Software Design & Architecture

Engr. Abdul-Rahman Mahmood

DPM, MCP, QMR(ISO9001:2000)

armahmood786@yahoo.com
alphapeeler.sf.net/pubkeys/pkey.htm
pk.linkedin.com/in/armahmood
www.twitter.com/alphapeeler
www.facebook.com/alphapeeler
abdulmahmood-sss alphasecure
armahmood786@hotmail.com
http://alphapeeler.sf.net/me

alphasecure@gmail.com
http://alphapeeler.sourceforge.net
http://alphapeeler.tumblr.com
armahmood786@jabber.org
alphapeeler@aim.com
mahmood_cubix 48660186
alphapeeler@icloud.com
http://alphapeeler.sf.net/acms/

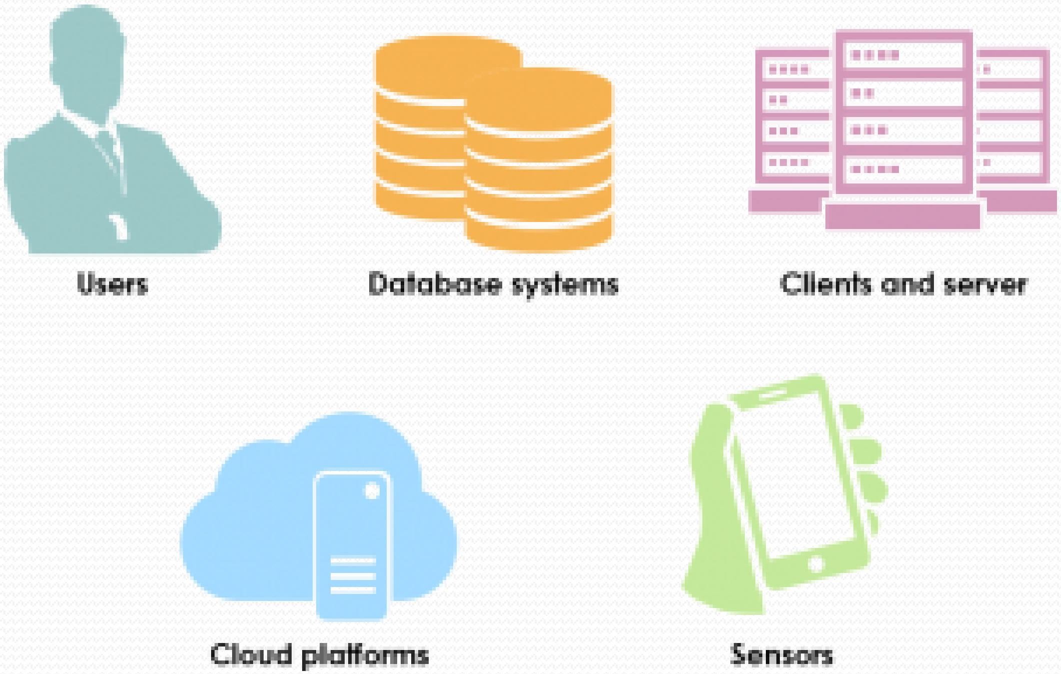
Use Cases

Primary vs Supporting Actors

- P**rimary** actor is often the actor triggering **use case**.
- **Primary Actor:** a user whose defined user goal is fulfilled by the system
  - The primary actor of a use case is the stakeholder that calls on the system to deliver one of its services. It has a goal with respect to the system – one that can be satisfied by its operation.
- **Supporting Actors:** a user who provides a service (e.g., information) to the system.
  - A supporting actor (secondary actor) is external actor that provides a service to the system under design. It might be a high-speed printer, a web service, or humans need to do some research & get back to us.

Another Classification of Actors

- Human
- Systems / Software
- Hardware
- Timer / Clock



- Typically external objects of the system that produce/consume data:
  - Must serve as sources and destinations for data
  - Must be external to the system

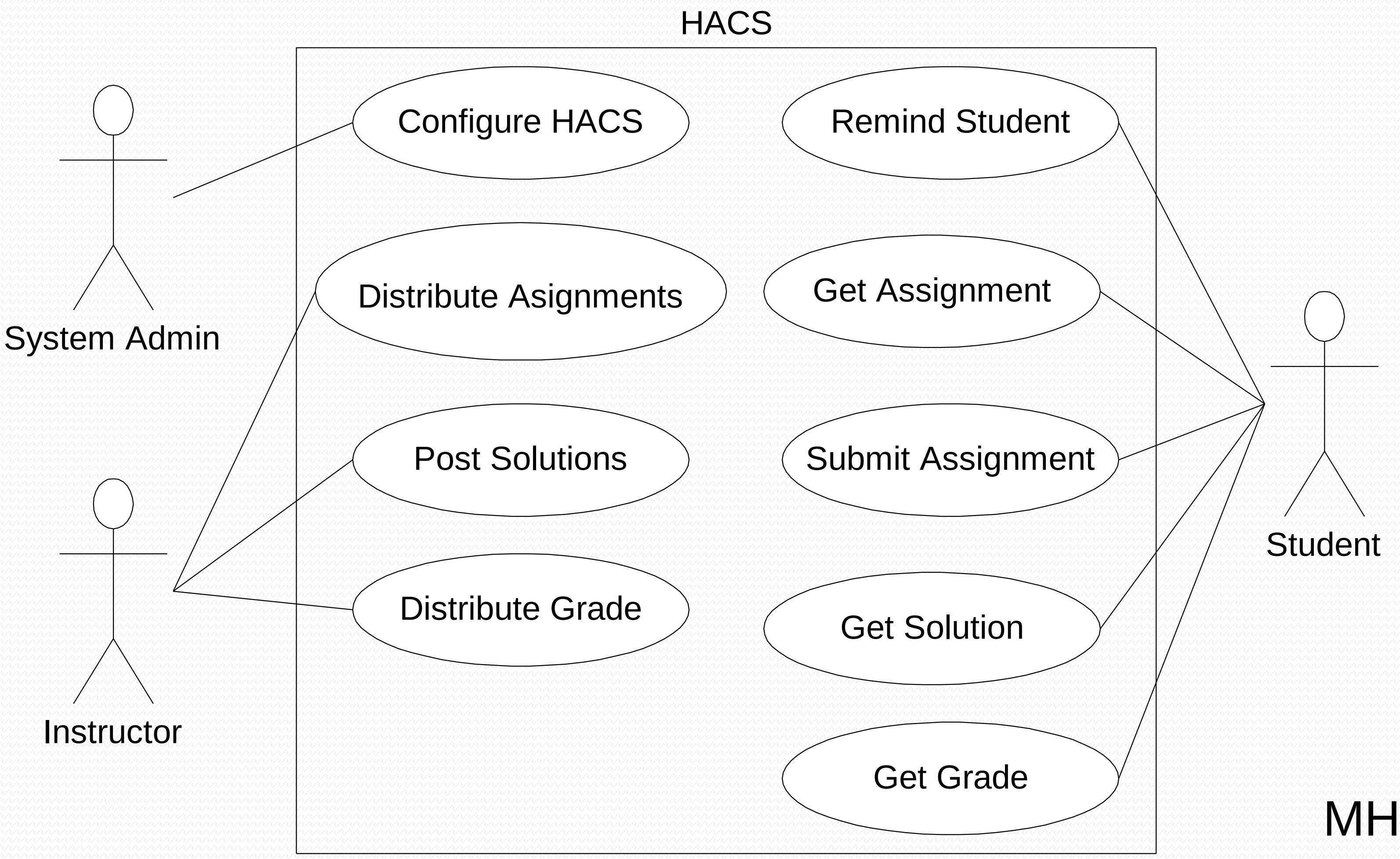
Use case vs Scenarios

- Use case is a collection of related success and failure scenarios.
- Scenario is basically one instance of a use case.
- A scenario is a specific sequence of actions and interactions between actors and the system.
- The scenarios describe actors using a system to support a goal.
- Use case describes a set of sequences. Each sequence is called a scenario
-

Case Study - Homework Assignment & Collection Sys

- Homework assignment and collection are an integral part of any educational system. Today, this task is performed manually. What we want the homework assignment distribution and collection system (HACS for short) to do is to automate this process.
- HACS will be used by the instructor to distribute the homework assignments, review the students' solutions, distribute suggested solution, and distribute student grades on each assignment.
- HACS shall also help the students by automatically distributing the assignments to the students, provide a facility where the students can submit their solutions, remind the students when an assignment is almost due, remind the students when an assignment is overdue.

HACS Use-Case Diagram



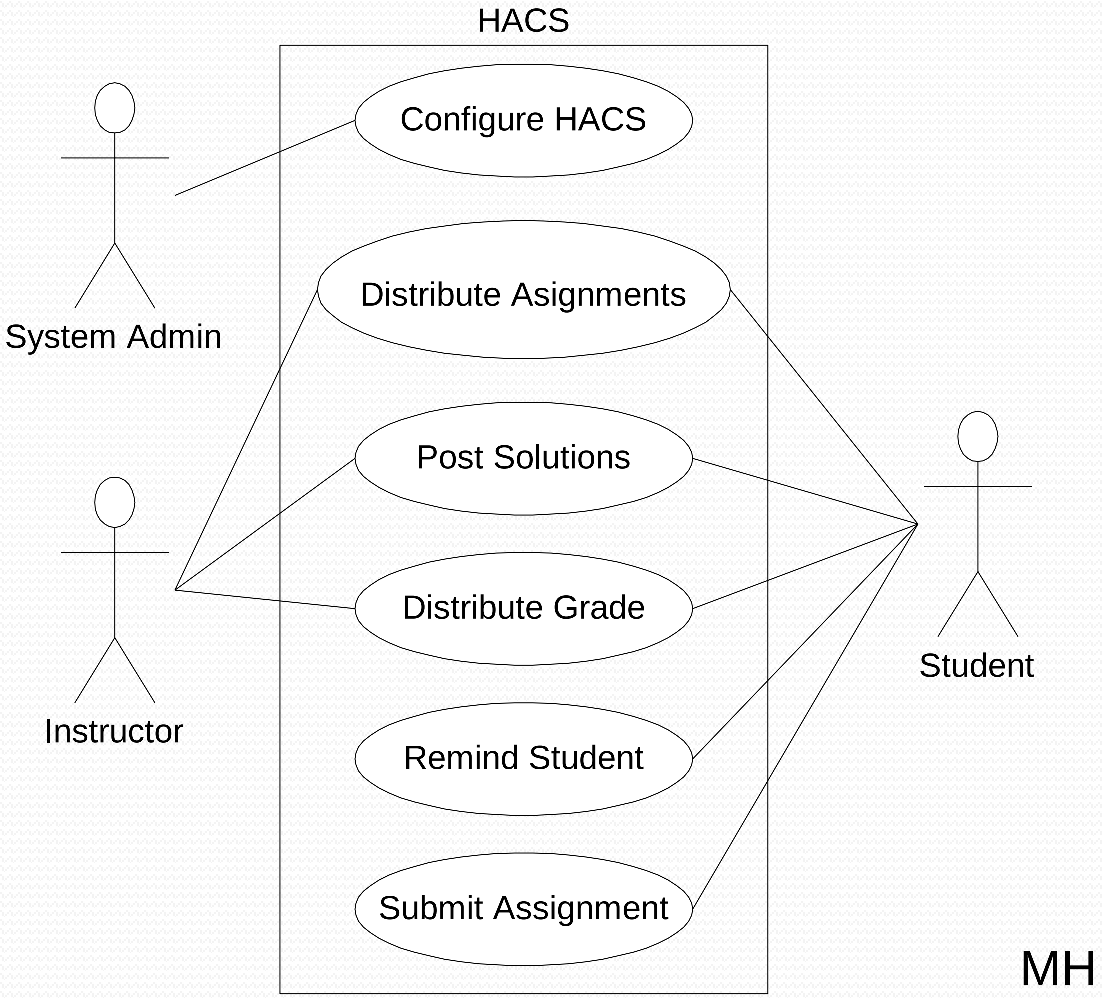
MH

HACS Use-Cases

Use case name: **Distribute Assignments**
Pre conditions: **Configure HACS**
Actors: Instructor (initiator)
Type: Primary and essential
Description: The Instructor completes an assignment and submits it to the system. The instructor will also submit the due date and the class the assignment is assigned for.

Cross Ref.: Requirements XX, YY, and ZZ
Use-Cases: *Configure HACS* must be done before any user (Instructor or Student) can use HACS
Post conditions: File transfer is completed, or file is sent on email.

Alternate HACS



MH

Alternate HACS Use-Cases

Use case: **Distribute Assignments**

Pre conditions: **Configure HACS**

Actors: Instructor (initiator), Student

Type: Primary and essential

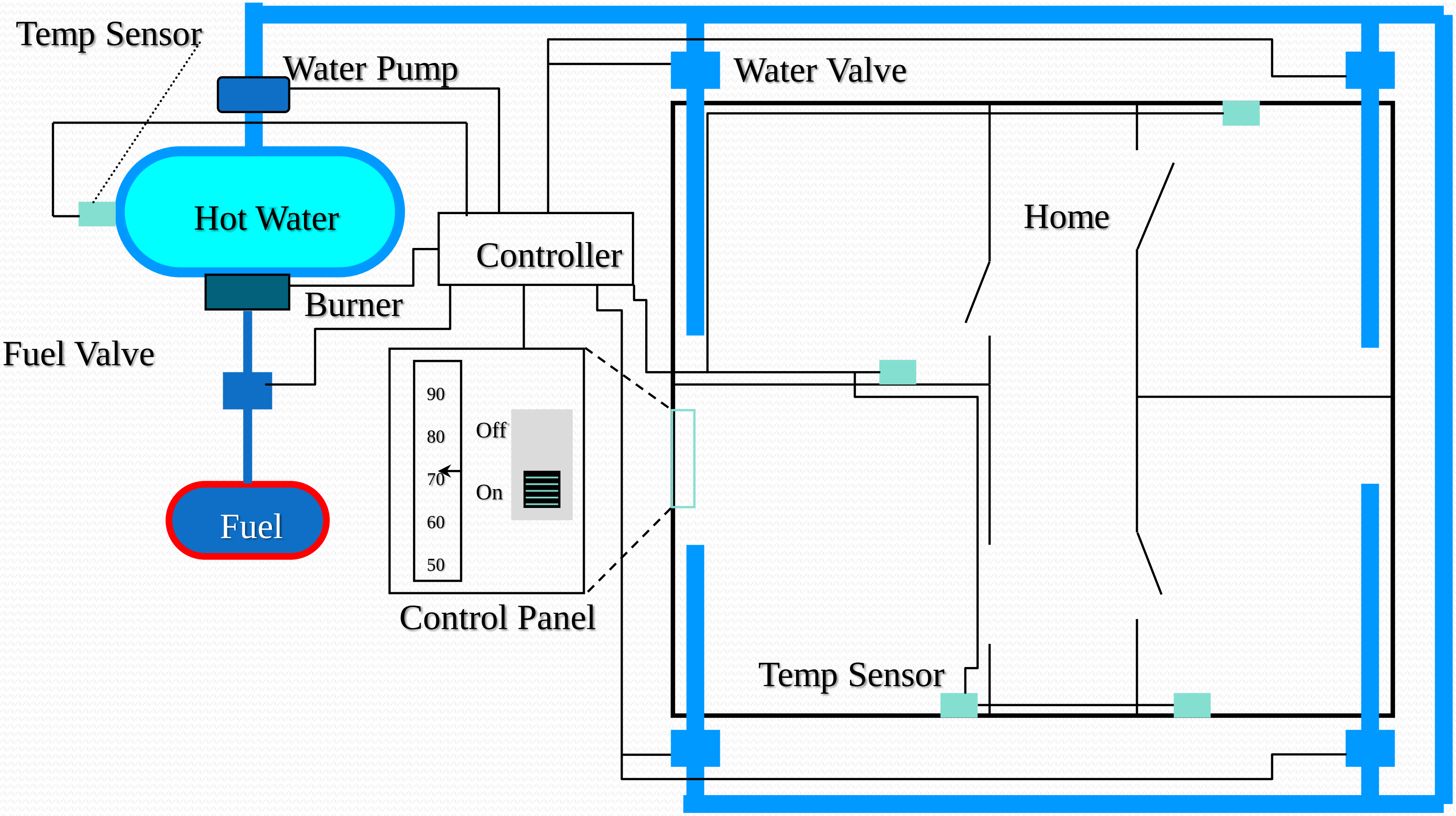
Description: The Instructor completes an assignment and submits it to the system. The instructor will also submit the delivery date, due date, and the class the assignment is assigned for. The system will at the due date mail the assignment to the student.

Cross Ref.: Requirements XX, YY, and ZZ

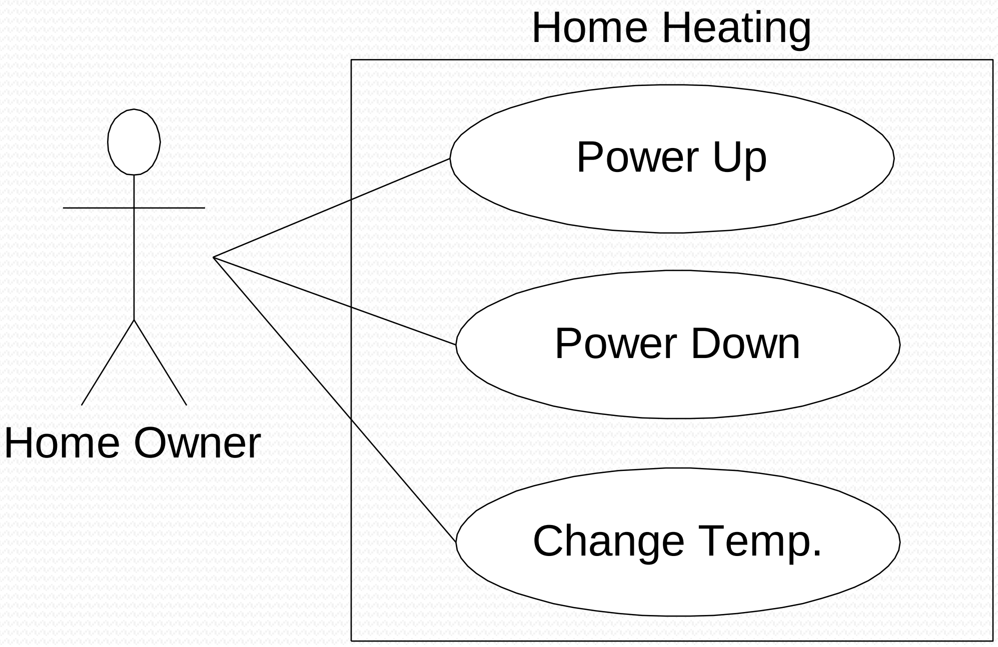
Use-Cases: *Configure HACS* must be done before any user
(Instructor or Student) can use HACS

Post conditions: File transfer is completed, or file is sent on email.

The Home Heating System



Home Heating Use-Case Diagram



MH

Home Heating Use-Cases

Use case: **Power Up**

Actors: Home Owner (initiator)

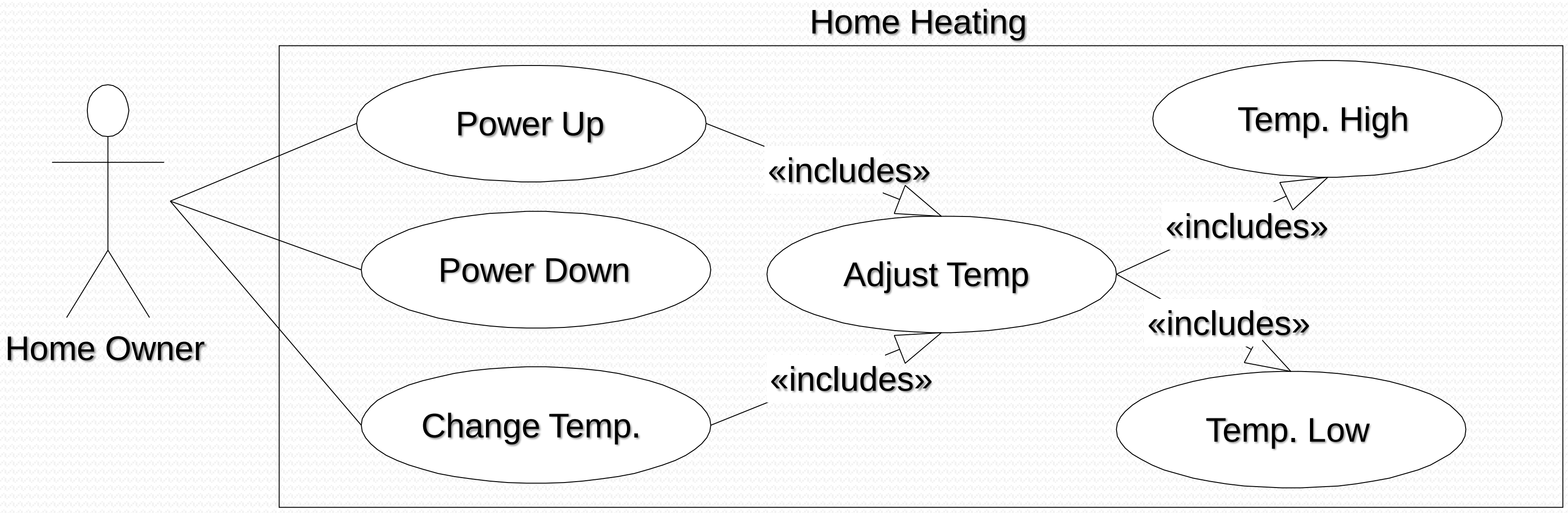
Type: Primary and essential

Description: The Home Owner turns the power on. Each room
is temperature checked. If a room is below the
the desired temperature the valve for the room is
opened, the water pump started. If the water temp falls
below threshold, the fuel valve is
opened, and the burner ignited.
If the temperature in all rooms is above the desired
temperature, no actions are taken.

Cross Ref.: Requirements XX, YY, and ZZ

Use-Cases: None

Modified Home Heating



MH

Modified:
Home Heating Use-Cases

Use case: **Power Up**

Actors: Home Owner (initiator)

Type: Primary and essential

Description: The Home Owner turns the power on. **Perform Adjust Temp**. If the temperature in all rooms is above the desired temperature, no actions are taken.

Cross Ref.: Requirements XX, YY, and ZZ

Use-Cases: Perform Adjust Temp

Modified:

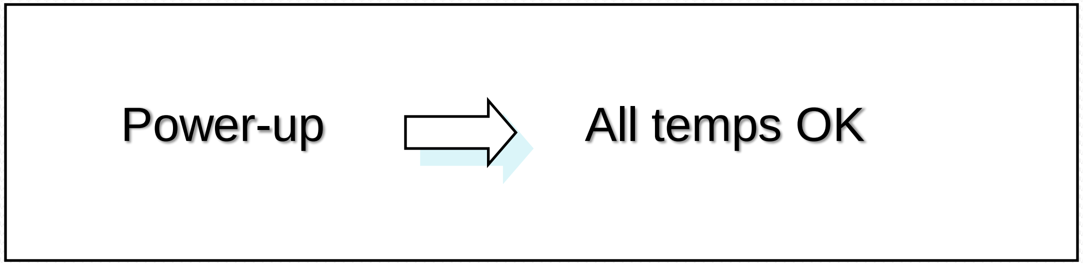
* Home Heating Use-Cases

Use case:	**Adjust Temp**
Actors:	System (initiator)
Type:	Secondary and essential
Description:	Check the temperature in each room. For each room: Below target: **Perform Temp Low** Above target: **Perform Temp High**
Cross Ref.:	Requirements XX, YY, and ZZ
Use-Cases:	Temp Low, Temp High

Modified:
Home Heating Use-Cases

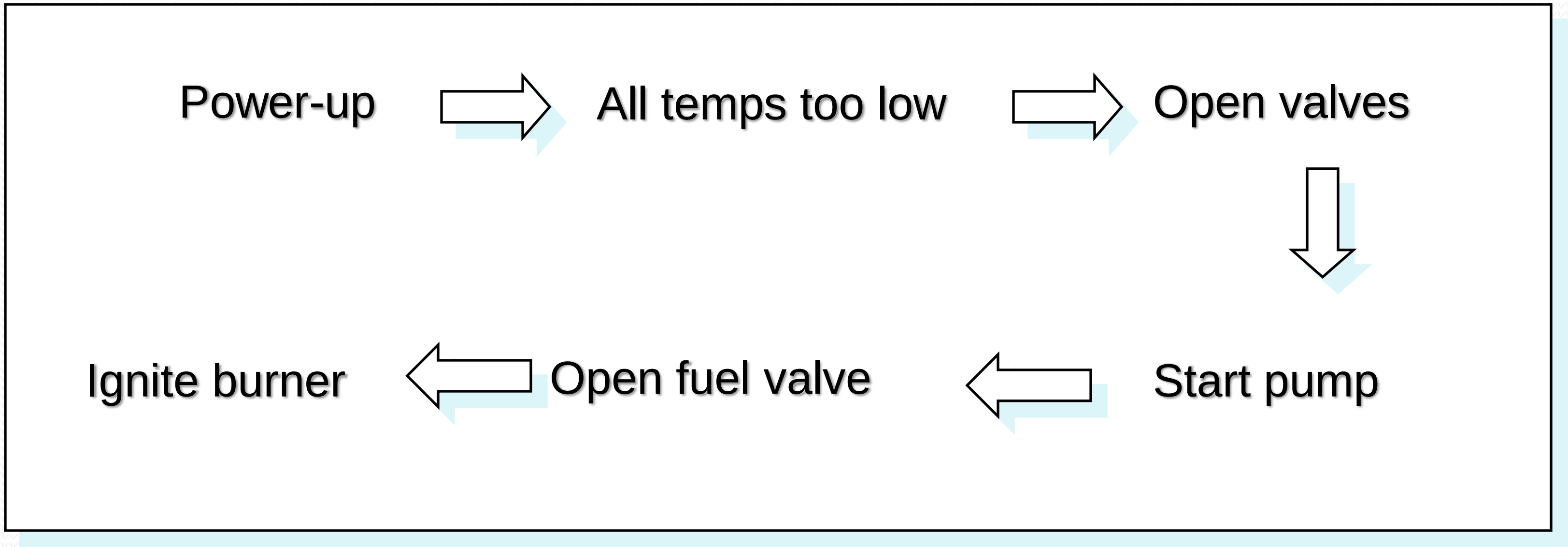
Use case:	**Temp Low**
Actors:	System (initiator)
Type:	Secondary and essential
Description:	Open room valve, start pump if not started. If water temp falls below threshold, open fuel value and ignite burner.
Cross Ref.:	Requirements XX, YY, and ZZ
Use-Cases:	None

Scenarios from One Use Case



Power-up

All temps OK



Power-up

All temps too low

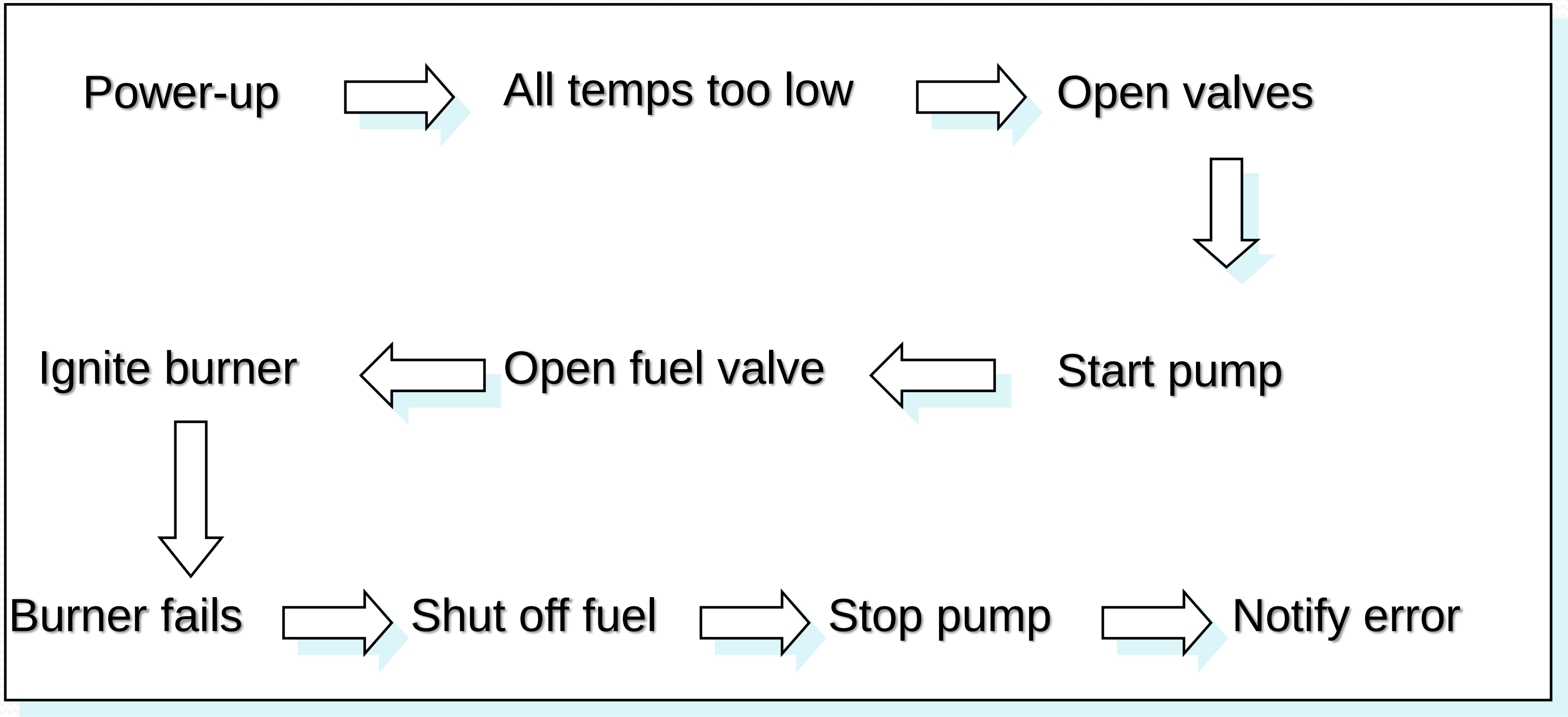
Open valves

Start pump

Open fuel valve

Ignite burner

Scenarios -- One Use Case



Power-up

All temps too low

Open valves

Start pump

Open fuel valve

Ignite burner

Burner fails

Shut off fuel

Stop pump

Notify error

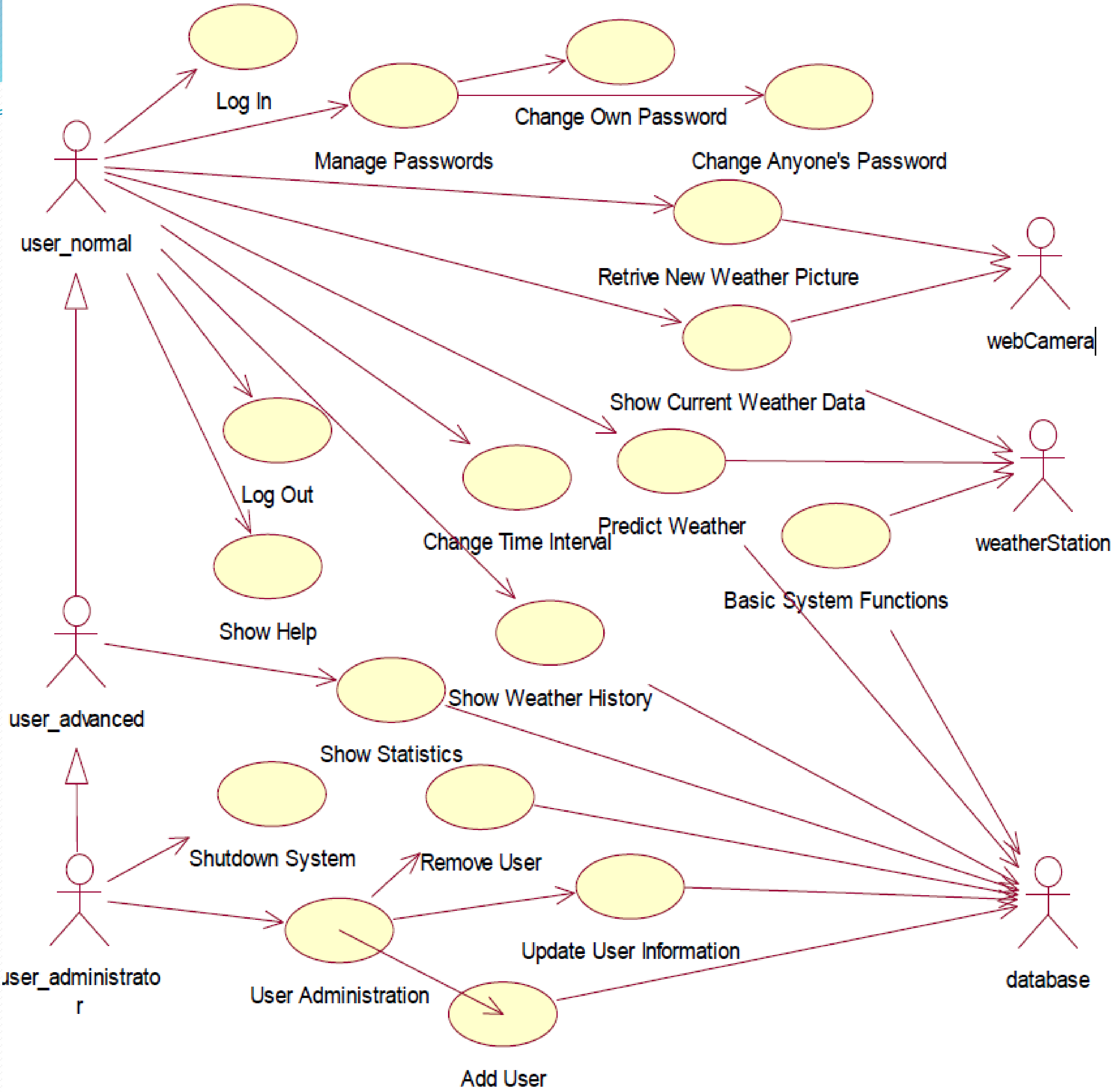
Fully dressed / extended use cases

- Here is an example of an extended use case template:

Use Case Name	Verb and Noun that expresses the goal of the use case, for example, "Deposit Funds."
Assumptions	Conditions that must test true before this use case will execute. This use case assumes that some other use case is taking responsibility to test the condition
Pre-conditions	Conditions that must true before this use case will execute. This use case is taking responsibility for testing the condition.
Initiation (Triggering event)	This use starts when … Idenity how the use case knows when it is time to try to accomplish its goal
Main flow of events	Describe the succesful dialog between the actor and the use case.
Exceptional flow of events	Describe a dialog that does not achieve the stated goal of the use case, i.e. an exception.
Post-conditions	Conditions that must test true before exiting this use case

The weather station project

- A system to get different information from a weather station was analyzed and designed. The system should be able to fetch, process, store and display current weather data from the weather station as well as historical data from a database. The system should also display a weather prediction based on a comparison of the current weather and stored weather data. A picture from a web camera should also display current weather. A special group of users should be able to compute some weather statistics from the weather system. Find use cases and actors for the weather system started the project. The use cases and the actors can be seen in figure on next page.


Log In
Manage Passwords
Change Own Password
Change Anyone's Password
user_normal
Retrive New Weather Picture
webCamera
Show Current Weather Data
Log Out
Change Time Interval
Predict Weather
weatherStation
Show Help
Basic System Functions
Show Weather History
user_advanced
Show Statistics
Shutdown System
Remove User
Update User Information
database
user_administrator
User Administration
Add User

Fully dressed / extended Use case description

- **Use case: Manage passwords**
- **Primary actors:** User, administrator
- **Interests:** User, administrator
- **Brief Description**
- The user or administrator request change of password.
- The system displays the correct dialog depending on the group membership (admin or not)
- **Flow of Events**
- *Basic Flow*
- 1. The user request change of password
- *Alternative Flows*
- 1a. If it is the administrator requesting change of password, *see use case Change everybody's password*.
- 1b. If it is not the administrator requesting change of password, see use case *Change own password*.
- **Pre-conditions**
- User/administrator logged in.
- **Post-conditions**
- A successful change of password.

- **Use case: Show weather statistics**
- **Primary actor:** Advanced user
- **Interests:** Advanced user
- **Brief Description :** The user requests weather statistics. The system collects the weather data from the database and presents the result on the screen.
- **Flow of Events**
- *Basic Flow*
- 1. The user requests statistics of the weather.
- 2. The system displays the “Statistics” dialog.
- 3. The user chooses between statistics for the last day, the last week or the last month.
- 4. The system collects weather data for the specified time interval from the database.
- 5. The system presents the data on the screen.
- *Alternative Flows*
- 4a-5. The database is unavailable:
- 1. The system displays an error message.
- **Pre-conditions**
- Advanced user logged in.
- The user is member of the advanced user group
- **Post-conditions**
- The statistics asked for were presented on the screen