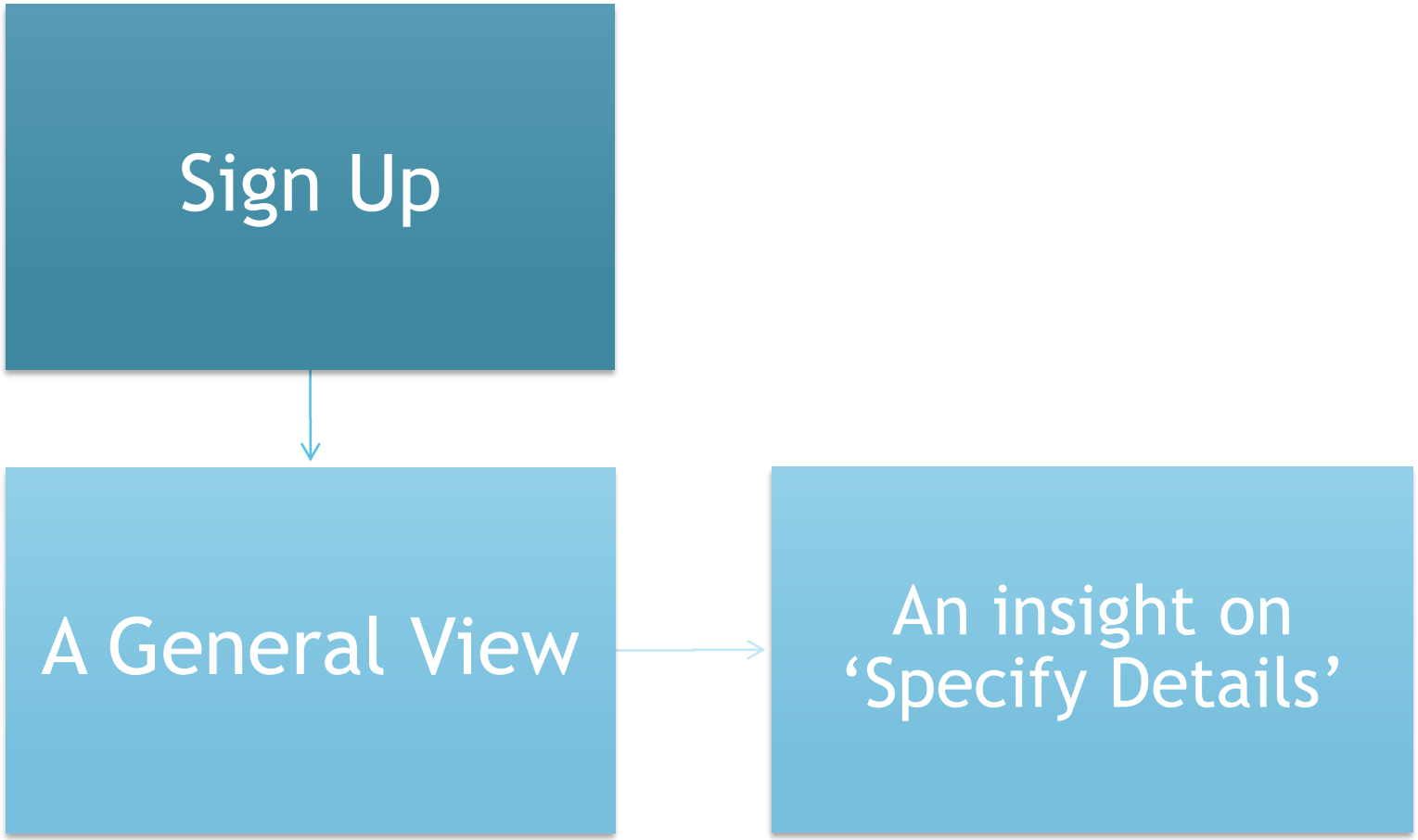
Software Engineering 2 Project, 2017/2018
Politecnico di Milano

Travlendar+

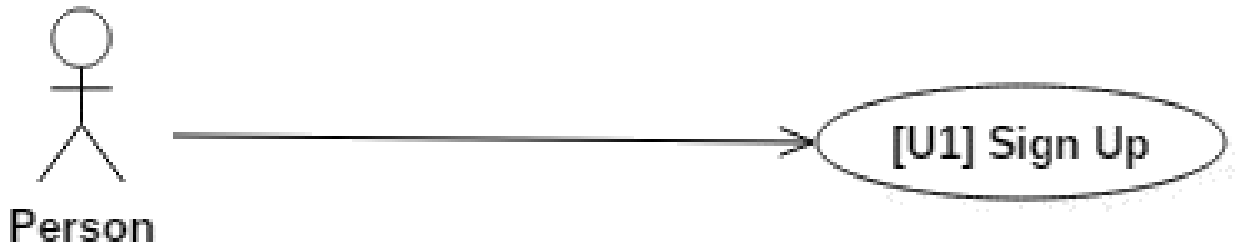
Mirko Salaris 895394
Piervincenzo Ventrella 898604
Pietro Cassarino 899152

Use Cases



Sign Up

Sign Up



Sign Up

Sign Up

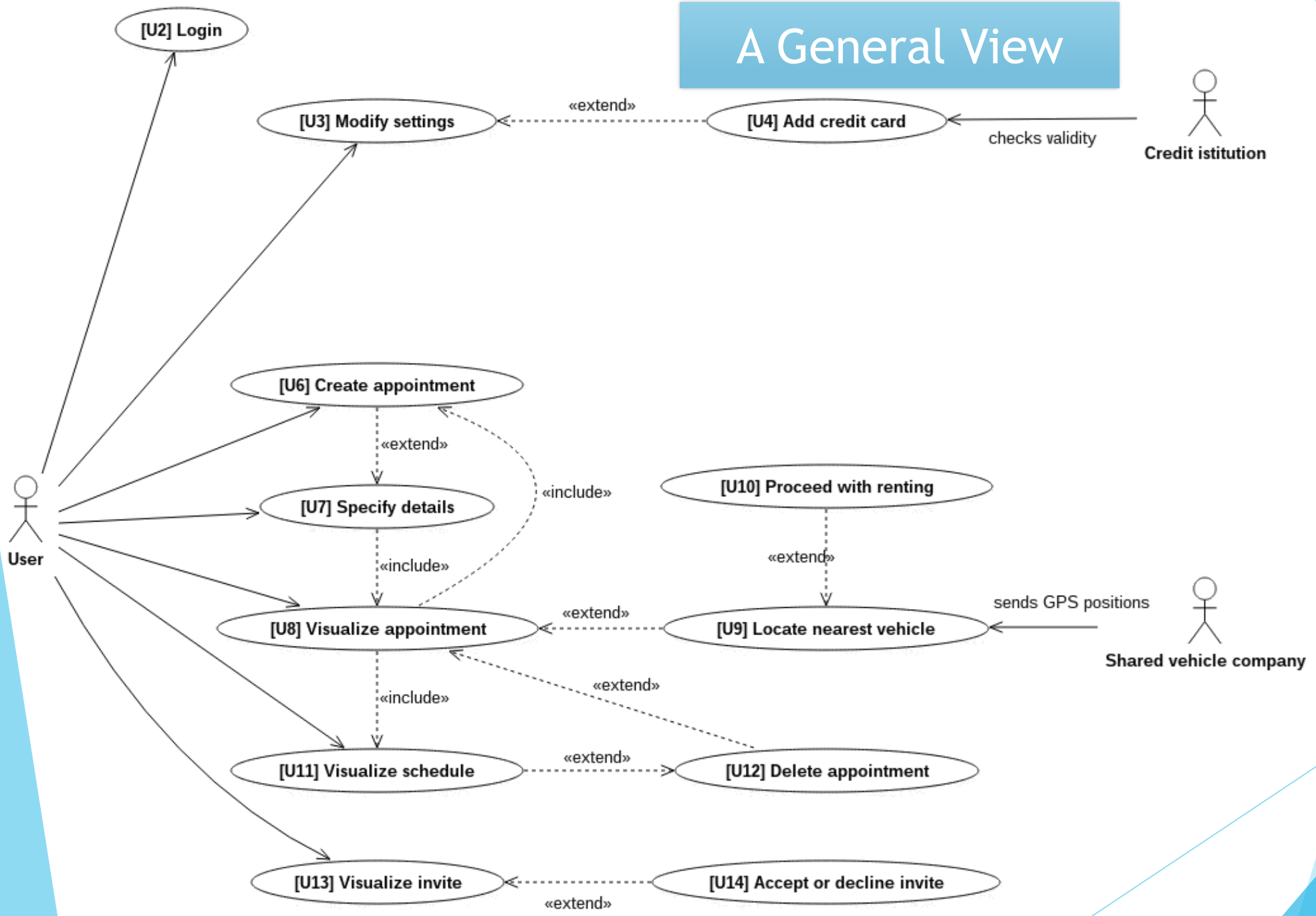
- [G1] a Person should be able to have his/her own Travlendar+ agenda

Sign Up

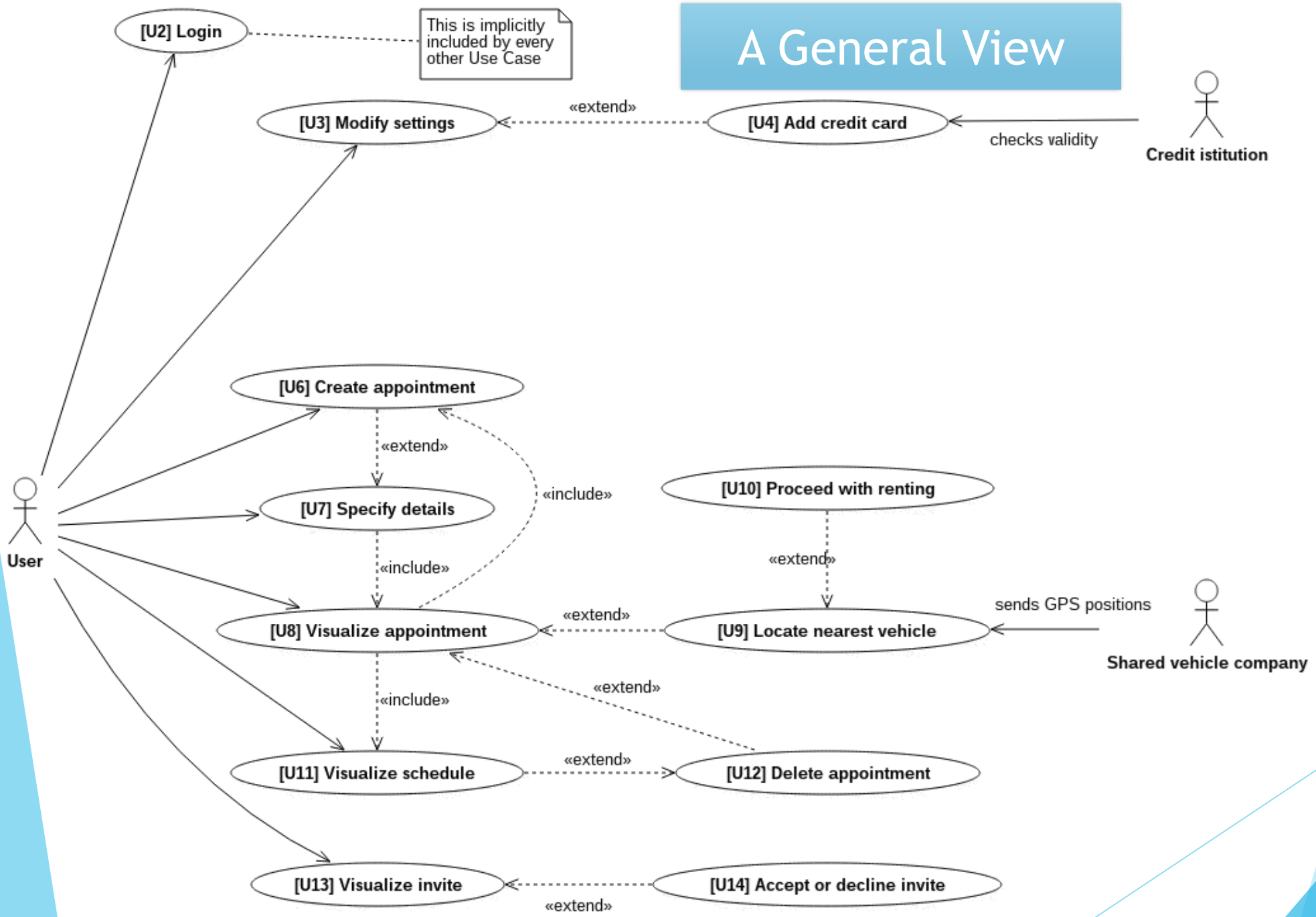
- [G1] a Person should be able to have his/her own Travlendar+ agenda

A General View

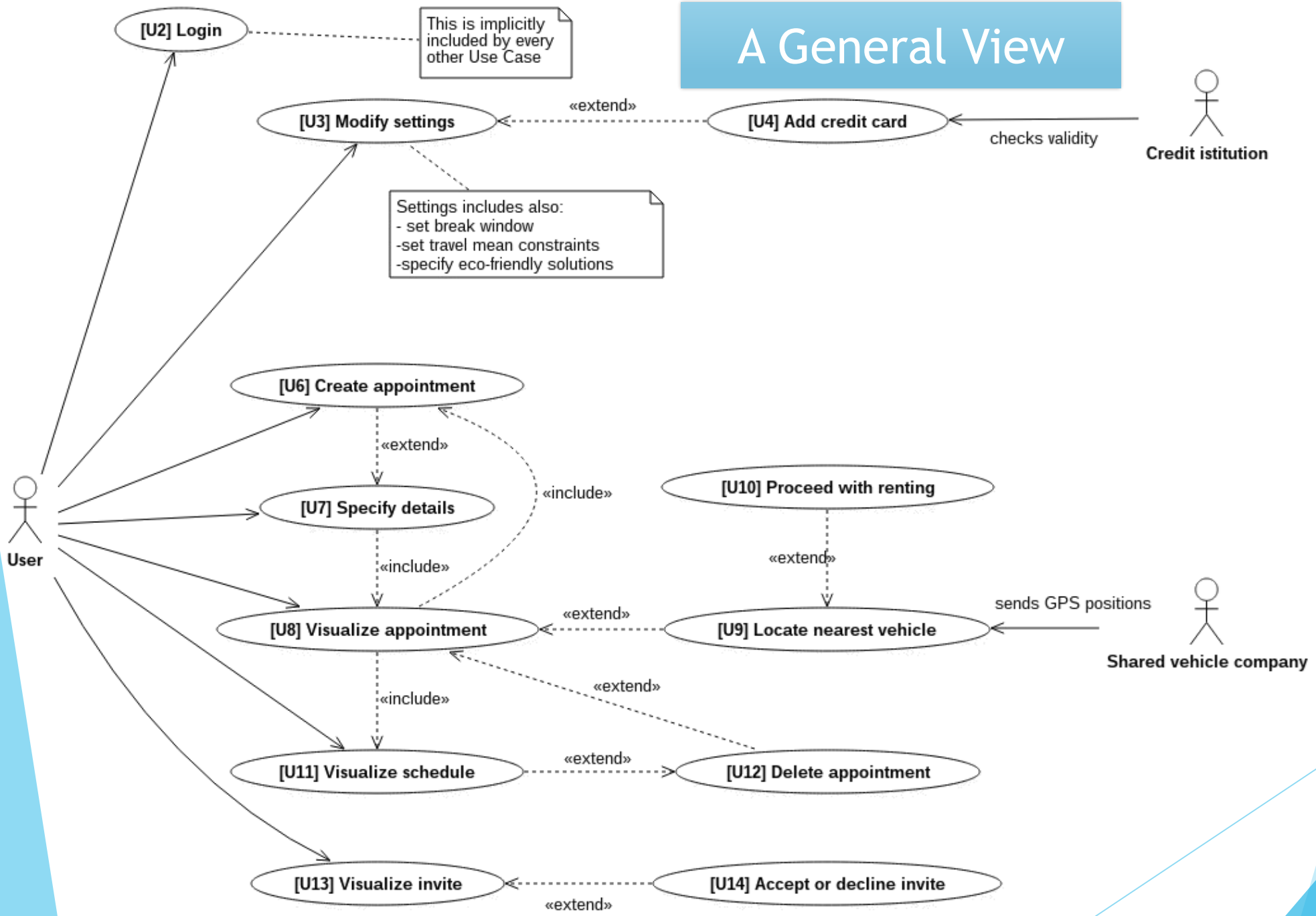
A General View



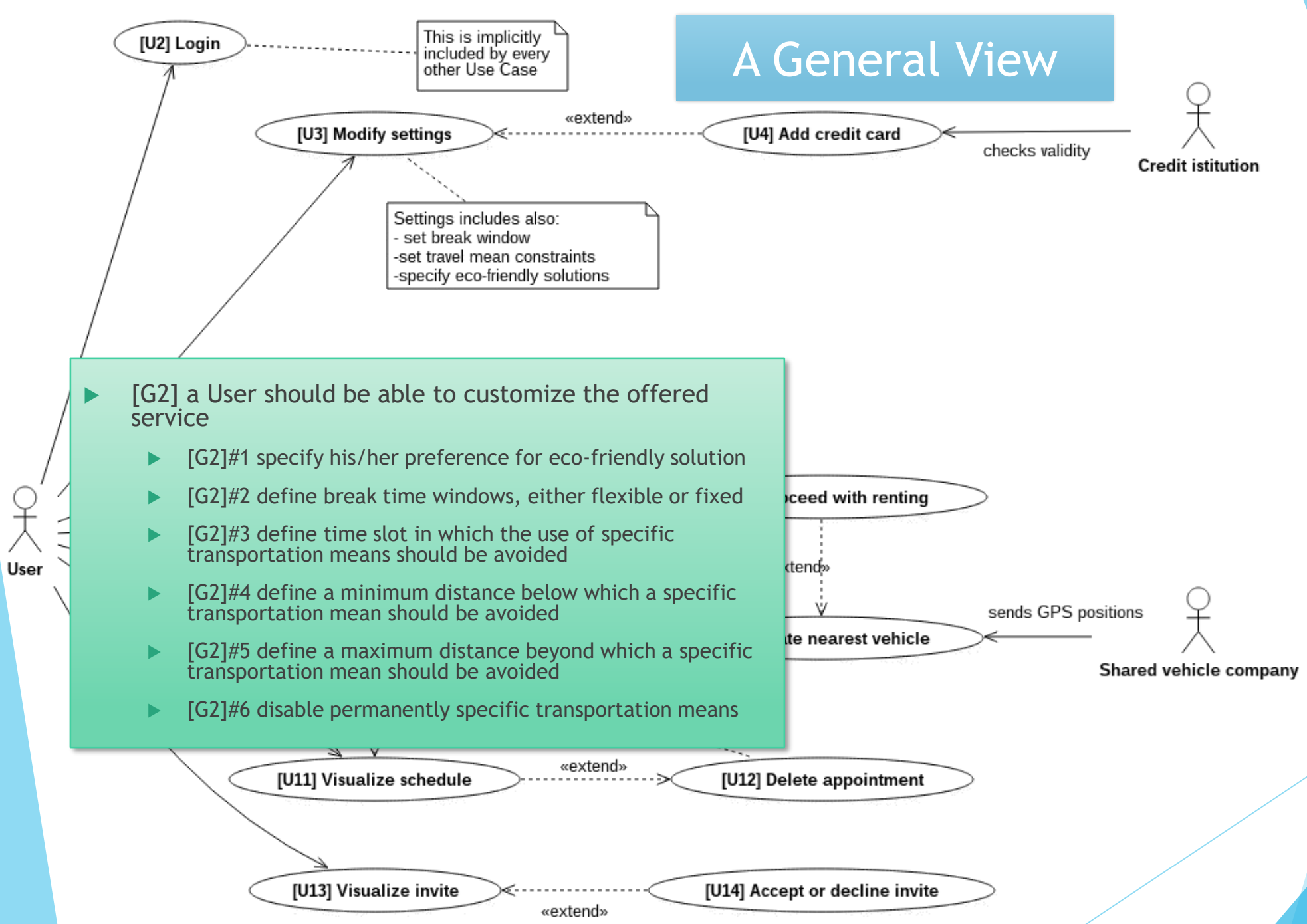
A General View



A General View

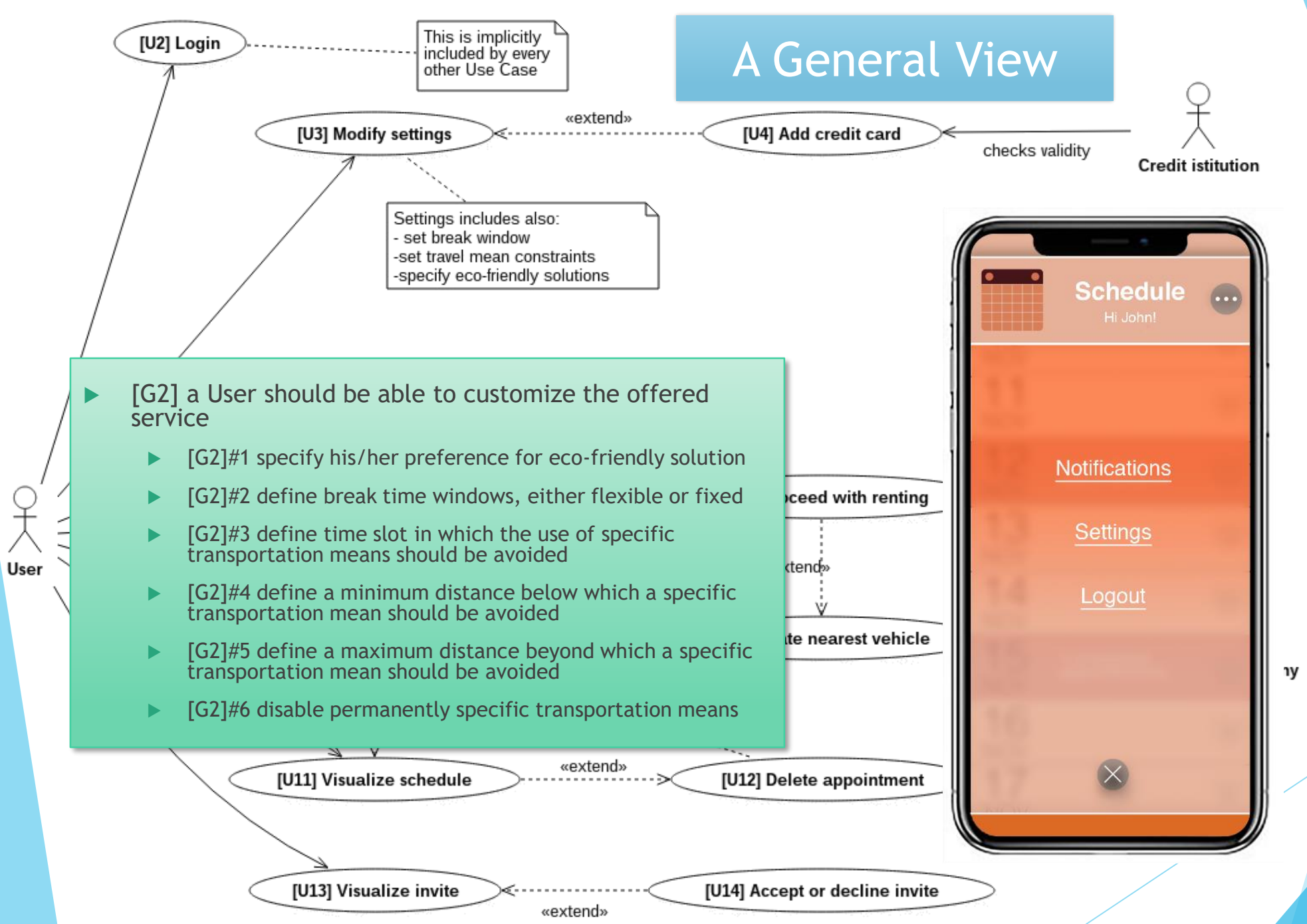


A General View



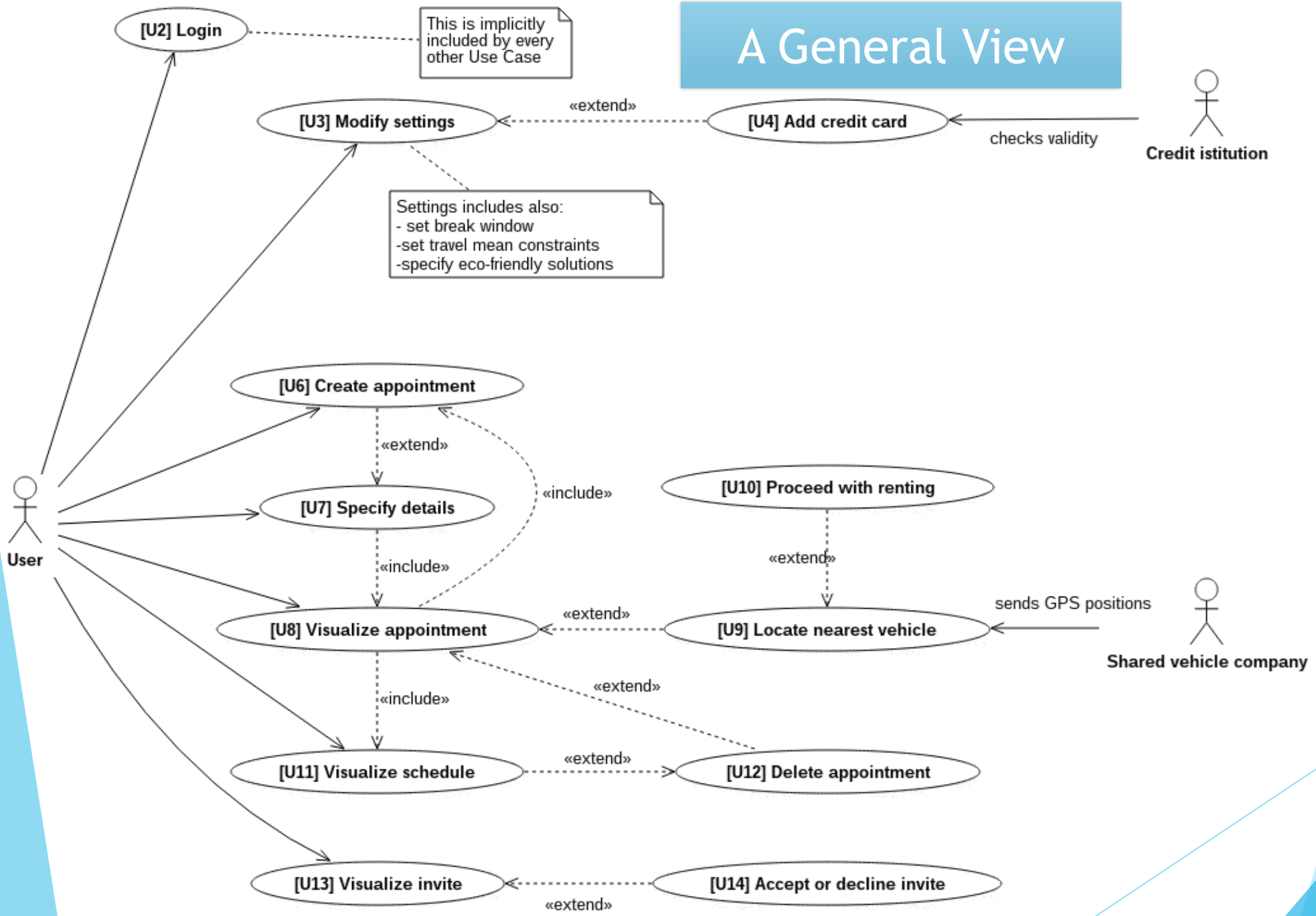
- [G2] a User should be able to customize the offered service
  - [G2]#1 specify his/her preference for eco-friendly solution
  - [G2]#2 define break time windows, either flexible or fixed
  - [G2]#3 define time slot in which the use of specific transportation means should be avoided
  - [G2]#4 define a minimum distance below which a specific transportation mean should be avoided
  - [G2]#5 define a maximum distance beyond which a specific transportation mean should be avoided
  - [G2]#6 disable permanently specific transportation means

A General View



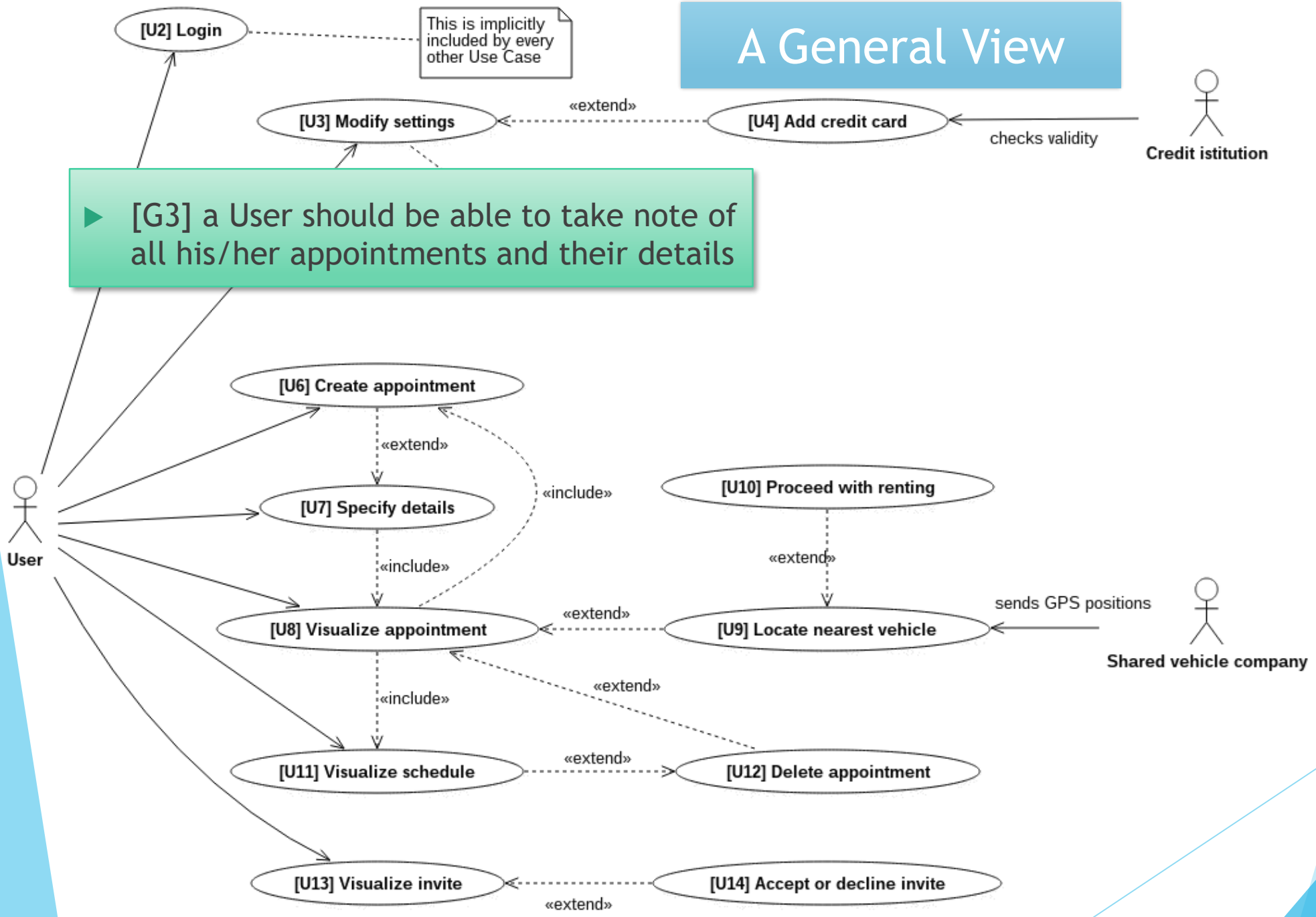
- [G2] a User should be able to customize the offered service
  - [G2]#1 specify his/her preference for eco-friendly solution
  - [G2]#2 define break time windows, either flexible or fixed
  - [G2]#3 define time slot in which the use of specific transportation means should be avoided
  - [G2]#4 define a minimum distance below which a specific transportation mean should be avoided
  - [G2]#5 define a maximum distance beyond which a specific transportation mean should be avoided
  - [G2]#6 disable permanently specific transportation means

A General View

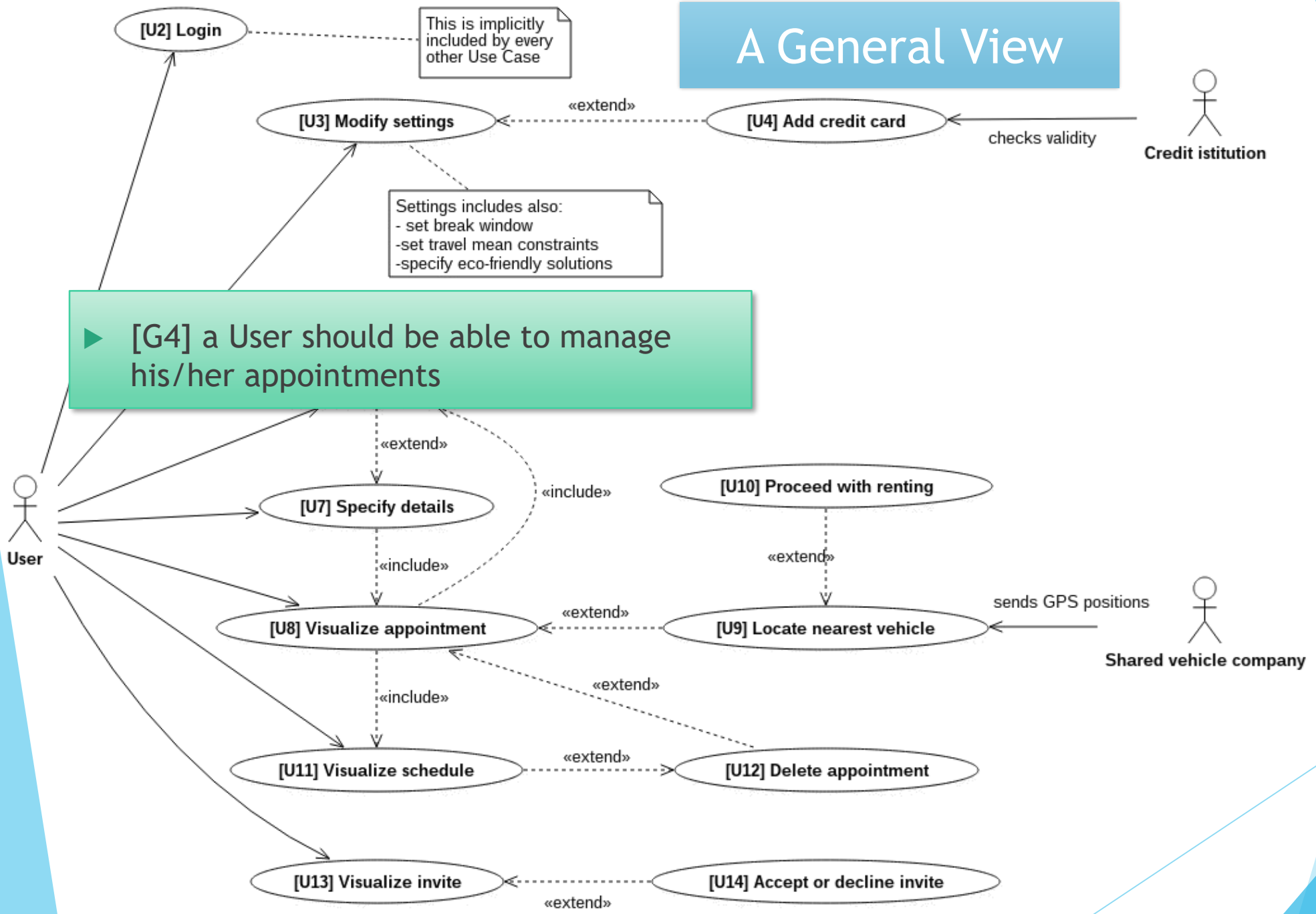


A General View

- [G3] a User should be able to take note of all his/her appointments and their details



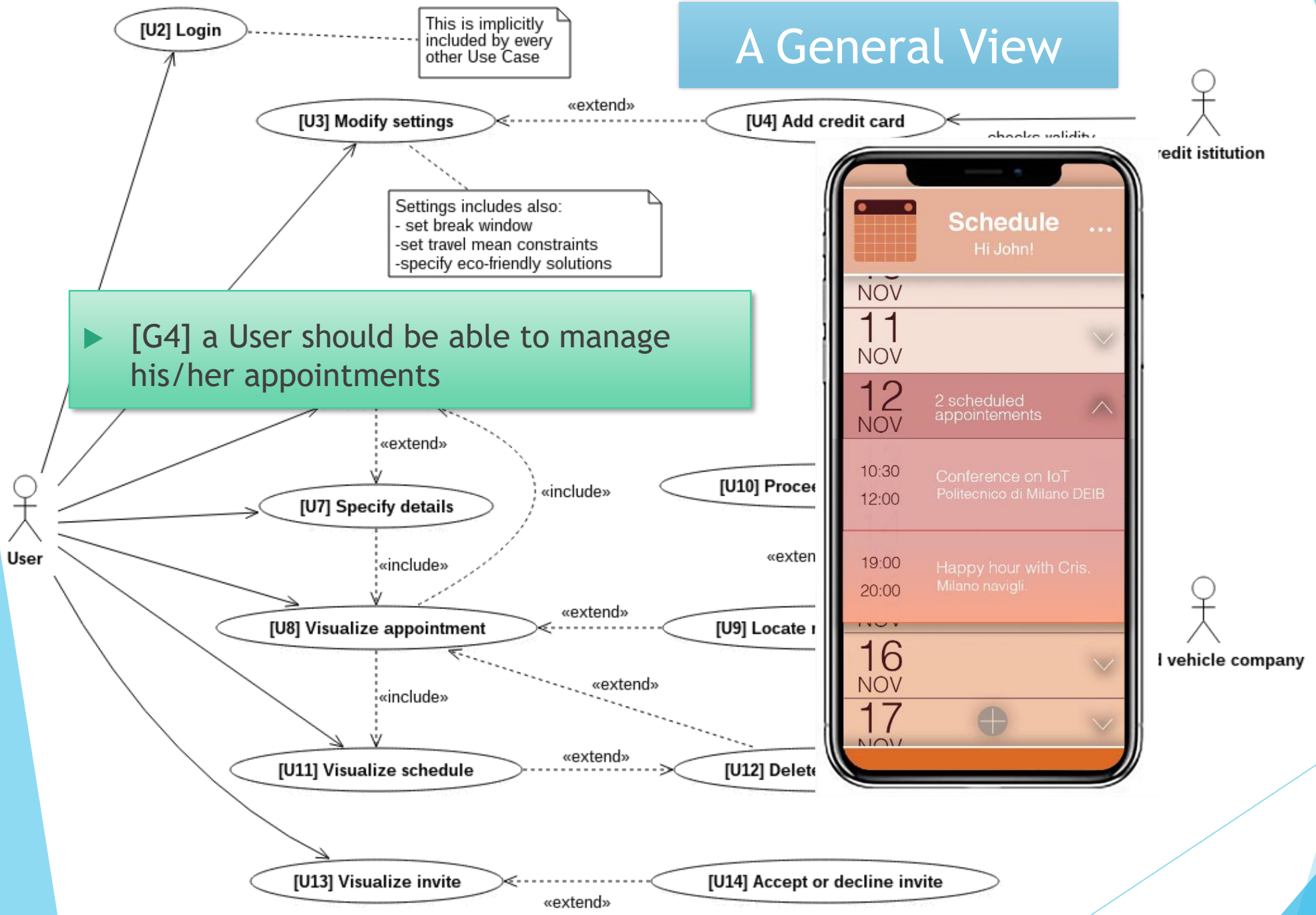
A General View



- [G4] a User should be able to manage his/her appointments

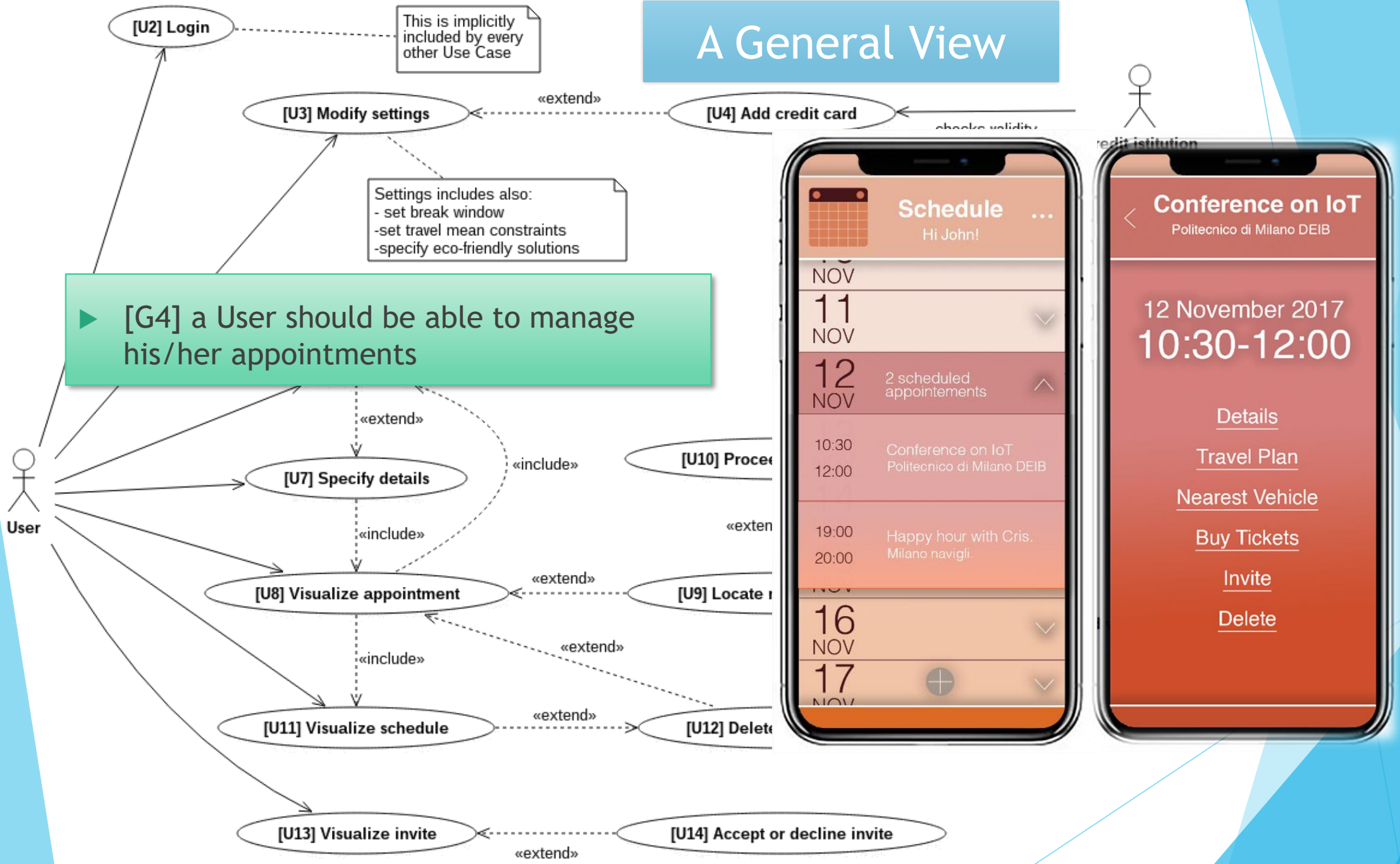
A General View

- [G4] a User should be able to manage his/her appointments

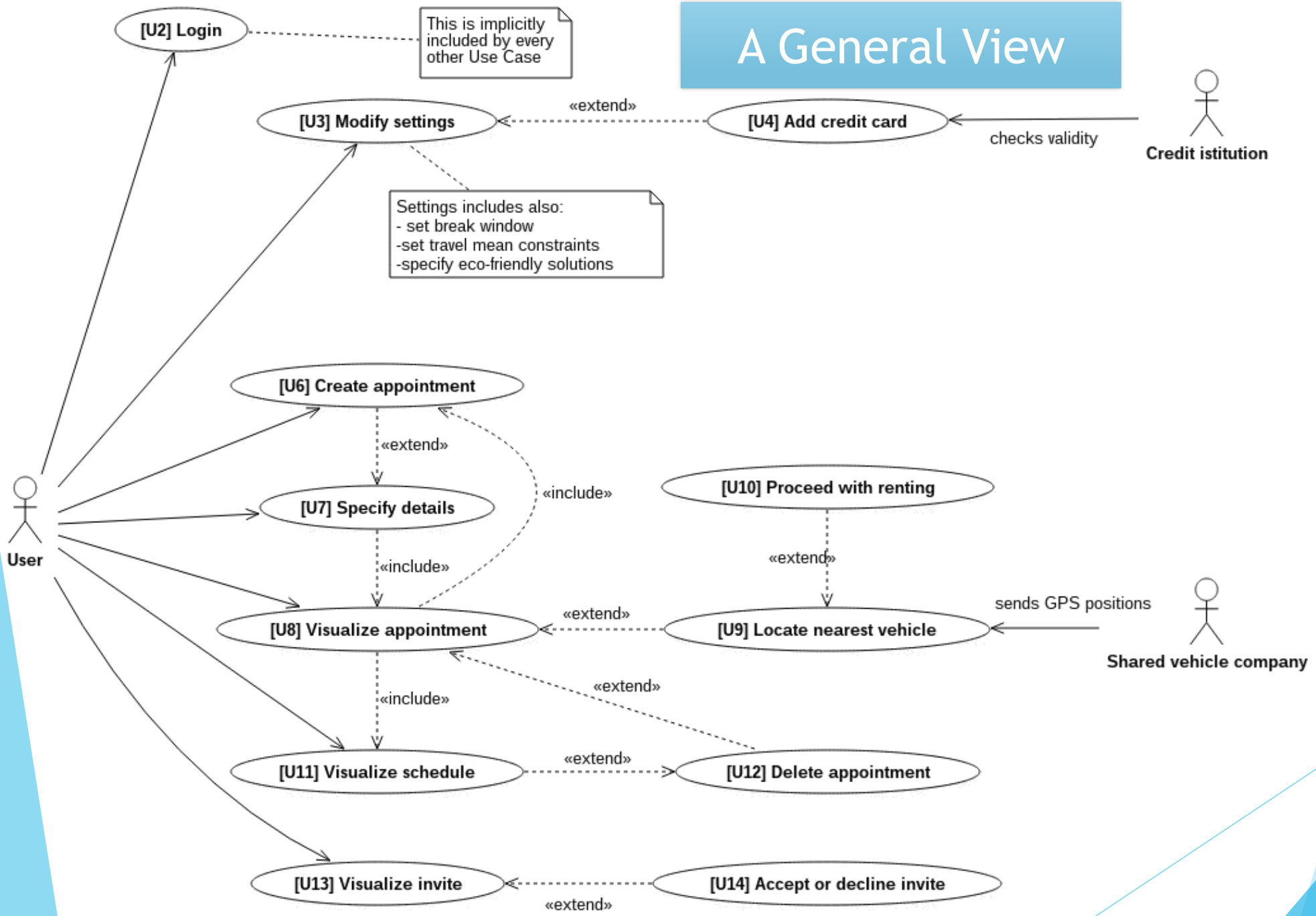


A General View

- [G4] a User should be able to manage his/her appointments

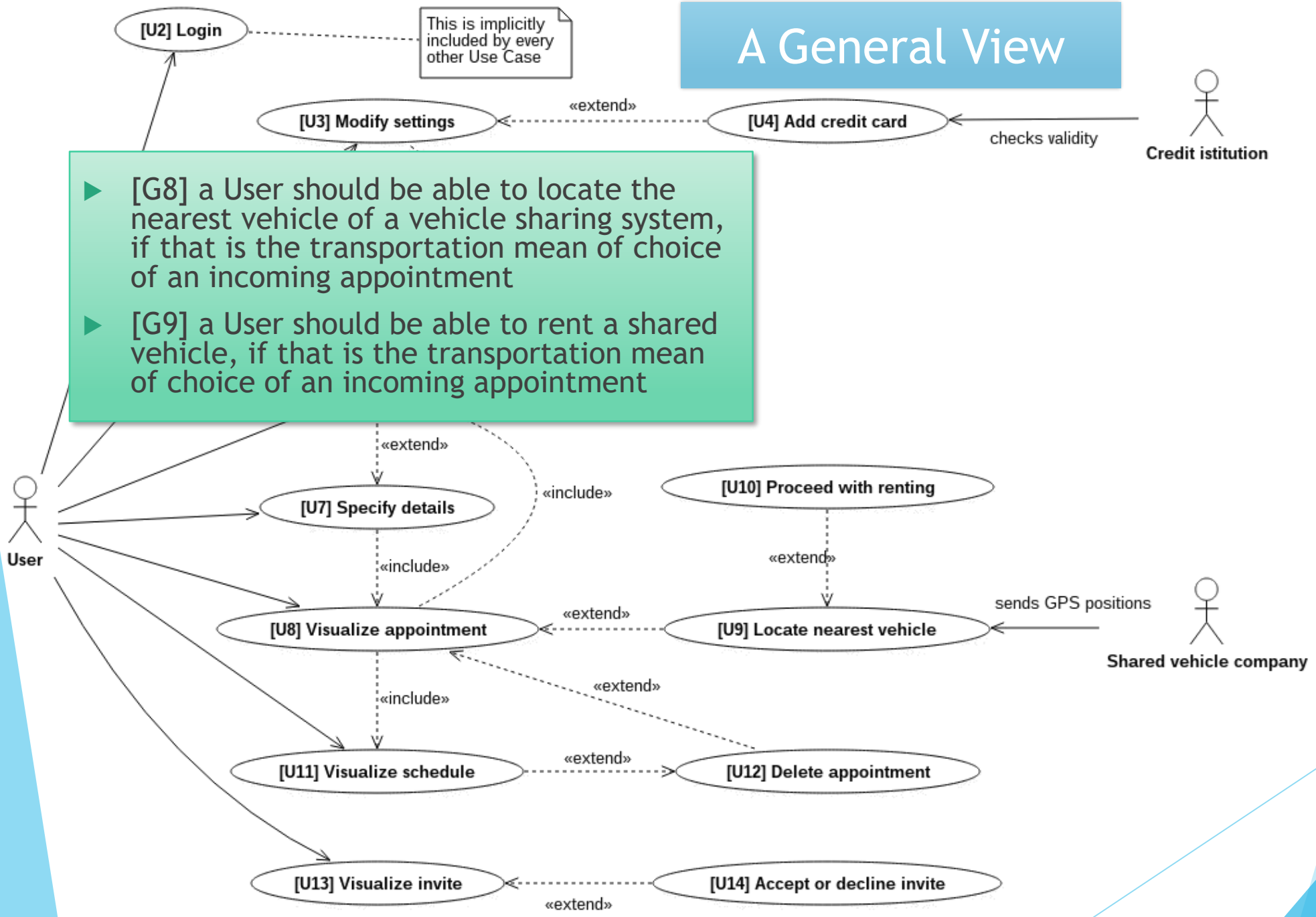


A General View



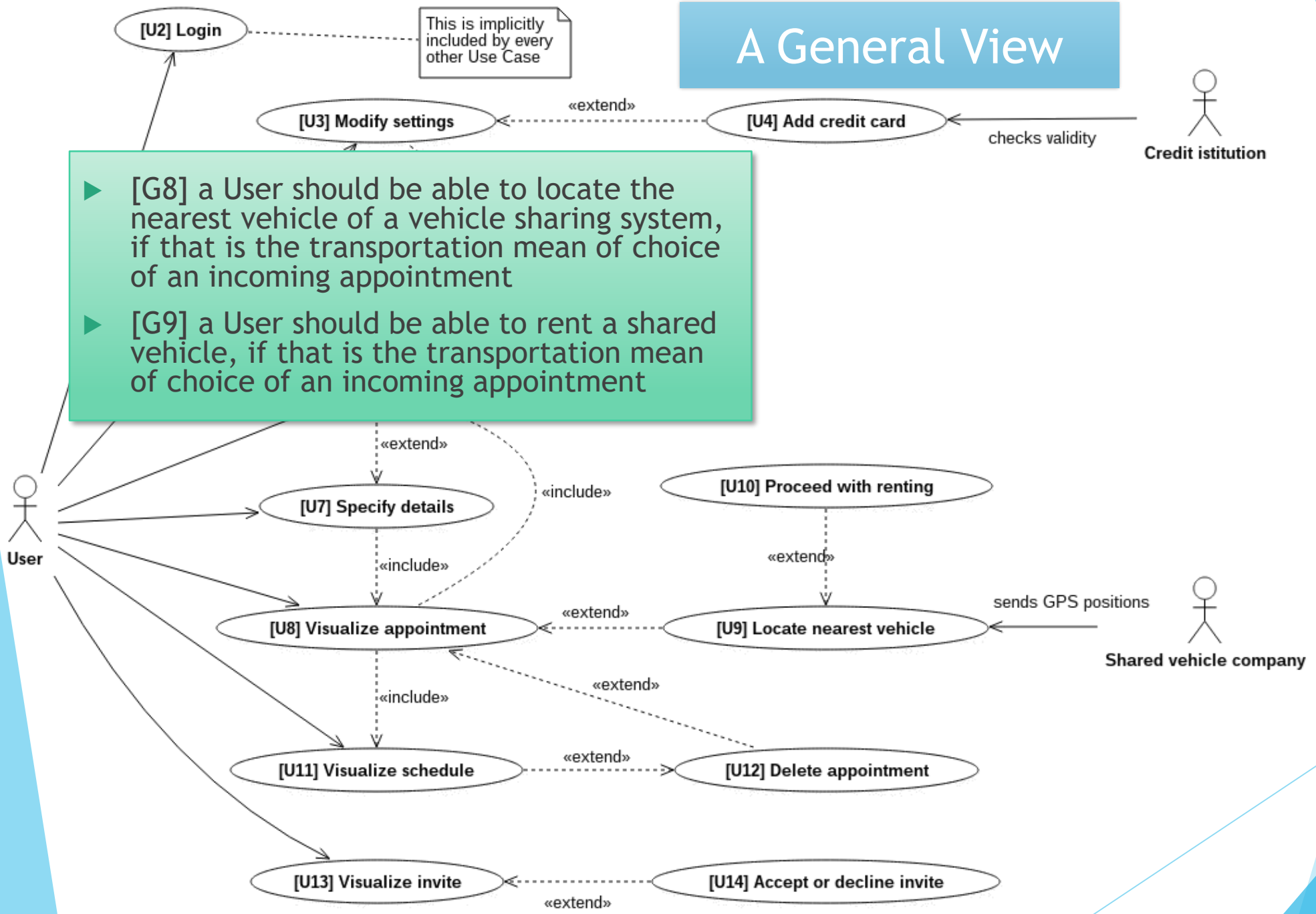
A General View

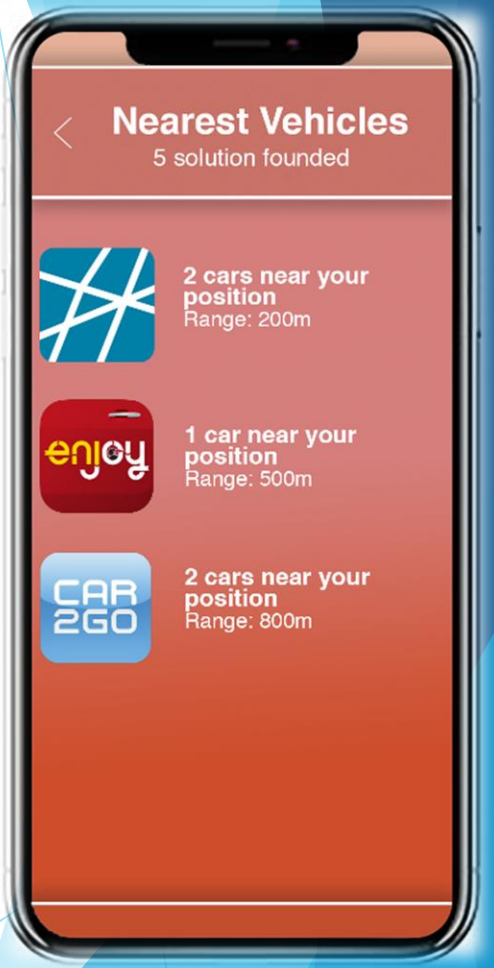
- [G8] a User should be able to locate the nearest vehicle of a vehicle sharing system, if that is the transportation mean of choice of an incoming appointment
- [G9] a User should be able to rent a shared vehicle, if that is the transportation mean of choice of an incoming appointment



A General View

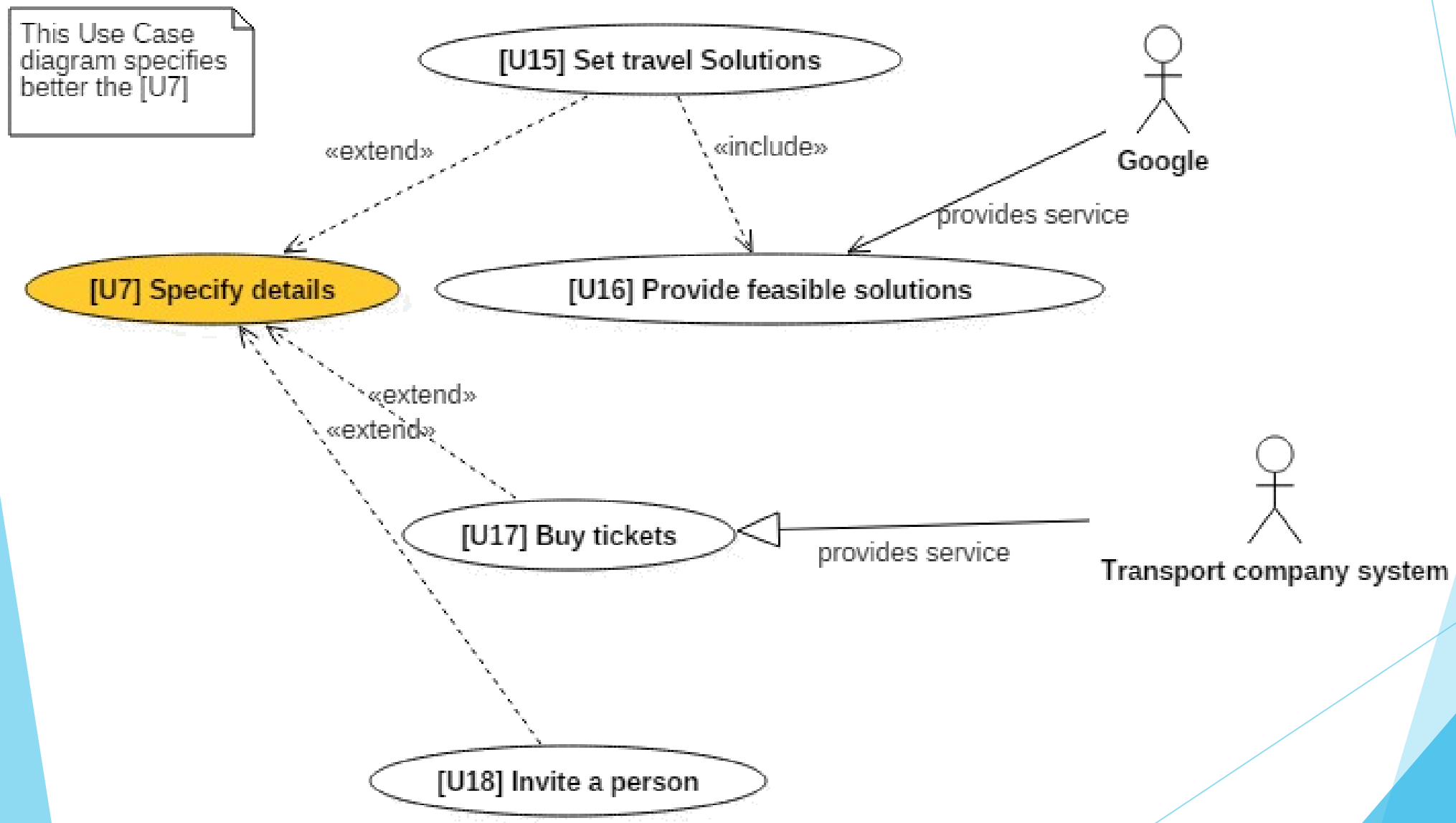
- [G8] a User should be able to locate the nearest vehicle of a vehicle sharing system, if that is the transportation mean of choice of an incoming appointment
- [G9] a User should be able to rent a shared vehicle, if that is the transportation mean of choice of an incoming appointment





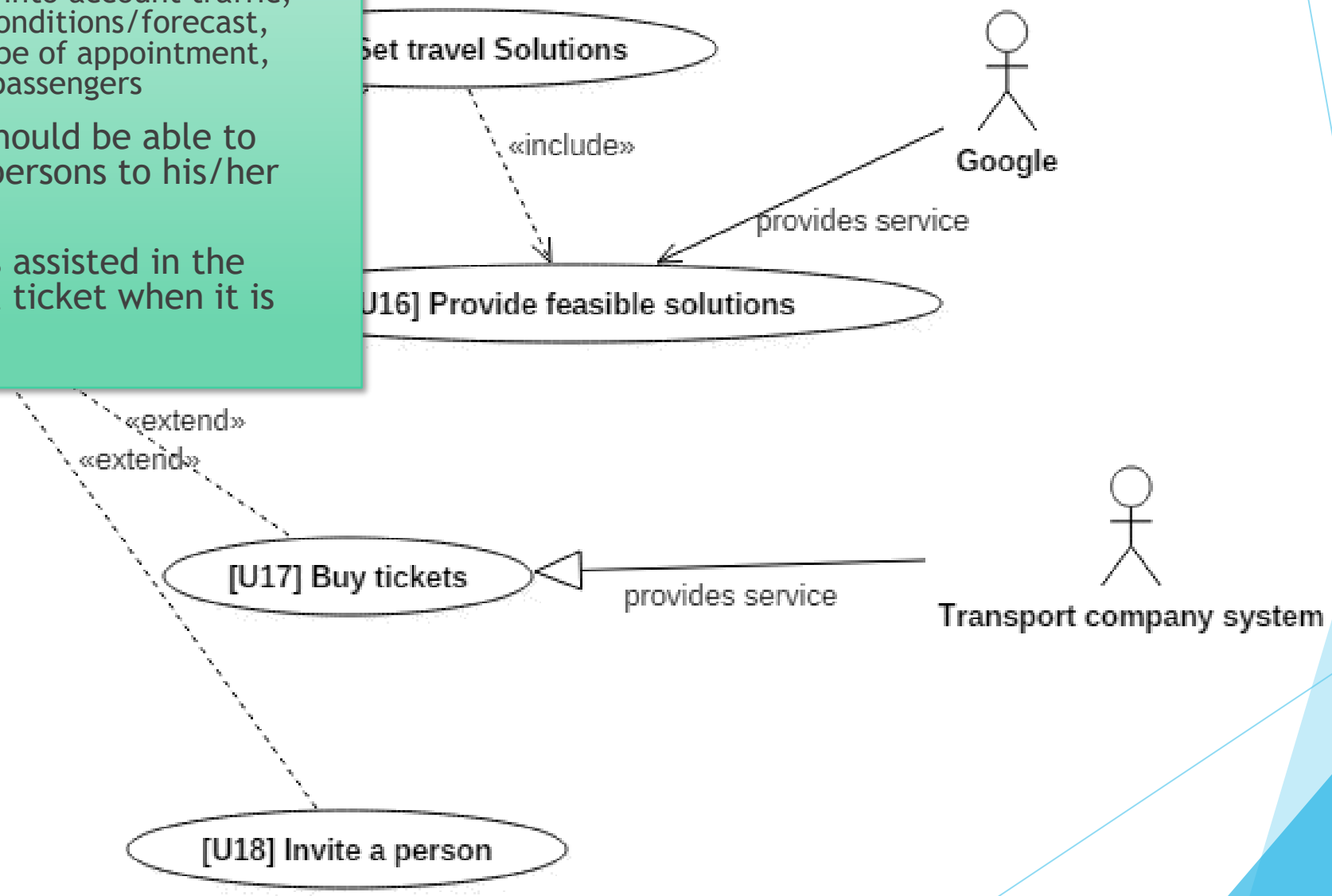
An insight on ‘Specify Details’

An insight on ‘Specify Details’



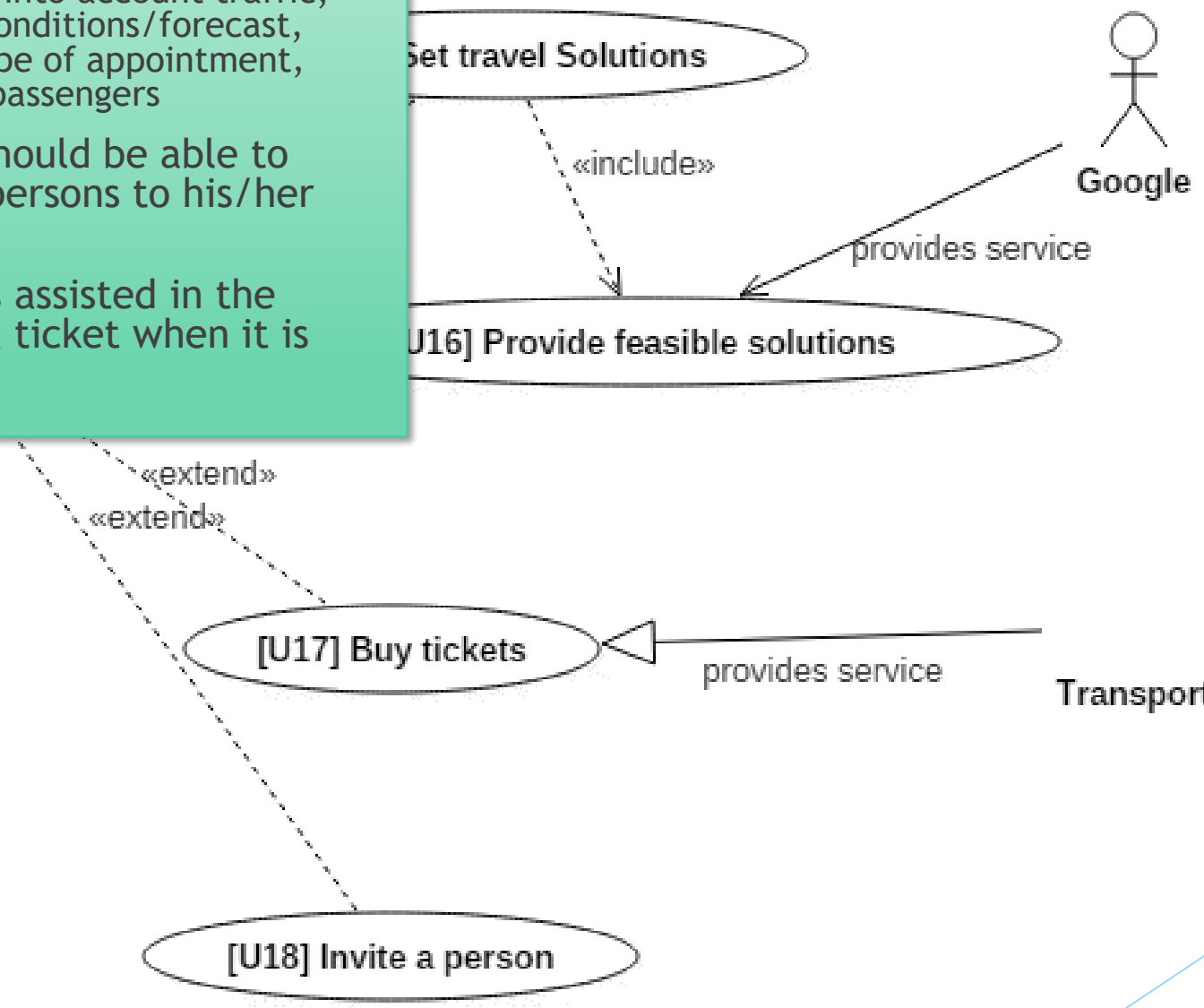
An insight on ‘Specify Details’

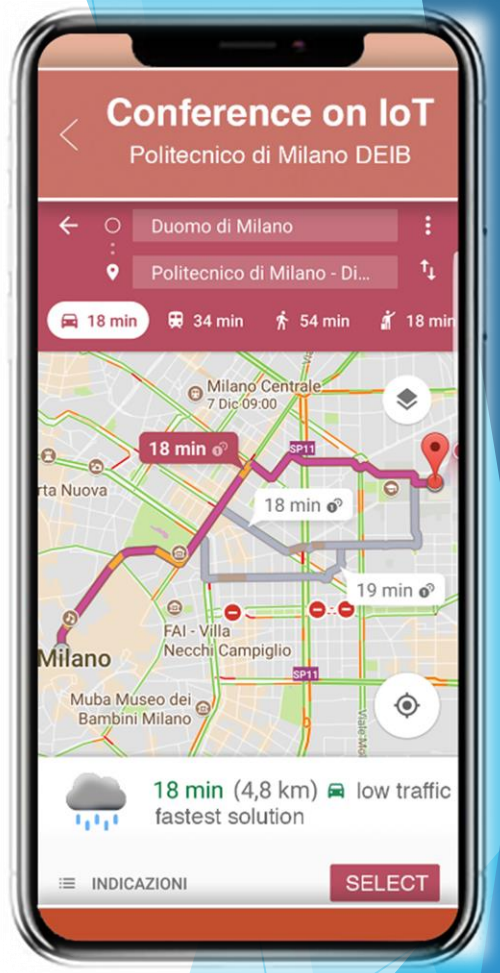
- [G5] for each appointment, the User should be assisted in the choice of the travel solution
  - [G5]#1 travel solution suggestions must take into account traffic, weather conditions/forecast, strikes, type of appointment, baggage, passengers
- [G6] a User should be able to invite other persons to his/her appointment
- [G7] a User is assisted in the purchase of a ticket when it is required



An insight on ‘Specify Details’

- [G5] for each appointment, the User should be assisted in the choice of the travel solution
  - [G5]#1 travel solution suggestions must take into account traffic, weather conditions/forecast, strikes, type of appointment, baggage, passengers
- [G6] a User should be able to invite other persons to his/her appointment
- [G7] a User is assisted in the purchase of a ticket when it is required





Architecture Overview

Architecture Overview

- Client/server

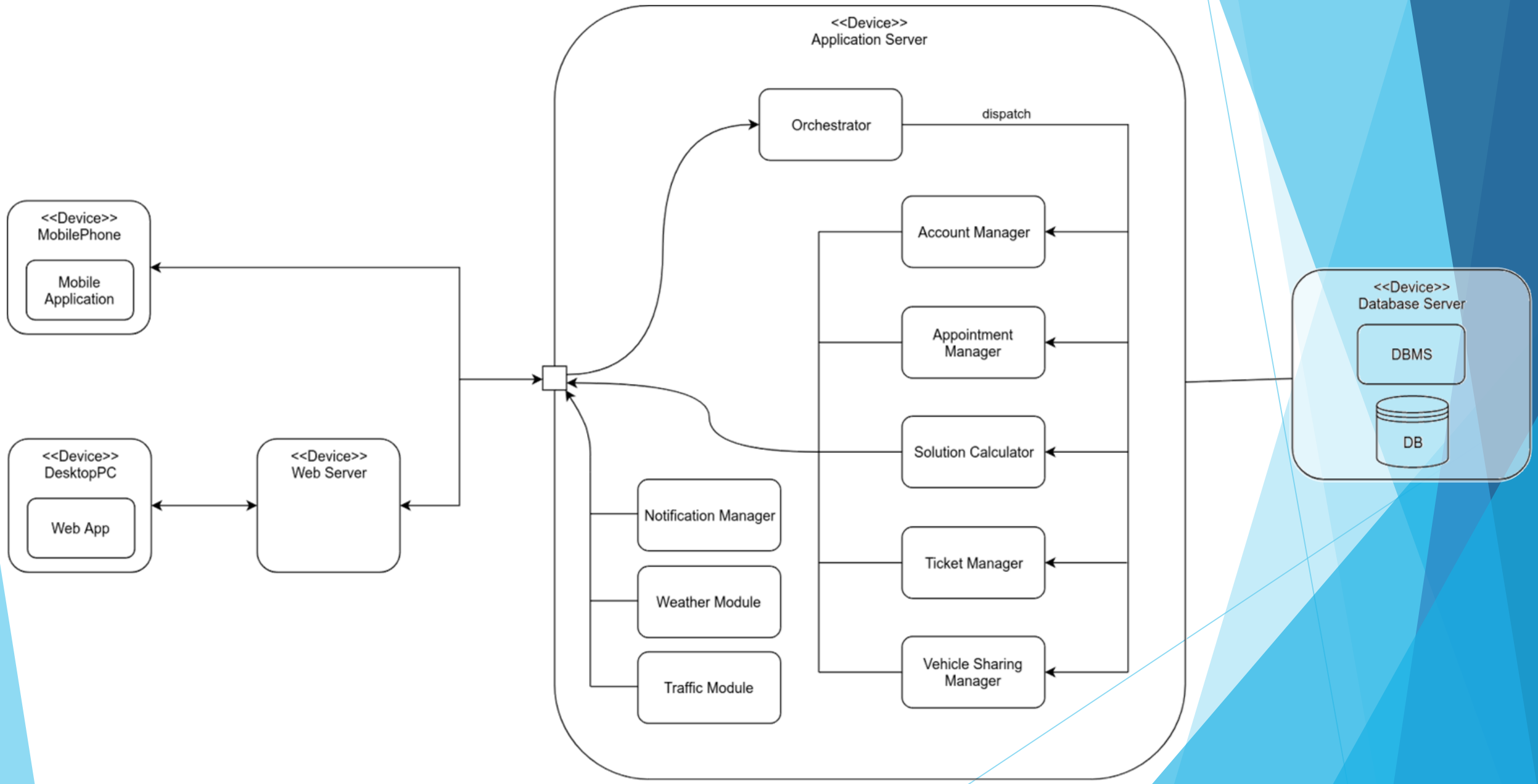
Architecture Overview

- Client/server
- Service Oriented Architecture

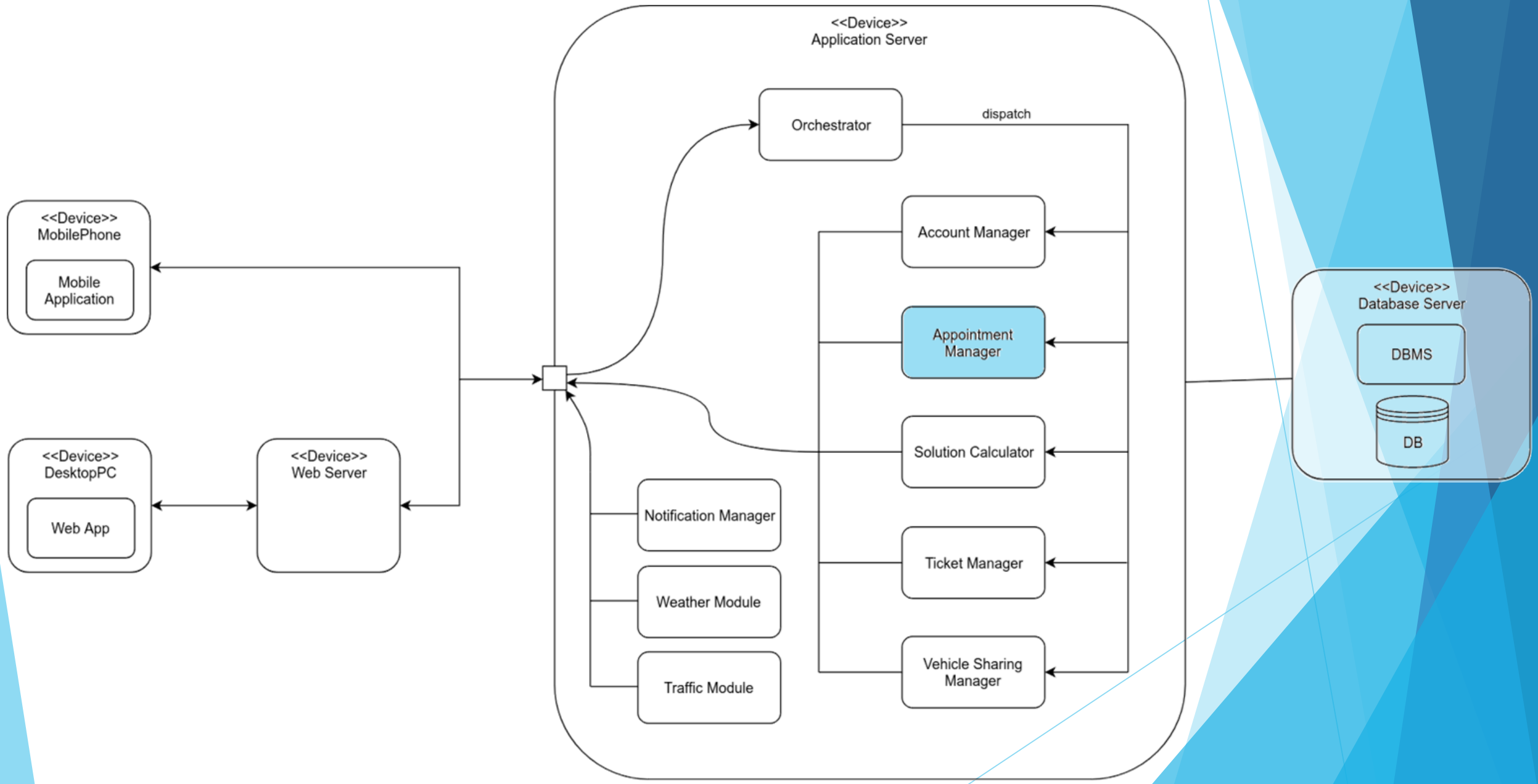
Architecture Overview

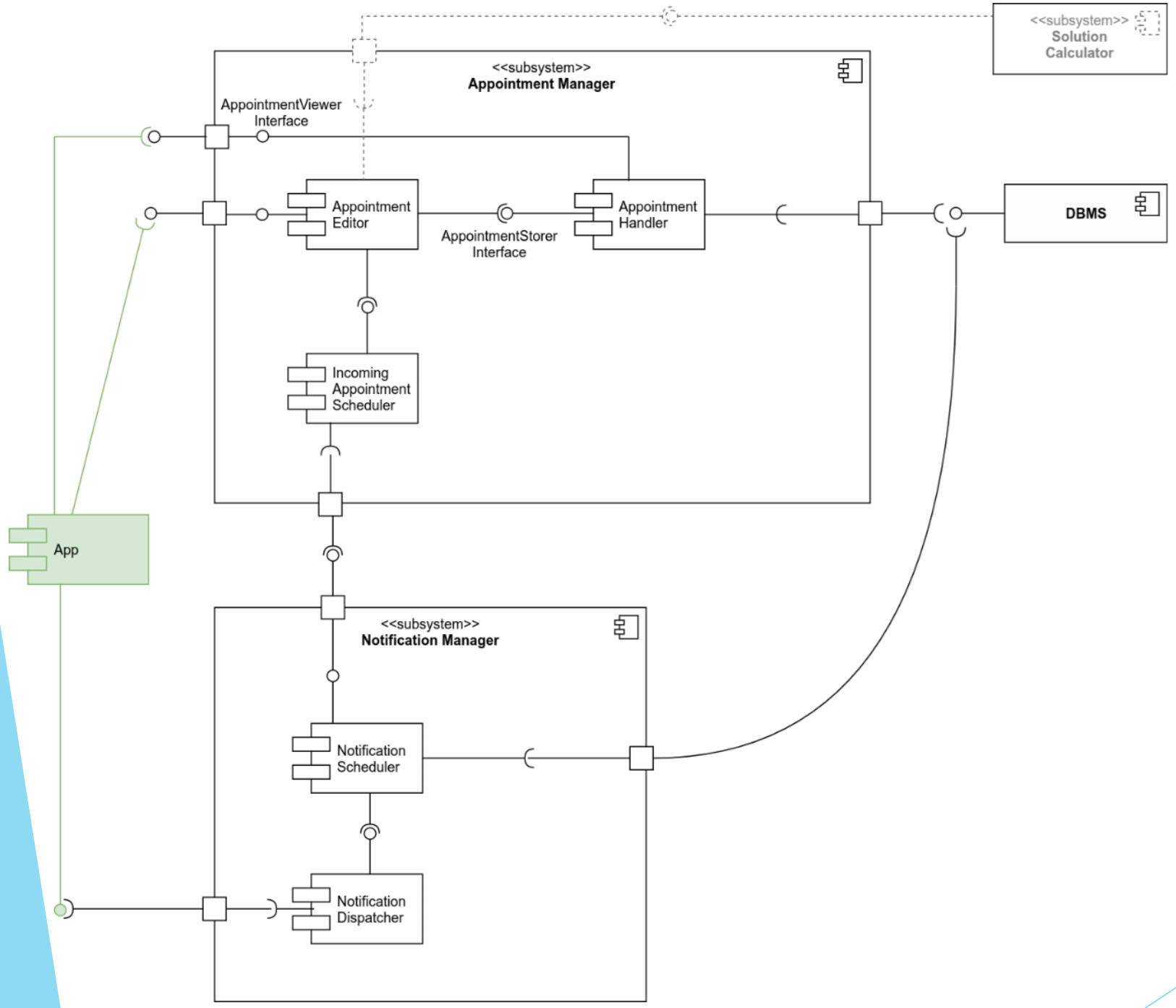
- Client/server
- Service Oriented Architecture
- Elastic Components

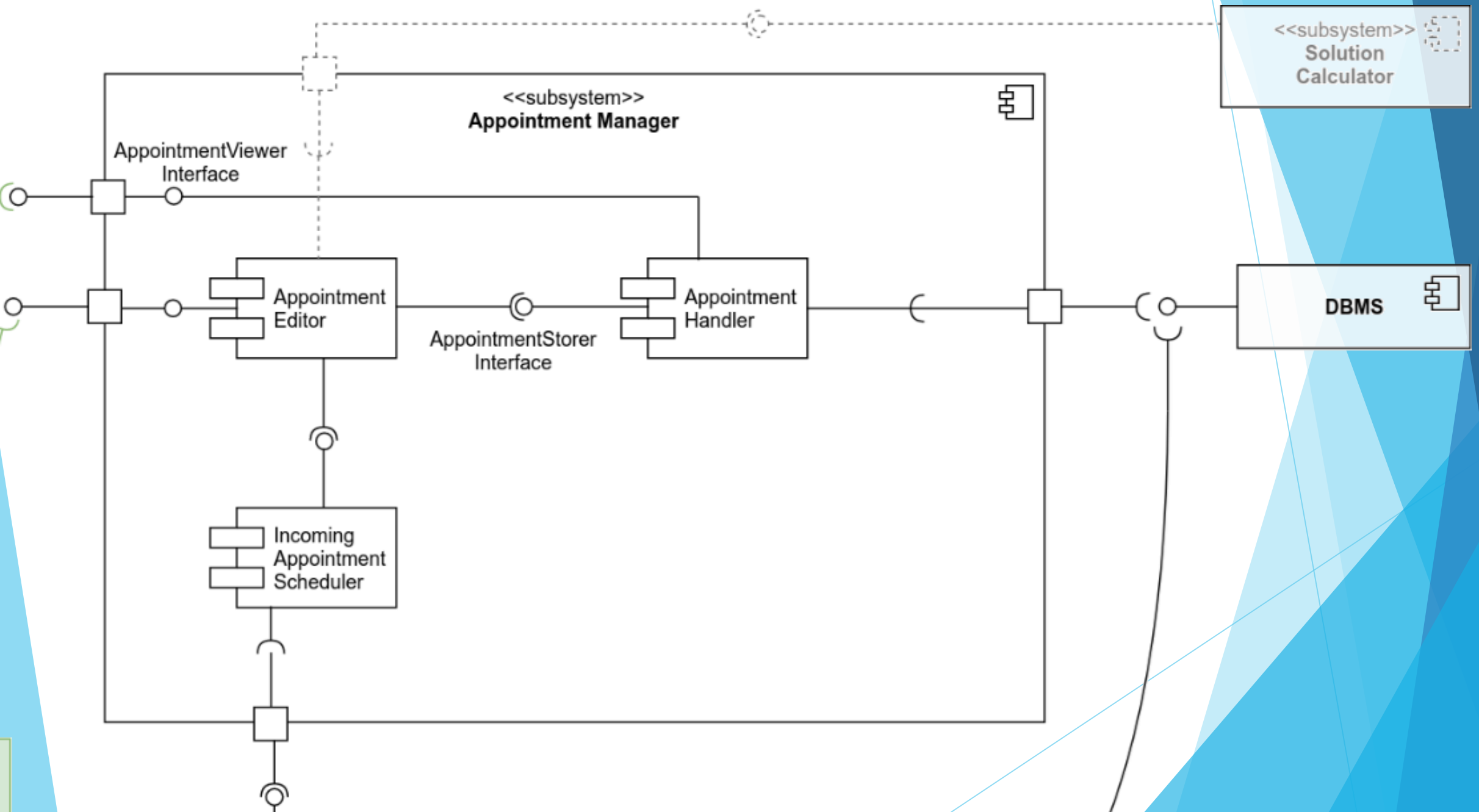
Architecture Overview

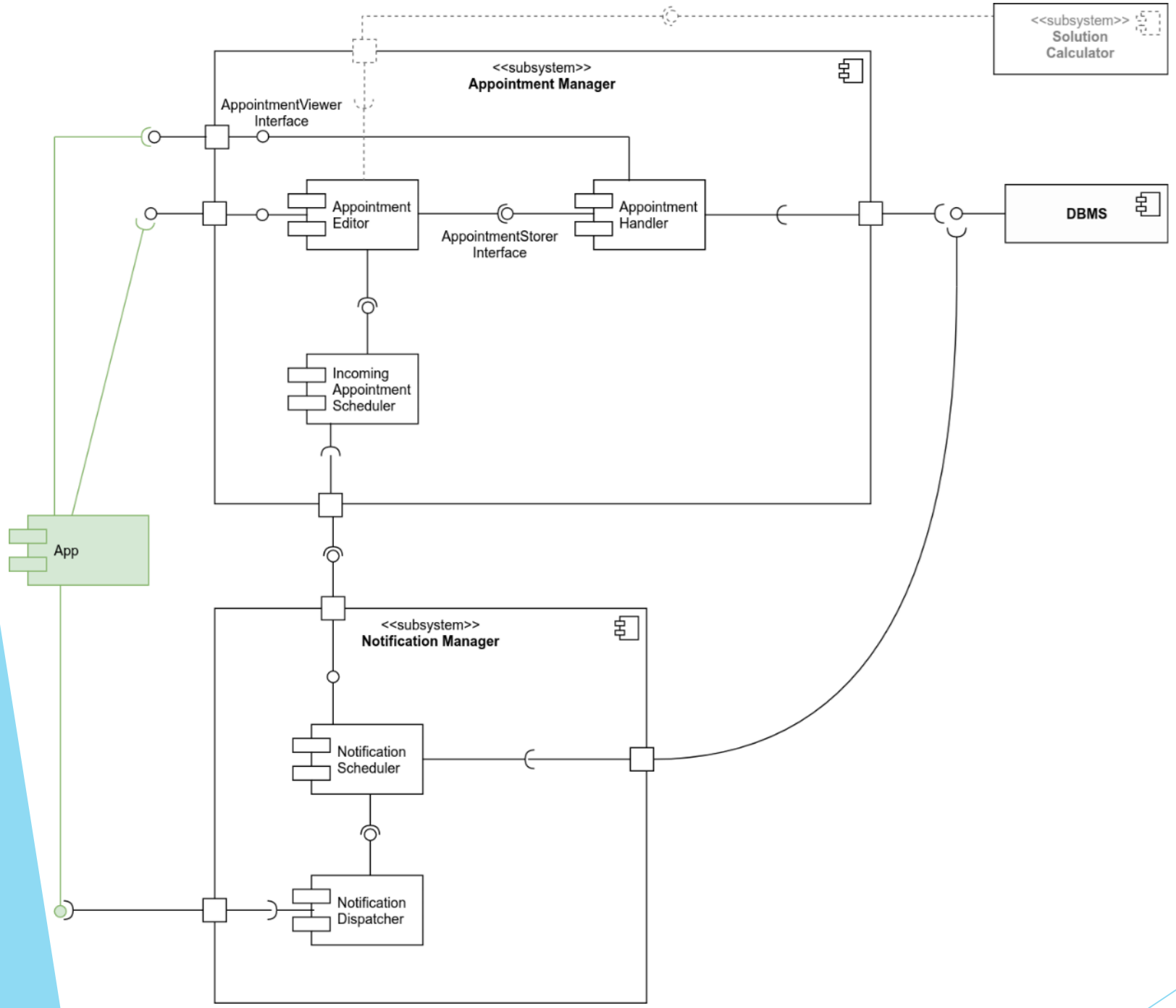


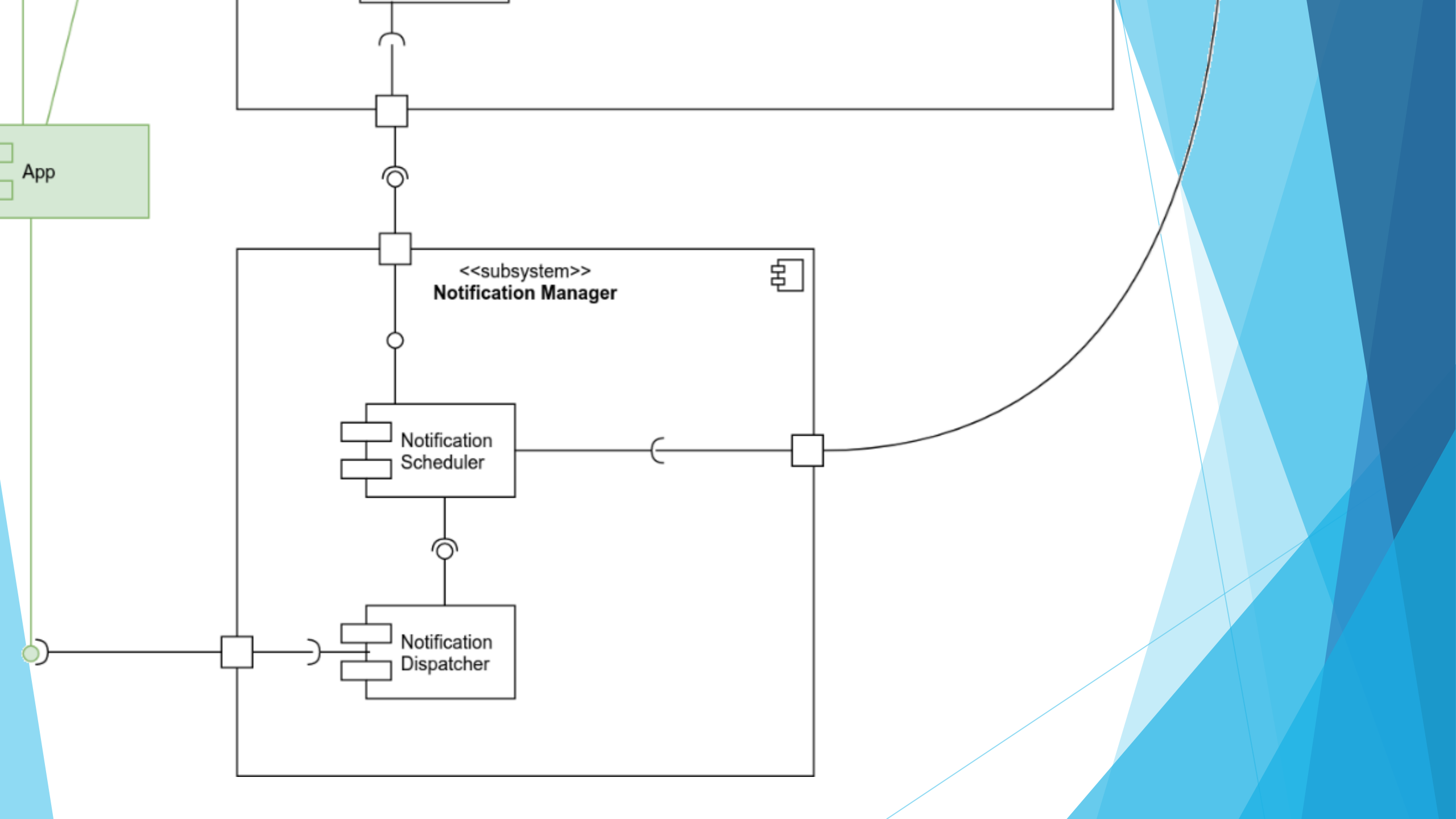
Architecture Overview

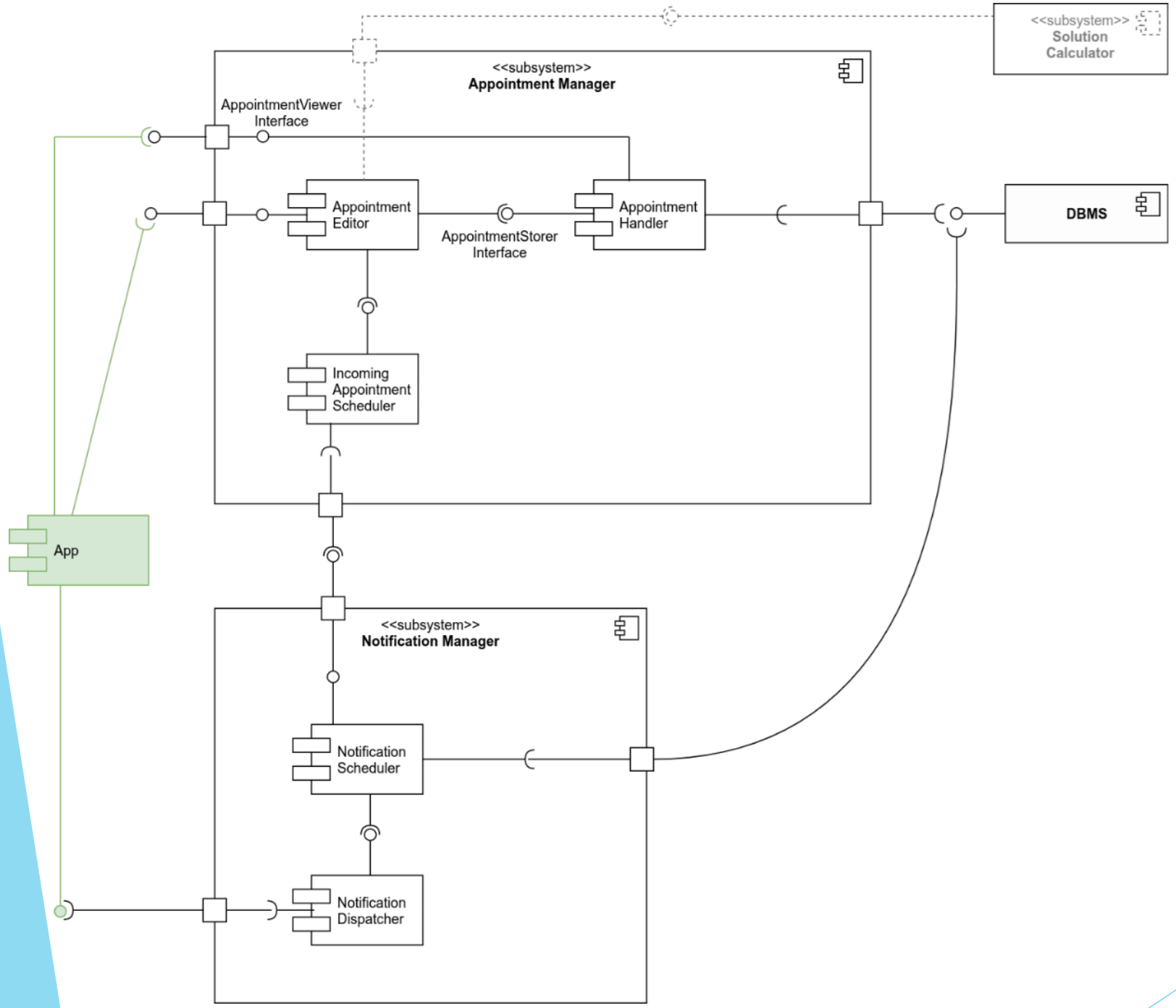



<<subsystem>>
Solution
Calculator
<<subsystem>>
Appointment Manager
AppointmentViewer
Interface
Appointment
Editor
AppointmentStorer
Interface
Appointment
Handler
DBMS
Incoming
Appointment
Scheduler
App
<<subsystem>>
Notification Manager
Notification
Scheduler
Notification
Dispatcher

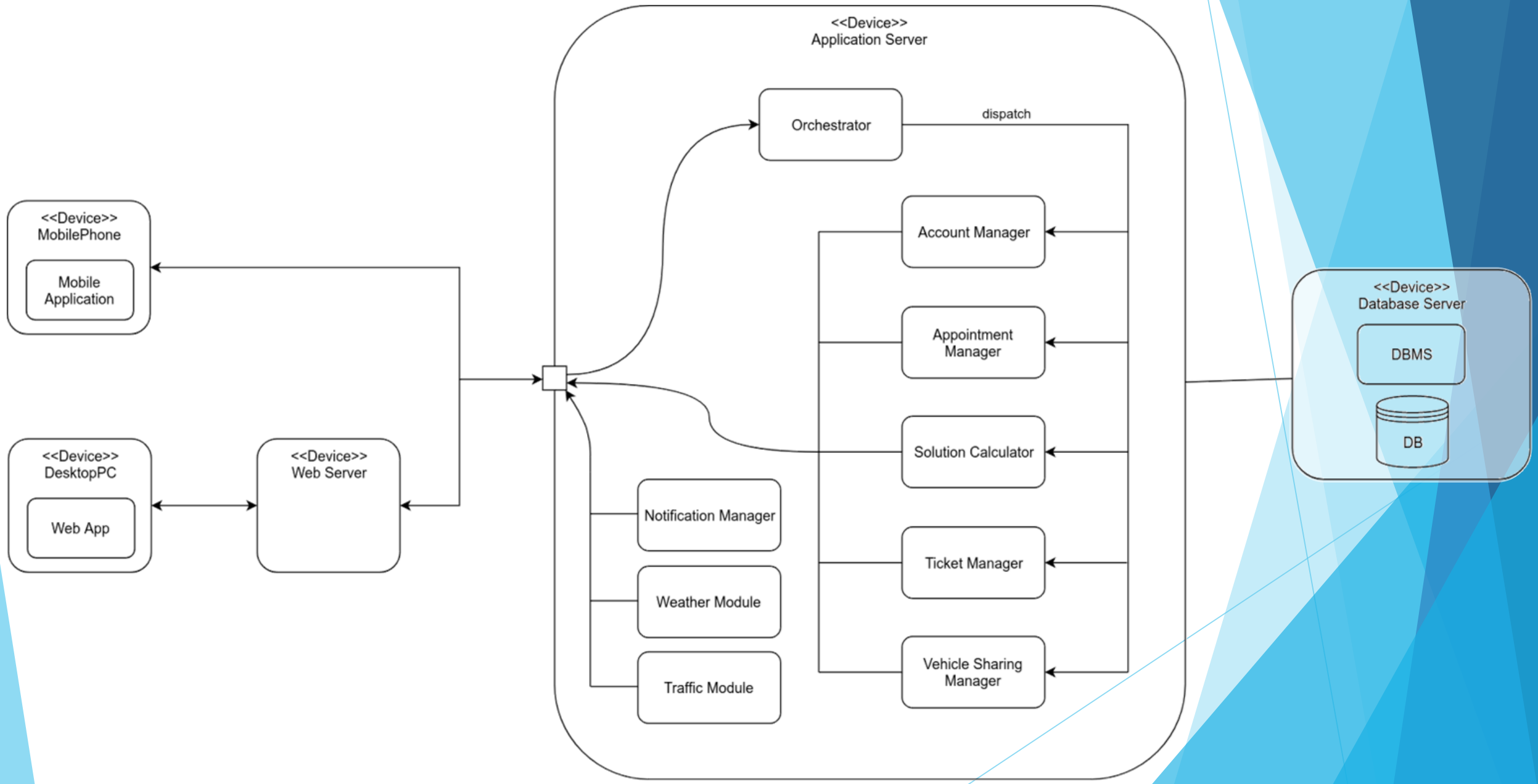

<<subsystem>>
Solution
Calculator
<<subsystem>>
Appointment Manager
AppointmentViewer
Interface
Appointment
Editor
AppointmentStorer
Interface
Appointment
Handler
DBMS
Incoming
Appointment
Scheduler


<<subsystem>>
Solution
Calculator
<<subsystem>>
Appointment Manager
AppointmentViewer
Interface
Appointment
Editor
AppointmentStorer
Interface
Appointment
Handler
DBMS
Incoming
Appointment
Scheduler
App
<<subsystem>>
Notification Manager
Notification
Scheduler
Notification
Dispatcher

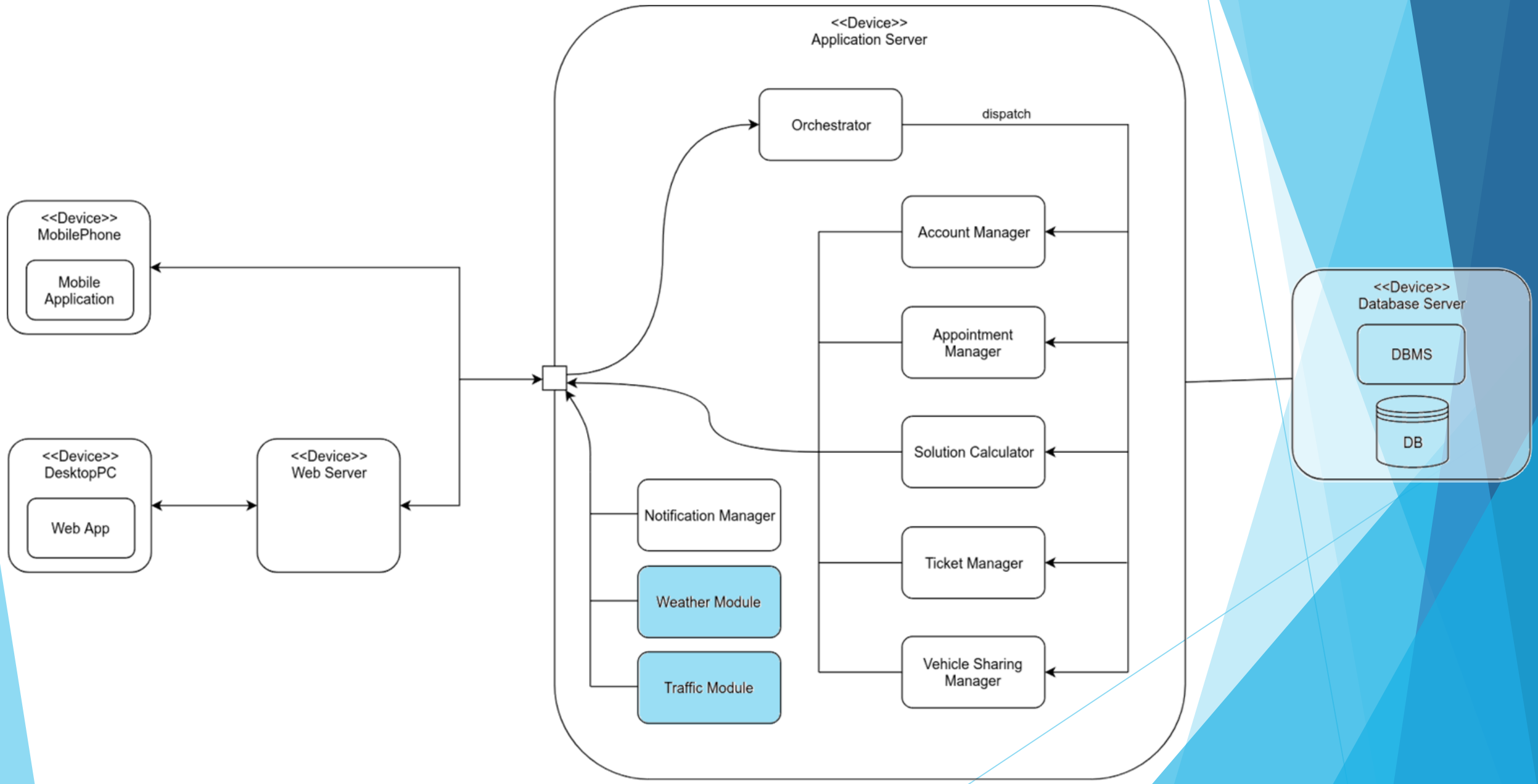

App
<<subsystem>>
Notification Manager
Notification
Scheduler
Notification
Dispatcher

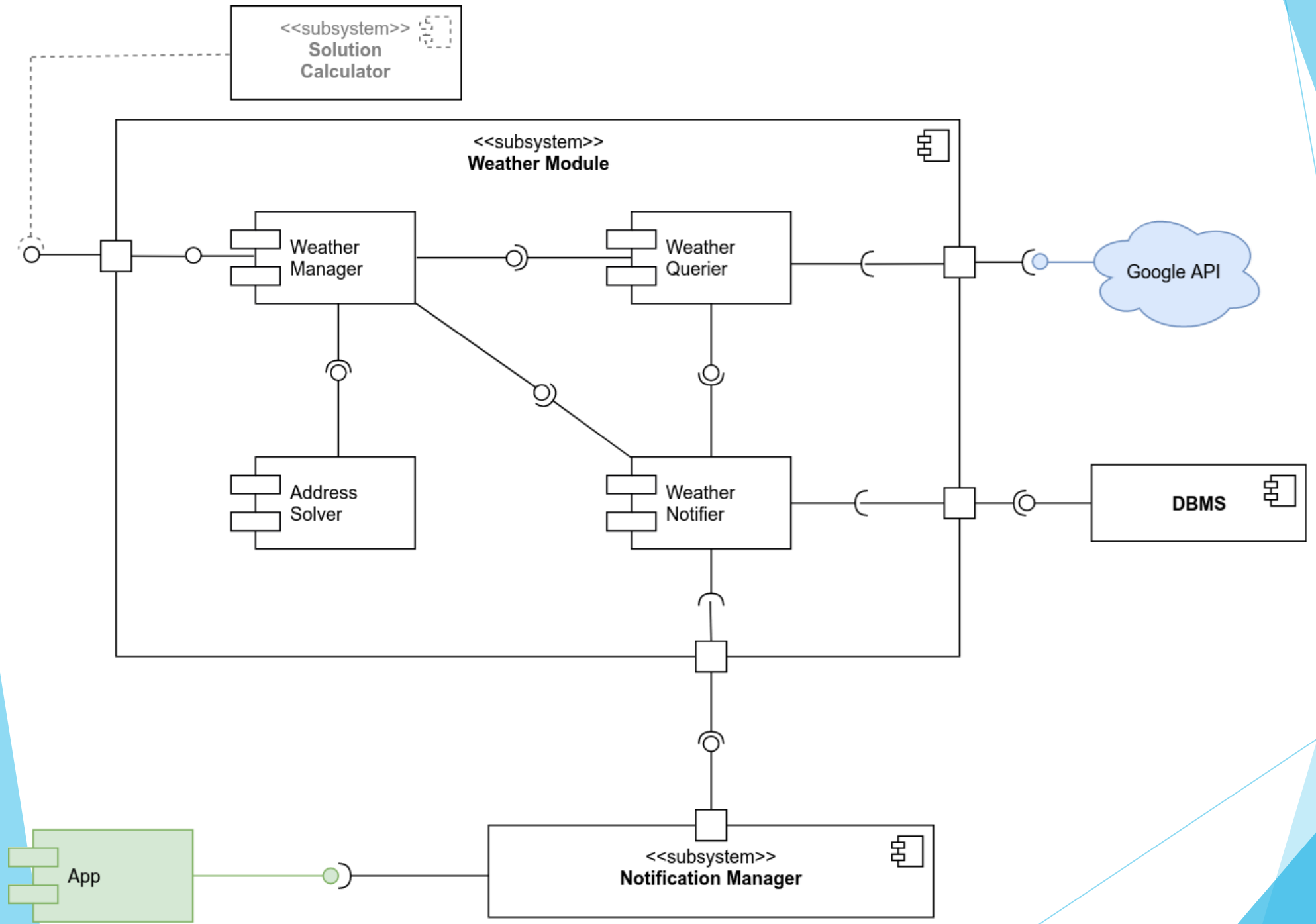

<<subsystem>>
Solution
Calculator
<<subsystem>>
Appointment Manager
AppointmentViewer
Interface
Appointment
Editor
AppointmentStorer
Interface
Appointment
Handler
DBMS
Incoming
Appointment
Scheduler
App
<<subsystem>>
Notification Manager
Notification
Scheduler
Notification
Dispatcher

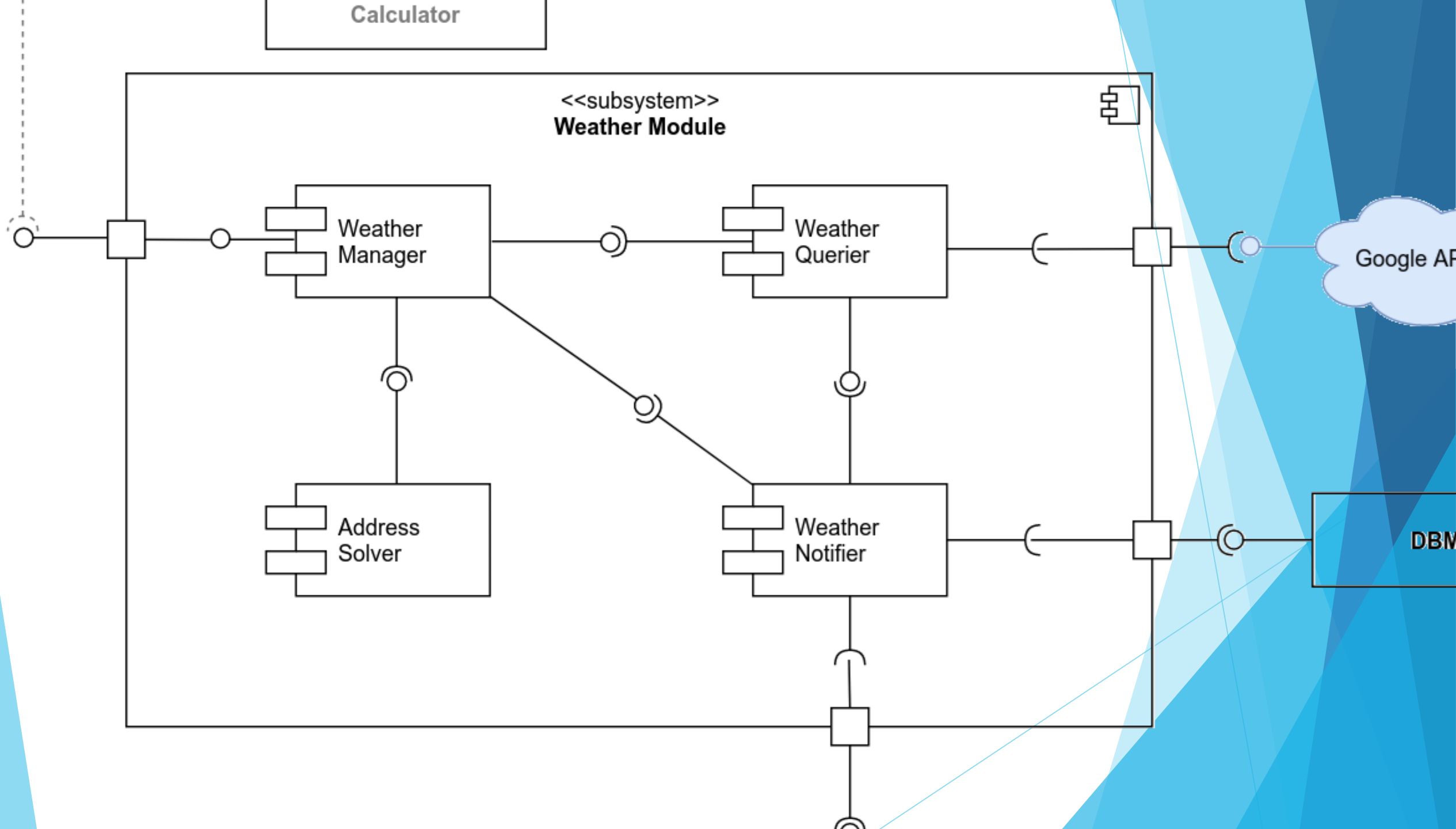
Architecture Overview

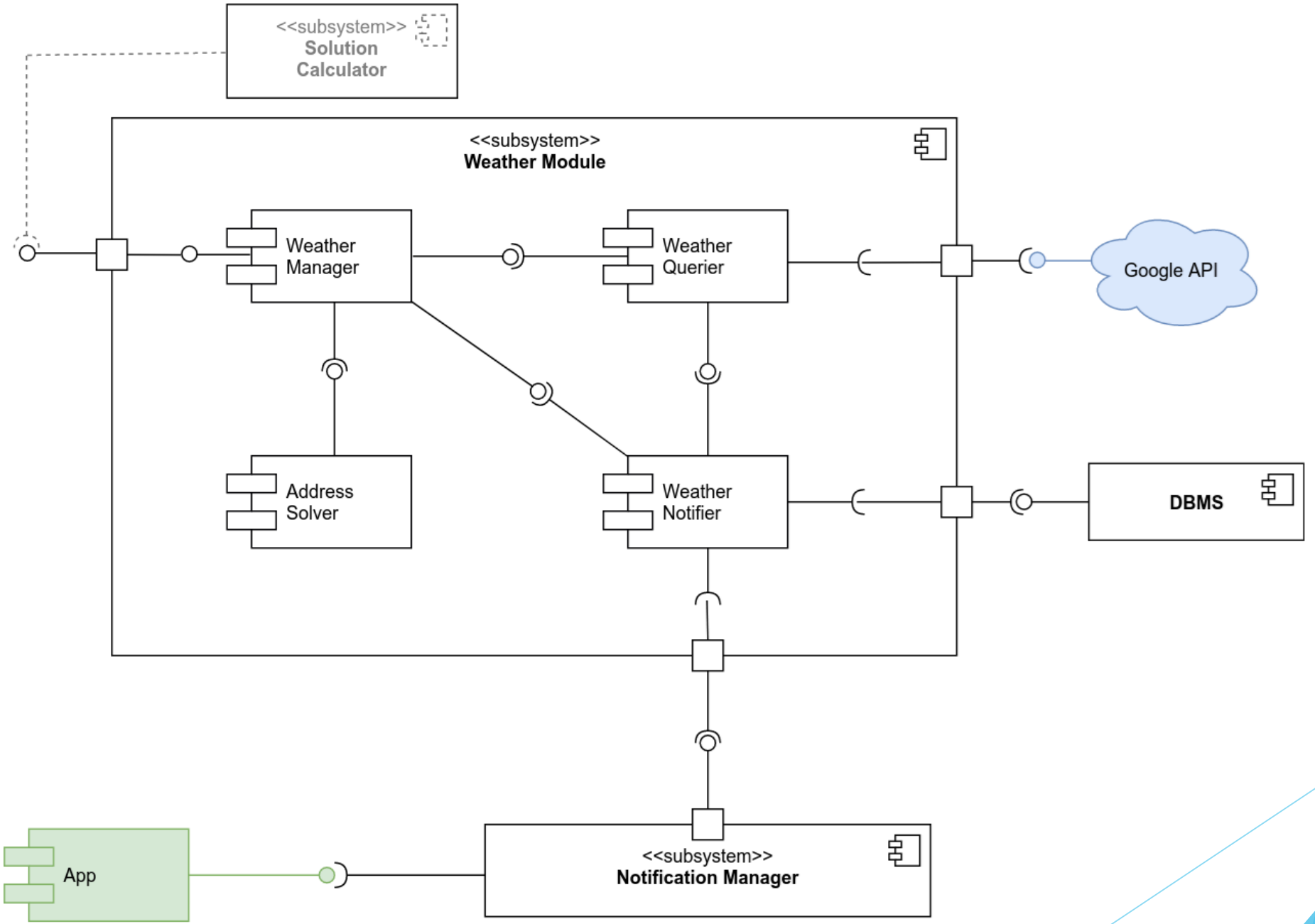


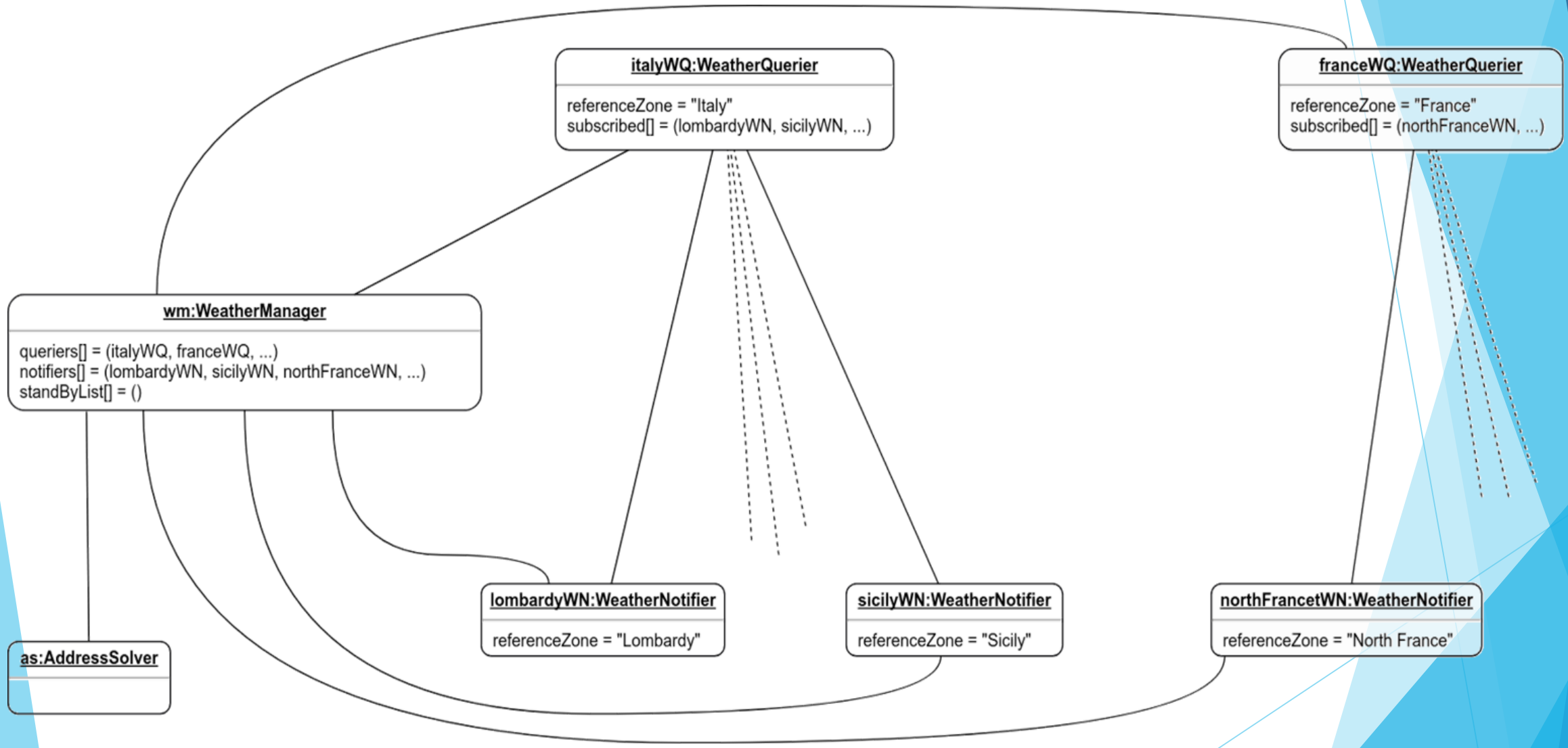
Architecture Overview




<<subsystem>>
Solution
Calculator
<<subsystem>>
Weather Module
Weather
Manager
Weather
Querier
Google API
Address
Solver
Weather
Notifier
DBMS
App
<<subsystem>>
Notification Manager


Calculator
<<subsystem>>
Weather Module
Weather Manager
Weather Querier
Google AP
Address Solver
Weather Notifier
DBM


<<subsystem>>
Solution
Calculator
<<subsystem>>
Weather Module
Weather
Manager
Weather
Querier
Google API
Address
Solver
Weather
Notifier
DBMS
App
<<subsystem>>
Notification Manager


italyWQ:WeatherQuerier
referenceZone = "Italy"
subscribed[] = (lombardyWN, sicilyWN, ...)
franceWQ:WeatherQuerier
referenceZone = "France"
subscribed[] = (northFranceWN, ...)
wm:WeatherManager
queriers[] = (italyWQ, franceWQ, ...)
notifiers[] = (lombardyWN, sicilyWN, northFranceWN, ...)
standByList[] = ()
lombardyWN:WeatherNotifier
referenceZone = "Lombardy"
sicilyWN:WeatherNotifier
referenceZone = "Sicily"
northFrancetWN:WeatherNotifier
referenceZone = "North France"
as:AddressSolver

Solution Calculation

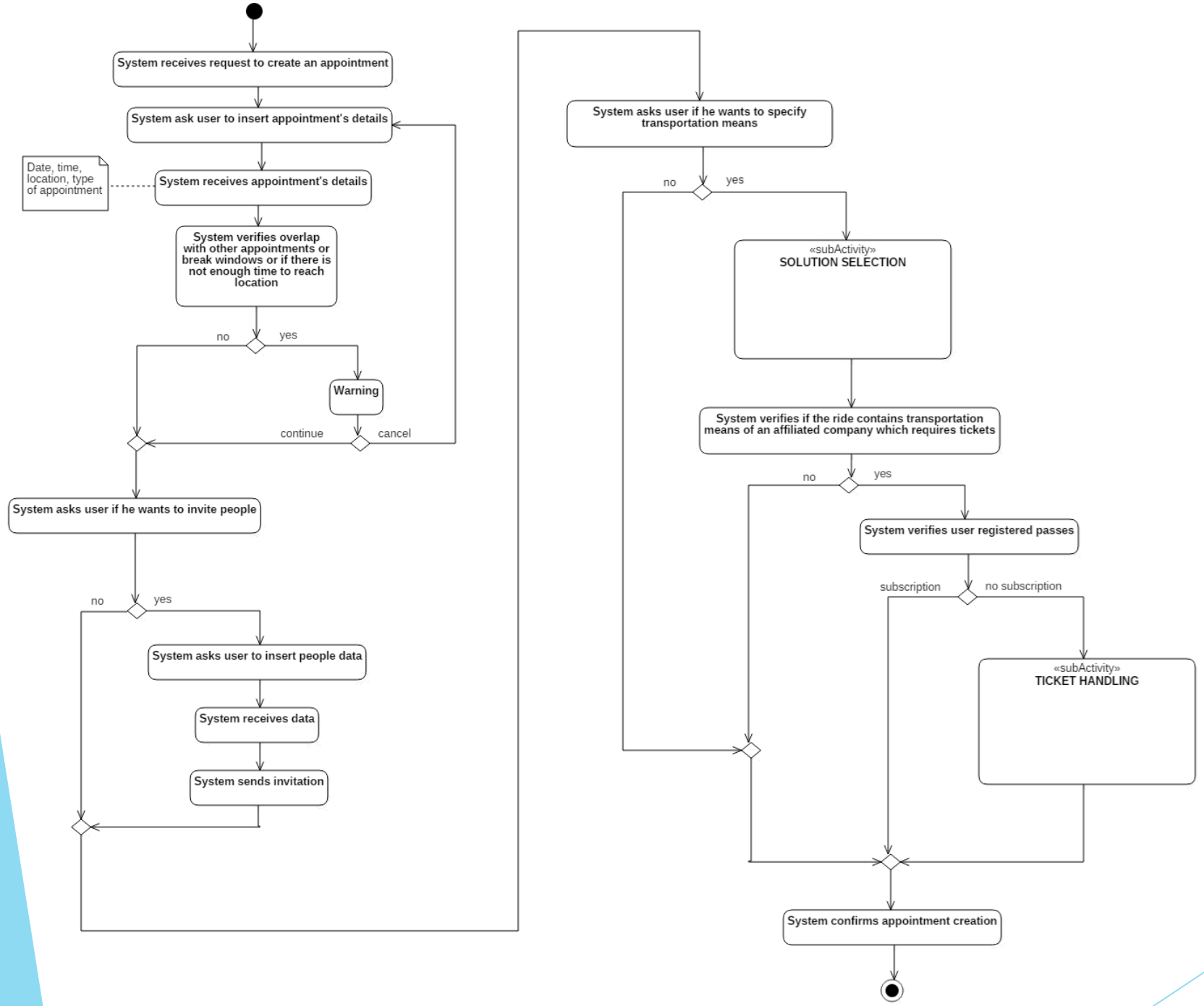
Solution Calculation

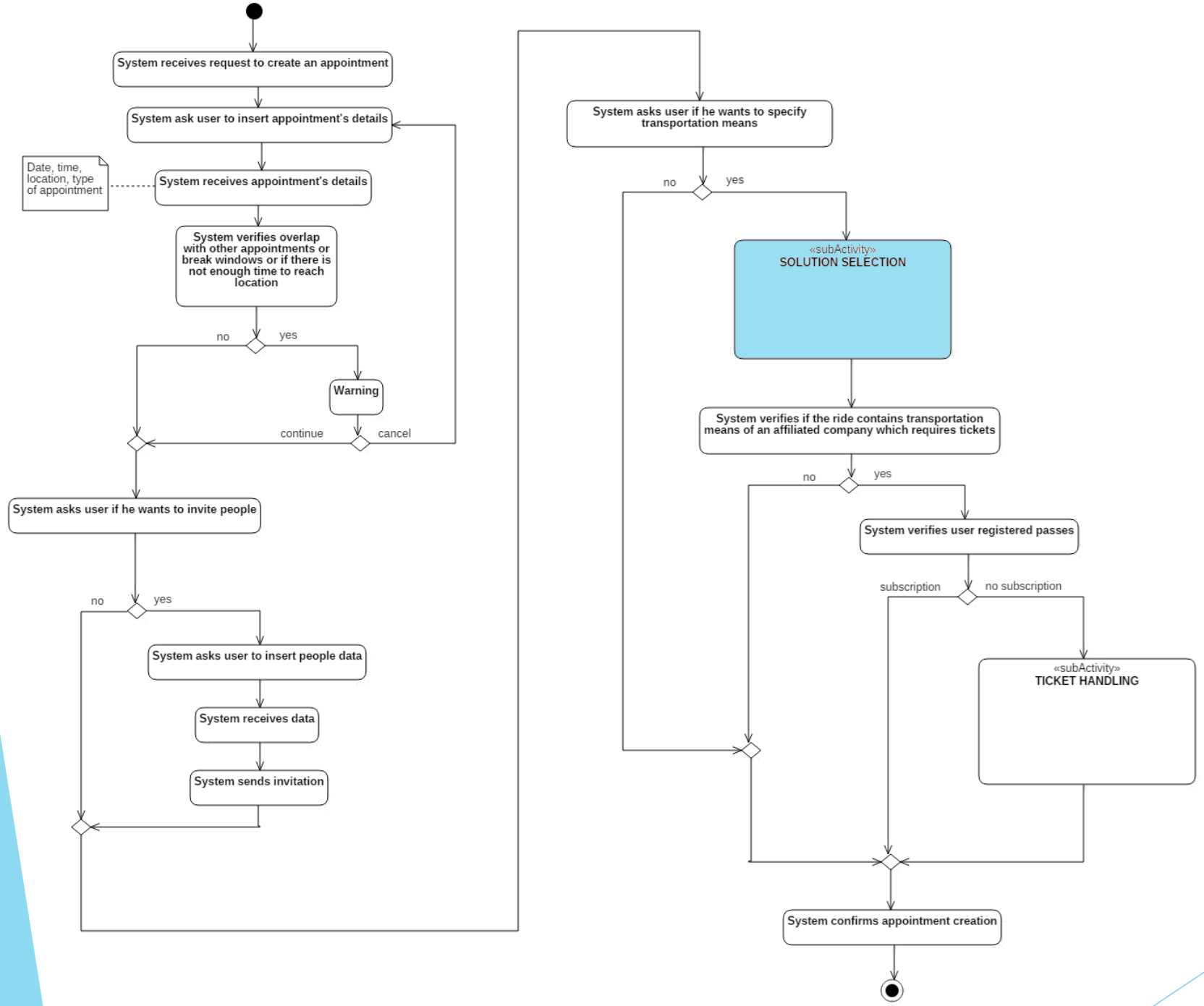
- Interaction with the user

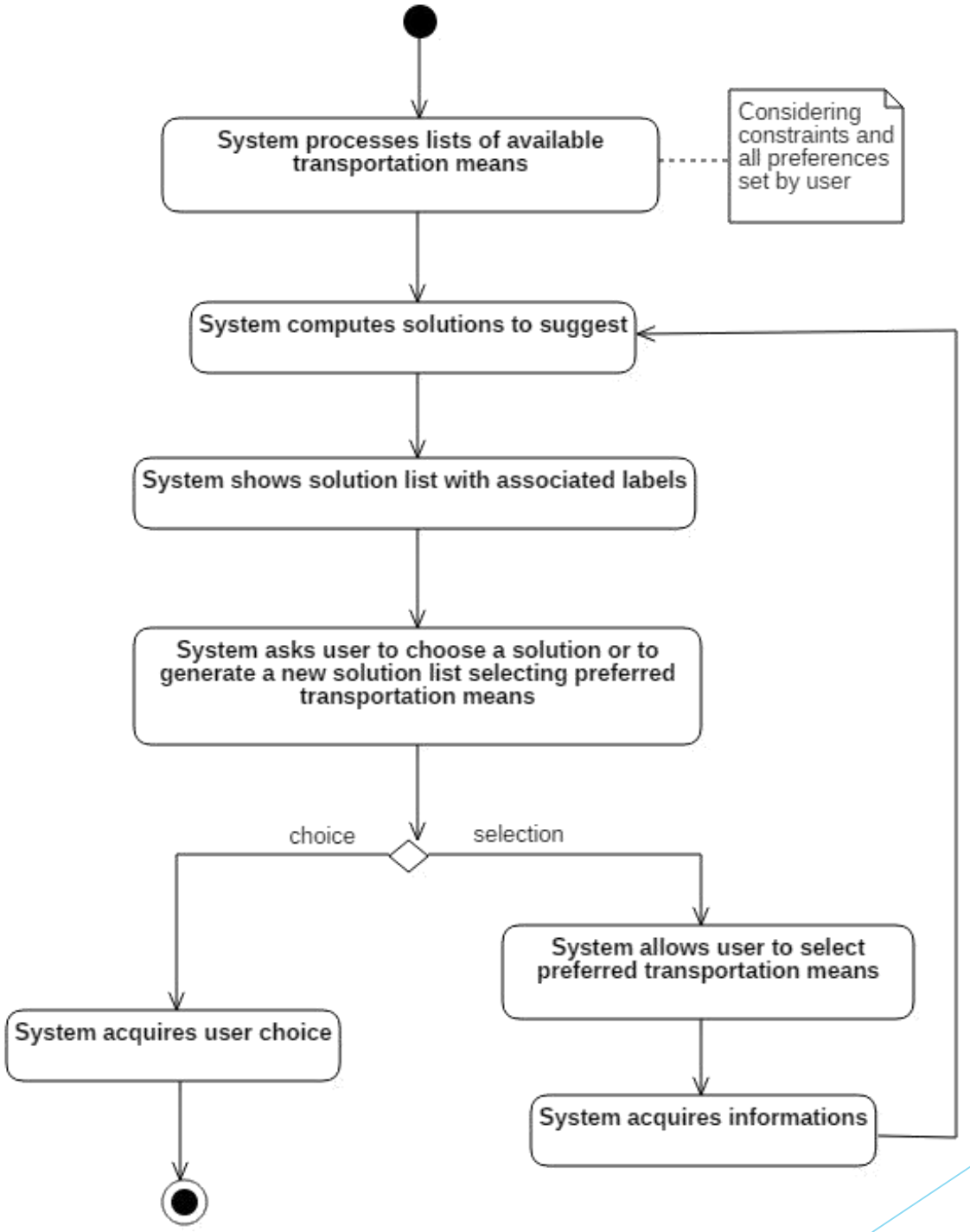
Solution Calculation

- Interaction with the user
- Overview of the algorithm

Interaction with the user


System receives request to create an appointment
System ask user to insert appointment's details
Date, time, location, type of appointment
System receives appointment's details
System verifies overlap with other appointments or break windows or if there is not enough time to reach location
no
yes
Warning
continue
cancel
System asks user if he wants to invite people
no
yes
System asks user to insert people data
System receives data
System sends invitation
System asks user if he wants to specify transportation means
no
yes
«subActivity»
SOLUTION SELECTION
System verifies if the ride contains transportation means of an affiliated company which requires tickets
no
yes
System verifies user registered passes
subscription
no subscription
«subActivity»
TICKET HANDLING
System confirms appointment creation


System receives request to create an appointment
System ask user to insert appointment's details
Date, time, location, type of appointment
System receives appointment's details
System verifies overlap with other appointments or break windows or if there is not enough time to reach location
no
yes
Warning
continue
cancel
System asks user if he wants to invite people
no
yes
System asks user to insert people data
System receives data
System sends invitation
System asks user if he wants to specify transportation means
no
yes
«subActivity»
SOLUTION SELECTION
System verifies if the ride contains transportation means of an affiliated company which requires tickets
no
yes
System verifies user registered passes
subscription
no subscription
«subActivity»
TICKET HANDLING
System confirms appointment creation


System processes lists of available transportation means
Considering constraints and all preferences set by user
System computes solutions to suggest
System shows solution list with associated labels
System asks user to choose a solution or to generate a new solution list selecting preferred transportation means
choice
selection
System acquires user choice
System allows user to select preferred transportation means
System acquires informations

- [G2] a User should be able to customize the offered service
  - […]
  - [G2]#3 define time slot in which the use of specific transportation means should be avoided
  - [G2]#4 define a minimum distance below which a specific transportation mean should be avoided
  - [G2]#5 define a maximum distance beyond which a specific transportation mean should be avoided
  - [G2]#6 disable permanently specific transportation means

- [G2] a User should be able to customize the offered service
  - […]
  - [G2]#3 define time slot in which the use of specific transportation means should be avoided
  - [G2]#4 define a minimum distance below which a specific transportation mean should be avoided
  - [G2]#5 define a maximum distance beyond which a specific transportation mean should be avoided
  - [G2]#6 disable permanently specific transportation means

- [G5] for each appointment, the User should be assisted in the choice of the travel solution
  - [G5]#1 travel solution suggestions must take into account traffic, weather conditions/forecast, strikes, type of appointment, baggage, passengers

Overview of the algorithm

Overview of the algorithm

Overview of the algorithm

- Retrieval of Feasible TravelPlanSolutions
- Filtering
- Labeling

Overview of the algorithm

- Retrieval of Feasible TravelPlanSolutions
  - Retrieve appointment information (through Appointment Manager)
  - interfaces with Google, which provides it with a list of possible travel solutions
- Filtering
- Labeling

Overview of the algorithm

- Retrieval of Feasible TravelPlanSolutions
- Filtering
- Labeling

Overview of the algorithm

- Retrieval of Feasible TravelPlanSolutions
- Filtering
  - Retrieve constraints on Travel Means and Eco Friendly preference
  - Filter out Solutions which contains Ride not compliant with the constraints
- Labeling

Overview of the algorithm

- Retrieval of Feasible TravelPlanSolutions
- Filtering
- Labeling

Overview of the algorithm

- Retrieval of Feasible TravelPlanSolutions
- Filtering
- Labeling
  - Cheapest TravelPlan
  - Fastest TravelPlan
  - Eco-Friendliest TravelPlan
  - Warning: Weather
  - Warning: Traffic

Overview of the algorithm

- Retrieval of Feasible TravelPlanSolutions
- Filtering
- Labeling
  - Cheapest TravelPlan
  - Fastest TravelPlan
  - Eco-Friendliest TravelPlan
  - Warning: Weather
  - Warning: Traffic

```
calculateEcoScore(TravelPlan tp){
        score = 0;
        for each Ride r in tp{
                score = score - r.travelMean.penalty * r.distance;
        }
        tp.Ecoscore = score;
}
```

Overview of the algorithm

- Retrieval of Feasible TravelPlanSolutions
- Filtering
- Labeling

Software Engineering 2 Project, 2017/2018
Politecnico di Milano

Travlendar+

Mirko Salaris 895394
Piervincenzo Ventrella 898604
Pietro Cassarino 899152