University Housing System

Team names:

1- Basant Benyamen
2- Mariam Nasser
3- Nada Maaman

Description :

University Housing System

Our system is responsible for the management of university housing and has many functions, including:

Manage the booking and payment through the student's nomination card and some other data, such as his place of residence, where the reservation is prioritized for students from distant cities, the booking is opened for a certain period, and in case of admission, the acceptance message appears on the system. Then the electronic payment door opens for a specified period. If the student skips it, his name will automatically be removed from the accepted list and another student will be added instead of him from reserve list. A code is created for each acceptable student. At this stage, the students will also choose if they want a comprehensive meal reservation or not.

The system manages students' departure and return times. There is a screen installed at the exit door. The student enters his or her student code and presses the exit button or the return button. Thus, students who miss the exact return time are automatically identified and warnings are automatically issued.

The system manages students' travel and return days through the installed screen, where the student enters his code and chooses to travel or return. The system also calculates the number of students in housing who will receive a meal that day.

Each student can receive his meal daily by informing the chef of his student code, which in turn puts a check mark in front of the name of the student who has received his meal.

- # Functions Requirement:

1- Booking in university housing

1.1- Login: Student shall log in with username and password

1.1.1- sign up: if the student doesn't have an account, he shall enter SSN, username and password

1.2- Book as a new student: the student shall enter his name, faculty name, level, SSN, his phone number, parent's phone number, date of birth, residence

1.2.1- After that the message shows "Please enter the site every day within the next 2 weeks to ensure that the booking has been accepted or refused.”

1.3- Renew booking: the student shall enter his code to check that he had warnings before or not

1.4- payment: choose if you want a meal or not then pay using a bank account card

1.4.1- if a new student generates code

Note: There is a screen installed next to the exit door on which the desktop application is installed

1- Travel and return

2.1-student shall enter his code

2.2- update list: verifying whether the student has a meal or not, updating the list of names that appear to the chef

2- Exit and arrive

3.1- student shall enter his code
3.2- record the time the student traveled, the time of return, and the date
3.3- if the time at which the student arrives exceeds the time specified for arrive, a message will appear that he received a warning
3.4- generate warnings that the housing manager can deal with

Nonfunctional requirement:

1. Performance:

Response Time: The website should load within 3 seconds to ensure a seamless user experience.

Scalability: The system should handle an increasing number of users during peak times, such as registration periods, without significant degradation in performance.

2. Reliability:

Availability: The website should be available, allowing students to access information or make bookings.

3. Security:

Data Encryption: All sensitive data, including personal information and payment details, should be encrypted to protect against unauthorized access.

User Authentication: A secure authentication mechanism should be in place to ensure that only authorized users can access certain features or data.

4. Usability:

User Interface (UI) Design: The website should have an intuitive and user-friendly interface to facilitate easy navigation for both novice and experienced users.

5. Scalability

User Growth: The website should easily handle an increasing number of users and housing listings as the student population grows.

6. Compatibility:

Browser Compatibility: The website should be compatible with major web browsers (Chrome, Firefox, Safari, etc.) to ensure a consistent experience for all users.

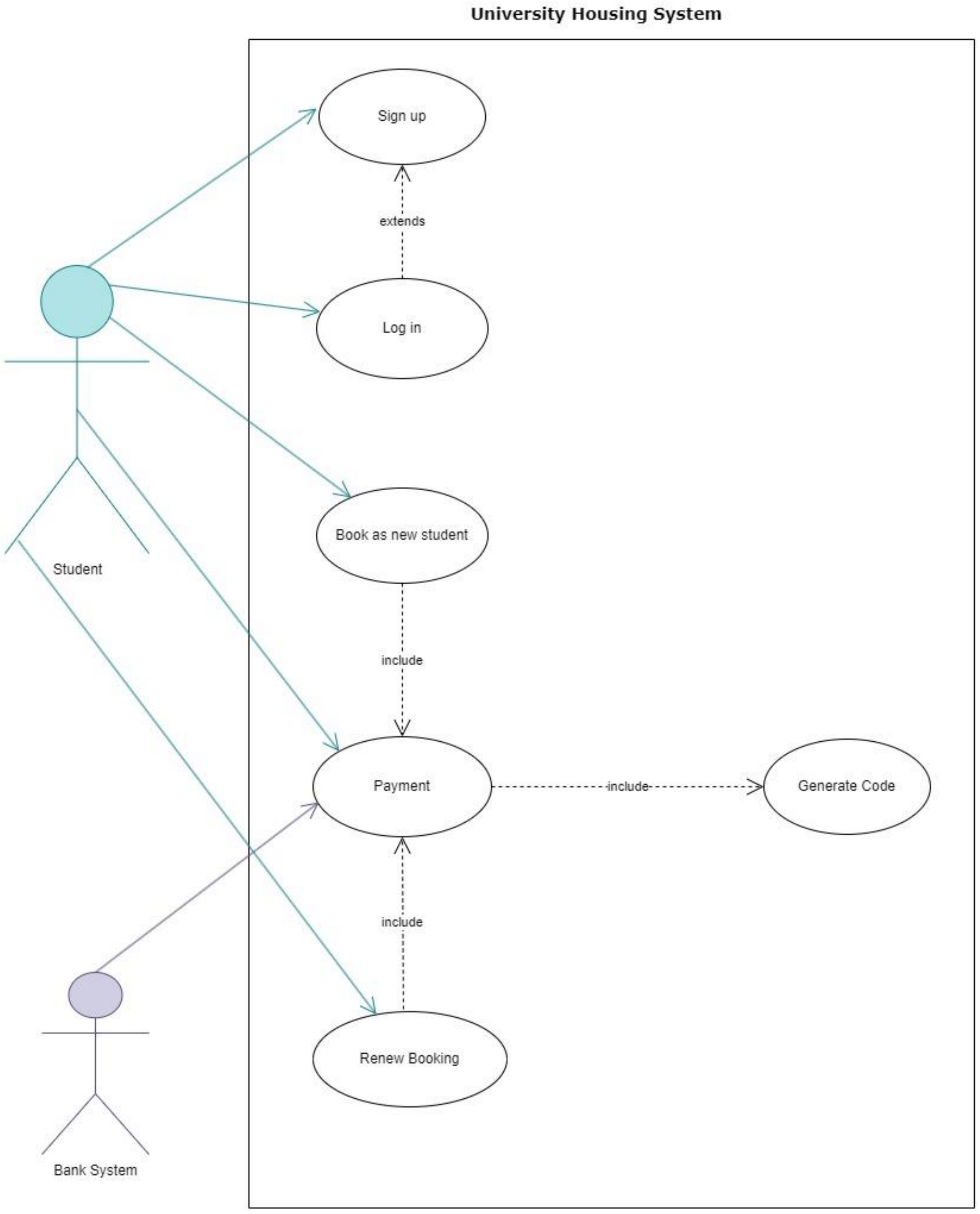
Device Compatibility: The website should be responsive and functional across various devices, including laptops, tablets, and smartphones.

7. Interoperability:

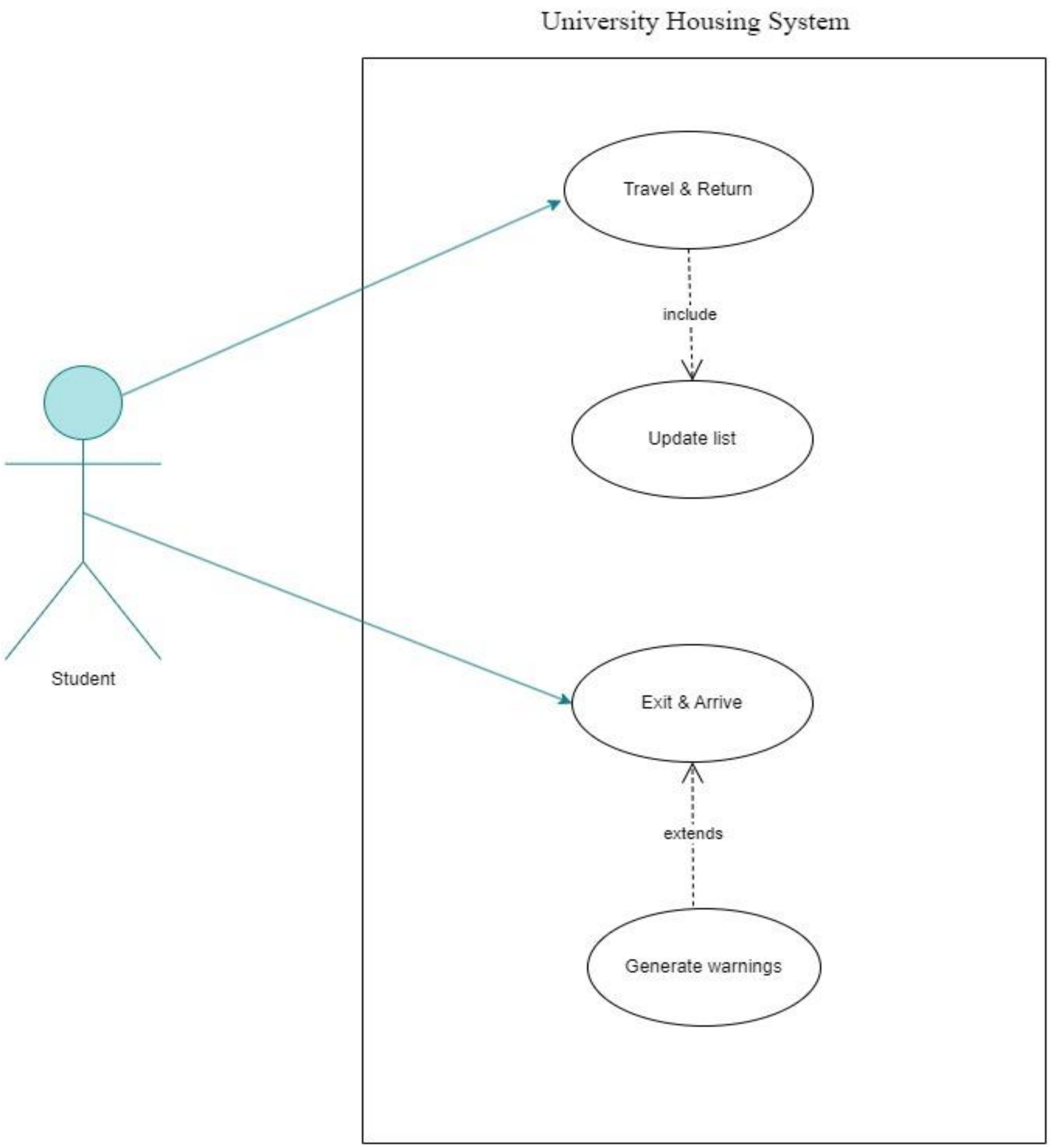
Integration with External Systems: If applicable, the website should be able to integrate with external systems or services, such as university databases or payment gateways.

Use case diagram:

Part 1


University Housing System
Sign up
extends
Log in
Book as new student
include
Payment
include
Generate Code
include
Renew Booking
Student
Bank System

Part 2



Use case scenario:

Use case Name: Sign up

Unique ID: UC1

Actors : Student

Importance level: High

Description: this use case describes the process of a new student registration for university housing.

Precondition:

1. University housing system is operational and accessible to users.
2. The student doesn’t have an existing account in the system.
3. The student has a stable internet connection to access the online page.

Normal flow of event:

1. **Access Sign-up page**: the student navigates to the university housing system’s sign-up page.
2. **Enter credentials:** the student provide required information(SSN, Username, Password)
3. **Submit Registration:** The student submit registration form.
4. **Confirm message:** If the registration is successful, the display a confirmation message.

Exceptional flow:

1. **Invalid credentials:** If the entered credentials are invalid or username is already in use, the system prompts the student to enter valid information.

Post conditions:

1. A new account is created for the student.
2. The student will be able to access further features within the system.

Use case Name: Login	**Unique ID: UC2**

Actors : Student

Importance level: High

Description: This use case describes the process of a registered student logging into the university housing system using their credentials.

Precondition:

1. Housing system is operational and accessible to users.
2. The student attempts to log in has a registered account.
3. The student has a stable internet connection to access the online page.
4. The student must be logged in with valid credentials.

Normal flow of event:

1. **Access Login page:** the student navigates to the university housing system's Login page**.**
2. **Enter credentials:** the student his Username and Password.
3. **Authenticate Credentials**: the system verify the credentials against the stored information.
4. **Successful login:** if the credentials are valid, the student gains access to system.

Exceptional flow:

1. **Invalid credentials:** If the entered credentials are invalid or username is already in use, the system prompts the student to enter valid information**.**

Post conditions:

1. **Availability for system activates:** The student gains access to various functionalities, including booking, renewing booking, making payments.

Use case Name: Book as a new student

Unique ID: UC3

Actors : Student

Importance level: High

Description: This use case outline the process of a new student booking university housing.

Precondition:

1. The student has successfully signed up for the university housing system.
2. The student has logged into the system using their valid credentials.

Normal flow of event:

1. **Include - Login (UC-2):** The system includes the Login use case to ensure the student is authenticated.
2. **Access booking section**: The student navigates to the booking section within the university housing system and chooses to book as a new student.
3. **Enter Personal Information:**
   The student enters the following details:
   {Name, Faculty Name, Level, Social Security Number (SSN), Phone Number
   Parent's Phone Number, Date of Birth, Residence Details}
4. **Submit Booking Request:** After entering the required information, the student submits the booking request.
5. **Confirmation Message:** The system displays a confirmation message, instructing the student to check the site daily for the next 2 weeks for acceptance or refusal notifications**.**

Exceptional flow:

1. **Invalid Information:** If the entered information is incomplete or invalid, the system prompts the student to provide accurate details.

Post conditions:

1. The student's booking request is recorded in the system for further processing.

Use case Name: Renew booking	**Unique ID: UC4**
Actors : Student	
Importance level: High	
Description: this use case describes the process of a student renewing their booking for university housing.	
Precondition: 1. The student has successfully signed up for the university housing system. 2. The student has logged into the system using their valid credentials. 3. The student has a previous booking that needs to be renewed.	
Normal flow of event: 1. **Include - Login (UC-2):** The system includes the Login use case to ensure the student is authenticated. 2. **Access booking section**: The student navigates to the booking section within the university housing system and chooses to renew booking. 3. **Enter Student Code:** The student enters their assigned code to check for any warnings or status updates related to their previous bookings 4. **View Warnings:** The system displays any warnings or relevant information associated with the student's booking history. 5. **Renewal confirmation:** If there are no warnings or problems, the student confirms the renewal of the booking 6. **Confirmation Message:** The system displays a confirmation message indicating the successful renewal of the booking.	
Exceptional flow: **1.** **Warnings Exist:** If there are warnings related to the student’s reservation record (for example, late return), the system informs the student that he cannot renew the reservation because the number of warnings has been exceeded.	
Post conditions: 1. The student's booking is successfully renewed for the upcoming academic term. 2. The system updates the student's booking status and provides confirmation.	

Use case Name: Payment | **Unique ID: UC5**

Actors : Student

Importance level: High

Description: This use case outlines the process of a student making a payment for their university housing, including the option to choose meal preferences.

Precondition:

4. The student has successfully signed up for the university housing system.
5. The student has logged into the system using their valid credentials.
6. The student has a previous booking that needs to be renewed.

Normal flow of event:

1. **Include - Login (UC-2):** The system includes the Login use case to ensure the student is authenticated**.**
2. **Access Payment Section:** The student navigates to the payment section within the university housing system.
3. **Choose the meal option:** The student determines whether he wants the meal option or not.
4. **Proceed to Payment:** After selecting the meal option, the student proceeds to make the payment using a bank account card.
5. **Payment Confirmation:** The system confirms the payment and updates the student's booking status accordingly.
6. **Include - Code Generation for New Student (UC6):** If the payment is successful and the student is a new user, the system generates a unique code for the student.

Exceptional flow:

1. **Payment Failed:** If the payment fails (for example, insufficient funds), the system informs the student that the payment was unsuccessful and to try again.

Post conditions:

1. The system updates the student's booking status and meal preferences.
2. For new students, the system generates a unique code after successful payment.

Use case Name: Generate Code **Unique ID: UC6**

Actors : Payment Function

Importance level: High

Description: this use case describes the process of a generating student code in the university housing.

Precondition:

4. The student has logged in.
5. The payment process has been completed successfully.

Normal flow of event:

5. **Generate student code :** If a new student: generate a new code which consist of 6 digit first two digits for year of joining university and the remaining four digits are unique and the character and then merge code with character that expresses meal state ('M' for meal , ' P' for no meal),
   If a student renews reservation: code will be the old code with determining the meal state.
6. **Showing generated code:** The system confirms the successful generation of the code and display it on

Exceptional flow:

2. **Not logging in:** show message to user to log in first and show log in button.
3. **Nonpayment:** show message ask user to pay first and show payment button.

Post conditions: displaying the generated code in the code label.

Use case Name: Travel & Return	**Unique ID: UC7**
Actors : Student	
Importance level: High	
Description: this use case describes the function or recording students travel home and return to residence time.	
Precondition: 1. Student has a unique code. 2. Entering the student code correctly. 3. The student’s status is ‘In residence’ in Exit & Arrive function.	
Normal flow of event: **1. Click on the ‘Travel & Return’ button.** **2. Entering student code.** **3. Click on ‘submit code’ button:** when student enter click on this button the state of ‘Travel & Return’ its state is switched from the current state to the other state. **4. Showing the new state:** Show message with the student info, updated state, and the time and date of this update. **5. Click on ‘Exit’ button:** to move from this window to the home window.	
Exceptional flow: **1. Submit without entering code:** show message asking student to enter his code. **2. Invalid code:** show message asking student to enter the code again correctly. **3. The state is ‘Exit’ in ‘Exit & Arrive’ function:** show message asking the student to edit this state first and show ‘Exit & Arrive’ button. **4. Not click on ‘Exit’ button:** moving to the home automatically window after one minute.	
Post conditions: Editing student ‘Travel & Return state’.	

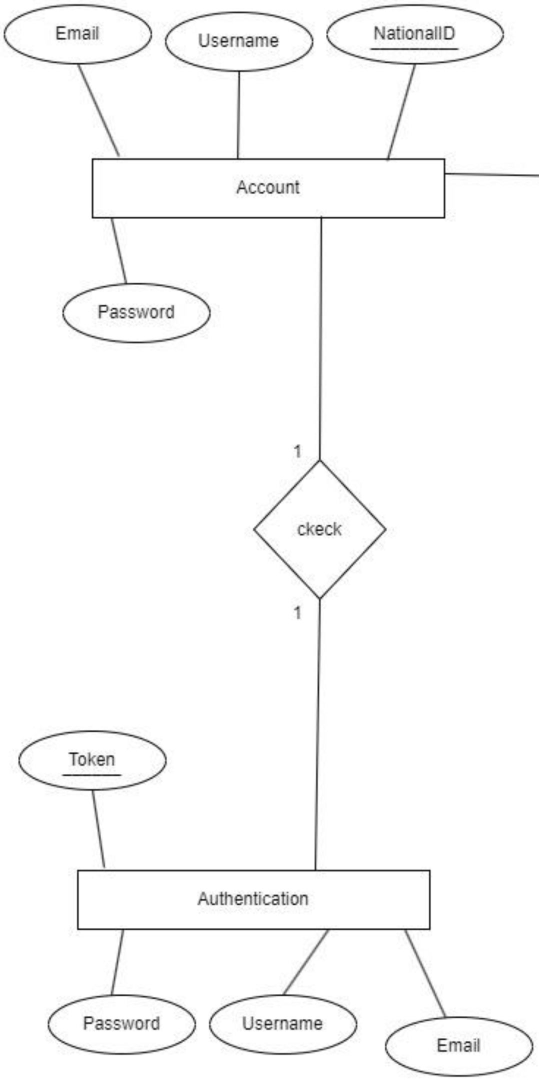
Use case Name: Update list	**Unique ID: UC8**
Actors : Travel & Return function	
Importance level: High	
Description: this use case describes the scenario of updating meal’s list by adding and removing students form it depending on ‘Travel & Return’ function.	
Precondition: 1. Updating Travel & Return status successfully.	
Normal flow of event: 1. **Checking code:** checking the first letter of the code, if it starts with ‘M’ so update meal list.	
Exceptional flow: **1. First letter of student code is ‘P’:** do nothing.	
Post conditions: Updating the list of number of students how will get meal and their code.	

Use case Name: Exit & Arrive	**Unique ID: UC9**
Actors : Student	
Importance level: High	
Description: this use case describes the flowing of ‘Exit and Arrive’ function	
Precondition: 1. Student has a unique code. 2. Entering the student code correctly. 3. The student’s status is ‘In residence’ in Travel & Return function.	
Normal flow of event: 1. **Click on the ‘Exit and Arrive’ button.** 2. **Entering student code.** 3. **Click on ‘submit code’ button:** when student enter click on this button the state of ‘Exit & Arrive’ its state is switched from the current state to the other state. 4. **Showing the new state:** Show message with the student info, updated state, and the time and date of this update. 5. **Click on ‘Exit’ button:** to move from this window to the home window**.**	
Exceptional flow: 1. **Submit without entering code:** show message asking student to enter his code**.** 2. **Invalid code:** show message asking student to enter the code again correctly**.** 3. **The state is ‘Travel’ in ‘Travel & Return’ function:** show message asking the student to edit this state first and show ‘Travel & Return**’** button**.** 4. **Not click on ‘Exit’ button:** moving to the home automatically window after one minute.	
Post conditions: update the ‘Exit and Arrive’ state.	

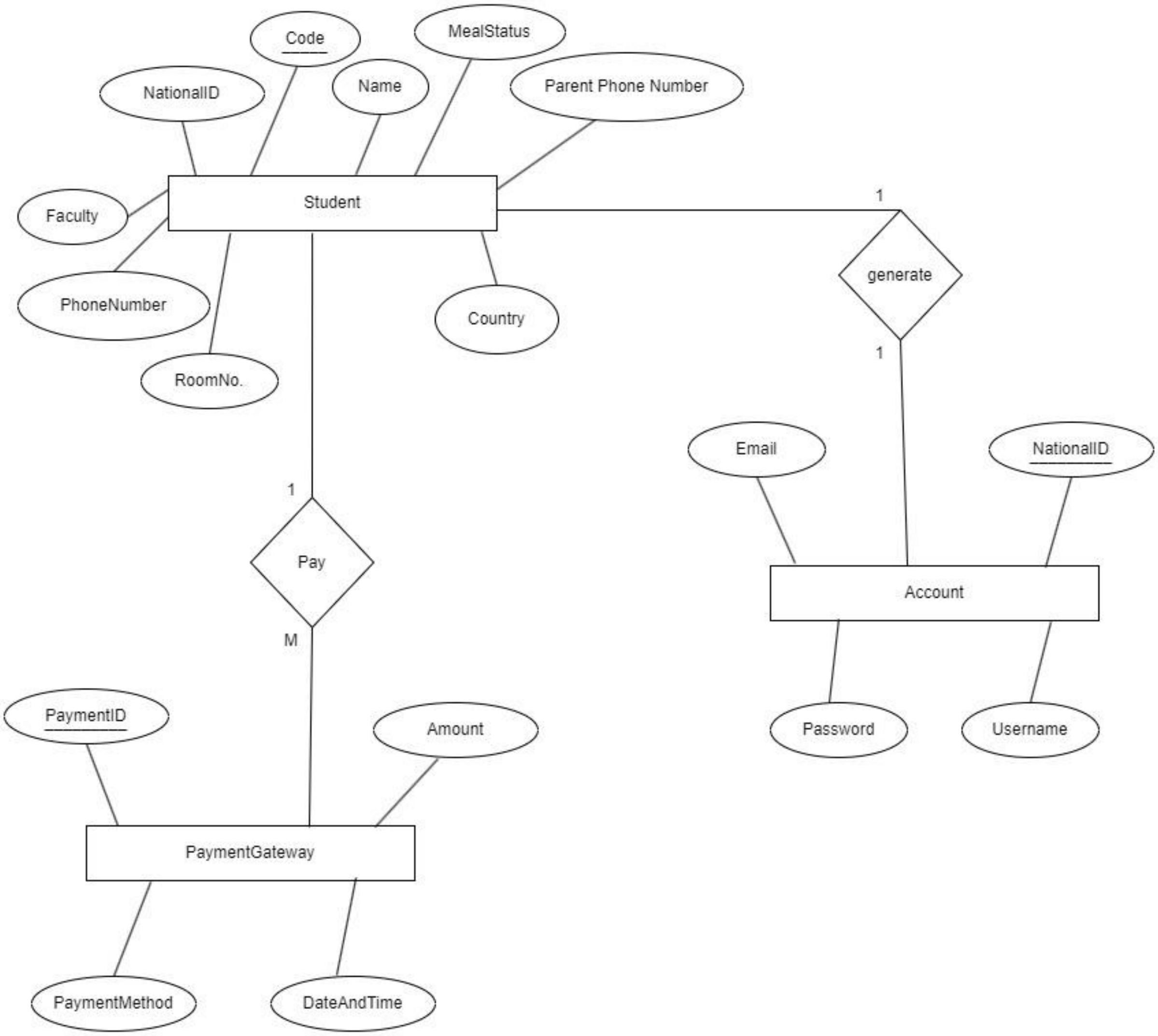
Use case Name: Generate Warnings	**Unique ID: UC10**
Actors : ‘Exit & Arrive’ function	
Importance level: High	
Description: this use case describes the process of generating warnings	
Precondition: 1. Arrive time exceeding the specified time at residence.	
Normal flow of event: **1. Increasing the number of times a student has been late by one.** **2. Showing warning message:** this message contains the date and time of arrive any number of previous delays**.**	
Exceptional flow: **1. Not exceeding the specified time:** do nothing.	
Post conditions: updating the number of warning for a student.	

ERD diagram:

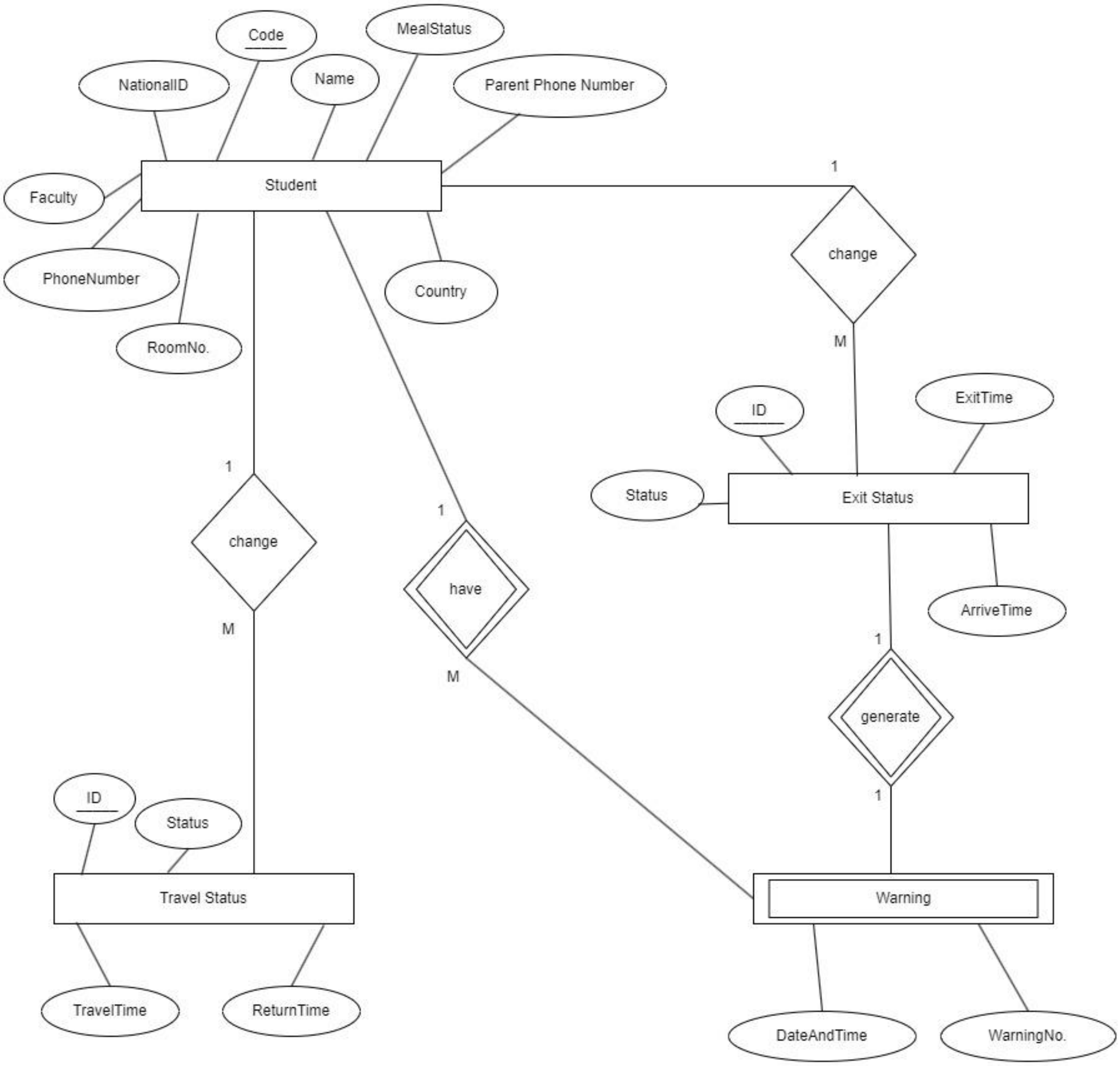
Part 1


Email
Username
NationalID
Account
Password
1
create
1
ckeck
1
1
Email
NationalID
Registration
Password
Username
Token
Authentication
Password
Username
Email

Part 2

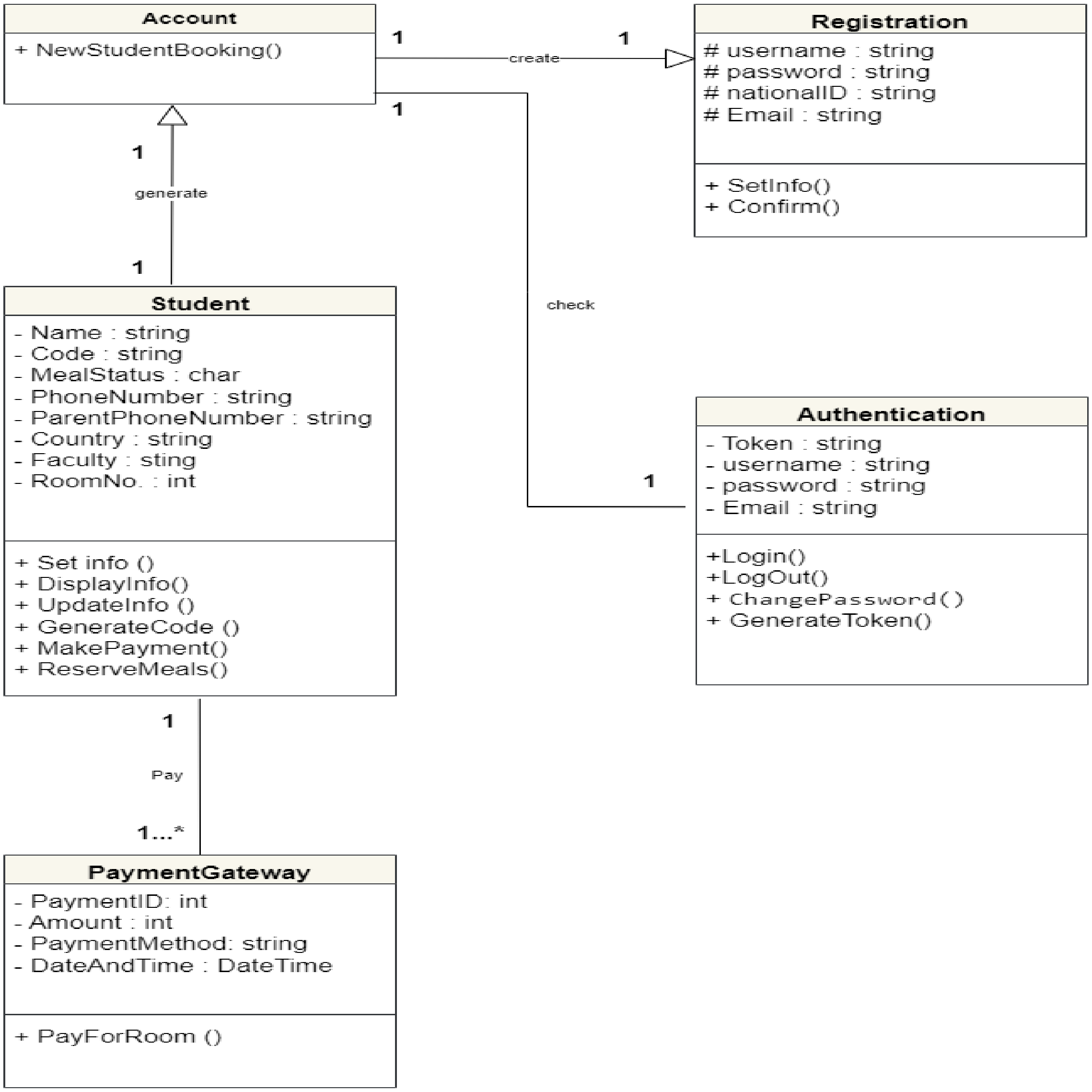

Code
MealStatus
NationalID
Name
Parent Phone Number
Faculty
Student
1
generate
PhoneNumber
Country
1
RoomNo.
Email
NationalID
1
Pay
Account
M
PaymentID
Amount
Password
Username
PaymentGateway
PaymentMethod
DateAndTime

Part 3



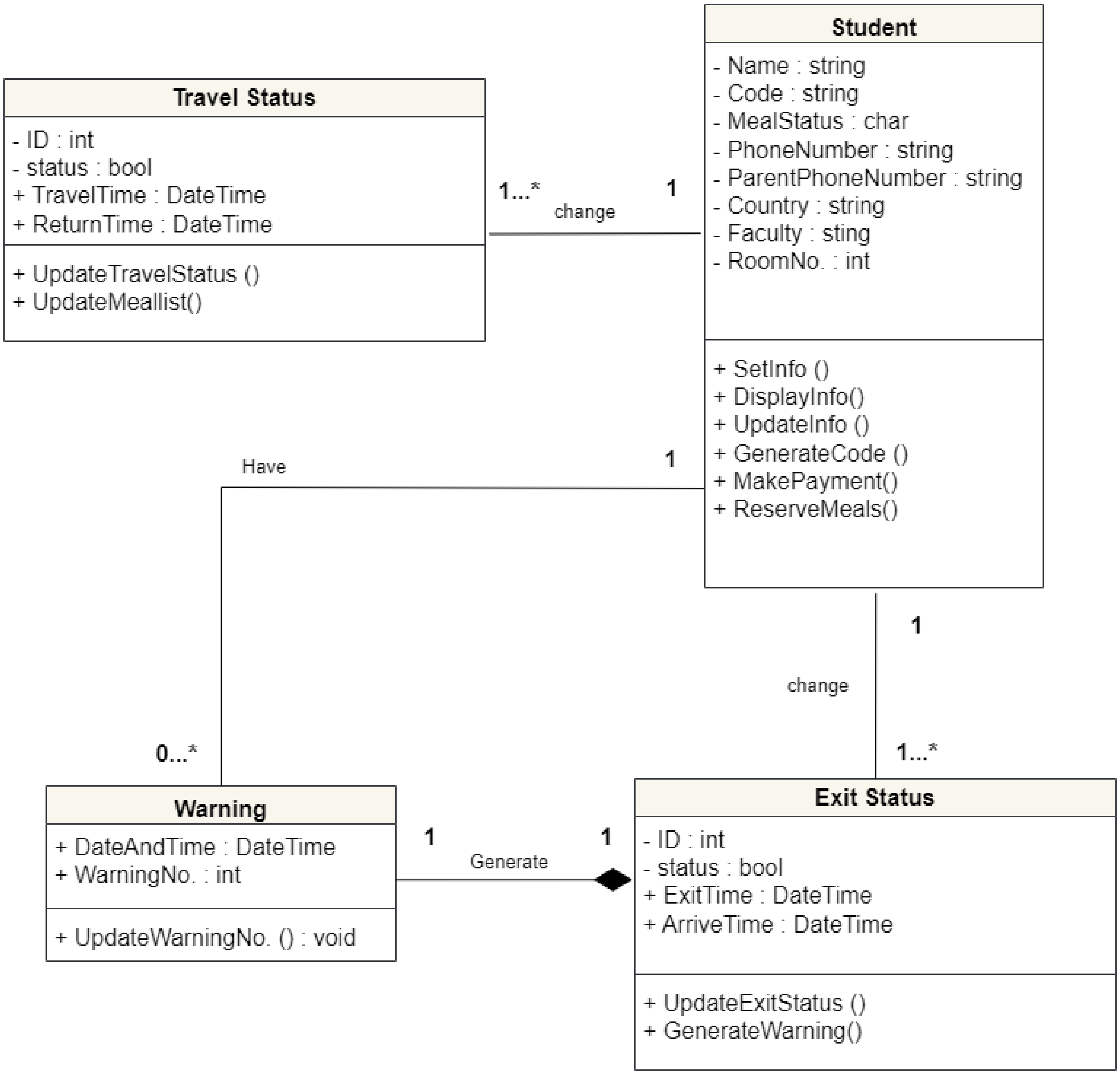
Class diagram:

Part 1



Account
+ NewStudentBooking()
1
create
1
Registration
# username : string
# password : string
# nationalID : string
# Email : string
+ SetInfo()
+ Confirm()
1
1
generate
1
check
Student
- Name : string
- Code : string
- MealStatus : char
- PhoneNumber : string
- ParentPhoneNumber : string
- Country : string
- Faculty : sting
- RoomNo. : int
+ Set info ()
+ DisplayInfo()
+ UpdateInfo ()
+ GenerateCode ()
+ MakePayment()
+ ReserveMeals()
1
Authentication
- Token : string
- username : string
- password : string
- Email : string
+Login()
+LogOut()
+ ChangePassword()
+ GenerateToken()
1
Pay
1...*
PaymentGateway
- PaymentID: int
- Amount : int
- PaymentMethod: string
- DateAndTime : DateTime
+ PayForRoom ()

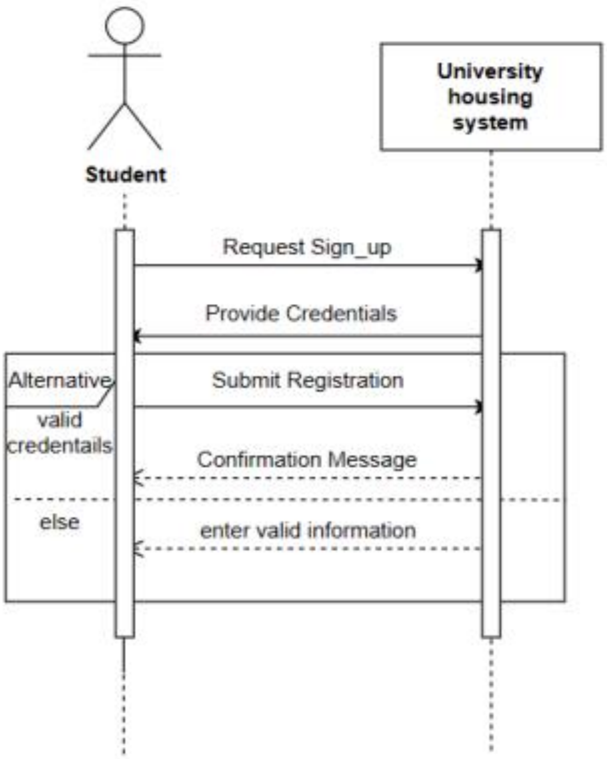
Part 2



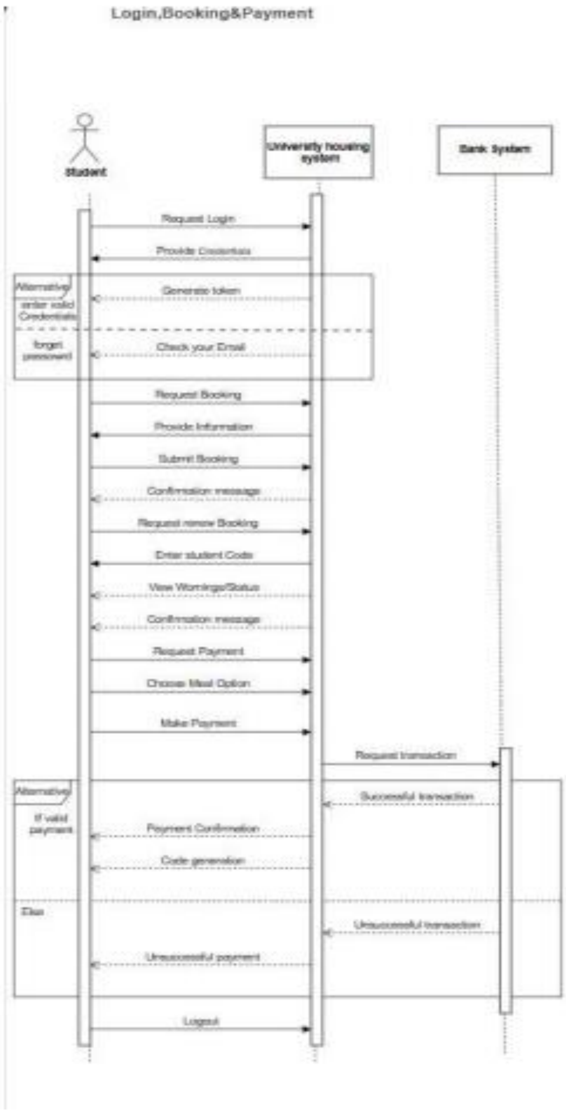
Sequence diagram:

Part 1

Sign_up

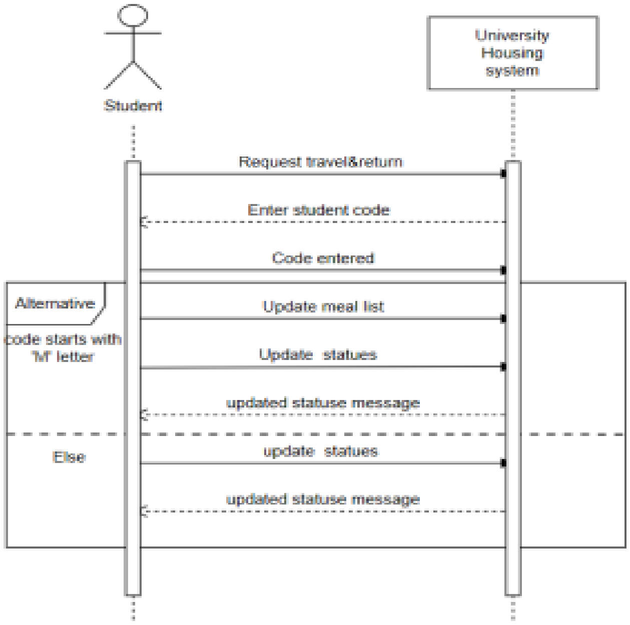


Part 2

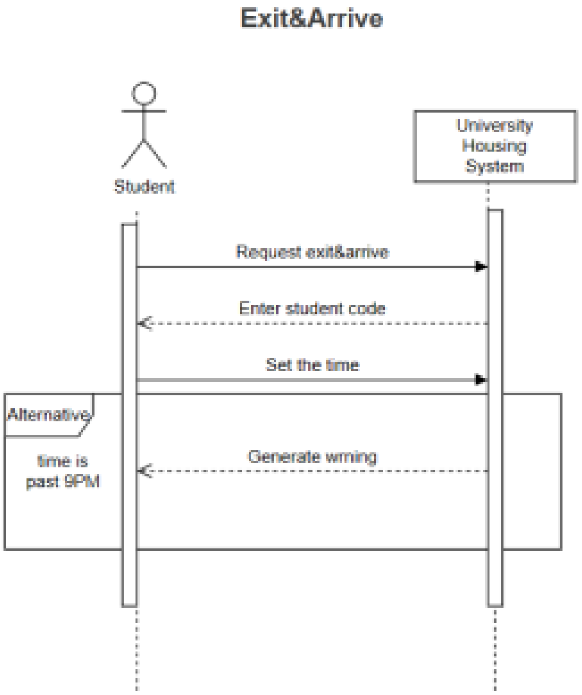

Login,Booking&Payment
Student
University housing system
Bank System
Request Login
Provide Credentials
Alternative
enter valid Credentials
Generate token
forget password
Check your Email
Request Booking
Provide Information
Submit Booking
Confirmation message
Request renew Booking
Enter student Code
View Warnings/Status
Confirmation message
Request Payment
Choose Meal Option
Make Payment
Request transaction
Alternative
If valid payment
Successful transaction
Payment Confirmation
Code generation
Else
Unsuccessful transaction
Unsuccessful payment
Logout

Part 3

Travel&return



Part 4


Exit&Arrive
Student
University Housing System
Request exit&arrive
Enter student code
Set the time
Alternative
time is past 9PM
Generate wrning