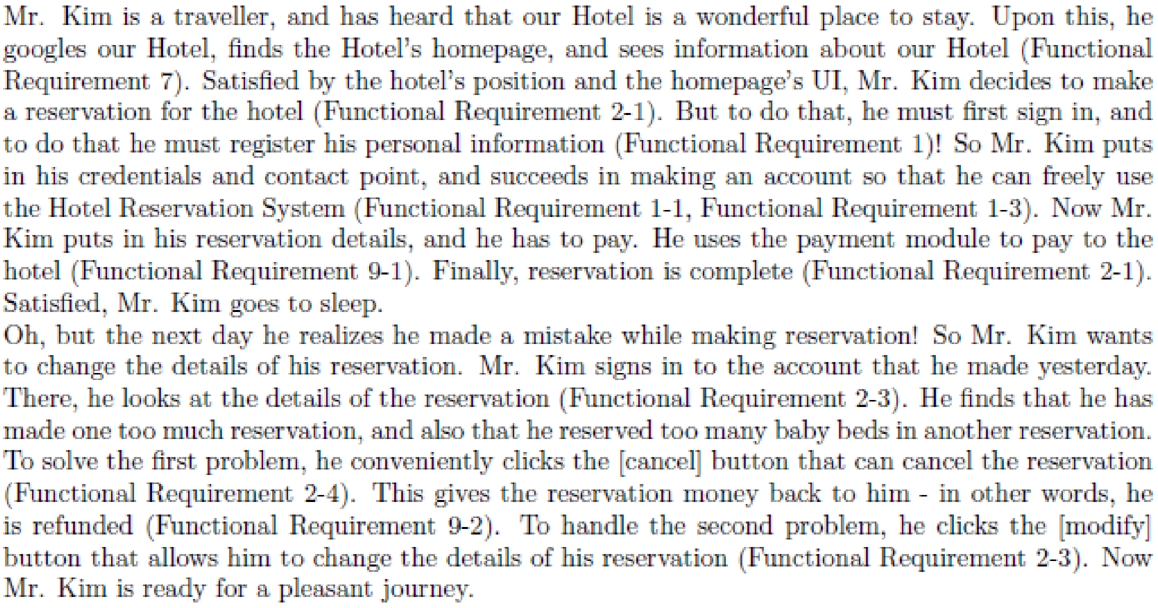
[CS350] 2018-Fall Team Project
Sprint 01: Samples

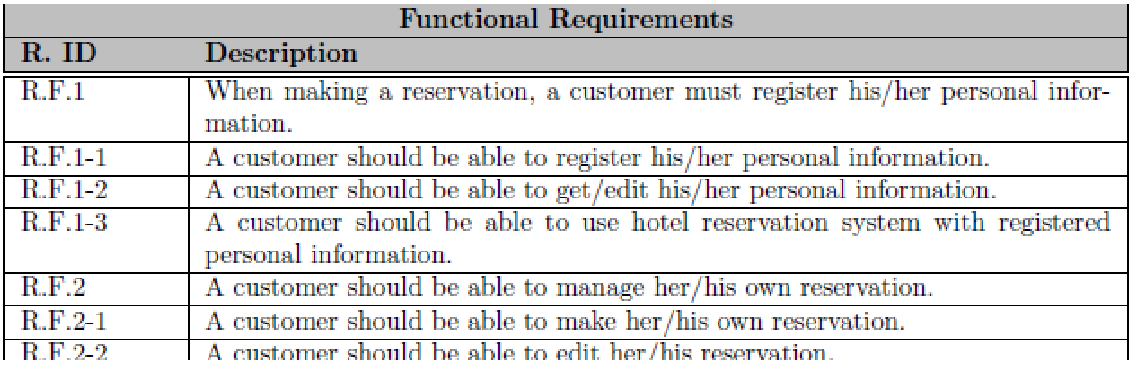
This document contains example models, description tables, and some guidelines for Project#03.

- **User Story**

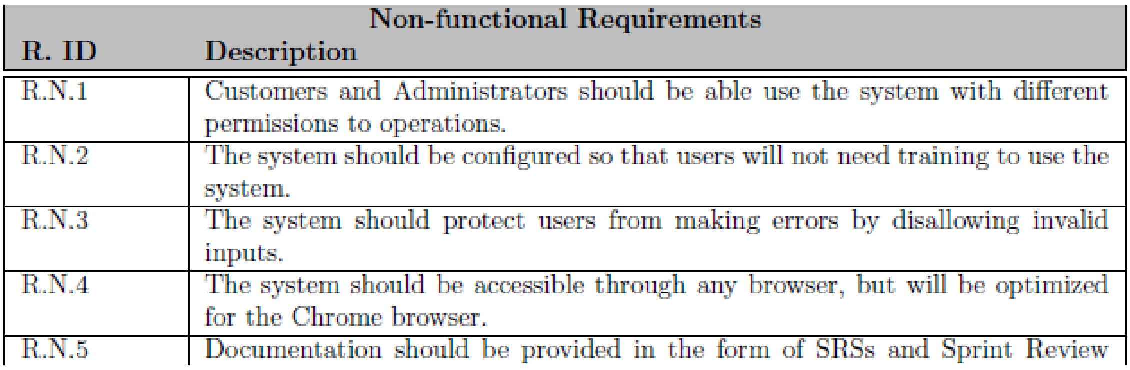


Mr. Kim is a traveller, and has heard that our Hotel is a wonderful place to stay. Upon this, he googles our Hotel, finds the Hotel's homepage, and sees information about our Hotel (Functional Requirement 7). Satisfied by the hotel's position and the homepage's UI, Mr. Kim decides to make a reservation for the hotel (Functional Requirement 2-1). But to do that, he must first sign in, and to do that he must register his personal information (Functional Requirement 1)! So Mr. Kim puts in his credentials and contact point, and succeeds in making an account so that he can freely use the Hotel Reservation System (Functional Requirement 1-1, Functional Requirement 1-3). Now Mr. Kim puts in his reservation details, and he has to pay. He uses the payment module to pay to the hotel (Functional Requirement 9-1). Finally, reservation is complete (Functional Requirement 2-1). Satisfied, Mr. Kim goes to sleep.
Oh, but the next day he realizes he made a mistake while making reservation! So Mr. Kim wants to change the details of his reservation. Mr. Kim signs in to the account that he made yesterday. There, he looks at the details of the reservation (Functional Requirement 2-3). He finds that he has made one too much reservation, and also that he reserved too many baby beds in another reservation. To solve the first problem, he conveniently clicks the [cancel] button that can cancel the reservation (Functional Requirement 2-4). This gives the reservation money back to him - in other words, he is refunded (Functional Requirement 9-2). To handle the second problem, he clicks the [modify] button that allows him to change the details of his reservation (Functional Requirement 2-3). Now Mr. Kim is ready for a pleasant journey.

- **Requirements Table**

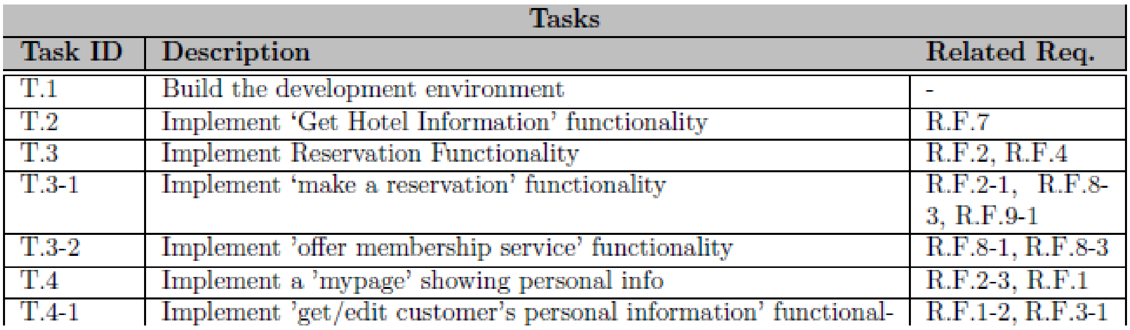


Functional Requirements	
R. ID	**Description**
R.F.1	When making a reservation, a customer must register his/her personal information.
R.F.1-1	A customer should be able to register his/her personal information.
R.F.1-2	A customer should be able to get/edit his/her personal information.
R.F.1-3	A customer should be able to use hotel reservation system with registered personal information.
R.F.2	A customer should be able to manage her/his own reservation.
R.F.2-1	A customer should be able to make her/his own reservation.
R.F.2-2	A customer should be able to edit her/his reservation.



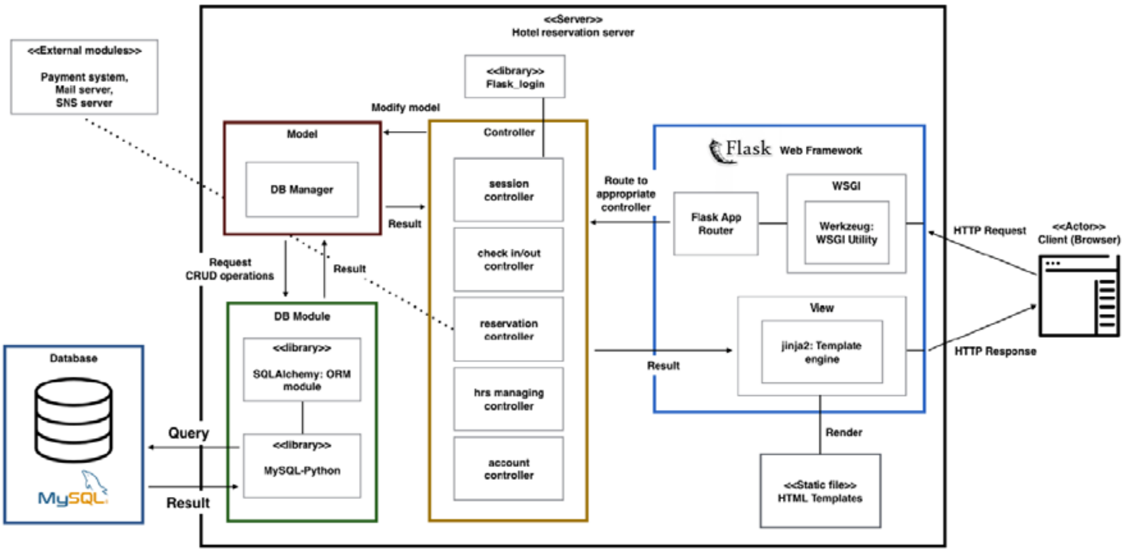
Non-functional Requirements	
R. ID	**Description**
R.N.1	Customers and Administrators should be able use the system with different permissions to operations.
R.N.2	The system should be configured so that users will not need training to use the system.
R.N.3	The system should protect users from making errors by disallowing invalid inputs.
R.N.4	The system should be accessible through any browser, but will be optimized for the Chrome browser.
R.N.5	Documentation should be provided in the form of SRSs and Sprint Review

- **Tasks**

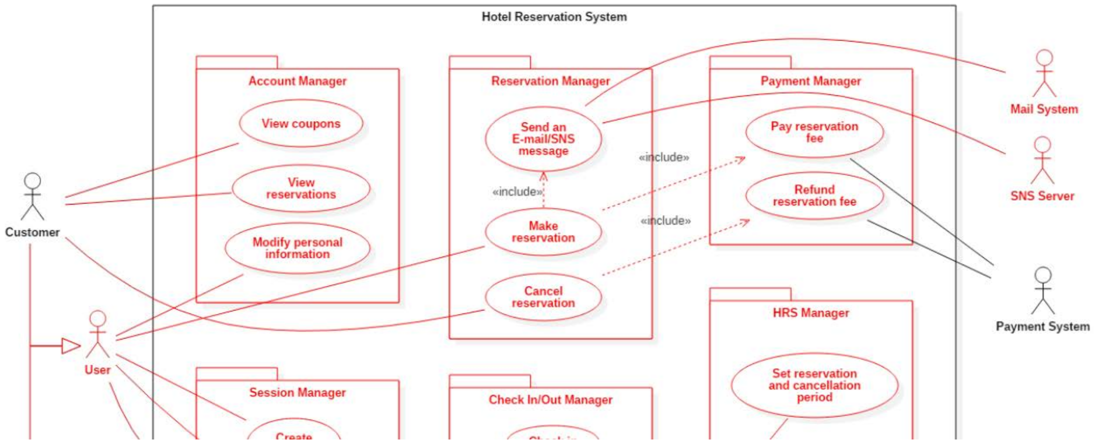


Tasks		
Task ID	Description	Related Req.
T.1	Build the development environment	-
T.2	Implement 'Get Hotel Information' functionality	R.F.7
T.3	Implement Reservation Functionality	R.F.2, R.F.4
T.3-1	Implement 'make a reservation' functionality	R.F.2-1, R.F.8-3, R.F.9-1
T.3-2	Implement 'offer membership service' functionality	R.F.8-1, R.F.8-3
T.4	Implement a 'mypage' showing personal info	R.F.2-3, R.F.1
T.4-1	Implement 'get/edit customer's personal information' functional-	R.F.1-2, R.F.3-1

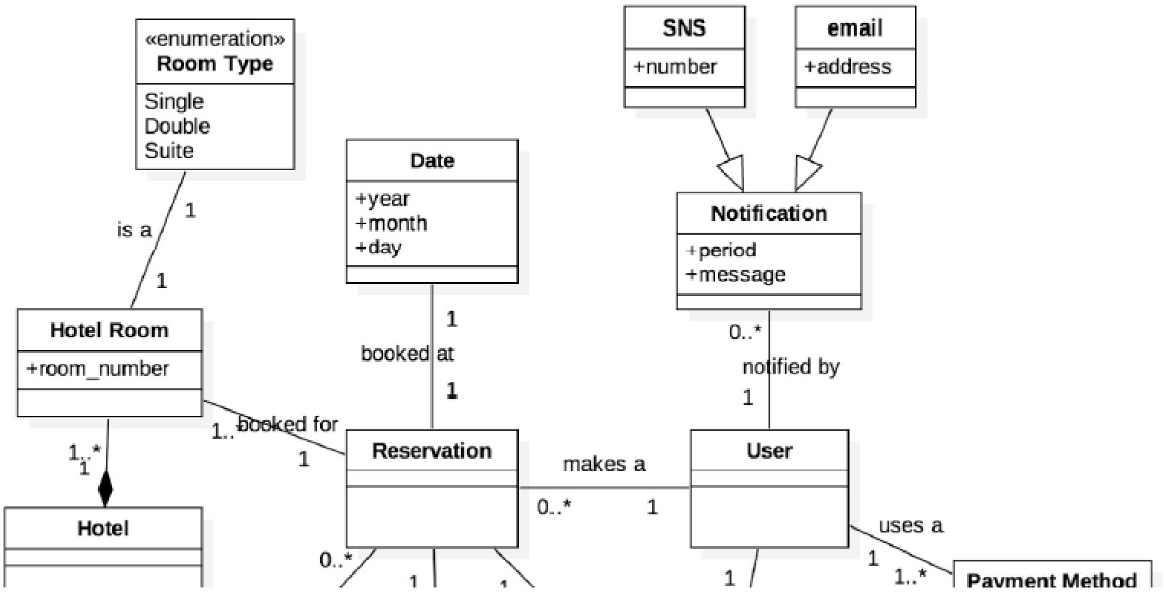
- **System Architecture**



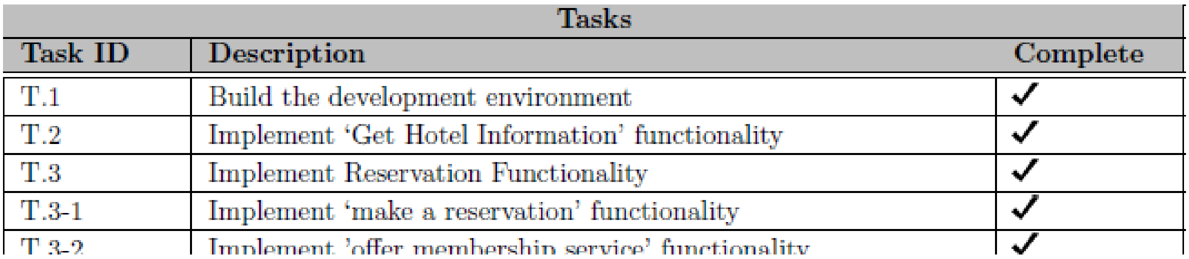
- **Refined Use Case Diagram**



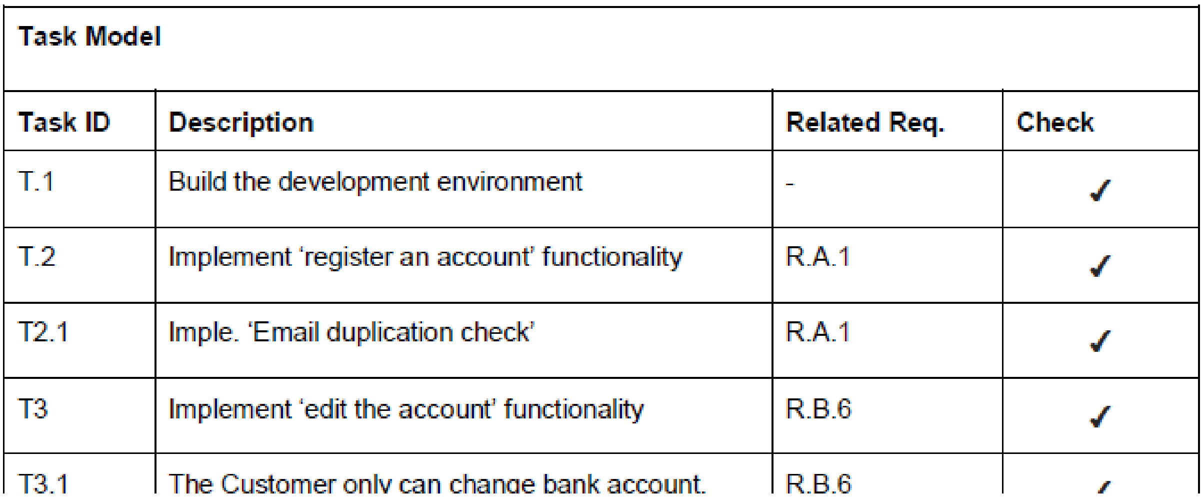
- **Class Diagram**



- **Achievements**

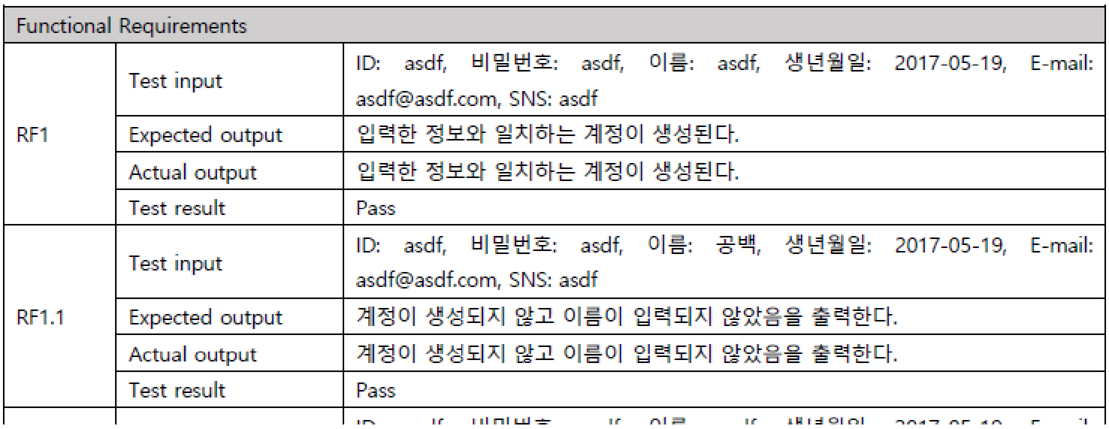


Tasks		
Task ID	Description	Complete
T.1	Build the development environment	✓
T.2	Implement 'Get Hotel Information' functionality	✓
T.3	Implement Reservation Functionality	✓
T.3-1	Implement 'make a reservation' functionality	✓
T.3-2	Implement 'offer membership service' functionality	✓

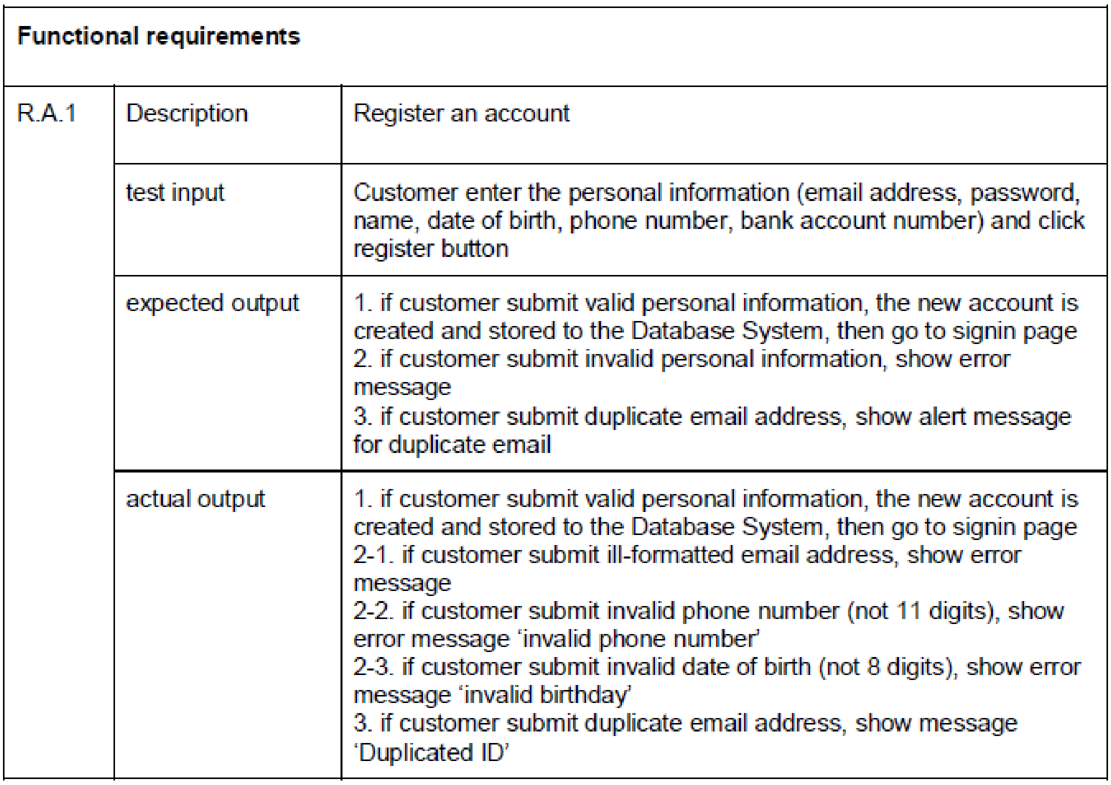


Task Model			
Task ID	Description	Related Req.	Check
T.1	Build the development environment	-	✓
T.2	Implement 'register an account' functionality	R.A.1	✓
T2.1	Imple. 'Email duplication check'	R.A.1	✓
T3	Implement 'edit the account' functionality	R.B.6	✓
T3.1	The Customer only can change bank account.	R.B.6	✓

- **Requirements Validation (Testing)**

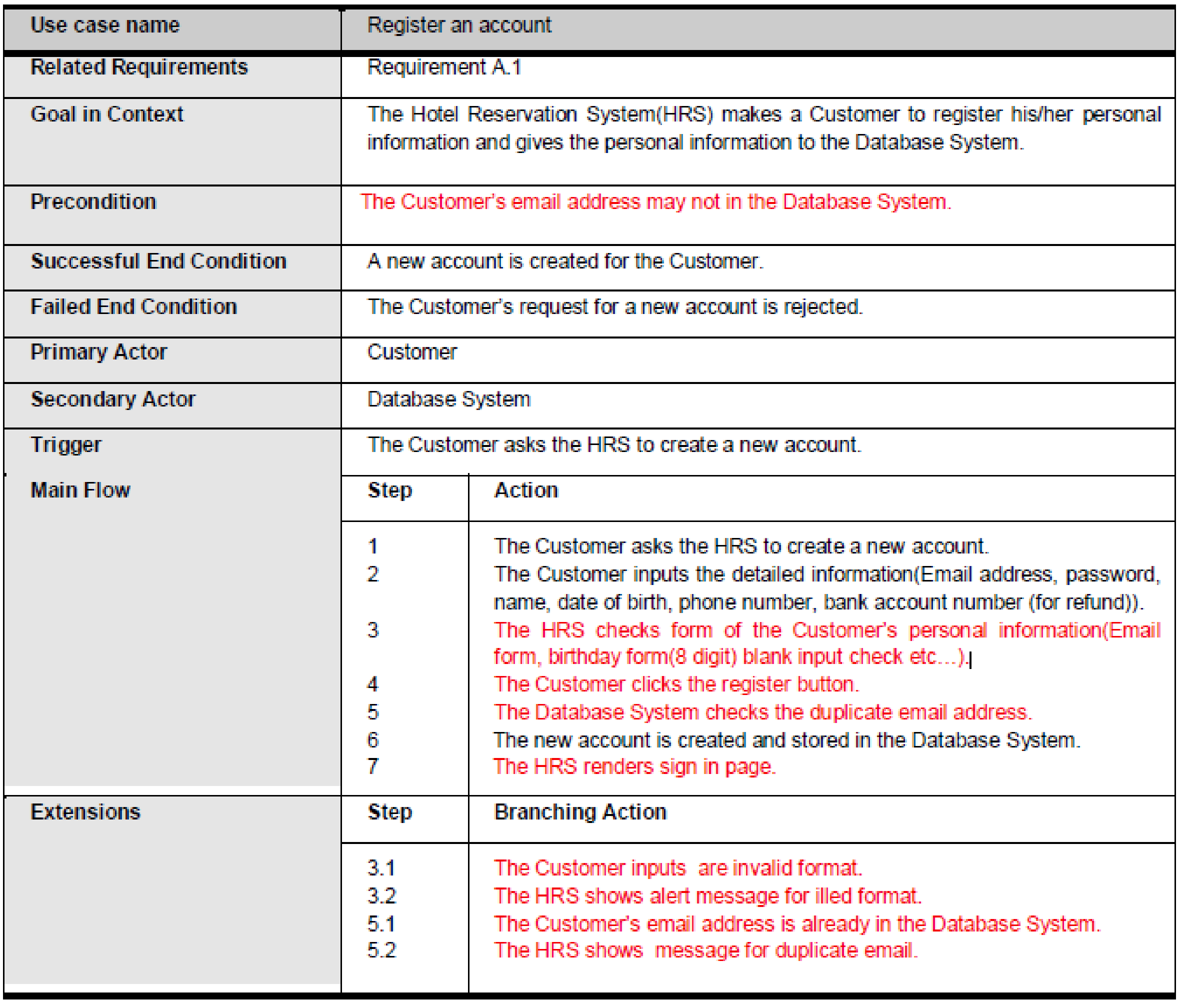


Functional Requirements		
RF1	Test input	ID: asdf, 비밀번호: asdf, 이름: asdf, 생년월일: 2017-05-19, E-mail: asdf@asdf.com, SNS: asdf
	Expected output	입력한 정보와 일치하는 계정이 생성된다.
	Actual output	입력한 정보와 일치하는 계정이 생성된다.
	Test result	Pass
RF1.1	Test input	ID: asdf, 비밀번호: asdf, 이름: 공백, 생년월일: 2017-05-19, E-mail: asdf@asdf.com, SNS: asdf
	Expected output	계정이 생성되지 않고 이름이 입력되지 않았음을 출력한다.
	Actual output	계정이 생성되지 않고 이름이 입력되지 않았음을 출력한다.
	Test result	Pass

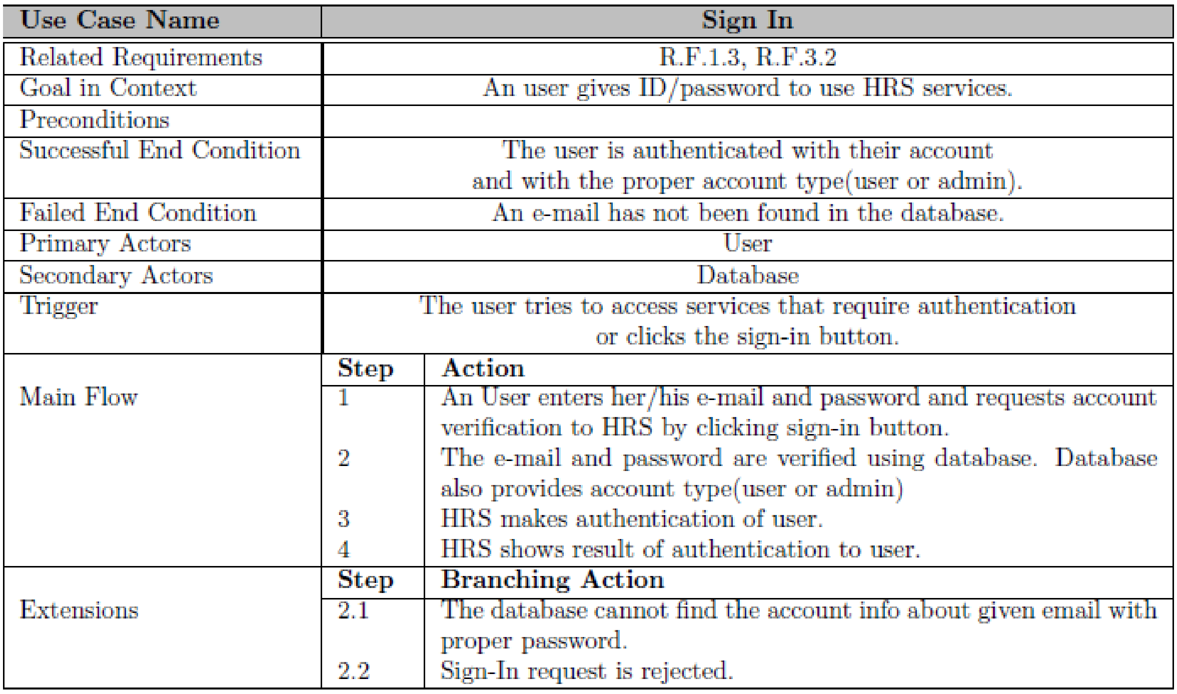


Functional requirements		
R.A.1	Description	Register an account
	test input	Customer enter the personal information (email address, password, name, date of birth, phone number, bank account number) and click register button
	expected output	1. if customer submit valid personal information, the new account is created and stored to the Database System, then go to signin page 2. if customer submit invalid personal information, show error message 3. if customer submit duplicate email address, show alert message for duplicate email
	actual output	1. if customer submit valid personal information, the new account is created and stored to the Database System, then go to signin page 2-1. if customer submit ill-formatted email address, show error message 2-2. if customer submit invalid phone number (not 11 digits), show error message 'invalid phone number' 2-3. if customer submit invalid date of birth (not 8 digits), show error message 'invalid birthday' 3. if customer submit duplicate email address, show message 'Duplicated ID'

- **Use Case Description**

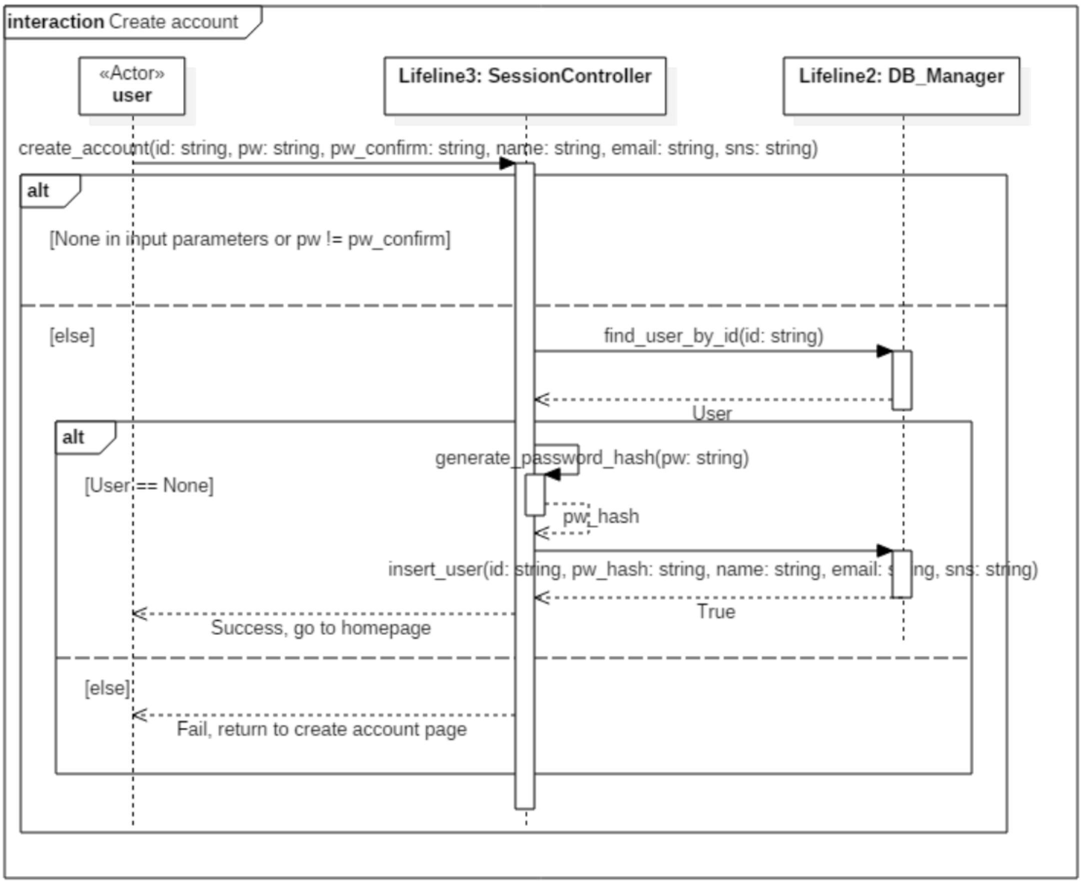


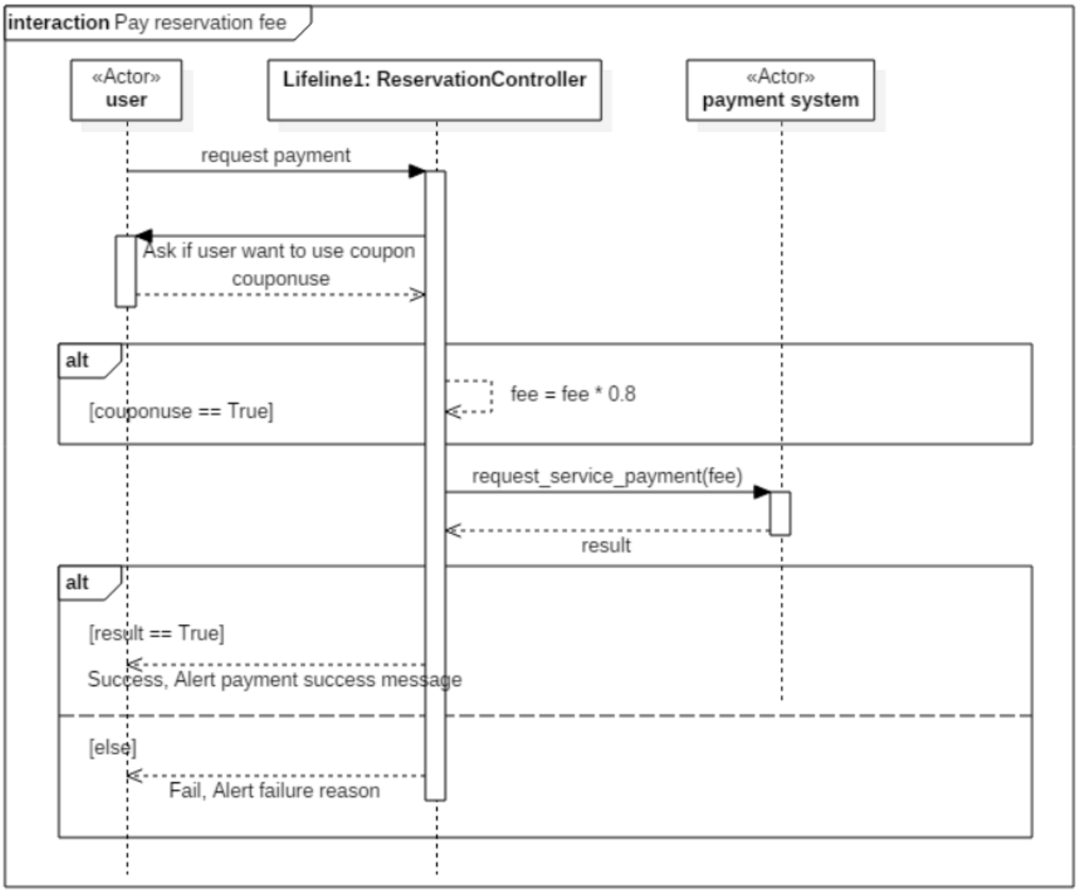
Use case name	Register an account	
Related Requirements	Requirement A.1	
Goal in Context	The Hotel Reservation System(HRS) makes a Customer to register his/her personal information and gives the personal information to the Database System.	
Precondition	The Customer's email address may not in the Database System.	
Successful End Condition	A new account is created for the Customer.	
Failed End Condition	The Customer's request for a new account is rejected.	
Primary Actor	Customer	
Secondary Actor	Database System	
Trigger	The Customer asks the HRS to create a new account.	
Main Flow	**Step**	**Action**
	1	The Customer asks the HRS to create a new account.
	2	The Customer inputs the detailed information(Email address, password, name, date of birth, phone number, bank account number (for refund)).
	3	The HRS checks form of the Customer's personal information(Email form, birthday form(8 digit) blank input check etc…).
	4	The Customer clicks the register button.
	5	The Database System checks the duplicate email address.
	6	The new account is created and stored in the Database System.
	7	The HRS renders sign in page.
Extensions	**Step**	**Branching Action**
	3.1	The Customer inputs are invalid format.
	3.2	The HRS shows alert message for illed format.
	5.1	The Customer's email address is already in the Database System.
	5.2	The HRS shows message for duplicate email.



Use Case Name	Sign In	
Related Requirements	R.F.1.3, R.F.3.2	
Goal in Context	An user gives ID/password to use HRS services.	
Preconditions		
Successful End Condition	The user is authenticated with their account and with the proper account type(user or admin).	
Failed End Condition	An e-mail has not been found in the database.	
Primary Actors	User	
Secondary Actors	Database	
Trigger	The user tries to access services that require authentication or clicks the sign-in button.	
Main Flow	**Step**	**Action**
	1	An User enters her/his e-mail and password and requests account verification to HRS by clicking sign-in button.
	2	The e-mail and password are verified using database. Database also provides account type(user or admin)
	3	HRS makes authentication of user.
	4	HRS shows result of authentication to user.
Extensions	**Step**	**Branching Action**
	2.1	The database cannot find the account info about given email with proper password.
	2.2	Sign-In request is rejected.

- **Sequence Diagram**


interaction Create account
«Actor»
user
Lifeline3: SessionController
Lifeline2: DB_Manager
create_account(id: string, pw: string, pw_confirm: string, name: string, email: string, sns: string)
alt
[None in input parameters or pw != pw_confirm]
[else]
find_user_by_id(id: string)
User
alt
generate_password_hash(pw: string)
[User == None]
pw_hash
insert_user(id: string, pw_hash: string, name: string, email: string, sns: string)
True
Success, go to homepage
[else]
Fail, return to create account page


interaction Pay reservation fee
«Actor»
user
Lifeline1: ReservationController
«Actor»
payment system
request payment
Ask if user want to use coupon
couponuse
alt
fee = fee * 0.8
[couponuse == True]
request_service_payment(fee)
result
alt
[result == True]
Success, Alert payment success message
[else]
Fail, Alert failure reason