Home Appliance Control System

Project Phase II

Final Project Report

Team:
Tech-9

Website:
http://ooad.ersoft.org

Team Members:
Abhishek Anand
Abhinav Devpura
Chaitanya Halbe
Eric Anderson
Jamie Smith
LeQiao
Madhura Bhatawdekar
Sandip Gaikwad
Vineet Singh

Table of Contents

1. INTRODUCTION:

Home appliance control systems (HACS) are not a new concept. They have existed in science fiction but also in implementation. X10 was the first such system and was created in 1975. Although it and other HACS have been available for over 30 years, they have not seen widespread adoption and remain only as a subject of science fiction to the general community. It is believed that the lack of adoption is due to the lack of extensibility, cost, and ease of existing systems.

The Home Appliance Control System (HACS) is developed to control various home appliances such as a Microwave, Oven, Air Conditioner, TV, etc., through one or more controllers such as a cell phone or PDA. In centralized controlled systems, home appliances will be connected to base station, which is called the Home Appliance Controller (HAC), installed in the house. Through HAC we can control all the home appliances by issuing commands from the mobile device. If the control is distributed home appliances will not be connected to base station, rather any appliance will be able to take control over other appliances. HACS will give us easy control over the home appliances even when we are away.

1.1 Scope

Tech-9 hopes to develop a HACS that corrects the listed insufficiencies of previous systems. This document's purpose is to show the development of such a system using some ideas presented by the WRSPM Model. This document's sections are modeled after the World, Requirements, and Specification of the model. Due to Tech-9's limited resources, this document does not provide a complete view of each, but instead just aims to provide just enough understanding of each to allow some traceability to the architecture and design of the system.

2 BRIEF DESCRIPTION OF PROJECT PHASE I

Initial understanding of the home appliance control system

- Use Case Diagram
- Use Case Template
- Sequence Diagram

System Design

- Class Diagram
- CRC Card

3 BRIEF DESCRIPTION OF PROJECT PHASE II

The purpose of the phase is to adapt the System development phase with changing requirement & refine the understanding of System in aid with different kinds of UML2.0 diagrams as possible, including class, use case, sequence, component, state transition, and activity and package diagrams, in carrying out an analysis, design and implementation of a home appliance control system (HACS).

It also aim in detecting & eliminating defects by using Internet search engines and appropriate literature, as well as creative imagination and interaction with team member & advisors

3.1 Changes/Additions to the initial description of the HACS

Since the elicitation of the initial description of the HACS, the potential user has expressed the needs for a couple of items:

1) In case of the failure of any of the home appliances controlled by the HACS, an individual or an organization capable of handling the failure shall be notified with appropriate information. For example, in case of fire, a fire department shall be notified immediately with the location, and possible cause, of the fire; for a break entry, a police department; etc. Of course, it goes without saying that the user also shall be notified of any such critial failure through a remote device s/he uses.

2) The HACS shall have a database of information about all the appliances it controls, including information about the date each home appliance was connected to the HACS, all the users, the vital actions the user(s) of the system has issued, etc. The system shall maintain not all the historical information but only for the period of one month, inclusive of “today”.

Initial understanding of the home appliance control system

- Use Case Diagram – Refined using UML2.0 notation.
- Sequence Diagram – Refined using UML2.0 notation.
- Communication Diagram

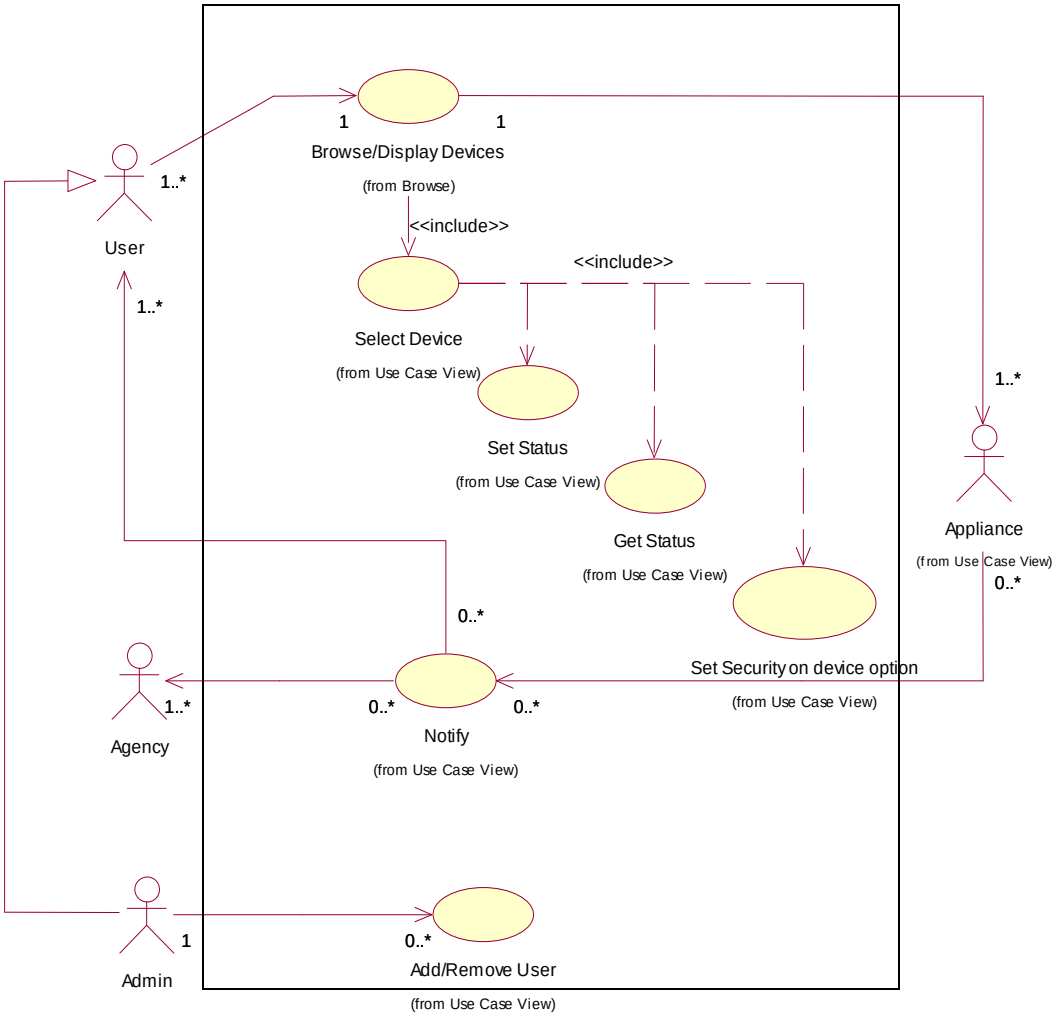
System Design

- Class Diagram – Refined using UML2.0 notation.
- State Chart Diagram
- Activity Diagram

4 USE CASE DIAGRAM – REFINED USING UML 2.0 NOTATION:

Overview of the Use Case diagram:

- We have recognized four actors of our system: a general user, an administrator who is a special kind of user, agency and a general actor which depicts all the devices in the house.
- The user can access the browse/display devices use case to have a look at all the devices that are connected in the network of HACS.
- All the appliances sit in the network, which is shown by the dotted boundary in the use case diagram.
- The display device option contains/includes the select device use case. It helps the user to select a device from the list of device the user gets by browsing all the devices.
- After selecting a particular device, the user can get the status of the device, that is, its current working state. He can change the state/ properties of the device using the set status use case.
- The user can also set security on device options, using the set security use case. For example, the parents don't want the children to view some channels. So they can lock that channel (which is a security feature) by entering a 4-digit pin number for that channel.
- All the use cases set status, get status and set security on device options come under one use case which is select device. This is because these use cases can be used only after user selects a device from the network.
- We have a use case notify, which only specific appliances use, if there is some change in their state or if there some failure or emergency. Suppose, there is an accidental fire in the house, so the fir alarm should send a signal to the user warning him that there is an emergency and also this message will be sent to the agency. Then the agency can take appropriate action, in this case it will be alerting the fire department about the fire.
- The administrator is a privileged user, as mentioned earlier, and he has access to all the use cases that a normal user can access and besides this, he can add or remove a user from the system.



Refinement to the Use Case Diagram:

- Multiplicity is added to the use case diagram according to the UML 2.0 specifications.
- One or more users will use the Browse or Display Devices use case provided by the HACS. This use case includes the 'select device',' set status',' get status' use cases.
- The 'browse/display device' use case will access one or more appliances on the network. One or more appliances because the system will have any number of devices from 1 to many.
- The appliances will use 'notify' zero or more times because an appliance will not notify the user and agency unless a request for sending notification is set explicitly or unless there is some device failure or emergency like fire. Hence, zero or more appliances will be accessing the notify use case and the same notification(s) will be sent to the user as well as agency.

4.1 Use Case Template – Browse/Display Device

Use Case Name:	Browse or Display Device
Actors:	User, Appliances
Description:	Gives list of devices connected to system
Preconditions:	1. All devices should be connected and working.
Post conditions:	
Normal Flow:	1. Log In to system. 2. User Request the list of Devices 3. Controller device sends the discover devices request to zeroconf. 4. Zeroconf returns the list of devices. 5. List of device will be displayed to user through UI.
Alternative Flows:	
Includes:	Log In.
Notes and Issues:	

4.2 Use Case Template – Select Device

Use Case Name:	Select Device
Actors:	User
Description:	One of devices connected to system is selected.
Preconditions:	1. All devices should be connected and working.
Post conditions:	
Normal Flow:	1. Discover devices connected to system 2. Give list of devices 3. Select device.
Alternative Flows:	
Includes:	Browse or display devices.
Notes and Issues:	

4.3 Use Case Template – Set Status

Use Case Name:	Set status
Actors:	User, Appliances
Description:	One of parameters of selected device is set.
Preconditions:	1. Selected device should be working.
Post conditions:	
Normal Flow:	1. Select device 2. Choose parameter to be set 3. Set value of parameter.
Alternative Flows:	
Includes:	Select Device.
Notes and Issues:	

4.4 Use Case Template – Get Status

Use Case Name:	Get status
Actors:	User, Appliances
Description:	Get value of parameters of selected device.
Preconditions:	1. Selected device should be working.
Post conditions:	
Normal Flow:	1. Select device 2. Choose parameter to be whose value is needed.
Alternative Flows:	
Includes:	Select Device.
Notes and Issues:	

4.5 Use Case Template – Set Security on Device Options

Use Case Name:	Set security on Device options.
Actors:	User
Description:	Security options are set for selected device.
Preconditions:	1. Selected device should be working 2. User must have entered correct PIN number.
Post conditions:	
Normal Flow:	1. Select device 2. Enter Pin number. 3. Set security options for device.
Alternative Flows:	
Includes:	Select Device.
Notes and Issues:	

4.6 Use Case Template – Notify

Use Case Name:	Notify
Actors:	User, Appliances, Agency
Description:	Appliance sends notification to User and to agency in case of any emergency.
Preconditions:	User has enabled notification option for device.
Post conditions:	
Normal Flow:	1. Appliance 2. Choose parameter to be set 3. Set value of parameter.
Alternative Flows:	
Includes:	
Notes and Issues:	

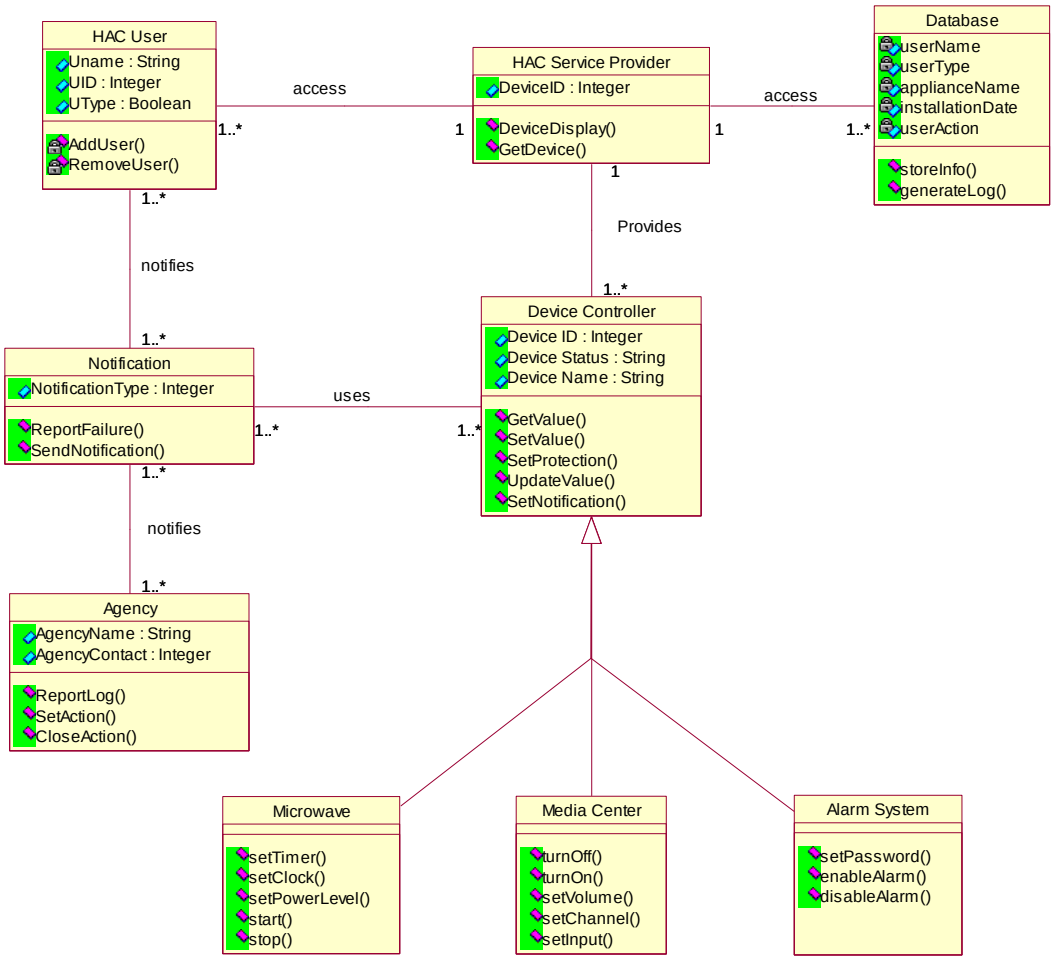
4.7 Use Case Template – Add Remove User

Use Case Name:	Add or Remove user.
Actors:	User, Admin
Description:	Gives access permission to new user.
Preconditions:	
Post conditions:	
Normal Flow:	1. Send request to Admin 2. He will set access permissions for user.
Alternative Flows:	
Includes:	
Notes and Issues:	

5 Class Diagram

HAC User class tells description about type of user and allows only valid user to enter system in accordance with UID. Now HAC user class can access HAC Service Provider. Now when user will select some kind of device then HAC Service Provider provides Device Controller for that particular device, which will contain attributes and operations of that particular device. Now that controller can use notification class to send notification to user, and also User can set up notification through that Notification class. Now there are different kinds of notification. If it is a normal notification then it is send only to user, and if its some emergency notification, then that notification is send to user as well as agency. Now Agency class has different operations like SetAction and CloseAction, also there will be log of all notifications that come to agency class. Note that Device Controller will manage the properties for each device.

CLASS DIAGRAM:

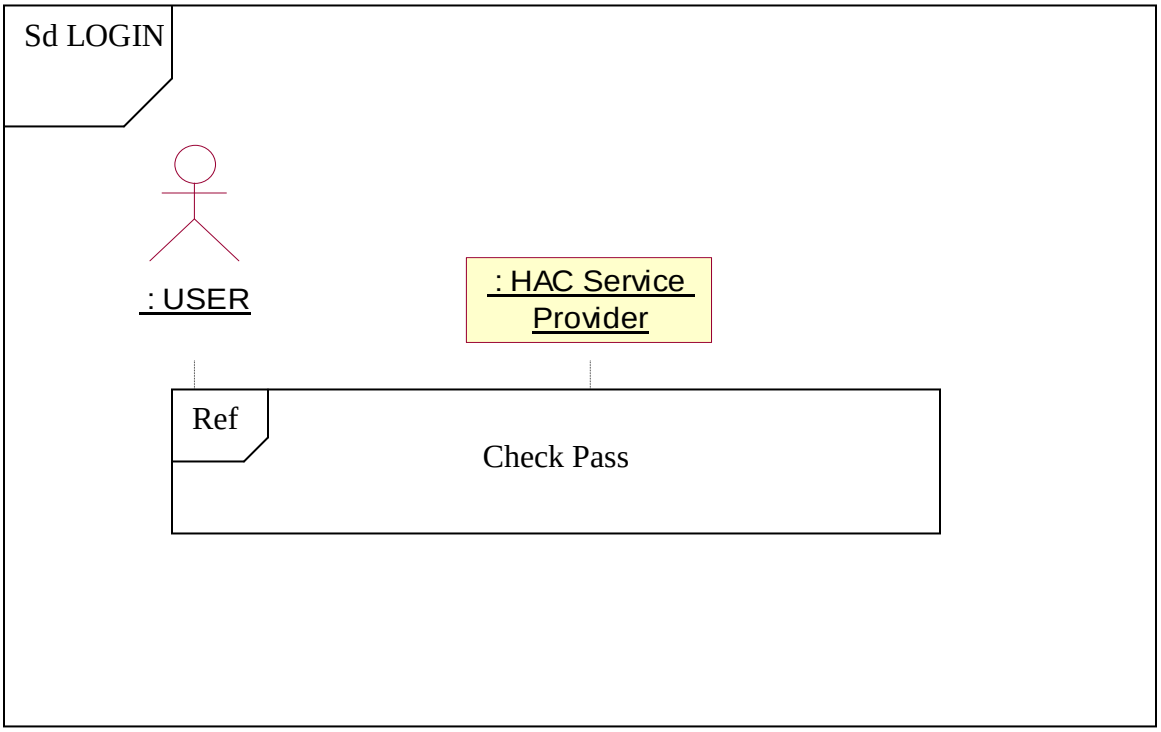


Refinement:

Microwave, MediaCenter & Alarm System is added as specialized classes of Device Controller class in order to reflect better visibility of System Model. Rest of the class diagram is same as the class diagram in the Project Phase I report.

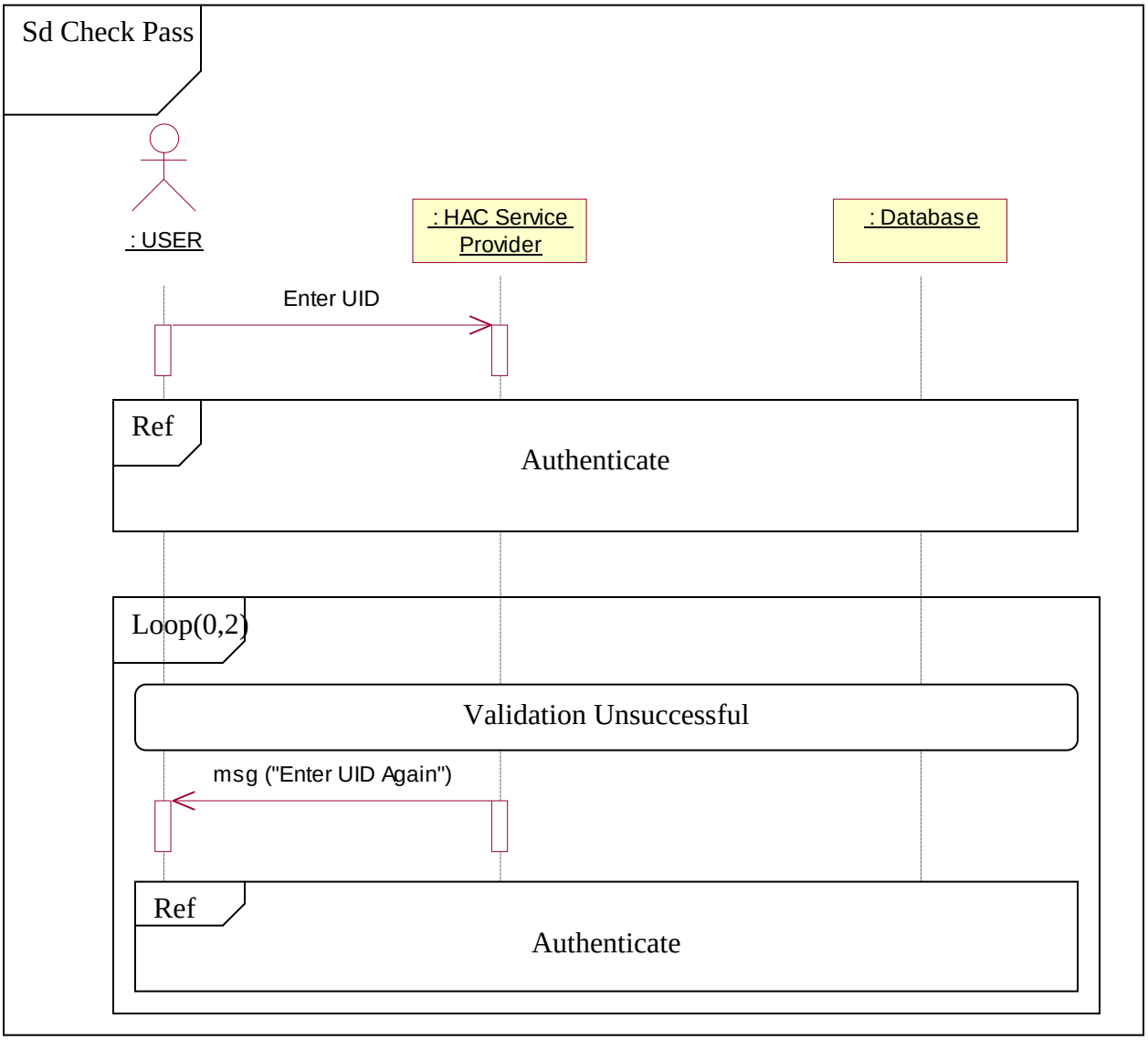
6 SEQUENCE DIAGRAM

6.1 Sequence Diagram for User Login



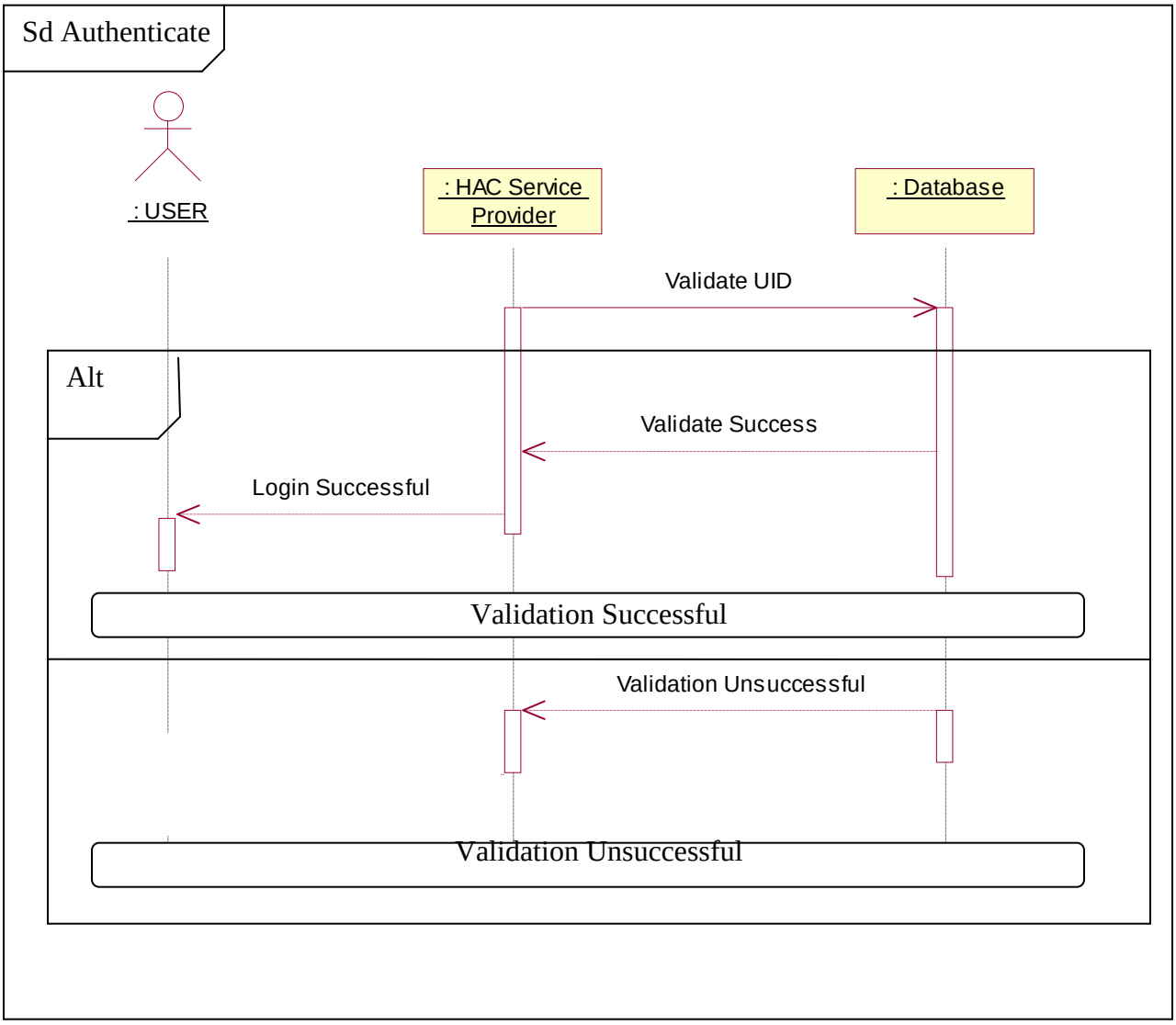
- We have used frames here to depict the user login.
- The box in the left-top corner gives the name of the sequence diagram.
- The frame ‘ref’ here is used to refer to the frame ‘Check pass’ which is elaborated in the next sequence diagram.
- When the user tries to log in to the HACS, the HAC Service provider will authenticate the user.
- The details of authentication and validation are given in the following sequence diagrams.

6.2 Sequence Diagram for Checking Passes



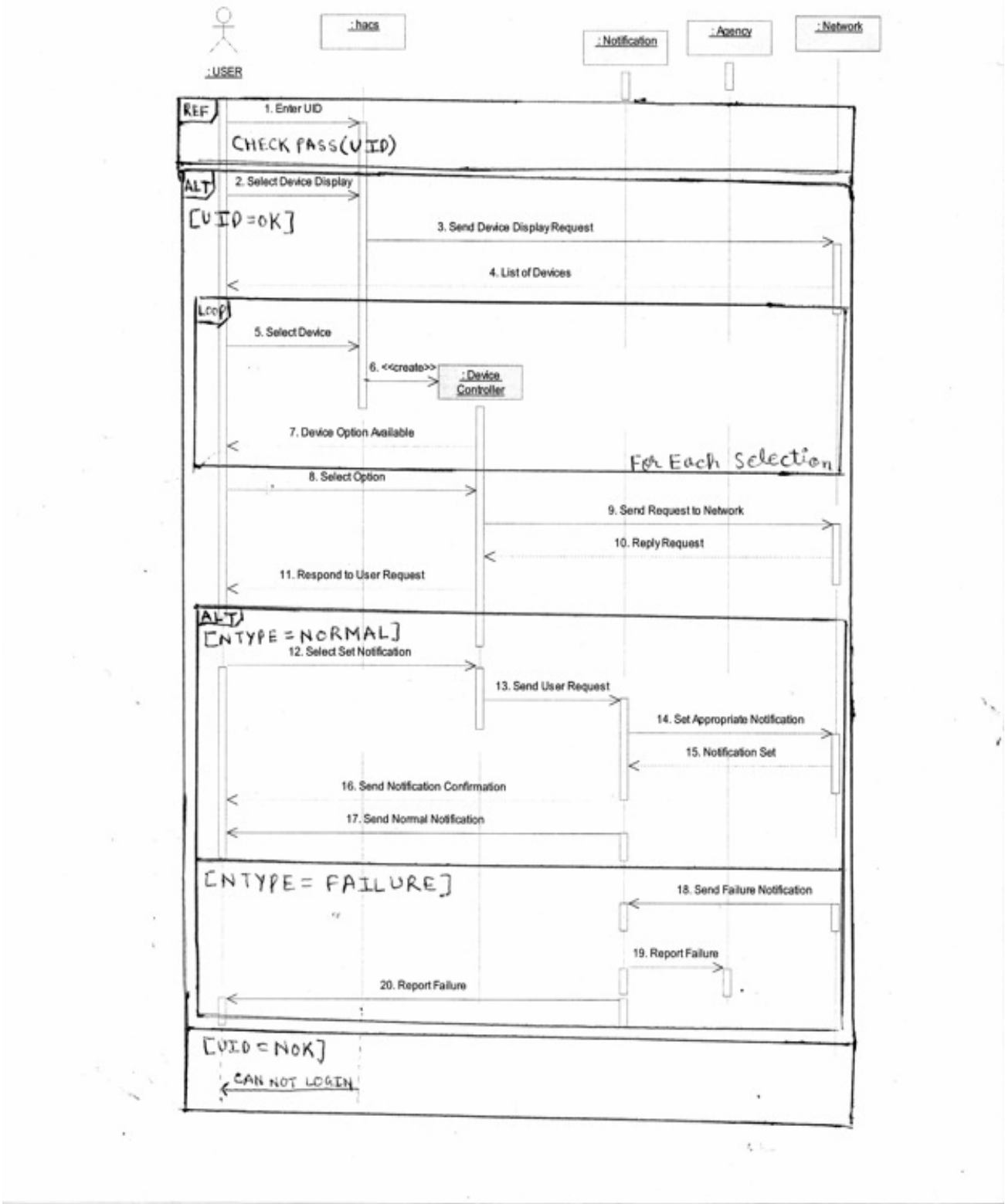
- When the user enters his/her UserId, the HAC Service Provider will authenticate the user. The authentication procedure has been given in the following sequence diagram.
- The 'Loop (0, 2)' frame depicts that the user will be given total 3 passes to enter the correct combination of his UserId and Password.
- If the UserId and password combination is correct, which is validated against the same stored in the database, user is given access to the HACS after authentication.
- If the validation is unsuccessful, user is given 2 more chances to enter his correct UserId and password.

6.3 Sequence Diagram for Authentication



- The entire authentication scenario is shown in the above ‘Authenticate’ frame.
- After the UserId is validated, if both UserId and password combination is correct the validation is successful and the user is given access to the system.
- If the validation is not successful, user is given 2 more chances to enter correct Id and password.

6.4 Sequence Diagram for Whole *System Operation*



- Above sequence diagram gives the entire system operation scenario. This is just an instance of how the system will work in one scenario.
- We have considered that the user will login to the system, choose a device, selects operation to be performed on that device, the device fails and then notification will be sent to the user.
- First, during the user login, the 'Check Pass' reference frame will be referred to.
- After successful login, the user will be displayed a list of all devices on the HACS network. The user selects a device after which a request is sent to Device Controller, which then gets all the options for that particular device, from which the user can select what operation he wants to perform.
- The user selects one of the options and the request is sent to the device through the Device controller and the device responds accordingly to the user.
- The user can set notification on devices, for instance, the user can set notification on the Microwave that 'Remind me after the food is cooked' and the user gets this normal type of notification wherever he is.
- There is another type of notification, which is sent by the devices themselves. This notification is sent to the user on device failure.
- A scenario which is not shown in the above diagram is that, if there is fire in the house then the fire alarm system will send 'Fire' notification to the user.

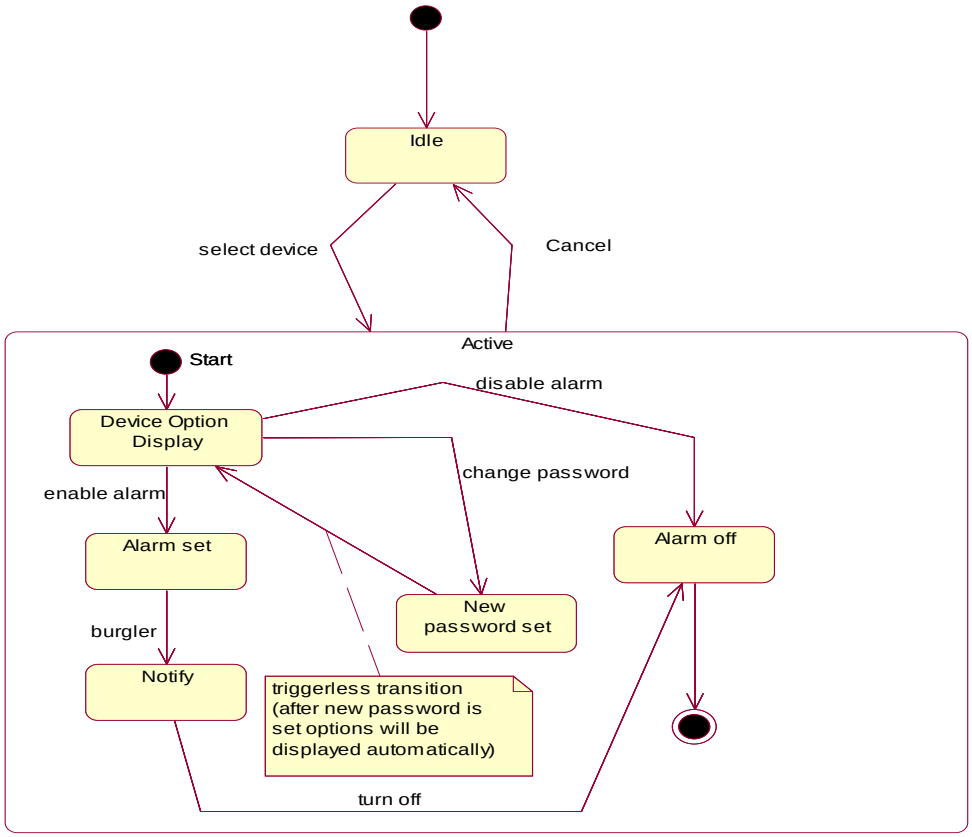
7 STATE CHART DIAGRAM

State chart diagrams, also referred to as State diagrams, are used to document the various modes ("state") that a class can go through, and the events that cause a state transition. For example, your television can be in the Off state, and when the power button is pressed, the television goes into the On state. Pressing the power button yet again causes a state transition from the On state to the Off state. In comparison the other behavioral diagrams which model the interaction between multiple classes, State diagrams typically model the transitions within a single class.

7.1 STATE CHART DIAGRAM FOR ALARM SYSTEM:

Flow of State Transitions:

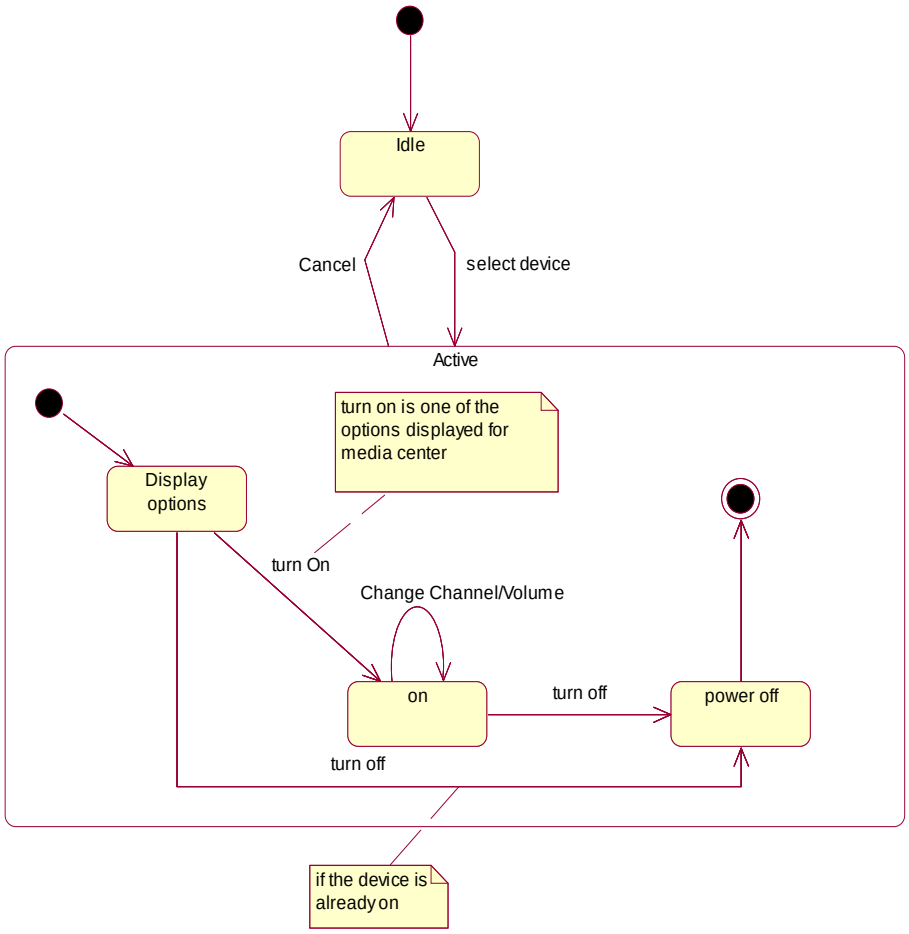
- Initially the system is in idle state.
- On event ‘device selected’ it moves to the ‘displaying option’ state by displaying the options.
- On event “enable alarm” alarm system get activated.
- Upon event “burglar” it trigger notification & then turns off when “ turn off” event is triggered to the alarm system



7.2 STATE CHART DIAGRAM FOR MEDIA CENTER:

Flow of State Transitions:

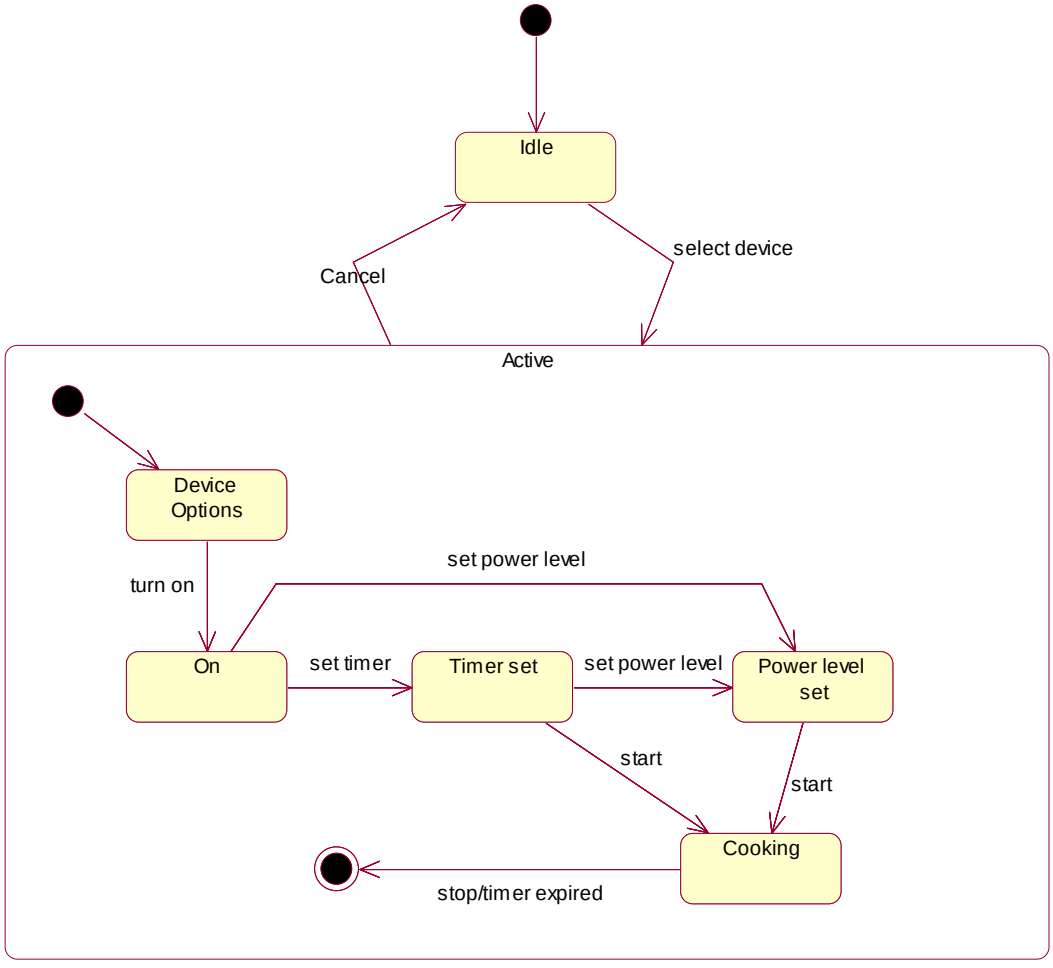
- Initially the system is in idle state.
- On event ‘device selected’ it moves to the ‘displaying option’ state by displaying the options.
- On event “turn On” Media Center makes the transition into On state
- Event “Change Channel” & “Volume” will change the volume & channel for media center in on state.
- Event “Turn Off” will switch the media center to Off State.



7.3 STATE CHART DIAGRAM FOR MICROWAVE:

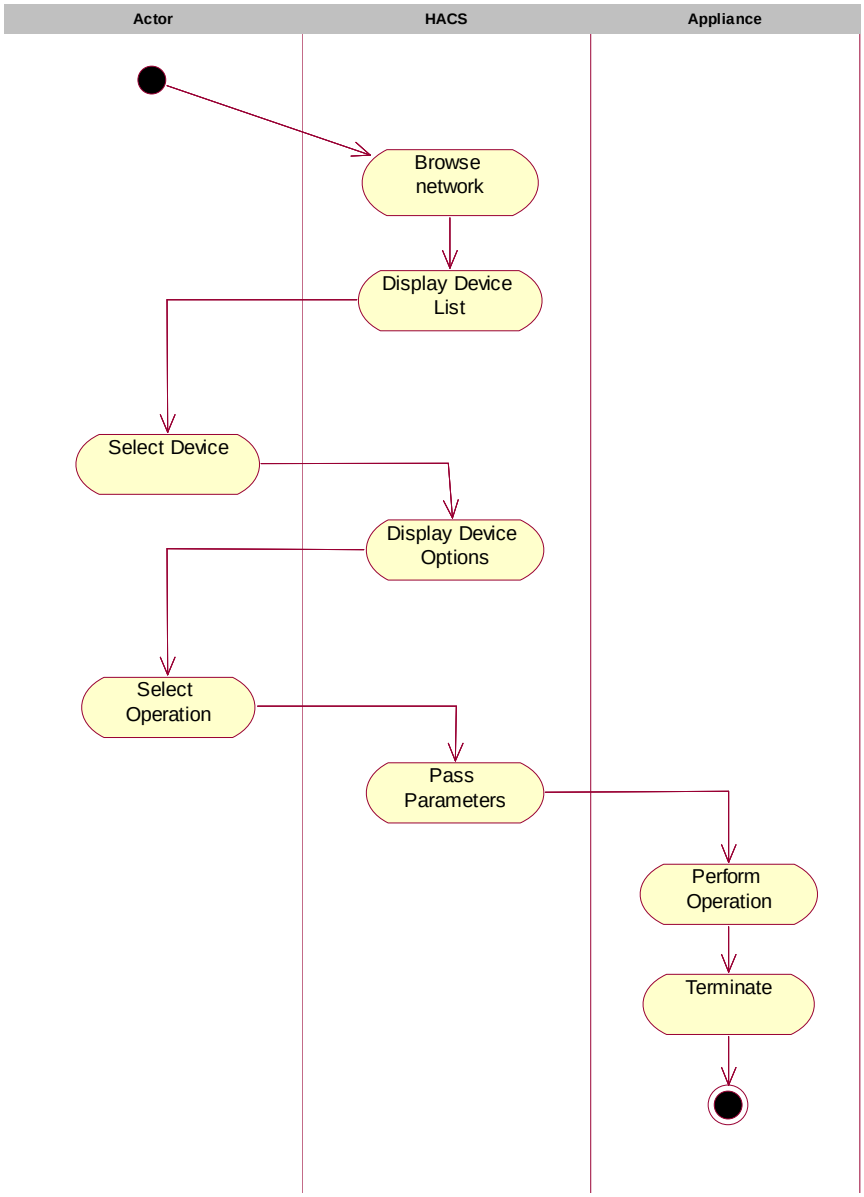
Flow of State Transitions:

- Initially the system is in idle state.
- On event ‘device selected’ it moves to the ‘displaying option’ state by displaying the options.
- On event “turn On” Micro Wave makes the transition into On state
- Event “Set timer” & “Set Power level” changes the Microwave state to Timer Set & Power level set respectively
- Event “Start” switch Micro Oven to cooking state.
- Micro wave terminates upon event stop/timer expires.



8 ACTIVITY DIAGRAM FOR THE WHOLE SYSTEM:

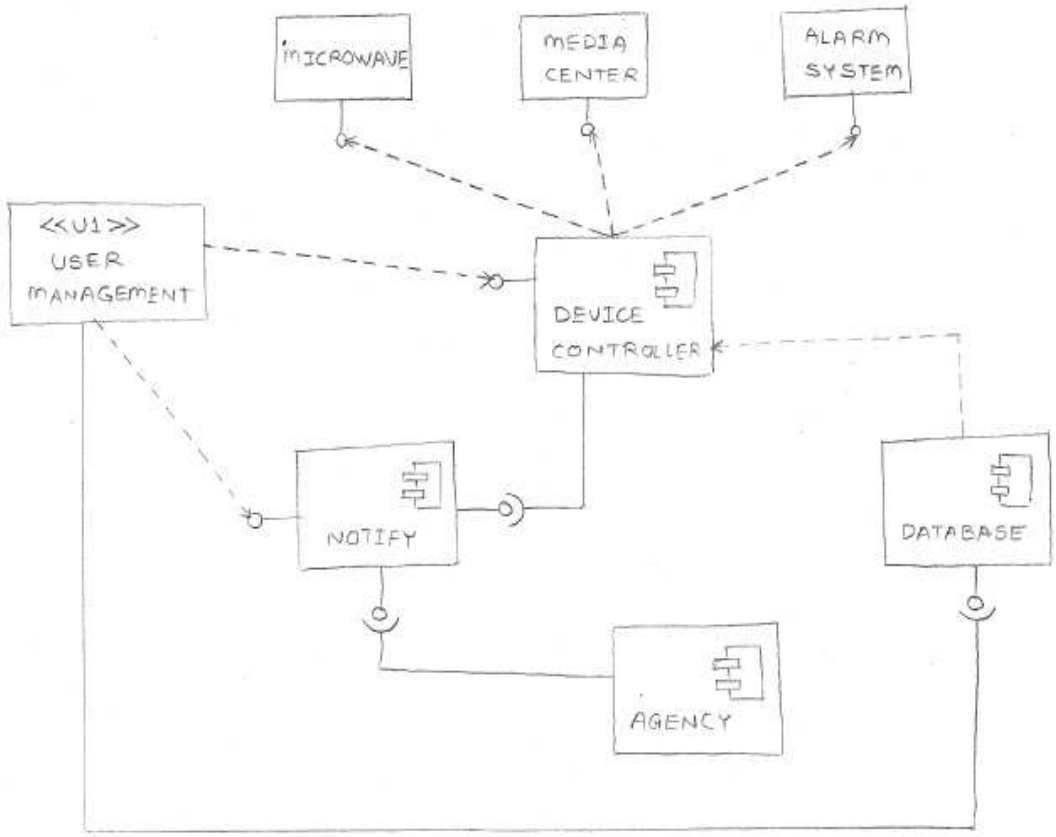
Activity diagrams describe the workflow behavior of a system. .The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel. Activity diagrams should be used in conjunction with other modeling techniques such as interaction diagrams and state diagrams. The main reason to use activity diagrams is to model the workflow behind the system being designed. Activity Diagrams are also useful for: analyzing a use case by describing what actions needs to take place and when they should occur; describing a complicated sequential algorithm; and modeling applications with parallel processes.



DESCRIPTION:

- There are three swim lanes User, HACS and device which specifies a set of activities that share some organizational property.
- In the User, the first activity is performed by an Actor which triggers the sensing of device over network in the HACS.
- HACS display the discovered device list.
- User selects one of the devices which trigger the HACS to display device options.
- User(actors) are allowed to perform operation over device
- The last activity performed is 'terminate' which signals termination

9 COMPONENT DIAGRAM



- In the above component diagram, the ‘User management’ is a user interface which is dependent on the interfaces provided by the device controller and notify. It uses the database to store the information and to validate the user during login.
- The devices which are also components i.e. microwave, Media center and alarm system provide interface to device controller which provides communication between the user and devices.
- The notify component provides interfaces to agency and device controller. Device controller uses the interface of the notify component to set notification and also to provide communication between notify and the devices. The notification sent by the devices will go though the device controller to notify.
- Notify will send that notification to user management and agency.
- Database component is dependent on device controller. The database will be updated according to the changes in the HACS. The changes can be addition of a device to the system, addition/deletion of a user, access rights of the user, etc.

10 TRACEABILITY

Traceability refers to the completeness of the information about every step in a process chain. The formal definition: **Traceability** is ability to chronologically interrelate the uniquely identifiable entities in a way that matters.

The HACS, we have proposed is a generic plug n play device control system. Initially to consolidate our understanding of a plug n play system, we used Use Case Diagrams, Use Case Templates, Sequence Diagrams, class diagrams and CRC cards in Phase I of our project.

In phase II of our project, we have refined the diagrams completed in Phase I reflecting the better understanding of our system. Some more diagrams like Sequence Diagrams with Frames and component diagrams are included to give over all understanding of HACS system. The classes in the class diagrams are realized in the implementation and the working prototype clearly states the fact.

11 REFERENCES

Web references:

http://www.utdallas.edu/~chung/OOAD/syllabus.htm
http://www.utdallas.edu/~weiminma/public/applications.htm
http://www.utdallas.edu/~chung/Fujitsu/UML_2.0/Rumbaugh--UML_2.0_Reference_CD.pdf
http://www.processimpact.com/process_assets/use_case_template.doc

12 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

CRC	-	Class Responsibility and Collaboration
HAC	-	Home Appliance Controller
HACS	-	Home Appliance Control System