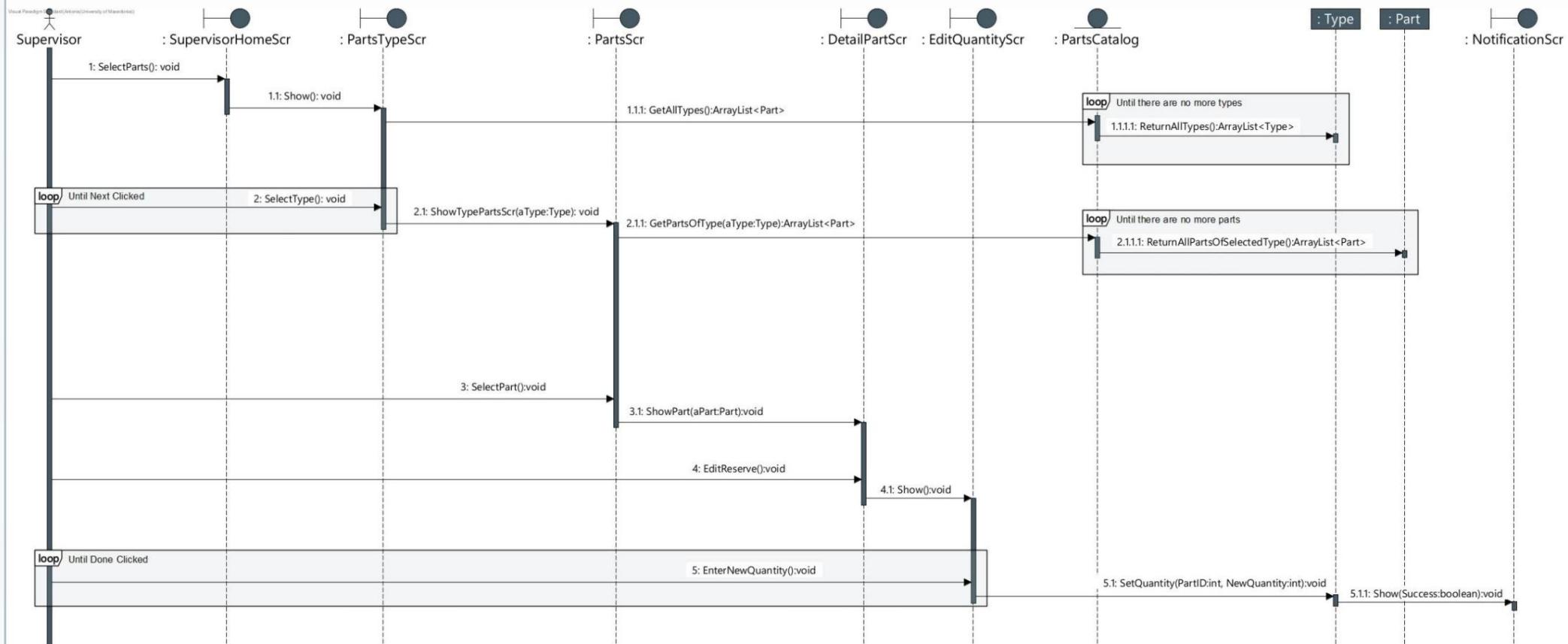
1. The supervising engineer from the home screen presses the "Parts" button

- 1.1. The system calls the parts type screen (PartsTypeScr

- 1.1.1. The called screen calls the part type list to populate the part type list

- 1.1.1.1. Catalog returns a list of all part types

1.1.1.1.T The part types screen shows all types and the 'Next' button



7. "Modify Parts List" Basic Flow



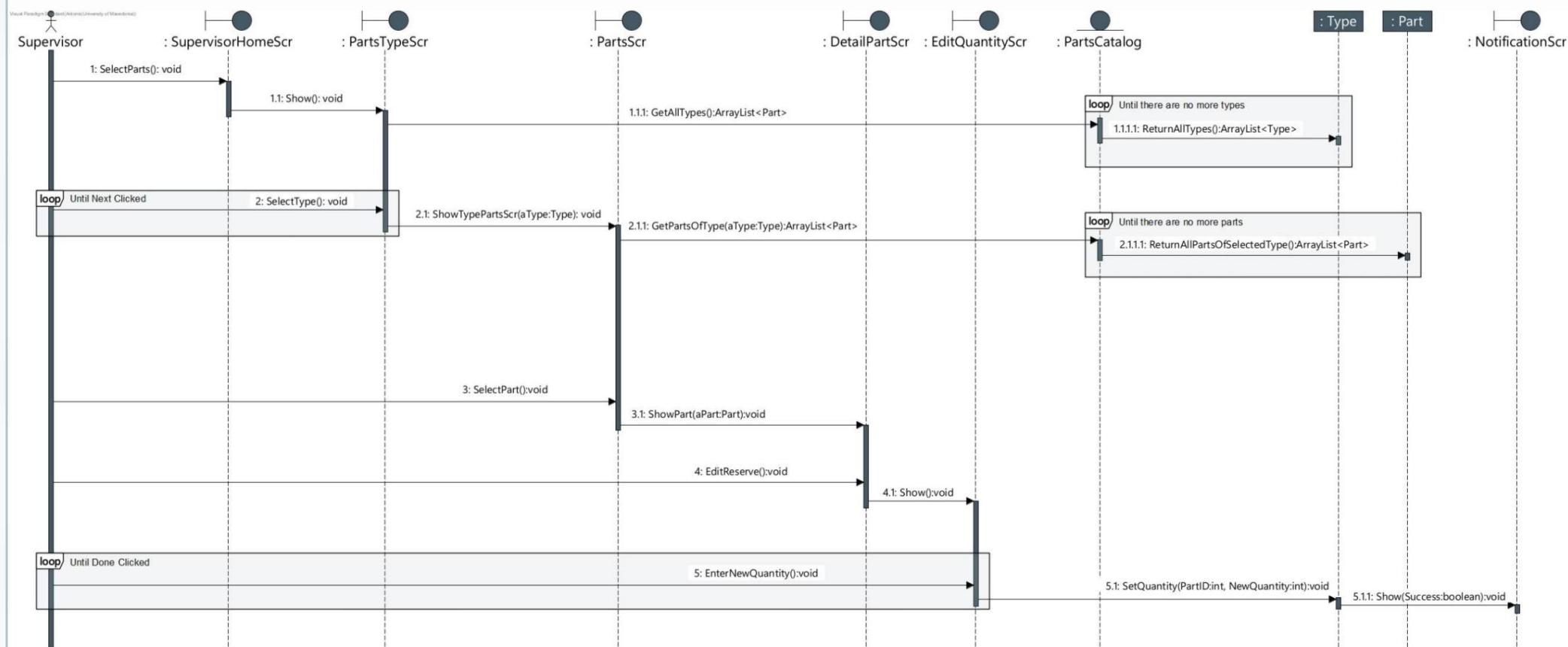
2. The supervising engineer from the part types screen selects the part types and then "Next"

2.1 The system calls the parts screen (PartsScr) according to the selected type

2.1.1. The parts screen calls up the parts catalog to populate the parts list

2.1.1.1. The vehicle list returns the parts belonging to the specified type

2.1.1.1.T The screen shows a list of requested spare parts



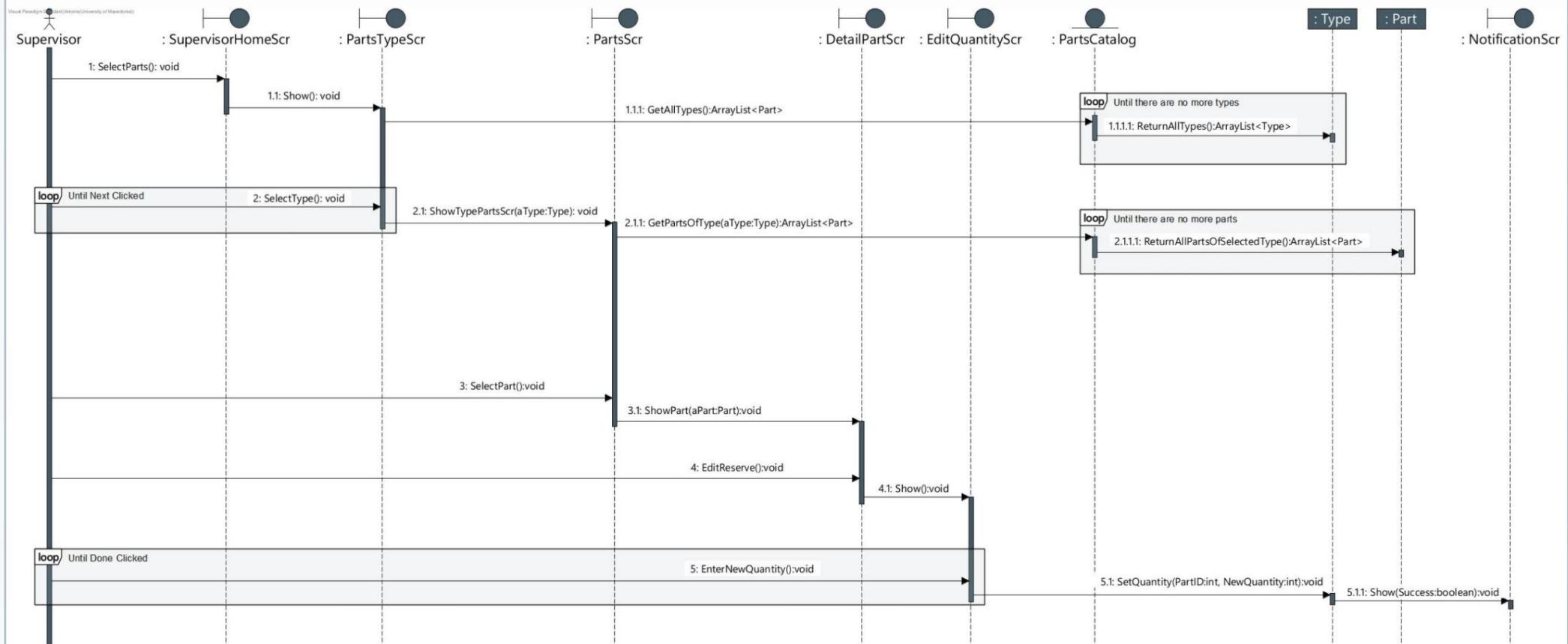
7. "Modify Parts List" Basic Flow



3. The supervising engineer from the parts screen a part

3.1 The system calls the parts detail screen (DetailPartScr)

3.1.T. The detail screen shows all the part details and the 'Modify Inventory' button



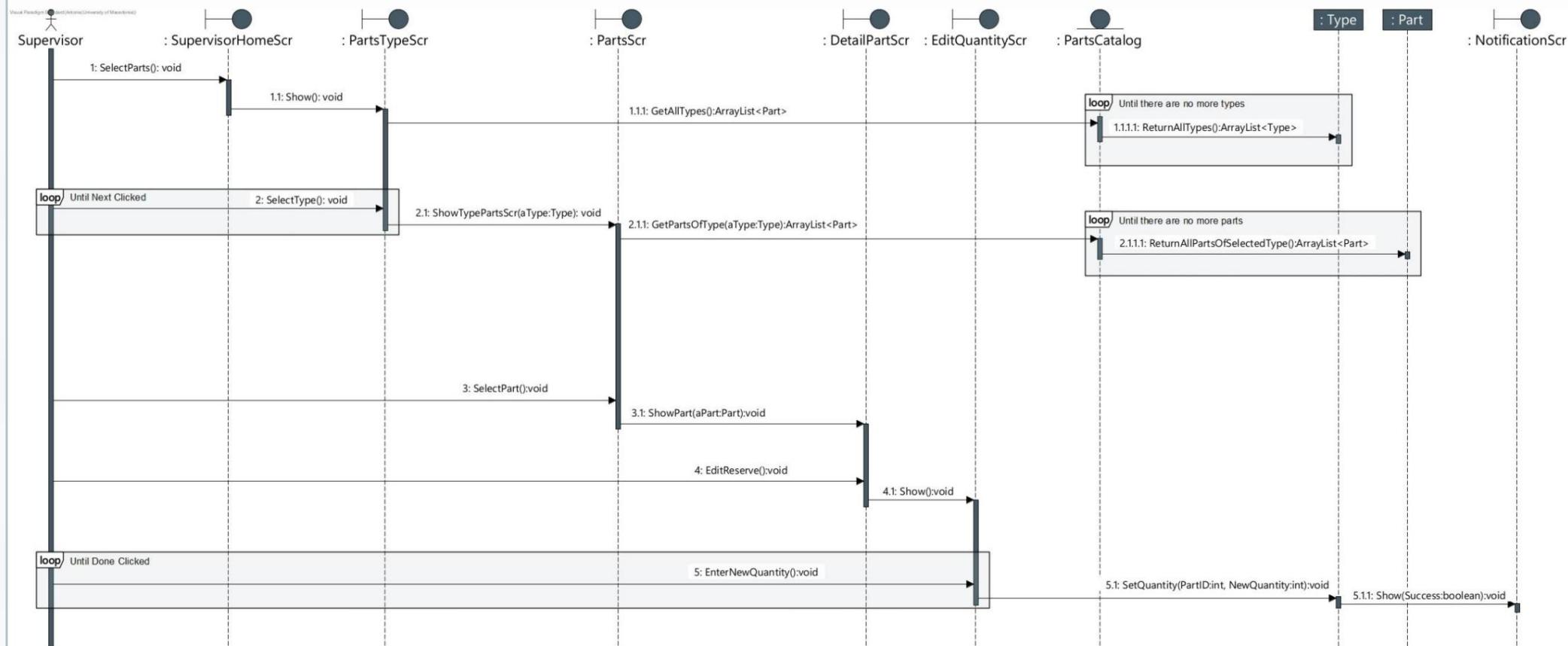
7. "Modify Parts List" Basic Flow



4. The supervising engineer from the detail screen taps "Modify Part"

4.1 The system calls the edit quantity screen (EditQuantityScr)

4.1.T. The screen shows a field to fill in and the "Save" button

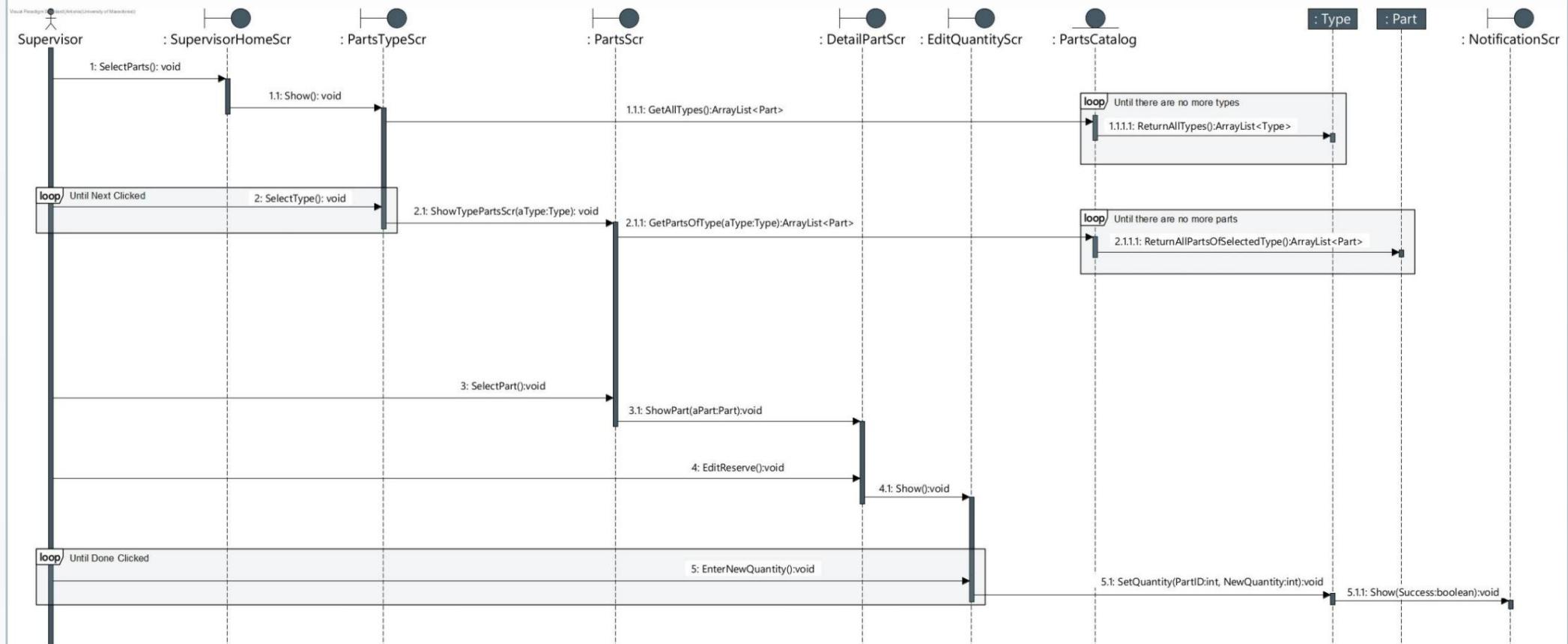


7. "Modify Parts List" Basic Flow



5.1. The system with the spare parts list updates the inventory and displays a message

5.1.T A successful update message appears on the screen



8. "Modify Task List" Basic Flow



1. The host engineer from the home screen presses the "Task List" button

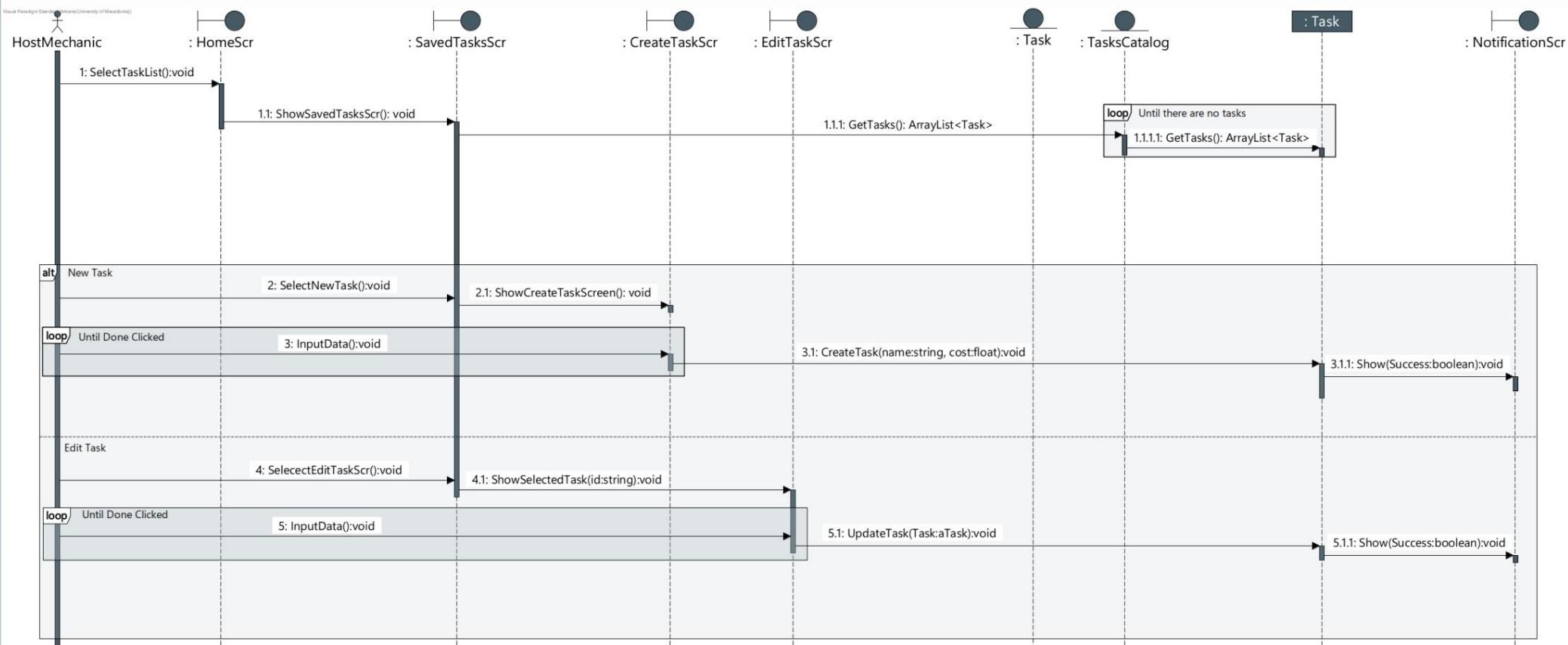
1.1. The system calls the task screen (SavedTasksScr)

1.1.1. The task screen calls the task list to populate the task list

1.1.1.1. Directory returns a list of all tasks

1.1.1.1.T The tasks screen shows an Add New Task button, a list of all tasks, and next to each one the "Modify" button

element



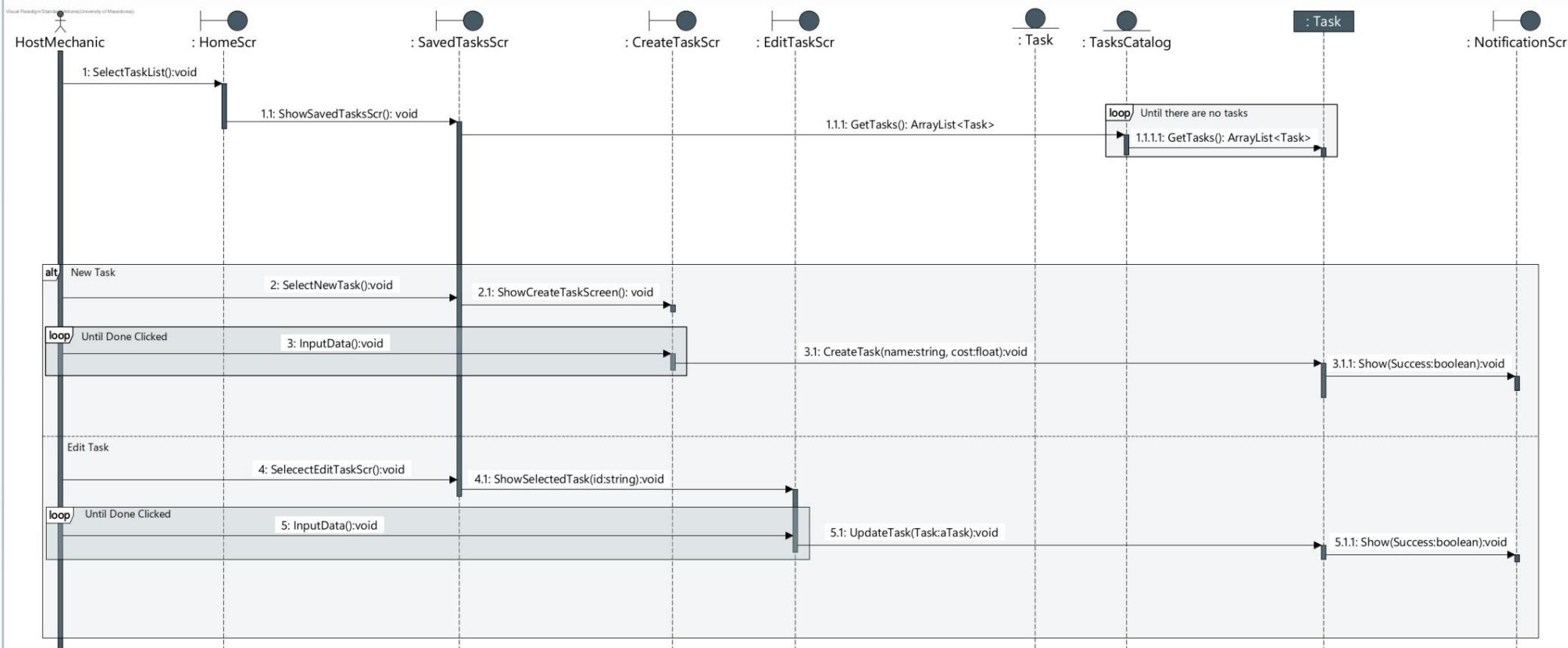
8. "Modify Task List" Basic Flow



2. The host engineer from a task screen clicks the "Add New Task" button

2.1 The system calls the task creation screen (CreateTaskScr)

2.1.T The screen shows empty fields to fill in and the "Save" button



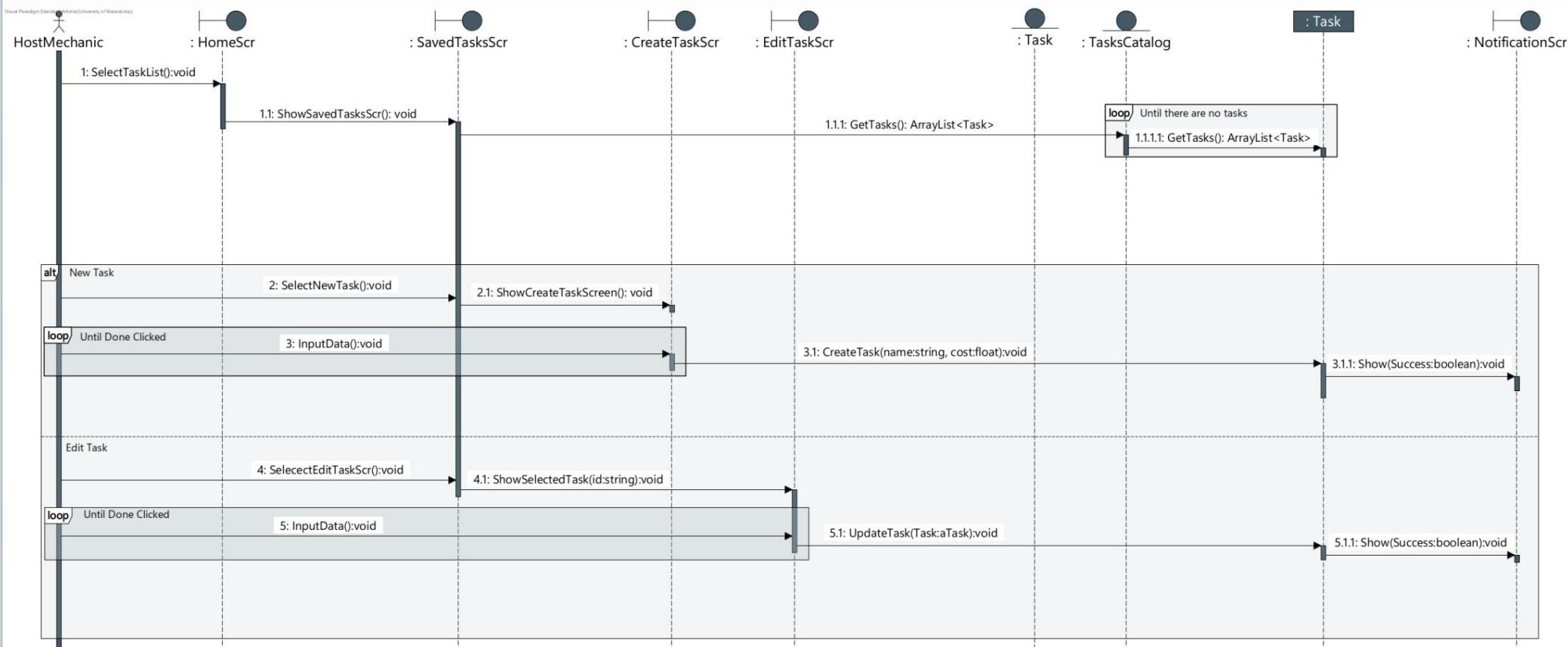
8. "Modify Task List" Basic Flow



3. The host engineer from the job creation screen fills in the fields and clicks "Save"

3.1. The system through the task list creates a new task and displays a message

3.1.T The screen displays the message "Job Creation Successful"



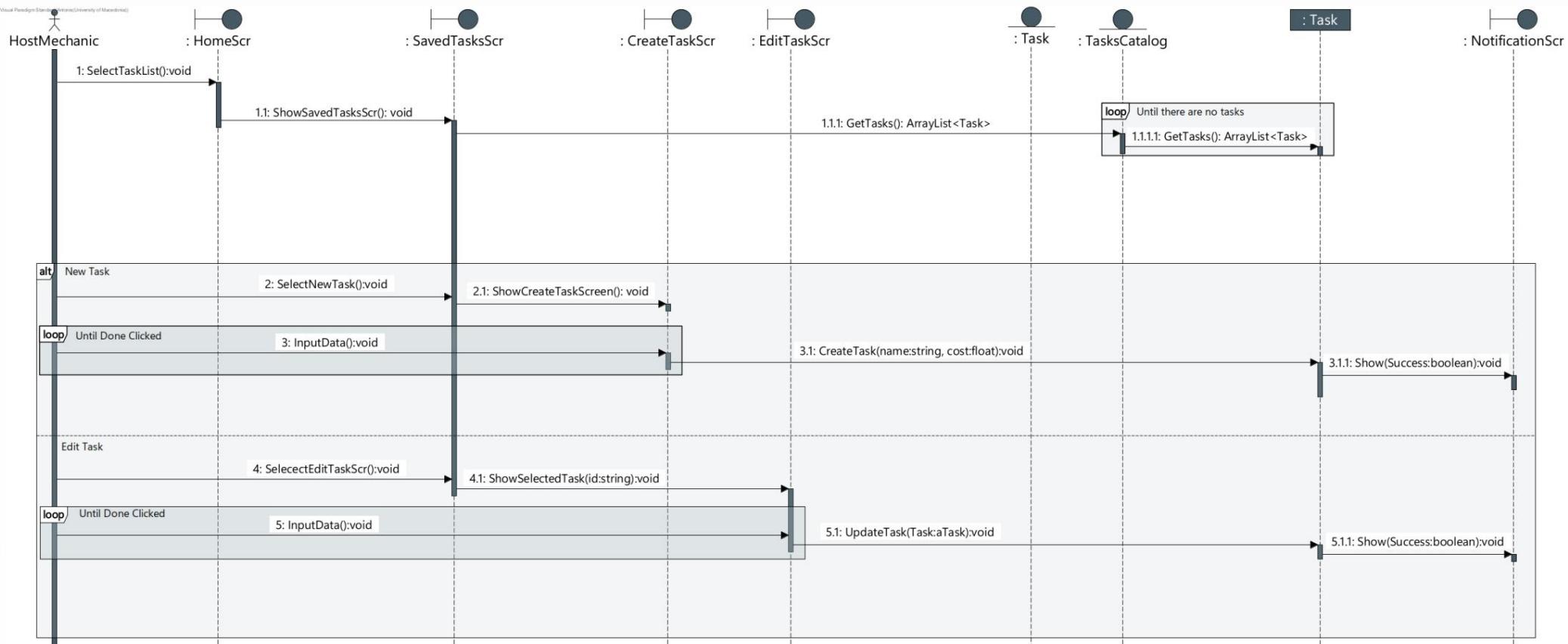
8. "Modify Task List" Basic Flow - Alternate Subflow of 1



4. The host engineer from a task screen presses the "Modify" button

4.1 The system calls the edit task screen (EditTaskScr)

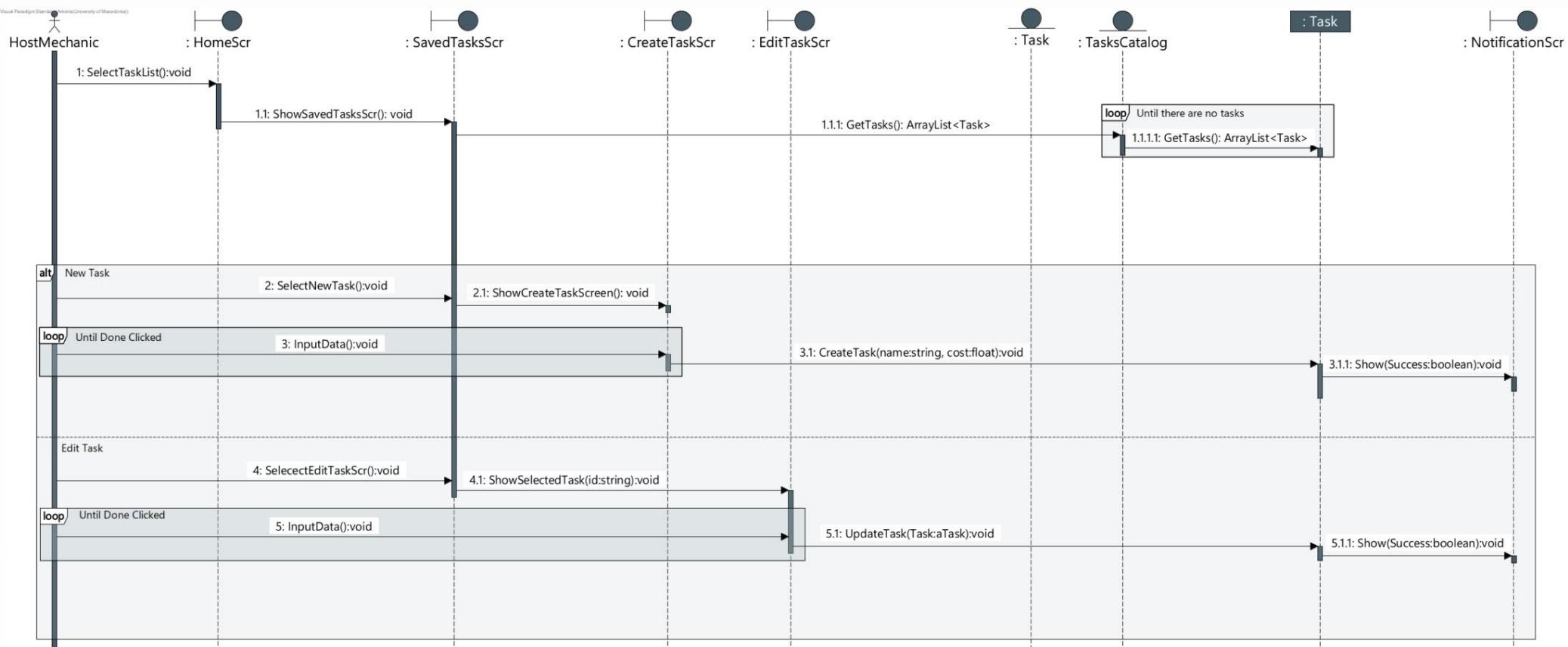
4.1.T The screen shows the job fields and the "Save" button



8. "Modify Task List" Basic Flow - Alternate Subflow of 1



5. The host engineer from the work processing screen updates the fields and presses "Save" 5.1. The system through the task list updates the new task and displays a message
 5.1.T "Job Processing Successful" message appears on the screen



9. "View Assignments" Basic Flow



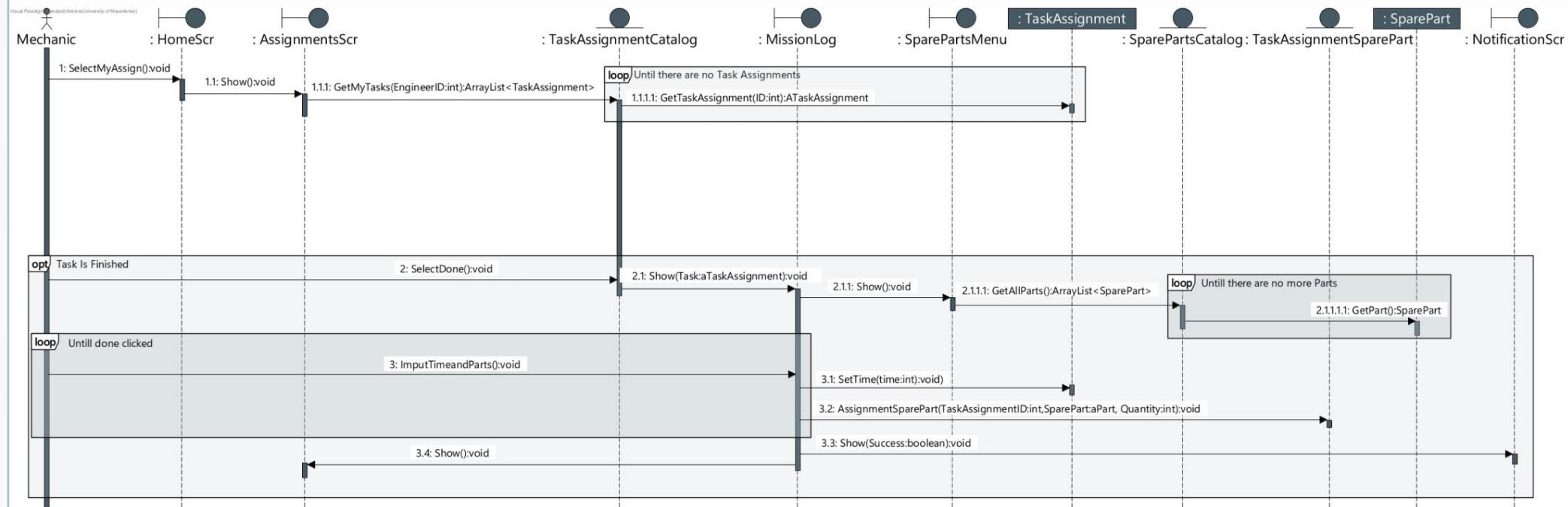
1. The engineer from the home screen presses the "View Assignments" button

1.1. System calls AssignmentsScr

1.1.1. The assignment screen calls the assignment list to populate the assignment list

1.1.1.1. Directory returns a list of tasks assigned to the engineer

1.1.1.1.T The assignments screen shows a list of assignments and next to each assignment is the "Complete" button



9. "View Assignments" Basic Flow

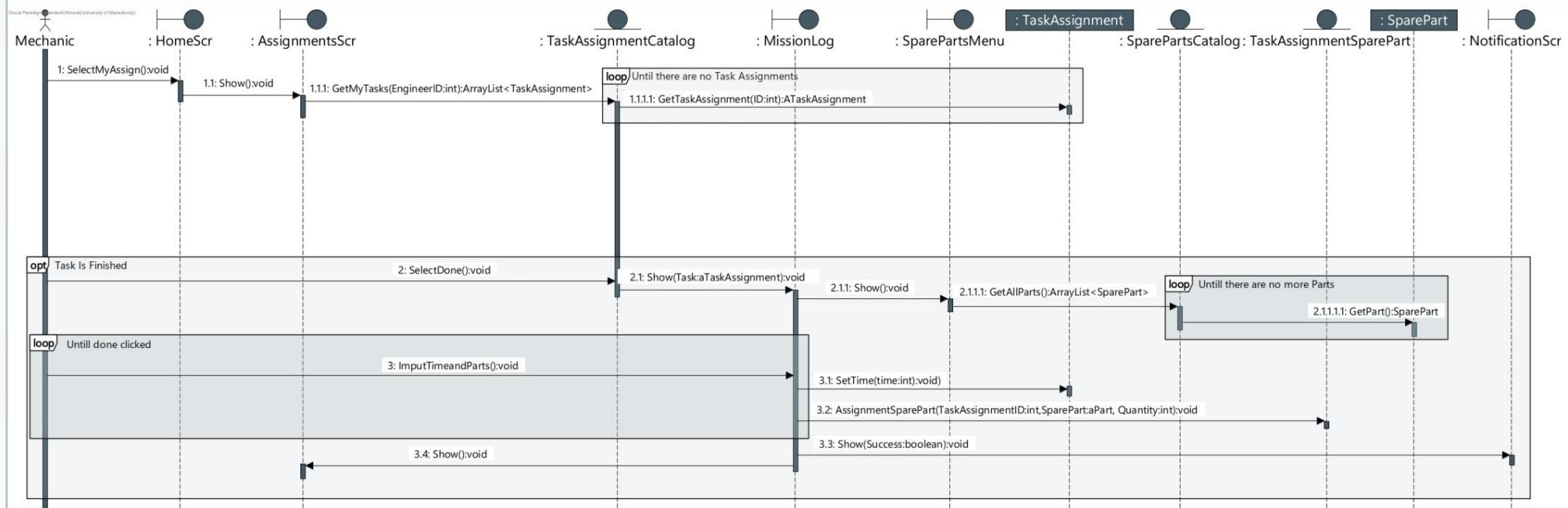


2. If a task has been completed, the engineer from the tasks screen presses "Completion"

2.1 The system calls the mission completion screen (MissionLog)

2.1.1. The completion screen calls up the parts catalog to populate the list of all parts

2.1.1.T The completion screen shows a job duration field, a list of all parts, and a "Finish" button



9. "View Assignments" Basic Flow



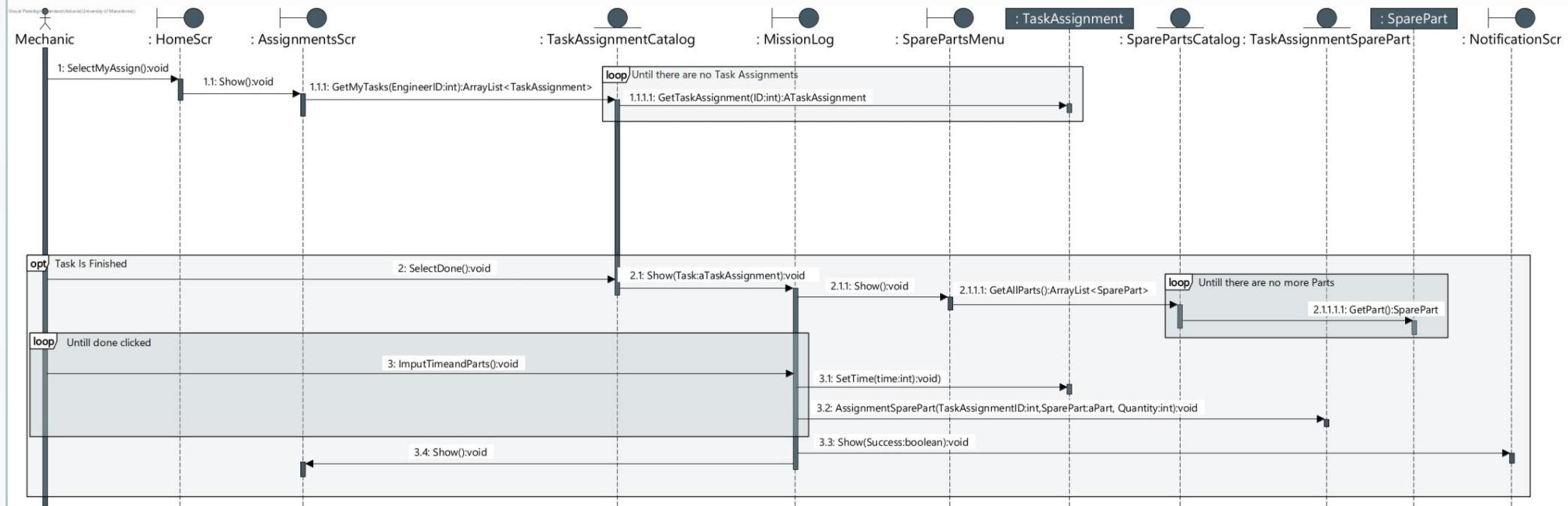
3. The engineer from the task completion screen fills in the fields and presses "Finish"

3.1. The system stores the repair duration

3.2 The system through the job list stores the list of used spare parts

3.3. The system calls up the assignment screen and displays a message

3.3.T. The assignment screen does not show the assignment and shows a successful completion message



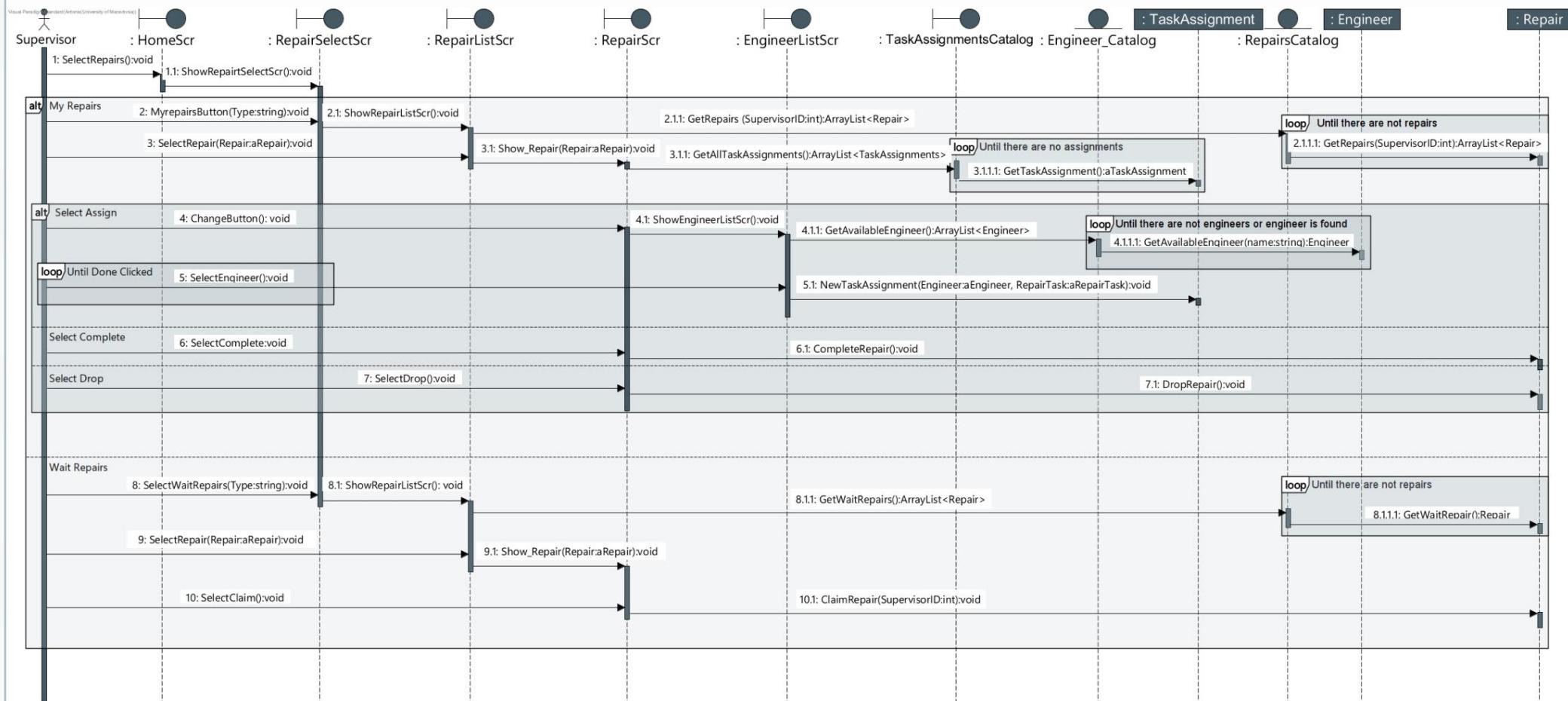
10. "Manage Available Repairs" Basic Flow



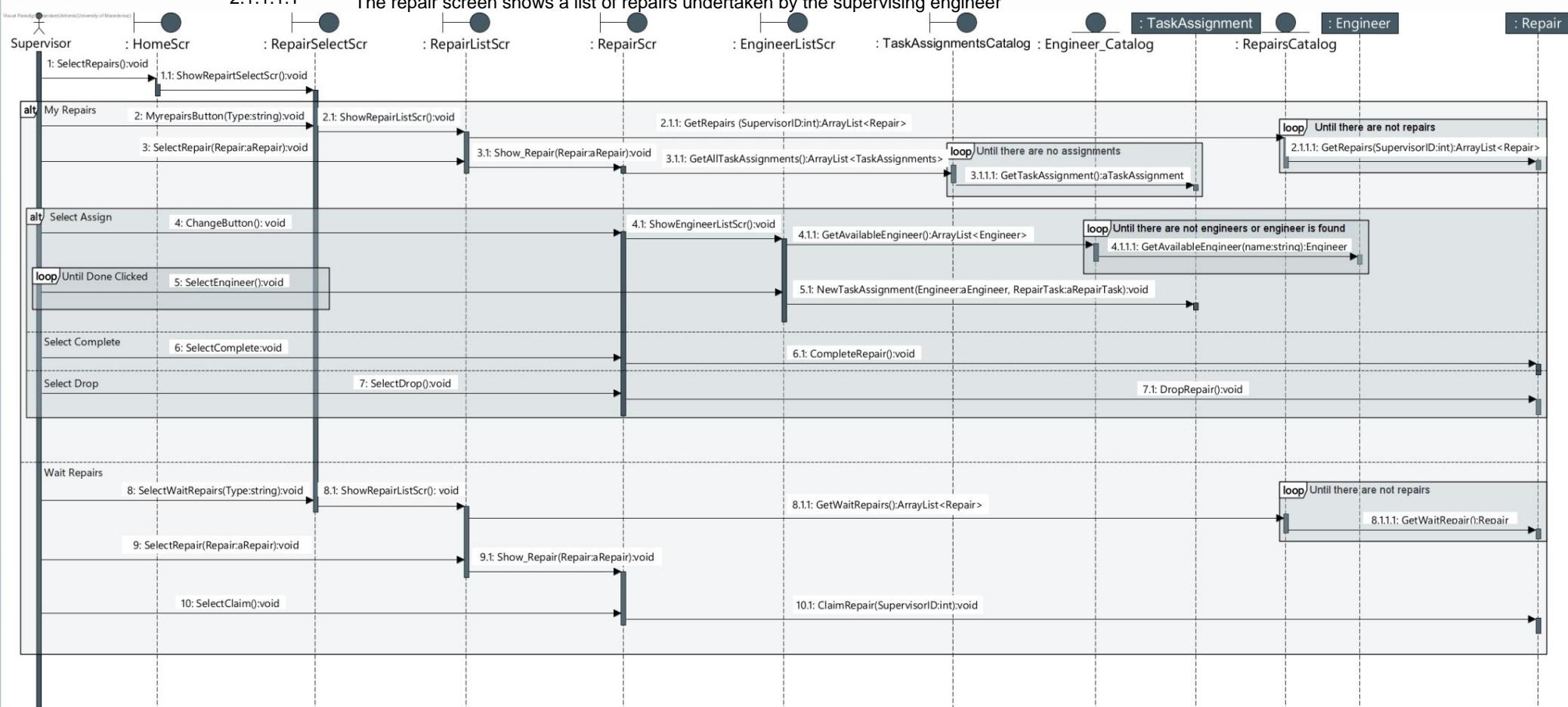
1. The Supervising engineer from the home screen presses the "Manage Repairs" button

1.1 The system calls the selection screen (RepairSelectScr)

1.1.T The "My Repairs" and "Pending Repairs" buttons are visible on the options screen



10. "Manage Available Repairs" Basic Flow



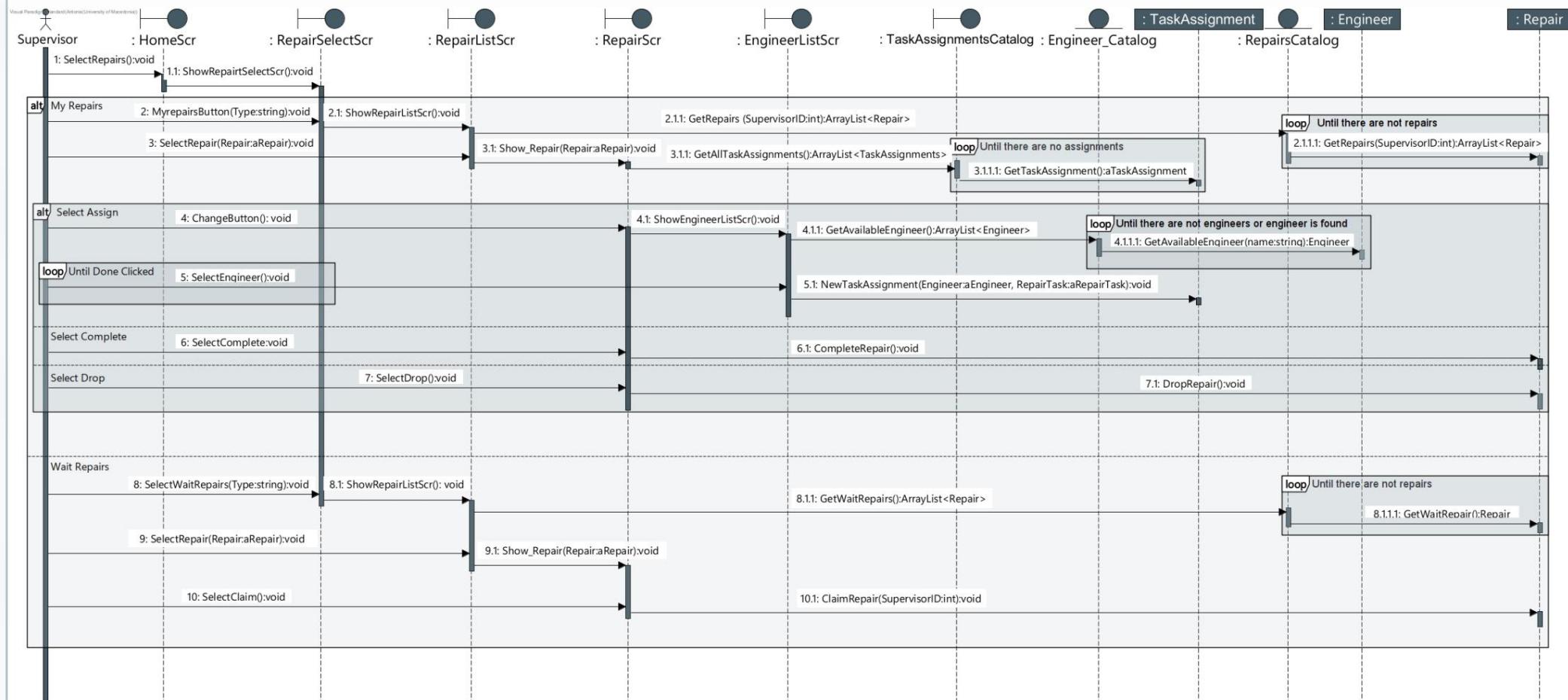
10. "Manage Available Repairs" Basic Flow



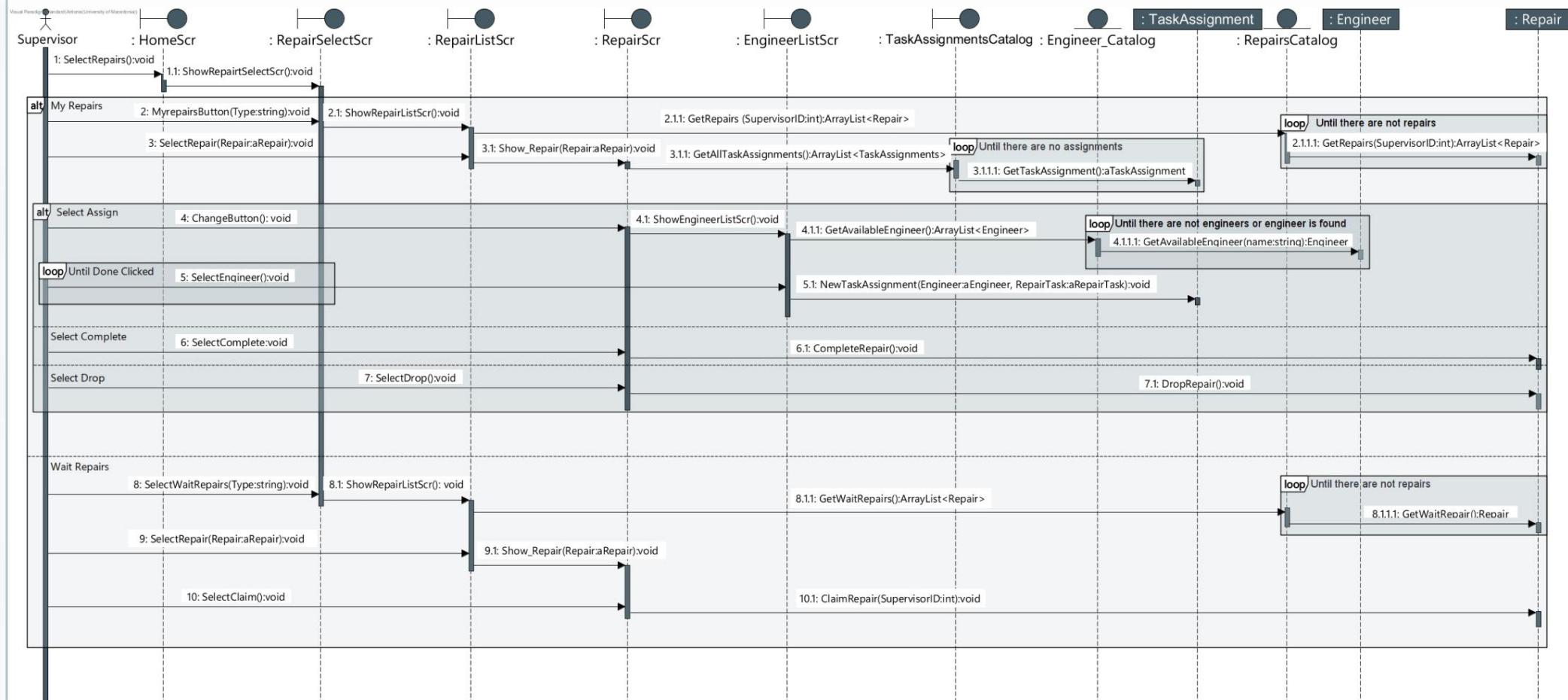
3. The supervising engineer selects the desired repair from the repair screen

3.1. The system calls the detailed repair screen (RepairScr)

3.1.1. The details screen calls up the assignment list to display the list of engineers assigned to this task



3.1.T The screen shows details of the repair, a list of mechanics and the "Change Assignment" buttons, "Complete" and "Leave"



10. "Manage Available Repairs" Basic Flow



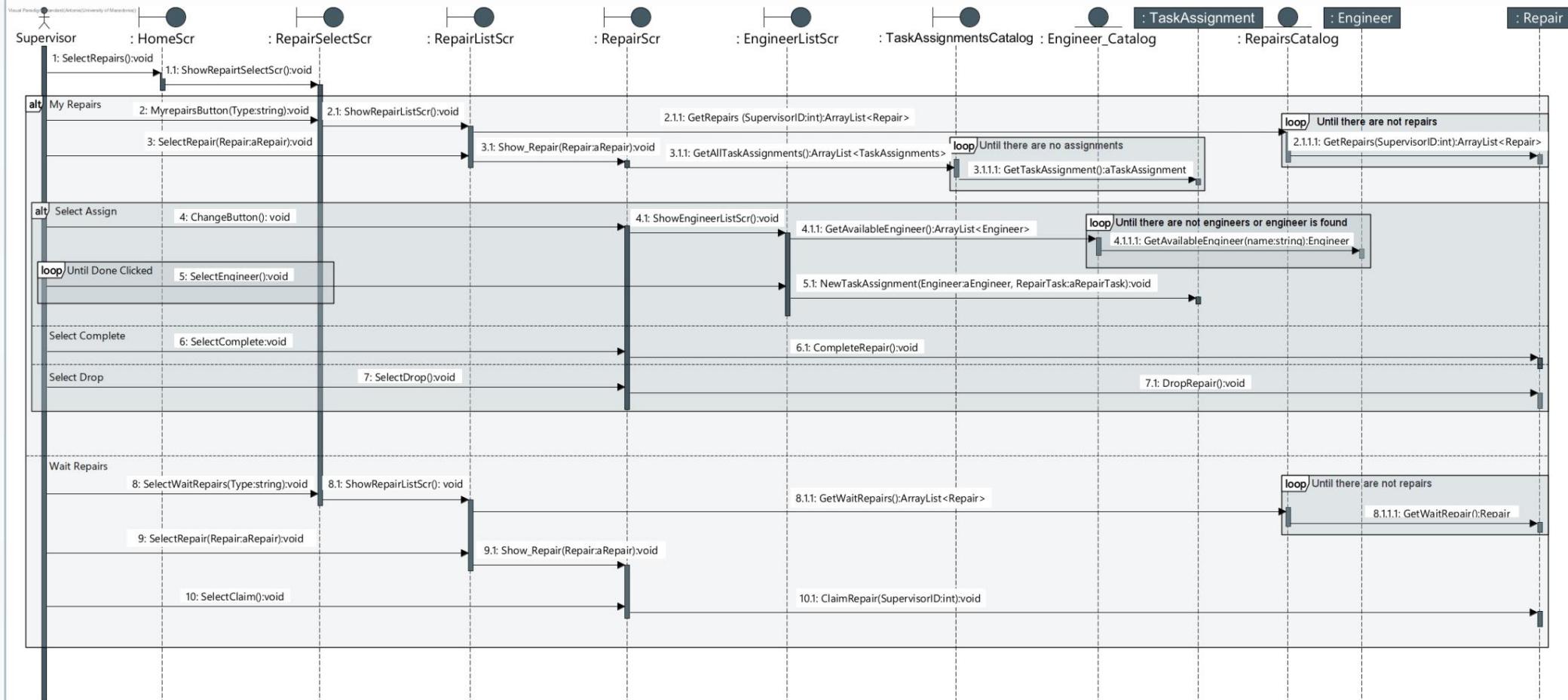
4. The supervising engineer from the repair screen presses the "Change Assignment" button

4.1 System calls engineer screen (EngineerListScr)

4.1.1 The mechanics screen calls the mechanics list to populate the mechanics list

4.1.1.1 The engineers list returns the names of all engineers

4.1.1.1.T The mechanics screen shows a list of all the mechanics



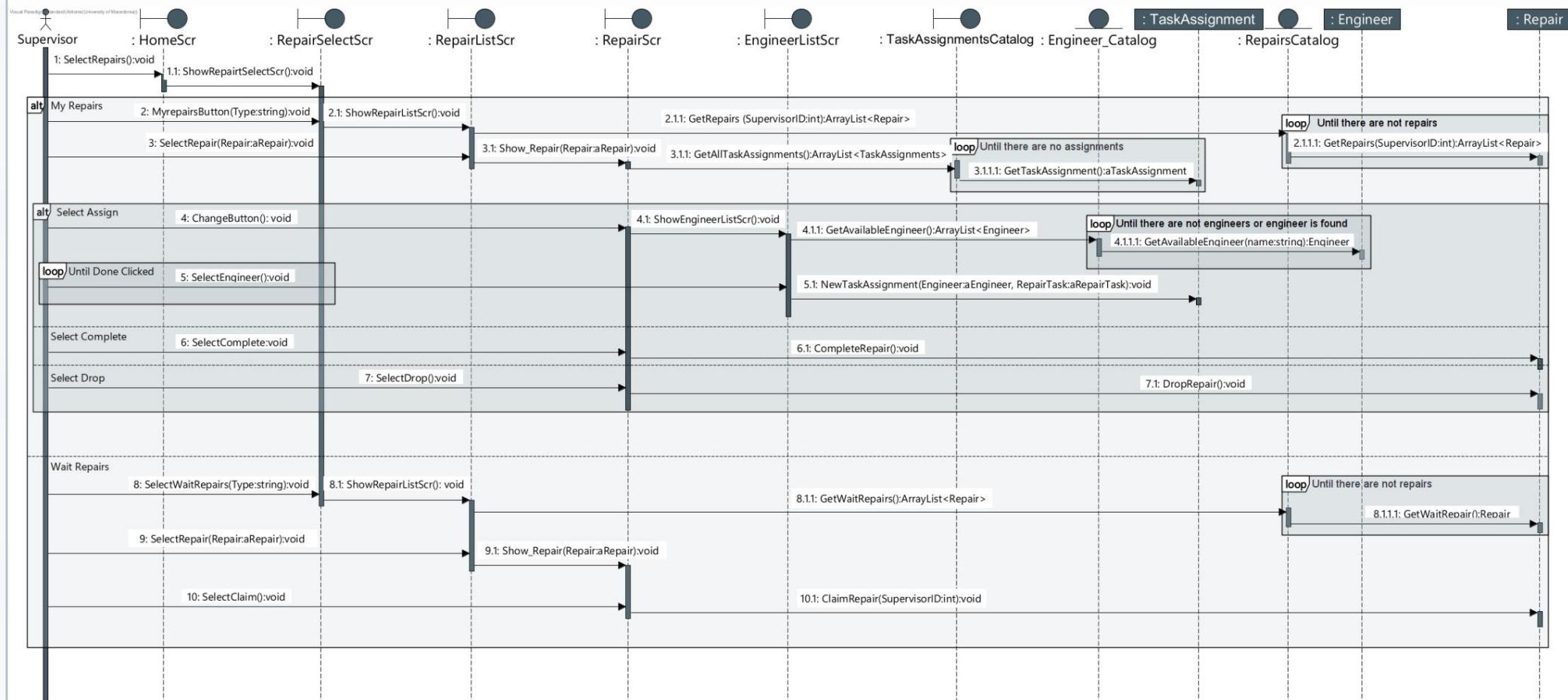
10. "Manage Available Repairs" Basic Flow



5. The supervising engineer from the engineers screen selects engineers for the repair and then "Done"

5.1. The system updates the repair details in the catalog and displays a message

5.1.T The repair screen shows a message of successful processing



10. "Manage Available Repairs" Basic Flow - Cont. Underflow of 1



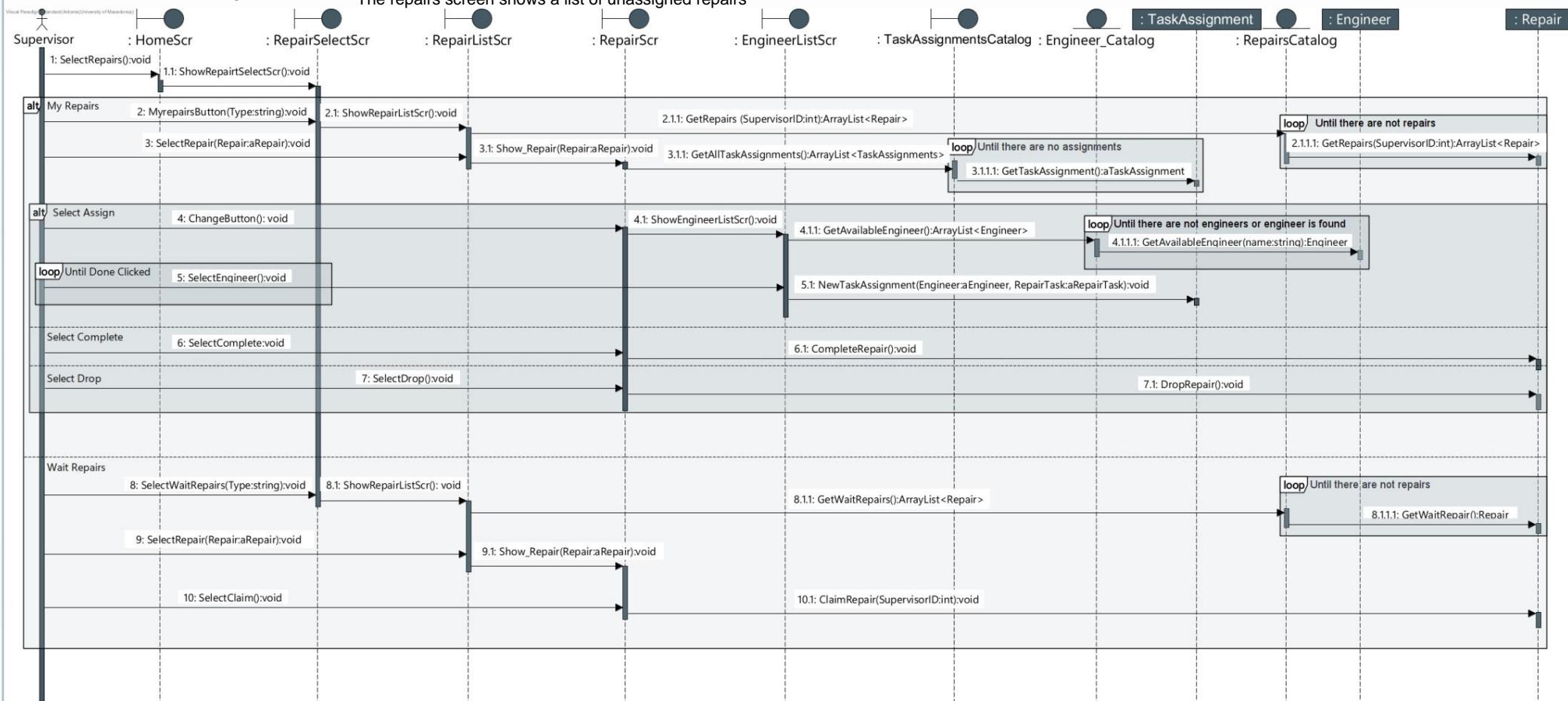
8. The Supervising Engineer from the options screen taps the "Pending Repairs" button

8.1 The system calls the repair screen (RepairListScr)

8.1.1 The repair screen calls the repair catalog to populate the repair list. The repair catalog returns the repairs that have

8.1.1.1 not been undertaken

8.1.1.1.T The repairs screen shows a list of unassigned repairs



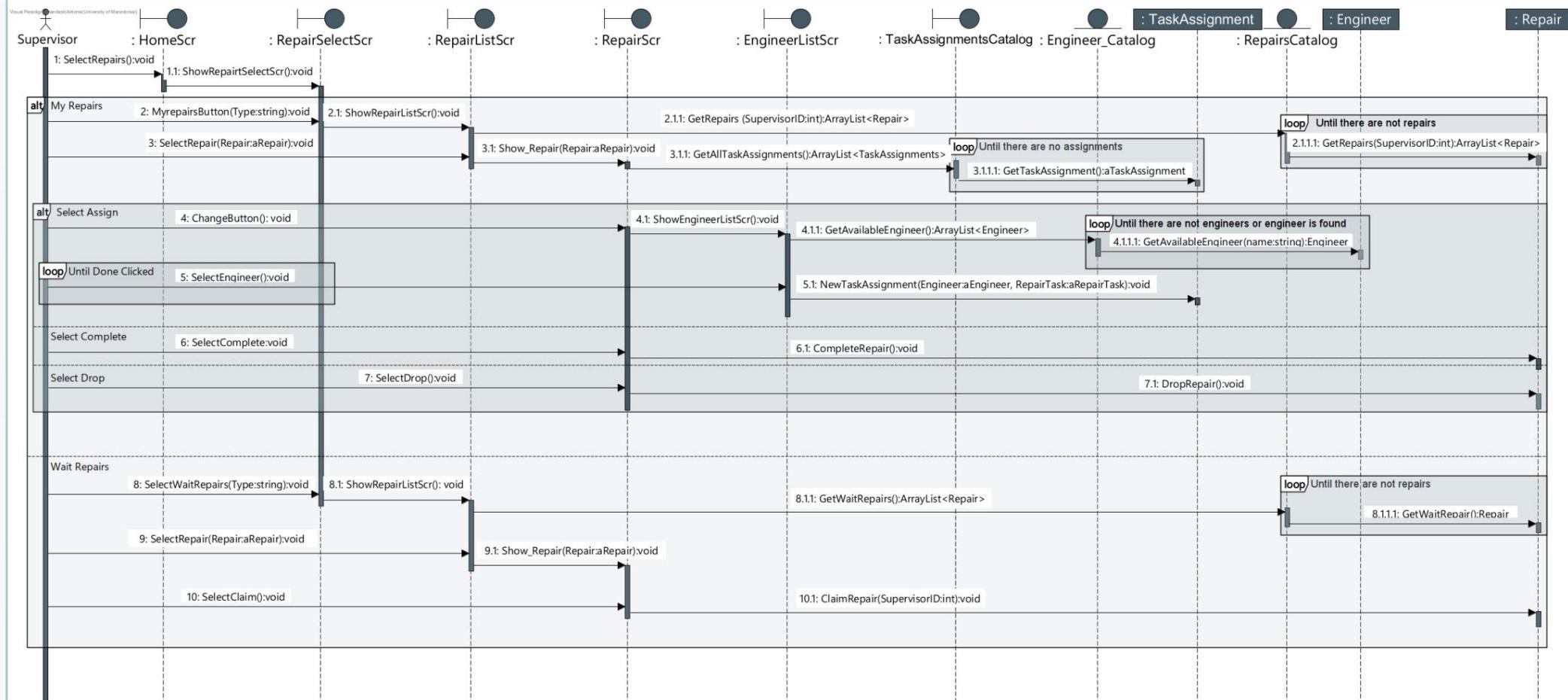
10. "Manage Available Repairs" Basic Flow - Cont. Underflow of 1



9. The supervising engineer selects the desired repair from the repair screen

9.1 The system calls the detailed repair screen (RepairScr)

9.1.T The screen shows the details of the repair and the "Accept" button



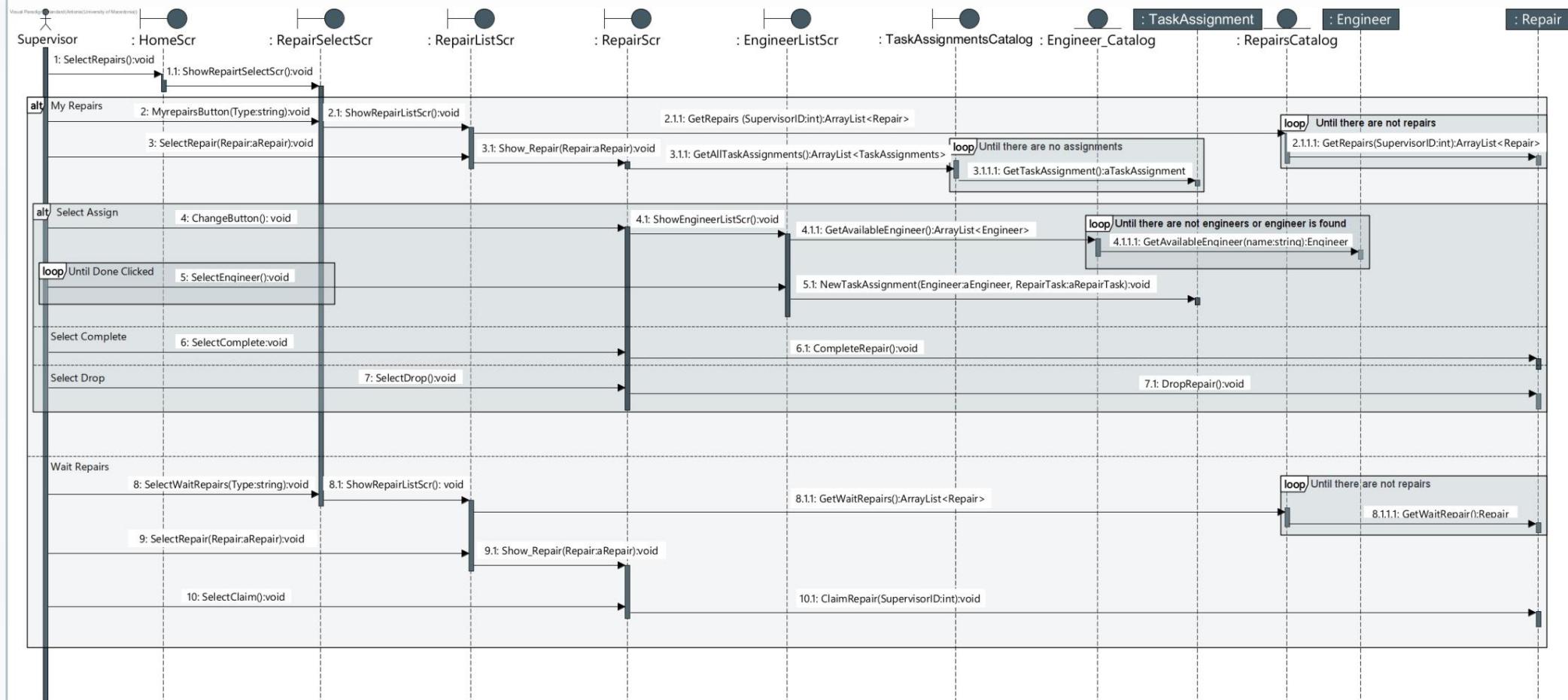
10. "Manage Available Repairs" Basic Flow - Cont. Underflow of 1



10. The supervising engineer from the repair screen presses the "Take over" button

10.1 The system updates the details of the repair in the catalog and displays a message

10.1.T A successful processing message appears on the repair screen



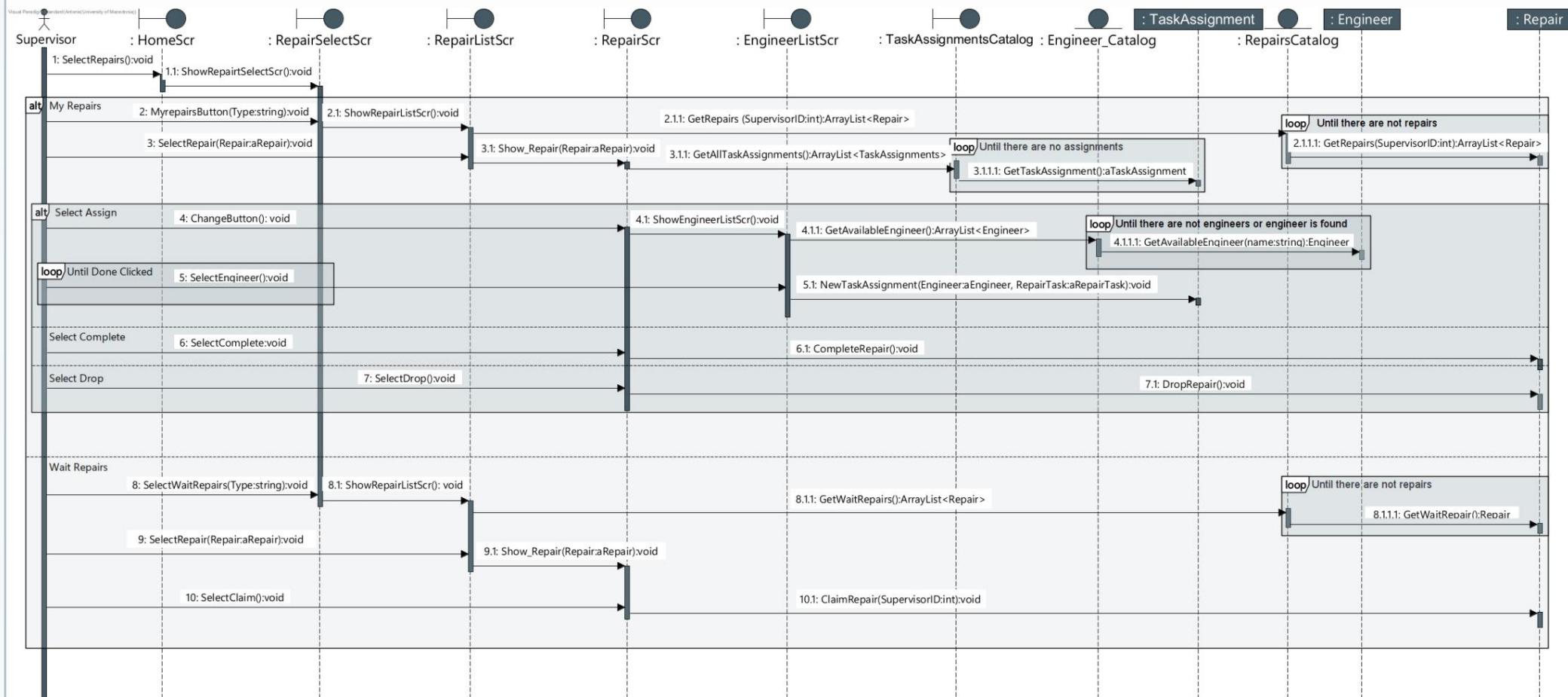
10. Basic Flow "Manage Available Repairs" - 1st Gen. Underflow of 3



6. The supervising engineer from the repair screen presses the "Finish" button

6.1 The system updates the repair as complete in the repair list

6.1.T The repair screen shows a message of successful processing



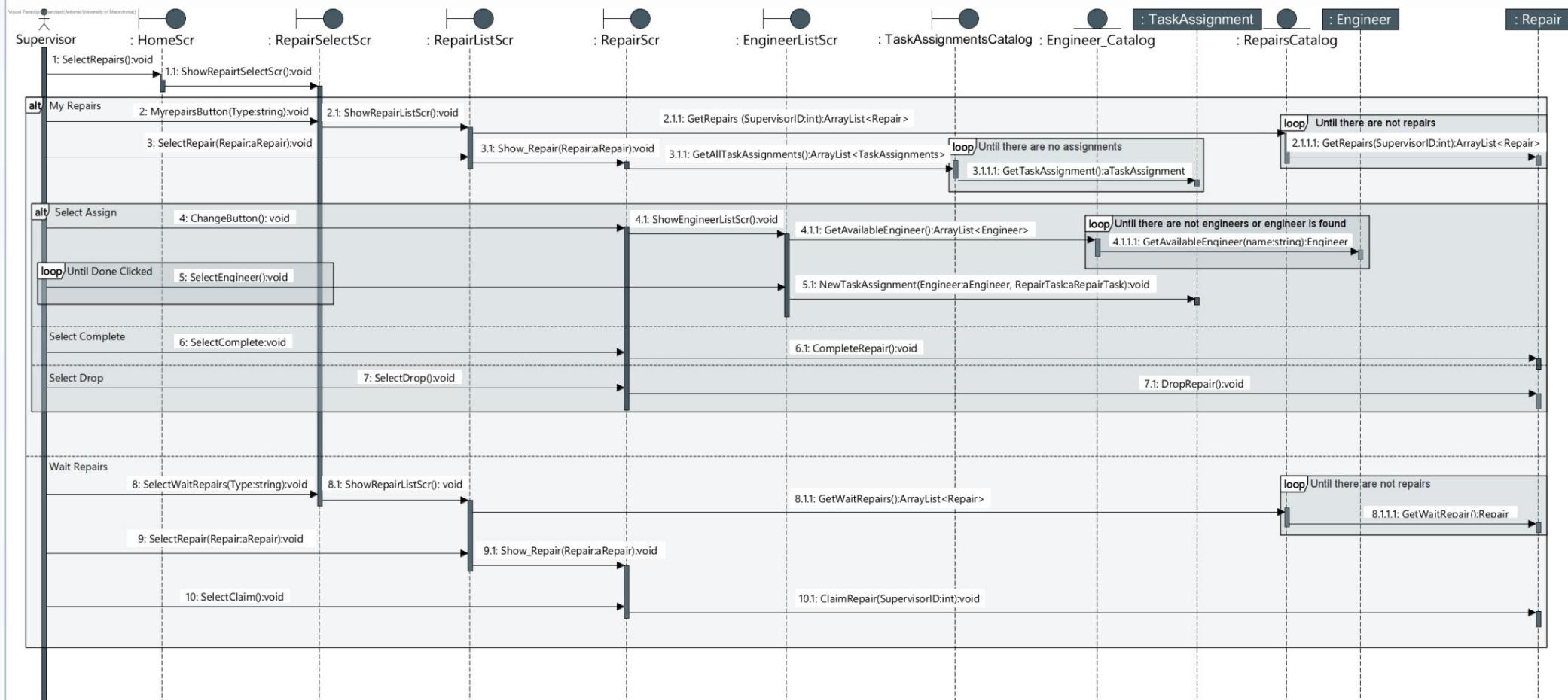
10. Basic Flow "Manage Available Repairs" - 2nd Gen. Underflow of 3



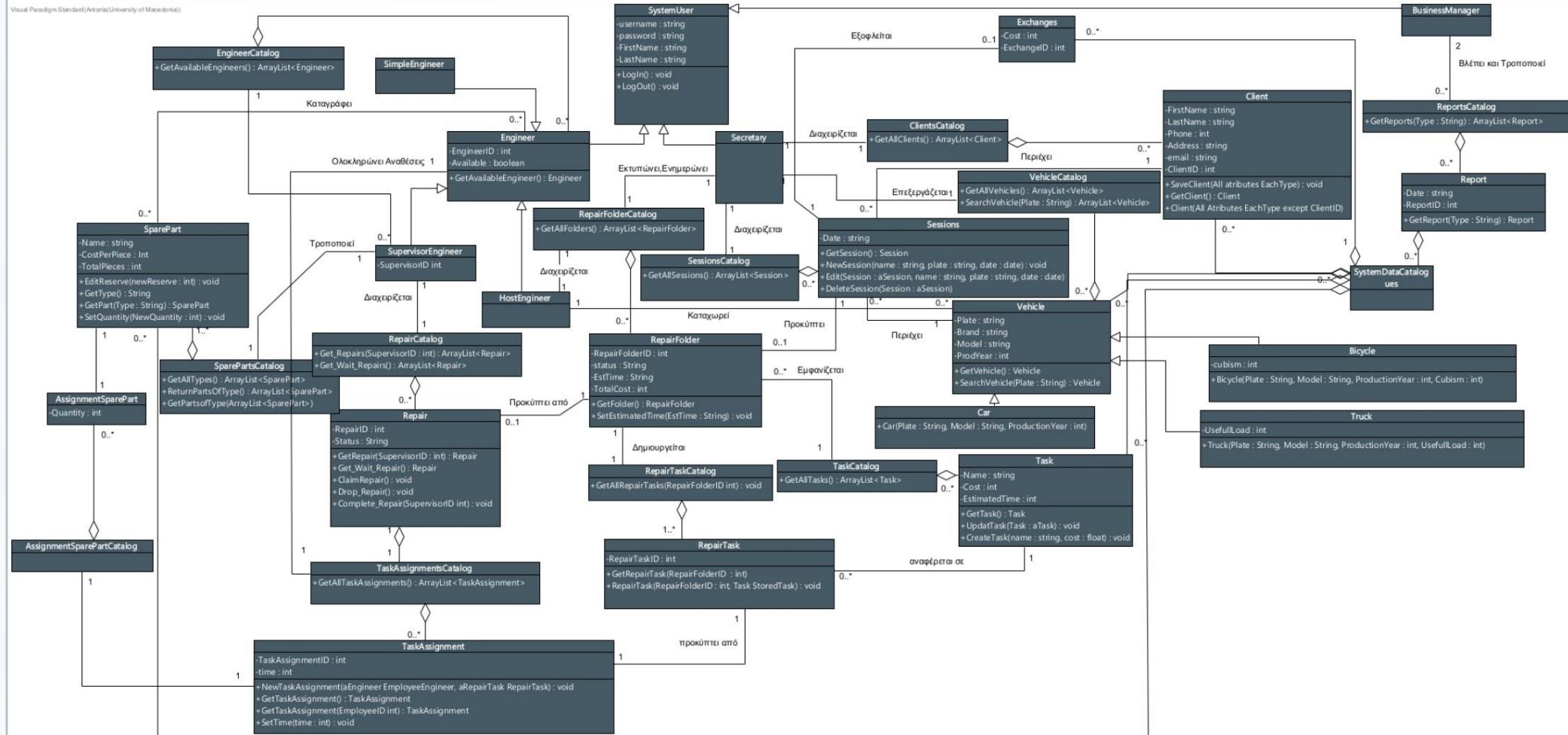
7. The supervising engineer from the repair screen presses the "Exit" button

7.1 The system updates the repair as not assigned in the repair list

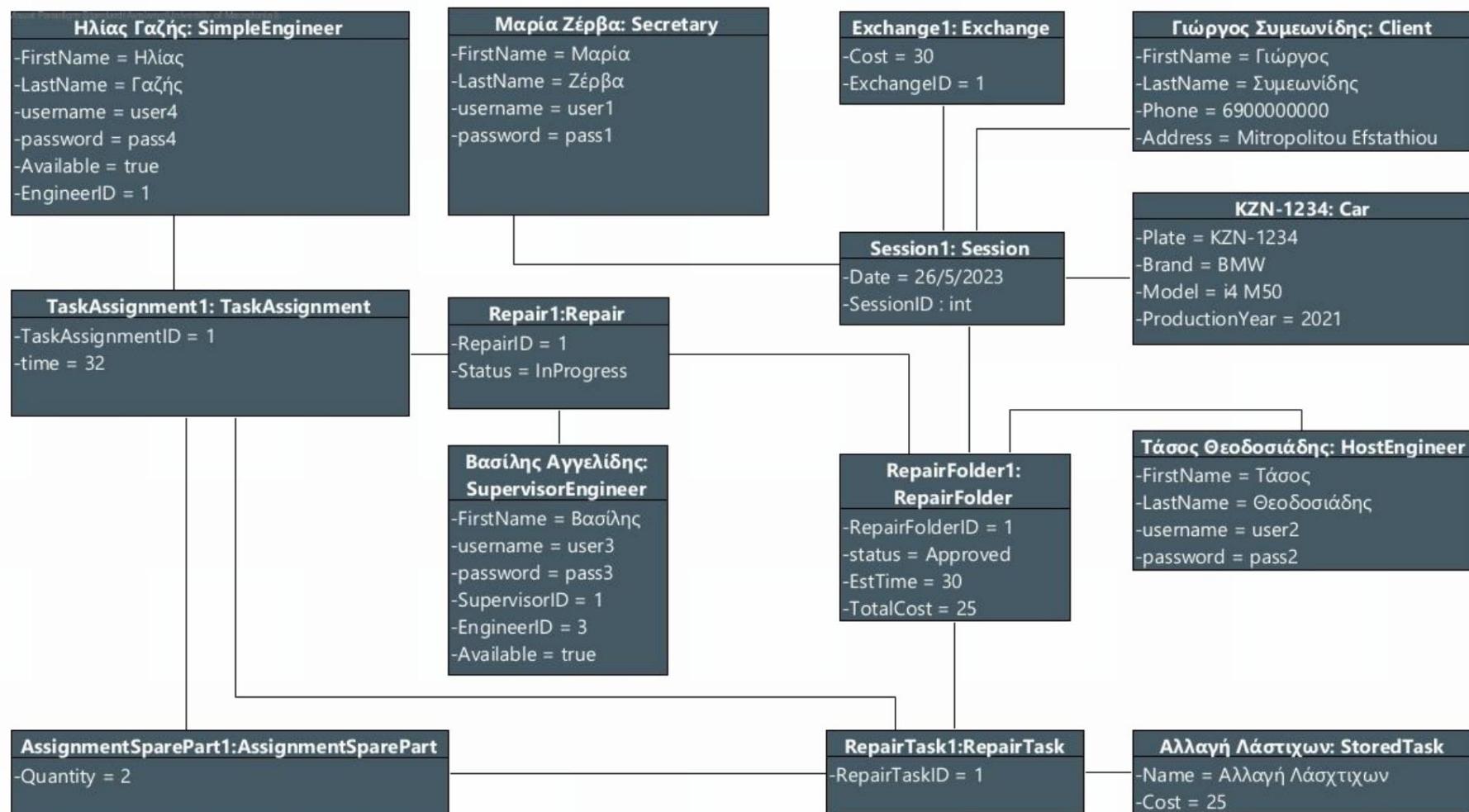
7.1.T The repair screen shows a message of successful processing

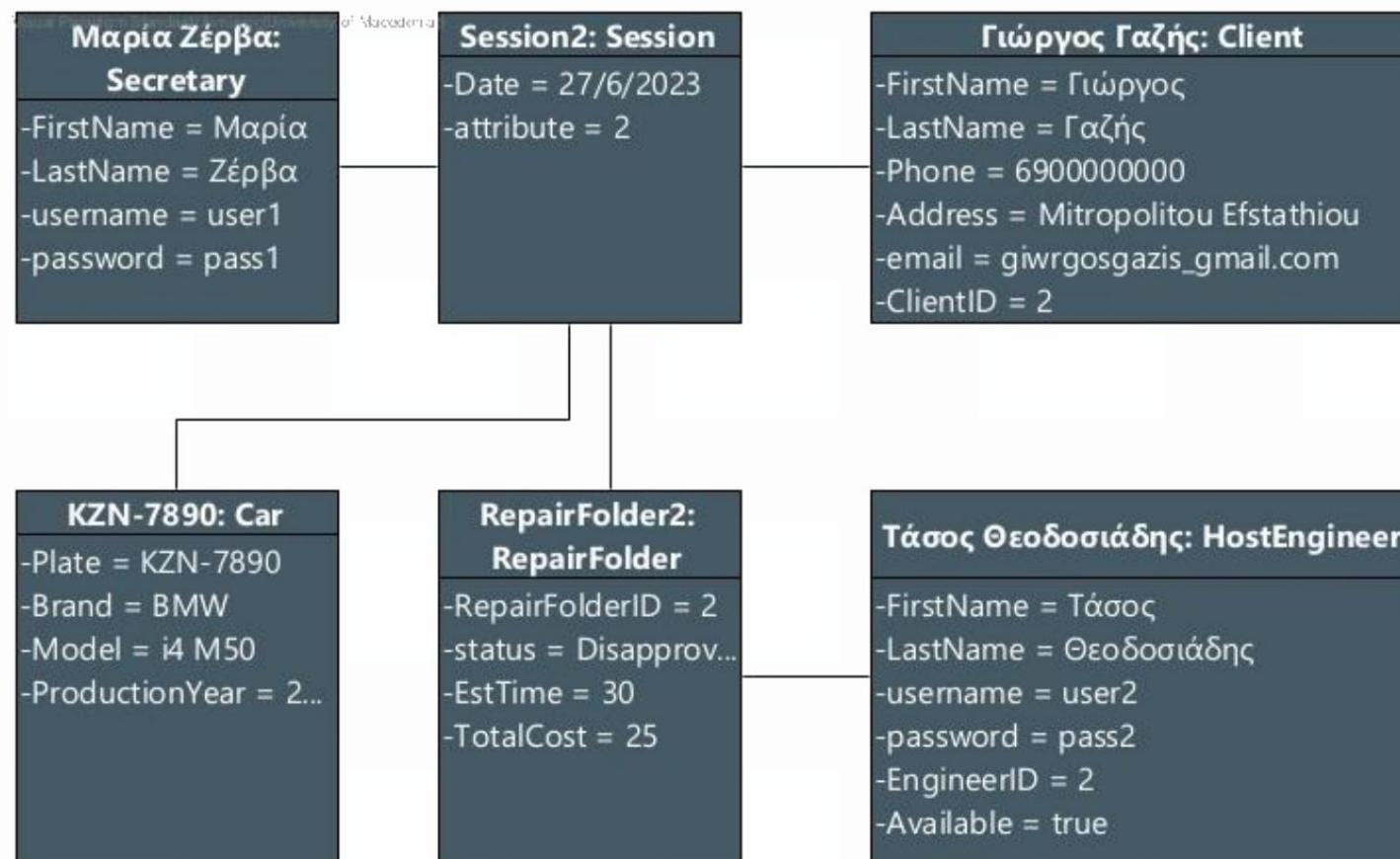


4.2 Class Diagram



4.3 Object Diagrams







IMPLEMENTATION



Execution instructions

The team completed the requested function of main and tried to build an additional GUI which was ultimately not completed by the submission date. However both Mains are **functional**.

Main Execution - MainWithoutScreens

Run through Eclipse

This main does not contain any external library and no additional ones are needed steps to run through Eclipse.

After it is imported from Eclipse and you open the MainWithoutScreens class it can be run by pressing Run.

Run GUI – Main

Because the team hasn't had time to implement the GUI, if you jump to it by the time this document is read on GitHub there's a chance it will be complete

Run via Bat

Click on the Clickme Bat file inside the CarOps - Run folder

Run via Eclipse 1. Download

the uploaded Zip file or clone the folder from GitHub 2. Install the e(fx)clipse

plugin from the online plugin store _____

of Eclipse

3. Install the JavaFX library.

From the Window menu select Preferences->Java->BuildPath->UserLibraries. Click the New option and type **JavaFX** as input , then click Add External JARs. Inside the unzipped folder go to this location CarOpsJavaProject\CarOps With GUI\lib\javafx-sdk-20.0.1\lib. Select all items in the folder and press Apply and Close. 4. Setting the Run Configurations.

From the Eclipse toolbar click on the Run Button (Green color only) down arrow. Then select Run Configurations->Arguments. Fill in the VM arguments box the command:

```
--module-path "lib\javafx-sdk-20.0.1\lib" --add-modules  
javafx.controls,javafx.graphics,javafx.fxml" --add-modules  
javafx.controls, javafx.graphics,javafx.fxml
```

Code Structure

GitHub Repository: <https://github.com/iis22023/CarOpsJavaProject>

Vehicle: Represents a vehicle object with properties such as License Plate, Make, Model and Year of Manufacture.

SystemUser: Represents a system user with properties such as username, password, first name, last name

Engineer : Represents a system engineer who extends to a system user and checks if he is available or not

VehicleCatalog : Represents a catalog of vehicles

Task : Represents the task with properties such as name, cost, and estimated time

SparePart : Represents spare parts with properties such as Name, Cost per Part, and Total Parts

TaskAssignments : Represents each engineer's task with properties such as Engineer ID, Repair ID, Status, Task Assignment ID, and Time

Report: Represents reports with parameters such as Date and Report ID

Repair : Represents repairs with parameters like repair id, folder repair, status, registry and total repairs

RepairFolder : Represents repair folders with parameters such as session, repair, repair folder, status, estimated time, total cost, and registry

Session : Represents the session of each repair with parameters such as date, session id, total sessions and folder id

Exchanges : Represents exchanges with parameters such as cost and exchange id

Registry : Contains the lists to be edited of the JavaProgram and the loops

Main : The entry point of the application, contains the main method to start the program.



CONCLUSIONS



Conclusions

Requirements Analysis & Understanding

Through the analysis of CarOps workshop requirements and preparation of the project, the team was led to important conclusions.

A deeper understanding of the company's business processes, as well as the client's needs, was necessary before developing the software. In addition, the value of organized collaboration to the successful completion of the project was confirmed, as the team had the opportunity to use various applications and tools in a collaborative environment.

Finally, it became clear that proper preparation is a cornerstone for success software design and development.

Planning

During the design of the system the team gained a deep understanding of UML diagrams which proved to be useful tools for understanding the operation of the system but also the broader visualization of user requirements.

Furthermore, by implementing sequence diagrams the team was able to deepen their understanding of user requirements, making a first introduction to the principles of object-oriented design.

Programming & Object Orientation

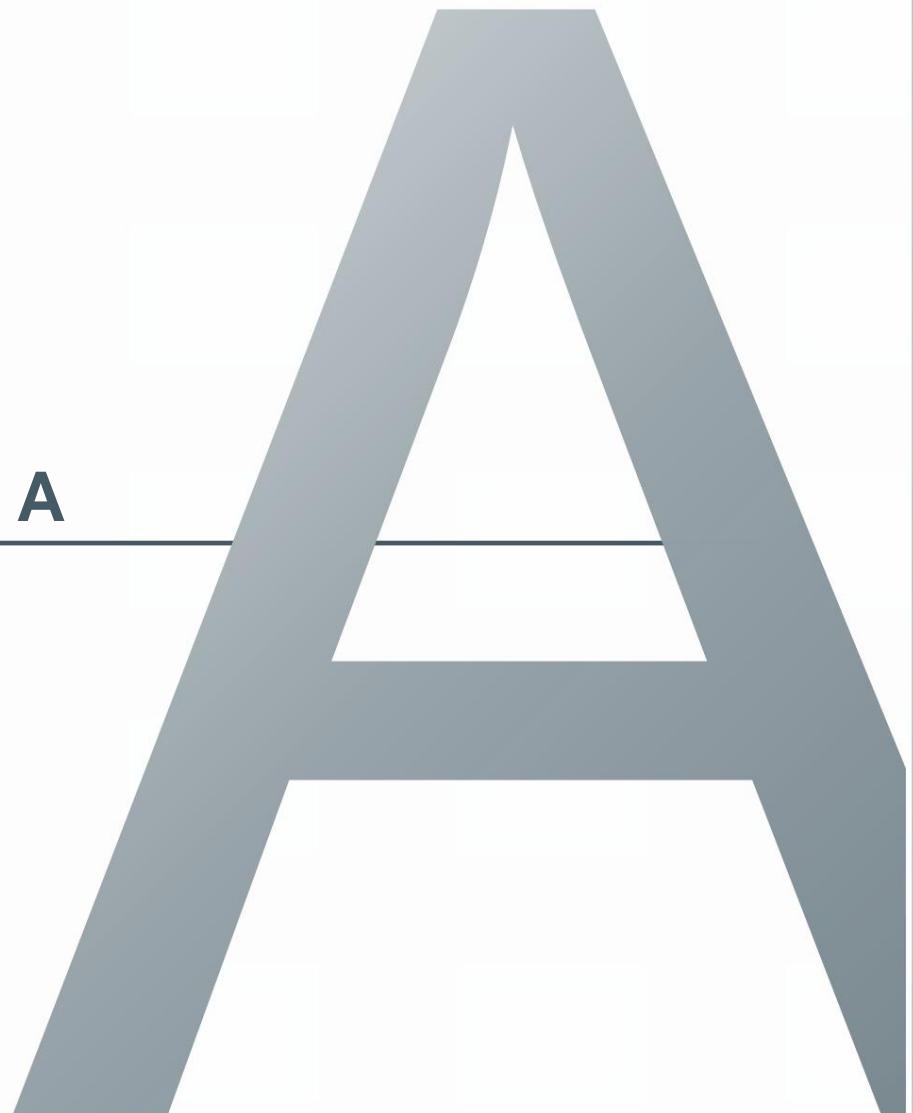
In the planning phase, the team managed to give concrete form to what was initially considered vague, answering all the questions of Who, What, Why, How, and proceeded to implement the system.

Coding in the Java programming language proved to be a pleasant process for developing the system, as it offered object-oriented techniques, better managing the complexity of the system classes.



APPENDIX A

Project management



Team Organization and Roles

During the group's first teleconference, the need to define a way of cooperation by establishing some procedures and assigning roles to all five members was deemed urgent.

In order to control and coordinate the team, it was decided to hold weekly teleconferences of a maximum duration of 1 hour where the week's slides will be reviewed first, tasks assigned in the previous teleconference will be checked and new ones will be assigned to the members based on the role they have chosen.

Clock

The team considered it appropriate to have a Project Manager to direct the other members, organize them, break down the requests into smaller ones and delegate them. This role was given to Antonis Giannakovitis.

Konstantinos Psyllas and Dimitris Lambrianidis chose the role of Business Analyst, Konstantinos specializes in representing business requirements and Dimitris in designing these requirements.

In the team it was considered necessary to have a Software Architect very important for software and technical issues. This role was given to Christos Sofianopoulos.

Finally, it was considered very important to have a Database Manager to be responsible for processing the data that will be produced for the company. This role was given to Nikos Siatra.

For each part of the work we decided to follow a flexible way of management. The Project Manager has first say in all three issues, then the Business Analyst would be responsible for the analysis and design issue, and the Software Architect would be responsible for the implementation issue.

Work style

After discussion the group was led to the conclusion that the issues are quite specific and therefore needed special attention and organization in the way they will be assigned and limited in time. The team in thinking to be assigned each issue to a single member was counterproductive because the burden was too great for a single person and would lack the feedback that exists in a group setting.

After further discussion, and with the final decision of the Project Manager, defined the way of working as well as assigning tasks to three or two members only since subgroups with this number were the key for maximum flexibility and proper coordination.

Tools and Applications

Asynchronous Communication & Repository

For the asynchronous communication of the group, the Google platform, Google Chats, was used since all members had a Google Workspace account and with each group on the platform a storage space was created on Google Drive. Finally, it was preferred for its simplicity but also the possibility to create separate communication channels within a group.

Modern Communication

The team's video conferences were held via the Google Meet platform for Business which is integrated into Google Chats.

Work Organization

Atlassian, Jira and Confluence products were used to organize the team. They greatly helped simplify the team's goals by highlighting the most urgent work each time.

Content Production

The Visual Paradigm platform was used to create the activity diagrams, use cases and classes, while the Balsamiq platform was used to create the screenshots in the verbal descriptions.

Finally, Microsoft Office 365 products were used both for the presentations during the teleconferences and for the creation of all documents.

Programming & Debugging

The Eclipse programming environment was used to write code in JAVA, while Large Language Models (LLM) such as ChatGPT and Copilot X were used for debugging. Finally, the code was hosted on GitHub.

Timesheets

Timesheet Giannakovitis Antonis

Tasks – 1st Deliverable	Duration
Implementation	
Staff Summary – Task E2	2 hours
Feasibility Study – Task P2 How to	3 hours
Gather Requirements – Task A2	1 hour
Backlog & User Stories – Task A3 Verbal	5 hours
Descriptions – Task A5 Quality	6 hours
Assurance	
System Application – Task P1	1 hour
Feasibility Study – Task P2 Lists &	1 hour
Diagrams – Task A1	6 hours
Backlog & User Stories – Task A3 Use	2 hours
Case Diagram – Task A4 Verbal Descriptions –	1 hour
Task A5 Other	3 hours
Meeting Scheduling Scrum Management Task	3 hours
Delegation and	3 hours
Team Management Meetings	6 hours
Teleconferences	8 hours
Assignments – 2nd Deliverable	
Teleconferences	5 hours
Sequence Charts – Task D1	22 hours
Assignments-3rd Deliverable	
Video Conferencing	10 hours
Code Generation Core Part Code	10 hours
Generation Graphical Interface	22 hours

Timesheet Lambrianidis Dimitrios

Tasks – 1st Deliverable	Duration
Implementation	
Introduction – Task E1	1 hour
Executive Summary – Task E2	1 hour
Lists & Diagrams – Task A1 Use Case	3 hours
Diagram – Task A4	3 hours
Domain Model – Task A6	2 hours & 30 minutes
Quality assurance	
Staff Summary – Task E2 Meetings	20 minutes
Teleconferences	4 hours
Tasks – 2 the Deliverable	
Teleconferences	4 hours
Sequence Charts – Task D1 Assignments –	6 hours
3 the Deliverable	
Video Conferencing	2 hours
Create Part Code	1 hour

Timesheet Siatras Nikolaos

Tasks – 1st Deliverable	Duration
Implementation	
System Application – Task P1	45 minutes
Feasibility Study – Task P2 How to	45 minutes
Gather Requirements – Task A2	1 hour & 45 minutes
Backlog & User Stories – Task A3	2 hours
Domain Model – Task A6	2 hours & 30 minutes
Quality assurance	
System Application – Task P1	30 minutes
Feasibility Study – Task P2 Lists &	30 minutes
Diagrams – Task A1 Use Case Diagram	30 minutes
– Task A4 Verbal Descriptions – Task A5	30 minutes
	5 hours
Domain Model – Task A6	30 minutes
Other	
Task Assignment and Team Management Task A6, A2 Meetings	1 hour
Teleconferences	8 hours
Tasks – 2 the Deliverable	
Teleconferences	5 hours
Sequence Charts – Task D1 Class Chart – Task	7 hours
D2 Assignments – 3rd Deliverable	4 hours
Video Conferencing	10 hours
Deliverable Tasks Code	8 hours
Generation Core Part Code Generation	16 hours
Graphical Interface	4 hours

Timesheet Sofianopoulos Ioannis-Christos

Tasks – 1st Deliverable	Duration
Implementation	
Introduction – Task E1	1 hour
System Application – Task P1 Lists	1 hour
& Diagrams – Task A1 Use Case	3 hours
Diagram – Task A4 Verbal Descriptions – Task A5	3 hours
Quality Assurance	1 hour & 30 minutes
Verbal Descriptions – Task A5	20 minutes
Meetings	
Teleconferences	4 hours
Tasks – 2 the Deliverable	
Teleconferences	5 hours
Sequence Charts – Task D1 Assignments – 3rd	6 hours
Deliverable	
Video Conferencing	6 hours
Code Generation Core Part	24 hours

Timesheet Psillas Konstantinos

Tasks – 1st Deliverable	Duration
Implementation	
System Application – Task P1 Lists	1 hour
& Diagrams – Task A1 Use Case	3 hours
Diagram – Task A4	3 hours
Domain Model – Task A6	2 hours & 30 minutes
But	25 minutes
Task Assignment and Group Management Task A4 Meetings	1 hour
Teleconferences	4 hours
Tasks – 2 the Deliverable	
Teleconferences	4 hours
Sequence Charts – Task D1 Assignments	6 hours
3rd Deliverable	
Teleconferencing	2 hours
Deliverable Tasks Code	1 hour
Generation Core Part	8 hours



APPENDIX B

Team Evaluation



Team Evaluation

Work Self-Assessment

Assessment

- Grade of work in terms of effort: **10**
- Degree of work in terms of the final result: **9**
- Final grade of your group work: **9**

Justification

As a team and individually, we believe that there is no other job that we have put more effort into, and for which we deserve the highest grade.

From the first days, when the teams were created, we started working on our own "Project Launch". With the information available to us, we simplified the requirements of the first deliverable, defined roles, and each of us was open about how much we were willing and able to work on.

We used to finish the speech prompts pretty early, but we always ended up turning in our work on the last day at 11:55pm because we always had a lot to add and shape. We aimed not only to list the requirements and "get rid of them", but to cover the enunciation and the theory.

But primarily, we want them to work, to look, to look good, to the person who will be reading this document now and in the future. Indicative:

Each diagram photo was enlarged with special tools so that it does not lose resolution during embedding. The document itself was set to too high an image quality which eventually started to become too heavy for us to process comfortable. At every point of the page there are links (or rather bookmarks) so that one can go directly to the contents of the document and also browse specifically the verbal descriptions. There is a common theme in the structure of the document etc.

Finally, we could not omit that the team was in the mood, 5 days before the examination of the course, to implement from the beginning a graphical interface for the system. This process was (and still is) incredibly time consuming, as none of us had this level of prior Java experience.

However, this was optional and the graphical interface is not why we claim a high score. In the end, our effort may seem to have been consumed for no reason, and we have put out our eyes unjustly, but for us this work is the best we have done.

Contribution





APPENDIX C

Bibliography

Bibliography

ALAN DENNIS, BH (2010). *Analysis and Design of Systems with UML 2.0: A object oriented approach*. KLEIDARITHMOS LTD.

FOWLER, M. (2006). *Introduction to UML: A Concise Guide to the Standard Object Modeling Language*. KLEIDARITHMOS PUBLICATIONS LTD.

javatpoint.com. (e.g.). Retrieved from Java Polymorphism: <https://www.javatpoint.com/java-tutorials>

Spurrier Gary, TH (2022). *Systems Analysis and Design-In an Age of Multiple Choices*. BROKEN HILL PUBLISHERS LTD.

Team, M. (2012, March 23). *MyMG*. Retrieved from Feasibility Study Template: How to Assess Project Viability: <https://mymanagementguide.com/feasibility-study-template/>

Visual Paradigm. (2014). Retrieved from What is Sequence Diagram?: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/?authuser=0>

w3schools.com. (e.g.). Retrieved from Java Tutorials: <https://www.w3schools.com/java/>