



Design Specification

ReadySetResource.com

Contents

[5.1 - Data Flow Diagrams 3](#_Toc514404194)

[5.2 - System Architecture 3](#_Toc514404195)

[5.3 - Use Case Descriptions 4](#_Toc514404196)

[5.4 - Use Case Diagrams 5](#_Toc514404197)

[5.5 - Activity Diagram 9](#_Toc514404198)

[5.6 - Class Diagram 12](#_Toc514404199)

[5.7 - Sequence and Communication Diagrams 13](#_Toc514404200)

[5.8 - State Diagrams 13](#_Toc514404201)

[5.9 - Normalisation 14](#_Toc514404202)

[5.10 - Data Model 15](#_Toc514404203)

[5.11 - Form Layouts 15](#_Toc514404204)

[5.12 - Object-Relational Mapping 15](#_Toc514404205)

[5.13 - Class Definition Document 15](#_Toc514404206)

[5.14 - Data Dictionary 15](#_Toc514404207)

[5.15 - Initial Entity Relationship Diagram 16](#_Toc514404208)

[5.16 - Extended Entity Relationship Diagram 17](#_Toc514404209)

[5.17 - Screen Layouts 18](#_Toc514404210)

[5.18 - Validation Control Forms 18](#_Toc514404211)

[5.19 - Event-Handling Forms 18](#_Toc514404212)

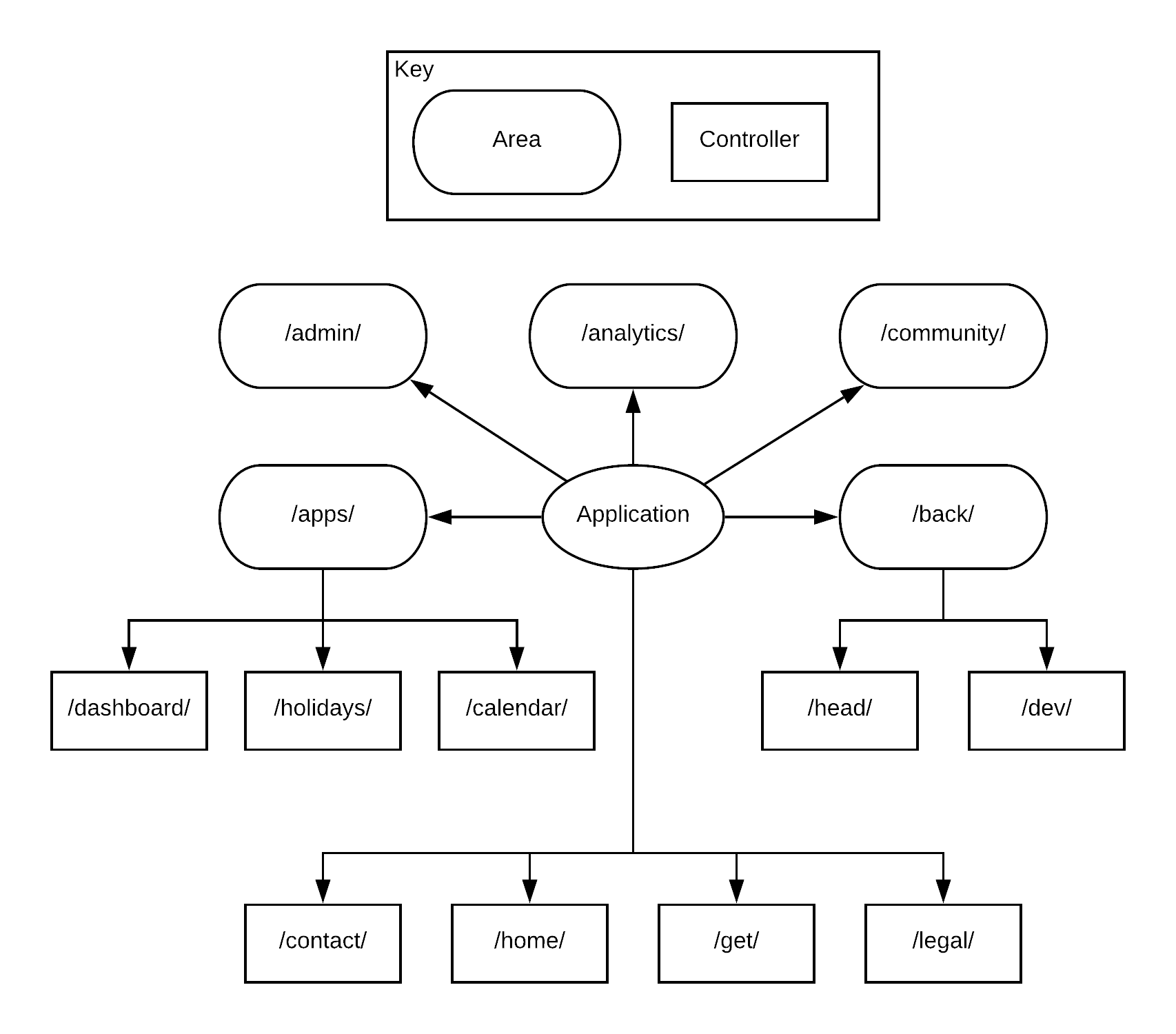
[5.20 - Identifier List 19](#_Toc514404213)

[5.21 - Bibliography 21](#_Toc514404214)

[5.22 - Software Requirements Spec. Work Log 21](#_Toc514404215)

# 5.1 - Data Flow Diagrams

# 5.2 - System Architecture



# 5.3 - Use Case Descriptions

Initial Use Case Description - Creating a Shift

1. Request booking of a shift
2. Check to see if user type can make shifts
3. Prompt the user for day of shift
4. Input day of shift
5. Check for availability for day
6. Prompt the user for start time of shift
7. Input start time of shift
8. Check for availability for start time
9. Prompt the user for end time of shift
10. Input end time of shift
11. Check for availability for end time
12. Accept shift
13. Update system

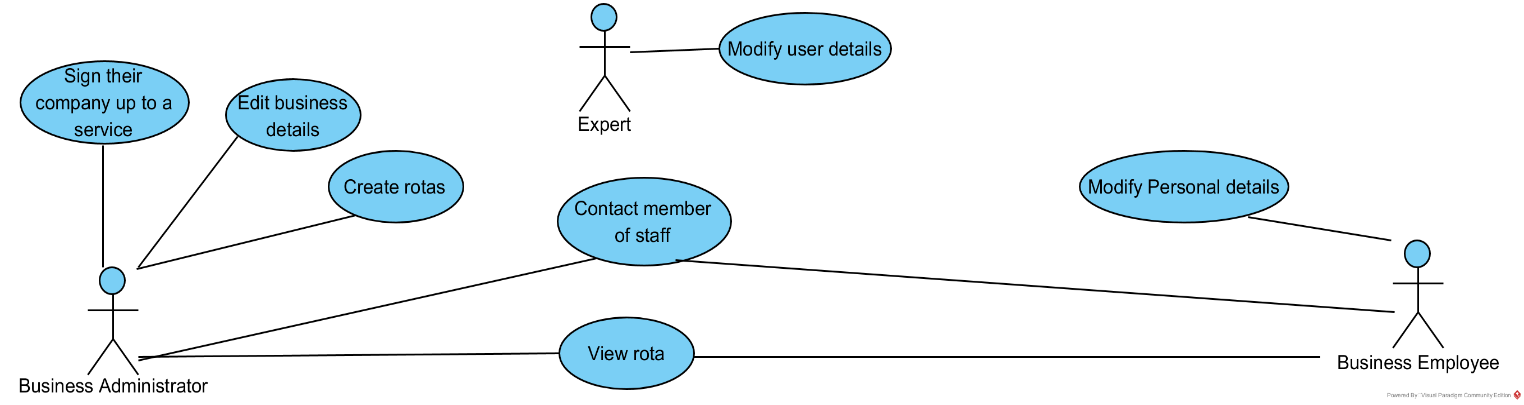
Extended Use Case Description

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | 01 | | |
| Use Case Name | Create A Shift | | |
| Created By | Aidan Marshall | Latest Updated By | Aidan Marshall |
| Date Created | 17/5/18 | Date Last Updated | 17/5/18 |

|  |  |
| --- | --- |
| Actor(s) | User, System |
| Description | The user creates a shift for a user |
| Trigger | The user clicks ‘Add Shift’ in the calendar view |
| Preconditions | The user is signed in  The user has editing rights for calendar |
| Postconditions | Shift is successfully created |
| Priority | High |
| Frequency of use | On demand |
| Normal Course of Events | 1. User requests booking of a shift 2. System checks to see if user type can make shifts 3. System prompts the user for day of shift 4. User inputs day of shift 5. System checks for availability for day 6. System prompts the user for start time of shift 7. User inputs start time of shift 8. System checks for availability for start time 9. System prompts the user for end time of shift 10. User inputs end time of shift 11. System checks for availability for end time 12. System accepts shift 13. System is updated |
|  | 1. User does not have editing rights    1. User is redirected    2. Process is terminated 2. The chosen date is not available    1. System offers to change date       1. User cancels       2. User is redirected    2. User chooses an alternative date    3. Continue to step 6   8. The chosen start time is not available   1. System offers to change start time    * 1. User cancels      2. User is redirected 2. User chooses an alternative start time 3. Continue to step 9   11. The chosen end time is not available   1. System offers to change end time    * 1. User cancels      2. User is redirected 2. User chooses an alternative end time 3. Continue to step 12 |
| Expectations | The user will be able to make a shift |
| Includes | Includes the register membership |
| Special Requirements | The user’s type allows for editing of the calendar |
| Assumptions |  |
| Notes |  |
| Issues | None to date |

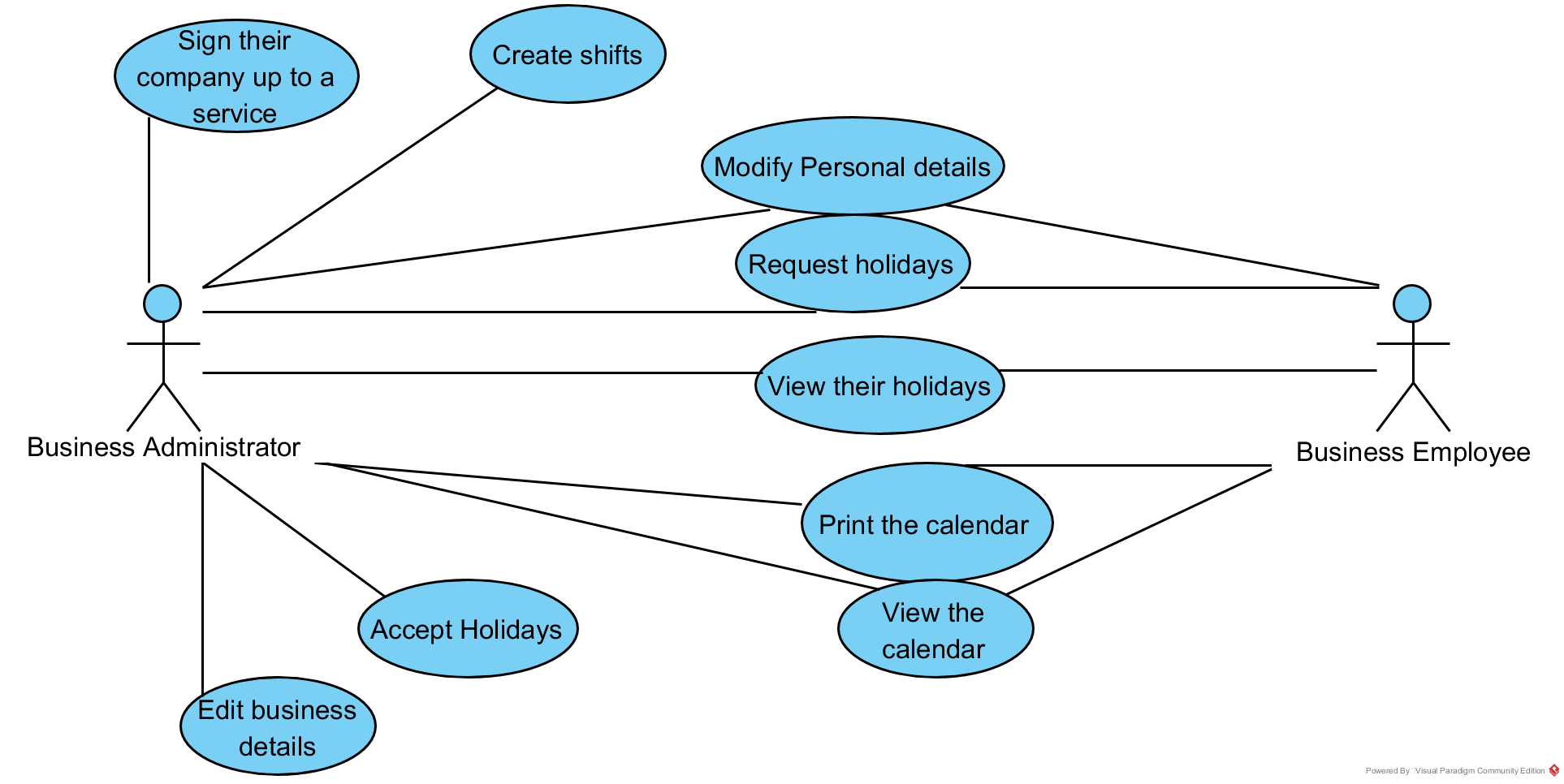
# 5.4 - Use Case Diagrams

Initial Use Case Diagram - V1

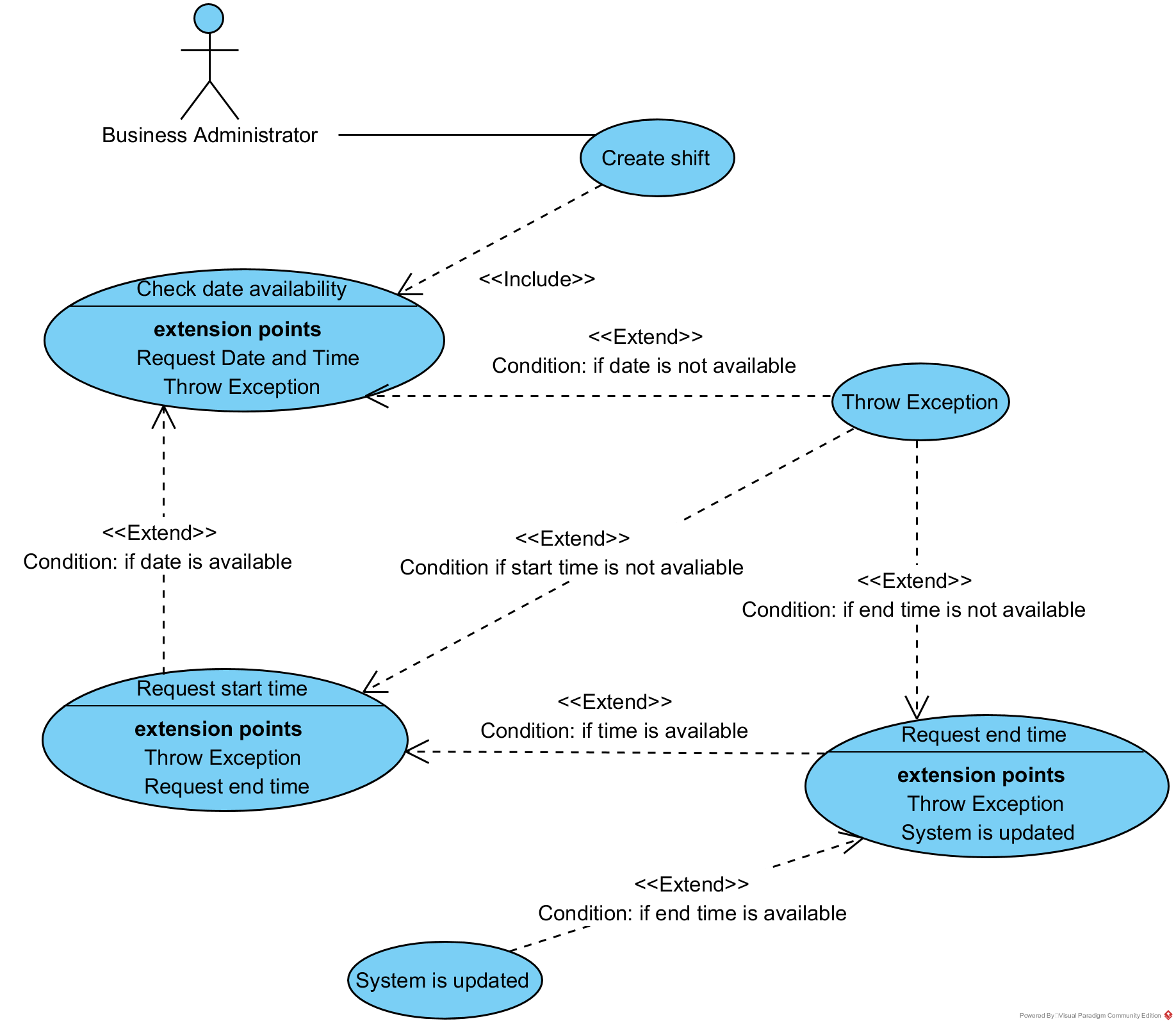


Initial Use Case Diagram - V2

Note: Graham introduced the idea of including the updating of the rota and production of reports. Due to this and a modification in system functionality, below is the revisited Initial Use Case Diagram. Rota has been changed to calendar as it was causing confusion with staff members and experts were removed as both the administrator of a business and the employee of a business are able to edit their details.

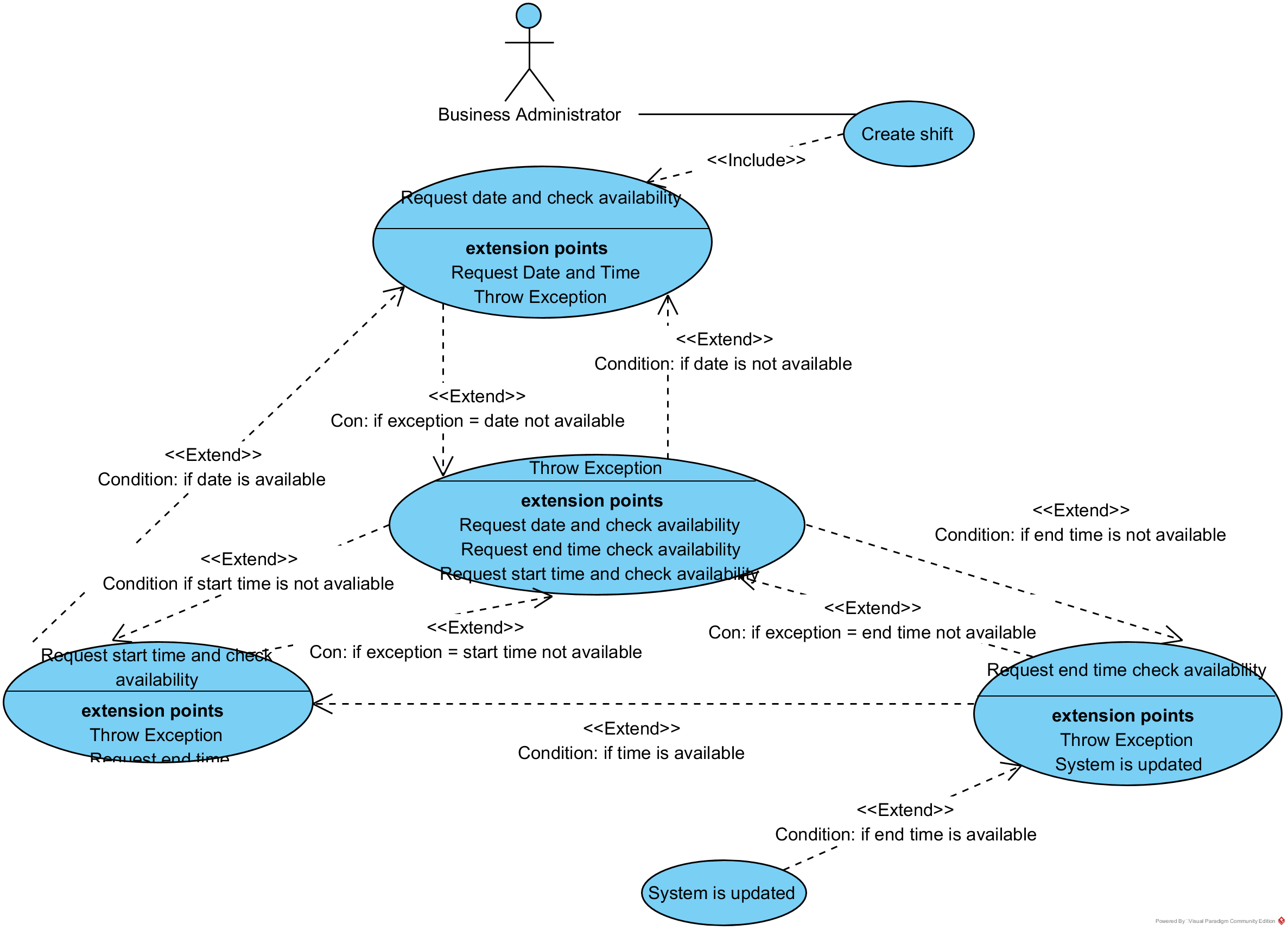


Extended Use Case Diagram - V1



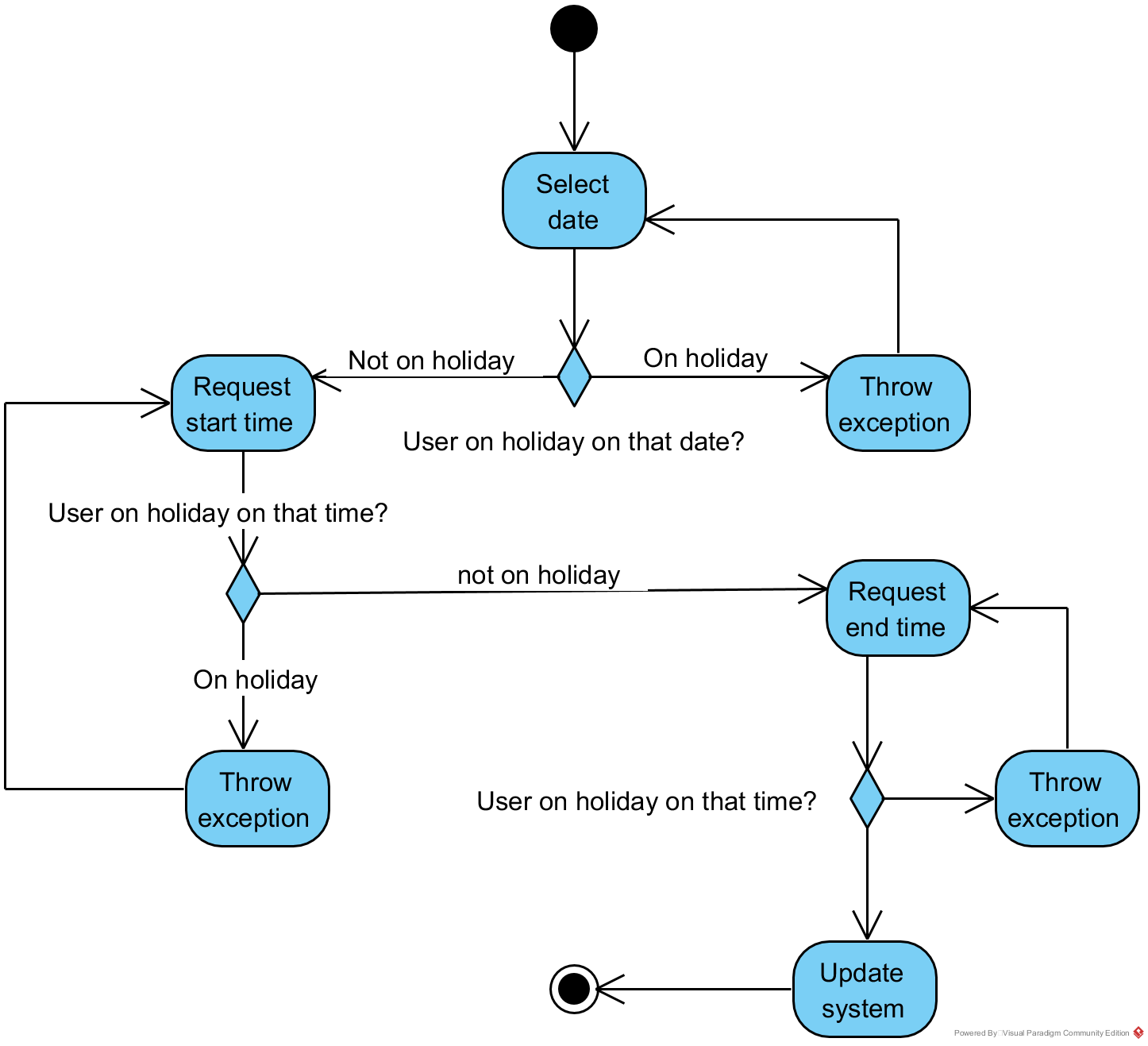
Extended Use Case Diagram - V2

Note: After discussions with Graham, it was pointed out to myself that nothing occurs after the exception is handled, also that the shift was being created and then the date was being checked, but there is nowhere to add a date in. Thus, I have made the appropriate changes displayed below. When an exception is thrown, the user needs to return to the previous step and try again.



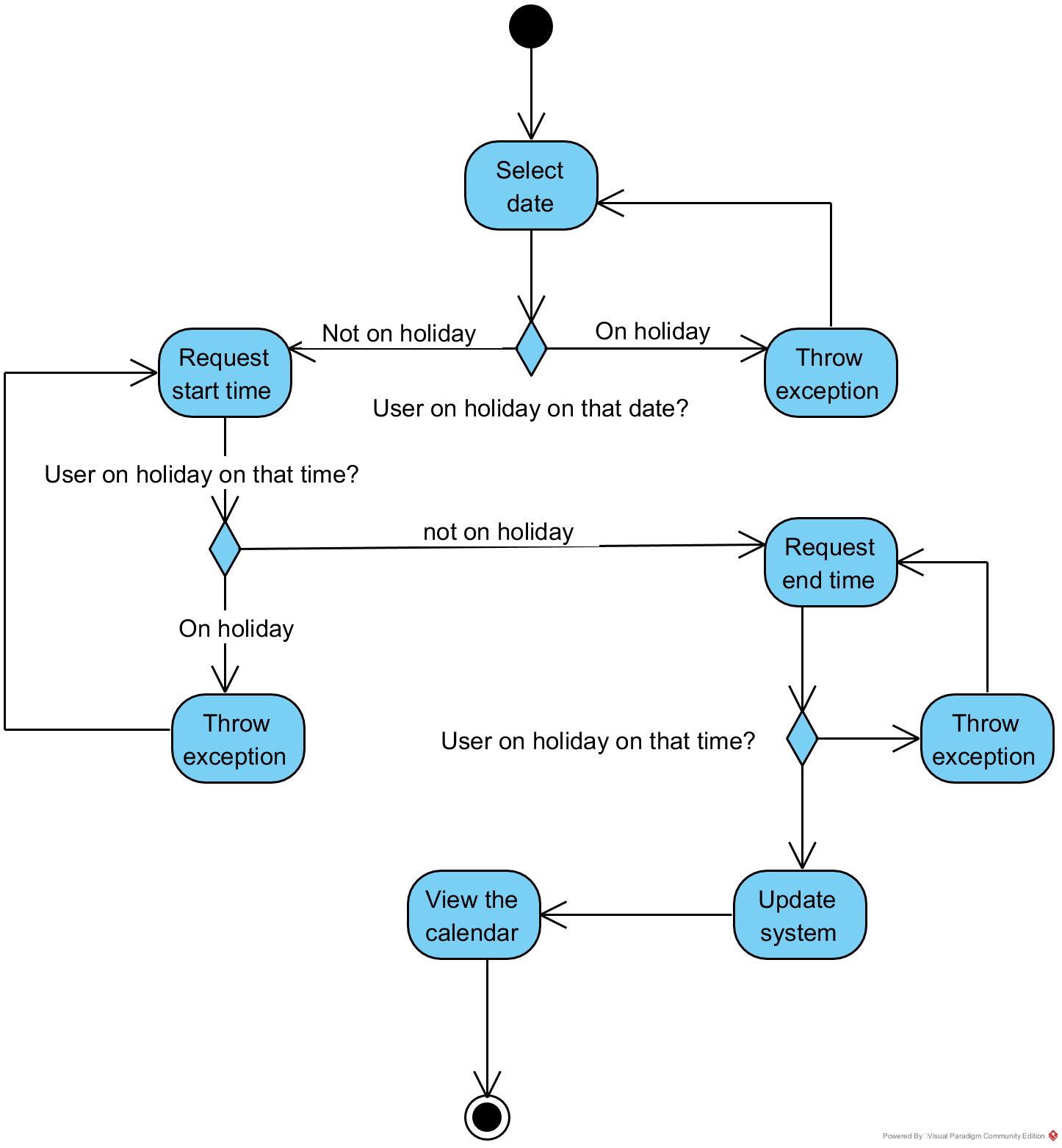
# 5.5 - Activity Diagram

Activity Diagram - V1



Activity Diagram - V2

Note: Upon receiving feedback from Graham, it was pointed out to myself that nothing occurs after the system is updated and what should happen is the user is redirected to the calendar where the user can view their shifts, their co-workers’ shifts, add a shift (if the type they are assigned to allows) and print the calendar. Below is the activity diagram with the redirect added, and below that is the algorithm that is used to display the calendar (the system will be the user of that activity diagram).



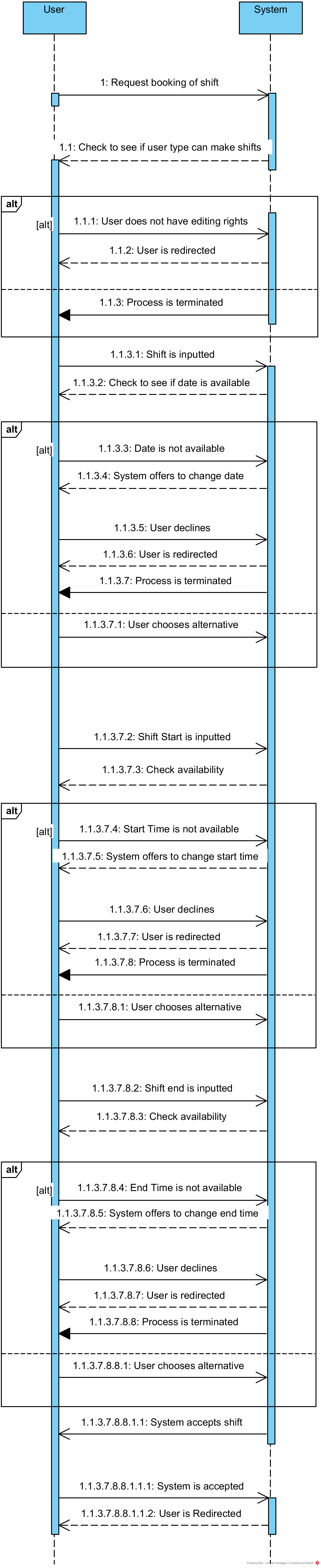
Algorithm Activity Diagram - V1

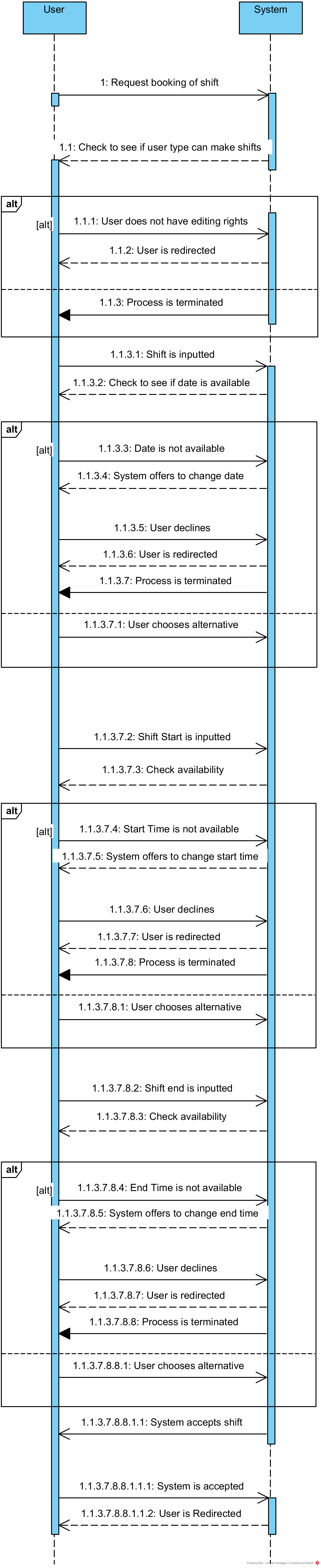
Note: The algorithm used in my calendar was by far my biggest challenge in the project and seems non-sensical at times. In my evaluation report I will discuss the difficulties I had to overcome, the errors that I made and how I can improve myself moving forward based on the experience.

# 5.6 - Class Diagram

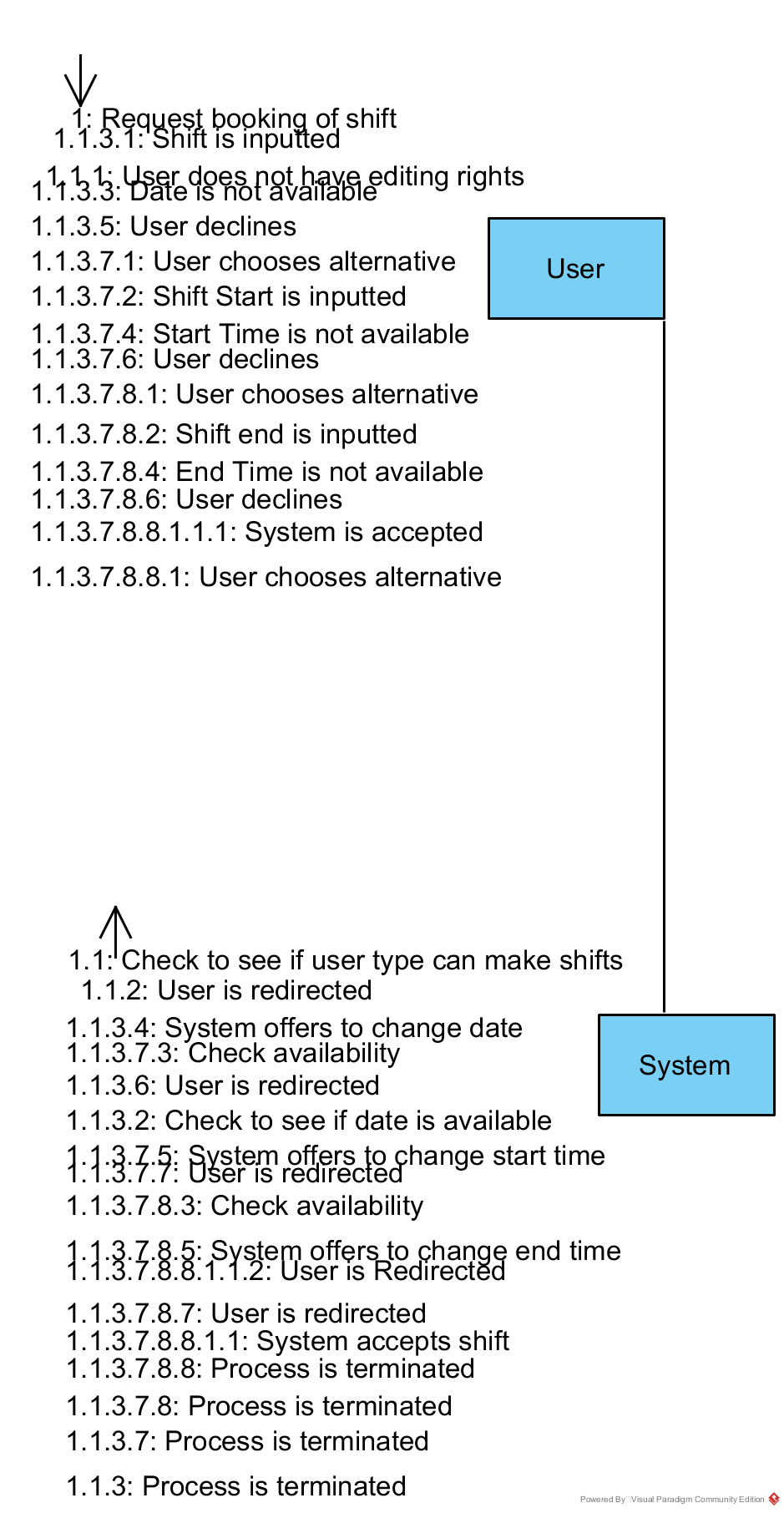
Note: Above is my completed class diagram, this will allow me to create the following ‘apps’ within the program: Calendar, Holidays, Meetings, Messages, Employee Store, Ideas, Updates. In addition, it would allow for business to be charged (Transactions) and a community section for questions and answers that the users can discuss in. Looking back, although I will implement this functionality in the future, I understand that I was being over-ambitious in the timescale provided.

# 5.7 - Sequence and Communications Diagrams

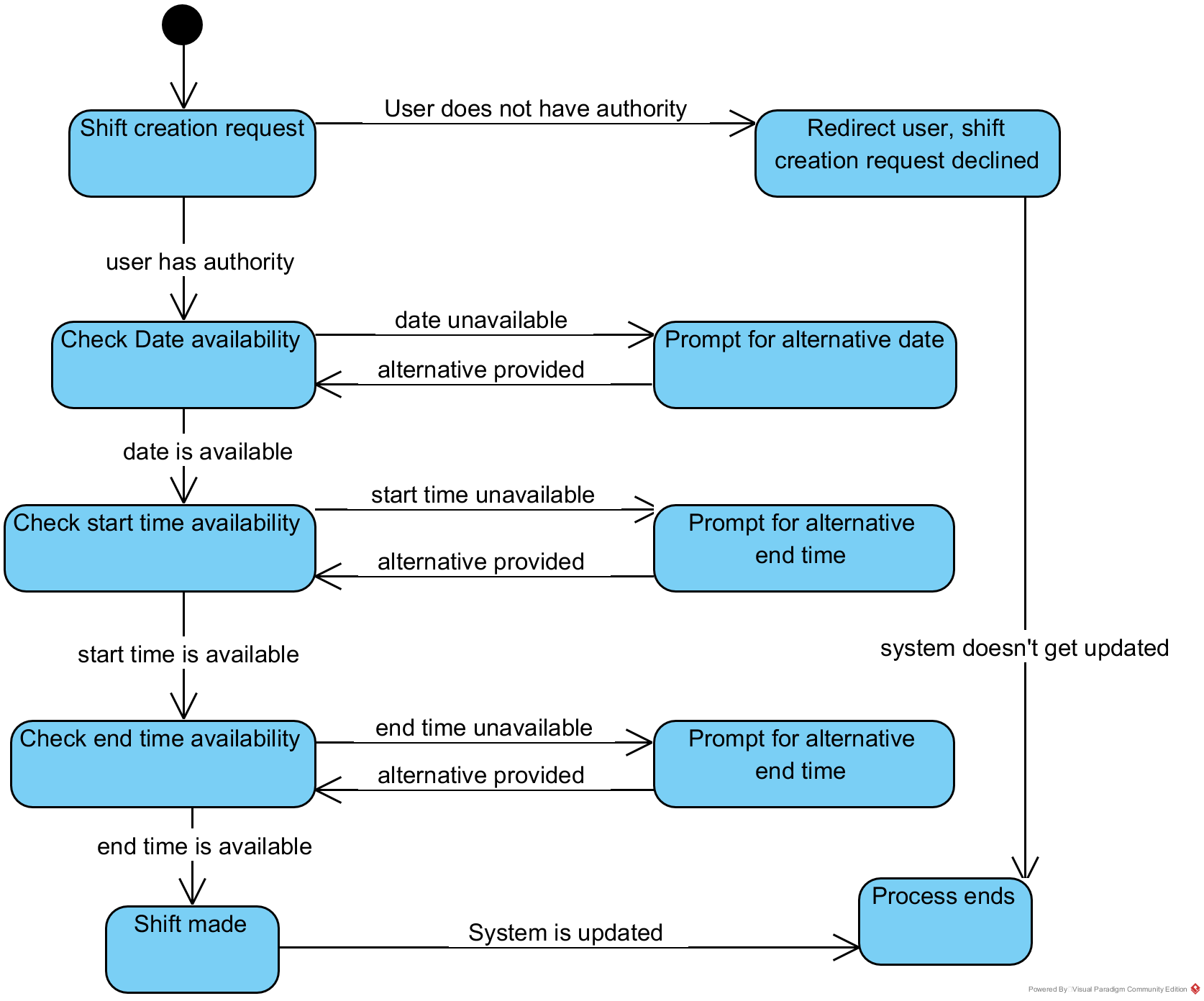
Sequence Diagram - V1

b

Communications Diagram - V1



# 5.8 - State Diagrams



# 5.9 - Normalisation

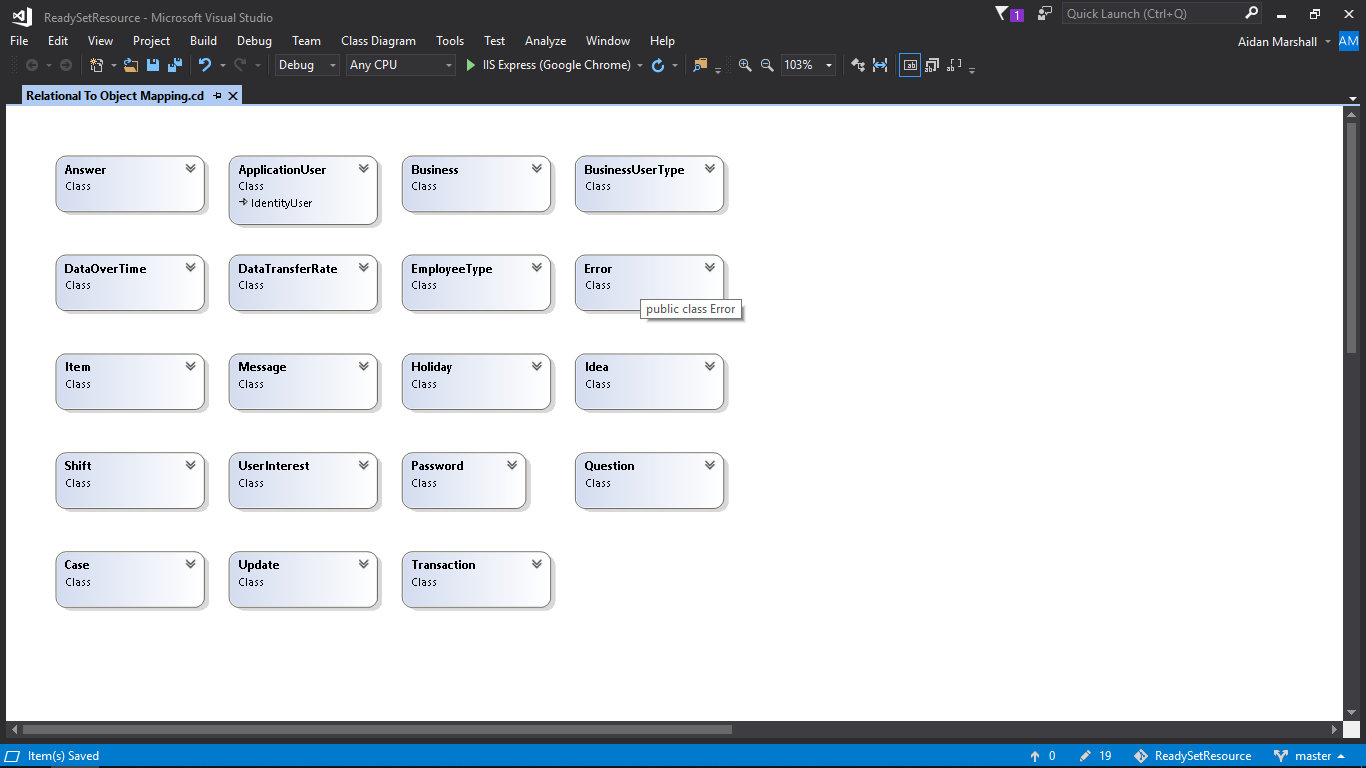
| UNF | 1NF | 2NF | 3NF |
| --- | --- | --- | --- |
| Order Id | Order Id | Order Id | Order Id |
| Order Date | Order Date | Order Date | Order Date |
| Staff Name | Staff Name | Staff Name | Staff Id\* |
| Staff Id | Staff Id | Staff Id | Customer Id\* |
| Customer Id | Customer Id | Customer Id | Quantity |
| Customer Name | Customer Name | Customer Name |  |
| Street | Street | Street |  |
| Town | Town | Town | Staff Id |
| Postcode | Postcode | Postcode | Staff Name |
| Customer Phone Number | Customer Phone Number | Customer Phone Number |  |
| Product Id |  |  | Customer Id |
| Product Name | Order Id | Order Id | Customer Name |
| Quantity | Product Id | Product Id | Street |
| Product Price | Product Name | Order Total | Town |
| Order Total | Quantity |  | Postcode |
|  | Product Price | Product Id | Customer Phone Number |
|  | Order Total | Product Name |  |
|  |  | Quantity | Product Id |
|  |  | Product Price | Product Name |
|  |  |  | Product Price |
|  |  |  |  |
|  |  |  | Order Id |
|  |  |  | Product Id |
|  |  |  | Order Total |

# 5.10 - Data Model

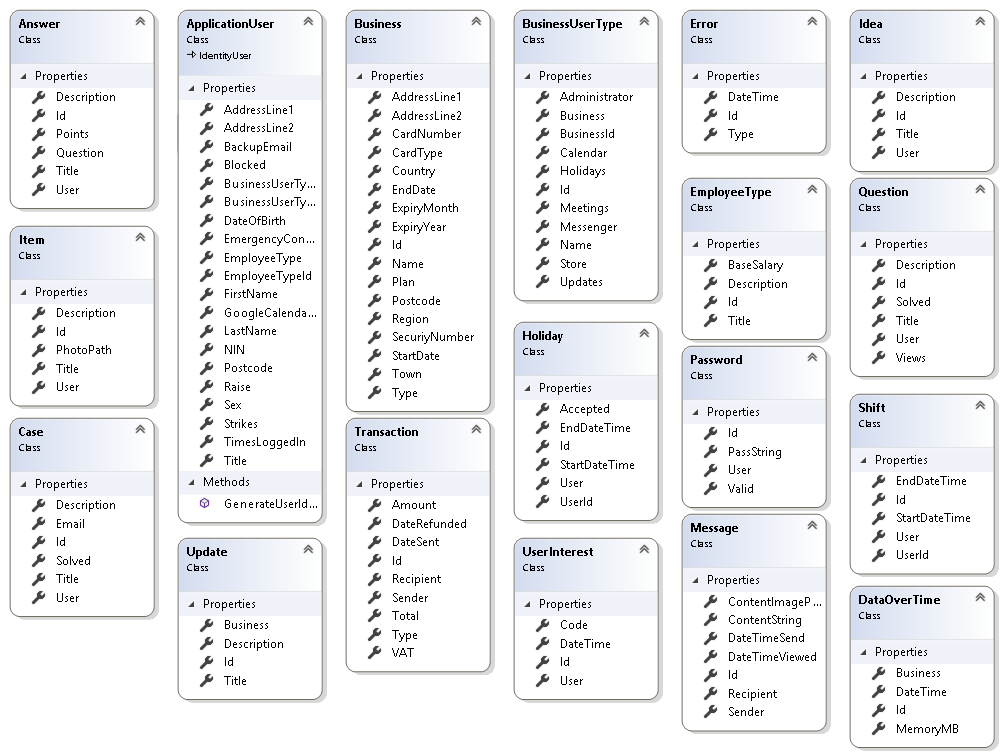
# 5.11 - Form Layouts

# 5.12 - Object-Relational Mapping

This diagram shows the updated class diagram without their properties:



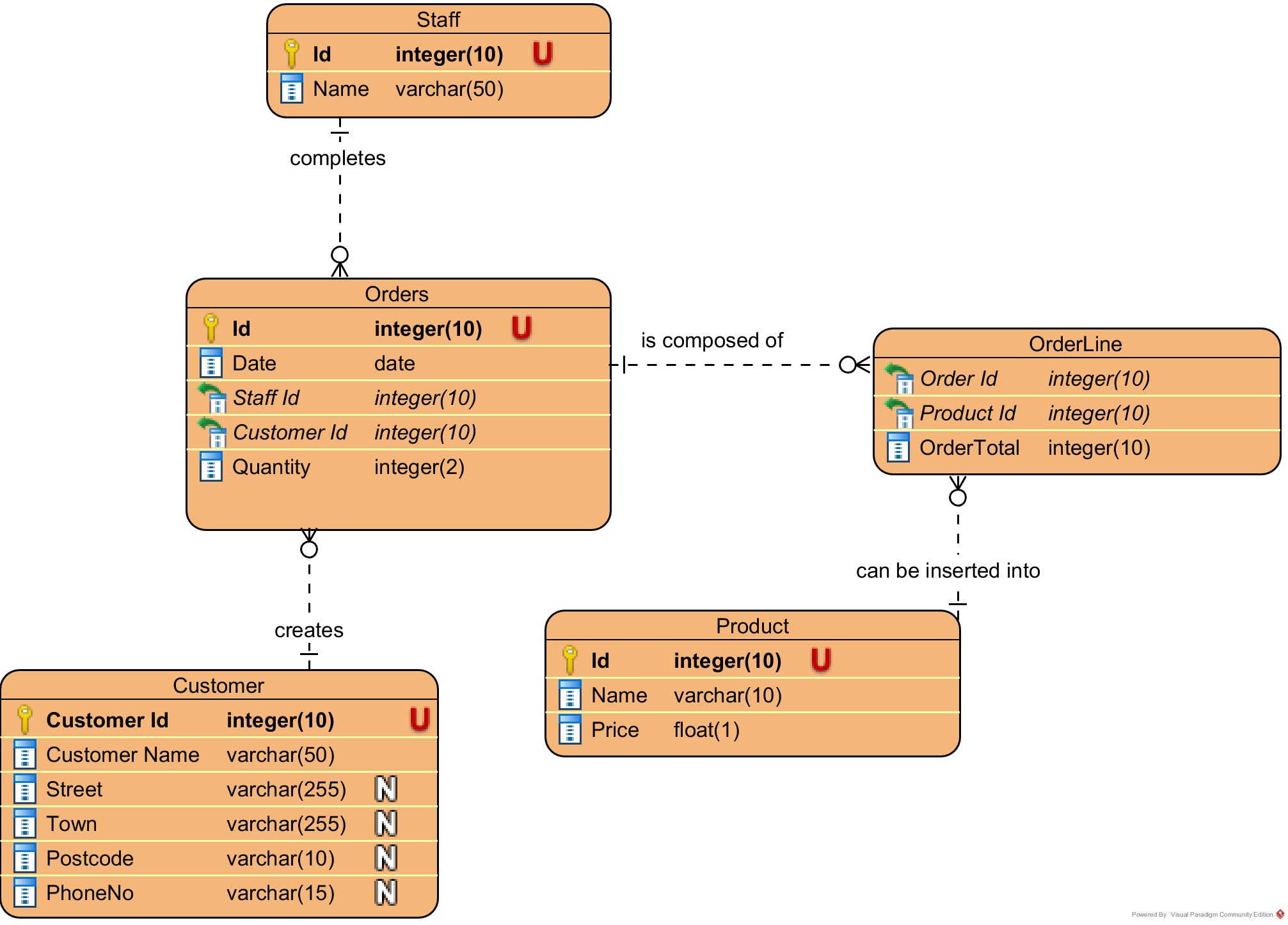
This diagram shows the updated class diagram with their respective properties:



# 5.13 - Class Definition Document

# 5.14 - Data Dictionary

# 5.15 - Initial Entity Relationship Diagram



Note: Over the course of a few days, I planned ways in which I could add to this ERD to create the program that I designed. I merged the customer and staff to create ‘user’, then I was going to change the orders, order line and product tables to calendars, calendar lines and shifts respectively, but I decided that was overcomplicated and it is derivable data as you can get the calendar from the date which is an attribute in each shift. After that I added tables and attributes which you can see in the next diagram.

# 5.16 - Extended Entity Relationship Diagram

# 5.17 - Screen Layouts

# 5.18 - Validation Control Forms

# 5.19 - Event-Handling Forms

# C:\Users\Aidan Marshall\Documents\Projects\ReadySetResource\Design\Diagrams\State Diagram.pngC:\Users\Aidan Marshall\Documents\Projects\ReadySetResource\Design\Diagrams\State Diagram.png5.20 -

# Identifier List

Users

| Name | Scope | Description |
| --- | --- | --- |
| Id | Global | Integer(10) - Used as a PK for users |
| Title | Global |  |
| FirstName | Global |  |
| LastName | Global |  |
| DateOfBirth | Global |  |
| Email | Global |  |
| BUEmail | Global |  |
| ProfilePicLink | Global |  |
| CountryCode | Global |  |
| PhoneNumber | Global |  |
| AddressLine1 | Global |  |
| AddressLine2 | Global |  |
| PostCode | Global |  |
| Town | Global |  |
| Region | Global |  |
| Country | Global |  |
| Blocked | Global |  |
| TimesLoggedIn | Global |  |
| Sex | Global |  |
| NIN | Global |  |
| EmergencyContact | Global |  |
| Raise | Global |  |
| Strikes | Global |  |

Users

| Name | Scope | Description |
| --- | --- | --- |
| Id | Global |  |
| Titile | Global |  |
| FirstName | Global |  |
| LastName | Global |  |
| DateOfBirth | Global |  |
| Email | Global |  |
| BUEmail | Global |  |
| ProfilePicLink | Global |  |
| CountryCode | Global |  |
| PhoneNumber | Global |  |
| AddressLine1 | Global |  |
| AddressLine2 | Global |  |
| PostCode | Global |  |
| Town | Global |  |
| Region | Global |  |
| Country | Global |  |
| Blocked | Global |  |
| TimesLoggedIn | Global |  |
| Sex | Global |  |
| NIN | Global |  |
| EmergencyContact | Global |  |
| Raise | Global |  |
| Strikes | Global |  |

# 5.21 - Bibliography

# 5.22 - Software Requirements Spec. Work Log

| Entry | Description | Person | Date | Mins Spent |
| --- | --- | --- | --- | --- |
| 1 | Formatted the document | Aidan Marshall | 25/11/17 | 6 |
| 2 | Started the Conceptual ERD | Aidan Marshall | 14/11/17 | 68 |
| 3 | Continued the Conceptual ERD | Aidan Marshall | 16/11/17 | 43 |
| 4 | Continued the Conceptual ERD | Aidan Marshall | 19/11/17 | 82 |
| 5 | Continued the Conceptual ERD | Aidan Marshall | 22/11/17 | 122 |
| 6 | Wrote the Conceptual ERD key | Aidan Marshall | 25/11/17 | 32 |
| 7 | Continued the Conceptual ERD | Aidan Marshall | 25/11/17 | 145 |
| 8 | Continued the Conceptual ERD | Aidan Marshall | 26/11/17 | 188 |
| 9 | Finished the Conceptual ERD | Aidan Marshall | 27/11/17 | 328 |
| 10 | Modified the conceptual ERD | Aidan Marshall | 28/11/17 | 120 |
| 11 | Decided to concentrate on critical requirements | Aidan, Graham | 26/3/18 | 12 |
| 12 | Took out ERD descriptions | Aidan Marshall | 26/3/18 | 4 |
| 11 | Created the use case diagrams | Aidan Marshall | 26/3/18 | 62 |
| 12 | Created the class diagram | Aidan Marshall | 26/3/18 | 26 |
| 11 | Created the ERDs | Aidan Marshall | 26/3/18 | 79 |
| 12 | Created the use case descriptions | Aidan Marshall | 26/3/18 | 16 |
| 13 | Created the identifier list | Aidan Marshall | 26/3/18 | 22 |
| 14 | Created Initial Use Case Diagram V2 | Aidan Marshall | 17/5/18 | 9 |
| 15 | Created Activity Diagram V2 | Aidan Marshall | 17/5/18 | 14 |
| 16 | Discussed class diagram | Aidan Marshall | 17/5/18 | 10 |
| 17 | Discussed initial ERD | Aidan Marshall | 17/5/18 | 5 |
| 18 | Added state and communications diagram | Aidan Marshall | 18/5/18 | 139 |
| 19 | Completed state diagram | Aidan Marshall | 18/5/18 | 22 |
| 20 | Added the system architrcture | Aidan Marshall | 2/6/18 | 17 |
| 21 |  |  |  |  |
| 22 |  |  |  |  |
| 23 |  |  |  |  |
| 24 |  |  |  |  |
| 25 |  |  |  |  |
| 26 |  |  |  |  |
| 1571 | | | | |