**Map Program Execution (UML1)**

Preconditions:

* None

Main Success Scenario:

1. xxx

Alternatives:

None

**Map Object Changes (UML2)**

Preconditions:

* None

Main Success Scenario:

1. xxx

Alternatives:

None

**Types of Diagrams**

Structure

* Class:
  + Describes the classes and objects that will make up a system. It is a static view of a system. Shows collaboration of elements in the static view. Describes the functionalities performed by the system. Helps in the construction of software applications using object oriented languages.
  + [Reference 1](http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/)
* Object:
  + Derived from class diagrams. So, object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.
* Component: xxx
* Composite Structure: xxx
* Package: xxx
* Deployment: Can describe the physical architecture and the deployment of components on that hardware architecture.

Behavior Diagrams

* Use Case:
  + Describes the boundary and interaction between the system and users. Corresponds in some respects to a requirements model. Used for requirements analysis and high level design, model the context of the a system, reverse engineering, forward engineering.
* Activity
  + Like a flow chart. Describes the workflows the system will implement. This diagram is made to understand the flow of activities and mainly used by the business users. So, this is a high level diagram. How this is different from a flow chart is that it has additional capabilities like branching, parallel flow, swimlane, and maybe a few more. The main usages of this diagram are modeling work flow by using activities, modeling business requirements, high level understanding of the systems functionalities, and investigate business requirements at a later stage.
  + [Reference 1](https://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep03/f_umlbasics_db.pdf)
* State: Describes the states or conditions that classes assume over time.
  + [Reference 1](http://www.sparxsystems.com/resources/uml2_tutorial/uml2_statediagram.html)

Interaction Diagrams (derived from more general Behavior Diagrams)

* Sequence: xxx
  + [Reference 1](http://www.ibm.com/developerworks/rational/library/3101.html)
* Communication: Describes how objects in the system will interact with each other to get work done.
* Timing: xxx
* Interaction Overview: xxx

Good UML Reference

* [Tutorials Point](http://www.tutorialspoint.com/uml/uml_class_diagram.htm)