
Software Requirements Specification

for

CRUISE SHIP MANAGEMENT SYSTEM (CSMS)

Version 1.0

**Prepared by Asad Mahmood
M.M. Arshad
Rehan Jahangir
Talha Ashar**

Lahore University of Management Sciences University

4/10/2020

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose.....	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Project Scope.....	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Features	2
2.3 User Classes and Characteristics	3
2.4 Operating Environment	3
2.5 Design and Implementation Constraints.....	3
2.6 User Documentation.....	3
2.7 Assumptions and Dependencies	3
3. Use Cases and System Features	3
3.1 Use Case List	3
3.2 Use Case Definitions.....	4
4. External Interface Requirements.....	20
4.1 User Interfaces	20
4.2 Hardware Interfaces	20
4.3 Software Interfaces.....	20
4.4 Communications Interfaces	20
5. Other Nonfunctional Requirements	21
5.1 Performance Requirements	21
5.2 Safety Requirements	21
5.3 Security Requirements	21
5.4 Software Quality Attributes.....	21
6. Other Requirements.....	21

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Cruise Ship Management System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

1.2 Document Conventions

No document conventions are being used at this time.

1.3 Intended Audience and Reading Suggestions

This document is intended for the developers of the system and for the staff of the course CS-340, Databases, for its approval.

1.4 Project Scope

The CSMS, Cruise Ship Management System, is a database application that will allow both its users and its administrators a complete mapping of the cruise ship's environment, internal workings, and the consumers that will be using the system for various self-serving needs. The CSMS will allow a robust and easy-to-use user interface for all participants involved to streamline the management and viewership of data. The data in the CSMS will not just be restricted to the cruise ship itself; the data will also be accessible to offshore management, regarding crewmates, and also travel agencies for passenger booking purposes. Of course, this non-ship access will be moderated by the database administrators in tandem with the cruise ship's highest ranking officers.

2. Overall Description

2.1 Product Perspective

The CSMS is a new, self-contained system that provides functionalities for cruise ship services consumption, operations, and logistics. The context diagram in *Figure 1* illustrates the external entities and system interfaces.

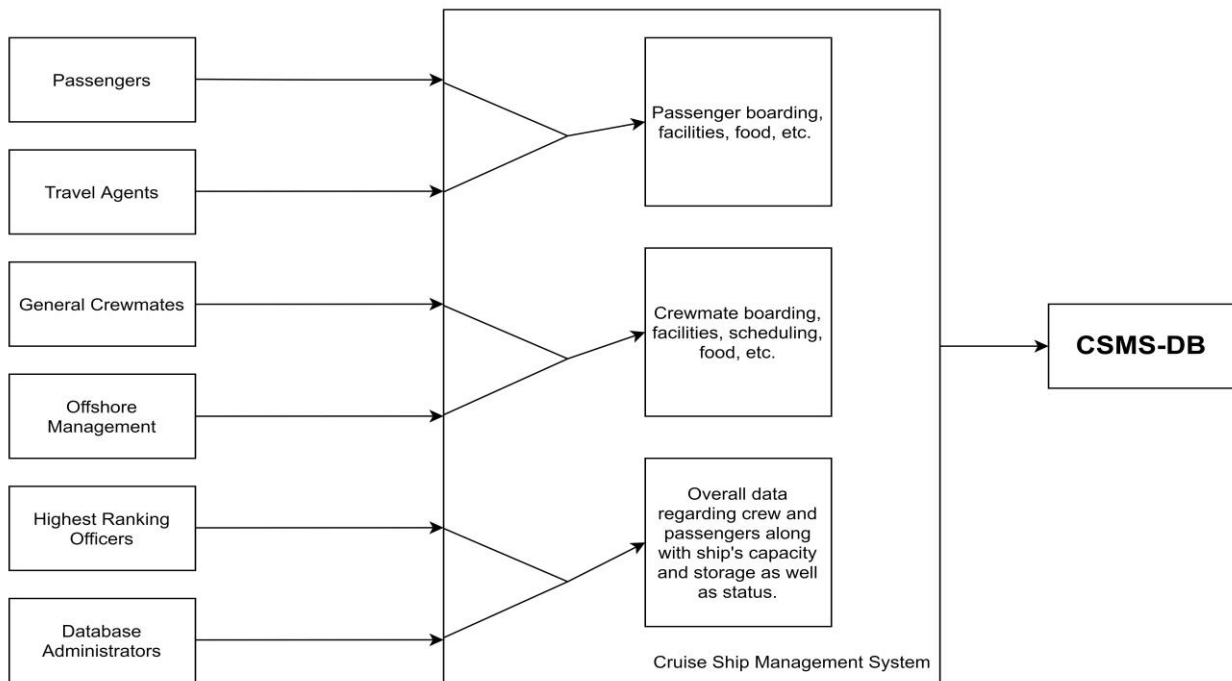


Figure 1: Entity-Interface Interaction Diagram.

2.2 Product Features

The major features of the CSMS are to make the viewership and management of necessary information easier and highly presentable to all users involved so that decision-making and planning goes smoothly for all concerned parties.

To follow the hierarchy set by *Figure 1*, the major features of the CSMS are as follows:

1. Booking and Boarding Management.
2. Facilities Availability and Scheduling.
3. Passenger Report Log.
4. Complaint Portal.
5. Staff Recruitment.
6. Food and Beverage Management.
7. Supplies and Inventory Management.
8. Promotional Campaigning Services.
9. Routing.
10. Journey Logging.

2.3 User Classes and Characteristics

Passengers (favored)	Passengers are the customers of the Cruise Ship Services. They use the CSMS to make and cancel bookings for a cruise, view schedule activities while on the ship, and lodge complaints using the complaints portal.
Travel Agents	Travel Agents are external companies that use the CSMS to offer cruise bookings and reservations, and provide information on ship facilities to prospective clients.
General Crewmates (favored)	General Crewmates refer to the staff employed for the cruise journey that has greater access and viewership to the database. Any and all managerial tasks are performed by them such as inventory management, supply management, and general logistics.
Offshore Management	Offshore Management may use the CSMS to access all the data related to potential personal which helps them facilitate crewmate recruitment, as well as performing promotional campaigns for upcoming cruises.
Highest Ranking Officers (favored)	These officers are personnel responsible for the supervision of the journey and will have access to all the data. They can use the CSMS for route management and control ship operations.
Database Administrators	Administrators are people that have all user privileges and can view, edit, and add data as per their discretion and handle data security.

Table 1: Classification of Users and External Parties

2.4 Operating Environment

The software will be operable in both Linux and Windows environments.

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

There is no user documentation as of this moment.

2.7 Assumptions and Dependencies

There are no such assumptions as of this moment.

3. Use Cases and System Features

3.1 Use Case List

Primary Actor	Use Cases
Travel Agents	Booking and Reservations Cancellation of Bookings
Passengers	Booking and Reservations Cancellation of Bookings Upgrading and Changing of Rooms Generating Complaint Tickets Complaint Tickets Status and Termination Menu Scheduling Room Service Journey Mapping
Offshore Management	Upgrading and Changing of Rooms Insertion and Updating of Crewmates Crewmate Management Promotional Campaigning Complaint Tickets Status and Termination
Crewmates	Upgrading and Changing of Rooms Activity Scheduling Menu Scheduling Freight Management Updating of Inventory Room Service Journey Mapping
Highest Ranking Officers	Journey Routing Journey Mapping

Table 2: List that contains all the Use Cases along with their actors.

3.2 Use Case Definitions

Use Case ID:	1		
Use Case Name:	Booking and Reservations		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Travel Agents, Passengers
Description:	Allow the Actors to book or reserve rooms on the cruise ship through appropriate views.
Trigger:	The customer clicks a button and is lead to a view is allowing them to specify their requirements.
Preconditions:	1. Cruise ship should not be at full capacity.
Postconditions:	1. Inserting the new passenger information into the necessary relations in the database. 2. Resulting pop-up to show the transaction was completed successfully and provides login ID and password. 3. Provision of transaction receipt.
Normal Flow:	1.0 Customer books a room. 1. Customer is provided view where they can input their details through a form. 2. The customer enters the required information. 3. If data is input successfully, insert the passenger details into the database. 4. Generate receipt from provided data and other necessary database information. 5. Output the receipt for customer to view. 6. Display successful transaction status.
Alternative Flows:	1.1 Customer input fails validation and duplication check (step 2). 1. Prompt the user to re-enter data. 2. Return to step 3 of normal flow.
Exceptions:	1.E.1 The form is partially filled or not submitted. 1. No changes are made to the database. 2. Terminate the use case after a specified amount of time.
Includes:	No other use cases included.
Priority:	High.
Frequency of Use:	High.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	2		
Use Case Name:	Cancellation of Bookings		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Travel Agents, Passengers
Description:	Allow the Actors to cancel any bookings or reserved rooms on the cruise ship through appropriate views.
Trigger:	The customer uses a button on their default view to cancel a booking.
Preconditions:	<ol style="list-style-type: none"> 1. The concerned room should exist in the database. 2. The concerned room should be allotted to the concerned customer.
Postconditions:	<ol style="list-style-type: none"> 1. Archiving the passenger information to the necessary relations in the database. 2. Resulting pop-up to show the transaction was completed successfully.
Normal Flow:	<p>2.0 Customer wishes to cancel booking.</p> <ol style="list-style-type: none"> 1. Customer logs in with provided ID and password. 2. They are provided a view showing status of booking. 3. A prompt or button would take them to the cancellation window. 4. The customer confirms cancellation. 5. Upon cancellation confirmation, check for refund eligibility. 6. Generate pop-up showing transaction status and any refund details. 4. Generate receipt from provided data and other necessary database information. 5. Upon closing pop-up, log out the user and rescind their login details.
Alternative Flows:	<p>2.1 Customer input fails validation and duplication check (step 2).</p> <ol style="list-style-type: none"> 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow.
Exceptions:	None.
Includes:	No other use cases included.
Priority:	High.
Frequency of Use:	Medium.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	3		
Use Case Name:	Upgrading and Changing of Rooms		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Passengers, Offshore Management, Crewmates
Description:	Allow the Actors to upgrade booking packages or swap rooms on the cruise ship through appropriate views.
Trigger:	An actor uses a button on their default view to upgrade or swap their
Preconditions:	<ol style="list-style-type: none"> 1. The concerned rooms should exist in the database. 2. The concerned rooms should be allotted to the concerned customer.
Postconditions:	<ol style="list-style-type: none"> 1. Updating the passenger information to the necessary relations in the database. 2. Resulting pop-up to show the transaction was completed successfully.
Normal Flow:	<p>3.0 Passenger wishes to upgrade or swap room.</p> <ol style="list-style-type: none"> 1. User logs in with provided ID and password. 2. They are provided a view showing status of booking. 3. A prompt or button would take them to the upgrade/swapping window. 4. The passenger inputs the desired room details. 5. The query with these details is sent to the Offshore Management. 6. Offshore Management accepts query. 7. Generate a pop-up about query acceptance. 8. After specified amount of time, Offshore Management makes required changes to database.
Alternative Flows:	<p>3.1 Invalid login details (step 1).</p> <ol style="list-style-type: none"> 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow. <p>3.2 Crewmate wishes to swap room (step 1).</p> <ol style="list-style-type: none"> 1. Crewmate logs in with work ID and password. 2. They are provided a view showing status of lodgings. 3. A prompt or button would take them to the swapping window. 4. Return to step 4 of normal flow. <p>3.3 Offshore Management reject query (step 6).</p> <ol style="list-style-type: none"> 1. Generate pop-up about query rejection.
Exceptions:	<p>3.E.1 The form is partially filled or not submitted.</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case after a specified amount of time.
Includes:	No other use cases included.
Priority:	High.
Frequency of Use:	Low.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	4		
Use Case Name:	Activity Scheduling		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Crewmates
Description:	Allow the Actors to view facilities on the cruise ship and schedule them through appropriate views.
Trigger:	When an actor wishes to schedule activities and facilities on the cruise ship, a view is provided allowing them to specify their requirements.
Preconditions:	<ol style="list-style-type: none"> 1. The activities should exist in the database. 2. Facilities should not be available for activities during downtimes.
Postconditions:	<ol style="list-style-type: none"> 1. Facility timings and flags are updated in database.
Normal Flow:	<p>4.0 Crewmate schedules facilities for activities.</p> <ol style="list-style-type: none"> 1. User logs in with provided ID and password. 2. They are provided a view showing the facilities relevant to their departments. 3. User can then change timings for the facilities as required as long as they do not clash. 4. Generate a pop-up about facility timing change.
Alternative Flows:	<p>4.1 Invalid login details (step 1).</p> <ol style="list-style-type: none"> 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow. <p>4.2 Clash timings (step 3).</p> <ol style="list-style-type: none"> 1. In case of two activity times clashing, prompt the user to change the timings for the activity. 2. Return to step 4 of normal flow.
Exceptions:	None.
Includes:	No other use cases included.
Priority:	High.
Frequency of Use:	High.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	5		
Use Case Name:	Promotional Campaigns		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Offshore Management
Description:	Allow Offshore Management to select and send promotional messages about upcoming cruises to former passengers through appropriate views.
Trigger:	When the actor wishes to send out a promotional message, they are given a view to select appropriate customers.
Preconditions:	1. There should be some future cruise planned.
Postconditions:	1. There will no change to the database 2. Relevant customers receive a promotional message from the Cruise Company.
Normal Flow:	5.0 Offshore Management sends promotional adverts. 1. The user will login with their work ID. 2. They are provided a view that shows all former passengers that have consented to receiving promotional content. 3. The user will be prompted to input the promotional message. 4. The system will ask the user to confirm whether they wish to send out the message. 5. Upon confirmation, the message will be sent.
Alternative Flows:	5.1 Invalid login details (step 1). 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow. 5.2 No customers having subscribed to promotional content (step 3). 1. Terminate use case.
Exceptions:	None.
Includes:	No other use cases included.
Priority:	Medium.
Frequency of Use:	Low.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	6		
Use Case Name:	Generating Complaint Tickets		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Passengers
Description:	The Passengers will be provided an appropriate view with a form where they can submit their concerns to the appropriate authorities (Offshore
Trigger:	A button visible to the Passengers on their login view which they can use to go to the complaint view.
Preconditions:	None.
Postconditions:	1. The details entered to the complaint form will be forwarded to the Offshore Management through an email.
Normal Flow:	6.0 Passenger lodges a complaint. <ol style="list-style-type: none"> 1. Passenger logs in with given ID and password. 2. They are shown a complaint button on the default view. 3. Clicking on the button forwards them to the complaint portal view. 4. The complaint is typed into the provided comment box. 5. The complaint is submitted and forwarded to the Offshore Management.
Alternative Flows:	6.1 Invalid login details (step 1). <ol style="list-style-type: none"> 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow. 6.2 Customer does not wish to go through with complaint (step 4) <ol style="list-style-type: none"> 1. User is returned to the default view.
Exceptions:	6.E.1 The form is partially filled or not submitted. <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case after a specified amount of time.
Includes:	No other use cases included.
Priority:	High.
Frequency of Use:	Medium.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	7		
Use Case Name:	Complaint Ticket Status and Termination		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Passengers, Offshore Management
Description:	The Offshore Management and Passengers can view the status of any lodged complaints. The Offshore Management can also decide to
Trigger:	Button which takes you to a view which shows all active and inactive complaint tickets.
Preconditions:	1. There should be some existing complaint.
Postconditions:	1. In case of ticket termination, passengers are notified. 2. Following ticket termination, the appropriate relations are updated to set current ticket status to terminated.
Normal Flow:	7.0 Passenger checks complaint ticket status. 1. Passenger logs in with given ID and password. 2. They are shown a complaint button on the default view. 3. Clicking on the button forwards them to the complaint portal view. 4. They are shown their history of lodged complaints. 5. Upon selection of complaint, they are shown that particular ticket's status.
Alternative Flows:	7.1 Invalid login details (step 1). 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow. 7.2 Offshore Management checks complaint ticket status (step 1). 1. Flow is same as Normal Flow until step 3. 2. They are shown all lodged passenger complaints. 3. Upon choosing a particular complaint, they can choose to terminate it. 4. Upon termination, the complaint ticket status is updated. 5. They are returned to the default complaint ticket view.
Exceptions:	None
Includes:	No other use cases included.
Priority:	High.
Frequency of Use:	High.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	8		
Use Case Name:	Insertion and Updating of Crewmates		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Offshore Management
Description:	When the Offshore Management can recruit crewmates for the cruise ship, they can enter their data as well as updating data for existing
Trigger:	Button on the default view.
Preconditions:	<ol style="list-style-type: none"> 1. Crewmate vacancies should exist. 2. In case of updating data, the relevant crewmate's record should exist.
Postconditions:	<ol style="list-style-type: none"> 1. New crewmate data inserted into the database, or an existing crewmate's data is updated.
Normal Flow:	<p>8.0 Offshore Management inserts new crewmate.</p> <ol style="list-style-type: none"> 1. User logs in with given ID and password. 2. They are shown a button on the default view. 3. Clicking on the button forwards them to the crewmate data view. 4. They are given a form to enter details of the new crewmate. 5. The data is committed to the database.
Alternative Flows:	<p>8.1 Invalid login details (step 1).</p> <ol style="list-style-type: none"> 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow. <p>8.2 Offshore Management updates existing crewmate data (step 3)</p> <ol style="list-style-type: none"> 1. They are shown a view of all existing crewmates. 2. Offshore Management may then select a specific crewmate and alter their personal details. 3. The data is committed to the database.
Exceptions:	<p>8.E.1 The form is partially filled or not submitted.</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case after a specified amount of time. <p>8.E.2 Offshore Management tries to update non-existent crewmate.</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case. <p>8.E.3 Offshore Management tries to insert existing crewmate.</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case.
Includes:	No other use cases included.
Priority:	High.
Frequency of Use:	High.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	9		
Use Case Name:	Crewmate Management		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Offshore Management
Description:	The Offshore Management can manage and update the crewmates' department allotments and scheduling.
Trigger:	Button on the default view.
Preconditions:	<ol style="list-style-type: none"> 1. The relevant crewmate's record should exist. 2. The relevant department data should also exist.
Postconditions:	1. Existing crewmates work data is updated in the database.
Normal Flow:	<p>9.0 Offshore Management enters crewmate work data.</p> <ol style="list-style-type: none"> 1. User logs in with given ID and password. 2. They are shown a button on the default view. 3. Clicking on the button forwards them to the crewmate data view. 4. They are shown another button which takes them to the crewmate work related data view. 5. They are given a form to enter details of the crewmate. 6. The data is committed to the database.
Alternative Flows:	<p>9.1 Invalid login details (step 1).</p> <ol style="list-style-type: none"> 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow. <p>9.2 Offshore Management updates crewmate work data (step 4).</p> <ol style="list-style-type: none"> 1. Offshore Management may then select a specific crewmate and alter their work details. 2. The data is committed to the database.
Exceptions:	<p>9.E.1 The form is partially filled or not submitted.</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case after a specified amount of time.
Includes:	Use Case 8 is included.
Priority:	High.
Frequency of Use:	High.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	10		
Use Case Name:	Menu Scheduling		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Crewmates, Passengers
Description:	Crewmates assigned to kitchen duties can decide the menu schedule for different days. Passengers and Crewmates can both view the menu.
Trigger:	Button on the default views.
Preconditions:	There are existing crewmates that have been assigned to the food and beverages department.
Postconditions:	The menu will get updated every week in advance and consequently the database will be updated.
Normal Flow:	<p>10.0 Crewmate wants to make changes to the menu.</p> <ol style="list-style-type: none"> 1. User logs in with given ID and password. 2. They are shown a button on the default view. 3. Clicking on the button forwards them to the menu view. 4. They can then select the appropriate cuisine for the menu. 5. The changes will be committed to the database.
Alternative Flows:	<p>10.1 Invalid login details (step 1).</p> <ol style="list-style-type: none"> 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow. <p>10.2 Passenger wishes to view the menu (step 4).</p> <ol style="list-style-type: none"> 1. In the passenger view, they will not have the authority to make any changes.
Exceptions:	<p>10.E.1 The form is partially filled or not submitted.</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case after a specified amount of time.
Includes:	No other use cases included.
Priority:	High.
Frequency of Use:	High.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	11		
Use Case Name:	Room Service		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Passengers, Crewmates
Description:	A passenger may request room services such as food and cleaning, and relevant resources will be allocated to perform them.
Trigger:	Button on the default views.
Preconditions:	The passenger must not already have an active room service request.
Postconditions:	The passenger's room service tab will be marked as active.
Normal Flow:	<p>11.0 The passenger makes a room service request.</p> <ol style="list-style-type: none"> 1. User logs in with given ID and password. 2. They are shown a button on the default view. 3. Clicking on the button forwards them to the room service menu. 4. They can then select the appropriate service they want. 5. A request ticket is forwarded to the relevant crewmate departments. 6. Relevant changes are committed to the database.
Alternative Flows:	<p>11.1 Invalid login details (step 1).</p> <ol style="list-style-type: none"> 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow. <p>11.2 Crewmate terminates request after execution of service (step 4).</p> <ol style="list-style-type: none"> 1. The relevant department crewmate can view room service request tickets that have been submitted for their particular service. 2. They can then terminate the requests depending on if the service was completed.
Exceptions:	<p>11.E.1 The request is not submitted</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case after a specified amount of time.
Includes:	No other use cases included.
Priority:	High.
Frequency of Use:	High.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	12		
Use Case Name:	Updating of Inventory		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Crewmates
Description:	Crewmates can view, insert, update, and delete inventory items over the course of the journey.
Trigger:	Button on the default views.
Preconditions:	If updating or deleting inventory items, those items should exist.
Postconditions:	The database will be updated with the new commits.
Normal Flow:	<p>12.0 Crewmate inserts a new item.</p> <ol style="list-style-type: none"> 1. User logs in with given ID and password. 2. They are shown a button on the default view. 3. Clicking on the button forwards them to the inventory view. 4. They are given a form to enter details of the new item. 5. The data is committed to the database.
Alternative Flows:	<p>12.1 Invalid login details (step 1).</p> <ol style="list-style-type: none"> 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow. <p>12.2 Crewmate updates an item (step 4).</p> <ol style="list-style-type: none"> 1. They are given a form to update details of the item and the changes are committed to the database. <p>12.3 Crewmate deletes an item (step 4).</p> <ol style="list-style-type: none"> 1. They are given a form to enter details of the item to be deleted and the changes are committed to the database.
Exceptions:	<p>12.E.1 The request is not submitted.</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case after a specified amount of time. <p>12.E.2 Crewmate tries to update or delete a non-existent item.</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case. <p>12.E.3 Crewmate tries to insert existing item.</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case.
Includes:	No other use cases included.
Priority:	High.
Frequency of Use:	High.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	13		
Use Case Name:	Freight Management		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Crewmates
Description:	Crewmates can view, insert, update, and delete freight over the course of the journey.
Trigger:	Button on the default views.
Preconditions:	If updating or deleting freight, those items should exist.
Postconditions:	The database will be updated with the new commits.
Normal Flow:	<p>13.0 Crewmate inserts a new item.</p> <ol style="list-style-type: none"> 1. User logs in with given ID and password. 2. They are shown a button on the default view. 3. Clicking on the button forwards them to the freight view. 4. They are given a form to enter details of the new item. 5. The data is committed to the database.
Alternative Flows:	<p>13.1 Invalid login details (step 1).</p> <ol style="list-style-type: none"> 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow. <p>13.2 Crewmate updates an item (step 4).</p> <ol style="list-style-type: none"> 1. They are given a form to update details of the item and the changes are committed to the database. <p>13.3 Crewmate deletes an item (step 4).</p> <ol style="list-style-type: none"> 1. They are given a form to enter details of the item to be deleted and the changes are committed to the database.
Exceptions:	<p>13.E.1 The request is not submitted.</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case after a specified amount of time. <p>13.E.2 Crewmate tries to update or delete a non-existent item.</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case. <p>13.E.3 Crewmate tries to insert existing item.</p> <ol style="list-style-type: none"> 1. No changes are made to the database. 2. Terminate the use case.
Includes:	Includes use case 12.
Priority:	High.
Frequency of Use:	Medium.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	14		
Use Case Name:	Journey Routing		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Highest Ranking Officer
Description:	The Highest Ranking Officers can use this feature to create an optimal journey route (based on its requirements) for the cruise ship from among a list of its designated journey locations.
Trigger:	Button on the default view.
Preconditions:	The list of locations should exist.
Postconditions:	A pop-up will be generated showing the list of locations on the optimal
Normal Flow:	14.0 Crewmate inserts a new item. 1. User logs in with given ID and password. 2. They are shown a button on the default view. 3. They select the destination. 4. The routing algorithm selects the optimal path. 5. A pop-up containing all the locations on this path is output. 6. User is redirected to default view upon closing pop-up.
Alternative Flows:	14.1 Invalid login details (step 1). 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow.
Exceptions:	14.E.1 The request is not submitted. 1. Should the destination input be the same as the current location, the use case will be terminated.
Includes:	Includes use case 15.
Priority:	High.
Frequency of Use:	Low.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

Use Case ID:	15		
Use Case Name:	Journey Mapping		
Created By:	Database Administrator	Last Updated By:	
Date Created:	6/10/2020	Date Last Updated:	

Actors:	Highest Ranking Officers, Crewmates, Passengers
Description:	The user will be able to see the current location of the cruise ship using a third-party API.
Trigger:	Button on the default views.
Preconditions:	There should be active internet connection.
Postconditions:	None.
Normal Flow:	<p>15.0 User wishes to view ship location.</p> <ol style="list-style-type: none"> 1. User logs in with given ID and password. 2. They are shown a button on the default view. 3. Every few minutes, the current location of the ship is randomly updated from the list of database locations. 4. Clicking on the button shows a pop-up that would display the ship's current location via a third-party API subject to an internet connection that allows updating of the location.
Alternative Flows:	<p>15.1 Invalid login details (step 1).</p> <ol style="list-style-type: none"> 1. Prompt the user to re-enter data. 2. Return to step 2 of normal flow.
Exceptions:	<p>15.E.1 The API is unable establish a connection.</p> <ol style="list-style-type: none"> 1. Display a pop-up with an error message. 2. Terminate the use case.
Includes:	None.
Priority:	Medium.
Frequency of Use:	Medium.
Business Rules:	None.
Special Requirements:	None.
Assumptions:	None.
Notes and Issues:	None.

4. External Interface Requirements

4.1 User Interfaces

The system will provide an easy to use web interface to users that will allow each user class to carry out their respective tasks after the authentication process. The interface will mainly involve buttons to allow users to trigger relevant tasks.

4.2 Hardware Interfaces

For the database management, there needs to be a server to respond to the queries from the user interface.

4.3 Software Interfaces

In order to facilitate the journey mapping feature, the system will need to interface with a third party API.

4.4 Communications Interfaces

The complaint portal will require email functionalities and as such communication between the server and client sides as well as the email server protocols.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

For tasks requiring user input through comment boxes or forms, there will be a timer associated with a task, and the failure of a user to submit the particular data within the allotted time frame will result in a termination of that particular task.

Queries sent from the user interface to the database are expected to be performed in no more than 15 seconds.

5.2 Safety Requirements

There will also be appropriate views presented to each user class to ensure they can only see the data they are meant to. Consequently, only authorized personnel can cause changes to be committed to the database.

5.3 Security Requirements

Users, regardless of their affiliations, are required to login to the CSMS portal in order to carry out any tasks.

5.4 Software Quality Attributes

The database shall be active for the duration of the cruise.

6. Other Requirements

No other requirement yet.

Appendix A: Glossary

CSMS	Cruise Ship Management System
------	-------------------------------

Appendix B: Analysis Models

No extra analysis models as of yet.

Appendix C: Issues List

No issues as of yet.