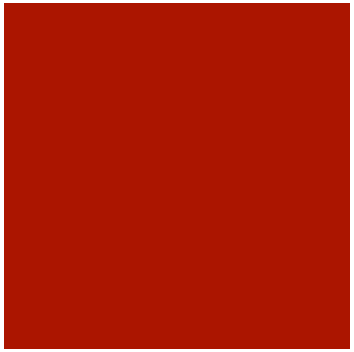
Practical 2: Use Cases

Where are we now?

- We have managed to do all of the following:
  - Produce a list of stakeholders
  - Identify some initial top level goals that stakeholders have for the system
  - Practiced boiler plating those goals into a formal requirement

BikeTour: Functionality



- Taking all of this information we are going to make up use cases for use of the BikeTour application.
- Start with the obvious Primary Actor: Cyclists
  - By this we mean any cyclist, we will not distinguish types of cyclists in this practical

Goals to build use cases for:

- User wants to find information about daily traffic on particular routes.
- User wants to find the distance travelled on a particular route from one point to another.
- User wants to find out how far a specific landmark is from a specific location.
- User wants to find all landmarks of interest from a particular within a particular radius of their route of travel.

Practical Activity 1: Define front matter (20 min)

- Assume the the primary actor is always the Cyclist.
- For each of the above (broad) goals, define each of the following:
  - Supporting actors
  - Pre-conditions
  - Minimal Post-condition(s)
  - Trigger

Practical Activity 2: Define scenarios (40 min)

- For each goal propose a sequence of actions that a Cyclist would need to undertake to complete their goal
  - Do not get in a loop arguing about whether the steps are absolutely correct – if the group can agree that the steps are plausible through consensus, that is sufficient for this activity.

Practical Activity 3: Exceptions (15 min)

- Identify steps in the scenario where errors or exceptions can occur.
- Write at least one exception scenario for each of the use cases.

Practical Step 4: Alternatives

- Consider Cyclists in two different categories – Commuters and Fitness
  - Are there any differences in the primary scenario?
  - Create alternative scenarios with different post conditions for these alternatives

Practical Step 5: Exceptions

- Look for places in your scenarios where there are errors or exceptions that could happen.
- Change your Use Case so that it addresses some of these exceptions – in particular create Exceptional scenarios with different post-conditions