HOTEL PLAZA MANDATORY ASSIGNMENT
1ST SEMESTER

Students:
Gkiokas Dimitrios
Kwiatkowska Karolina
Martidis Ilias

TABLE OF CONTENTS

Requirements/Needs list:

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. Requirements analysis involves all the tasks that are conducted to identify the needs of different stakeholders. Therefore requirements analysis means to analyze, document, validate and manage software or system requirements. High-quality requirements are documented, actionable, measurable, testable, traceable, helps to identify business opportunities, and are defined to facilitate a system design. A software requirement is a functional or non-functional need to be implemented in the system. Functional means providing a particular service to the user. A software requirement can also be a non-functional, it can be a performance requirement. For
example, a non-functional requirement is where every page of the system should be visible to the users within 5 seconds.

Customer requirements:

- The system must be easy to comprehend/use
- It must do the job to the complete satisfaction of the user

Architectural requirements:

- The system must be built in Java and operate on Windows machines.

Structural requirements:

- The system must be well structured

Behavioral requirements:

- The system must perform as intended

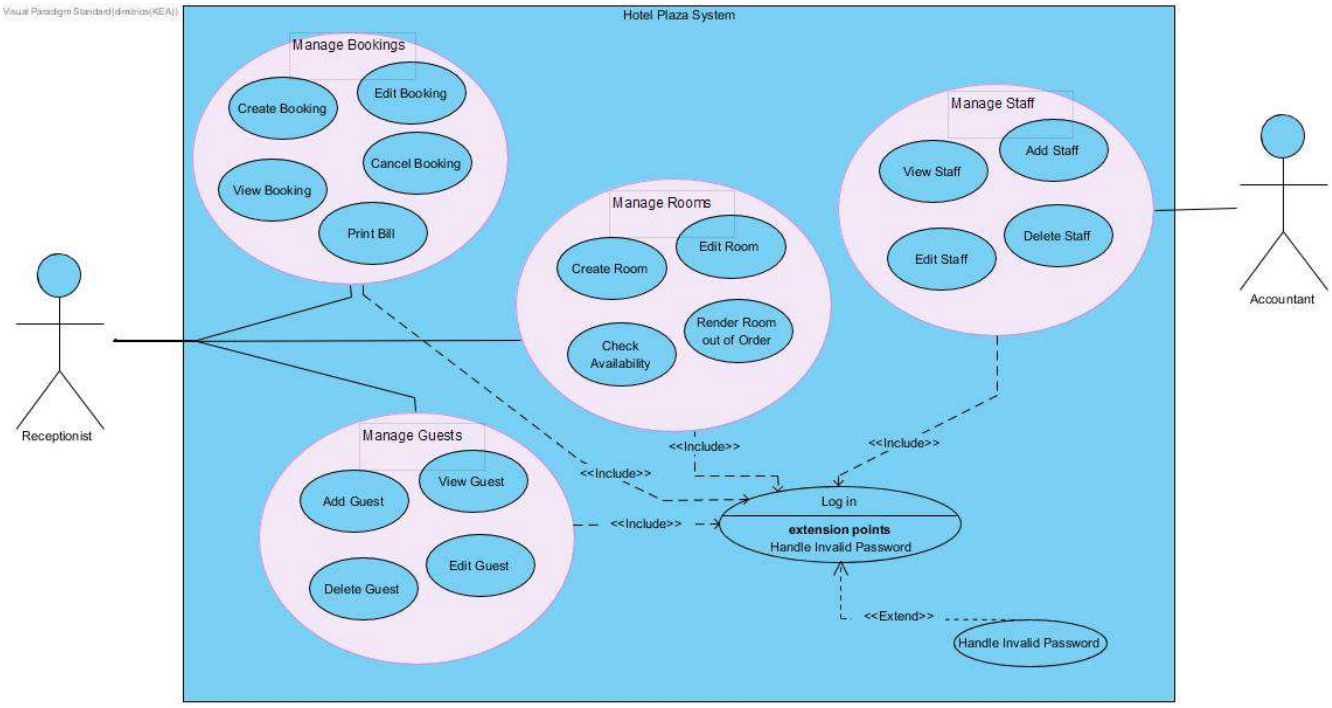
Performance requirements:

- The system's response times must be minimal so the receptionist doesn't lose time and make the guests wait.

Design requirements:

- **The system must be low maintenance and memory effective**

USE CASE DIAGRAM



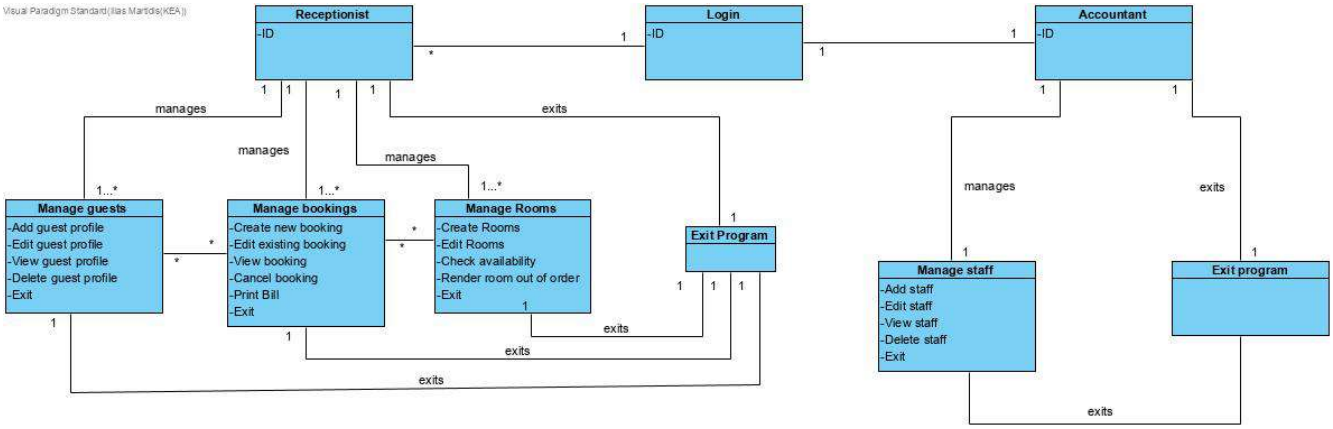
VISION OF THE HOTEL PLAZA HMS

The Hotel Plaza HMS system will increase the efficiency of the hotel processes and procedures by making the booking system quick and efficient. The satisfaction of our customer, Hotel Plaza, will lead to other projects with other hotels and keep existing customers like Hotel Plaza.

GLOSSARY (Alphabetical)

- **Accountant – The financial director of the Hotel and Human Resources Manager**
- **Availability – The number of free rooms available**
- **Booking – The creation of a new room reservation**
- **Chambermaids – The cleaning staff of the Hotel**
- **Double Rooms - Rooms with 2 beds, can accommodate up to 2 people**
- **GM – The general manager of the Plaza Hotel**
- **Guest – The customer who will stay in the hotel**
- **Out of order – A term which means that a room cannot be used temporarily**
- **Plaza Hotel – Our customer who is financing the project**
- **Receptionist – The administrator who handles the hotel reservations**
- **Single Rooms – Rooms with 1 bed, can accommodate 1 person**
- **Suites – Rooms with 4 beds, can accommodate up to 4 people**
- **Username and password - data which allows the user to log in into the system (The accountant and receptionist have different passwords)**

DOMAIN MODEL



HOTEL PLAZA: Functional/Non-Functional Requirements

Functional Requirements

1. Default functionality
    1. System shall allow to login to the system and recognize whether it's a receptionist or an accountant.
    2. System shall allow to exit program at each stage.
    3. System shall allow to "go to the main menu" option at each stage.
2. Guests profiles management
    1. The system shall record the customer's first name.
    2. The system shall record the customer's last name.
    3. The system shall record the customer's address.
    4. The system shall record the customer's phone number.
    5. The system shall create the guest's ID and profile.
    6. The system shall allow to edit guest's details.
    7. The system shall allow to delete guest's profile.
3. Rooms management
    1. The system shall record the room's ID.
    2. The system shall record the floor on which room is located. (Assign the room to the floor)
    3. The system shall record the number of beds in the room.

4. The system shall record the default room price.
5. The system shall record whether the room has internet access or not.
6. The system shall allow to edit room's data.
7. The system shall allow to delete room's data.

4. Reservation/Booking management
    1. The system shall record reservation.
    2. The system shall record the number of occupants.
    3. The system shall inform if no room is available.
    4. The system shall assign the room number to the guest's profile.
    5. The system shall display the default room rate.
    6. The system shall allow the default room rate to be changed.
    7. The system shall record the expected check-in date and time.
    8. The system shall record the expected checkout date and time.
    9. The system shall check-in customers.
    10. The system shall allow reservations to be modified without having to reenter all the customer information.
    11. The system shall allow reservations to be cancelled.
    12. The system shall checkout customers.
    13. The system shall display the amount owed by the customer.
    14. To retrieve customer information the last name or room number shall be used
    15. The system shall record that the room is empty.
    16. The system shall record the payment.
    17. The system shall print the bill.
5. Staff management
    1. The system shall record the employee's first name.
    2. The system shall record the employee's last name.
    3. The system shall record the employee's phone number.
    4. The system shall record the employee's salary.
    5. The system shall allow to edit employee's details.
    6. The system shall allow to delete employee's details.
    7.

Non-functional

1. Performance Requirements
    - The load time for user interface screens shall take no longer than two seconds.
    - The log in information shall be verified within five seconds.
    - Queries shall return results within five seconds.
2. Design Constraints
    - The System shall be a stand-alone system running in a Windows environment. The system shall be developed using Java.
3. Standards Compliance
    - There shall be consistency in variable names within the system.

4. Reliability
    - The system has to be operational on a 24 hour basis, to preserve data in case of shut down.
5. Availability
    - The system shall be available during normal hotel operating hours.
6. Security
    - Receptionist will be able to log in to the Guest, Room and Booking Management System. Accountant will have access to the Staff Management System. Access to the various subsystems will be protected by a user log in screen that requires a user name and password.
7. Maintainability
    - The System is being developed in Java. Java is an object oriented programming language and shall be easy to maintain.
8. Portability
    - The System shall run in any Microsoft Windows environment that contains Java Run time.

ALL BRIEF USE CASES

MANAGE GUESTS

Brief Use Case of Add guest profile(Manage guests by the Receptionist)

When the user logs-in as a Receptionist he can add a guest's profile. The guest information is comprised of the GuestID, firstName, lastName, address and telephone number.

Brief Use Case of Add guest profile(Manage guests by the Receptionist)

When the user logs-in as a Receptionist he can edit a guest's profile. The user can edit the firstName, lastName, address and telephone number. The guest is found using the GuestID.

Brief Use Case of View guest profile(Manage guests by the Receptionist)

When the user logs-in as a Receptionist he can view a guest's profile. The guest is found using the GuestID.

Brief Use Case of Delete guest profile(Manage guests by the Receptionist)

When the user logs-in as a Receptionist he can delete a guest's profile. The guest is found using the GuestID.

Brief Use Case of Exit Program(Manage guests by the Receptionist)

When the user logs-in as a Receptionist he can select to exit the program. The program terminates.

MANAGE BOOKING

Brief Use Case of Create new booking(Manage bookings by the Receptionist)

When the user logs-in as a Receptionist he can create a new booking in the system. The user has to provide the startDate, endDate, number of days, roomID and GuestID to make the booking.

Brief Use Case of Edit existing booking(Manage bookings by the Receptionist)

When the user logs-in as a Receptionist he can edit an existing booking in the system. The user can edit the duration of the stay and the booking is found through the roomID or GuestID.

Brief Use Case of View booking(Manage bookings by the Receptionist)

When the user logs-in as a Receptionist he can view a booking in the system. The booking is found through the roomID or GuestID.

Brief Use Case of Cancel booking(Manage booking by the Receptionist)

When the user logs-in as a Receptionist he can cancel a booking in the system. The booking is found through the roomID or GuestID.

Brief Use Case of Print Bill(Manage booking by the Receptionist)

When the user logs-in as a Receptionist he can print the bill for a booking in the system. The booking is found through the roomID or GuestID.

Brief Use Case of Exit Program(Manage booking by the Receptionist)

When the user logs-in as a Receptionist he can select to exit the program. The program terminates.

MANAGE ROOMS

Brief Use Case of Create Rooms(Manage rooms by the Receptionist)

When the user logs-in as a Receptionist he can create the rooms that the hotel has in the system. The user has to provide roomID(room number), number of beds, internet access, price pr. night, floor(the first digit of the room number).

Brief Use Case of Edit Room(Manage rooms by the Receptionist)

When the user logs-in as a Receptionist he edit the rooms that the hotel has in the system. The user can edit the number of beds, internet access, price pr. night. The room is found by the the RoomID.

Brief Use Case of Check Availability(Manage rooms by the Receptionist)

When the user logs-in as a Receptionist he can check the available rooms in the system.

Brief Use Case of Render Room out of order(Manage rooms by the Receptionist)

When the user logs-in as a Receptionist he can cancel the use of a room by rendering it OOO (Out of order). The room is found by the the RoomID.

Brief Use Case of Exit Program(Manage rooms by the Receptionist)

When the user logs-in as a Receptionist he can select to exit the program. The program terminates.

Brief Use Case of Exit Program

When the user logs-in as a Receptionist he can select to exit the program. The program terminates.

Brief Use Case of Add staff (Manage staff by the Accountant)

When the user logs-in as an Accountant he can add a staff member's profile. The staff information is comprised of the Title, first name, last name, telephone number and salary.

Brief Use Case of View Staff(Manage staff by the Accountant)

When the user logs-in as an Accountant he can view a staff member's profile information. All the staff with their Titles, first names, last names, telephone numbers and salaries are printed.

Brief Use Case of Edit staff (Manage staff by the Accountant)

When the user logs-in as an Accountant he can edit a staff member's profile. The user can edit the Title, first name, last name, telephone number and salary. The staff member is found using the first and last names.

Brief Use Case of Delete staff (Manage staff by the Accountant)

When the user logs-in as an Accountant he can delete a staff member's profile. The staff member is found using the first and last names.

Brief Use Case of Exit Program (Manage staff by the Accountant)

When the user logs-in as an Accountant he can select to exit the program. The program terminates.

CASUAL USE CASES

Casual Use Case of Add staff (Manage staff by the Accountant)

Main success scenario: The accountant logs in with his username and password. He selects “Add staff” from the submenu of “Manage staff”. He enters the Title, first name, last name, telephone number and salary to the system. The information is saved and updated. The accountant is taken back to the “Manage staff submenu”.

Alternative scenario 1: The username or password don't match. The systems prompts the user to re-enter the login information.

Alternative scenario 2: The user enters illegal data as the Title e.g numbers. The systems prompts the user to re-enter the Title.

Alternative scenario 3: The user enters illegal data as the first name e.g numbers. The systems prompts the user to re-enter the first name.

Alternative scenario 4: The user enters illegal data as the last name e.g numbers. The systems prompts the user to re-enter the last name.

Alternative scenario 5: The user enters illegal data as the telephone number e.g. letters. The systems prompts the user to re-enter the telephone number.

Alternative scenario 6: The user enters illegal data as the salary e.g. letters. The systems prompts the user to re-enter the salary.

Casual Use Case of View staff (Manage staff by the Accountant)

Main success scenario: The accountant logs in with his username and password. He selects "View staff" from the submenu of "Manage staff". All the staff with their Titles, first names, last names, telephone numbers and salaries are printed. The accountant is taken back to the "Manage staff submenu"

Alternative scenario 1: The username or password don't match. The systems prompts the user to re-enter the login information.

Casual Use Case of Edit staff (Manage staff by the Accountant)

Main success scenario: The accountant logs in with his username and password. He selects "Edit staff" from the submenu of "Manage staff". He enters the first name and last name of the staff member. The system asks the user to select what he would like to edit (Title, first name, last name, telephone number or salary). The user selects what he would like to edit. He enters the new information in the system. The information is saved and updated. The accountant is taken back to the "Manage staff submenu".

Alternative scenario 1: The username or password don't match. The systems prompts the user to re-enter the login information.

Alternative scenario 2: The user enters illegal data as the Title e.g numbers. The systems prompts the user to re-enter the Title.

Alternative scenario 3: The user enters illegal data as the first name e.g numbers. The systems prompts the user to re-enter the first name.

Alternative scenario 4: The user enters illegal data as the last name e.g numbers. The systems prompts the user to re-enter the last name.

Alternative scenario 5: The user enters illegal data as the telephone number e.g. letters. The systems prompts the user to re-enter the telephone number.

Alternative scenario 6: The user enters illegal data as the salary e.g. letters. The systems prompts the user to re-enter the salary.

Alternative scenario 7: The system cannot find a staff member with the provide name. The systems prompts the user to re-enter the first name and the last name.

Casual Use Case of Delete staff (Manage staff by the Accountant)

Main success scenario: The accountant logs in with his username and password. He selects "Delete staff" from the submenu of "Manage staff". He enters the first name and last name of the staff member. The system asks the user to confirm the deletion. The user confirms. The information is updated. The accountant is taken back to the "Manage staff submenu".

Alternative scenario 1: The username or password don't match. The systems prompts the user to re-enter the login information.

Alternative scenario 2: The system cannot find a staff member with the provide name. The systems prompts the user to re-enter the first name and the last name.

Casual Use Case of Exit Program (Manage staff by the Accountant)

Main success scenario: The accountant logs in with his username and password. He selects "Exit program" from the submenu of "Manage staff". The program terminates.

Alternative scenario 1: The username or password don't match. The systems prompts the user to re-enter the login information.

Fully Dressed Use Case 1: Create Booking

Use Case: Create Booking

Scope: Hotel Plaza System
Level: User Goal
Primary Actor: Receptionist
Stakeholders and Interests:

- Guest: Wants to book a room and calls Hotel Plaza.
- Receptionist: Wants to create a booking with no mistakes in order to satisfy the guest.
- Hotel Plaza Management: Wants to provide excellent hospitality and service.

- Support: Wants to easily perform maintenance and repairs.

Preconditions: The Guest must have requested for a booking to be made. The Receptionist must be able to log-in and use the System. The 'Create Booking' option must be available.
Success Guarantee (or Postconditions): Booking is saved. Room availability is updated. Bill is generated.

Main Success Scenario (or Basic Flow):

1. Receptionist logs in. (Subflow)
2. System displays main menu.
3. Receptionist selects 'Manage Bookings' from the options available.
4. System displays 'Bookings' menu.
5. Receptionist selects 'Create New Booking' from the options available.
6. Receptionist enters check-in and check-out date.
7. System displays how many rooms available.
8. Receptionist enters room type and quantity.
9. Systems displays total price.
10. Receptionist confirms total price.
11. Receptionist enters guest's details (first name, last name, telephone, address).
12. System displays guest details.
13. Receptionist confirms guest details.
14. System generates bill.
15. System records booking.
16. End of use case.

Extensions (or Alternative Flows):

Rooms Unavailability

At the 'System shows how many rooms available' of the Basic Flow, if all rooms are occupied, then

- System informs Receptionist that there is no availability on selected dates.
- System asks if the Receptionist wants to change the dates.
  - If yes, then the use case resumes from step 6.
  - If no, then the use case ends.

Booking Multiple Rooms

At the 'Receptionist confirms room details and total price' step of the Basic Flow, if the Receptionist wants to add more rooms, then

- Receptionist selects 'Add more rooms' from the options available
- Use case resumes from step 8.

Existing Customer

At the 'Receptionist enters guest's details' step of the Basic Flow, if first name and last name correspond to a guest in our database, then

- System displays guest with the given name.
  - If the guest details correct, then Receptionist confirms the use case resumes from step 14.
  - If not, the Basic Flow resumes from step 11.

Cancelling Session

If at any step of the use case the Receptionist wants to cancel the booking, then

- System asks for confirmation.
- If Receptionist confirms, then
  - If yes, System cancels the operation and the use case ends.
  - If no, use case resumes.

Error Handling

1. Handle 'Check-in date is a past date'

At the 'Receptionist enters check-in date' step of the Basic Flow, if the date entered is a date is a past date, then

- The System informs the Receptionist that the input is invalid.
- The use case resumes from step 6.

2. Handle 'Check-out date earlier than check-in date'

At the 'Receptionist enters check-out date' step of the Basic Flow, if the date entered is a date is earlier than the check-in date, then

- The System informs the Receptionist that the input is invalid.
- The use case resumes from step 6.

3. Handle 'First name has illegal data'

At the 'Receptionist enters first name' step of the Basic Flow, if the Receptionist enters illegal data (e.g. numbers), then

- The System informs the Receptionist that the input is invalid.
- The use case resumes from step 11.

4. Handle 'Last name has illegal data'

At the 'Receptionist enters last name' step of the Basic Flow, if the Receptionist enters illegal data (e.g. numbers), then

- The System informs the Receptionist that the input is invalid.
- The use case resumes from step 11.

5. Handle 'Telephone number has illegal data'

At the 'Receptionist enters telephone number' step of the Basic Flow, if the Receptionist enters illegal data (e.g. letters), then

- The System informs the Receptionist that the input is invalid.
- The use case resumes from step 11.

6. Handle 'Address has illegal data'

At the 'Receptionist enters first name' step of the Basic Flow, if the Receptionist enters illegal data (e.g. numbers), then

- The System informs the Receptionist that the input is invalid.
- The use case resumes from step 11.

Subflow

- User log-in

Other Hotel Plaza System functions

- Extend booking

- Change room price
- Update guest name
- Create staff member
- Update staff salary
- Create guest
- Update guest telephone

Fully Dressed Use Case 2: Add Staff

Use Case: Add Staff

Scope: Hotel Plaza System
Level: User Goal
Primary Actor: Accountant
Stakeholders and Interests:

- Accountant: Wants to create a user profile for a new employee.
- Employee: Wants to get his/her data registered in a safe way.
- Hotel Plaza Management: Wants to maintain an updated database of its employees.
- Support: Wants to easily perform maintenance and repairs.

Preconditions: The Management must have requested for a new user profile to be created. The Accountant must be able to log-in and use the System. The 'Manage Staff' option must be available.

Success Guarantee (or Postconditions): New user profile is created.

Main Success Scenario (or Basic Flow):

1. Accountant logs-in. (Subflow)
2. System displays main menu.
3. Accountant selects 'Manage Staff' from the options available.
4. System displays 'Manage Staff' menu.
5. Accountant selects 'Register staff' from the options available.
6. System displays 'Staff roles' menu.
7. Accountant selects 'Cleaning Staff' from the options available.
8. Accountant enters staff details (title, salary, first name, last name, telephone, address, username, password).
9. Accountant saves new employee.
10. System records entry.
11. End of use case.

Extensions (or Alternative Flows):

Existing Staff Member

At the 'Accountant enters staff details (title, salary, first name, last name, telephone, address, username, password)' step of the Basic Flow, if the Accountant attempts to enter a new member with the same first name and last name as an existing staff member in our system, then

- System displays existing staff member
  -If Accountant confirms then
    - System informs Accountant that the staff member already exists.

- System cancels registration of new staff
- Use case ends.

- If Accountant wants to proceed, then
    - System informs that the name should not be different to avoid duplicates.
    - Accountant enters new name.
    - Use case resumes.

User Input Handling

1. Handle 'Staff Role does not exist'

At the 'Accountant selects "Cleaning Staff" from the three options available' step of the Basic Flow, if the number entered is not a valid room number, then

- The System informs the Receptionist that the input is invalid.
- The use case continues from step 7.

2. Handle 'First name has illegal data'

At the 'Accountant enters first name' step of the Basic Flow, if the Accountant enters illegal data (e.g. numbers), then

- The System informs the Accountant that the input is invalid.
- The use case continues from step 8.

3. Handle 'Last name has illegal data'

At the 'Accountant enters last name' step of the Basic Flow, if the Accountant enters illegal data (e.g. numbers), then

- The System informs the Accountant that the input is invalid.
- The use case continues from step 9.

4. Handle 'Telephone number has illegal data'

At the 'Accountant enters telephone number' step of the Basic Flow, if the Accountant enters illegal data (e.g. letters), then

- The System informs the Receptionist that the input is invalid.
- The use case continues from step 10.

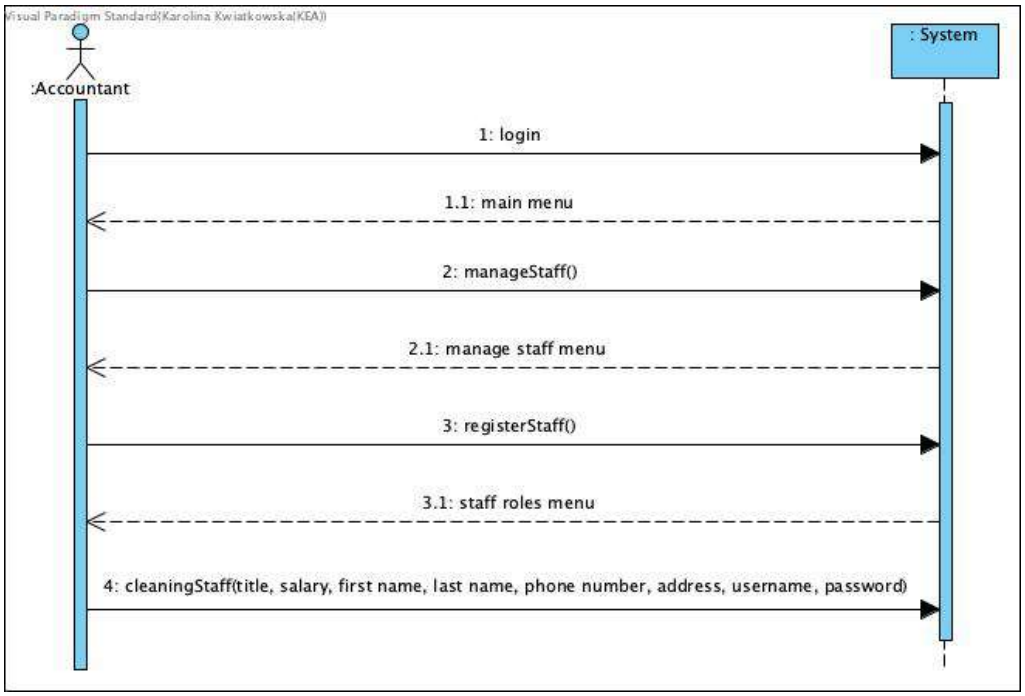
Subflow

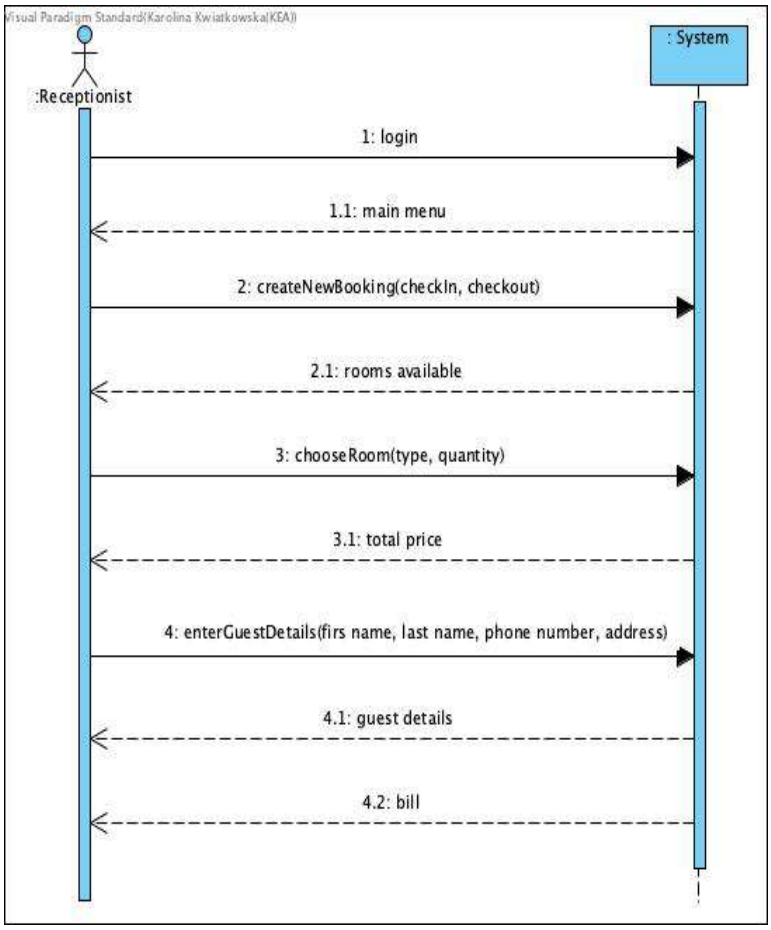
- User log-in

Other Hotel Plaza System functions

- Create booking
- Extend booking
- Change room price
- Update guest name
- Update staff salary
- Create guest
- Update guest telephone

SYSTEM SEQUENCE DIAGRAMS





OPERATION CONTRACT

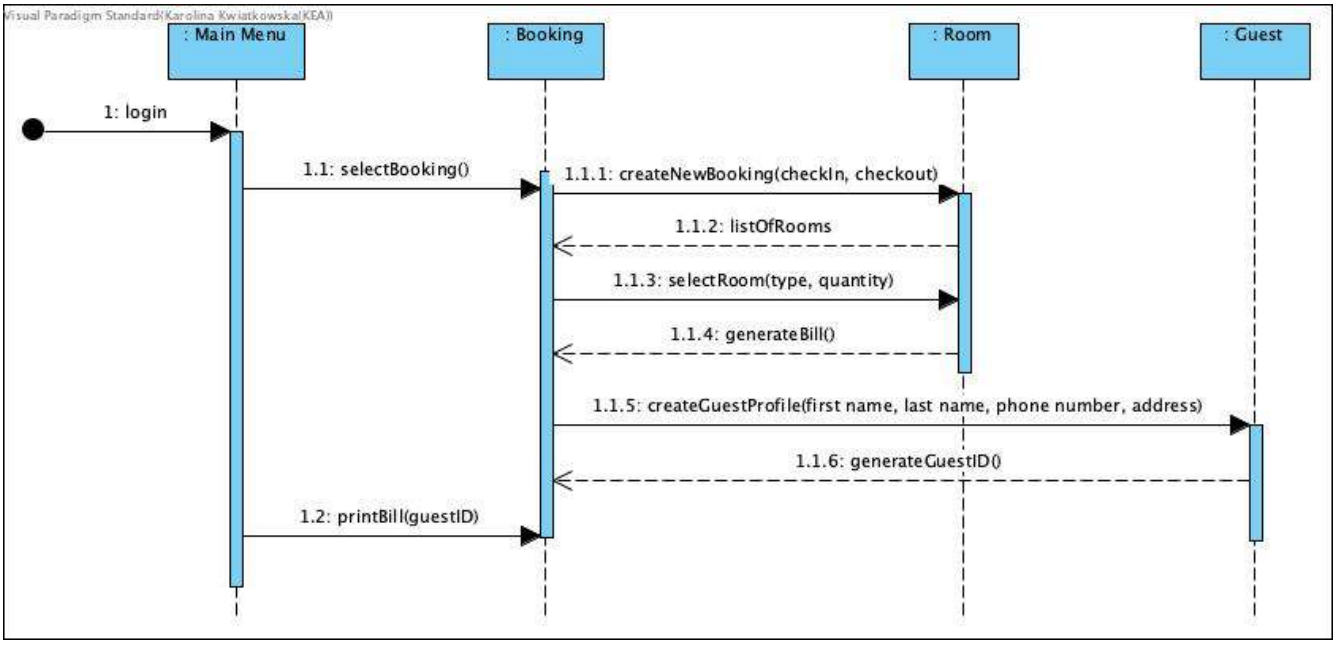
Hotel Plaza System: Contracts

Notebook:	Software Design		
Created:	06/11/2019 16.56	**Updated:**	06/11/2019 18.14
Author:	dimitris.giokas1@gmail.com		

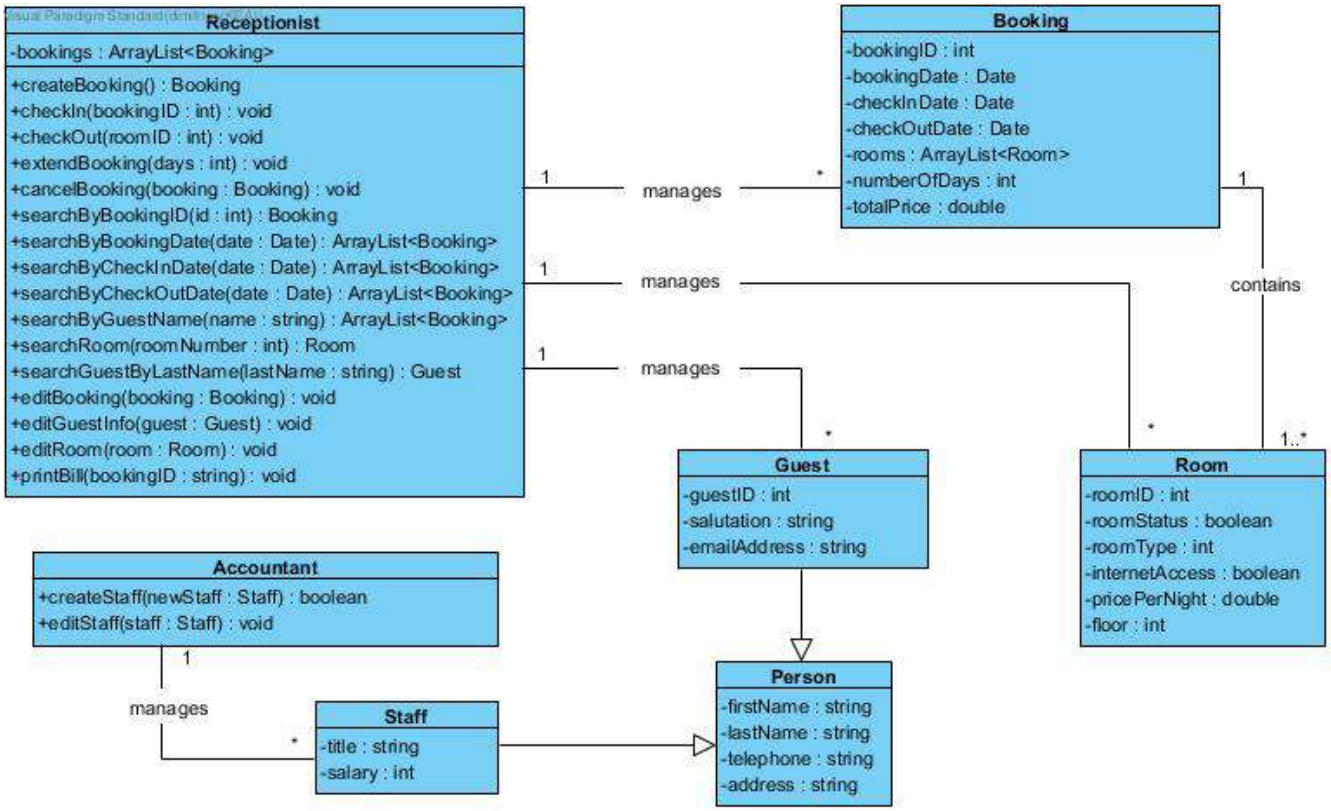
Contract 1: Create Booking	
Operation:	createBooking(dayIn : Date, dateOut : Date, roomType : RoomType, roomID : Room, firstName : String, lastName : String, telephone : int)
Cross References:	Use Case: Create Booking
Preconditions:	none
Postconditions:	• A Booking instance (booking) was created. • A Guest instance (guest) was created. • Attributes of booking were initialized. • Attributes of guest were initialized.

Contract 3: Confirm Booking	
Operation:	confirmBooking()
Cross References:	Use Case: Create Booking
Preconditions:	There is a booking underway.
Postconditions:	• Booking was associated with the Room. • Attributes of the associated room changed (isAvailable = false). • Attribute of the Booking instance changed (isCompleted = true).

SEQUENCE DIAGRAM FROM FULLY DRESSED



CLASS DIAGRAM



GANTT CHART

* = an automatically calculated cell

	START DATE	DAY OF MONTH*	END DATE	DURATION* (WORK DAYS)	DAYS COMPLETE*	DAYS REMAINING*	TEAM MEMBER	PERCENT COMPLETE
Design								
Write Requirements/Needs list	11/5	5	11/22	17	17	0	Elias	100%
Identify all Actors and draw a Use Case Diagram	11/5	5	11/22	17	17	0	Dimitris	100%
Write a Vision & create a Glossary	11/5	5	11/22	17	17	0	Elias	100%
Create a Domain Model	11/5	5	11/22	17	17	0	Elias	100%
Write Functional/Non-Functional or FURPS+ from Requirements/Needs	11/5	5	11/22	17	17	0	Karolina	100%
Write all brief use case descriptions	11/5	5	11/22	17	17	0	Elias	100%
Write 2 casual case descriptions from brief		5	11/22	17	17	0	Elias	100%
Write 2 fully dressed use case descriptions from casual	11/5	5	11/22	17	17	0	Dimitris	100%
Create 2 System Sequence Diagram	11/7	5	11/22	17			Karolina	100%
Create an Operation Contract list	11/6	5	11/22	17	17	0	Dimitris	100%
Create one or more Sequence Diagram from Fully dressed UC	11/7	5	11/22	17	17	0	Karolina	100%
Create a Design Class Diagram	11/5	5	11/22	17	17	0	Dimitris	100%