

Software Development Methods

School Year 2019/2020

1st Phase Report

**Magellan’s Race – 500 years**

**Alexandre Correia** (53298)

**Hugo Rodrigues** (53309)

**Ricardo Ferreira** (52915)

**Vasco Martins** (53299)

Lab P4 | João Araújo

23/10/2019

Table of Contents

[Assumptions](#_30j0zll) **2**

[Use Case Diagram](#_1fob9te) **3**

[Use Case Specifications](#_3znysh7) **4**

[Register Team](#_2et92p0) 4

[Use Case Specification](#_tyjcwt) 4

[Alternative Flows Specifications](#_1t3h5sf) 5

[Form Crew](#_4d34og8) 6

[Use Case Specification](#_2s8eyo1) 6

[Alternative Flows Specifications](#_17dp8vu) 7

[High-Sea Technical Reparations](#_3rdcrjn) 8

[Use Case Specification](#_26in1rg) 8

[Alternative Flows Specifications](#_lnxbz9) 9

[Refill Provisions](#_35nkun2) 10

[Use Case Specification](#_1ksv4uv) 10

[Alternative Flows Specifications](#_44sinio) 11

[Send Message to Family](#_2jxsxqh) 12

[Use Case Specification](#_z337ya) 12

[Alternative Flows Specifications](#_3j2qqm3) 13

[Activity Diagrams](#_1y810tw) **14**

[Register Team](#_4i7ojhp) 14

[Form Crew](#_2xcytpi) 15

[High-Sea Technical Reparations](#_1ci93xb) 16

[Refill Provisions](#_4gbbciumj4xp) 17

[Send Message to Family](#_332ftaiwzy45) 18

[Class Diagram](#_qsh70q) **19**

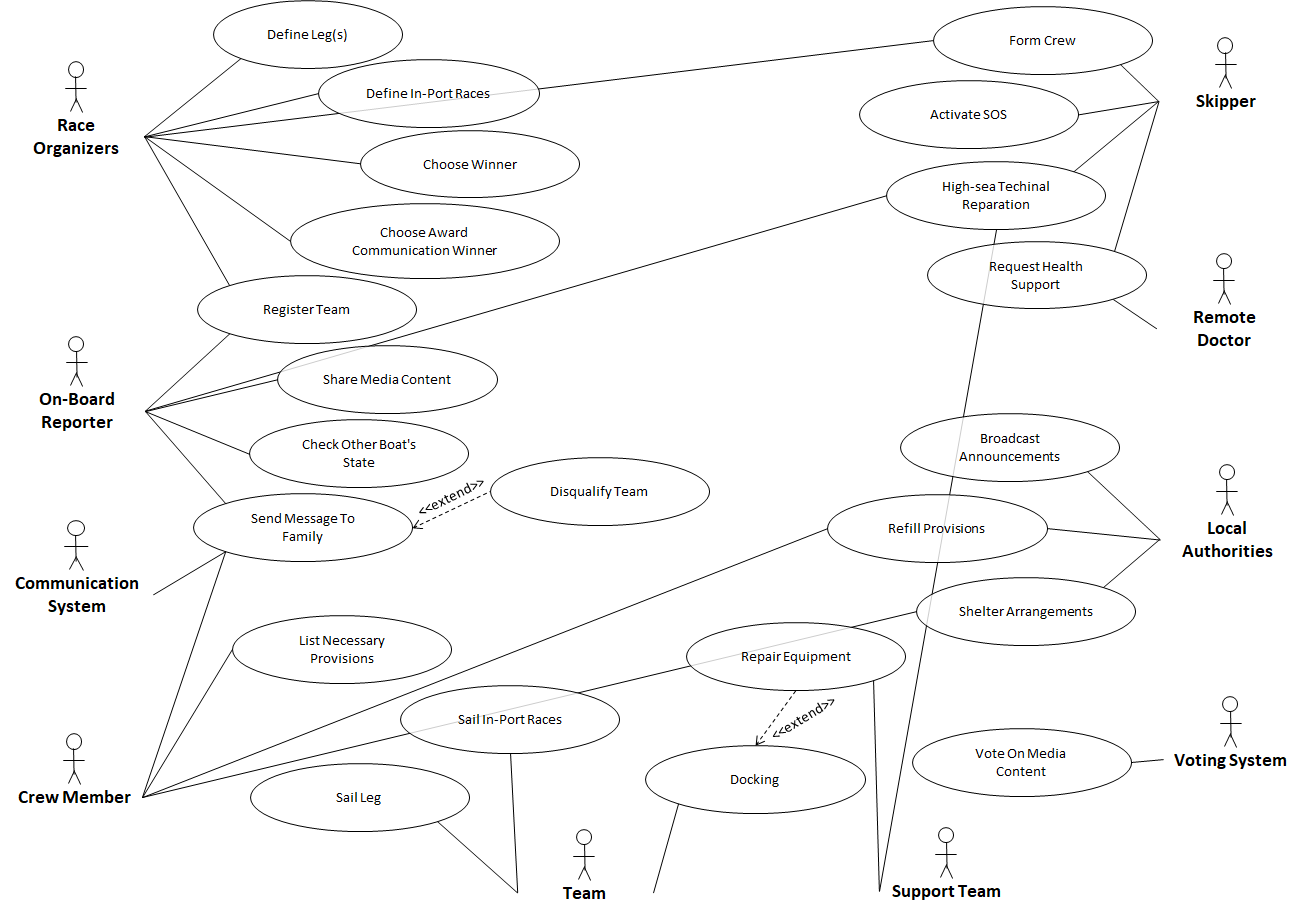
# Assumptions

We assume that the Race Organizer “manages” a team, this means, the Race Organizer is responsible for register and disqualify a team. When the Race Organizer wants to register a new team to the race, it asks the Skipper for every information about the boat and the Support Team. About the disqualifying process, we assume that the only thing that the OBR does is to report a violation committed by the team to the Race Organizer, being the Race Organizer the one that disqualifies the team.

It was assumed that all the communications are made through a communication system and that the viewers’ vote on the media content, which are shared by the On-Board Reporters, are managed by a voting system.

When docked, the teams can buy provisions to the local authorities, and for that purpose, the Skipper assigns one crew member.

# Use Case Diagram



# Use Case Specifications

## Register Team

### Use Case Specification

**Name:** Register Team

**Id:** 1

**Description:** Register a team in the system

**Actors:**

* Main: Race Organizer, Skipper
* Secondary: *None*

**Pre-conditions:** *None*

**Main-Flow:**

1. The UC starts when the Race Organizer decides to register a team.
2. The Race Organizer inserts the team’s name.
3. The system validates the team’s name.
4. The Race Organizer associates an On-Board Reporter to the team.
5. The system validates the On-Board Reporter.
6. The Race Organizer associates the Skipper to the team.
7. The system validates the Skipper.
8. The Skipper submits the ship information.
9. The system validates the ship.
10. The Skipper submits a list of the Support Team members.
11. The Race Organizer receives the list of the Support Team members.
12. The Race Organizer associates the Support Team to the team.
13. The system validates the Support Team.
14. The system sends a notification saying “Team registered”.
15. The UC ends.

**Alternative Flows:**

* Name associated to another team.
* OBR associated to another team.
* Support team associated to another team.
* Ship does not meet the requirements.

**Post-Conditions:** Team registered in the system.

### Alternative Flows Specifications

**Name:** OBR associated to another team  
**Id:** 1.1  
**Description:** Describes the steps when an OBR is associated to another team  
**Actors:**

* Main: Race Organizer
* Secondary: OBR

**Pre-conditions:** *None***Main-Flow:**

1. The UC starts when the system does not validate the OBR.
2. The system notifies the Race Organizer that the OBR is associated to another team.
3. The system provides the available OBRs.
4. The Race Organizer associates an available OBR to the team.
5. The system validates the OBR.
6. Returns to step 6 of the “Register team” Main-flow.

**Alternative Flows:** None  
**Post-Conditions:** None

**Name:** Ship does not meet the requirements  
**Id:** 1.2  
**Description:** Describes the steps when a ship does not meet the race requirements  
**Actors:**

* Main: Race Organizer
* Secondary: Skipper

**Pre-conditions:** *None***Main-Flow:**

1. The UC starts when the system does not validate the ship.
2. The system notifies the Race Organizer that the ship does not meet the race requirements.
3. The Race Organizer requests that the Skipper chooses another boat.
4. The Skipper submits the information about the new boat.
5. The Race Organizer register the new boat.
6. The system validates the new boat.
7. Returns to step 11 of the “Register team” Main-flow.

**Alternative Flows:** None  
**Post-Conditions:** None

## Form Crew

### Use Case Specification

**Name:** Form Crew

**Id:** 2

**Description:** Updates a team’s crew for next leg

**Actors:**

* Main: Skipper
* Secondary: Race Organizer

**Pre-conditions:**

* Team already registered
* Boat docked

**Main-Flow:**

1. The UC starts when the Skipper decides to change the crew before a leg or an In-Port Race.
2. The Skipper submits the male/female ratio.
3. The Race Organizer receives the new male/female ratio.
4. The Race Organizer inserts the new male/female ratio.
5. The system validates the ratio.
6. The Skipper submits the information about each crew member.
7. The Race Organizer receives the crew members’ information.
8. The Race Organizer inserts the crew members’ information.
9. The system validates the new crew members.
10. The UC ends.

**Alternative Flows:**

* Male/Female ratio is not valid.
* Crew composition doesn’t match male/female ratio.

**Post-Conditions:**  Crew has been updated in the system.

### Alternative Flows Specifications

**Name:** Male/Female ratio not valid  
**Id:** 2.1  
**Description:** Describes the steps when the ratio is not approved  
**Actors:**

* Main: Race Organizer
* Secondary: Skipper

**Pre-conditions:** *None***Main-Flow:**

1. The UC starts when the ratio is not approved
2. The system notifies the Race Organizer that the ratio is not valid.
3. The system provides the acceptable ratios.
4. The Race Organizer informs the Skipper about the possible ratios.
5. The Skipper submits a new and acceptable ratio.
6. The Race Organizer register the new ratio.
7. The system validates the new ratio.
8. Returns to step 6 of the “Form Crew” Main-Flow.

**Alternative Flows:** None  
**Post-Conditions:** Male/Female ratio approved

**Name:** Crew composition doesn’t match male/female ratio  
**Id:** 2.2  
**Description:** Describes the steps when the crew composition does not match the male/female ratio  
**Actors:**

* Main: Race Organizer
* Secondary: Skipper

**Pre-conditions:** *None***Main-Flow:**

1. The UC starts when the crew composition does not match the male/female ratio
2. The system notifies the Race Organizer that crew composition does not match the male/female ratio.
3. The Race Organizer informs the Skipper that the crew composition is not valid.
4. The Race Organizer requests a new composition.
5. The Skipper submits a new list of members.
6. The Race Organizer registers the new members.
7. The system validates.
8. The UC ends.

**Alternative Flows:** None  
**Post-Conditions:** Crew composition approved

## High-Sea Technical Reparations

### Use Case Specification

**Name:** High-sea technical reparations

**Id:** 3

**Description:** Describes the steps of a high-sea technical reparation

**Actors:**

* Main: Skipper
* Secondary: Support Team, OBR

**Pre-conditions:** The boat is not docked

**Main-Flow:**

1. The UC starts when the Skipper requests technical support or when the support team detects a problem.
2. The system establishes a connection between the Skipper and the Support Team.
3. The Skipper identifies the problem.
4. The Skipper asks for technical advice.
5. The Support Team gives technical advice.
6. The Skipper fixes the problem.
7. The Skipper confirms that the problem has been fixed.
8. The system closes the connection.
9. The system notifies the OBR with the sent messages.
10. The UC ends.

**Alternative Flows:**

* Team is disqualified.
* Problem has not been fixed.

**Post-Conditions:** Problem fixed

### Alternative Flows Specifications

**Name:** Team is disqualified  
**Id:** 3.1  
**Description:** Describes the steps when the OBR decides to disqualify a team  
**Actors:**

* Main: OBR
* Secondary: Team, Race Organizer

**Pre-conditions:** Violation committed  
**Main-Flow:**

1. The UC starts when the OBR decides to disqualify a team.
2. OBR detects a violation in a message exchanged.
3. The OBR informs the Race Organizers about the violation.
4. The Race Organizer informs the team that they are disqualified.
5. The Race Organizer disqualifies the team.
6. The UC ends.

**Alternative Flows:** *None***Post-Conditions:** Team disqualified

**Name:** Problem has not been fixed  
**Id:** 3.2  
**Description:** Describes the steps when boat problem has not been fixed  
**Actors:**

* Main: Skipper
* Secondary: Support Team

**Pre-conditions:** Malfunction detected  
**Main-Flow:**

1. The UC starts when the boat problem has not been fixed.
2. The Skipper informs the Support Team that the problem has not been fixed.
3. The Support Team gives advice on how to fix the problem.
4. The Skipper fixes the problem.
5. Returns to step 7 of the “High-sea technical reparations” Main-flow.

**Alternative Flows:** None  
**Post-Conditions:** None

## Refill Provisions

### Use Case Specification

**Name:** Refill provisions

**Id:** 4

**Description:** Describes the steps for the team to buy provisions from the local authorities.

**Actors:**

* Main: Crew Member
* Secondary: Local Authorities, Race Organizer

**Pre-conditions:** The boat is docked.

**Main-Flow:**

1. The UC starts when the crew wants to buy provisions from the Local Authorities.
2. The Skipper assigns a crew member to buy the provisions.
3. The assigned crew member gives the list of provisions needed to the local authorities.
4. The local authorities accept the list.
5. The local authorities confirm that they have those provisions in stock.
6. The assigned crew members buys the provisions from the local Authorities.
7. The Local Authorities submit the products bought by the Crew Member.
8. The system notifies the Race Organizer about the purchase.
9. The UC ends.

**Alternative Flows:**

* Products are not available
* Fresh food is not allowed

**Post-Conditions:** Provisions have been sold

### Alternative Flows Specifications

**Name:** Products are not available  
**Id:** 4.1  
**Description:** Describes the steps when the local authorities do not have the requested products  
**Actors:**

* Main: Local Authorities
* Secondary: Crew Member

**Pre-conditions:** Products have been requested  
**Main-Flow:**

1. The UC starts when the Crew Member requested some products and the Local Authorities does not have those products in stock.
2. The Local Authorities inform that the products that are not available.
3. The Local Authorities ask the Crew Member if he wants the other requested products.
4. The Crew Member confirms.
5. The assigned crew members buys the rest of the provisions from the local Authorities.
6. Returns to step 6 of the “Refill provisions” Main-flow.

**Alternative Flows:** *None***Post-Conditions:** *None*

**Name:** Fresh food is not allowed  
**Id:** 4.2  
**Description:** Describes the steps when fresh food is not allowed  
**Actors:**

* Main: Local Authorities
* Secondary: Crew Member

**Pre-conditions:** Fresh food have been requested  
**Main-Flow:**

1. The UC starts when the Crew Member asks for fresh food.
2. The Local Authorities indicate that fresh food is not allowed on board.
3. The Local Authorities ask if the Crew Member wants the other requested products.
4. The Crew Member confirms.
5. The assigned crew members buys the rest of the provisions from the local Authorities.
6. Returns to step 6 of the “Refill provisions” Main-flow.

**Alternative Flows:** None  
**Post-Conditions:** None

## Send Message to Family

### Use Case Specification

**Name:** Send message to family

**Id:** 5

**Description:** Describes the steps when a crew member wants to send a message to his/her family.

**Actors:**

* Main: Crew Member
* Secondary: OBR, Communication System

**Pre-conditions:** *None*

**Main-Flow:**

1. The UC starts when the crew member wants to make a call to a family member.
2. The Crew Member requests to OBR to send a message to his/her family.
3. The OBR approves the request.
4. The system establishes a connection between the Crew Member and the Communication System.
5. The Crew Member send messages through the Communication System.
6. The Communication System notifies the Crew Member about a response
7. The Crew Member ends the communication.
8. The system closes the connection.
9. The system notifies the OBR about the messages exchange.
10. The UC ends.

**Alternative Flows:**

* Team is disqualified.
* OBR doesn't approve crew member to sending a message to a family member.

**Post-Conditions:** Crew member sent a message to his/her family

### Alternative Flows Specifications

**Note:** The alternative flow “Team is disqualified” in the use case “Send message to family” is described by the “Team is disqualified” use case, which id is 3.1.

**Name:** OBR doesn't approve crew member to sending a message to a family member  
**Id:** 5.1  
**Description:** Describes the steps when the OBR does not approve crew member to send a message to a family member  
**Actors:**

* Main: OBR
* Secondary: Crew Member

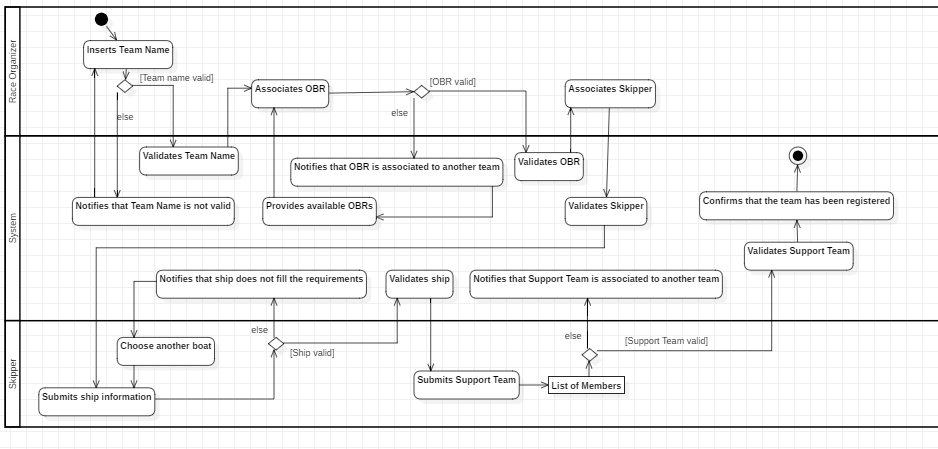
**Pre-conditions:** Crew member requested to send a message to a family member  
**Main-Flow:**

1. The UC starts when the OBR does not approve to send a message.
2. The OBR informs the crew member that is not possible to send the message.
3. The OBR indicates a prevision of the time that the crew member can send the message.
4. The UC ends.

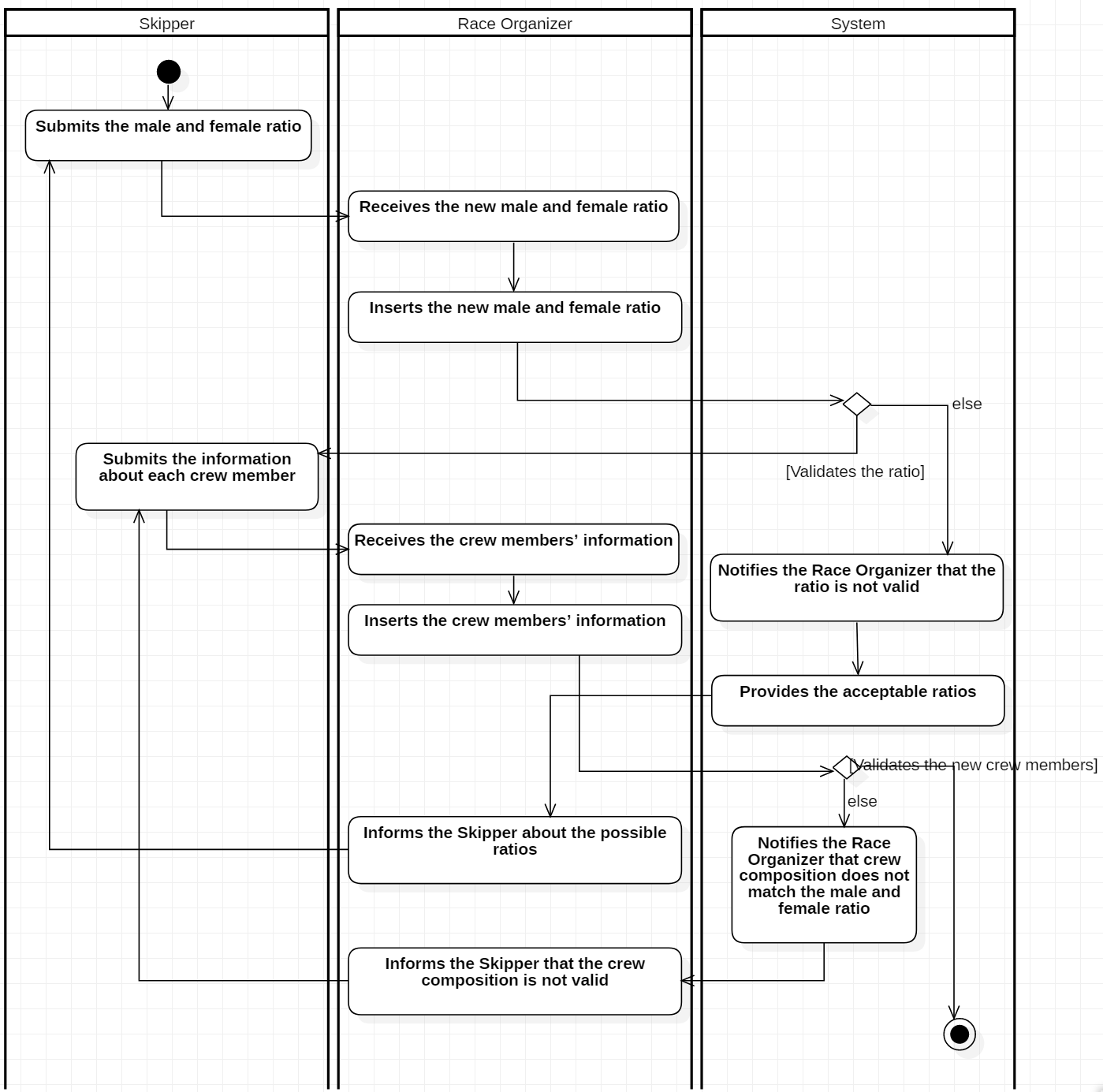
**Alternative Flows:** *None*  
**Post-Conditions:** *None*

# Activity Diagrams

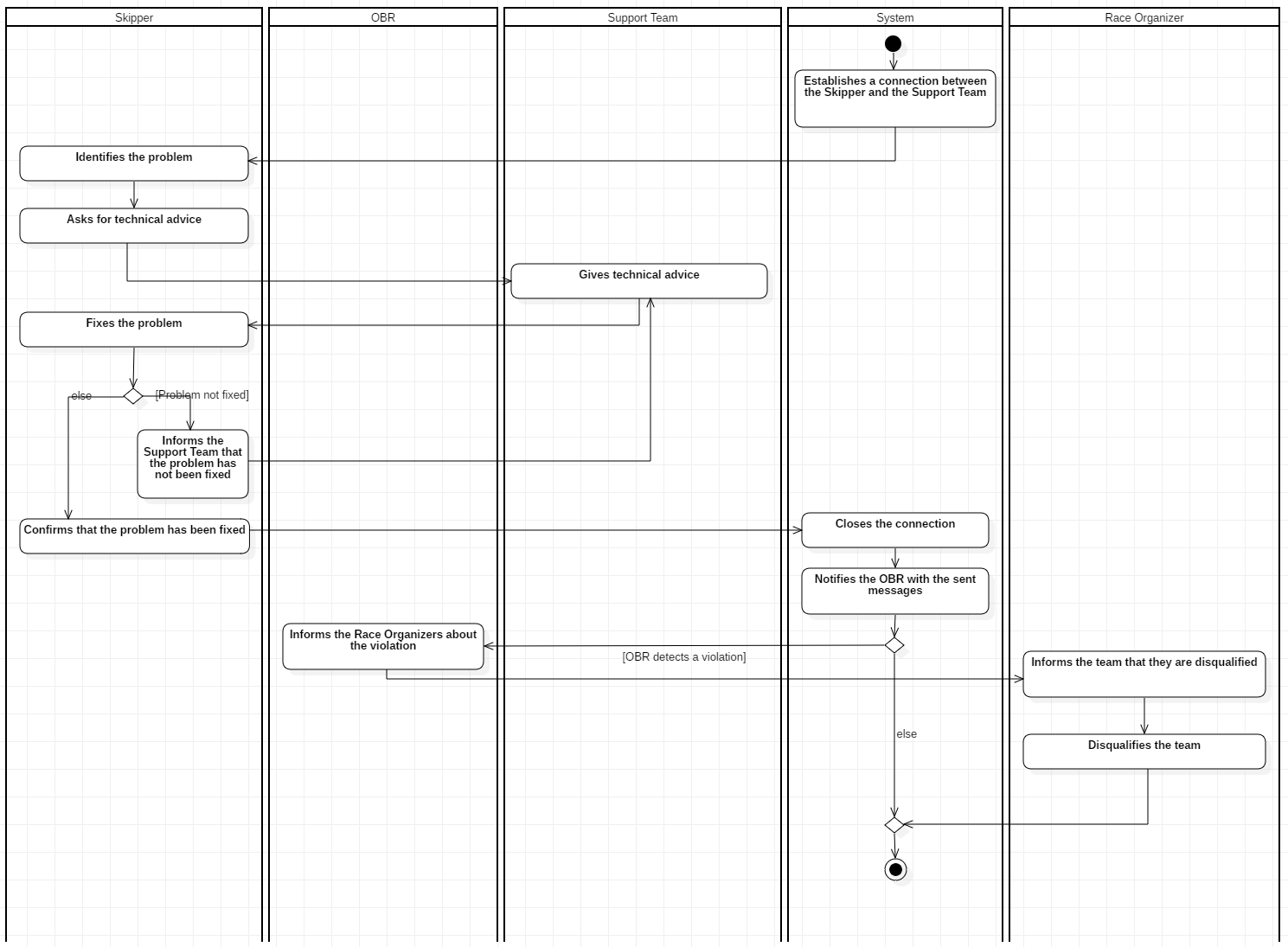
## Register Team



## Form Crew

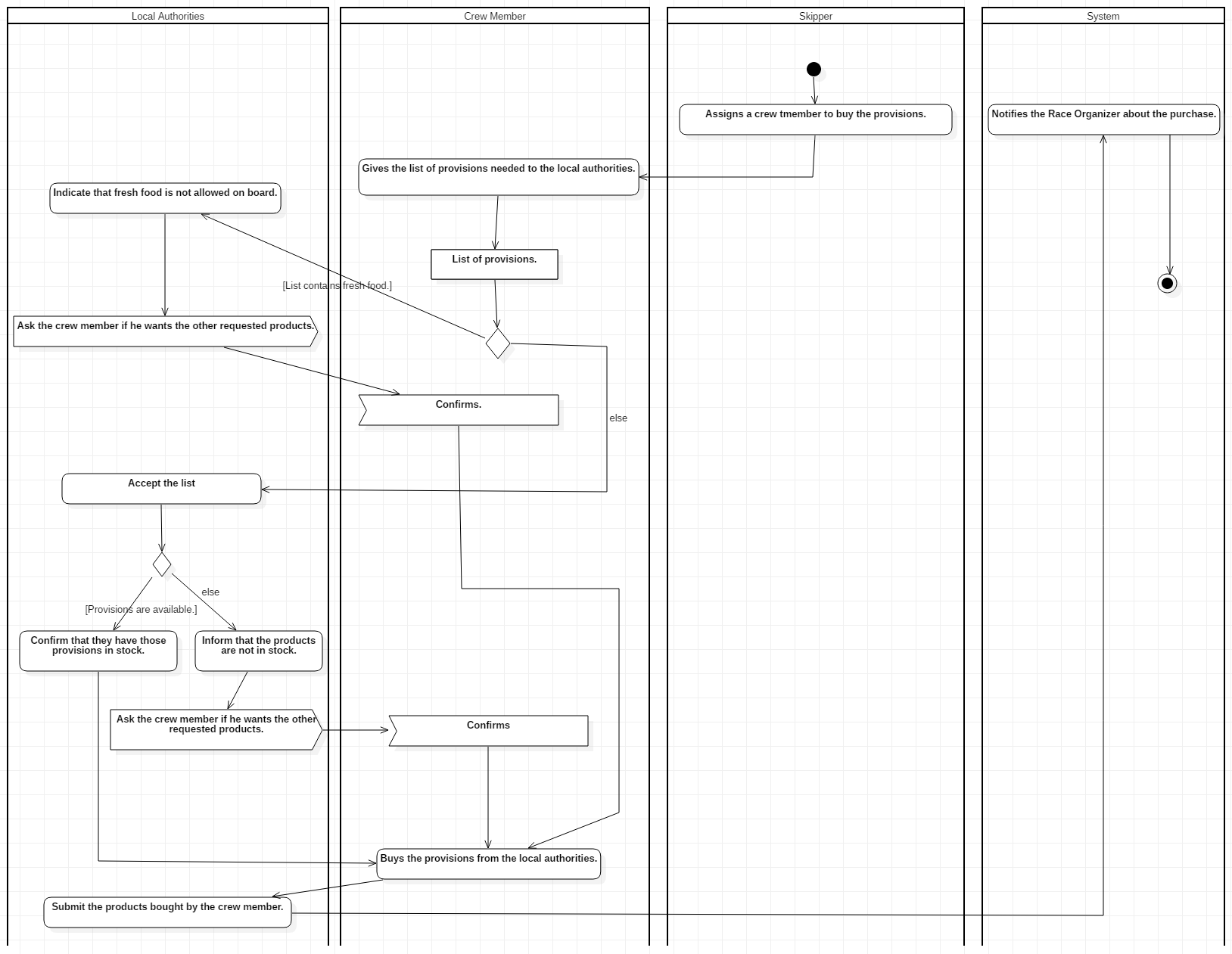


## High-Sea Technical Reparations



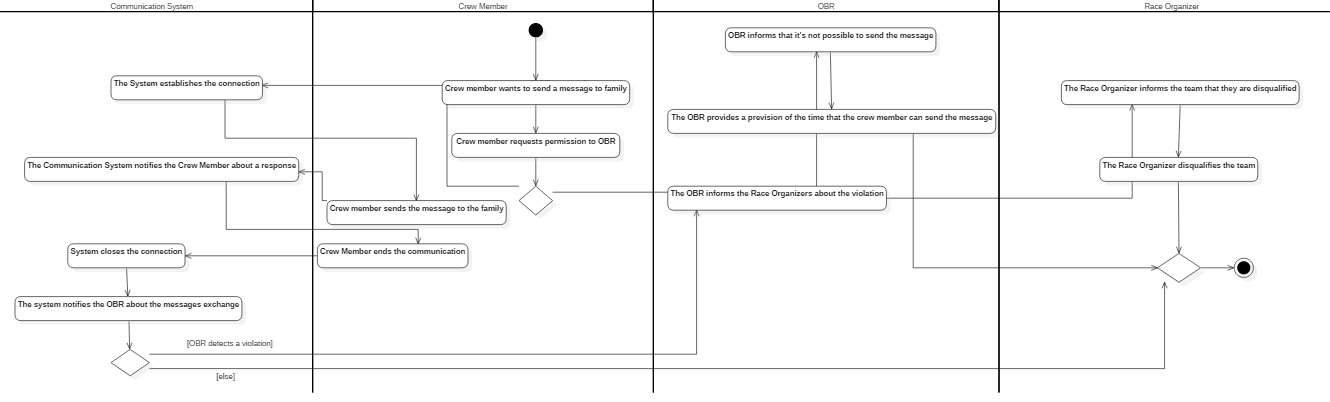
## 

## Refill Provisions



## 

## Send Message to Family



# Class Diagram

