**Actors identifying**

* **Guest:** Person that’s not registered yet. He can see the login and the registration page.
* **User:** Person registered and logged to the app. He can access to his calendar to see his appointments, setup new meetings or manage them, change his preferences, …

**Goals**

The aim of the system is to give the user the following functionalities:

* registration to the service and preferences set up;
* anytime management of personal and mobility preferences;
* create and schedule a new event choosing time and location;
* edit previously added event data;
* change travel options on automatically created trips to/from event;
* mobility companies’ tickets purchase via built-in browser;
* provide high customizability of personal preferences regarding user daily routines.

and to provide on its own these other ones:

* manage in a clever way trips to/from user events, relying on the map of the surroundings, time of the day, public and private transports’ timetables and stops, shared means, traffic, weather forecast, possible strikes, event information and user preferences;
* notify the user on time of upcoming trips;
* warn him in case of an exception.

**Domain properties and assumptions**

We suppose that the following conditions are true in the analysed world:

* the geographical area of the city is included in the coverage area of most common mobile communication technologies (3g, 4g) offered by main telecommunications companies,
* users must be subscripted to a sharing service if they want to use it,
* APIs used by the application will always be updated on traffic status, eventual incidents and weather conditions,
* sharing services’ APIs signals their means if and only if the means are where the APIs say, and they are not occupied or booked,
* users always have a working internet connection,
* half an hour is enough warning time for users to start a trip,
* the user cannot ride two means of transport at the same time.

**Functional requirements**

The aim of the system is to give the user the following functionalities, arranged in sections:

* Registration process:
* the application will provide a registration process,
* users can access to the functionalities provided by the system if and only if they register to it,
  + there are not users with privileges, every registered user can access to the same features of the other ones and the system is safe even without supervisors,
* all data will be stored on the Travlendar+ server to backup user information and to allow it to be shared on different user’s devices.
* Initial settings, always adjustable:
  + set user personal information:
    1. google account to synchronize calendar and maps,
    2. house location, work location, new location,
    3. default location to reach after appointments, to be chosen between the favourites (usually home).
    4. user break times and guaranteed amount of time to keep free from trips in every break to have lunch or another dynamic event,
    5. time of the day after which bike (owned or shared) and public transports will not be considered anymore in planning the trip.
  + set desired user transport means:
    1. car possession, bike possession,
    2. car sharing account(s), bike sharing account(s),
    3. public and private transports (possibility to insert season ticket),
    4. maximum walking distance (to destination or sharing vehicle).
* Scheduled event creation:

1. set day(s),
2. set time of beginning and end,
3. set location.

* Calendar consultation and edit:
  1. see future scheduled events and meetings,
  2. modify events previously added.
* Planned trips consultation and edit:
  1. desire to minimize carbon footprint,
  2. choose between any other trip possibilities in case the one recommended by Travlendar+ is not suitable,
  3. purchase tickets for public transports before the trip,
  4. presence of passengers,
  5. show the best travel option selected by the app on various factors,
  6. consult the map showing the best route to reach the destination, the sharing vehicle or the public transport stop.
* On arrival of scheduled meeting:
  1. notification service informing the user he needs to leave to the next meeting in half an hour,
  2. notification service telling the user the time to leave to the next meeting has come,
  3. alert the user of trips’ changes between the first reminder and the departure time.
* In case of exceptions:
  1. warn the user of overlapping events creation and ask to choose primary event,
  2. warn the user of possible inability to arrange trip that takes him in time to an unrechable future event.
* On trip planning:
  1. the Google Maps API is used to determine best route (also considering traffic),
  2. time of the day is considered on means of transport’ choice,
  3. public and private transports’ timetables, stops and position (obtained through their APIs) are employed to evaluate their possible usage,
  4. shared means position (obtained through their APIs) are employed to evaluate their possible usage,
  5. weather forecasts are used to evaluate the best time to schedule lunch (or other dynamic events) between trips and the possibility to go by bike,
  6. possible strikes are considered on means’ choice,
  7. event information is used to set trip’s destination, time of arrival or departure and location,
  8. user preferences are considered to choose means and dynamic events allocation,
  9. season pass information will be used to check the necessity of a ticket for the trip,
  10. the best trip is processed for every means of transport not excluded,
  11. the suggested trip is chosen according to the user trip preference (fastest, eco-friendly, customized).

**Non-functional requirements**

The system will also undertake the following constraints:

* trips solutions are updated daily until the day of the trip, then every hour until half an hour before the trip, finally every minute until the trip begins,