Software Engineering

Group 10

UNIVERSITY OF BIRMINGHAM

Design of Smart Home System for Smart Cities and Smart Nations

Abasin Entegam	2180250	axe150
Chit Lee	2070900	cxl203
Daniel Ho	2417053	dxh253
Ching Hong Lai	2341862	cxl234
Pratham Arora	2338168	pxa168
Praveen Rajendran	2422147	pxr247
Yeni Jung	2308283	yxj183
Yugal Khanal	2302704	yck104

A. Requirements Engineering

A1.

Background

Smart city is a hot topic as with the current technological advancement, we can analyse big data to improve the quality of life of citizens. Therefore data collection is very important to decision-making.

Also, with the rise in living costs and increasing environmental awareness. Families want to use energy wisely and avoid unnecessary energy consumption in order to reduce costs. But without knowing where the energy is spent, it is hard to do that.

Our aim

Therefore we propose this smart home system so that the user can precisely know which home appliances are using how much electricity so they can make educated decisions on how to save energy.

We also aim to streamline submitting energy readings, by having the smart home system report the energy usage to the energy provider directly and get the bill.

Another aim is to help the city council understand the energy usage patterns of its residents. So they can give better advice on how to save energy, as well as make better policies.

Basic functionality

The smart home system is a supervisory control and data gathering software, it allows users to control appliances in their home and collect energy usage data from appliances. The user can see the energy usage of different appliances, turn on/off appliances, operate smart devices, and set up routines. The smart home system also warns about unusual energy usage, sends energy usage and receives energy bills from the energy provider, and sends energy usage to the city council.

What it is connected to

The smart home system itself includes a smart home device and a smartphone app for users to interact with the system, it connects to

- Smart appliances so that the smart home system can operate them
- Smart outlets, for the non-smart appliances, the smart outlet can help turn on/off the appliances, as well as track the energy usage
- The energy provider, to submit energy usage and receive energy bill
- The city council, providing energy usage data to them

Assumptions

The scope of this smart home system is the software that is on the smart home system device.

- The smart home system includes a smart home device, and a smartphone app for the user to interact with the system
- The energy usage that we track is only electricity
- Smart appliances in the user's home can track how much energy they are using
- The user will plug in smart outlets before any non-smart appliances, which enables:
    - The smart home system is able to identify all appliances connected to the home's energy grid
    - The smart home system is able to monitor the energy usage of all appliances connected to the home's energy grid
- The smart home system is able to cut/restore electricity from each appliance by sending the command to the smart outlet
- The smart home system is connected to the home wifi, so it is able to communicate with the user's app, smart devices, and the energy provider

Details of each functionality

Track energy usage of appliances

The smart home system constantly monitors the energy usage of all the appliances by receiving the energy readings from the smart devices it's connected to and logging them.

Show the user the energy usage

The user will be presented with the continuously updated usage date on the smartphone app.

Operate smart appliances

The user can control smart appliances via the app, they would select the smart device on the app and select the action they want to perform.

Turn on/off appliances

The user can turn on/off appliances by selecting them on the app. If the appliance is a smart device, the smart home system will send a signal to them to turn itself on/off. However, if the appliance is not smart, the smart outlet connected will cut/restore the power.

Warn about unusual energy usage

While the smart home system is monitoring the energy usage, if anything abnormal about the energy usage is detected, the smart home system will notify the user.

Routines

The smart home system can be configured to do certain actions when a condition is met.
The trigger (cause of action) can be a certain time or a certain signal from a smart device.
The action can be to turn on/off an appliance or operate a smart device.

Send energy usage to the energy provider and receive bills

At the end of the month, the smart home system will automatically send the energy usage data to the energy provider and receive the energy bill from them.

Send energy usage data to the council

At the end of the month, the smart home system will send the energy usage data to the city council if the user opted for this service.

A2.

The following functional and non-functional requirements are how we think the system should behave after brainstorming and consulting with our friends.

Functional requirements

1) Operation
    - The smart home system should be able to register/unregister smart outlets using their MAC address
    - The smart home system should be able to register/unregister smart appliances using their respective communication
2) Store energy usage of appliances
    - The smart home system should store the energy usage data from the smart outlets/appliances
    - The above data should be stored for a year
    - The smart home system should recognise if a smart outlet/appliance was unplugged and plugged in again
3) Show the user the energy usage
    - The smart home system should show the energy usage data when the user requests it
    - The data should be presented graphically and numerically
4) Operate smart appliances
    - The smart home system should operate smart appliances when the user requests it
    - It should do so by calling the API of that smart appliance
5) Turn on/off appliances
    - The smart home system should turn on/off an appliance when the user requests it
    - It should do so by calling the API of the smart outlet or smart appliance
6) Warn about unusual energy usage
    - The smart home system should warn the user about unusual energy usage
    - The smart home system should detect if energy usage is higher than usual
7) Routines
    - A routine consists of a trigger, which can be a specific time, or a signal from a connected appliance
    - A routine consists of an action, which is to send a signal to an appliance
    - The smart home system should be able to add routine
    - The smart home system should be able to edit routine
    - The smart home system should be able to remove routine

    The smart home system should carry out the routine when the condition is right
8) Send energy usage to the energy provider and receive bills
    - The smart home system should send the energy usage data to the energy provider
    - The transmission should be encrypted
    - The smart home system should be able to authenticate itself with the energy provider

- The smart home system should receive the energy bill from the energy provider

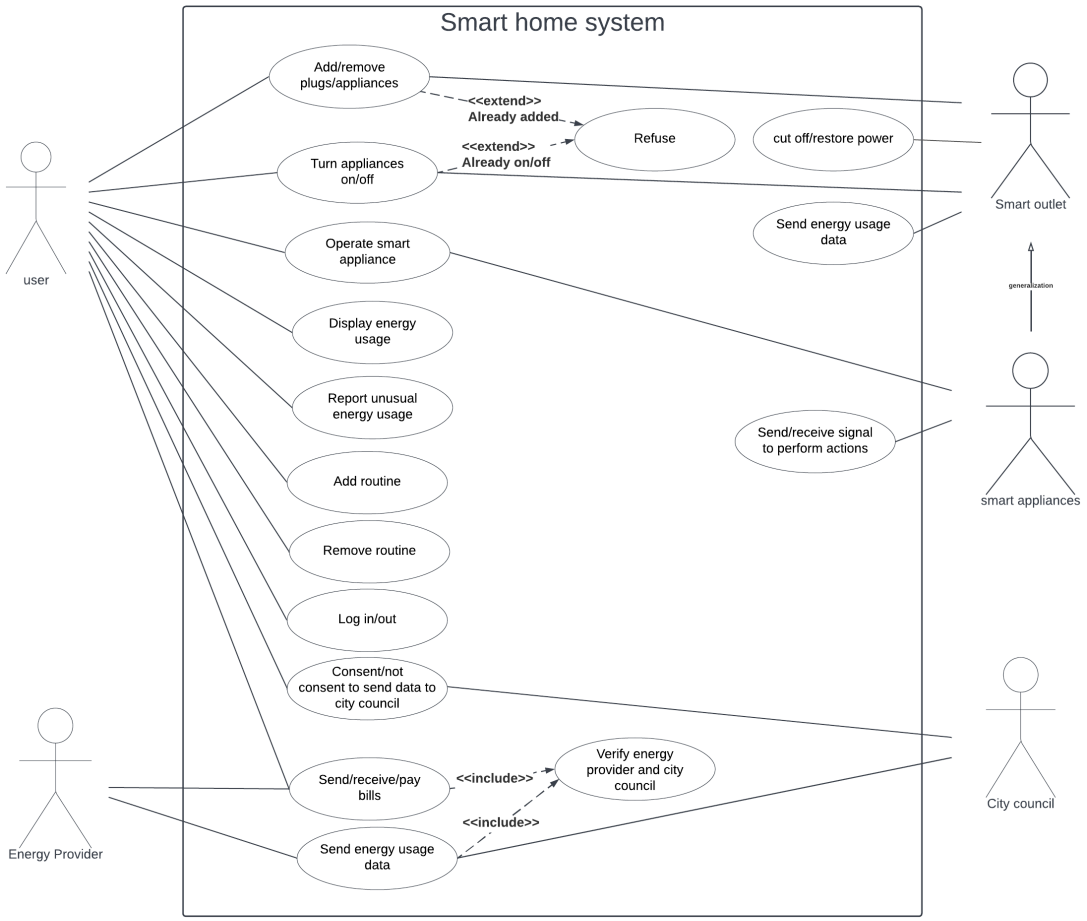
9) Send energy usage data to the council
   - The smart home system should send the energy usage data to the council if the user opted in
   - The user should be able to opt in/out of sending energy usage data to the council

Non-functional requirements

1) Availability
   a) The smart home system should operate 24/7
2) Security
   a) The smart home system energy usage storage should not be manipulated in unintended ways
   b) The smart home system should not be able to be controlled by unauthorised users
   c) The smart home system requires 2-step verification from the user
3) Reliability
   a) The smart home system should provide an accurate energy reading
   b) The smart home system should make sure the command sent to smart outlets are executed
   c) The smart home system should make sure the command sent to smart appliances are executed
4) Scalability
   a) The smart home system should support having more than 100 smart outlets and more than 100 smart appliances
5) Efficiency performance
   a) The smart home system should use as little energy as possible
   b) The smart home system should display energy usage to the user within 1 second
   c) The smart home system should turn on/off appliances within 1 second
   d) The smart home system should operate smart appliances within 1 second
6) Usability
   a) The text should be big enough so normal users can see them
   b) There should be a readability mode with larger text and high contrast so people with bad eyesight can use the system
7) Legislative privacy
   a) The smart home system should only send the energy usage data to the city council if the user consents
   b) The user should agree with the terms of service and privacy agreement before using the system
8) Compatibility
   a) The smart home app should be usable on different mobile operating system

B. Software Specification, Analysis and Design with U

B1.



B2.

Use case: add routines	
Actors	The user
Preconditions	The smart home system is powered up. The appliances that will be involved in this routine are connected to the smart home system. The smart home system is connected to the network The app is on the home screen.
The flow of event	1. The use case starts when the user clicks the set routine and then the add routine button 2. The app switches to the add routine interface 3. The system loads the connected appliances 4. The user can choose a trigger for the routine, between a certain time, or a smart appliance a. If “time” is selected, the user will need to input what time of the day, and how often this repeats b. If “smart appliance” is selected, the user will need to choose from a list of all smart appliances that support sending a signal to the smart home system 5. The user can choose an appliance to operate as the action a. If a smart appliance is chosen, the app will list out all supported actions b. If a non-smart appliance is chosen, the only action possible is to turn it on/off 6. The app displays the combination the user chooses, and asks the user to confirm 7. After the user clicks to confirm, the smart home system adds the routine, the app acknowledges it has been added and goes back to the main screen
Post-conditions	The app is in the home screen The new routine is added to storage The new routine should be active
Alternate flow	In step 2-6, if the user clicks cancel, the app returns to the home screen, the and routine is discarded and does not get stored. In steps 4 and 5, if the user can cancel the previous selection by clicking back, and the user can select another option from the previous page, the choice will be updated when the user selects another item

Use case: Send energy usage data	
Actors	Energy provider, city council
Preconditions	Smart outlets and appliances are functioning properly Smart home system is powered up All devices needed are connected to the network
Flow of events	1. The use case starts when the date reaches the end of month 2. The energy usage calculator verifies the energy provider host addresses stored 3. The energy usage calculator verifies the city council host address stored 4. The energy usage calculator verifies itself to the energy provider and city council 5. The energy usage calculator sends the energy usage data to the energy provider 6. If the user consented a. The energy usage calculator sends the energy usage data to the city council 7. The smart home system awaits confirmation from the energy provider and the city council
Post-conditions	The energy provider and the city council receive energy usage from the smart home system. The smart home system gets confirmation of the usage data being received by the energy provider and city council.

Use case: Operate smart appliance	
Actors	Users, Smart appliances
Preconditions	The smart home system is powered up The smart devices are connected to the smart home system The app is in the home screen
Flow of events	1. The use case starts when the user clicks “operate smart appliance” in app 2. A list of smart devices is displayed on the screen by the app 3. The user issues a specific command to the corresponding device 4. Once the operation is selected, the smart home system sends a signal to the smart appliance to perform action 5. The smart device that receives an execution signal performs the corresponding operation and sends a returning signal that confirms it is performing the operation 6. The smart home system reflects the signal from smart appliances and changes the current state of the device in the app 7. The app displays the list of appliances in the event that other actions are further carried out a. If the actions are selected, step 3 to 6 is repeated
Post-conditions	The smart appliance performs the operation The app displays the map of smart devices
Alternate flow	During step 2-3, if the user hit the back button, the app will go back to the previous step

Use case: Display energy usage	
Actors	Users
Preconditions	The smart home system is powered up The energy usage is tracked by the smart home system which connects all smart devices The energy usage is stored The app is in the home screen
Flow of events	1. The use case starts when the user press the “view energy usage” button 2. The app requests the data from the smart home system 3. The smart home system calculates the data and sends it back 4. The app reads the accumulated energy usage from local storage and displays it on the screen a. Energy usage is initialised on the first day of each month b. Energy usage is updated every 10 minutes to reflect current usage c. The screen also shows how much energy is being used per second by which device d. The monthly energy usage graph for a year can be checked on a split screen, the energy usage is only stored for a year and disposed of 5. If the user clicks exit, the app returns to the home screen
Post-conditions	The app displays either energy usage of the smart home or home screen

B3.

1. User James opens the app and selects the “set routine” and “add routine” button which takes him to the add routine menu. He wants to add a routine based on the time of the day, thus he clicks the time button. The app requests what time he wants before showing a list of actions that can be done. Joe then chooses 12:30 pm. The app shows Joe a list of connected appliances. From here, James selects the fan in the living room and selects turn on. After that, the app shows him the summary of actions supposed to be carried out at 12:30 pm. The screen shows “12:30 pm - Turn on the living room fan” and asks him to confirm the action. James selects the confirm button, prompting the app to store the active routine in the storage and he is brought back to the main menu.

2. User John signed a contract with the energy provider on January 1st 2022 and consented to the city council to collect energy usage data from him. A month after the contract was signed - January 31, the smart home system recognizes that tomorrow is the billing date. It first verifies the energy provider and city council host address stored, and verifies itself to the energy provider and city council. Then, it gathers the energy usage amount and sends it to the energy provider, and gathers the energy usage data, which is more detailed, and sends it to the city council. The energy provider sends the confirmation back to the smart home system to confirm that John uses 900kWh electricity this month. The city council sends the confirmation that the data is received.

3. User Justin wants to switch off the lights in the living room. First, Justin will have to select the “Operate Smart Device” function from the menu. Then he will have to pick the smart device that he wants to take control of, in this case, it would be the lights in his living room. Then Justin will be prompted with the option of turning on or off the lights in his living room. If Justin chooses to turn off the lights, the smart home system will then send a signal to the smart lights to be turned off. After that happens, Justin still can select other smart devices to take control of or he can return to the main menu.

4. User Joseph wants to check the cumulative energy usage from the beginning of the month to the present and how much energy he is currently consuming. He accesses an app with a smart home system and clicks "Display energy usage". The converted app screen displays the accumulated data from the 1st of the month to the present and the data that shows the current usage of energy of bedroom lights, kitchen gas stoves, and living room TV using 1W, 167W, and 30W of energy per 10 minutes, respectively. After checking the energy usage, he presses the “Return to Home Screen” button and switches the app to Home Screen.

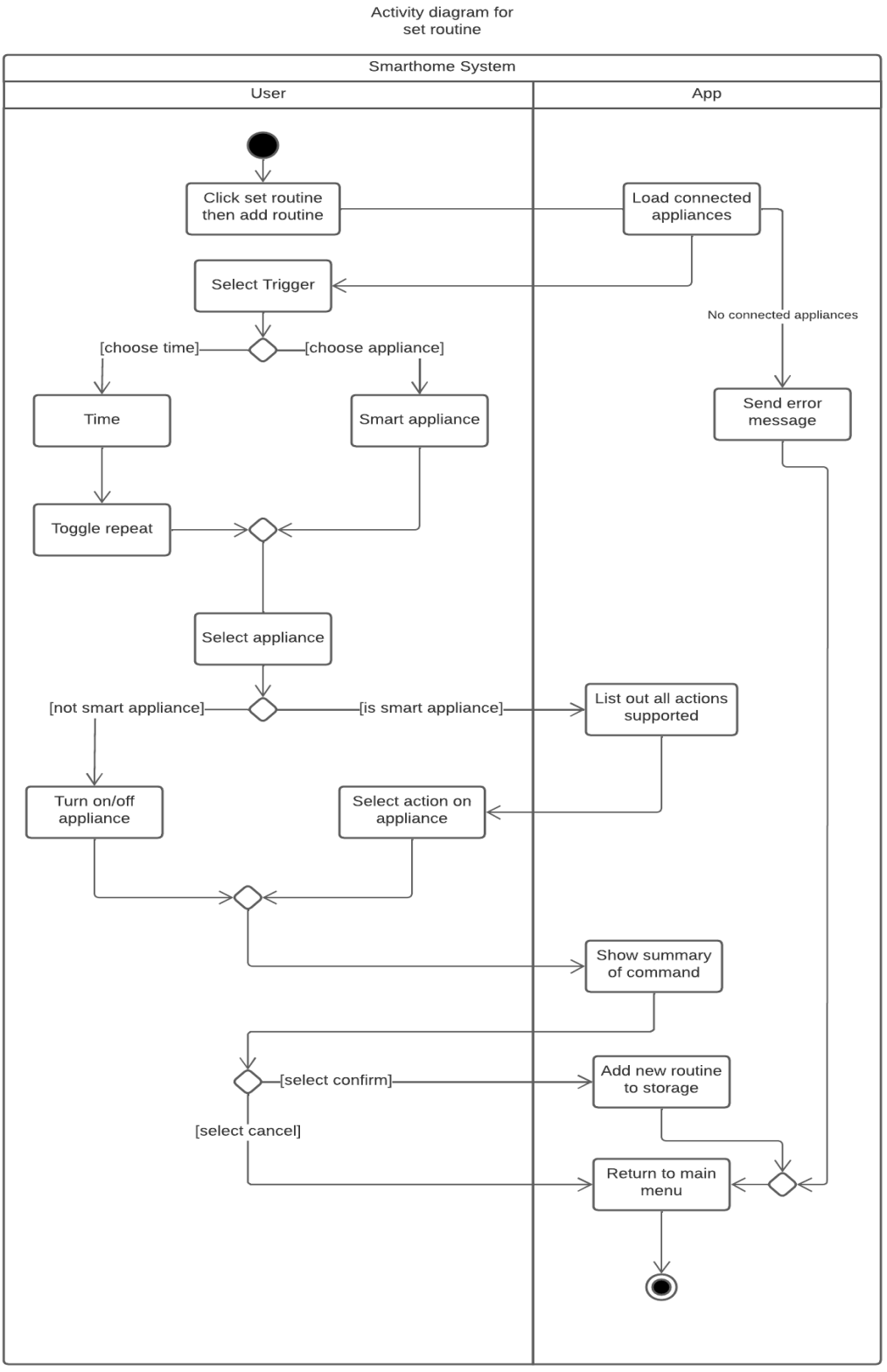
5. User Joe opens the app and selects the “set routine” and “add routine” buttons, which takes him to the add routine menu. He selects a smart appliance trigger. The app loads the connected smart appliances. Joe selects the entrance hallway sensor as a trigger when crossed. The app then shows a list of appliances as the action. He selects the lights appliance to turn on upon receiving the trigger. The app shows the summary of the command and asks him to confirm the action. He selects the confirm button and the active routine is stored in the storage while he is brought back to the main menu.

6. User Mary opens the app and tries to turn on the water kettle, she selects the water kettle on the “Operate Smart Device” function from the menu and presses the turn-on button. After the button was pressed the app sent a signal to the smart outlet that is attached to the water kettle and turned on the water kettle. Mary sees on the app that the water kettle is turned on.

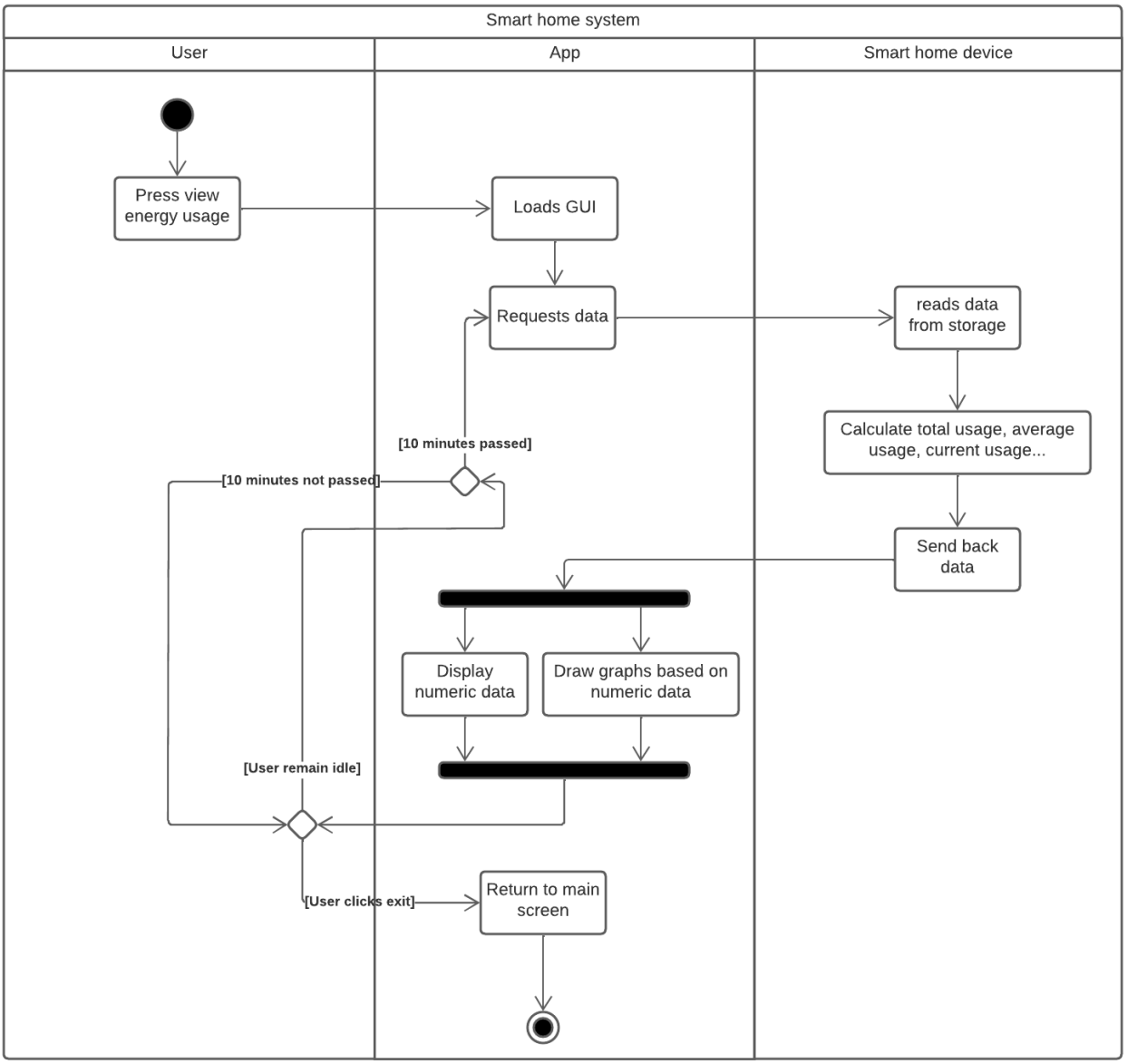
7. User Sam signs an energy contract with Jason Energy Co but didn’t consent to share energy usage data with the city council. After a month, the smart home system verifies the energy provider host address, verifies itself to the energy provider, and sends the energy usage data to the energy provider. Then it receives a confirmation from the energy provider.

8. User Oliver wants to check how much energy was used for a computer in an hour. He opens the app and selects “Your energy usage” and selects the smart outlet which is connected to the computer. The app shows the usage data of the energy that has been used in the past 1 hour is 140 watts, after that Oliver presses the back button and returns to the main menu.

B4.


Activity diagram for
set routine
Smarthome System
User
App
Click set routine
then add routine
Load connected
appliances
Select Trigger
No connected appliances
[choose time]
[choose appliance]
Time
Smart appliance
Send error
message
Toggle repeat
Select appliance
[not smart appliance]
[is smart appliance]
List out all actions
supported
Turn on/off
appliance
Select action on
appliance
Show summary
of command
[select confirm]
Add new routine
to storage
[select cancel]
Return to main
menu

Activity diagram for
display energy usage



B5.

A. Noun - Verbal analysis

Specifications	Analysis
Operation a) The smart home system should be able to register/unregister smart outlets using their MAC address b) The smart home system should be able to register/unregister smart appliances using their respective communication	Classes identified based on common nouns : -> Smart outlet -> Smart appliance -> Connected devices Operations identified based on doing verb: ->Register ->Unregister Members identified based on common nouns: -> MAC address on smart outlets -> serial / connection number on smart appliance
Store energy usage of appliances a) The smart home system should store the energy usage data from the smart outlets/appliances b) The above data should be stored for a year c) The smart home system should recognise if a smart outlet/appliance was unplugged and plugged in again	Classes identified based on common nouns: -> Energy usage storage -> Smart outlet -> Smart appliance Operations identified based on doing verb: -> Store data -> Input data Members identified based on common nouns: -> Energy usage date -> Plug detection
Show the energy usage a) The smart home system should show the energy usage data when the user requests it b) The data should be presented graphically and numerically	Classes identified based on common nouns: -> Smart outlet -> Energy usage storage -> Connected devices Inheritance identified based on being verb: -> Smart outlet being an extension of Smart/non-smart appliances: -> Smart Outlet inherits Smart appliances Operations identified based on doing verb: -> Display energy usage -> Monitoring energy usage
Operate smart appliances	Classes identified based on common nouns: -> Appliances controller -> Routine executor

a) The smart home system should operate smart appliances when the user requests it b) It should do so by calling the API of that smart appliance	-> Smart appliances -> Smart outlet Operations identified based on doing verb: -> Operate Appliance -> Operate -> Turn on -> Turn off
Turn on/off appliances a) The smart home system should turn on/off an appliance when the user requests it b) It should do so by calling the API of the smart outlet or smart appliance	Classes identified based on common nouns: ->Routine executor -> Appliances controller -> Smart outlets -> Smart appliances Operations identified based on doing verb: -> Operate appliance -> Turn on -> Turn off -> Operate
Warn about unusual energy usage a) The smart home system should warn the user about unusual energy usage b) The smart home system should detect if energy usage is higher than usual	Classes identified based on common nouns: ->Unusual energy usage detector -> Energy usage storage -> Energy usage calculator Operations identified based on verb: -> Check for unusual energy data -> Retrieve data
Routines a) A routine consists of a trigger, which can be a specific time, or a signal from a connected appliance b) A routine consists of an action, which is to send a signal to an appliance c) The smart home system should be able to add routine d) The smart home system should be able to edit routine e) The smart home system should be able to remove routine f) The smart home system should carry out the routine when the condition is right	Classes identified based on common nouns: -> Routine executor -> Smart appliances -> Smart outlets -> Appliance controller Operations identified based on verb: -> Add routine -> View routine -> Remove routine -> Operate -> Turn on -> Turn off -> Get appliances -> Operate appliances
Send energy usage to the energy provider and receive bills a) The smart home system should send the energy usage data to the energy provider b) The transmission should be encrypted	Classes identified based on common nouns: -> Energy usage storage -> Energy usage calculator -> Energy provider Operation identified based on verb:

c) The smart home system should be able to authenticate itself with the energy provider d) The smart home system should receive the energy bill from the energy provider	-> Retrieve data -> Get usage summary -> Send data -> Receive bill
Send energy usage data to the council a) The smart home system should send the energy usage data to the council if the user opted in b) The user should be able to opt in/out of sending energy usage data to the council	Classes identified based on common nouns: -> City council -> Energy usage calculator -> Energy usage storage Operations identified based on verb: -> Get usage summary -> Retrieve data -> Send data

B. CRC cards:

App	
Create account Log in Log out	Routine executor Connected devices Energy usage storage Unusual energy usage detector Energy provider API

Energy-usage storage	
Store energy usage Give out energy usage data	Smart appliances Smart outlet Energy usage calculator

Routine executor	
Store routines given by user Execute a routine on smart appliances given by user	Connected devices Smart appliance

Connected devices	
List and store connected smart appliances List and store connected smart outlets Send controls to smart appliance/outlets	Smart outlets Smart appliance

Smart appliance	
Have serial number / connection number Carry out actions supported by smart appliance Send signals	Appliance controller Routine storage

Smart outlet	
Have MAC address Track energy usage Turn on/off an outlet Receive routines Connect to the smart home system	Appliance controller Routine storage

City council API	
Receive data from the smart home system Verify data sent	Energy-usage calculator Verifier

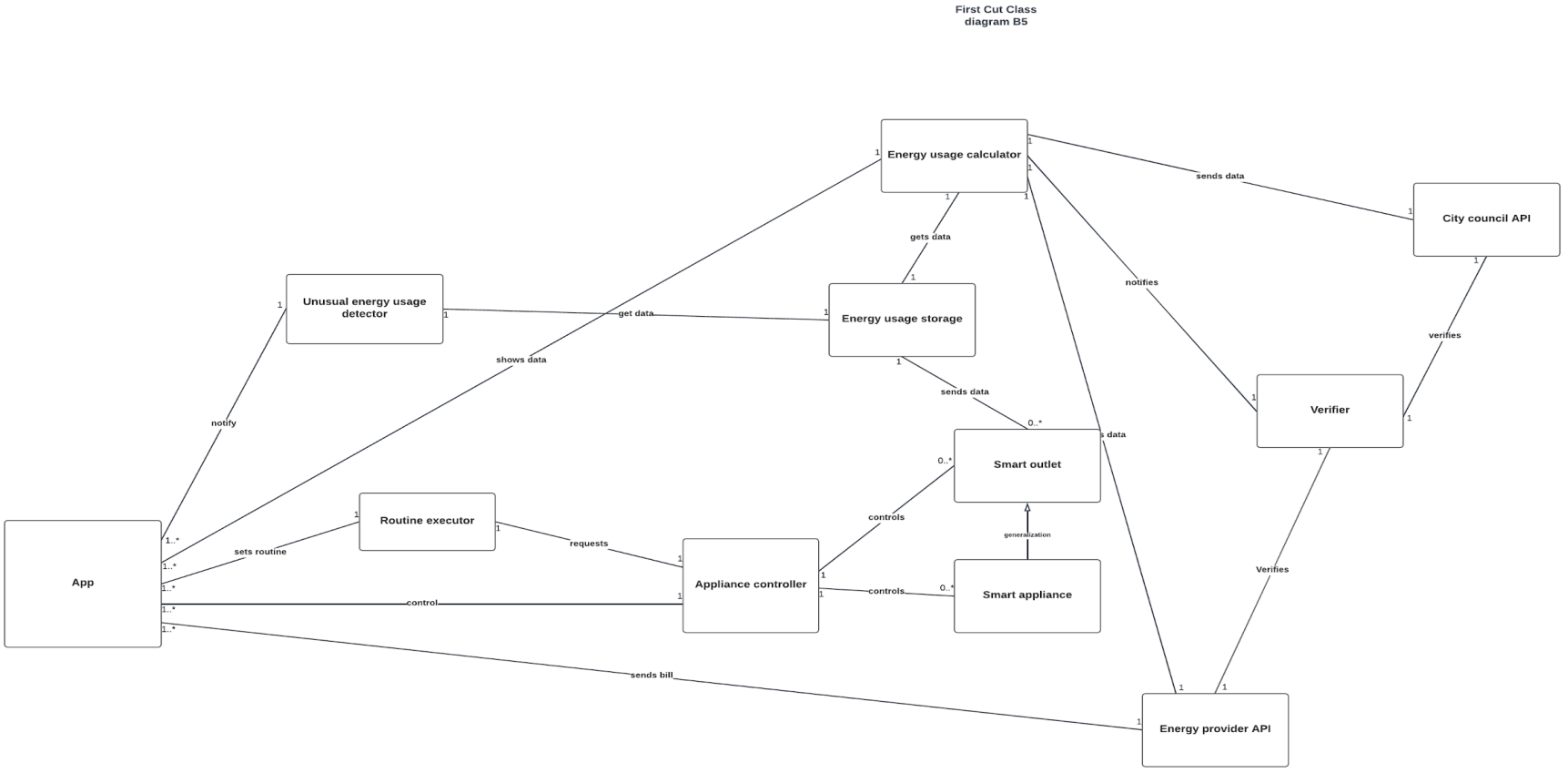
Energy provider API	
Receive data from the smart home system Verify data sent Send bills	Energy-usage calculator Verifier

Verifier	
Verify energy provider or city council	City council API Energy provider API

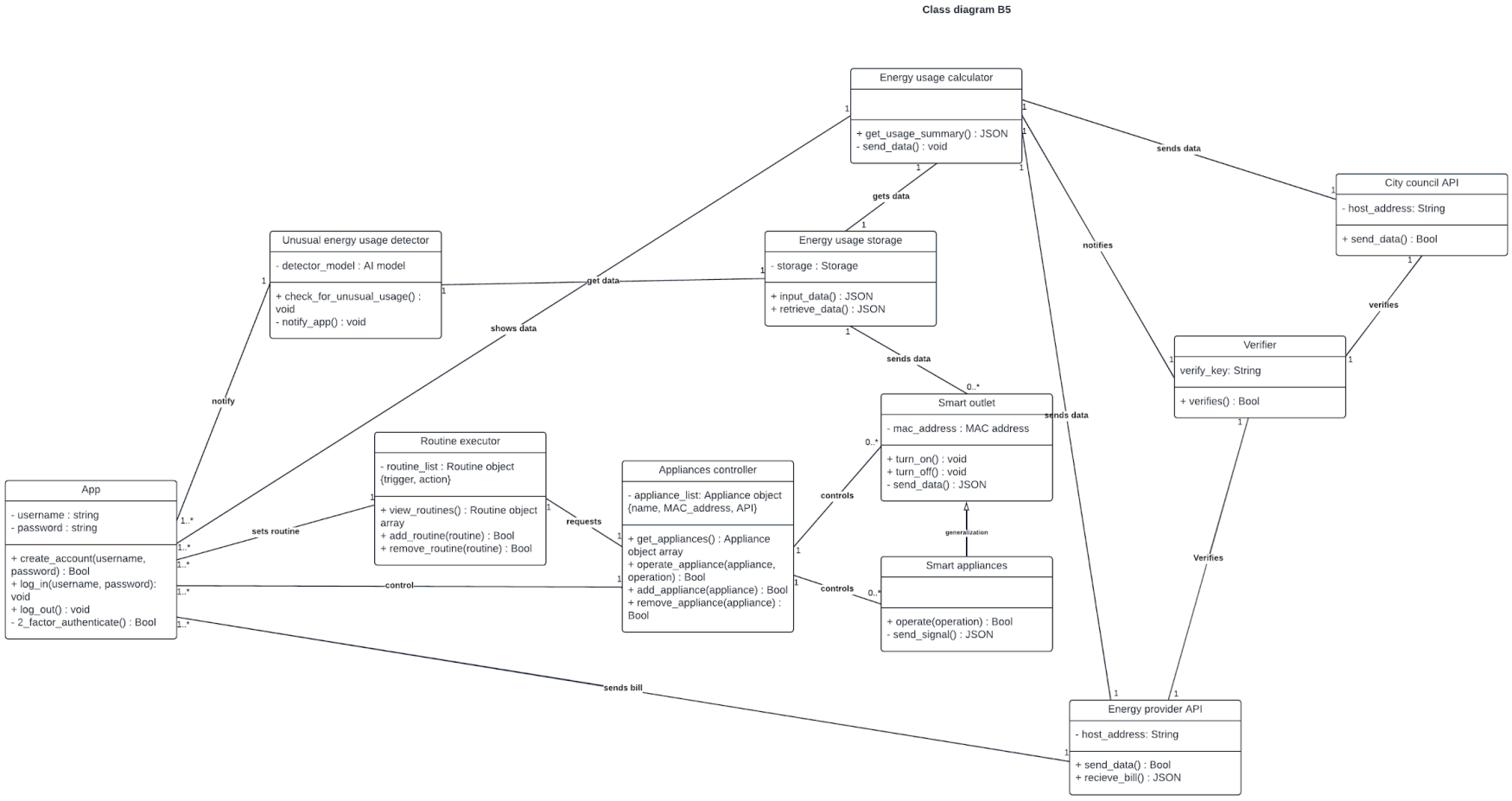
Energy usage calculator	
Get energy usage summary Send data	Verifier City council API Energy provider API Energy usage storage App

Unusual energy usage detector	
Check for unusual energy usage Notify app	App Energy usage storage

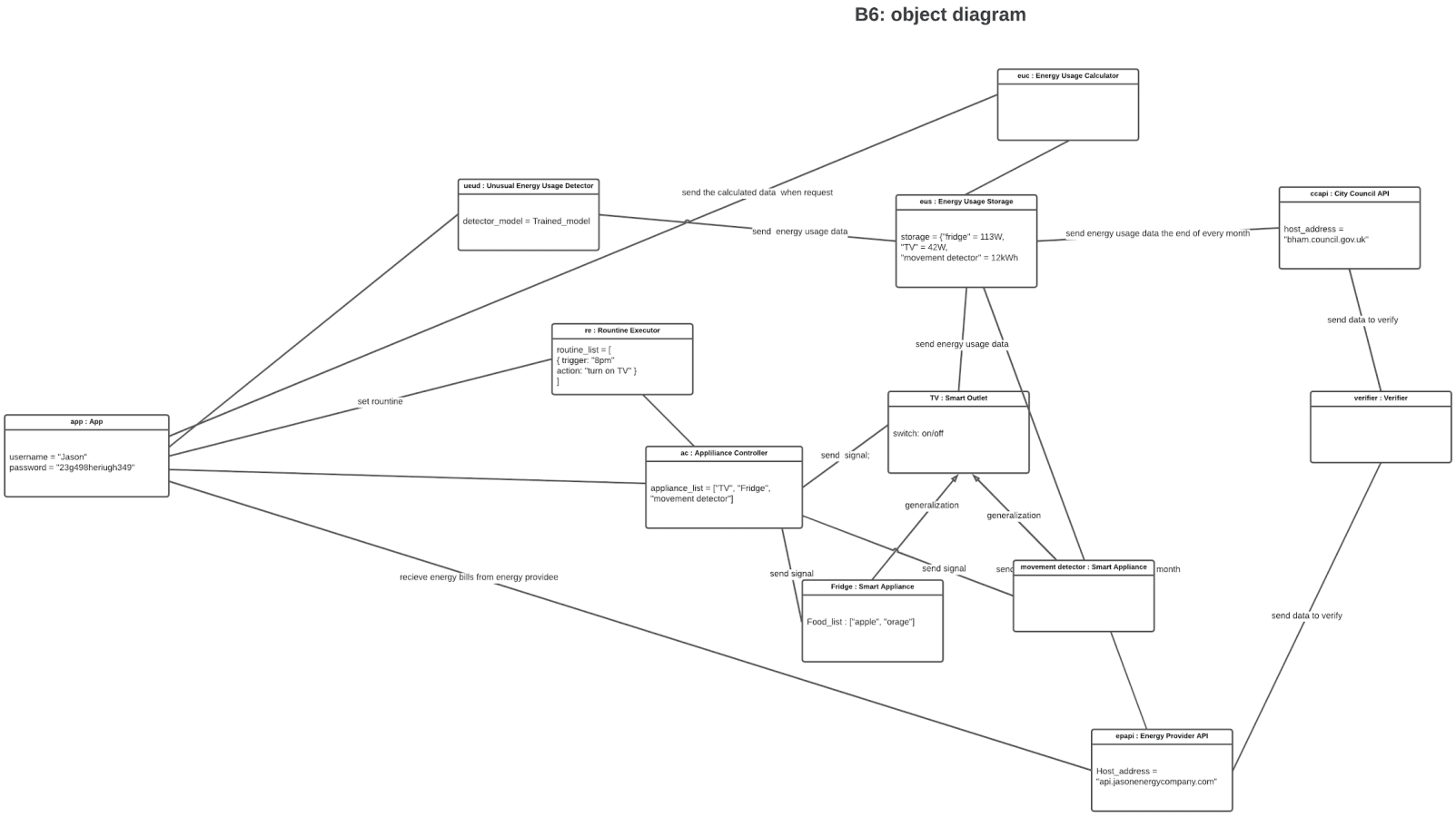
C. First Cut Class Diagram

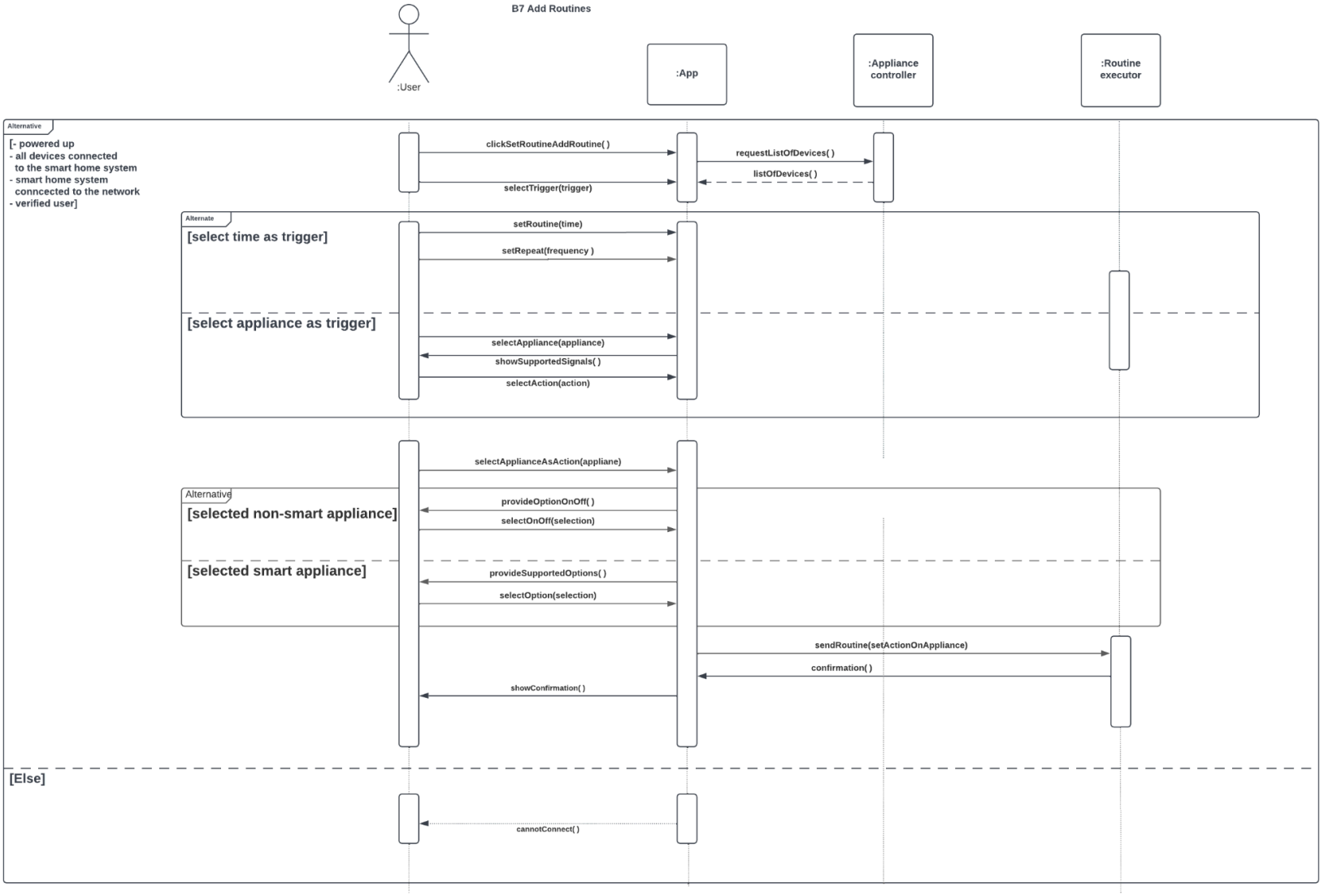


D. Detailed Class Diagram

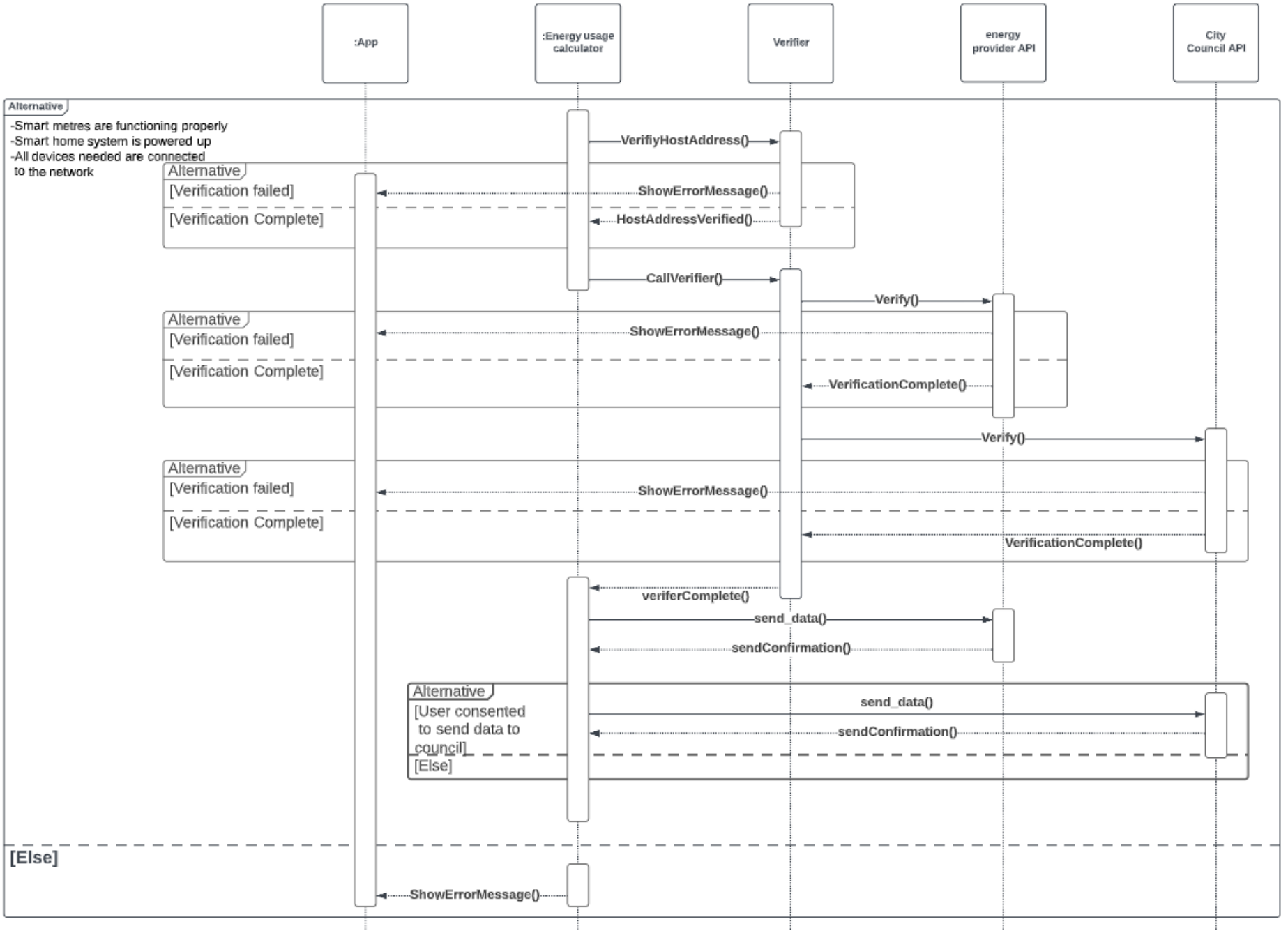


B6.

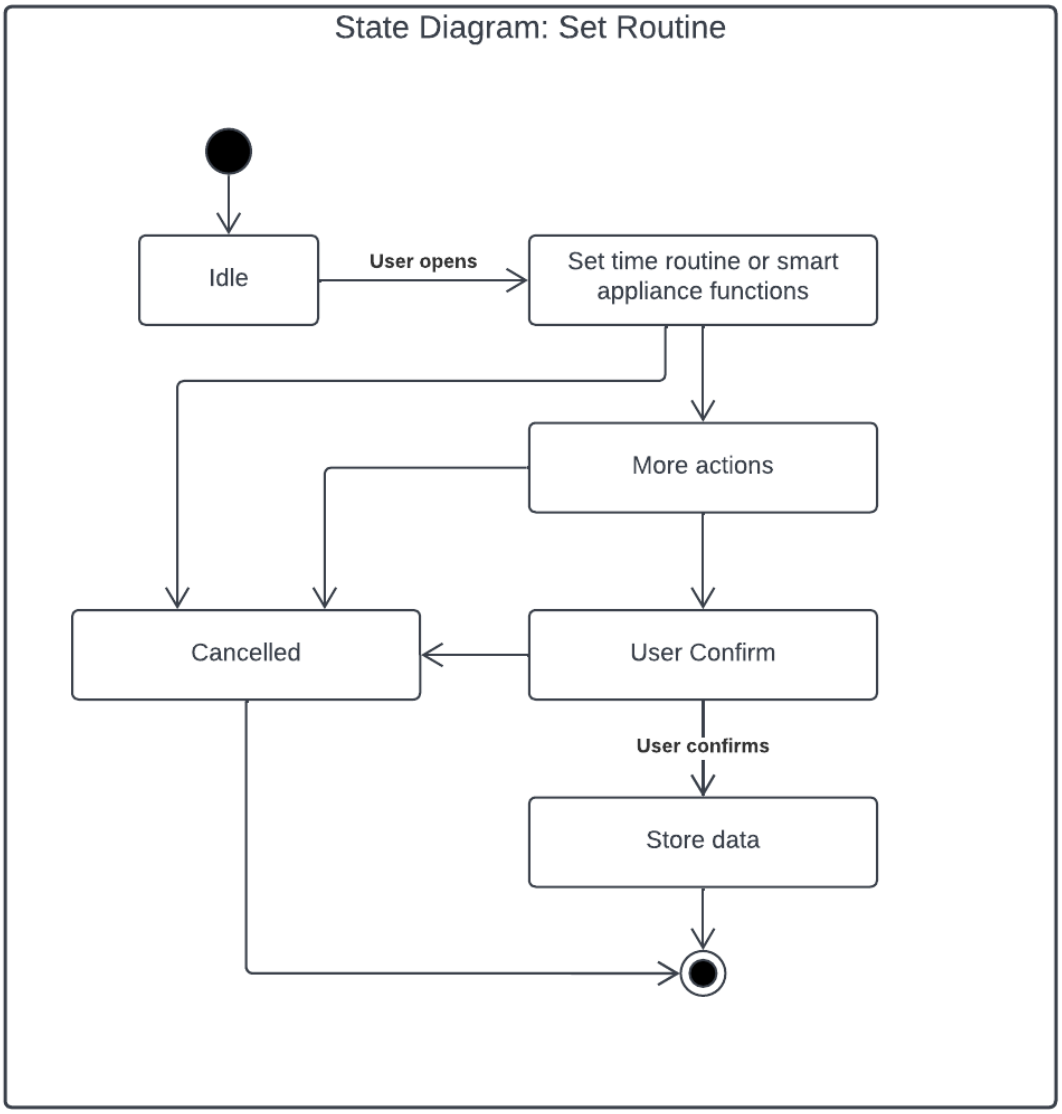

B6: object diagram
euc : Energy Usage Calculator
ueud : Unusual Energy Usage Detector
detector_model = Trained_model
send the calculated data when request
send energy usage data
eus : Energy Usage Storage
storage = {"fridge" = 113W,
"TV" = 42W,
"movement detector" = 12kWh
send energy usage data the end of every month
ccapi : City Council API
host_address =
"bham.council.gov.uk"
send data to verify
re : Rountine Executor
routine_list = [
{ trigger: "8pm"
action: "turn on TV" }
]
set rountine
send energy usage data
TV : Smart Outlet
switch: on/off
verifier : Verifier
app : App
username = "Jason"
password = "23g498heriugh349"
ac : Appliance Controller
appliance_list = ["TV", "Fridge",
"movement detector"]
send signal;
generalization
generalization
send signal
send signal
month
Fridge : Smart Appliance
Food_list : ["apple", "orage"]
movement detector : Smart Appliance
recieve energy bills from energy provideo
send data to verify
epapi : Energy Provider API
Host_address =
"api.jasonenergycompany.com"

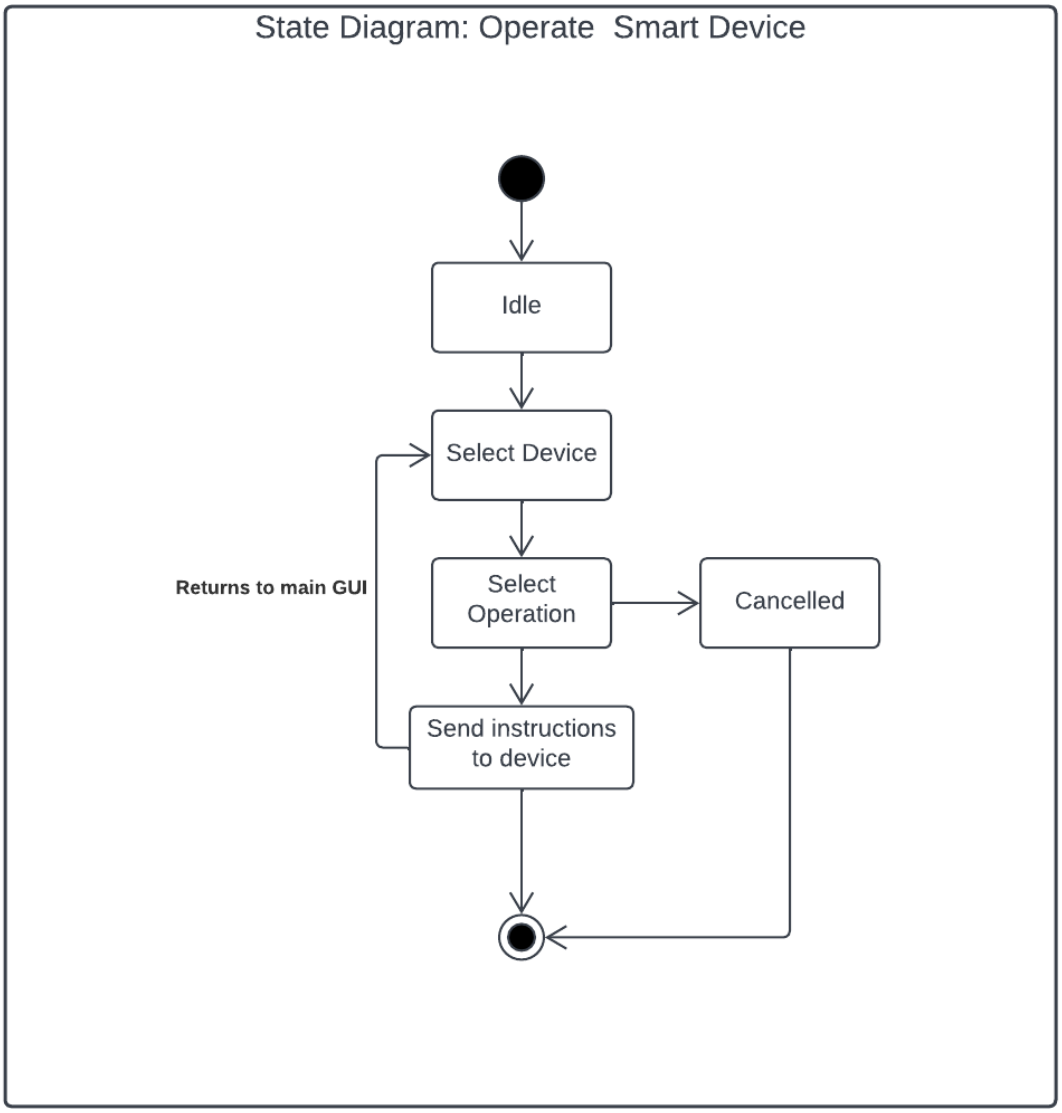

B7 Add Routines
:User
:App
:Appliance controller
:Routine executor
Alternative
[- powered up
- all devices connected to the smart home system
- smart home system conncected to the network
- verified user]
clickSetRoutineAddRoutine( )
requestListOfDevices( )
listOfDevices( )
selectTrigger(trigger)
Alternate
[select time as trigger]
setRoutine(time)
setRepeat(frequency)
[select appliance as trigger]
selectAppliance(appliance)
showSupportedSignals( )
selectAction(action)
selectApplianceAsAction(appliane)
Alternative
[selected non-smart appliance]
provideOptionOnOff( )
selectOnOff(selection)
[selected smart appliance]
provideSupportedOptions( )
selectOption(selection)
sendRoutine(setActionOnAppliance)
confirmation( )
showConfirmation( )
[Else]
cannotConnect( )

B7.


:App
:Energy usage calculator
Verifier
energy provider API
City Council API
Alternative
-Smart metres are functioning properly
-All devices needed are connected to the network
VerifiyHostAddress()
Alternative
[Verification failed]
ShowErrorMessage()
[Verification Complete]
HostAddressVerified()
CallVerifier()
Verify()
Alternative
[Verification failed]
ShowErrorMessage()
[Verification Complete]
VerificationComplete()
Verify()
Alternative
[Verification failed]
ShowErrorMessage()
[Verification Complete]
VerificationComplete()
veriferComplete()
send_data()
sendConfirmation()
Alternative
[User consented to send data to council]
send_data()
sendConfirmation()
[Else]
[Else]
ShowErrorMessage()

B8.

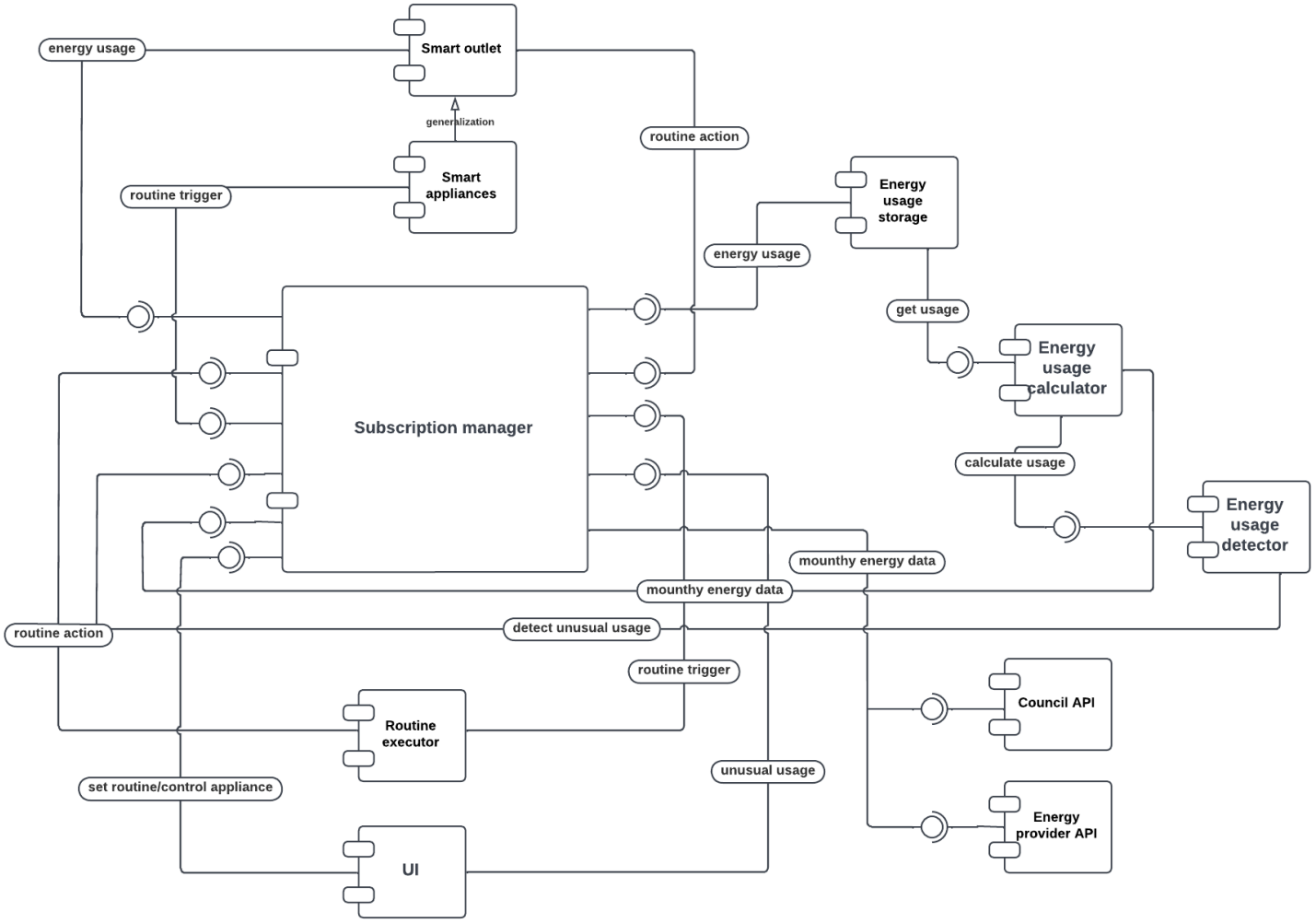

State Diagram: Set Routine
Idle
User opens
Set time routine or smart appliance functions
More actions
Cancelled
User Confirm
User confirms
Store data


State Diagram: Operate Smart Device
Idle
Select Device
Select Operation
Cancelled
Returns to main GUI
Send instructions to device

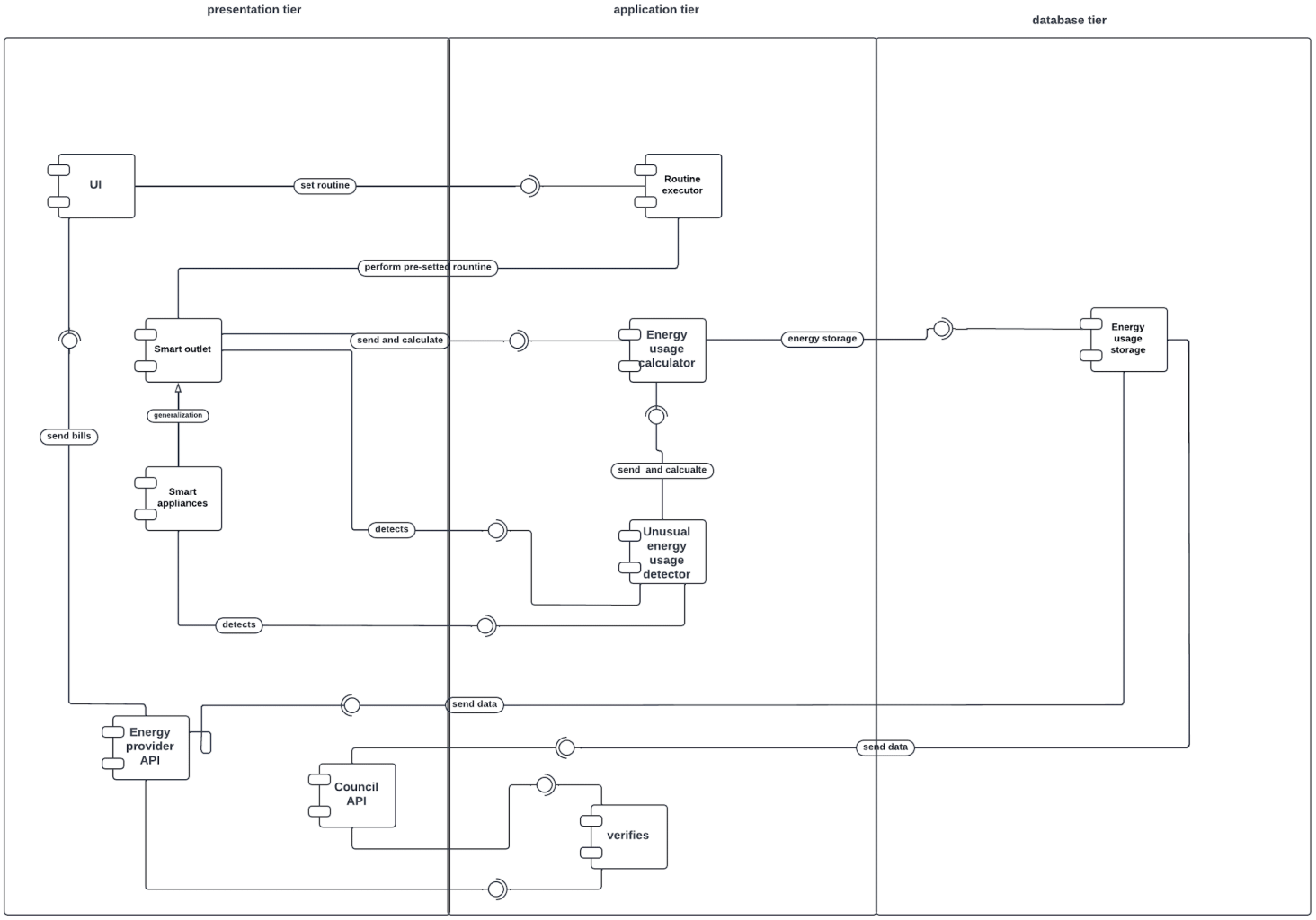
C. Software Architecture Style, Modelling and Evaluation (Unit 3 and Unit 4):

C1.

Publisher-Subscriber architecture style

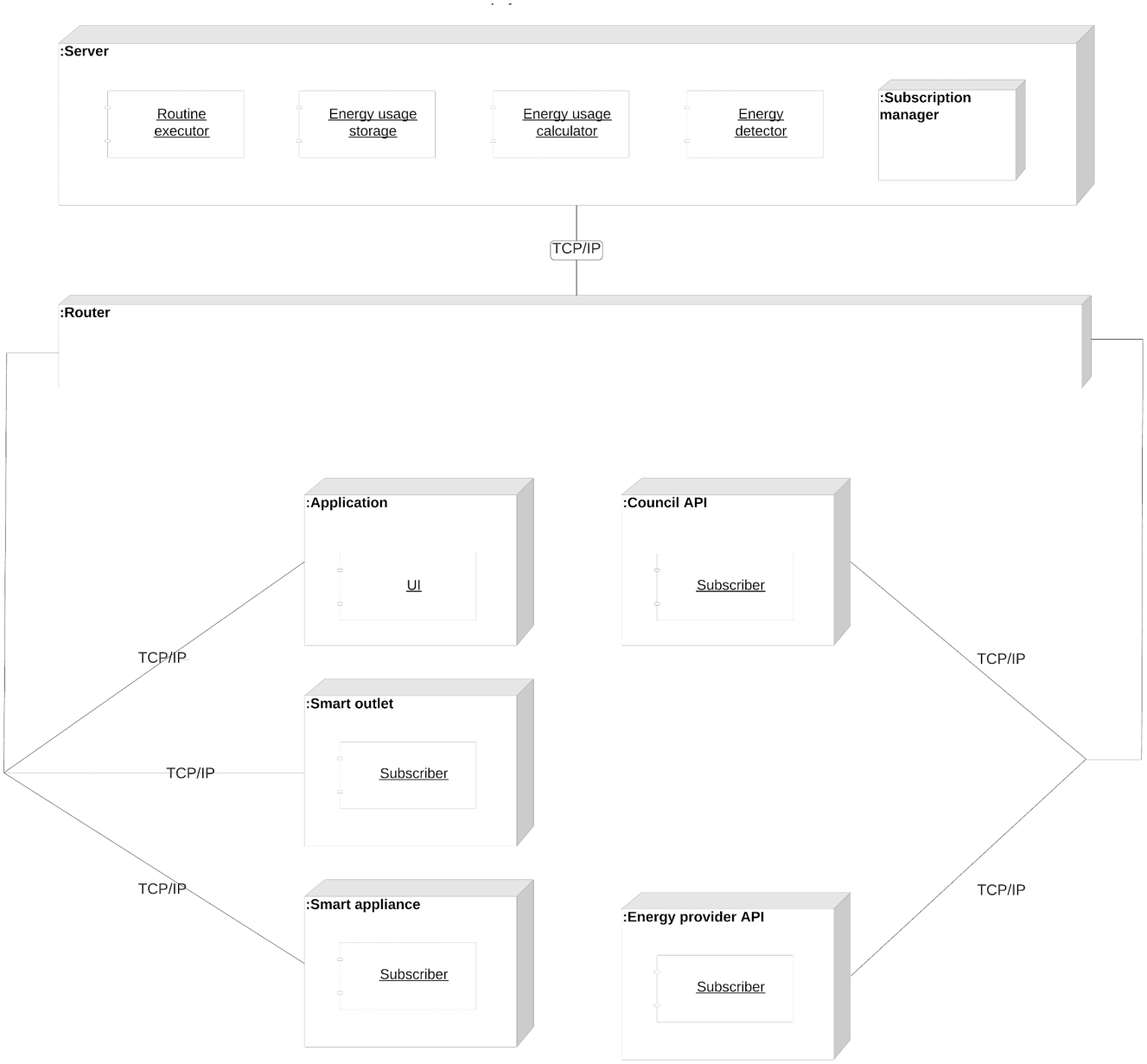


3-tier architecture style

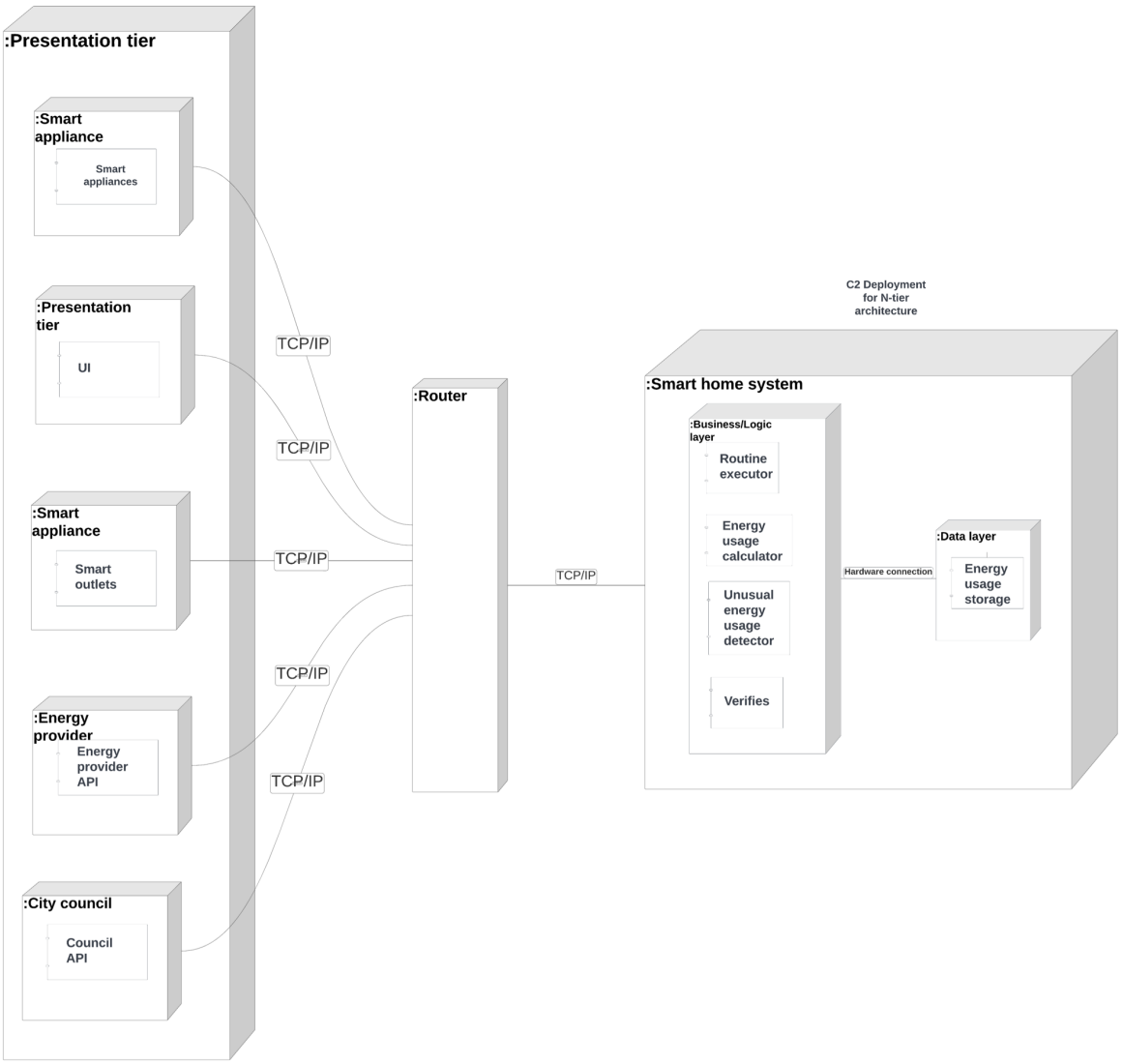


C2.

Publisher-Subscriber architecture style



3-tier architecture style



C3.

The advantage of the publisher-subscriber model is loose coupling. The appliance controller doesn't need to know the API of each appliance, it just needs to publish the message to the message broker, and the message broker will take care of that.

However, the advantage of having real-time events isn't applicable. Also, for use cases like sending energy usage and showing energy usage. We will be fetching data periodically and the advantage of flexibility isn't applicable too, as the user is most likely going to have the same set of appliances most of the time, there is no need to dynamically change the list of appliances.

The trade-off is that it creates unnecessary complexity for our system. For a system for one household, we expect up to 100 appliances. Therefore the appliance controller is able to support storing the API of each appliance and having a message broker just complicates things. Also, the publisher has no knowledge of the status of the subscriber since things are loosely coupled, if the appliance controller tells an appliance to do something, it doesn't know if the appliance did it, since it is merely a subscriber of the event published.

The advantage of the N-tier model is the ease of development compared to the publisher-subscriber model. We do not need to set up any extra components such as an event broker. Also, each layer can be developed by separated teams as each layer is separated and only communicates through APIs.

Another advantage is security. Since the presentation tier must go through the application tier to get to the data tier, components outside our system cannot bypass the application logic and access the data. In our case, it is the energy usage data, so it offers better security than the publisher-subscriber model, where often publishers don't know who the subscribers are.

The drawback of the N-tier model is increased latency. Unlike the publisher-subscriber model, where everything simply goes through the event broker. In the N-tier model, components inside the presentation tier have to go through the application tier, which will increase the latency of response. However, since our system isn't time-critical, it is fine.

Therefore, from the small scale of our system, and how execution speed isn't a big concern. We believe the advantage of the ease of development and better security justifies us choosing the 3-tier model over the publisher-subscriber model.

D. Software Testing

Test Plan

Introduction

The plan specifies the things to be tested, the features to be tested, the types of testing to be performed to complete testing, and the risks connected with the plan. As needed to ensure that our system meets the quality specifications defined by the client and discover any bugs that may arise.

Test Items

Features To Be Tested

Features to be tested include the following:

Functional Requirements

1. Operation TC-10
2. Store energy usage of appliances TC-20
3. Show the user the energy usage TC-30
4. Operate smart appliances TC-40
5. Turn on/off appliances TC-50
6. Warn about unusual energy usage TC-60
7. Routines TC-70
8. Send energy usage to the energy provider TC-80

Non-functional Requirements

1. Availability TC-90
2. Efficiency performance TC-100
3. Scalability TC-110

- Add the test case ID

Features Not To Be Tested

- Usability
- Efficiency space
- Compatibility

The features above will not be tested as the features is not critical to the functioning of the software. For example, a feature that provides a minor convenience for the user, but does not affect the core functionality of the software, may not be worth the time and effort required to test it. The feature is too difficult to test. If testing a feature would require a significant amount of time, resources, or expertise, it may not be feasible to test it. For example, if there are any security issues, it would be fixed through bug fixes. Besides that, compatibility should not be thoroughly tested as it will already be tested throughout the development process.

Approach

Each test in the test plan has been categorised into either a black or white testing method. We will have a separate quality assurance team that will deal with each respective type of testing

White box testing, also known as clear box testing or glass box testing, is a technique in which the tester has detailed knowledge of the internal workings and structure of the software being tested. This allows the tester to design test cases that exercise specific paths through the code and ensure that all aspects of the software are functioning as intended.

Black box testing, on the other hand, is a technique in which the tester has no knowledge of the internal workings of the software. Instead, the tester focuses on the functionality of the software from the perspective of the end user, and designs test cases based on the requirements and specifications of the software. Black box testing is often used to validate the functionality of the software from the user's perspective, and to ensure that it behaves as expected.

Both white box and black box testing have their own advantages and disadvantages. White box testing allows for more thorough and detailed testing of the software, but it can be time-consuming and may not always provide a realistic representation of how the software will be used in the real world. Black box testing, on the other hand, is typically faster and more closely reflects the user experience, but it may not provide as comprehensive a view of the software's internal workings. As a result, many testing efforts use a combination of both approaches to provide a well-rounded view of the software's functionality.

Pass Fail Criteria

Functional requirements

Test Case ID	Test Description	White/Black Box	Test Steps	Test Data	Expected Result	Actual result	Pass / Fail	Test Comments
TC-10	Verify smart home system can register a smart outlet	Black	Go to “Devices” and then “Register Device” and then select “Smart outlet”	Smart outlet MAC address: 00-B0-D0-63-C2-26	Smart home system should be able to register the smart outlet			
TC-11	Verify smart home system can unregister a smart outlet	Black	Go to “Devices” and then select a smart home outlet and then select “Unregister Device”	Smart outlet MAC address: 00-B0-D0-63-C2-26	Smart home system should be able to unregister the smart outlet			
TC-12	Verify smart home system can register a smart appliances	Black	Go to “Devices” and then “Register Device” and then select “Smart appliance” and select a smart appliance from a list	Smart appliance: Smart light	Smart home system should be able to connect to the smart appliance			
TC-13	Verify smart home system can unregister a smart appliance	Black	Go to “Devices” and then select a smart appliance and then “Unregister device”	Smart appliance: Smart light	Smart home system should be able to disconnect from the smart appliance			
TC-20	Verify the smart home system can	White	- While a smart outlet/appliance is	Smart outlet: Test smart	Smart outlet usage data			

	store the energy usage data from the smart outlets/ appliance		connected, connect an electrical device and turn it on. - Go to “Usage Data” and find the corresponding smart outlet/appliance data	outlet	should be displayed on			
TC-21	Verify the energy usage data can be stored for a year	White	- There must be existing smart outlet data in the usage data - Go to “Settings” and change the date to the next year	Date: 1/1/2022 -> 1/1/2023	Previous existing smart outlet data should not be visible			
TC-30	Verify the smart home system can show the energy usage data from the smart outlets	Black	Go to “Usage Data” and find data from the smart outlets	Usage data: Test smart outlet	Test smart outlet usage data should be displayed on the screen graphically and numerically			
TC-40	Verify that the smart home system is able to operate smart appliances	Black	Go to “operate smart appliance” and select the smart appliance to take control of	Usage data: Appliances that are connected	The smart appliance to perform the operations requested.			
TC-50	Verify that the	Black	Go to “operate	Usage data:	The smart			

	smart home system is able to turn on or off appliances		smart appliance" and select the application to turn on or off	Appliances that are connected	appliance to perform the operations requested.			
TC-60	Verify that the smart home system is able to warn the user about unusual energy usage	Black	Install a high energy-consuming device in the home after running the system for a while.	Usage data: outlet that a high consuming device is connected to	The application should notify the user			
TC-70	Verify the smart home system is able to add a routine to the storage	Black	Go to "Routine" select "Add routine" and add a test routine	Routine: Test routine	Test routine should be added to storage of the smart home system and displayed on "Routine" menu			
TC-71	Verify the smart home system is able to edit a routine	Black	Go to "Routine", select a routine and select "Edit"	Routine: Test routine	Test routine should be edited and updated in the storage of the smart home system and displayed on "Routine" menu			
TC-72	Verify the smart	Black	Go to "Routine",	Routine: Test	Test routine			

	home system is able to delete a routine		select a routine and select “Delete”	routine	should be deleted and removed from the storage of the smart home system and removed on “Routine” menu			
TC-80	Verify the smart home system is able to send energy bills to the energy provider	Black	The smart home system should do this automatically	The data sent to the energy provider	The latest bills showing the total usage should be updated on your account with the energy provider and the total usage should match the recorded usage on the system			
TC-81	Verify the smart home system is able to display the latest bills from the energy provider	Black	Go to “Billing” and select the latest bill to view	The bills stored	The latest bill should show the total usage energy usage that was sent to the energy provider.			

Non-functional requirements

Test Case ID	Test Description	White/Black box	Test Steps	Test Data	Expected Result	Actual result	Pass / Fail	Test Comments
TC-90	Ensure that the smart home system can be accessed at all times	Black	Access the smart home system and perform different tasks at different times. This test should be multiple times a day for a month.	The app is able to run	The smart home system should be able to run at all times without any trouble			
TC-91	Verify the smart home system can operate 24/7	Black	Add a routine every hour for 24 hours	App notification every 1 hour interval from 00:00 to 23:00	The smart home system should send a notification every hour for 24 hours			
TC-100	Verify the smart home system can't be forced to manipulate the database table in unintended ways	White	Perform a SQL injection attack on the database tables	Database table: Smart outlet	The database should reject or scrub the bad inputs from the user			
TC-101	Verify the smart home system is not able to be controlled by unauthorised users	White	Search for vulnerabilities in the smart home system	N/A: Vulnerabilities are general	The smart home system should have no vulnerabilities			

TC-110	Verify the smart home app is compatible on different mobile operating systems	White	Download the app on different phones	Phone operating systems: iOS, Android	All features should work as normal on both iOS and Android phones			

Exit Criteria

The entry criteria refer to the desired conditions for beginning test execution. When all of the activities mentioned to be tested are 100% performed, the test can begin its execution phase.

Exit criteria are the conditions or requirements that must be met before a software testing effort can be deemed complete. These criteria typically include the testing of all specified functional and non-functional requirements, the identification and resolution of all defects, and the verification that the software meets all specified acceptance criteria. The exact exit criteria for a given software testing effort will depend on the specific goals and requirements of the project.

Exit Criteria

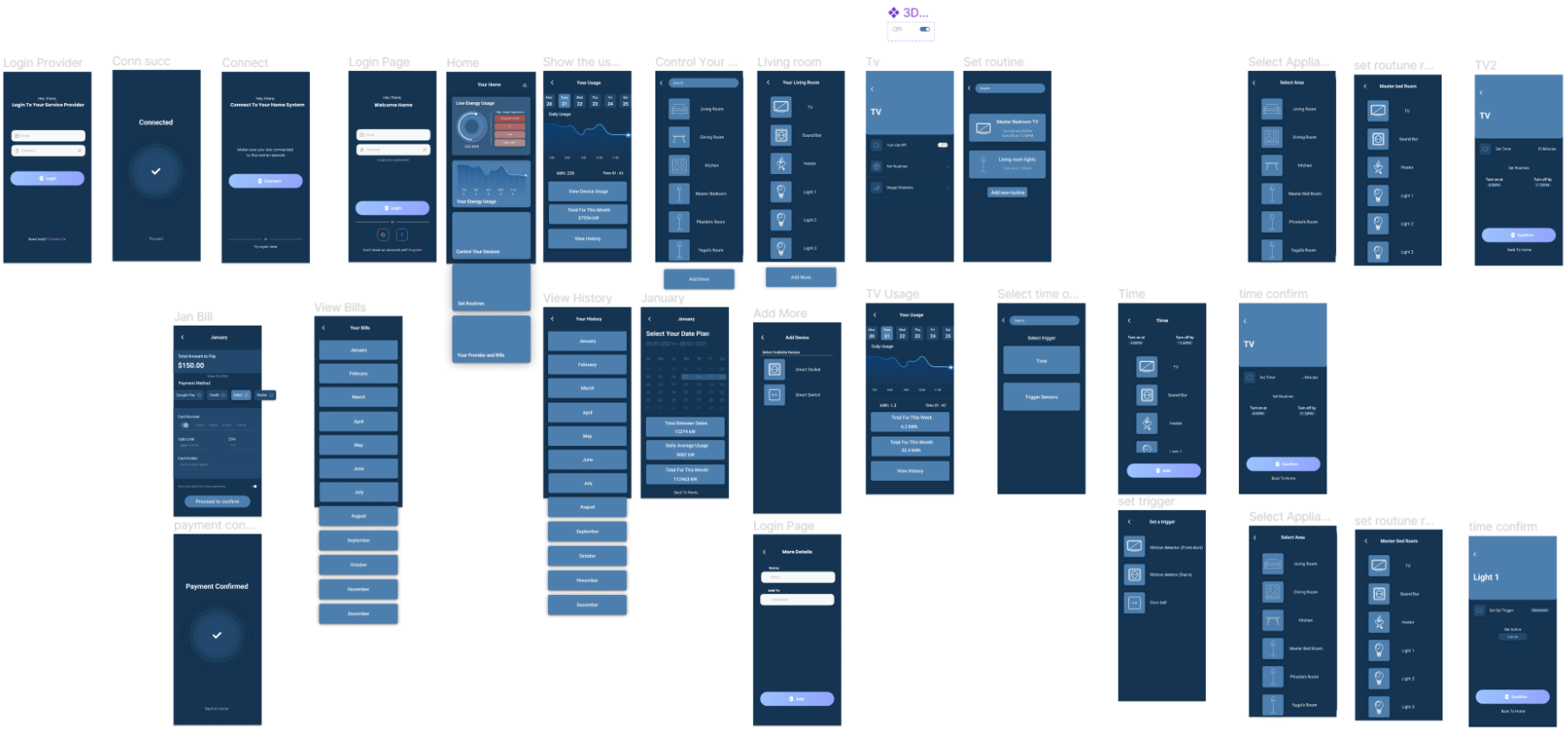
- All tests have been executed
- 95% pass rate of the tests
- No critical defects
- Defects found must be documented for future releases

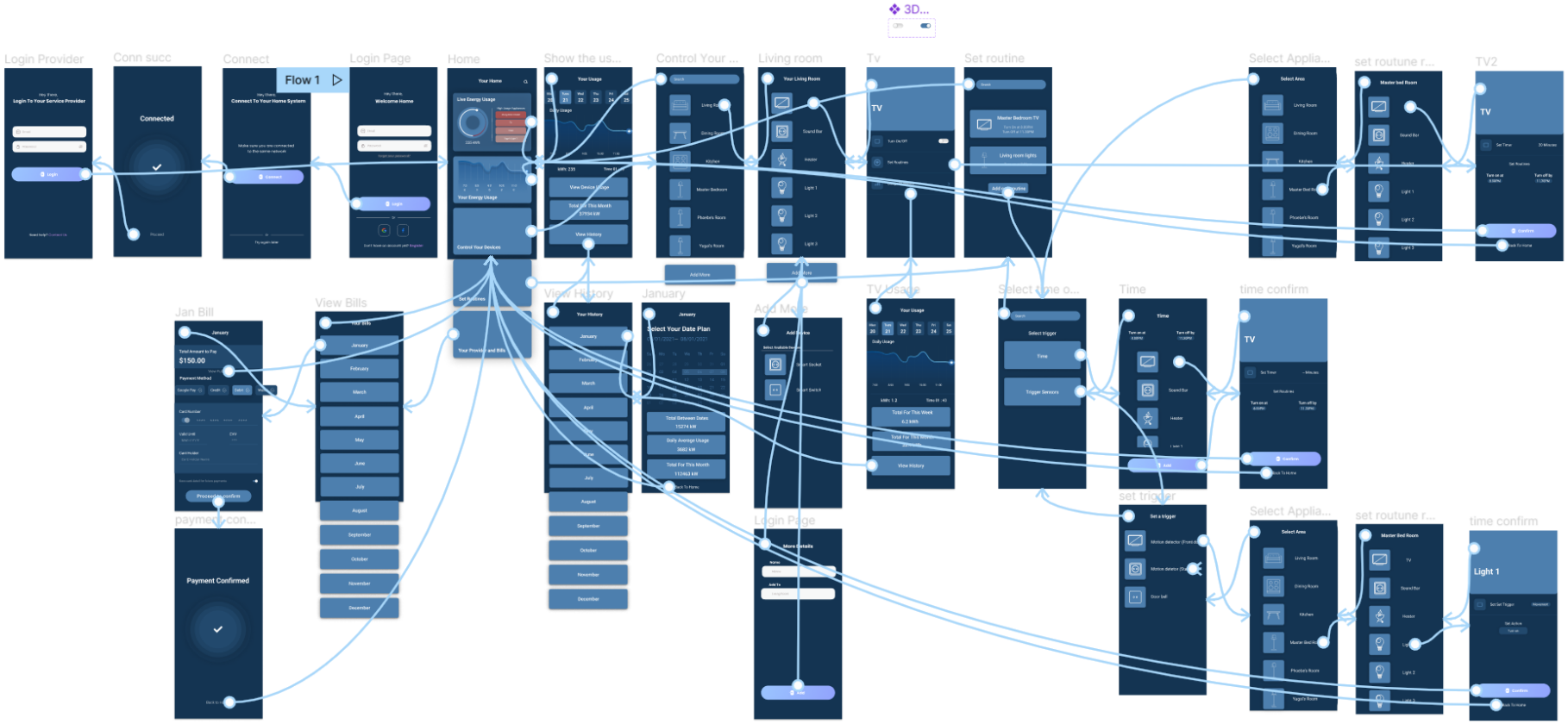
Assumption

E. Usability and Prototyping

E1.

The interactive prototype can be viewed with the following link
https://www.figma.com/file/cj6T5ELwwc2eqx1bOd2uqQ/Smart-Home-system?node-id=0%3A1&t=HmPzlc5t60x2dWvI-1
We will also screenshot the prototype, the screenshot is all the screens. The second screenshot includes the prototype.


3D...
Login Provider
Conn succ
Connect
Login Page
Home
Show the us...
Control Your...
Living room
Tv
Set routine
Select Applia...
set routune r...
TV2
Jan Bill
View Bills
View History
January
Add More
TV Usage
Select time o...
Time
time confirm
payment con...
Login Page
set trigger
Select Applia...
set routune r...
time confirm


Login Provider
Conn succ
Connect
Flow 1
Login Page
Home
Show the us...
Control Your ...
Living room
Tv
Set routine
Select Applia...
set routune r...
TV2
Jan Bill
View Bills
View History
January
Add More
TV usage
Select time o...
Time
time confirm
payment con...
Login Page
set trigger
Select Applia...
set routune r...
time confirm

E2.

Please find the corresponding video in the mp4 attachment

F. Ethics and Professional Practice – (Unit 8) – self-reading:

The smart home system followed the ACM code of ethics and professional conduct. The study of ethics involves analysing human attributes and social structures. The purpose of ethics is to help us distinguish between right and wrong.

In Section 1, general ethical principles. Our smart home system contributes to society by providing users with their own energy usage, making them more aware of them and encouraging users to use less energy (Code 1.1 Contribute to society and to human well-being). Our system provides users with transparency and a clear understanding of how much and when they consume energy. We do not give false energy usage to cheat our users into thinking any appliances are more energy efficient than they really are, or to falsely increase the energy usage to increase the energy bill for the energy provider. (Code 1.3 Be honest and trustworthy). The smart home system benefits the environment because utilities (i.e. energy providers) are able to regulate peak times, reduce congestion, detect overloads, and encourage more efficient use of power sources based on the data provided by the smart home system(Code 1.1 Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing.). Additionally, the integration of renewable energy sources into the Smart Home System by providing relevant data helps maintain the environment in some way. Furthermore, privacy should be considered. Our smart home system allows users to choose whether or not they consent to send energy usage data to the city council, and if they do not consent, the system will not send it. Also, the smart home system also doesn't send energy usage data to any third party (GDPR regulations and Code 1.6 Respect Privacy). It is true that cloud-based systems, such as those developed by Google and Amazon, provide ease of use, but they often come at the expense of privacy, as the company's servers receive an unrelenting stream of personal information. A locally-hosted alternative often requires greater effort to set up but offers increased security as the data remains within your personal network (Code 1.7 Honor Confidentiality).

In section 2, professional responsibilities. Our system allows our customers to request consent from their customers(Code 2.5 Give comprehensive and thorough evaluations). According to the state of consent, we let them configure how the system works. Regular reviews were conducted throughout this project to minimise issues/defects in each phase, the work is distributed to different groupmates, but everyone has a chance to review and provide feedback on it. This minimises the chance of someone making a mistake, as it would likely be spotted by another groupmate. (Code 2.4 Accept and provide professional review). The multi-factor authentication capability in our user verification ensures that the system is robust and secure and that it is unlikely for someone not authorised to have access to the system. Also, before our system sends the energy usage data to the energy provider or city council, it verifies them, as well as verifies itself, the smart home system to them. Therefore malicious third-party actors wouldn't be able to pretend to be the energy provider, city council, or our system. (Code 2.9 robustly and usably secure)

In Section 3, professional leadership principles. We recognise and care for our system, which will become embedded in society's infrastructure. We hope as more and more people adapt to this smart home system, we can form a smart city that uses the data generated to benefit citizens (Code 3.2 evaluates the fulfilment of social responsibilities). Within our group, we ensured that others felt comfortable speaking their minds and facilitated the dialogue so that other colleagues could reach a strategic conclusion, so each of us can learn from the work. (Code 3.5 Create opportunities for members).

In Section 4, Compliance With The Code, We ensured that the above vision was properly translated and that actionable objective were carried out in an ethical manner by reference to the code of conduct often. (Code 4.1 Uphold, promote, and respect the principles of the Code.).

Project Management and Moderation

We met the deadline by having meetings about once a week and distributing the remaining task to groupmates that is available.

Working progress tracker		
part of the project	**allocation**	**edited (not allocated to you, but you did things)**
A1	Chit	
A2	Chit	
B1	Abasin, Chit, Daniel, Jason, Pratham, Praveen, Yeni, Yugal	
B2	Abasin, Chit, Daniel, Yeni, Yugal	Yugal, Praveen
B3	Daniel, Jason, Praveen, Yeni	Chit
B4	Chit, Daniel	Yugal
B5a	Pratham	Daniel, Chit, Yugal
B5b	Pratham	Daniel, Chit
B5c	Pratham	Chit
B5d	Pratham	Chit
B6	Jason	Chit
B7	Abasin, Yugal	Chit, Yugal
B8	Praveen	
C1	Chit, Jason	
C2	Yeni, Yugal	Chit
C3	Abasin, Chit	
D	Abasin, Daniel, Praveen	
E1	Praveen	Chit, Yugal
E2	Film: Abasin, Chit, Daniel, Jason, Praveen, Yeni, Yugal Edit: Praveen	
F	Pratham	Chit, Jason

Meeting records	
date	**summary**
26-Oct	1. Read the coursework document and decide on which topic to proceed with the project.
2-Nov	1. Discuss and decide by which part to finish at the next meeting. 1-1. Each person was assigned a part to finish by taking charge of A1 to B3. 2. Creates shared documents necessary to proceed the task together.
10-Nov	1. Discuss the progress of the parts that have been allocated so far, their respective reviews, and points to be revised. 2. Determined who the remaining B parts (B4-B8) will be allocated to.
17-Nov	1. Discuss the progress of the parts that have been allocated so far, their respective reviews, and points to be revised.
20-Nov (online)	1. Discuss the progress of the parts that have been allocated so far, their respective reviews, and points to be revised. 2. Determine who will work on the UML diagram and deployment diagram of part c and who will work on the testing case in d part.
23-Nov	1. Discuss the progress of the parts that have been allocated so far, their respective reviews, and points to be revised.
27-Nov (online)	1. Discuss the progress of the parts that have been allocated so far, their respective reviews, and points to be revised. 2. As slow progress is concerned, change of the format and duration of the meeting for faster progress is decided.
28-Nov	1. Working together as a team to speed up progress
30-Nov	1. Review progress, assign groupmates to remaining tasks, talked about video recording
3-Dec (online)	1. Reviewed the interactive prototype of the system made so far and discussed improvements. 2. Delievered general progress of the whole coursework; reviewed work by parts and decided when to do peer review.
4-Dec	1. Worked on recording video presentation. 2. Improved the UI of the app
6-Dec	1. Discussed what needed to be done before submitting

Feedback session records	
date	**Simplified**
27-Oct	Asked about the topic
3-Nov	Asked about the topic, and what assumptions we should make. How do we start working.
9-Nov	Asked about use case diagram, what are the actors. about our project idea.
17-Nov	Reviewed the state diagram and use case scenario
18-Nov	Asked about actors in the use case diagram. reviewed the activity diagram, asked about software architectures
23-Nov	Talked about the class diagram, testing, component diagram
29-Nov	Talked about deployment diagram, sequence diagram, use case
1-Dec	Reviewed the component diagram, deployment diagram, asked about testing
6-Dec	Reviewed ethics and deployment diagram

Extra work done	
Person	**Work**
Chit	- held every meeting, set up time, set up topics, facilitate discussions - attend all but 2 feedback sessions - distributed work between teammates - contacted teammates about the work done - monitored the progress - update progress on discord group - proofread entire document
Abasin	- Attended every group meeting - attended some feedback sessions proofread the document, made spell and grammar checks. - held meeting -created original tracker
Yugal	- Attended every group meeting, attended some feedback sessions, proofread the document, made spelling and grammar checks, helped with recording the video for E2, helped make the UI for the app
Daniel	- Attended all group meetings, attended some feedback sessions, directed and recorded the presentation video, helped tidy up some other members work, fixed grammar, helped with making the video for E2, helped with making the UI of the app
Jason	-Attended all group meetings -review parts that are done by other groupmates
Pratham	
Praveen	- Attended all group meetings - Attended some feedback sessions -Helped the other group members for some parts as I have prior experience in this subject -review the document from time to time
Yeni	- Attended all group meetings - Attended some feedback sessions - Proofread some part of the project - Record the meeting, feedback session and keep on track of work allocation