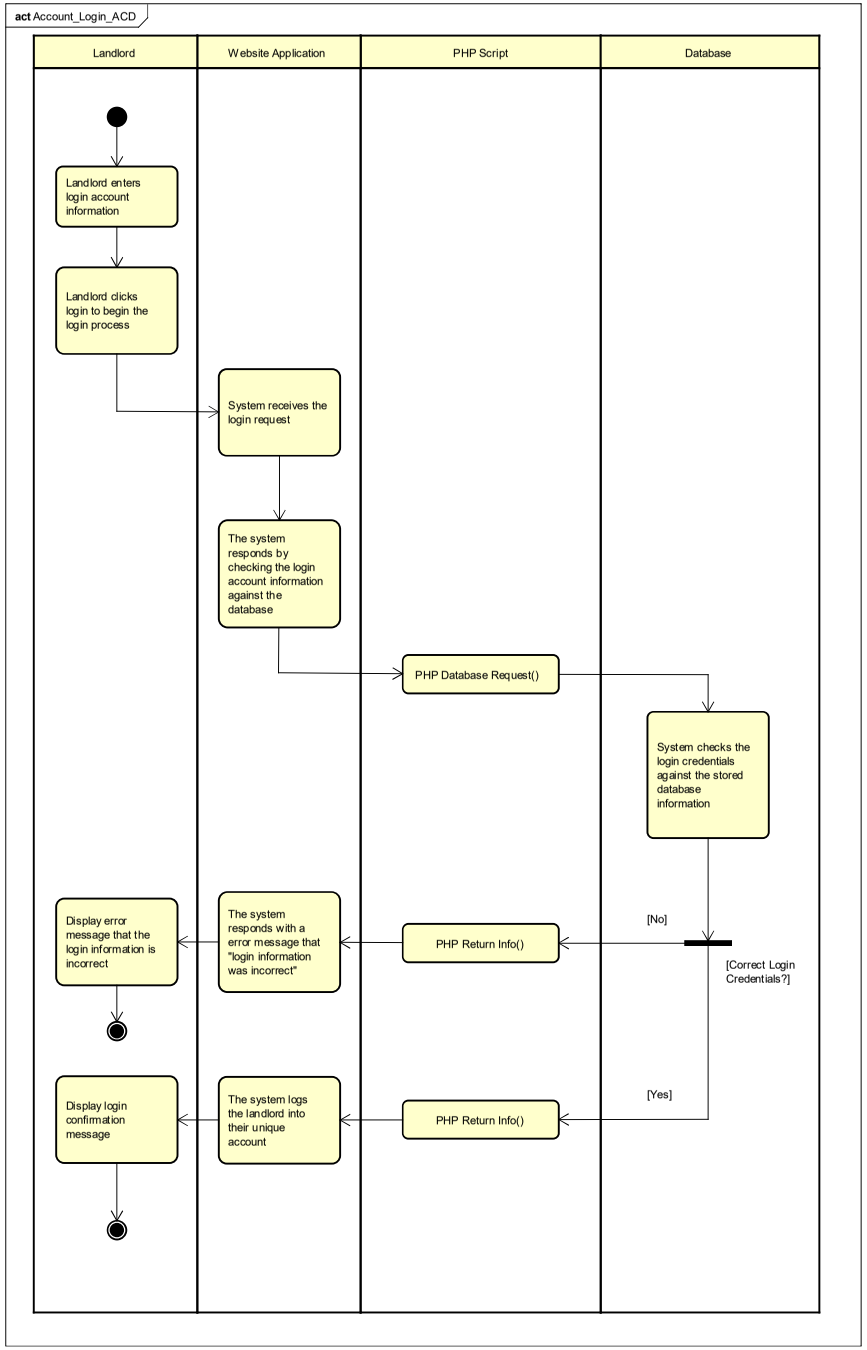
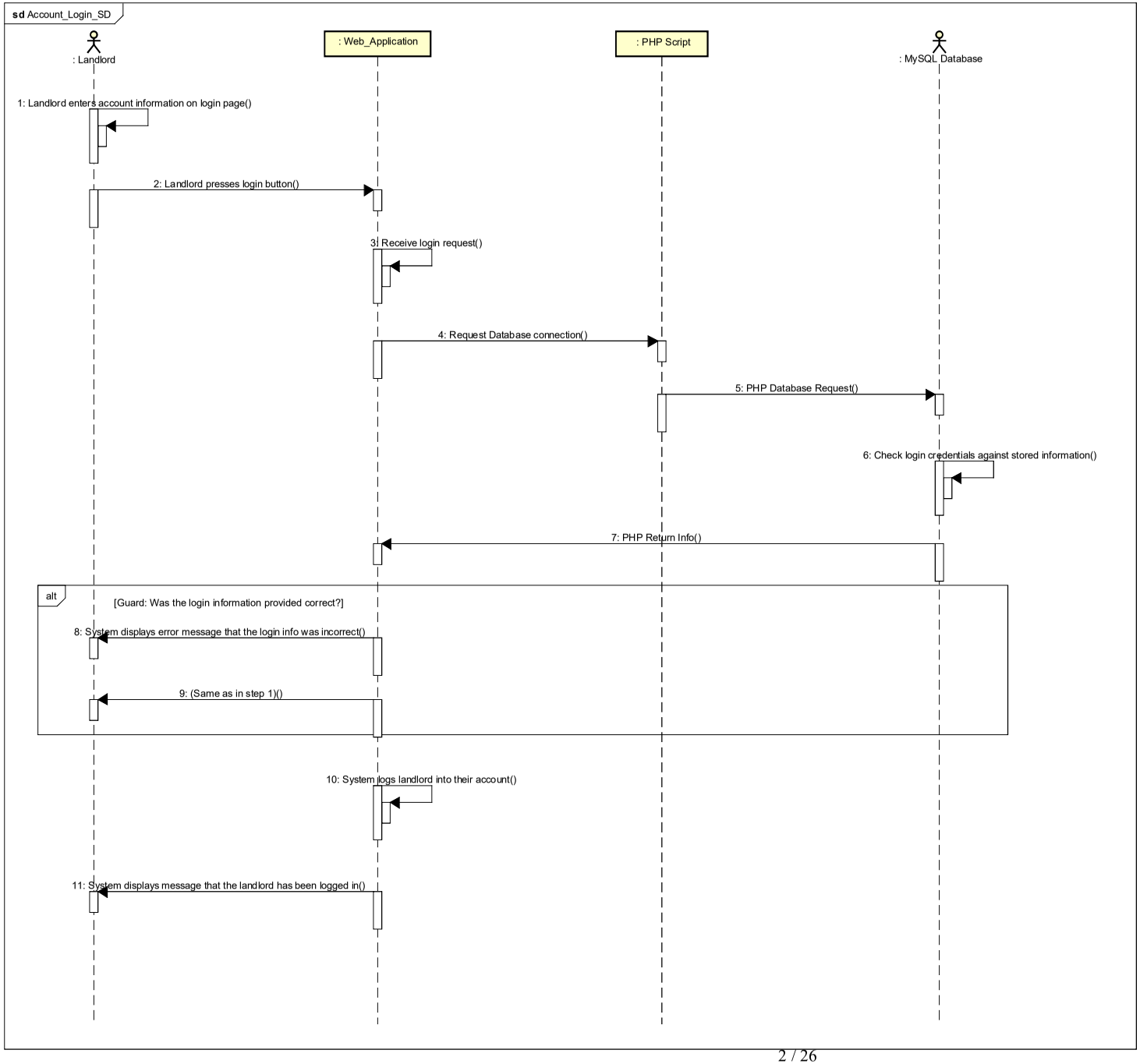
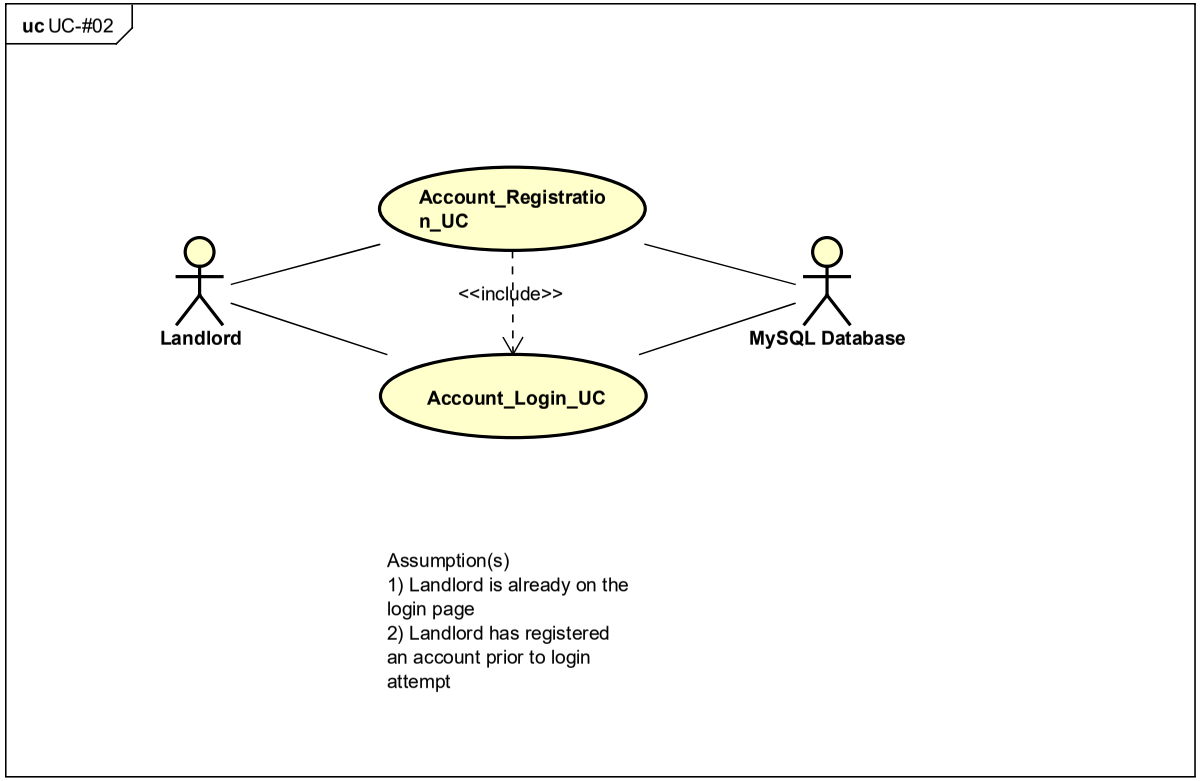
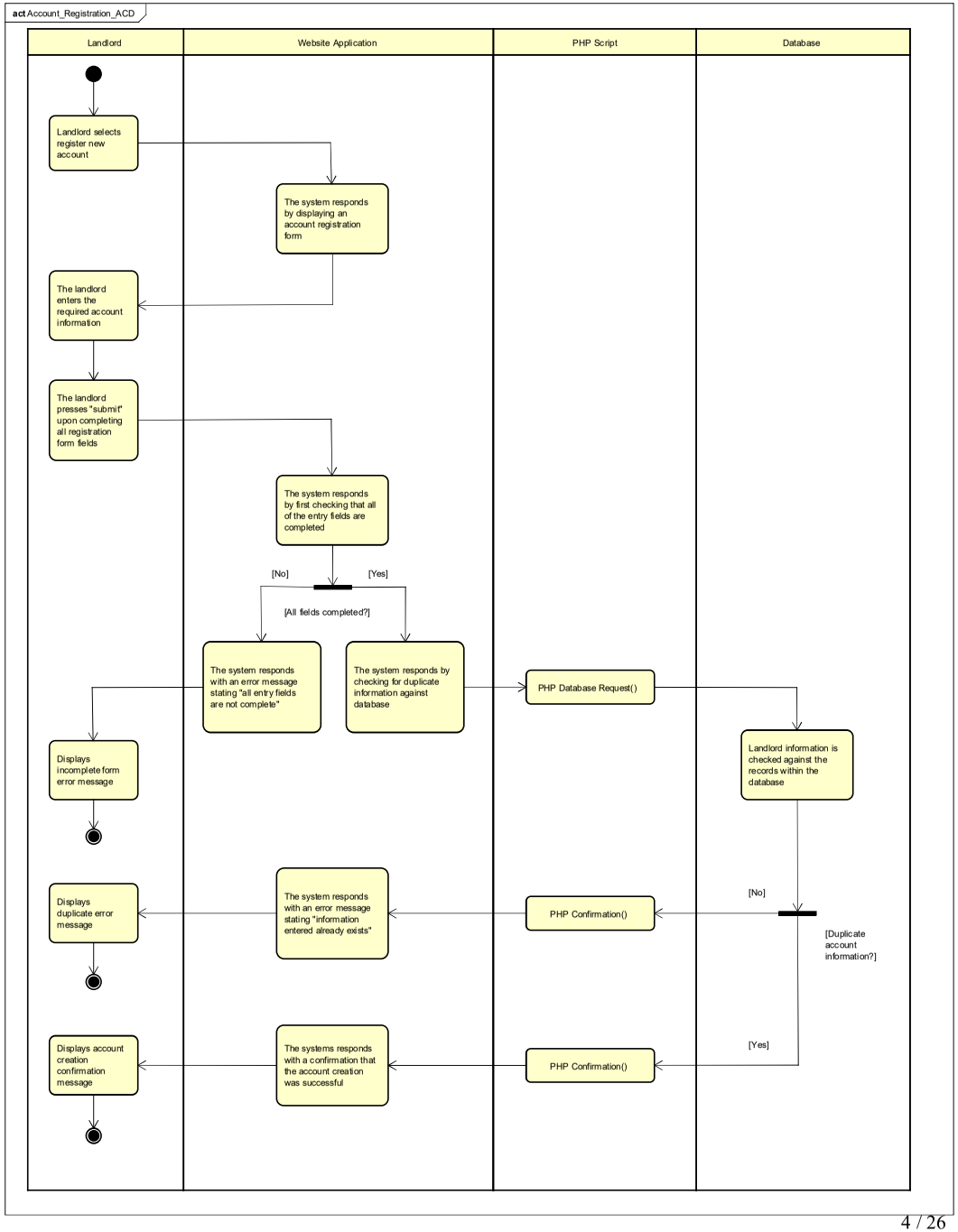
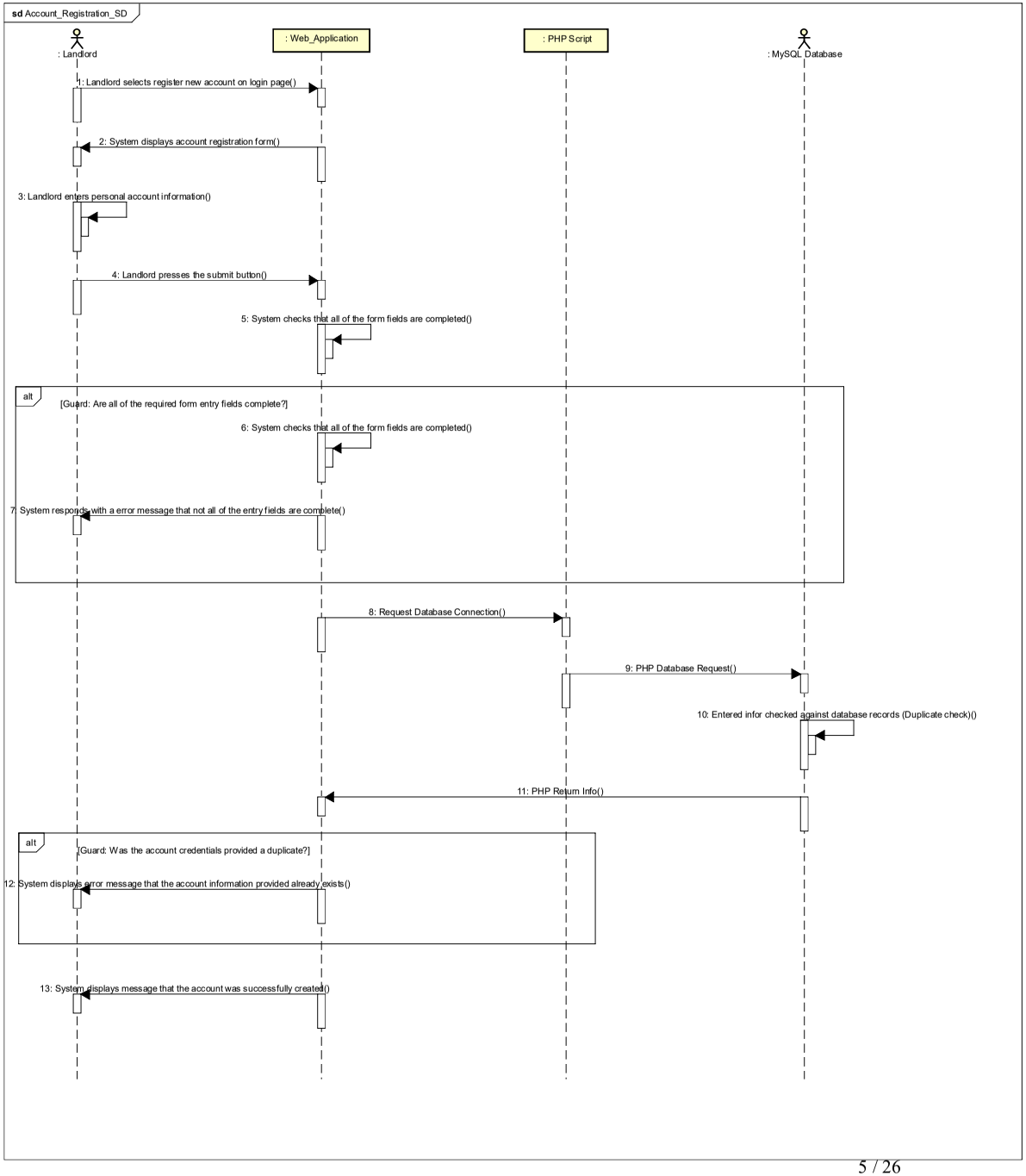
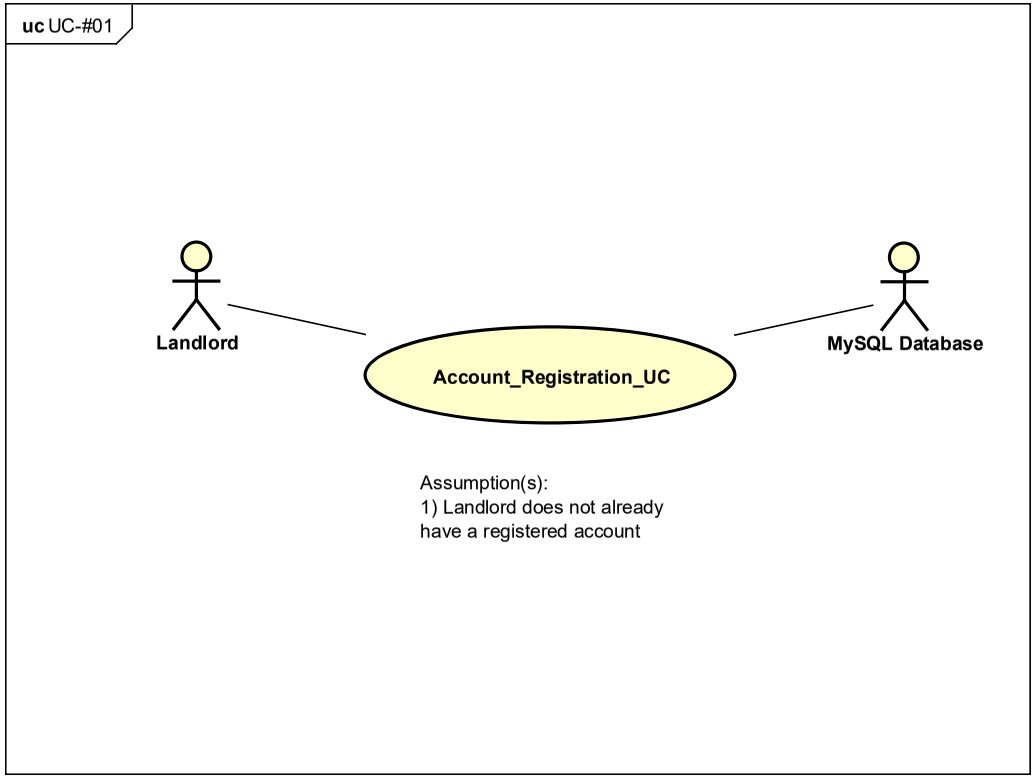

act Account_Login_ACD
Landlord
Website Application
PHP Script
Database
Landlord enters login account information
Landlord clicks login to begin the login process
System receives the login request
The system responds by checking the login account information against the database
PHP Database Request()
System checks the login credentials against the stored database information
[No]
[Correct Login Credentials?]
Display error message that the login information is incorrect
The system responds with a error message that "login information was incorrect"
PHP Return Info()
[Yes]
Display login confirmation message
The system logs the landlord into their unique account
PHP Return Info()

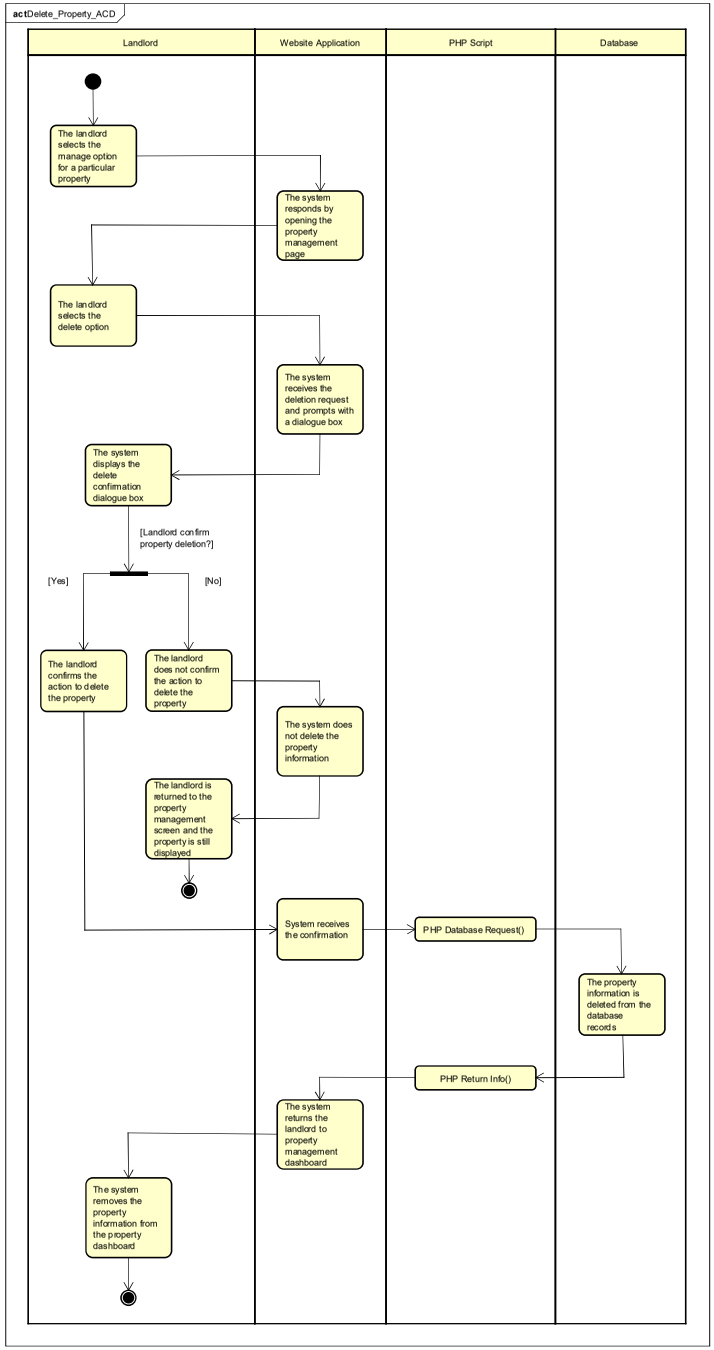

sd Account_Login_SD
: Landlord
: Web_Application
: PHP Script
: MySQL Database
1: Landlord enters account information on login page()
2: Landlord presses login button()
3: Receive login request()
4: Request Database connection()
5: PHP Database Request()
6: Check login credentials against stored information()
7: PHP Return Info()
alt
[Guard: Was the login information provided correct?]
8: System displays error message that the login info was incorrect()
9: (Same as in step 1)()
10: System logs landlord into their account()
11: System displays message that the landlord has been logged in()

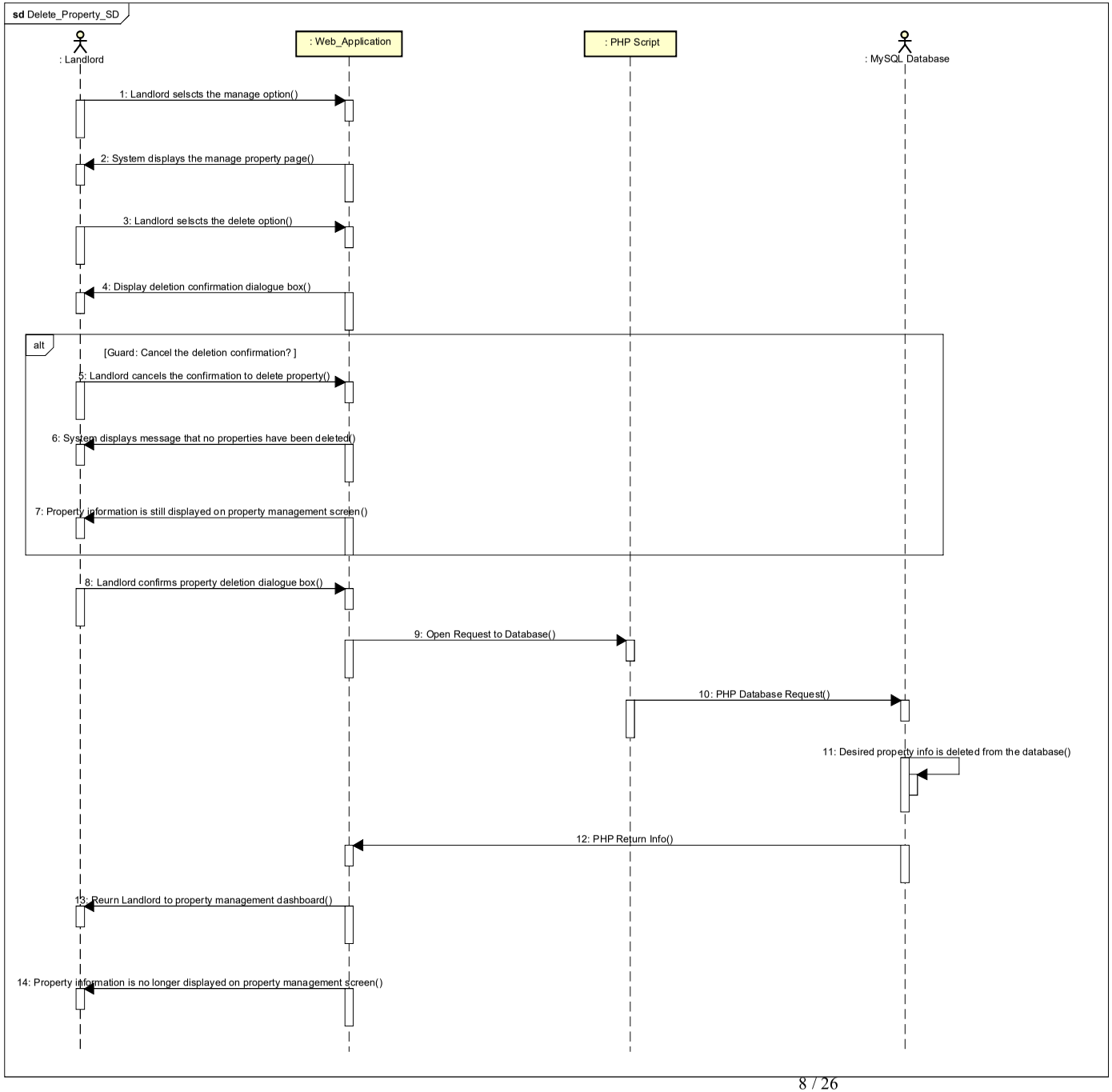

uc UC-#02
Account_Registratio
n_UC
<<include>>
Landlord
MySQL Database
Account_Login_UC
Assumption(s)
1) Landlord is already on the
login page
2) Landlord has registered
an account prior to login
attempt

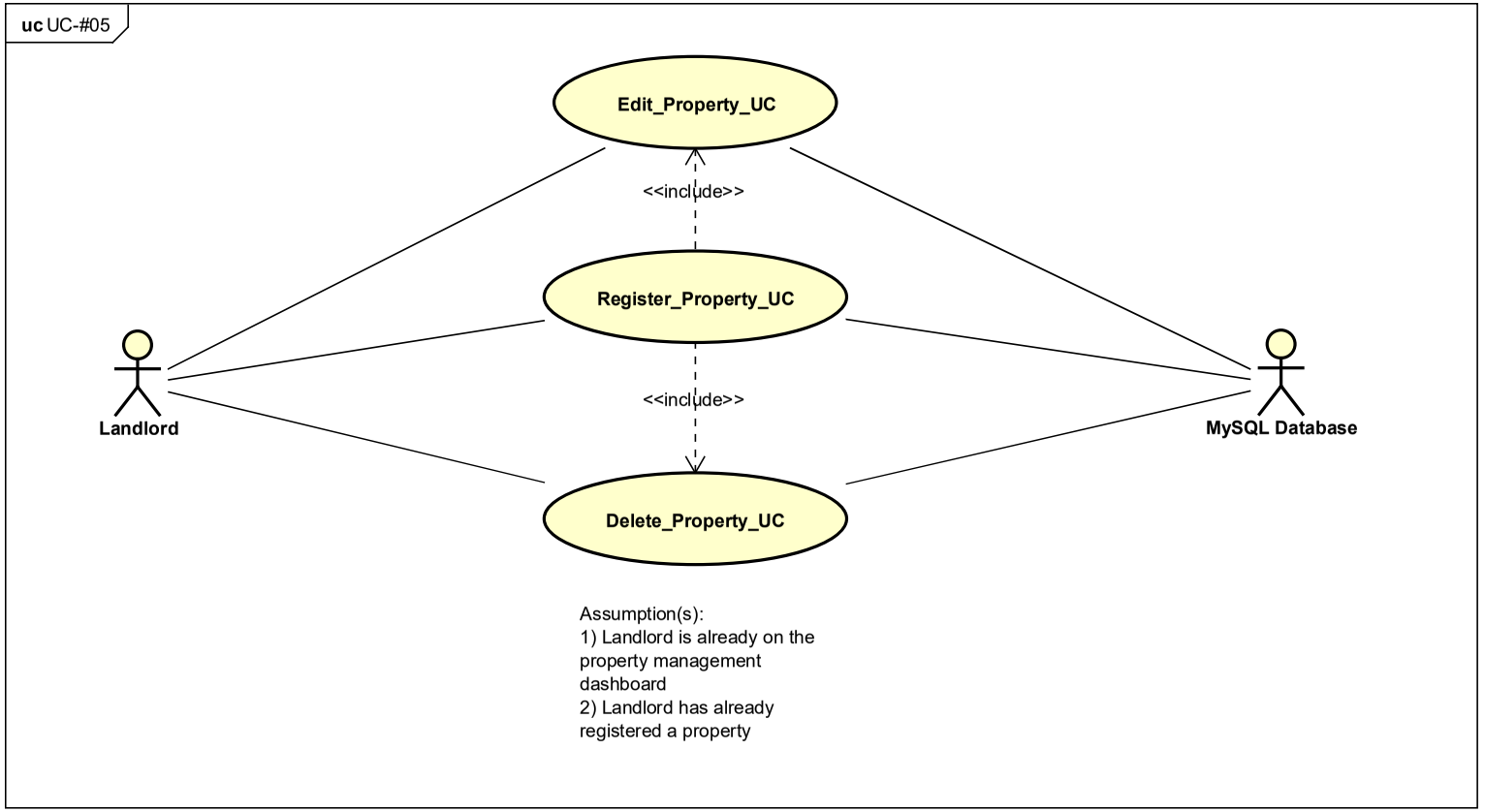

act Account_Registration_ACD
Landlord
Website Application
PHP Script
Database
Landlord selects register new account
The system responds by displaying an account registration form
The landlord enters the required account information
The landlord presses "submit" upon completing all registration form fields
The system responds by first checking that all of the entry fields are completed
[No]
[Yes]
[All fields completed?]
The system responds with an error message stating "all entry fields are not complete"
The system responds by checking for duplicate information against database
PHP Database Request()
Displays incomplete form error message
Landlord information is checked against the records within the database
Displays duplicate error message
The system responds with an error message stating "information entered already exists"
PHP Confirmation()
[No]
[Duplicate account information?]
Displays account creation confirmation message
The systems responds with a confirmation that the account creation was successful
PHP Confirmation()
[Yes]

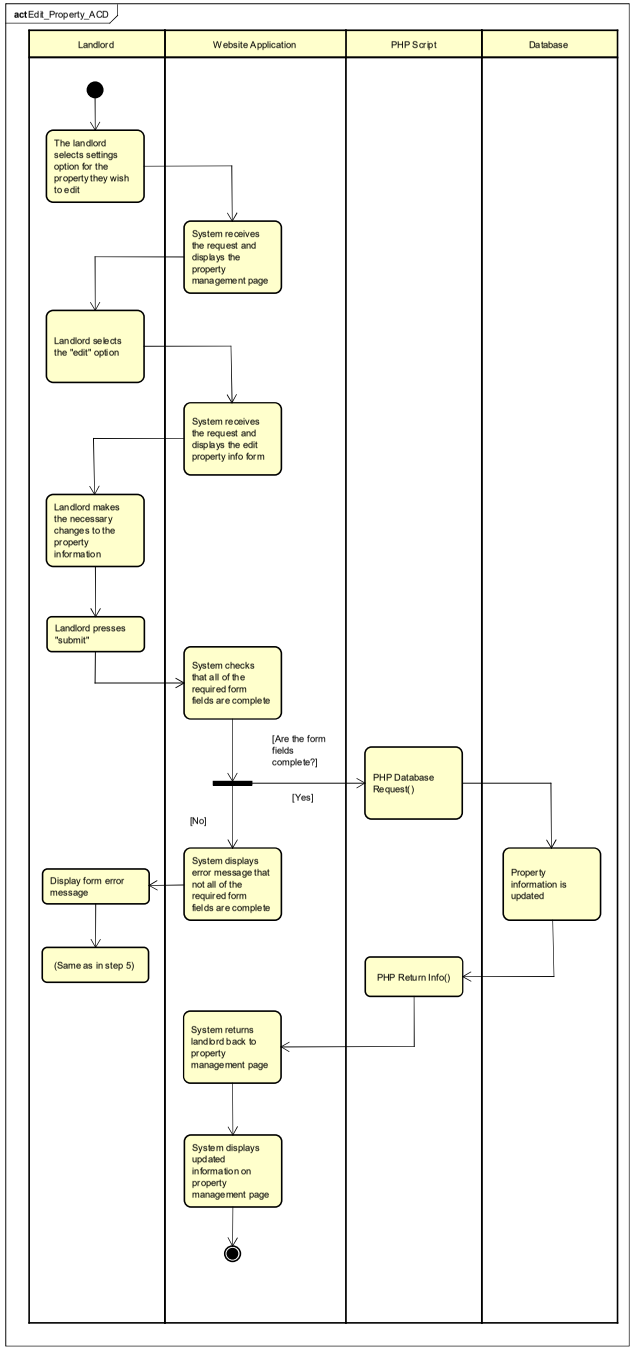

sd Account_Registration_SD
: Landlord
: Web_Application
: PHP Script
: MySQL Database
1: Landlord selects register new account on login page()
2: System displays account registration form()
3: Landlord enters personal account information()
4: Landlord presses the submit button()
5: System checks that all of the form fields are completed()
alt
[Guard: Are all of the required form entry fields complete?]
6: System checks that all of the form fields are completed()
7: System responds with a error message that not all of the entry fields are complete()
8: Request Database Connection()
9: PHP Database Request()
10: Entered infor checked against database records (Duplicate check)()
11: PHP Return Info()
alt
[Guard: Was the account credentials provided a duplicate?]
12: System displays error message that the account information provided already exists()
13: System displays message that the account was successfully created()

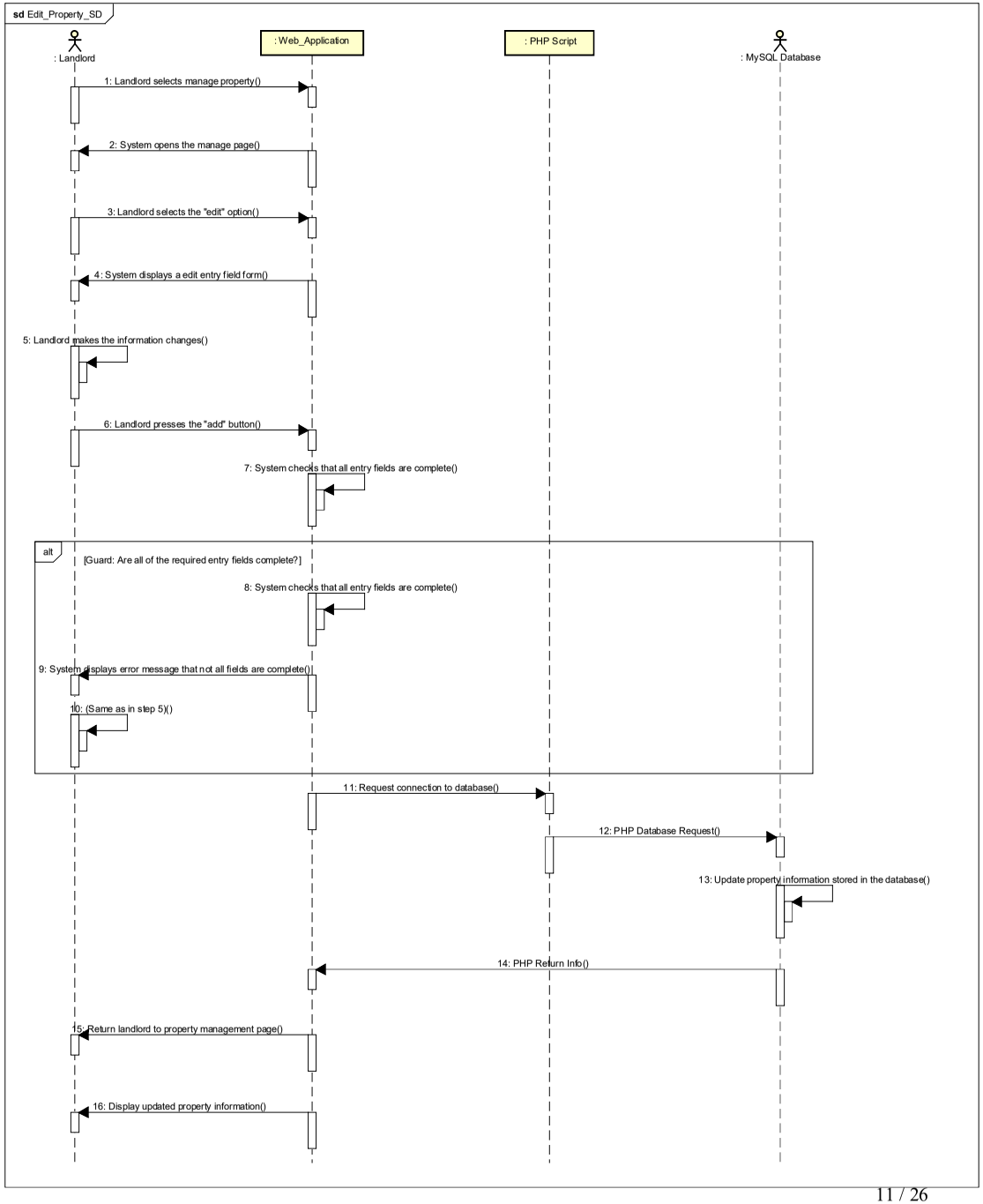

uc UC-#01
Landlord
Account_Registration_UC
MySQL Database
Assumption(s):
1) Landlord does not already
have a registered account

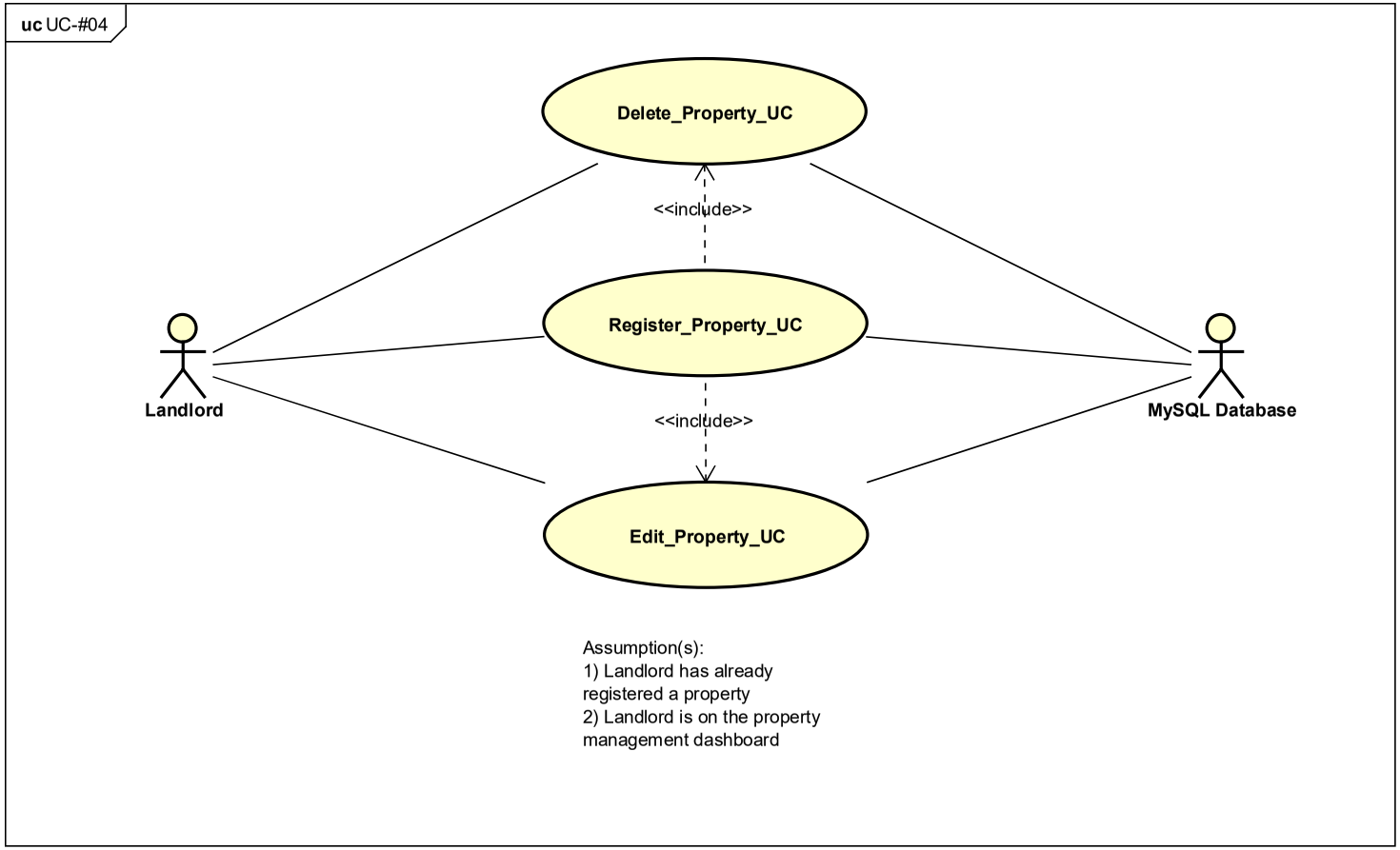

actDelete_Property_ACD
Landlord
Website Application
PHP Script
Database
The landlord selects the manage option for a particular property
The system responds by opening the property management page
The landlord selects the delete option
The system receives the deletion request and prompts with a dialogue box
The system displays the delete confirmation dialogue box
[Landlord confirm property deletion?]
[Yes]
[No]
The landlord confirms the action to delete the property
The landlord does not confirm the action to delete the property
The system does not delete the property information
The landlord is returned to the property management screen and the property is still displayed
System receives the confirmation
PHP Database Request()
The property information is deleted from the database records
PHP Return Info()
The system returns the landlord to property management dashboard
The system removes the property information from the property dashboard

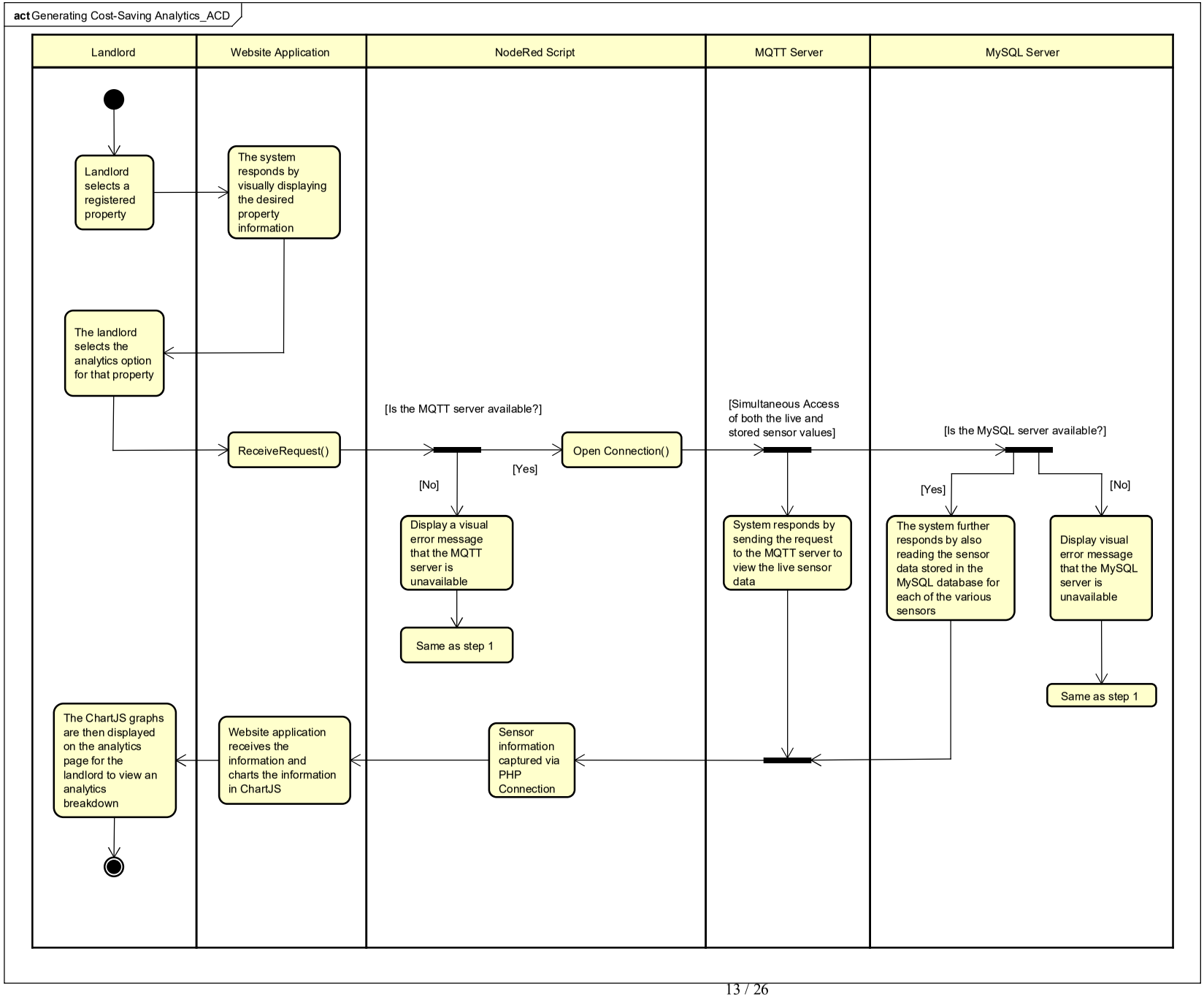

sd Delete_Property_SD
: Landlord
: Web_Application
: PHP Script
: MySQL Database
1: Landlord selscts the manage option()
2: System displays the manage property page()
3: Landlord selscts the delete option()
4: Display deletion confirmation dialogue box()
alt
[Guard: Cancel the deletion confirmation?]
5: Landlord cancels the confirmation to delete property()
6: System displays message that no properties have been deleted()
7: Property information is still displayed on property management screen()
8: Landlord confirms property deletion dialogue box()
9: Open Request to Database()
10: PHP Database Request()
11: Desired property info is deleted from the database()
12: PHP Return Info()
13: Reurn Landlord to property management dashboard()
14: Property information is no longer displayed on property management screen()

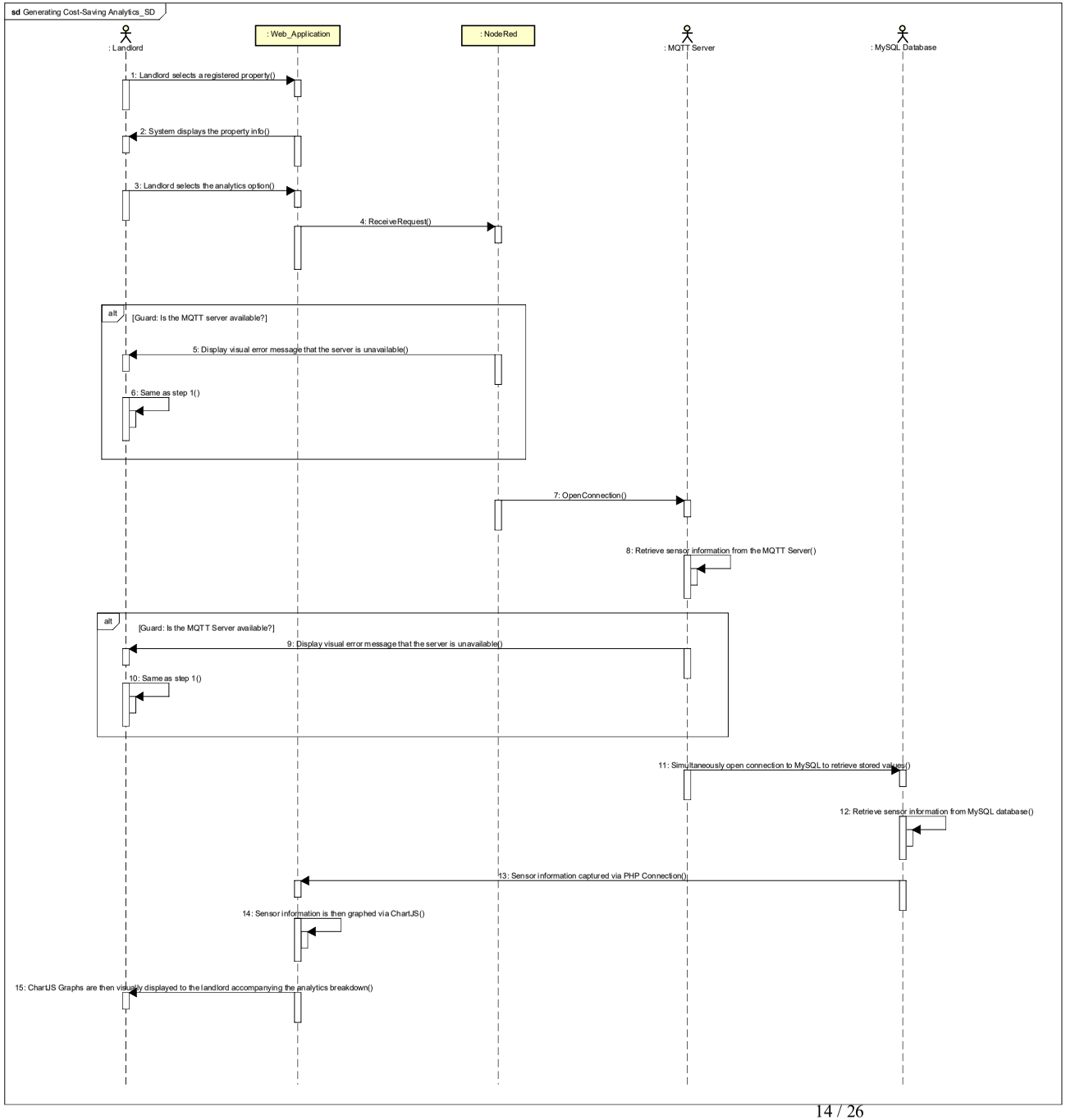

uc UC-#05
Edit_Property_UC
<<include>>
Register_Property_UC
<<include>>
Delete_Property_UC
Landlord
MySQL Database
Assumption(s):
1) Landlord is already on the property management dashboard
2) Landlord has already registered a property

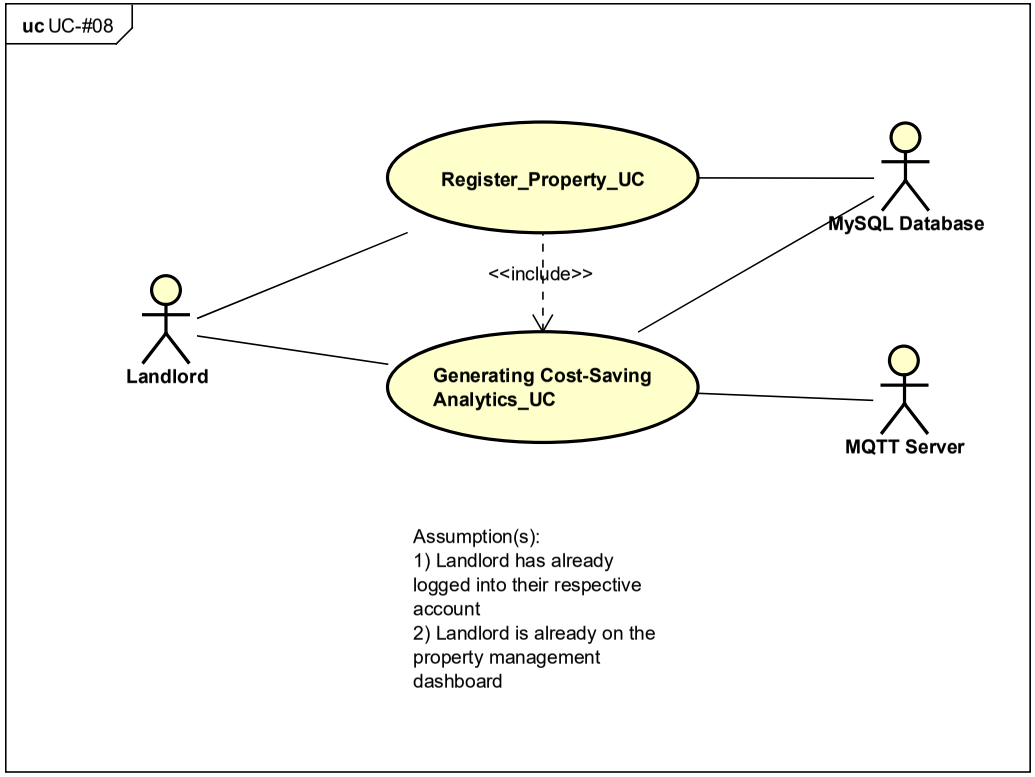

actEdit_Property_ACD
Landlord
Website Application
PHP Script
Database
The landlord selects settings option for the property they wish to edit
System receives the request and displays the property management page
Landlord selects the "edit" option
System receives the request and displays the edit property info form
Landlord makes the necessary changes to the property information
Landlord presses "submit"
System checks that all of the required form fields are complete
[Are the form fields complete?]
[Yes]
[No]
PHP Database Request()
Property information is updated
System displays error message that not all of the required form fields are complete
Display form error message
(Same as in step 5)
PHP Return Info()
System returns landlord back to property management page
System displays updated information on property management page

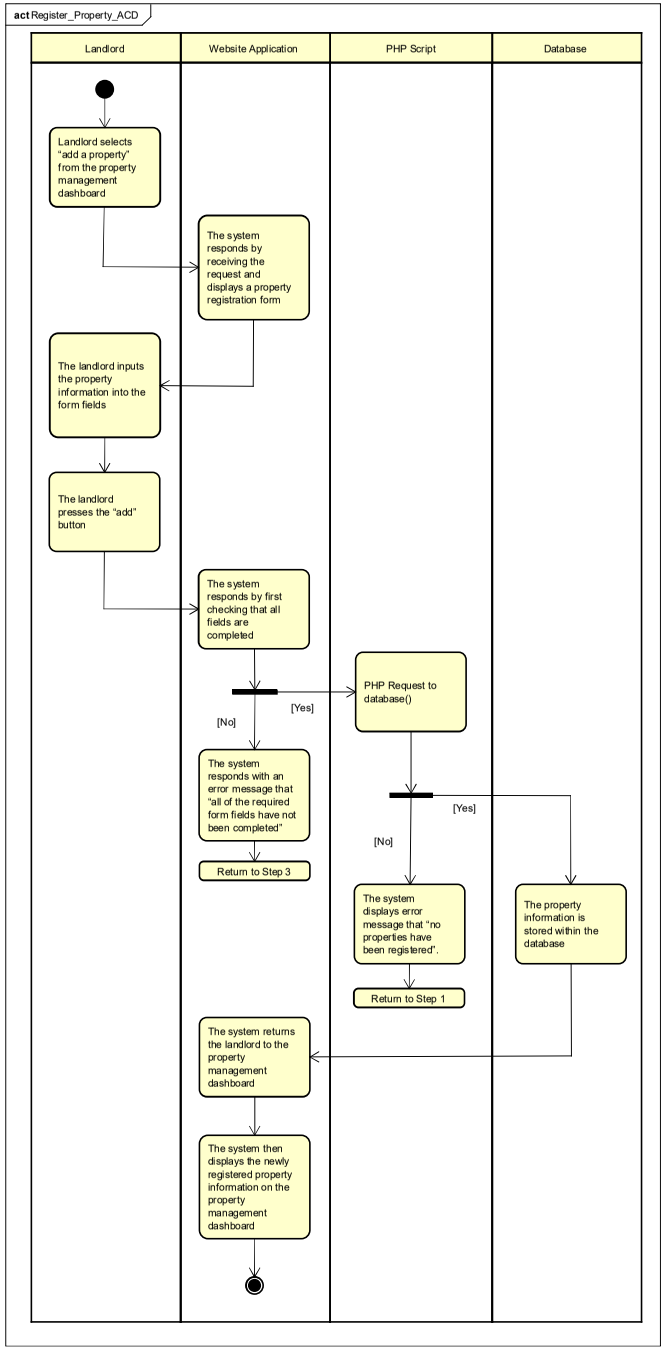

sd Edit_Property_SD
: Landlord
: Web_Application
: PHP Script
: MySQL Database
1: Landlord selects manage property()
2: System opens the manage page()
3: Landlord selects the "edit" option()
4: System displays a edit entry field form()
5: Landlord makes the information changes()
6: Landlord presses the "add" button()
7: System checks that all entry fields are complete()
alt
[Guard: Are all of the required entry fields complete?]
8: System checks that all entry fields are complete()
9: System displays error message that not all fields are complete()
10: (Same as in step 5)()
11: Request connection to database()
12: PHP Database Request()
13: Update property information stored in the database()
14: PHP Return Info()
15: Return landlord to property management page()
16: Display updated property information()

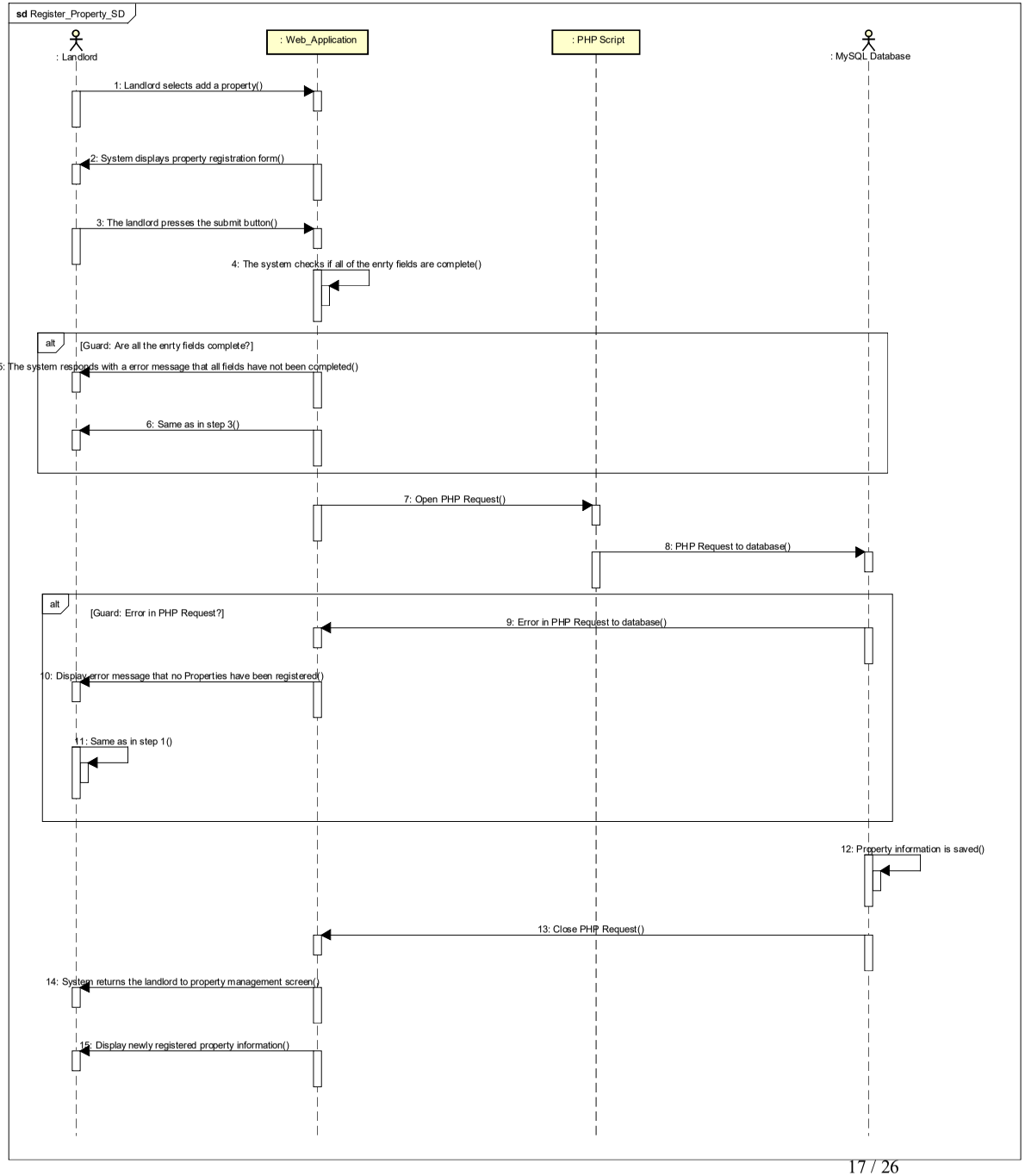

uc UC-#04
Delete_Property_UC
<<include>>
Register_Property_UC
<<include>>
Edit_Property_UC
Landlord
MySQL Database
Assumption(s):
1) Landlord has already registered a property
2) Landlord is on the property management dashboard

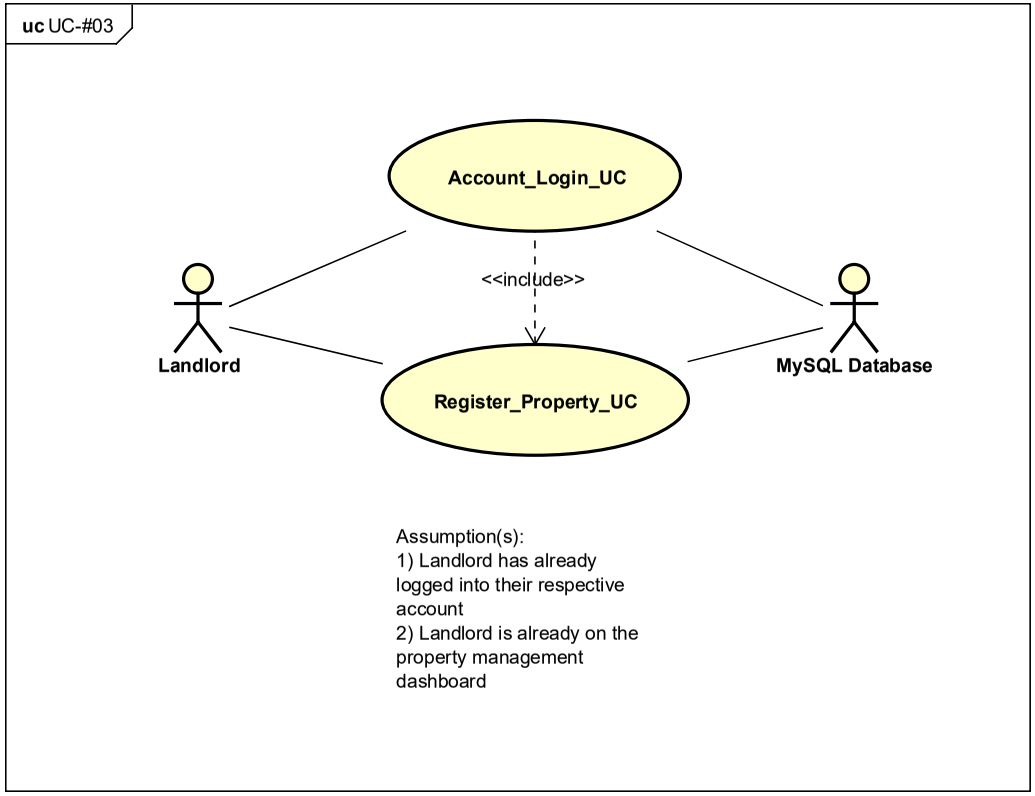

act Generating Cost-Saving Analytics_ACD
Landlord
Website Application
NodeRed Script
MQTT Server
MySQL Server
Landlord selects a registered property
The system responds by visually displaying the desired property information
The landlord selects the analytics option for that property
ReceiveRequest()
[Is the MQTT server available?]
[No]
[Yes]
Open Connection()
Display a visual error message that the MQTT server is unavailable
Same as step 1
[Simultaneous Access of both the live and stored sensor values]
System responds by sending the request to the MQTT server to view the live sensor data
[Is the MySQL server available?]
[Yes]
[No]
The system further responds by also reading the sensor data stored in the MySQL database for each of the various sensors
Display visual error message that the MySQL server is unavailable
Same as step 1
Sensor information captured via PHP Connection
Website application receives the information and charts the information in ChartJS
The ChartJS graphs are then displayed on the analytics page for the landlord to view an analytics breakdown

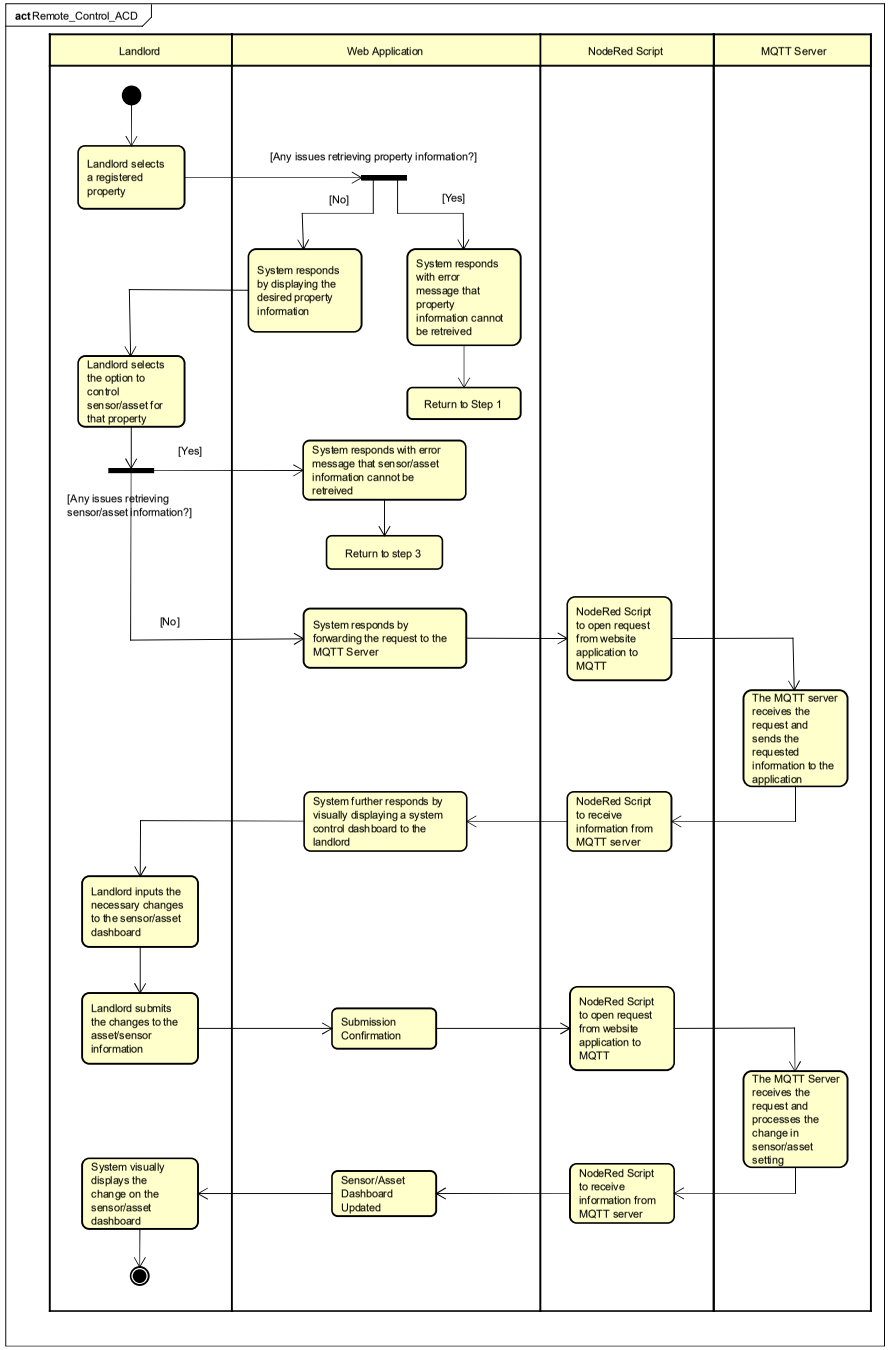

sd Generating Cost-Saving Analytics_SD
: Landlord
: Web_Application
: NodeRed
: MQTT Server
: MySQL Database
1: Landlord selects a registered property()
2: System displays the property info()
3: Landlord selects the analytics option()
4: ReceiveRequest()
alt
[Guard: Is the MQTT server available?]
5: Display visual error message that the server is unavailable()
6: Same as step 1()
7: OpenConnection()
8: Retrieve sensor information from the MQTT Server()
alt
[Guard: Is the MQTT Server available?]
9: Display visual error message that the server is unavailable()
10: Same as step 1()
11: Simultaneously open connection to MySQL to retrieve stored values()
12: Retrieve sensor information from MySQL database()
13: Sensor information captured via PHP Connection()
14: Sensor information is then graphed via ChartJS()
15: ChartJS Graphs are then visually displayed to the landlord accompanying the analytics breakdown()

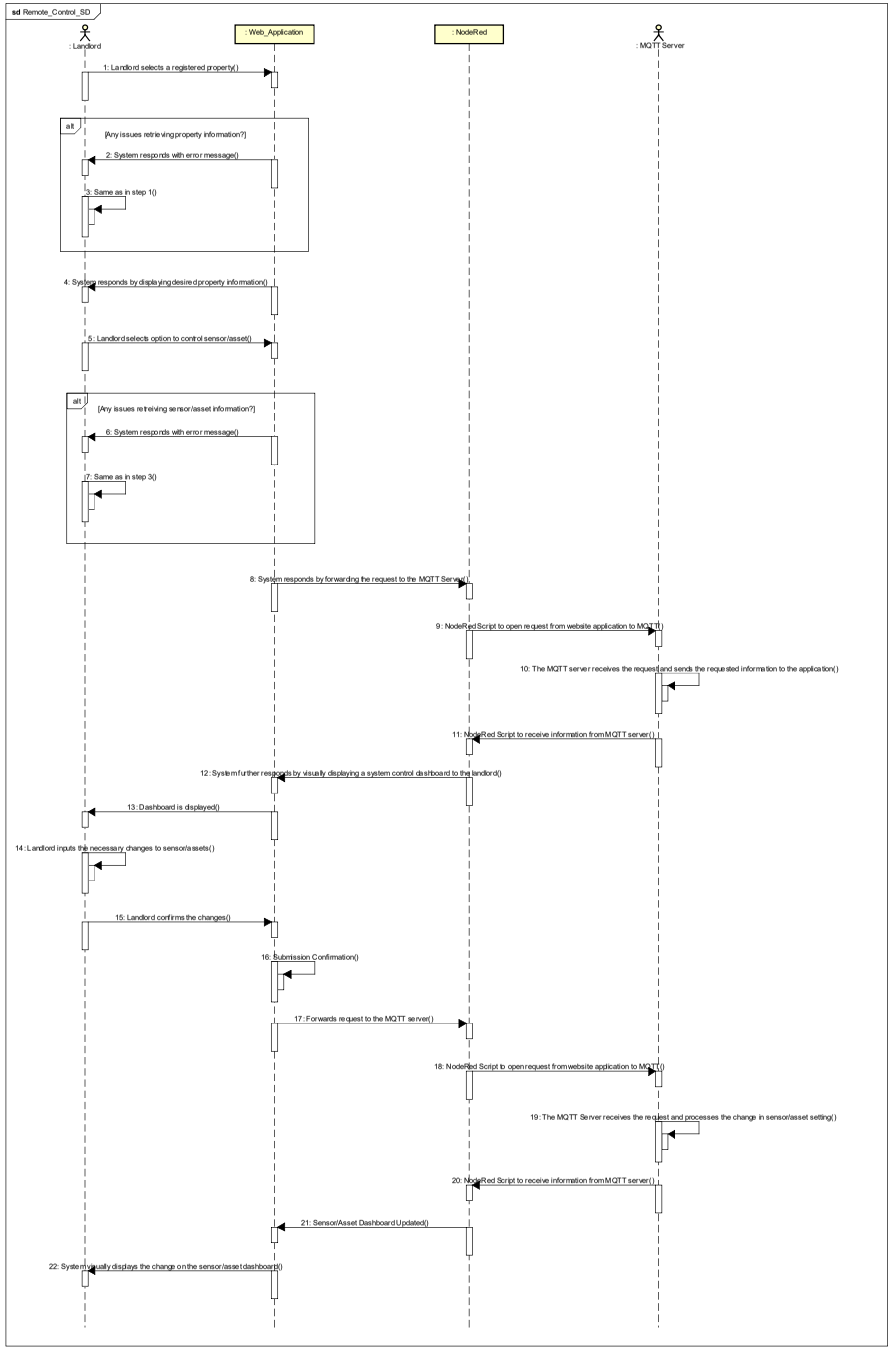

uc UC-#08
Register_Property_UC
MySQL Database
<<include>>
Landlord
Generating Cost-Saving Analytics_UC
MQTT Server
Assumption(s):
1) Landlord has already logged into their respective account
2) Landlord is already on the property management dashboard

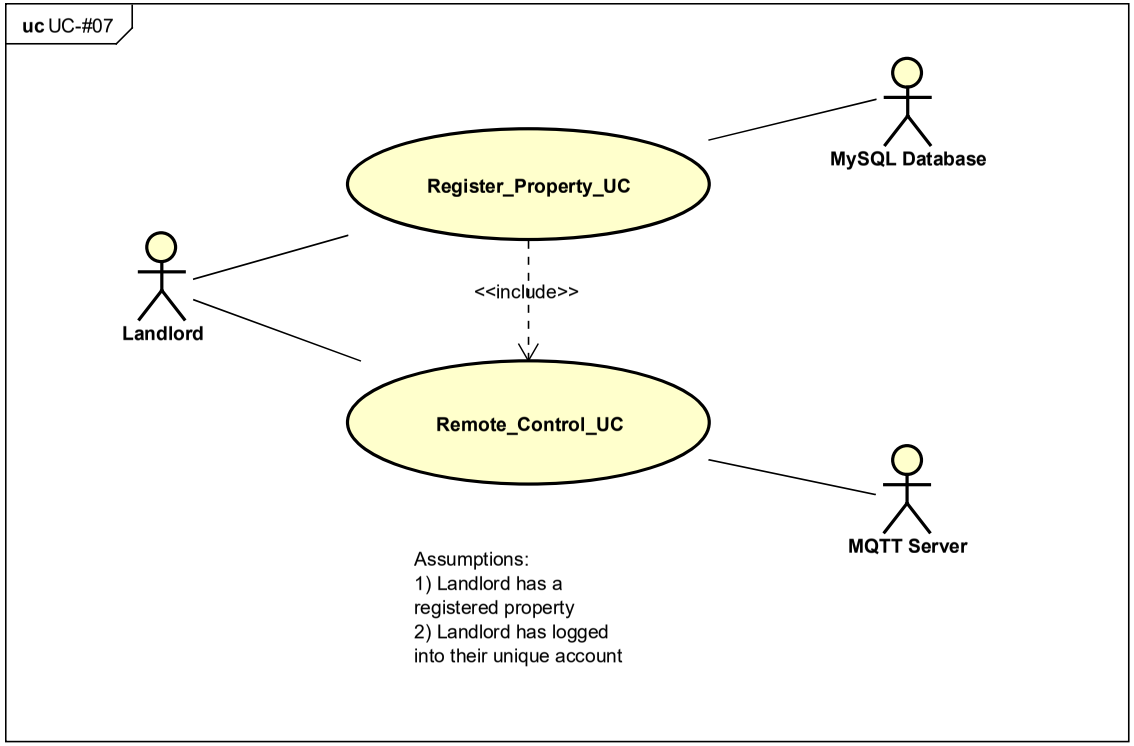

act Register_Property_ACD
Landlord
Website Application
PHP Script
Database
Landlord selects "add a property" from the property management dashboard
The system responds by receiving the request and displays a property registration form
The landlord inputs the property information into the form fields
The landlord presses the "add" button
The system responds by first checking that all fields are completed
[Yes]
PHP Request to database()
[No]
The system responds with an error message that "all of the required form fields have not been completed"
Return to Step 3
[Yes]
[No]
The system displays error message that "no properties have been registered".
Return to Step 1
The property information is stored within the database
The system returns the landlord to the property management dashboard
The system then displays the newly registered property information on the property management dashboard

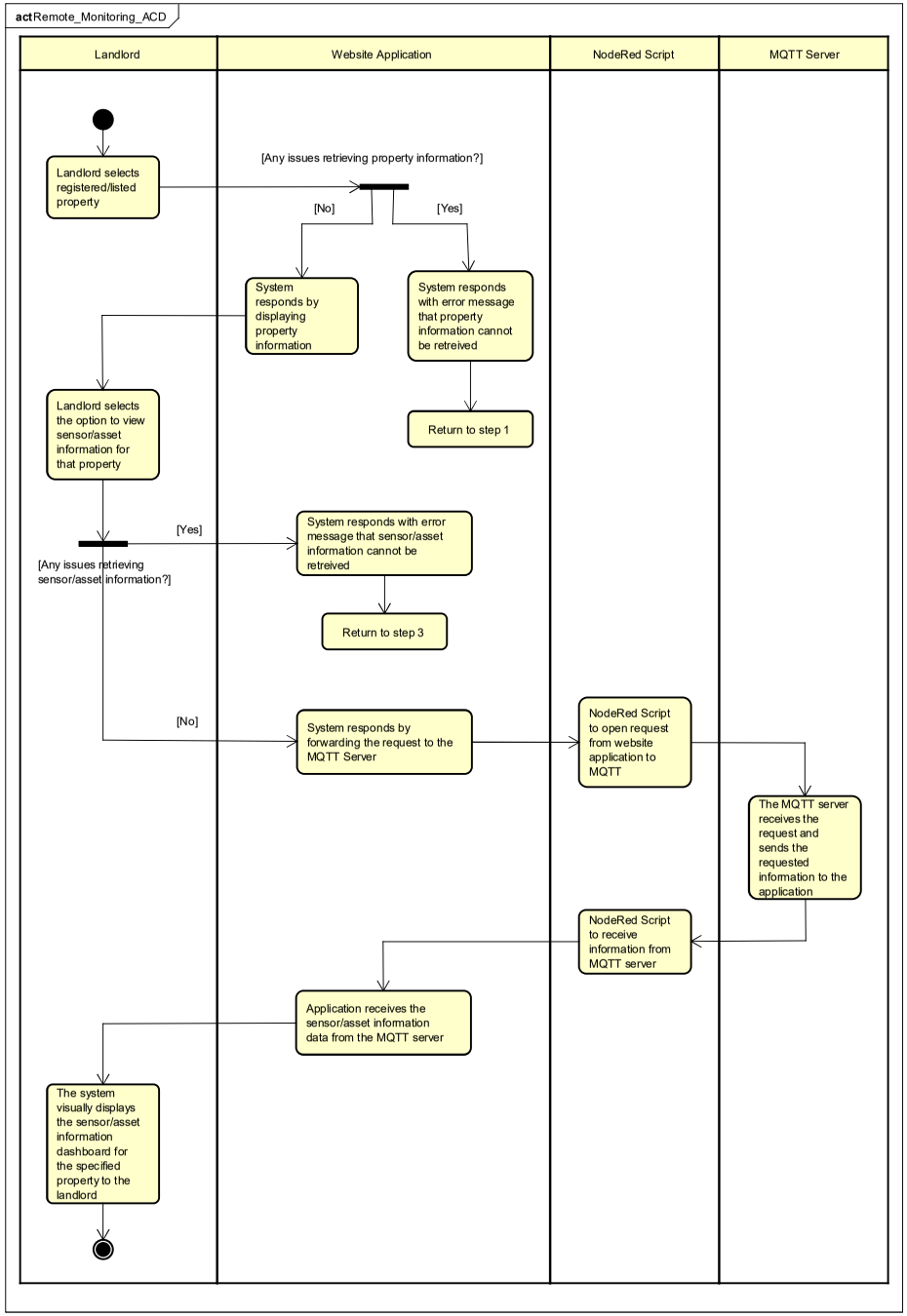

sd Register_Property_SD
: Landlord
: Web_Application
: PHP Script
: MySQL Database
1: Landlord selects add a property()
2: System displays property registration form()
3: The landlord presses the submit button()
4: The system checks if all of the enrty fields are complete()
alt
[Guard: Are all the enrty fields complete?]
5: The system responds with a error message that all fields have not been completed()
6: Same as in step 3()
7: Open PHP Request()
8: PHP Request to database()
alt
[Guard: Error in PHP Request?]
9: Error in PHP Request to database()
10: Display error message that no Properties have been registered()
11: Same as in step 1()
12: Property information is saved()
13: Close PHP Request()
14: System returns the landlord to property management screen()
15: Display newly registered property information()

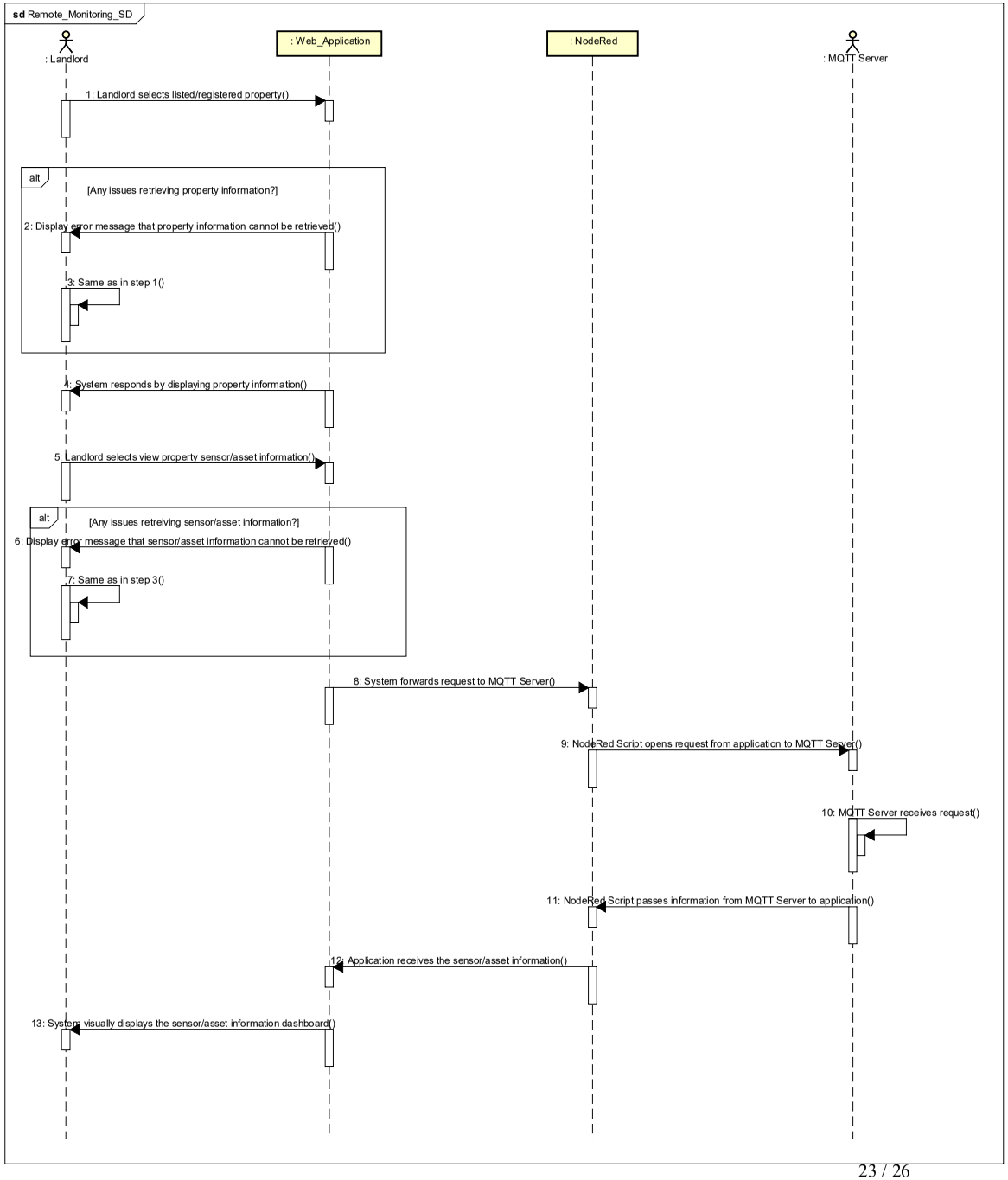

uc UC-#03
Account_Login_UC
<<include>>
Landlord
MySQL Database
Register_Property_UC
Assumption(s):
1) Landlord has already logged into their respective account
2) Landlord is already on the property management dashboard

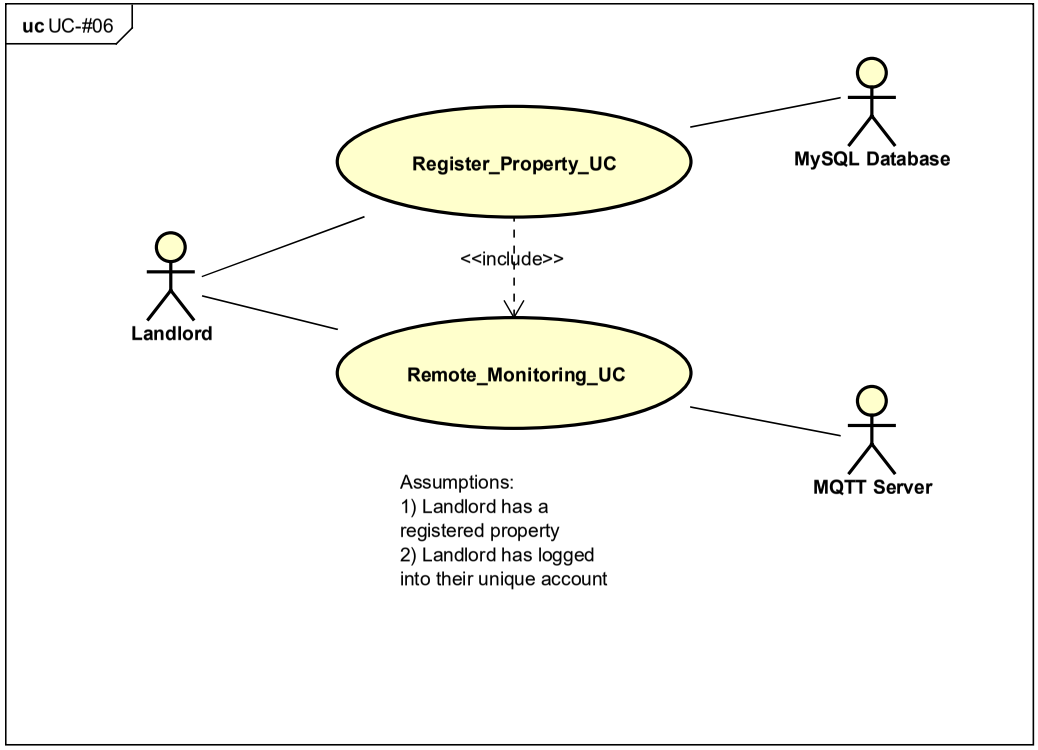

act Remote_Control_ACD
Landlord
Web Application
NodeRed Script
MQTT Server
Landlord selects a registered property
[Any issues retrieving property information?]
[No]
[Yes]
System responds by displaying the desired property information
System responds with error message that property information cannot be retreived
Return to Step 1
Landlord selects the option to control sensor/asset for that property
[Yes]
System responds with error message that sensor/asset information cannot be retreived
[Any issues retrieving sensor/asset information?]
Return to step 3
[No]
System responds by forwarding the request to the MQTT Server
NodeRed Script to open request from website application to MQTT
The MQTT server receives the request and sends the requested information to the application
System further responds by visually displaying a system control dashboard to the landlord
NodeRed Script to receive information from MQTT server
Landlord inputs the necessary changes to the sensor/asset dashboard
Landlord submits the changes to the asset/sensor information
Submission Confirmation
NodeRed Script to open request from website application to MQTT
The MQTT Server receives the request and processes the change in sensor/asset setting
System visually displays the change on the sensor/asset dashboard
Sensor/Asset Dashboard Updated
NodeRed Script to receive information from MQTT server

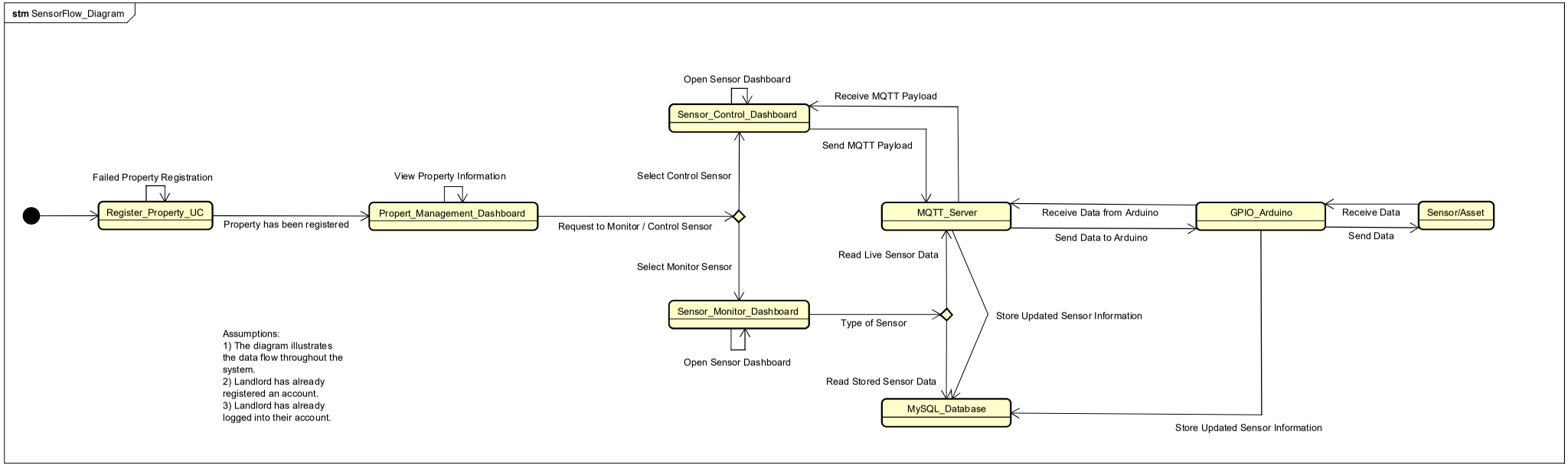

sd Remote_Control_SD
: Landlord
: Web_Application
: NodeRed
: MQTT Server
1: Landlord selects a registered property()
alt
[Any issues retrieving property information?]
2: System responds with error message()
3: Same as in step 1()
4: System responds by displaying desired property information()
5: Landlord selects option to control sensor/asset()
alt
[Any issues retreiving sensor/asset information?]
6: System responds with error message()
7: Same as in step 3()
8: System responds by forwarding the request to the MQTT Server()
9 : NodeRed Script to open request from website application to MQTT()
10: The MQTT server receives the request and sends the requested information to the application()
11: NodeRed Script to receive information from MQTT server()
12: System further responds by visually displaying a system control dashboard to the landlord()
13: Dashboard is displayed()
14: Landlord inputs the necessary changes to sensor/assets()
15: Landlord confirms the changes()
16: Submission Confirmation()
17: Forwards request to the MQTT server()
18: NodeRed Script to open request from website application to MQTT()
19: The MQTT Server receives the request and processes the change in sensor/asset setting()
20: NodeRed Script to receive information from MQTT server()
21: Sensor/Asset Dashboard Updated()
22: System visually displays the change on the sensor/asset dashboard()

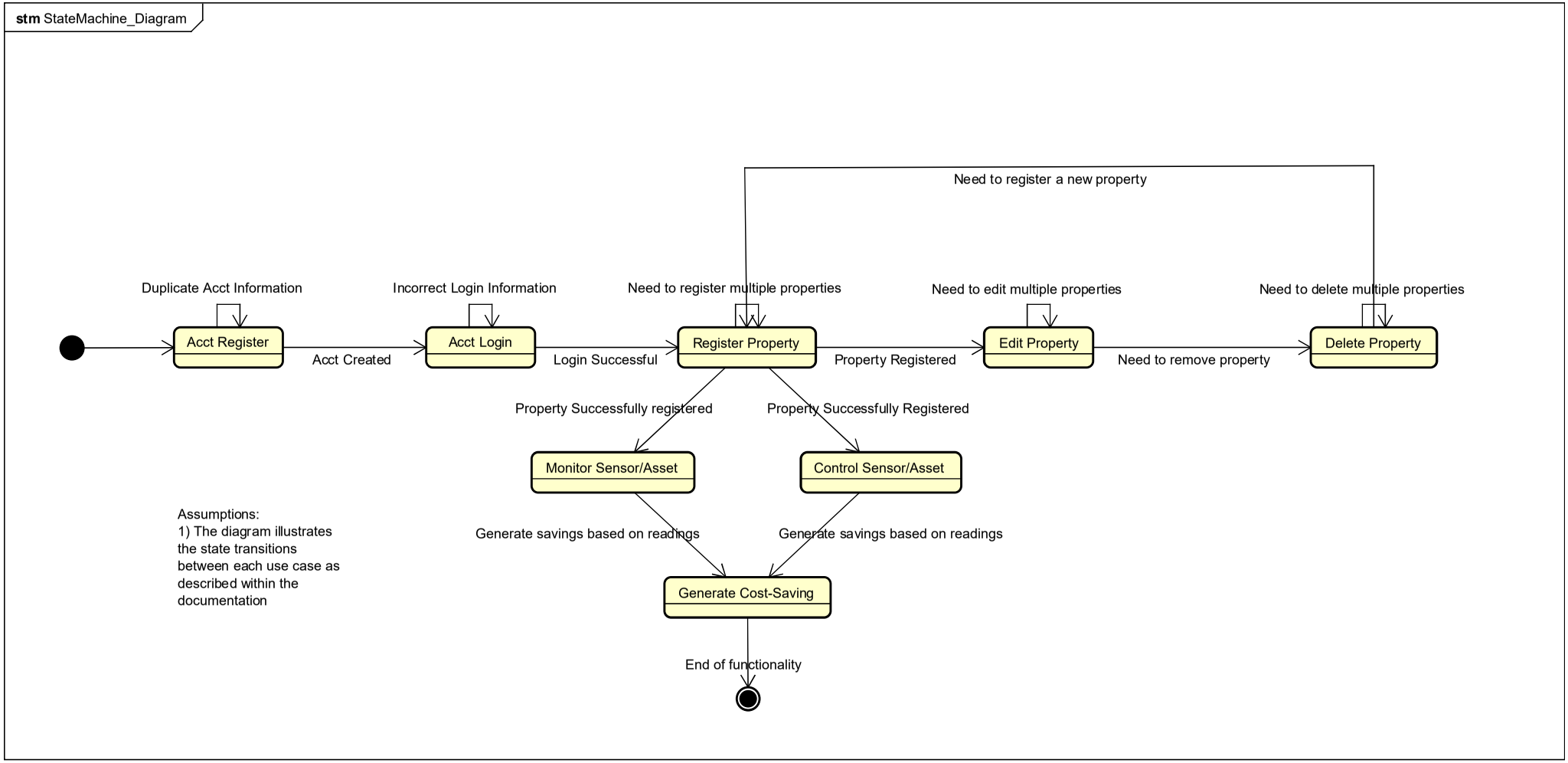

uc UC-#07
MySQL Database
Register_Property_UC
Landlord
<<include>>
Remote_Control_UC
MQTT Server
Assumptions:
1) Landlord has a registered property
2) Landlord has logged into their unique account


actRemote_Monitoring_ACD
Landlord
Website Application
NodeRed Script
MQTT Server
Landlord selects registered/listed property
[Any issues retrieving property information?]
[No]
[Yes]
System responds by displaying property information
System responds with error message that property information cannot be retreived
Return to step 1
Landlord selects the option to view sensor/asset information for that property
[Yes]
[Any issues retrieving sensor/asset information?]
System responds with error message that sensor/asset information cannot be retreived
Return to step 3
[No]
System responds by forwarding the request to the MQTT Server
NodeRed Script to open request from website application to MQTT
The MQTT server receives the request and sends the requested information to the application
NodeRed Script to receive information from MQTT server
Application receives the sensor/asset information data from the MQTT server
The system visually displays the sensor/asset information dashboard for the specified property to the landlord


sd Remote_Monitoring_SD
: Landlord
: Web_Application
: NodeRed
: MQTT Server
1: Landlord selects listed/registered property()
alt
[Any issues retrieving property information?]
2: Display error message that property information cannot be retrieved()
3: Same as in step 1()
4: System responds by displaying property information()
5: Landlord selects view property sensor/asset information()
alt
[Any issues retreiving sensor/asset information?]
6: Display error message that sensor/asset information cannot be retrieved()
7: Same as in step 3()
8: System forwards request to MQTT Server()
9: NodeRed Script opens request from application to MQTT Server()
10: MQTT Server receives request()
11: NodeRed Script passes information from MQTT Server to application()
12: Application receives the sensor/asset information()
13: System visually displays the sensor/asset information dashboard()


uc UC-#06
Register_Property_UC
MySQL Database
<<include>>
Landlord
Remote_Monitoring_UC
MQTT Server
Assumptions:
1) Landlord has a
registered property
2) Landlord has logged
into their unique account


stm SensorFlow_Diagram
Open Sensor Dashboard
Receive MQTT Payload
Sensor_Control_Dashboard
Send MQTT Payload
Failed Property Registration
View Property Information
Select Control Sensor
Register_Property_UC
Property has been registered
Propert_Management_Dashboard
Request to Monitor / Control Sensor
MQTT_Server
Receive Data from Arduino
GPIO_Arduino
Receive Data
Sensor/Asset
Send Data to Arduino
Send Data
Read Live Sensor Data
Select Monitor Sensor
Sensor_Monitor_Dashboard
Type of Sensor
Store Updated Sensor Information
Assumptions:
1) The diagram illustrates the data flow throughout the system.
2) Landlord has already registered an account.
3) Landlord has already logged into their account.
Open Sensor Dashboard
Read Stored Sensor Data
MySQL_Database
Store Updated Sensor Information


stm StateMachine_Diagram
Need to register a new property
Duplicate Acct Information
Incorrect Login Information
Need to register multiple properties
Need to edit multiple properties
Need to delete multiple properties
Acct Register
Acct Created
Acct Login
Login Successful
Register Property
Property Registered
Edit Property
Need to remove property
Delete Property
Property Successfully registered
Property Successfully Registered
Monitor Sensor/Asset
Control Sensor/Asset
Assumptions:
1) The diagram illustrates the state transitions between each use case as described within the documentation
Generate savings based on readings
Generate savings based on readings
Generate Cost-Saving
End of functionality