

Leader Josh Wilson

Members Josh Wilson, Brad Buckingham,
Lanshi Li, Bailey Hollman, Nick Cook

Project Vision We aim to create a tool for schedule management in the workplace which will allow for managers to automate the creation of employee schedules by processing employee schedule requests (desired days, hours, etc.) in order to optimize managerial workflow.

Use Case: Request Schedule Change
ID: UC06
Actors: Employee
Preconditions: 1. The Employee is logged into the system
Flow of Events: 1. Employee selects "Request Schedule Change" 2. System opens dialog for requested change 3. Employee enters desired date/time they wish to change 4. Employee enters the reason for the schedule to be changed 5. System sends generated request to Manager for approval 6. System forwards "Schedule Change Request" notification to email in database
Postconditions: 1. Schedule pending approval from Manager
Alternative Flow: 1. At any point the dialog may be closed without clicking to confirm, which discards any inputted information and does not create a new request.

Use Case: Approve Request
ID: UC08
Actors: Manager
Preconditions: 1. The manager is logged into the system
Flow of Events: 1. Manager receives a request from the System 2. Manager either confirms or rejects the request.
Postconditions: 1. If the request was approved, the schedule is updated
Alternative Flow: 1. At any point the dialog may be closed, which discards any inputted information and does not update a schedule
Postconditions: If the request was rejected, an email is forwarded to the Employee's email provided in the System database and the request is removed

Use Case: Print Employee Schedule
ID: UC09
Actors: Employee
Preconditions: 1. The Employee is logged into the system
Flow of Events: 1. The Employee selects "Print my schedule" 2. The Employee indicates what schedule they wish to print 3. The System finds the indicated schedule and extracts only the data pertaining to the Employee 4. The System returns the dates and times the Employee is designated to work.
Postconditions: 1. The Employee's Schedule is displayed on the screen.
Alternative Flow: 1. At any point the dialog may be closed, which discards any inputted information and does not show the Employee's Schedule

Use Case: Add Employee
ID: UC01
Actors: Manager
Preconditions: 1. The manager is logged into the system on an administrator account
Flow of Events: 1. Manager selects "Register New Employee" 2. System opens dialog for employee information 3. Manager enters requested information about employee 4. System adds employee to employee database
Postconditions: 1. New employee added to database
Alternative Flow: 1. At any point the dialog may be closed, which discards any inputted information and does not create a new employee in the database.
Postconditions:

Use Case: Remove Employee
ID: UC02
Actors: Manager
Preconditions: 1. The manager is logged into the system on an administrator account
Flow of Events: 1. Manager selects the "Remove Employee" option while viewing the employee in the employee list 2. System opens confirmation dialog 3. Manager confirms removal of employee 4. System removes employee/employee's information from database
Postconditions: 1. Selected employee removed from database
Alternative Flow: 1. The confirmation dialog may be closed, which cancels removal of the employee.
Postconditions:

Use Case: Update Employee
ID: UC03
Actors: Manager
Preconditions: <ol style="list-style-type: none"> 1. The manager is logged into the system on an administrator account. 2. The manager already has the Employee data. 3. The employee is already registered to the system.
Flow of Events: <ol style="list-style-type: none"> 1. Manager selects "Update Employee" from database 2. System opens dialog for employee information 3. Manager enters requested information about employee 4. System adds employee to employee database
Postconditions: <ol style="list-style-type: none"> 1. Employee data is updated in database.
Alternative Flow: <ol style="list-style-type: none"> 1. At any point the dialog may be closed, which discards any inputted information and does not create a new employee in the database.
Postconditions:

Use Case: Create 9-5 schedule for Monday, Wednesday, and Friday
ID: UC04
Actors: Manager
Preconditions: 1. The manager is logged into the system on an administrator account
Flow of Events: <ol style="list-style-type: none"> 1. Manager selects "Schedules" from home menu 2. System opens schedules screen 3. Manager selects "Create New Schedule" 4. System opens name choice dialog 5. Manager enters a name for the schedule and clicks "Select" 6. System closes name choice dialog and opens schedule creation dialog with options/info 7. Manager clicks "Time Details" on Monday 8. System opens time details dialog for Monday 9. Manager enters 9:00 AM for start time and 5:00 PM for end time, then clicks "Apply to All" 10. System returns to creation dialog and updates all days to have set times 11. Manager removes Tuesday and Thursday from day selection and clicks "Create" 12. System creates a new schedule with the name, days, and times chosen by the manager, and stores it in the schedule database.
Postconditions: <ol style="list-style-type: none"> 1. New schedule generated and added to list of schedules for manager's department
Alternative Flow: <ol style="list-style-type: none"> 1. Manager can exit dialog without clicking to confirm, which cancels generation of the schedule
Postconditions:

Use Case: List Employee
ID: UC04
Actors: Manager
Preconditions: 1. The manager is logged into the system on an administrator account
Flow of Events: 1. Manager selects "List Employee's" 2. System displays list of all employee names in the database.
Postconditions: 1. All Employee's from the data base are displayed to the screen.
Alternative Flow: 1. If no Employees are listed in the data base an error message will be displayed.
Postconditions:

Use Case: Generate Schedule
ID: UC05
Actors: Manager
Preconditions: 1. The manager is logged into the system on an administrator account
Flow of Events: 1. Manager selects "Schedules" from main menu 2. System opens schedules screen 3. Manager selects "Generate New Schedule" 4. System opens a dialog with options/info 5. Manager inputs desired options and constraints and clicks confirm 6. System generates a new schedule based off of employee info and manager's constraints
Postconditions: 1. New schedule generated and added to list of schedules
Alternative Flow: 1. Manager can exit dialog without clicking to confirm, which cancels generation of the schedule
Postconditions:

Use Case: Account/Log-in
ID: UCD05
Actors: Employee/Manager
Preconditions: <ol style="list-style-type: none"> 1. The user is at the login page
Flow of Events: <ol style="list-style-type: none"> 1. The user inputs username and password 2. The user selects enter 3. System authenticates login
Postconditions: <ol style="list-style-type: none"> 1. Pending System access verification.
Alternative Flow: <ol style="list-style-type: none"> 1. At any point the dialog may be closed without clicking to confirm, which discards any inputted information and does not create a new request.
Alternative Flow: <ol style="list-style-type: none"> 1. The user selects "Change/Forgot password" 2. System prompts for new password 3. The Inputs new password 4. System updates database 5. Continue at step 1 of normal flow

Use Case: PTO Request
ID: UC07
Actors: Employee
Preconditions: 1. The employee is logged into the system.
Flow of Events: 1. Employee selects "PTO Request" from main menu 2. System opens PTO Request Screen 3. Employee inputs desired date/time for request. 4. System sends generated request to Manager for approval. 5. System forwards "PTO Request" notification to email in database.
Postconditions: 1. Pending manager approval
Alternative Flow: 1. Employee can exit dialog without clicking to confirm, which cancels generation of the PTO Request.
Postconditions:

Use Case: Update Holiday Hours
ID: UC12
Actors: Manager
Preconditions: 1. The manager is logged into the system on an administrator account
Flow of Events: 1. Manager select "Update Hollyday Hours" 2. System opens dialog for requested change 3. Manager enters desired date/time they wish to update 4. System generate new schedule and print out, request for confirmation. 5. System forwards "Holyday Hours Update" notification to employees.
Postconditions: 1. Schedule updated.
Alternative Flow: 1. At any point the dialog may be closed without clicking to confirm, which discards any inputted information and does not create a new request.

Use Case: Request Shift Swap
ID: UC11
Actors: Employee
Preconditions: 1. The employee is logged into the system
Flow of Events: 1. User(A) choose a target time registered by another employee(B) 2. Pop out a dialog to request a confirmation for the operation and ask for the reason. 3. If user confirm this operation. Sent a request to manager account. 4. If manager approve the request, sent a request to employee(B). 5. If employee(B) approves the request, sent message to user(A) and manager.
Postconditions: 1. Update the new schedule..
Alternative Flow: 1. At any point the dialog may be closed, which discards any inputted information and do nothing. 2. If the request wasn't approved by manager, do nothing and sent a message to User(A). 3. If the request wasn't approved by employee(B), do nothing and sent a message to manager and User(A).
Postconditions: Do nothing.

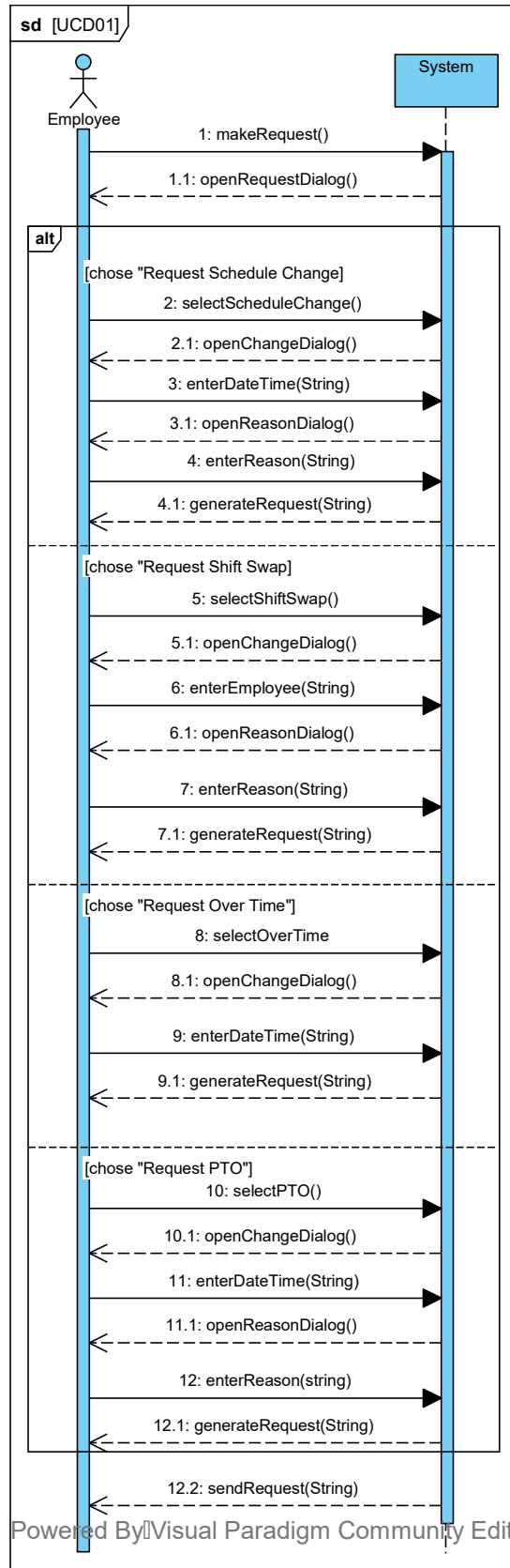
Use Case: Print Full Schedule
ID: UC09
Actors: Manager
Preconditions: 1. The manager is logged into the system on an administrator account
Flow of Events: 1. The Manager selects "Print Full schedule"
Postconditions: 1. The Full Schedule is displayed on the screen.
Alternative Flow:

Use Case: Request Overtime
ID: UC13
Actors: Employee
Preconditions: 1. The employee is logged into an account in the system
Flow of Events: 1. Employee selects "Request Overtime" 2. System opens interface for overtime request 3. Employee enters dates and hours planned to work overtime 4. Employee enters reason for overtime 5. System generates request for manager approval 6. System notifies user for request to email in database
Postconditions: 1. Pending approval request in Manager inbox
Alternative Flow: 1. User cancels overtime request

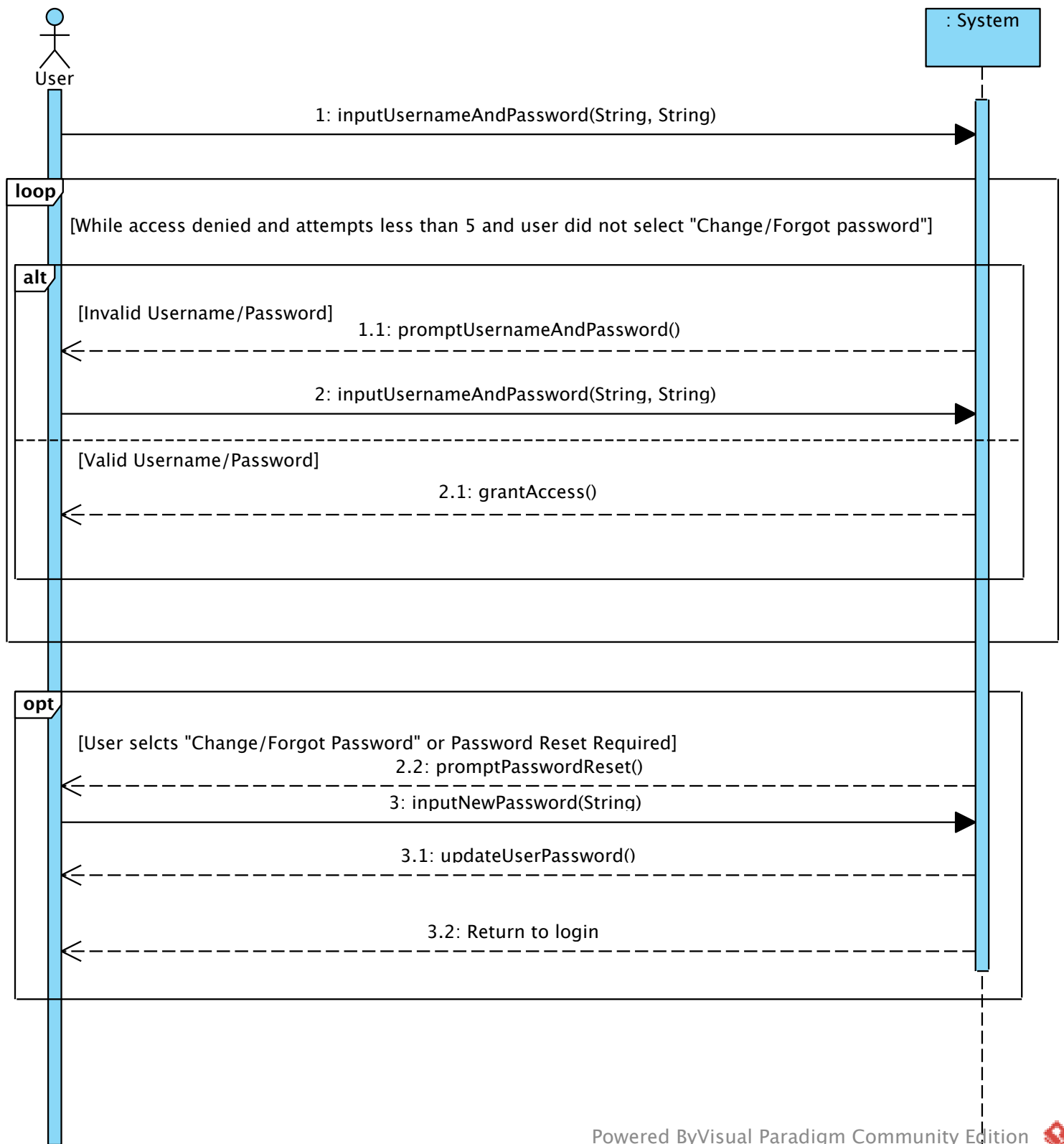
Use Case: Login
ID: UC14
Actors: Manager, Employee
Preconditions: 1. The employee or manager has an account in the system
Flow of Events: 1. User (Employee or Manager) enters username 2. User enters password 3. System authenticates the user credentials 4. User is granted access
Postconditions: 1. User is logged in and given permissions based on the account permissions
Alternative Flow: 1. Login is able to be canceled at any time 2. If login fails more than 5 times, system locks account
Postconditions: 1. Account is locked until manager unlocks account

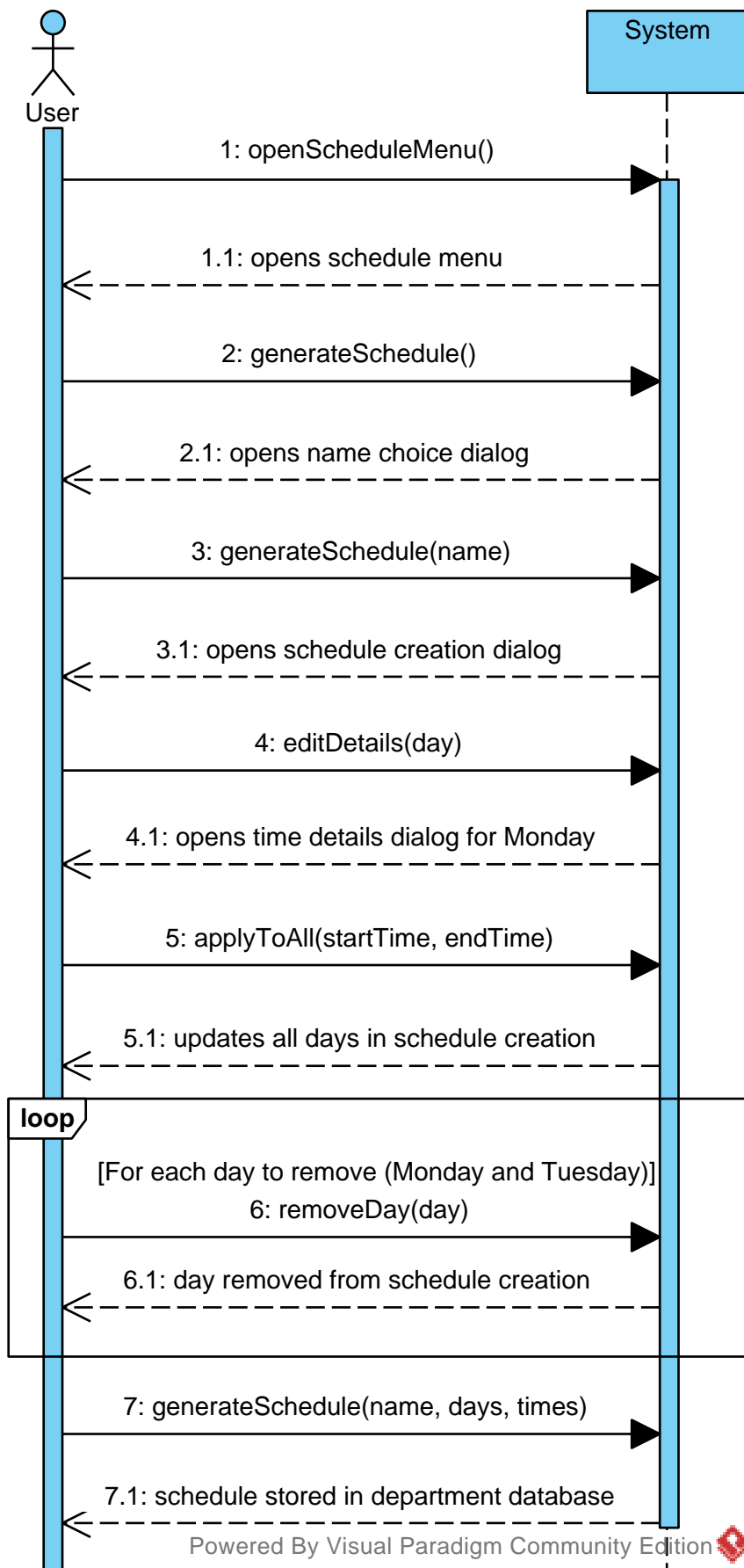
Use Case: Logout
ID: UC15
Actors: Manager, Employee
Preconditions: 1. The employee or manager is logged into an account in the system
Flow of Events: 1. User selects logout button 2. System checks for confirmed logout 3. User confirms logout 4. System returns to login page
Postconditions: 1. System logs user out of all open pages
Alternative Flow: 1. User cancels logout
Postconditions: 1. User is still logged into all pages

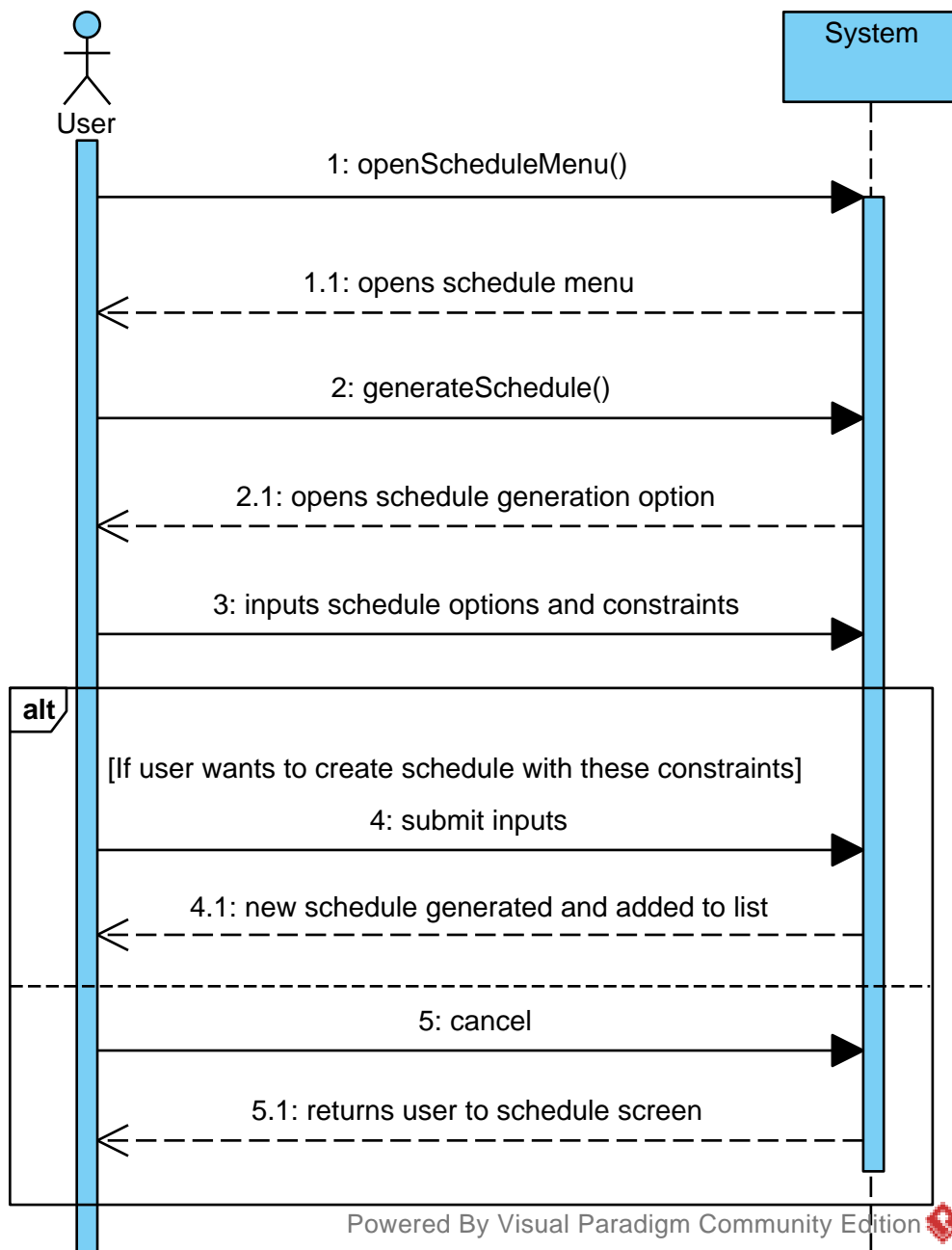
Use Case: Make Request
ID: UCD01
Actors: Employee
Preconditions: 1. The Employee is logged into the system
Flow of Events: 1. The Employee selects "Make Request" 2. A dialog box opens prompting the Employee to choose what type of request to make 3. The Employee selects "Request Schedule Change" 4. Employee enters desired date/time they wish to change 5. Employee enters the reason for the schedule to be changed 6. System sends generated request to Manager for approval 7. System forwards "Schedule Change Request" notification to email in database
Postconditions: 1. Schedule pending approval from Manager
Alternative Flow: 1. At any point the dialog may be closed without clicking to confirm, which discards any inputted information and does not create a new request.
Alternative Flow: 1. The Employee selects "Request Shift Swap" instead of "Request Schedule Change" 2. Employee enters which Employee they wish to swap schedules with 3. Employee enters the reason for the schedules to be swapped 4. Continue at step 5 of normal flow
Alternative Flow: 1. The Employee selects "Request Over Time" instead of "Request Schedule Change" 2. Employee enters the desired date/time they wish to work overtime 3. Continue at step 5 of normal flow
Alternative Flow: 1. The Employee selects "Request PTO" instead of "Request Schedule Change" 2. Employee enter the desired date/time they wish to take off 3. Employee enters the reason they wish to take PTO 4. Continue at step 5 of normal flow

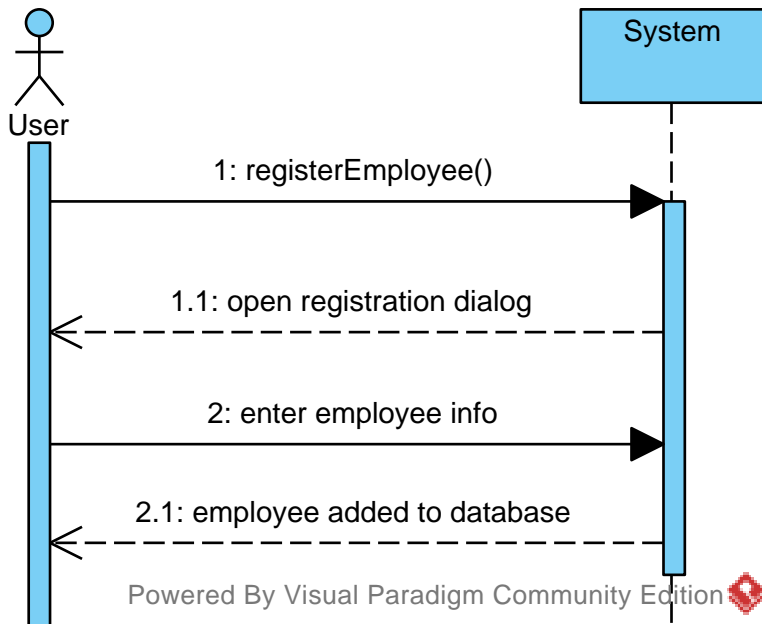


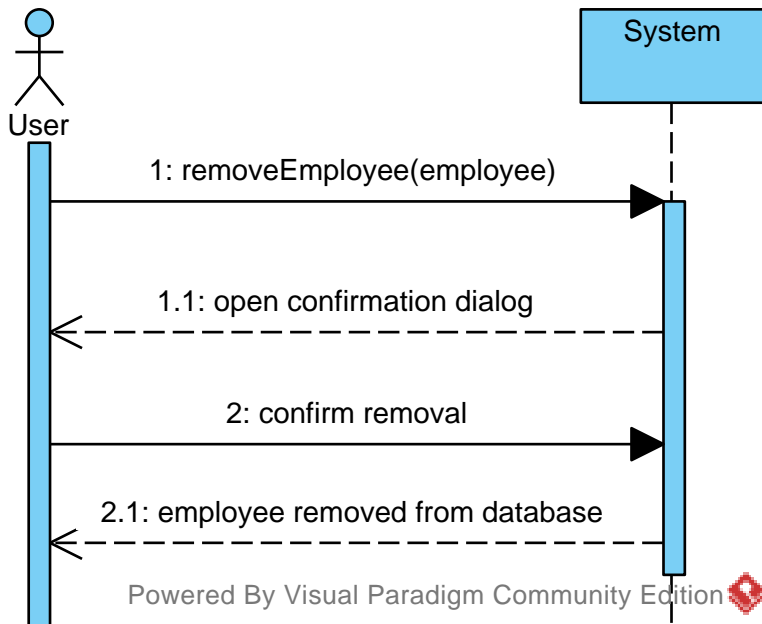
sd [UC5 SSD]

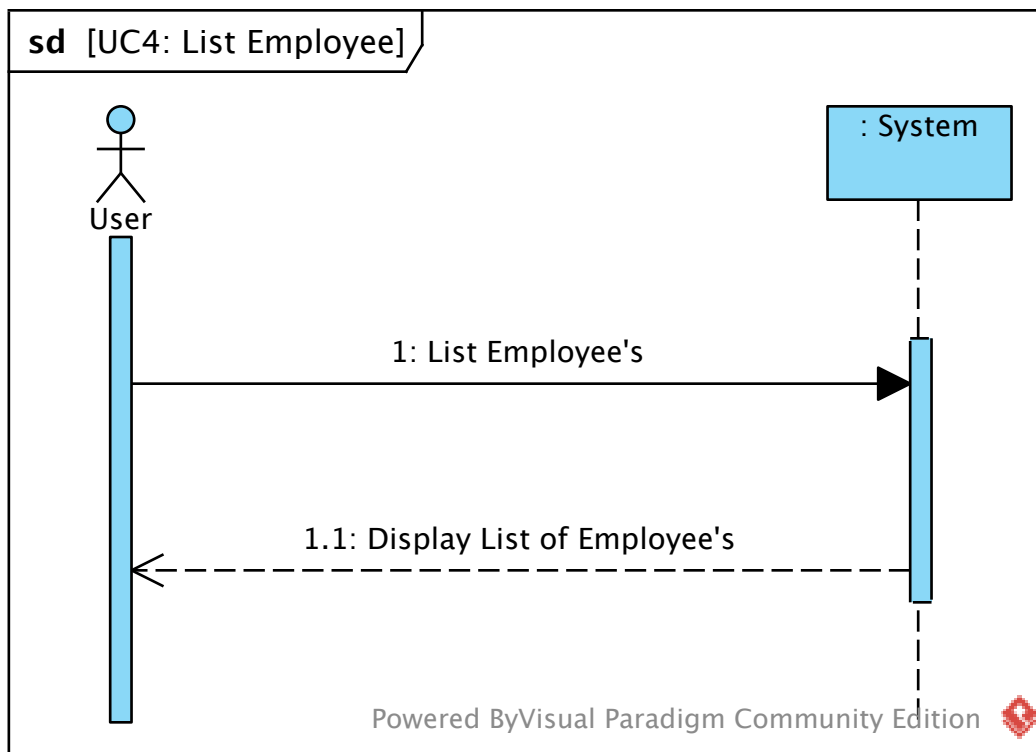




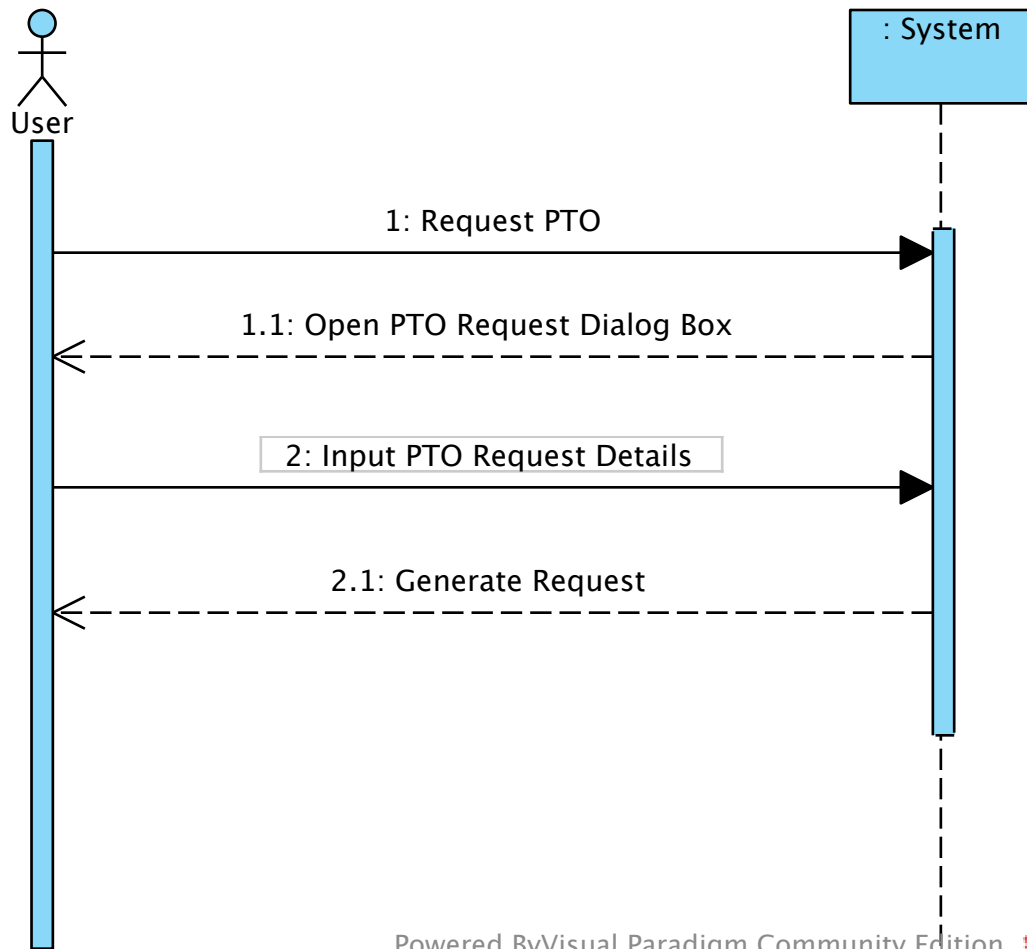




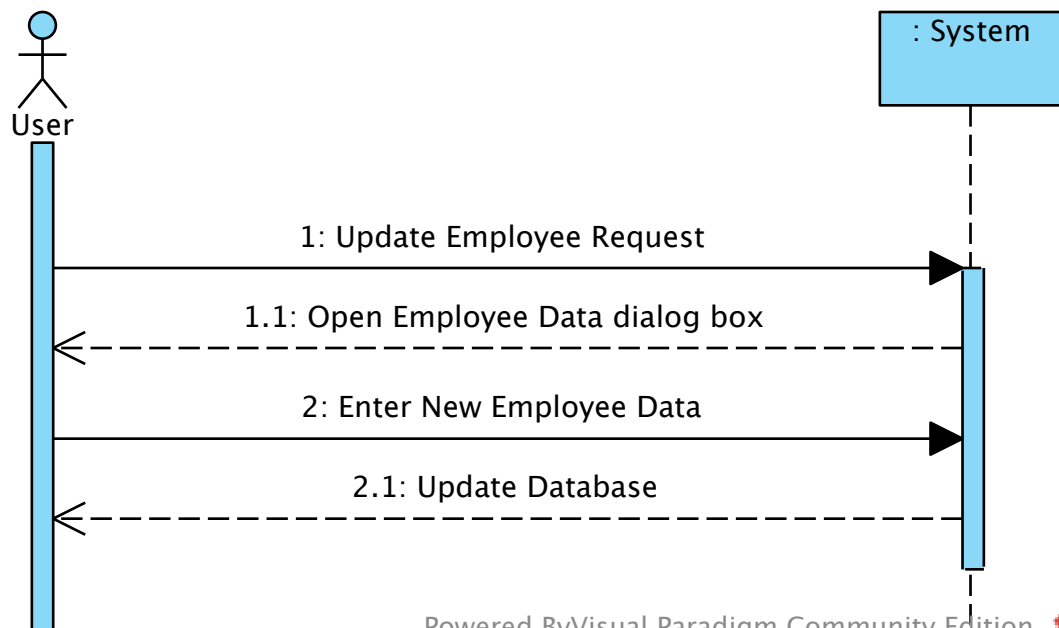




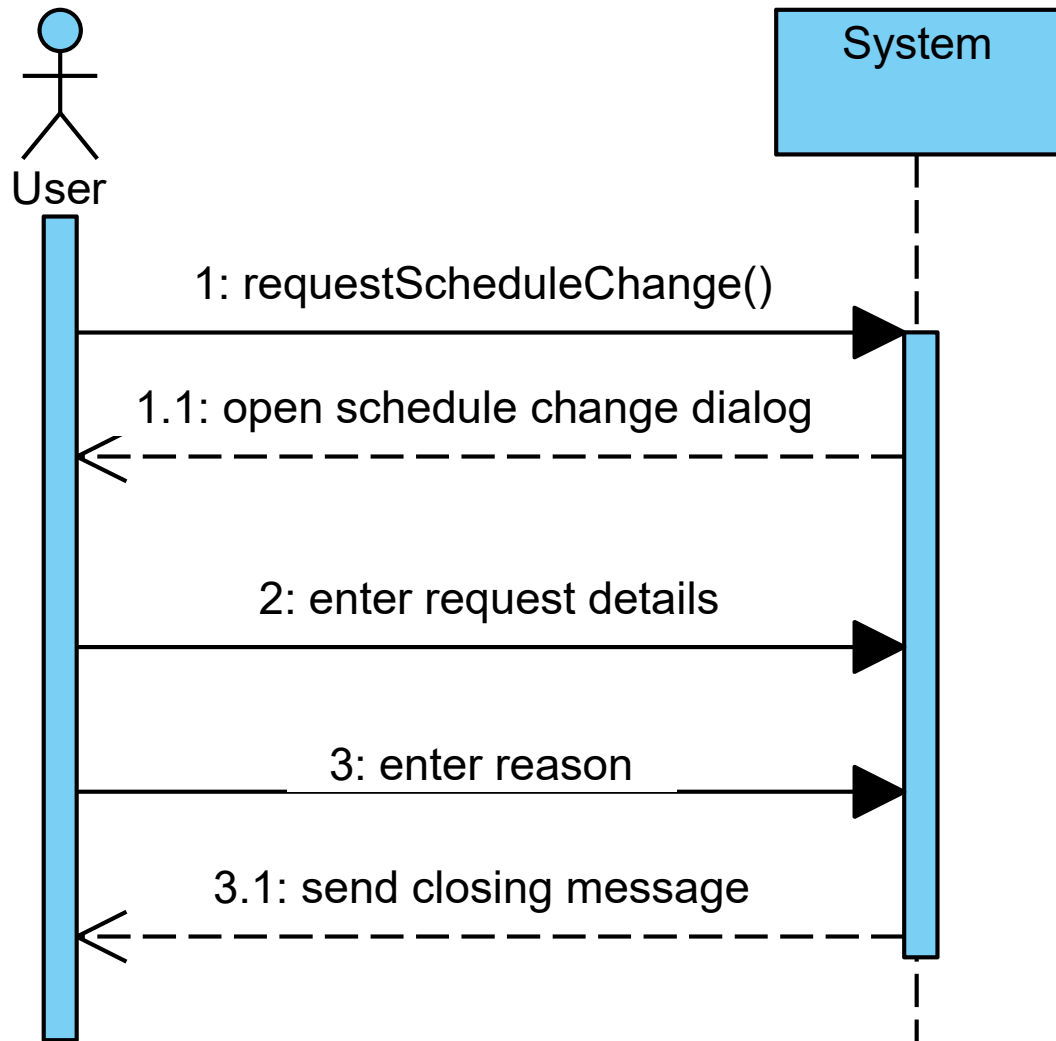
sd [UC7: PTO Request]



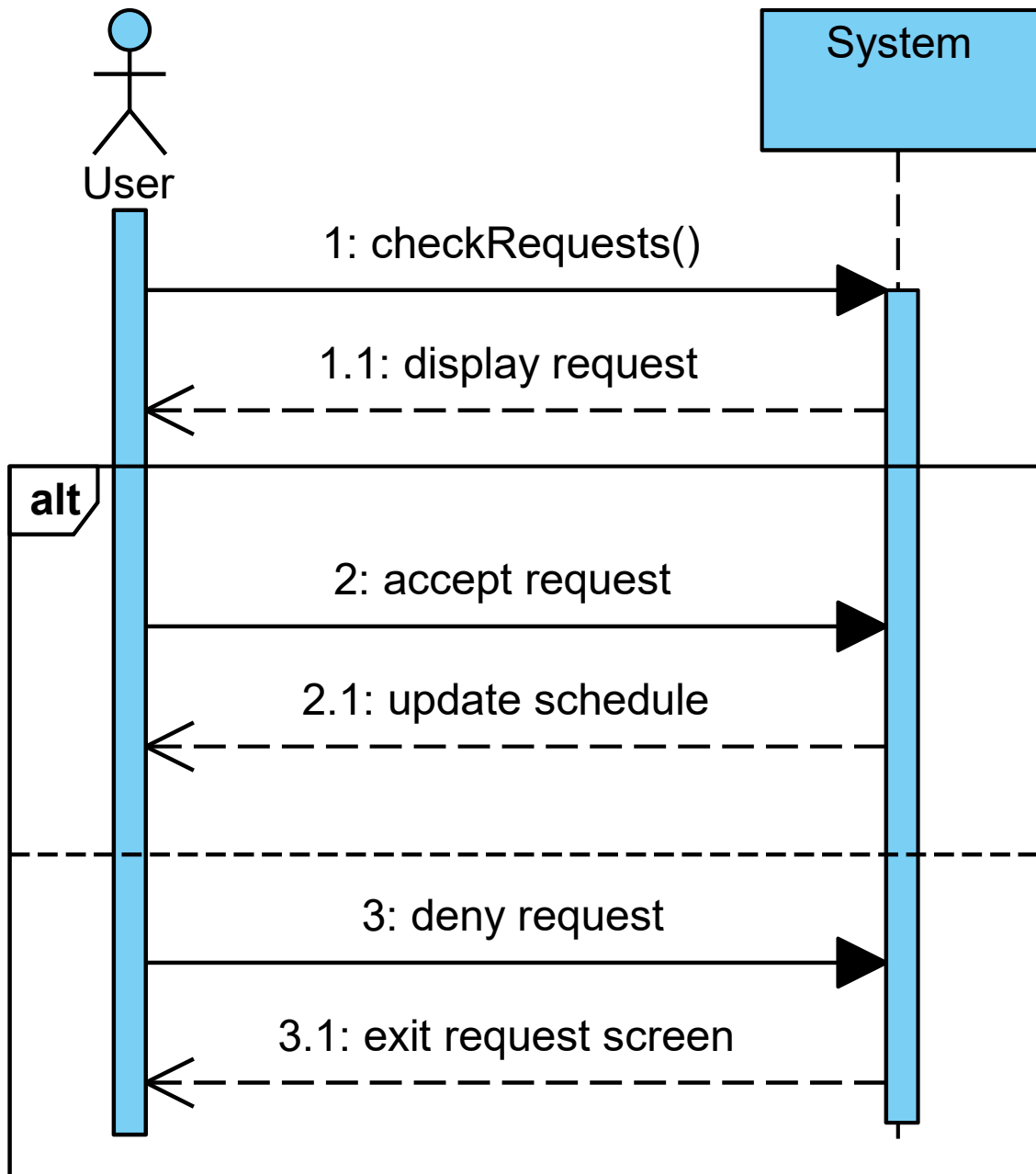
sd [UC3: Update Employee Request]



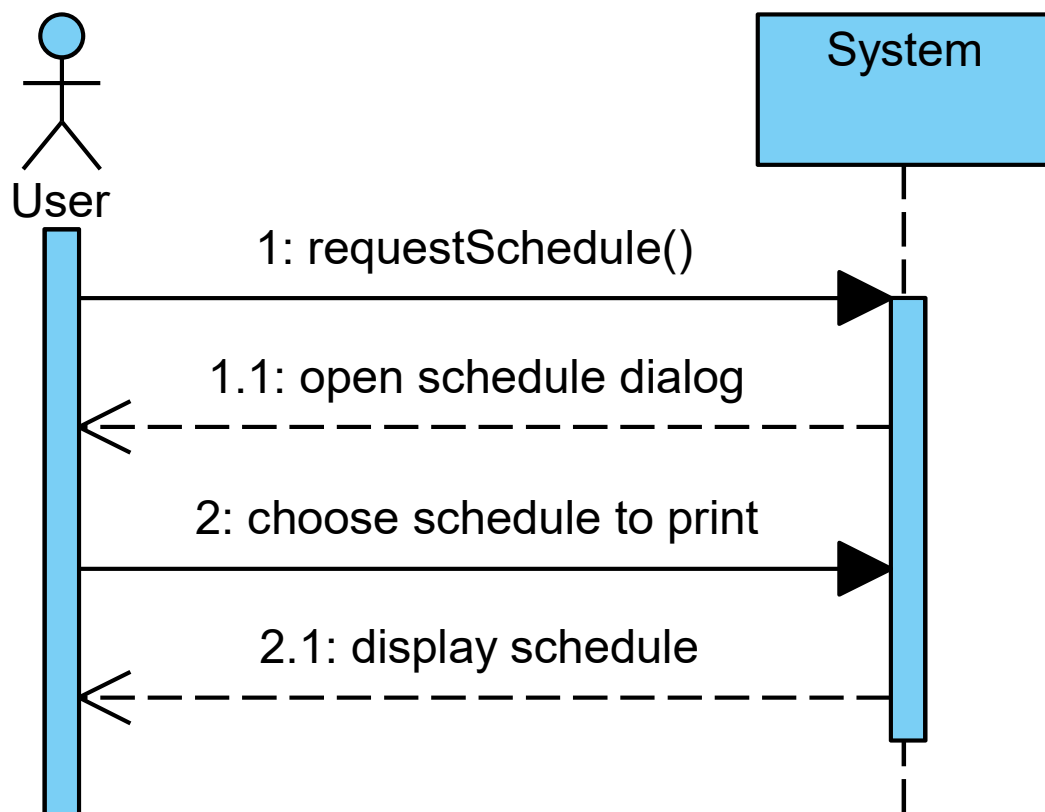
sd [UC06SSD]



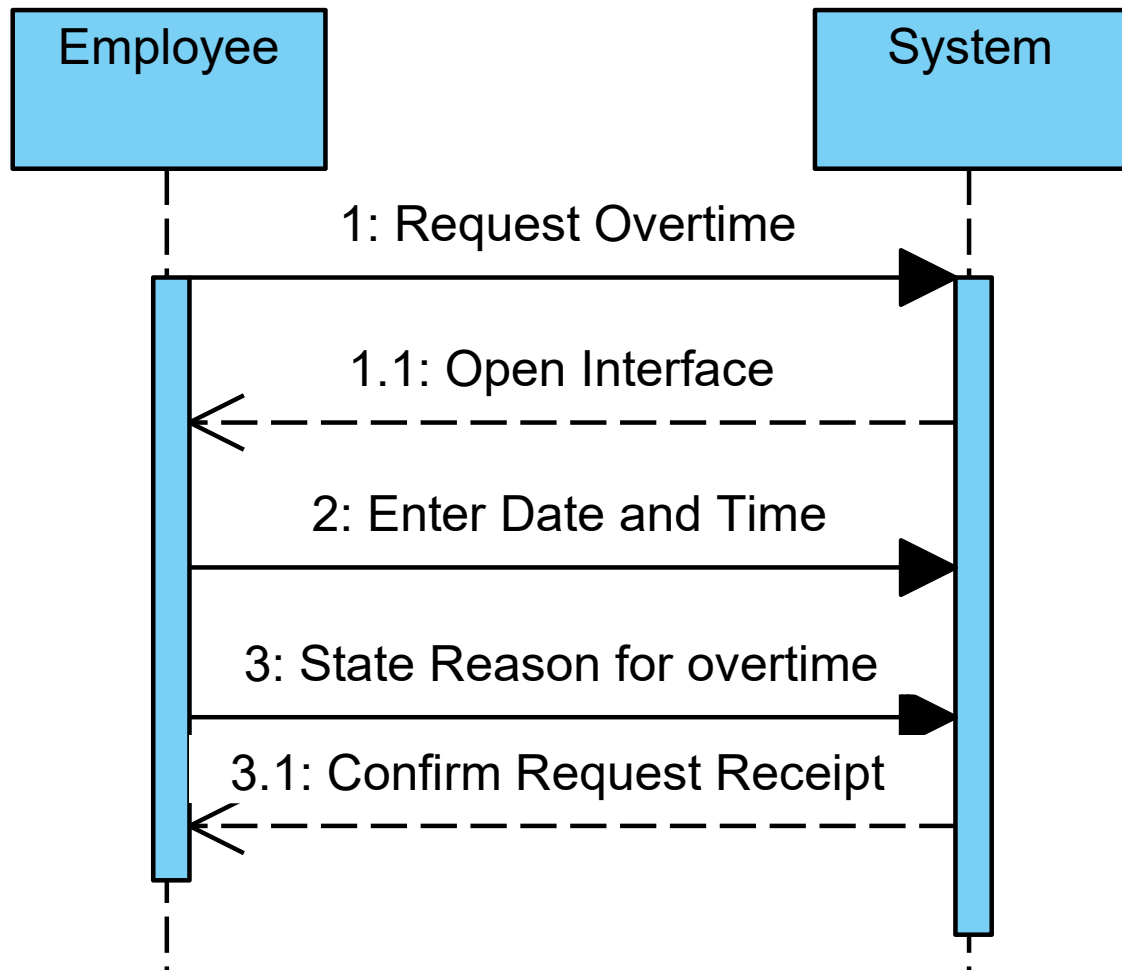
sd [UC07]



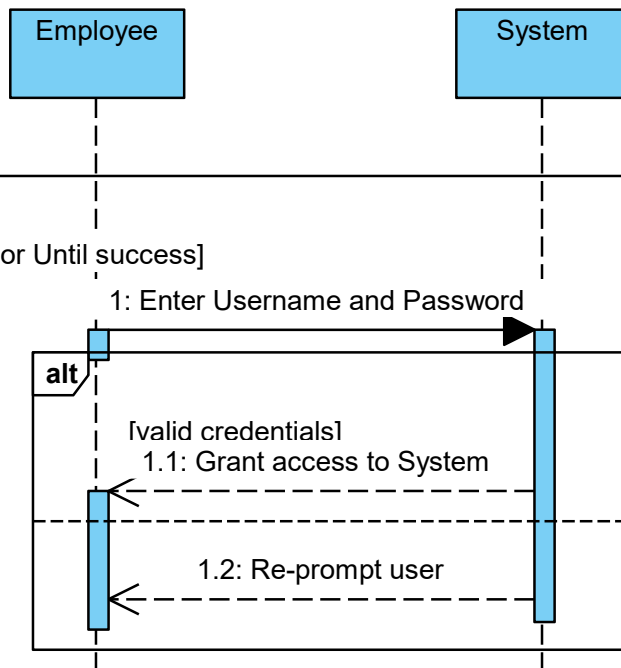
sd [UC09]



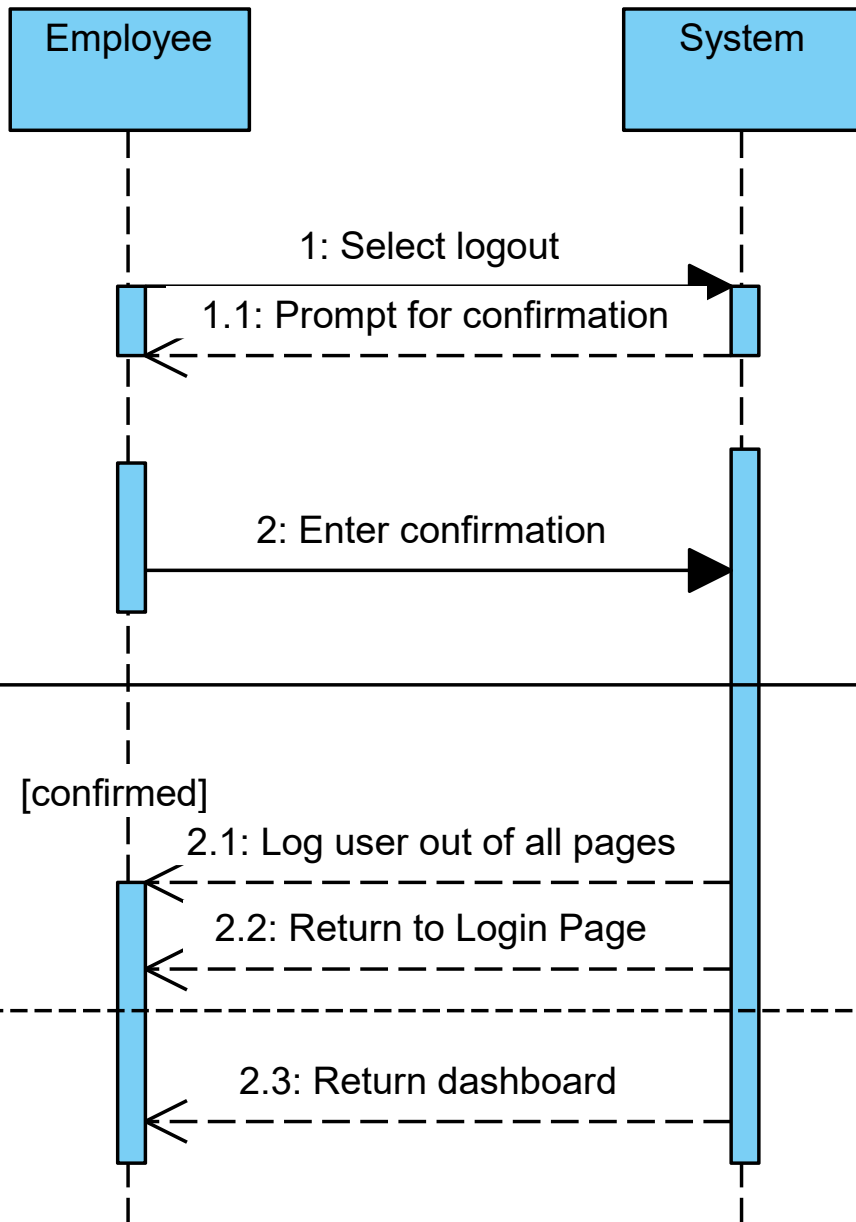
sd [UC13 SSD]

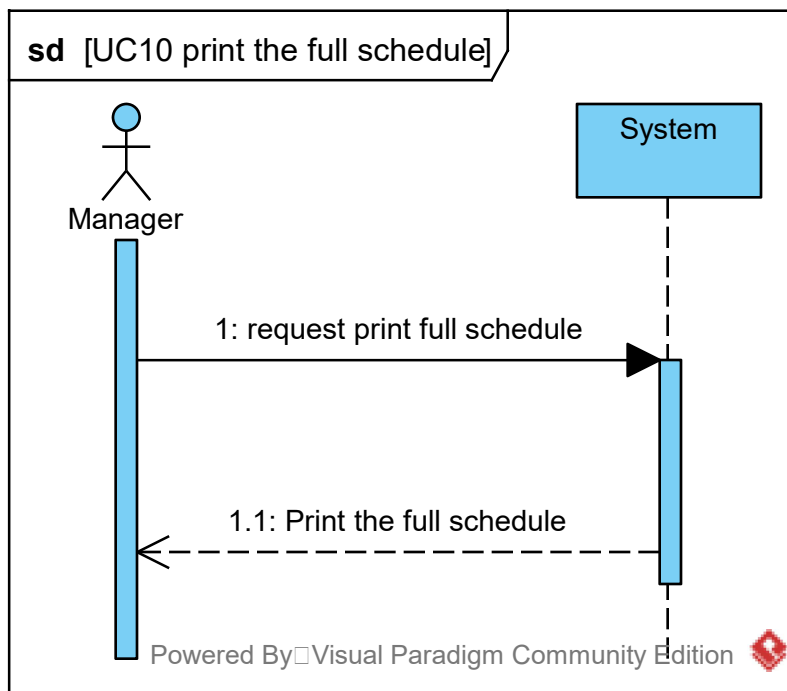


sd [UC14 SSD]



sd [UC15 SSD]





sd [UC12 Update Holliday Hours]

User

: System

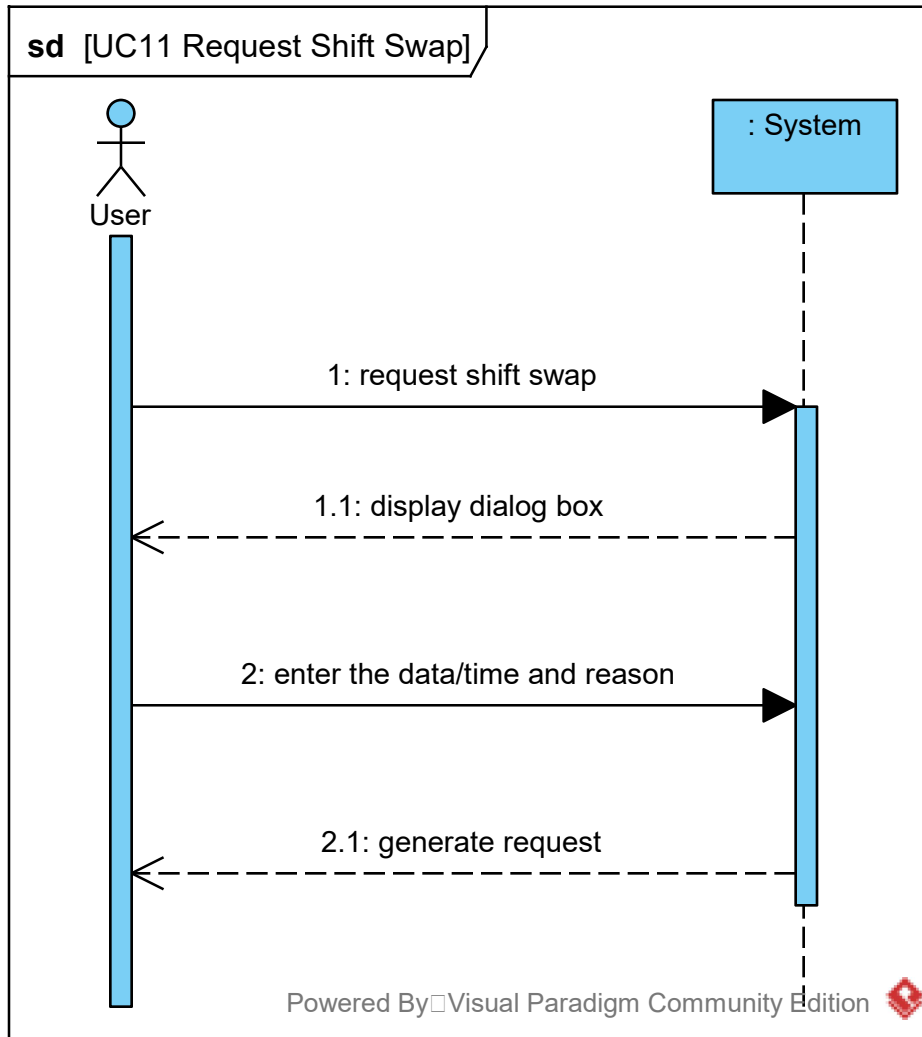
1: Update Holiday Hours

1.1: pop out dialog

2: enter desire data/time

2.1: display new schedule to scren





Employee View

Log out

◀
August
▶

M	T	W	T	F	S	S
	1	2	3	4		

Print Schedule

Make a Request

Manager view

[illegible]

Log IN

User name:

Pass Word:

Change/forget Password

EMPLOYEE REQUEST View

Date:

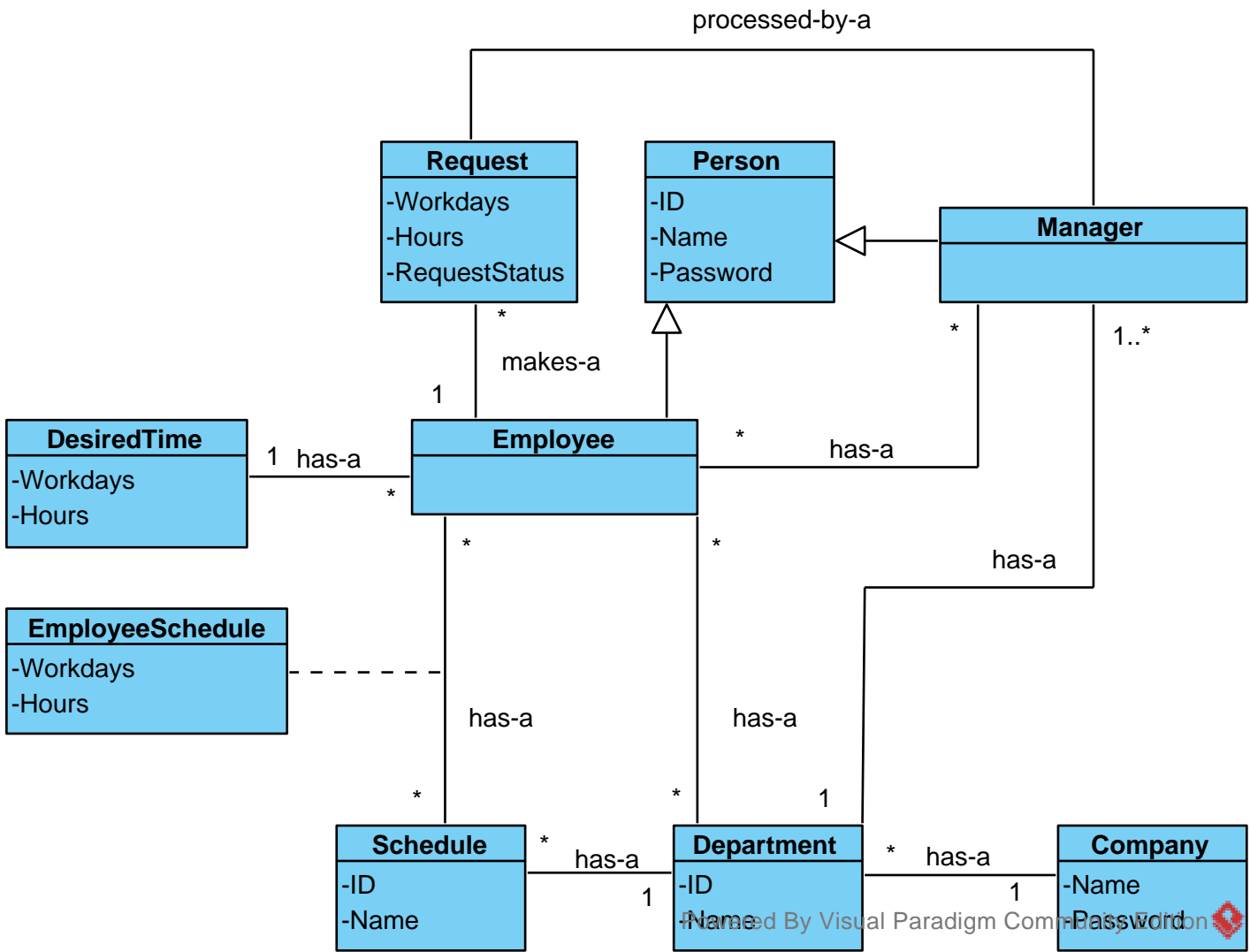
Time: Start END

TYPE:

Reason:

Manager Request View

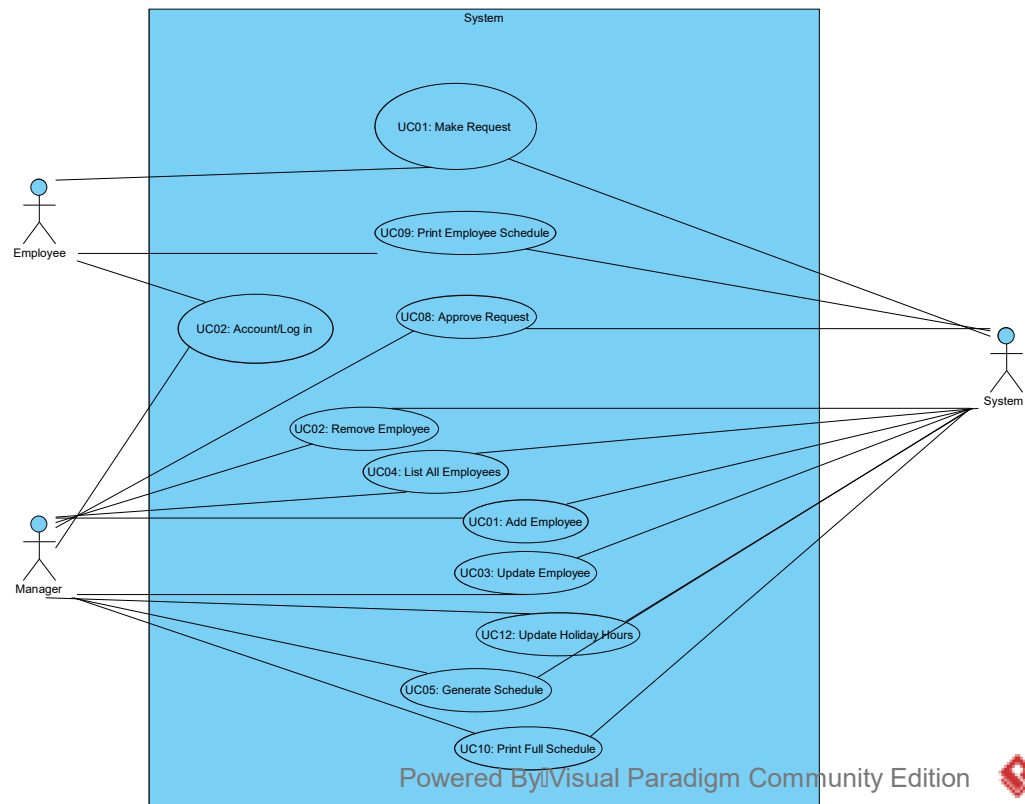
Status	Name	Date: Start/End	TYPE	Team 1
Pending	John Doe	01/01/2025: 0900-1700	PTO	
~	~	~ ~	~	
~	~	~ ~	~	
~	~	~ ~	~	
~	~	~ ~	~	
~	~	~ ~	~	



		UC1 : Add Employee	UC2 : Remove Employee	UC3 : Update Employee	UC4 : List All Employees	UC5 : Generate Schedule	UC6 : Request Schedule Change	UC7 : PTO Request	UC8 : Approve Request	UC9 : Print Employee Schedule	UC10 : Print Full Schedule	UC11 : Request Shift Swap	UC12 : Update Holiday Hours	UC13 : Request Overtime	UC14 : Login	UC15 : Logout
1. Create Optimized Schedule					X		X				X		X			
2. Adjust Employee Schedule			X		X			X				X				
3. List Time Constraints			X			X	X				X		X			
4. Modify Employees	X	X	X											X	X	
5. Create or Remove Employees	X	X														
6. List All Employees				X												
7. Request Schedule Changes					X	X	X									
8. Approve/Deny Employee Requests			X					X								
9. Print Schedules									X	X						

uc [Use Case Iteration1]

Authors:
Josh Wilson
Brad Buckingham
Nick Cook
Bailey Hollman
Lanshi LJ



Powered By Visual Paradigm Community Edition

