Chabot supported with knowledge base

Intro:

V.I.K.I is the name of our Chabot ,this Chabot is used in medical assistance for poor people around the world who can't pay money to visit a doctor for medical diagnosis or medical description , the Chabot is built by rasa framework and python using database neo4j , VIKI speaks English and formulate its own experience form text file speaks about diseases and some dataset to know how to find the answer and generate a friendly answer this use the framework, it's recommended to use image classification algorithms to detect disease from images or injury ,VIKI must be fast and use logic based on knowledge base to classify the symptoms to its correct diseases , VIKI shell be human like conversation engine

1- What shell V.I.K.I do(functional requirement)?

1-do medical diagnosis or medical description according the knowledge it has

2-answer questions about anything (can be specified in medical only)

3- Ask questions and manage dialog with user

4- It has only patients that use it as free services

5- Use web application like messenger

6- it shell send/receive images and photo from the user

Obligatory

2- Analysis

Use case
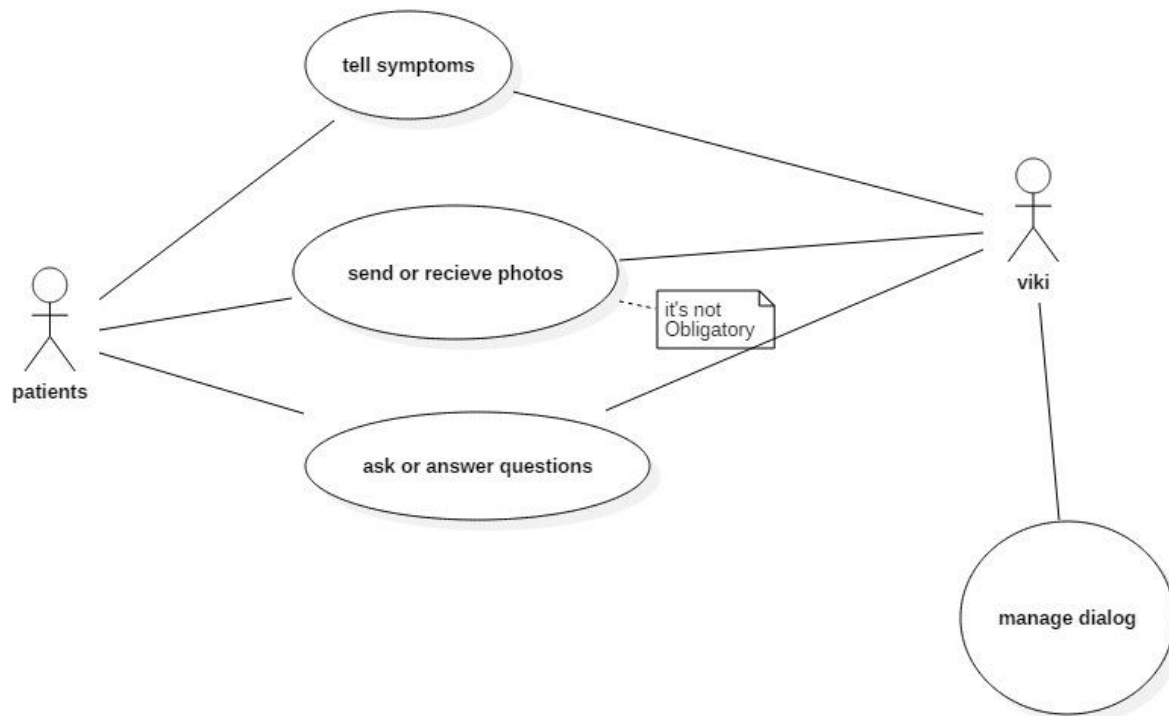
From above we get three 4 actions happens in the system we are developing, the user can tell symptoms or VIKI can give him a symptoms and the user can say yes

It's not obligatory to send image from VIKI or to it

They both can ask question and answer it

Figure 2-1: use case design
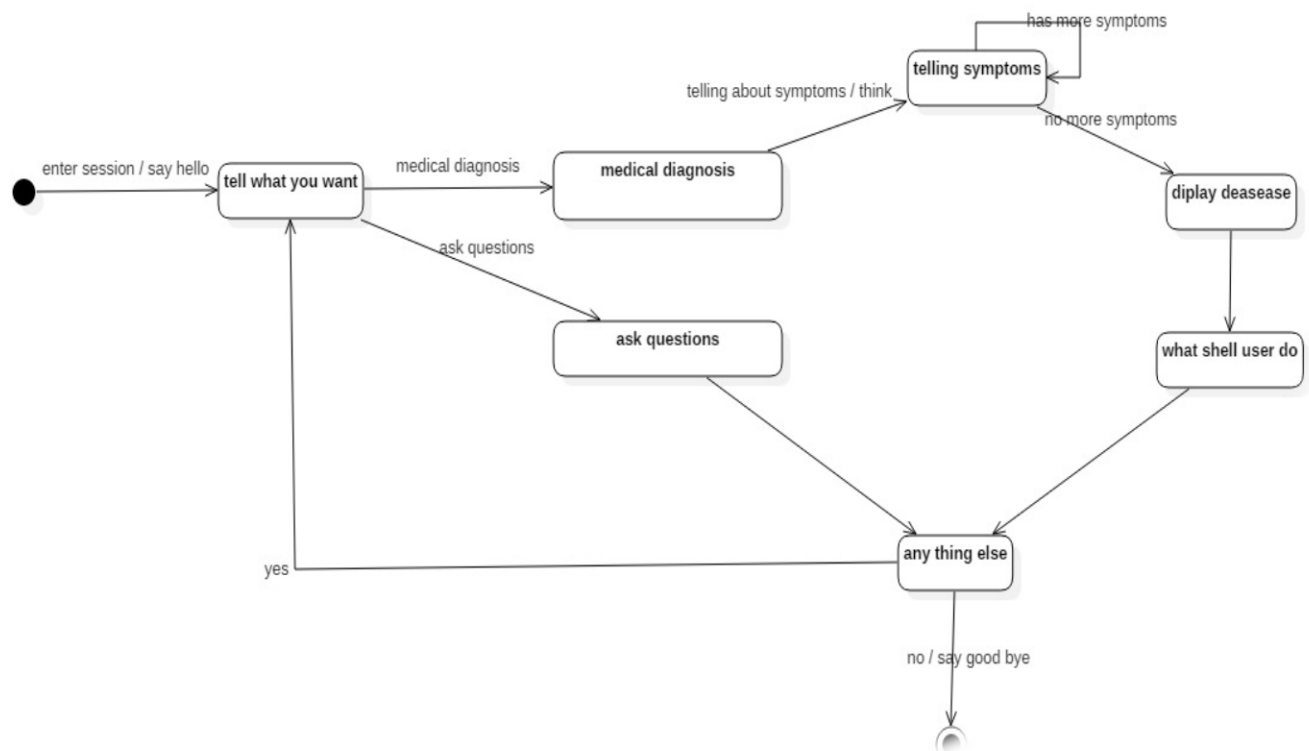
Finite state machine

Figure 2-2: this is finite state machine is for dialog without image sending/receiving

This finite state machine describe the general conversation how should behave from welcome until saying good bye

Finite state machine for filter symptoms

Filters here is for filtering the most optimal and satisfying theorems to given parameters (symptoms)

state diagram proving theorms

start search

recieve symptoms

inilize searchobject
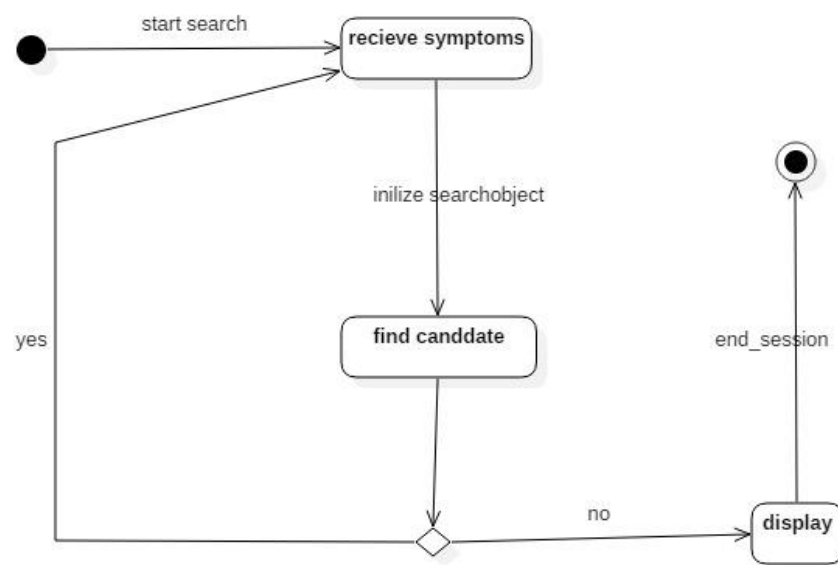
find canddate

yes

end_session

no

display

Figure 2-3: finite state machine for theorem(disease proving)

This finite state machine describe how the search should be done in implementation level

how the system should behave?

1- Question answering :this scenario is easy and like any question-answer conversation we have with any one

ask question

user	rasa

- welcome user
- user enter session
- user ask question
- filter database
- rank results
- get most ranked diplay it
- read
- user have question
- yes
- no