Now, I would like to continue with an important project stage – analysis. As we were assured that proper analysis can have a huge impact on next stages like Design and Implementation, we spent a lot of time in this stage.

At the beginning, we stated 3 main problems from the interview.

* First one was the problem that how we will store data about members, lecturers and events.
* Second one was the need of VIPASANA to be able to search for a proper lecturer for instance by given category
* Last one was how to make the system easy to extend.

In order to find proper solutions was necessary to think ahead and imagine how our system could work. We summed up delimitations as follows:

* This system won’t use databases for data storage
* Initial data will be provided from files, not by real user
* The issue about storing feedback we found not very efficient as this system is supposed to be only single-user system. Therefore, the administrator would must collect and upload manually all feedback data to the system.
* Due to lack of time and knowledge, this system is not able to send reminding emails to members who haven’t paid.
* What is more, searching for an event in specific time period isn’t implemented in the system.

The conclusion from interview, problem statement and delimitations gave us the list of requirements. I would divide functional requirements into 4 groups.

* Regarding members and participants the administrator must be able to store info about new member and also for example to sign up member/participant to an event.
* As for a lecturer, admin must be able to store data about new lecturer and also search for lecturers of given category

Let me take a detailed look into the base sequence of this requirement. First of all the system initializes new empty list of suitable lecturers. Then it goes through the list of all lecturers and check for every lecturer in the list whether his list of categories contains category that is equal to desired category. If yes it adds this lecturer to the list of suitable lecturers and in the end the system just return the list of suitable lecturers.

* Regarding Event there are important requirements about creating and modifying event or searching for not finalized events to finalize them.
* As for generating newsletter the system must collect the list of lecturers who wants a fee or advertisement and the list of finalized events.

Our non-functional requirements are listed as follows:

* The system must be implemented in Java
* Secondary storage is done with files
* The parts of the system must be accessible among them

This is how looks USE CASE DIAGRAM created from requirements. In this diagram is described what is administrator able to do.

We found interesting and important to describe more one of our activity diagrams – Plan an event. The steps to plan an event are as follows:

* In the beginning , choose the type of the event – Lecture, Seminar, Workshop or Trip
* Then if user wants, he can add a topic
* In the next step the system checks whether the event is of type Trip
  + If yes, user decides if he wants to add a location
  + If not, user has the possibility to add a lecturer (using USE CASE: Search for lecturer)
* Then either way if user wants, he can add a start and end date and the price as well
* At the end the system checks whether the event contains all necessary data, it means whether can be finalized
  + If yes, it enables checkbox FinalizeEvent and user decides if he wants to finalize it by selecting this checkbox
* Either way the system saves the event.

The next part of presentation belongs to Daniela.