coursework 1

Enterprise Application Development

Manuel Benavent Lledo – w1749459

Coursework 1

Enterprise Application Development

University of Westminster

Index

[1. Part A: Requirements 2](#_Toc23859983)

[2. Part B: Use Case Diagrams 3](#_Toc23859984)

[3. Part C – Classes 4](#_Toc23859985)

[1) CRC table 4](#_Toc23859986)

[2) Domain model 5](#_Toc23859987)

# Part A: Requirements

These are the requirements needed for building the time management and reporting tool for personal use:

**R1. The software shall allow the user to insert a new contact and edit the existing ones.**

R1.1. The information about the contact will be stored in a data base

R1.2. The contact must have a first name and the system must validate it, that is, it can be composed only by alphabetic characters.

R1.3. The contact may have an email address, if so, the software will validate it.

R1.4. The software will optionally allow the user to add a telephone number, it will check that is only composed by numbers (no more validations will be required since each country has a different format).

**R2. The software shall allow the user to create (or edit existing) events.**

R2.1. All events must have a name.

R2.2. All events must have a start and an end date and time.

R2.3. Events may be one-off or recurring. If the event is recurring, the user will be able to introduce the number of days, weeks, months or years that it will be recurring for.

R2.4. There will be different types of events: appointments, tasks, lectures and tutorials

R2.5. These events will be stored in the database.

R2.6. An event may have one or more contacts associated.

R2.7. An event may have a location

**R3. The software shall display the user interface with the following options:**

R3.1. The user interface will allow the user to introduce all the information mentioned above.

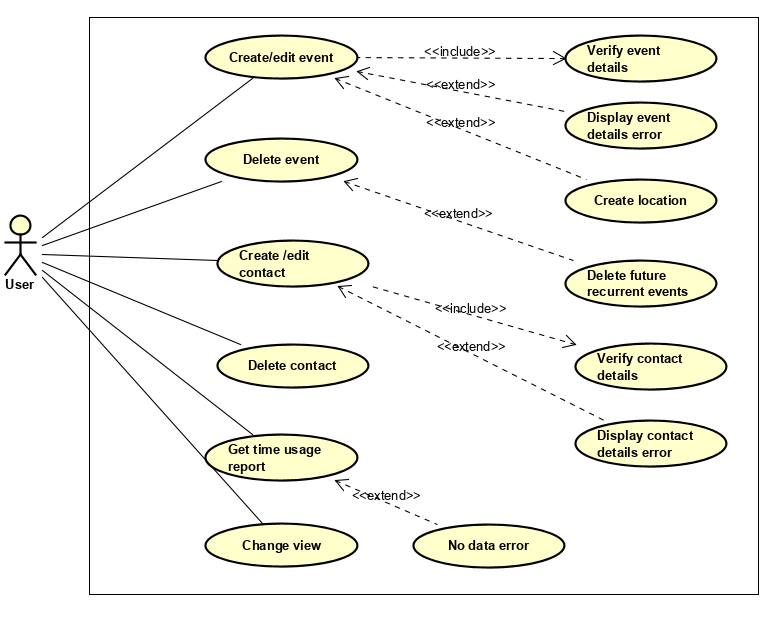
R3.2. The system will provide the user with different views of the calendar: weekly, monthly and schedule.

R3.3. The system will provide an option to show a list of the contacts and an option to edit each of them.

R3.4. The system will provide the user with a time usage report for the next 4 weeks based on the previous weeks’ time usage.

# Part B: Use Case Diagrams

The designed system has the following use case diagram:



The uses cases shown above have the following descriptions:

Use Case: Create/edit event

Id: UC-001

Description:

The user wants to create or edit an event

Primary Actor:

User: the user of the system

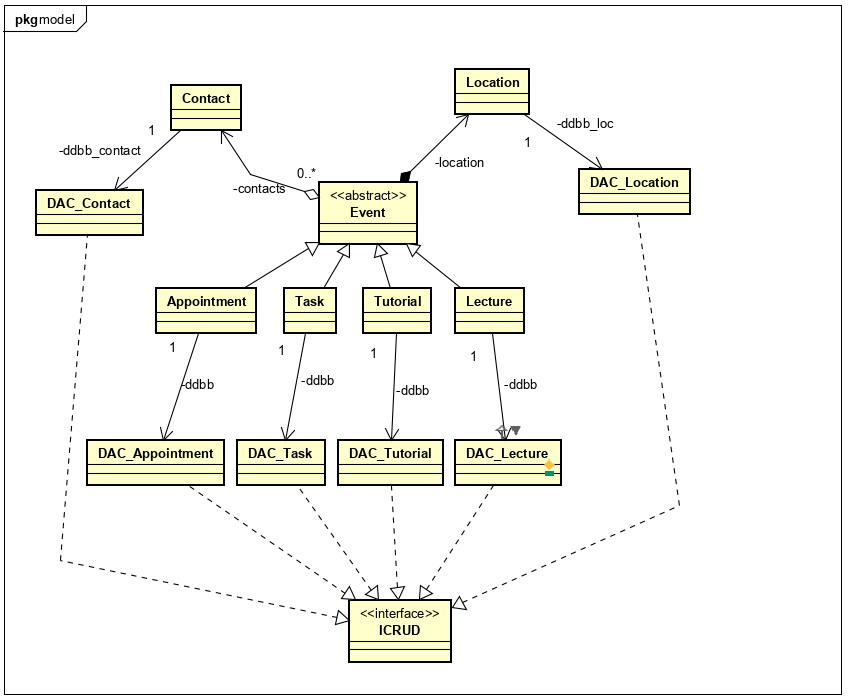
# Part C – Classes

## CRC table

|  |  |  |  |
| --- | --- | --- | --- |
| Class Name | Type | Responsibility | Collaborations |
| Event | Controller | Interacts with its model to obtain data and pass it to the views | DAC\_Event |
| DAC\_Event  (DAC=Data Access Component) | Model | Interact with the DDBB obtaining event’s information | Event |
| EventView | View | View showing the information of an event | Event, Contact |
| EventEditView | View | View for creating/editing events | Event, Contact |
| Contact | Controller | Interacts with its model to obtain data and pass it to the views | DAC\_Contact |
| DAC\_Contact | Model | Interacts with the DDBB obtaining contact’s information | Contact |
| ContactView | View | View showing the information of a contact | Contact |
| ContactEditView | View | View for creating/editing contacts | Contact |
| ContactListView | View | View for showing all the contacts stored for edition/deletion of any of them | Contact |
| WeeklyView | View | This is the main view of the program, displays a weekly view of the calendar and some different options for adding events or contacts, changing the view type and displaying the time usage prediction | Event, Contact |
| ScheduleView | View | View that may replace the MainView which is WeeklyView, it has the same options but the information is displayed in a different form | Event, Contact |
| TimePredictionView | View | This view will show the report of expected time use | Event |

## Domain model

The system will be using the following domain model:



The classes that implement part of the view or the controller are not included in the domain model. However, since some of these classes will be able to throw different types of exceptions there is a domain model for the exceptions:

