

CSP 586 Project

Weather Notification System



Darshankumar Zala
(dzala@iit.edu)
Snehal Mahalle
(smahalle@iit.edu)

Table of Contents

1. Project Overview.....	2
2. Requirements/Features of Application.....	2
3. Use Cases	3
3.1. Actors	3
3.1.1. Primary Actor	3
3.1.2. Secondary Actors	3
3.1.3. Off-stage Actors	3
3.2. Use Cases – Brief Format	3
3.3. Use Cases in Fully-Dressed format.....	5
3.3.1. Use Case: Show current weather condition.....	5
3.3.2. Use Case: Share current weather condition	6
3.3.3. Notify weather alerts	7
4. Use Case Diagram	8
5. Domain Model	9
6. Sequence Diagrams.....	10
6.1. Sequence Diagram for view current weather condition.....	10
6.2. Sequence Diagram for sharing weather condition	11
6.3. Sequence Diagram for Change temperature settings	12
7. Design Model	13
7.1. Class Diagram	13
8. Design Patterns	14
8.1. Observer Pattern.....	14
8.2. Strategy Pattern	15
8.3. Abstract Factory.....	16

1. Project Overview

- ▶ Weather notification system app is used to display the current weather conditions as well as it alerts user with location-specific weather warnings.
- ▶ When severe weather such as tornados, lightning, flooding, hurricanes, and snowstorms threaten to strike users area, system sends alerts to user by email, text or by call.
- ▶ This system uses the data provided by the National weather system and Emergency Alert System (E.A.S.) for notification and for displaying the current weather conditions.

2. Requirements/Features of Application

- System should ask user for his/her current location.
- System should display current weather conditions such as temperature, humidity, dew point, visibility, pressure, sunrise, sunset
- System should display weather forecast for 24 hours, 2 days, 10 days, 1 month
- System should display severe weather conditions such as tornados, lightning, flooding, hurricanes, and snowstorms.
- System should notify weather alerts to the subscribed users.
- User should be able to share weather conditions with his/her family and friends by email, Facebook post or by sending text.
- System should notify the subscribed user in case of natural calamities.

3. Use Cases

3.1. Actors

3.1.1. Primary Actor

- Application User

3.1.2. Secondary Actors

- National Weather Service
- Emergency Alerts System (E.A.S.)
- Geo spacious System
- E-mail service provider
- Facebook
- Call/Text Service provider

3.1.3. Off-stage Actors

- Weather Stations

3.2. Use Cases – Brief Format

1. Search location by providing zip-code or street address

- User starts application
- User enters location either ZIP-Code or street address
- System validate location
- System request for the weather details for the location
- System displays weather details.

2. View current weather conditions

- User starts application
- System gets current location from Geo Spacious System
- System request for the weather details for the location
- System displays weather details.

3. View weather forecast

- User starts application
- User enters location either ZIP-Code or street address
- System validate location
- System request for the weather forecast details.
- System displays weather details.

4. View severe weather alerts

- User starts application
- User enters location either ZIP-Code or street address
- System validate location
- System request for the weather alerts for the location.

- System displays weather alerts.

5. Share weather alerts

- User starts application
- User enters location either ZIP-Code or street address
- System validates location.
- System request for the weather alerts for the location.
- System displays weather alerts.
- User share weather alerts by Facebook, E-mail or Text.

6. Change temperature setting

- User starts application
- User select change settings option.
- User select change Temperature Settings.
- User set the temperature either in Celsius or Fahrenheit.
- System display weather condition in appropriate settings.

7. Subscribe for weather notification alerts.

- User starts application
- User select change settings option.
- User select weather alerts option
- User subscribes for the weather alert notification either by Text, Call or Mail.

8. Unsubscribe form weather notification alerts

- User starts application
- User select change settings option.
- User select weather alerts option
- User unsubscribes for the weather alerts.

9. Notify subscribed user for severe weather alerts

- System search for the severe weather alerts for location registered by user.
- System notify user for the severe weather condition by E-mail, Call or Text message.

3.3. Use Cases in Fully-Dressed format

3.3.1. Use Case: Show current weather condition

Use Case Section	Comment
Use Case Name	Show current weather condition
Scope	Weather Notification system for 3G smartphone
Level	User-goal
Primary Actor	End user
Stakeholders and Interests	End user: <ul style="list-style-type: none">○ Wants to get weather conditions Notional Weather Service: <ul style="list-style-type: none">○ Provide weather data to the system
Preconditions	<ul style="list-style-type: none">○ There should be an internet connection to connect cell phone with the server○ User should enter valid location
Success Guarantee	<ul style="list-style-type: none">○ Current weather condition for the specific location is displayed.
Main Success Scenario	<ul style="list-style-type: none">○ User starts the app to check the weather○ User enters the desired location○ System fetches the weather details from server for the desired location○ System displays the weather details○ User gets the current weather conditions
Extensions	At any time, internet connection fails: 1. If the Wi-Fi connection fails the system will try to acquire the 3G internet connection.
Special Requirements	<ul style="list-style-type: none">○ Application should response within 20 sec
Frequency of Occurrence	Depends on users use
Miscellaneous	-

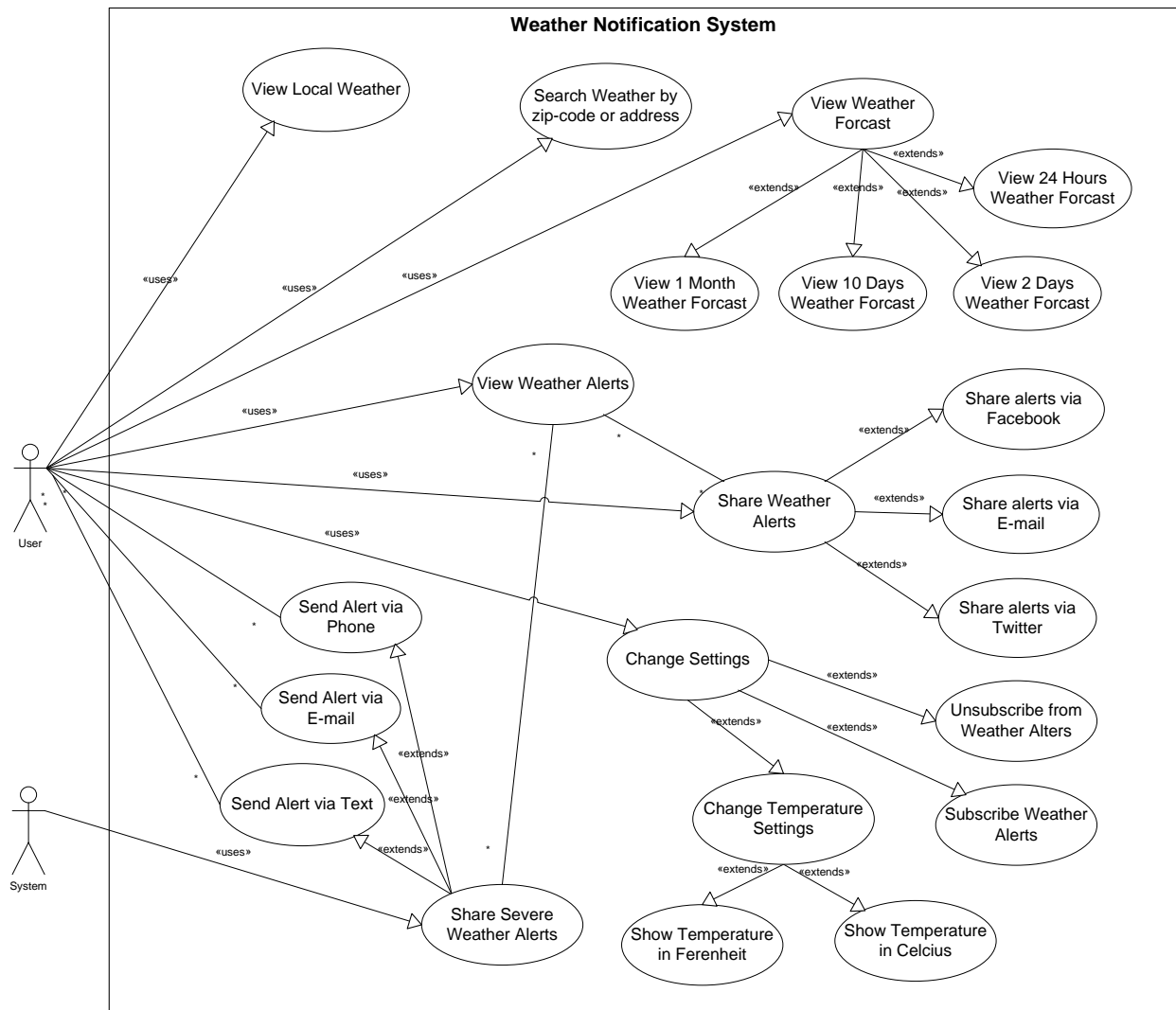
3.3.2. Use Case: Share current weather condition

Use Case Section	Comment
Use Case Name	Share current weather conditions with family and friends
Scope	Weather Notification system for 3G smartphone
Level	User-goal
Primary Actor	End user
Stakeholders and Interests	End user: <ul style="list-style-type: none"> ○ Wants to share weather conditions Notional Weather Service: <ul style="list-style-type: none"> ○ Provide weather data to the system Text Service Provider : <ul style="list-style-type: none"> ○ Provides text service to user Email Service Provider: <ul style="list-style-type: none"> ○ Provides email service to user Facebook: <ul style="list-style-type: none"> ○ Provides facebook service to user
Preconditions	<ul style="list-style-type: none"> ○ There should be an internet connection to connect cell phone with the server ○ User should enter valid location ○ User should enter valid phone number of receiver ○ User should have Facebook account and email id
Success Guarantee	<ul style="list-style-type: none"> ○ Current weather condition for the specific location is shared either by email, text or facebook post
Main Success Scenario	<ul style="list-style-type: none"> ○ User starts the app to check the weather ○ User enters the desired location ○ System fetches the weather details from server for the desired location ○ System displays the weather details ○ User gets the current weather conditions ○ User shares weather details via email, text or facebook
Extensions	At any time, internet connection fails: 1. If the Wi-Fi connection fails the system will try to acquire the 3G internet connect.
Special Requirements	<ul style="list-style-type: none"> ○ Application should response within 20 secs
Frequency of Occurrence	Depends on users use
Miscellaneous	-

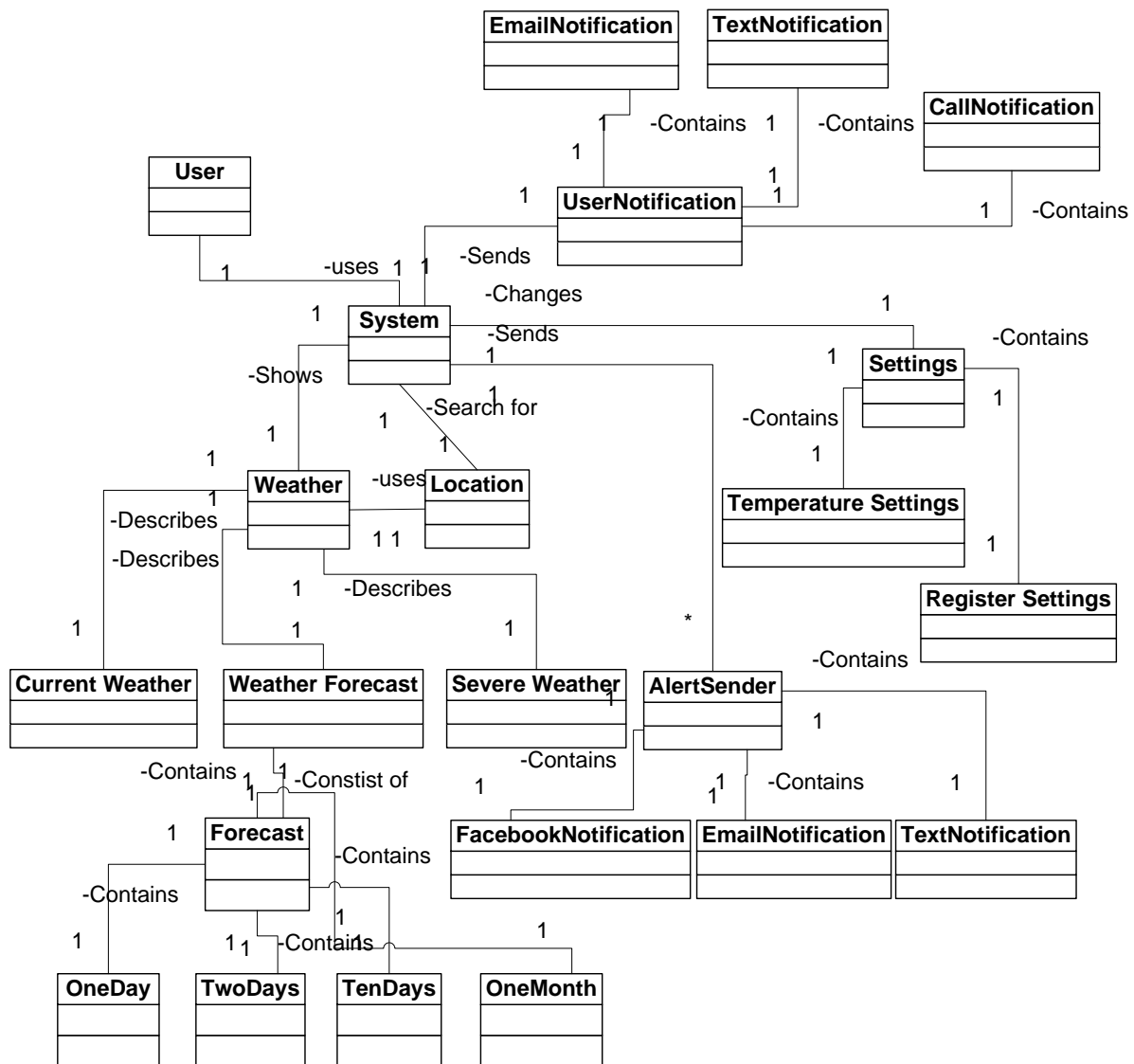
3.3.3. Notify weather alerts

Use Case Section	Comment
Use Case Name	Notify weather alters to the subscribed users
Scope	Weather Notification system for 3G smartphone
Level	System-goal
Primary Actor	System
Stakeholders and Interests	<p>System:</p> <ul style="list-style-type: none">○ Notify subscribed users for severe weather alerts <p>End user:</p> <ul style="list-style-type: none">○ Wants to receive severe weather alerts <p>Notional Weather Service:</p> <ul style="list-style-type: none">○ Provide weather data to the system <p>EAS:</p> <ul style="list-style-type: none">○ Provides severe weather alerts <p>Call / Text Service Provider :</p> <ul style="list-style-type: none">○ Provides call / text service to user <p>Email Service Provider:</p> <ul style="list-style-type: none">○ Provides email service to user
Preconditions	<ul style="list-style-type: none">○ User should be subscribed to get severe weather alerts.○ User should have ongoing cell phone service○ User should enter valid phone number during subscription○ User should have registered valid email id
Success Guarantee	<ul style="list-style-type: none">○ Subscribed user is notified with severe weather alerts
Main Success Scenario	<ul style="list-style-type: none">○ If the server weather conditions occur in subscribed users area system send alters to the user via call, email or text.
Extensions	-
Special Requirements	
Frequency of Occurrence	Depends on users use
Miscellaneous	-

4. Use Case Diagram

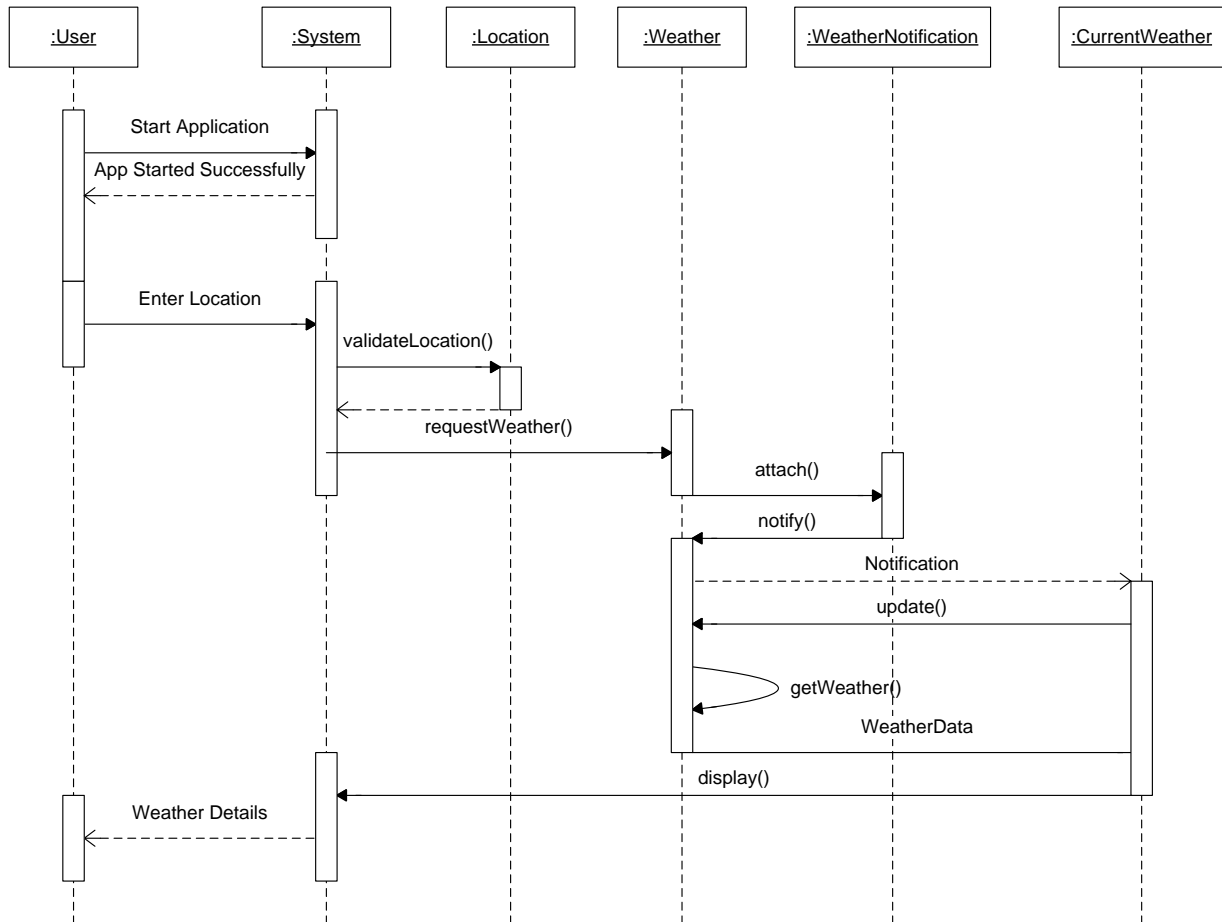


5. Domain Model

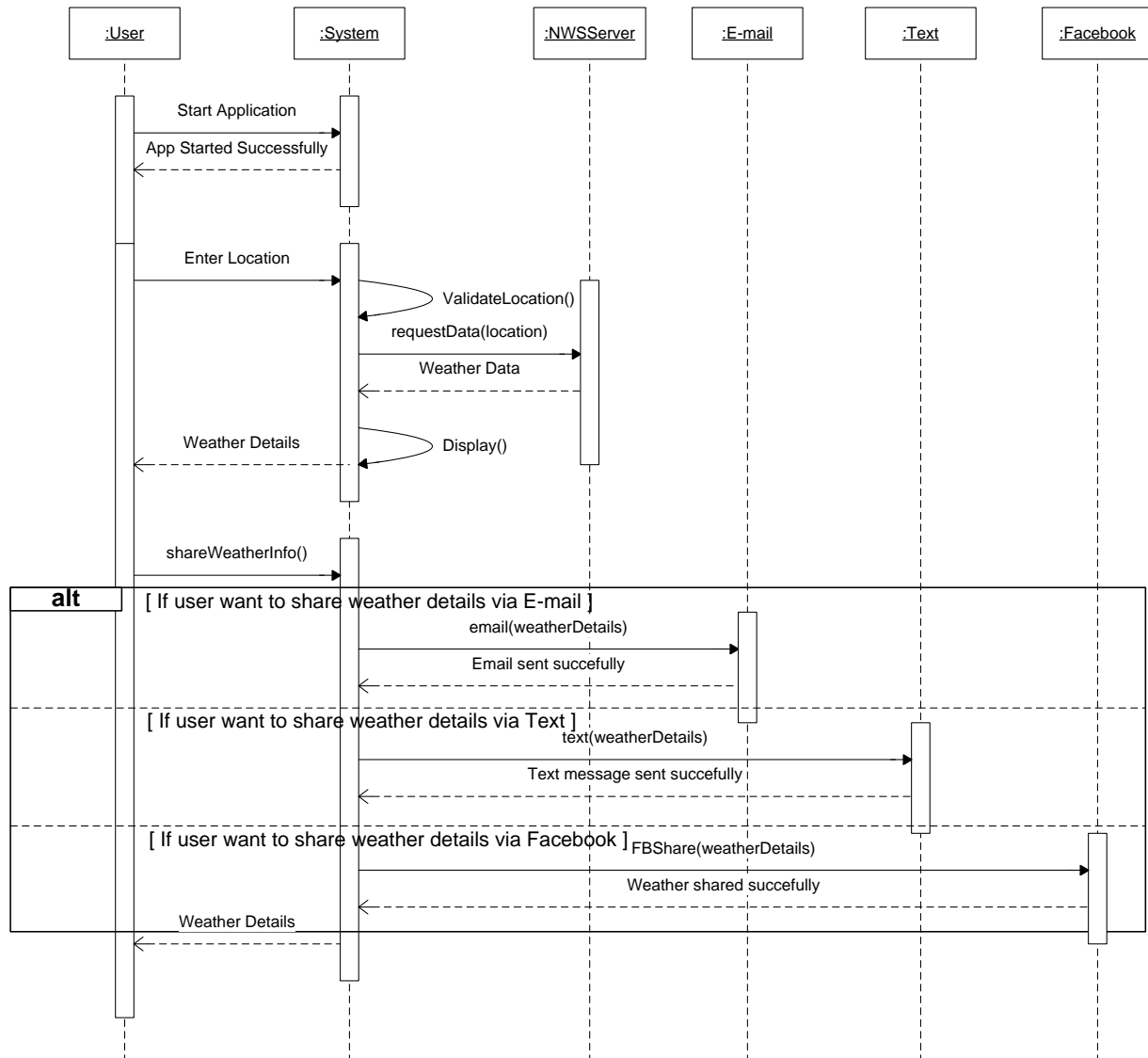


6. Sequence Diagrams

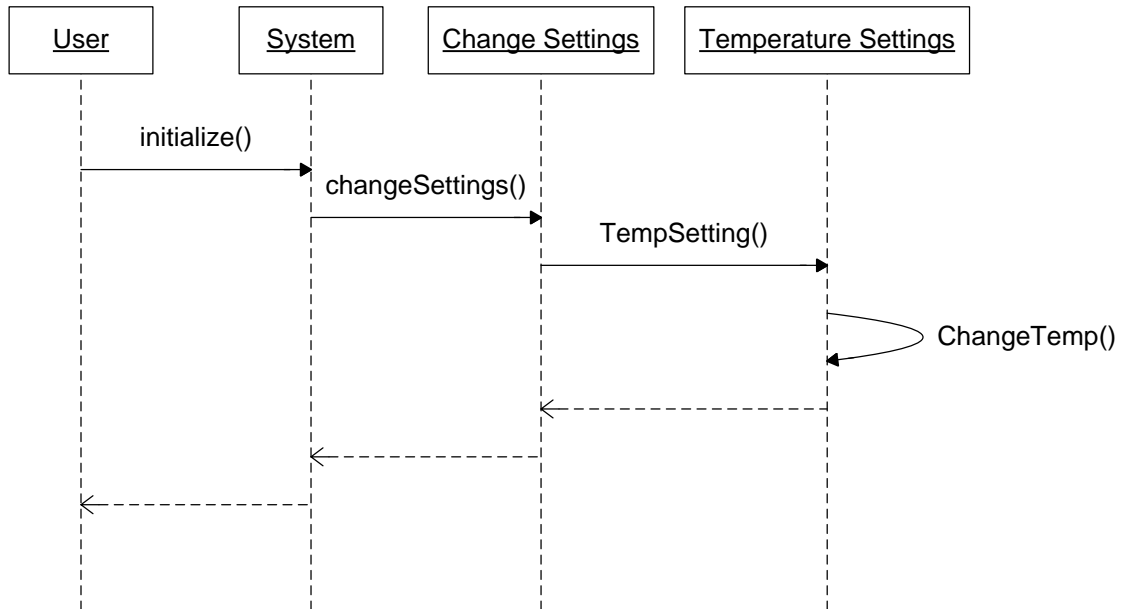
6.1. Sequence Diagram for view current weather condition



6.2. Sequence Diagram for sharing weather condition

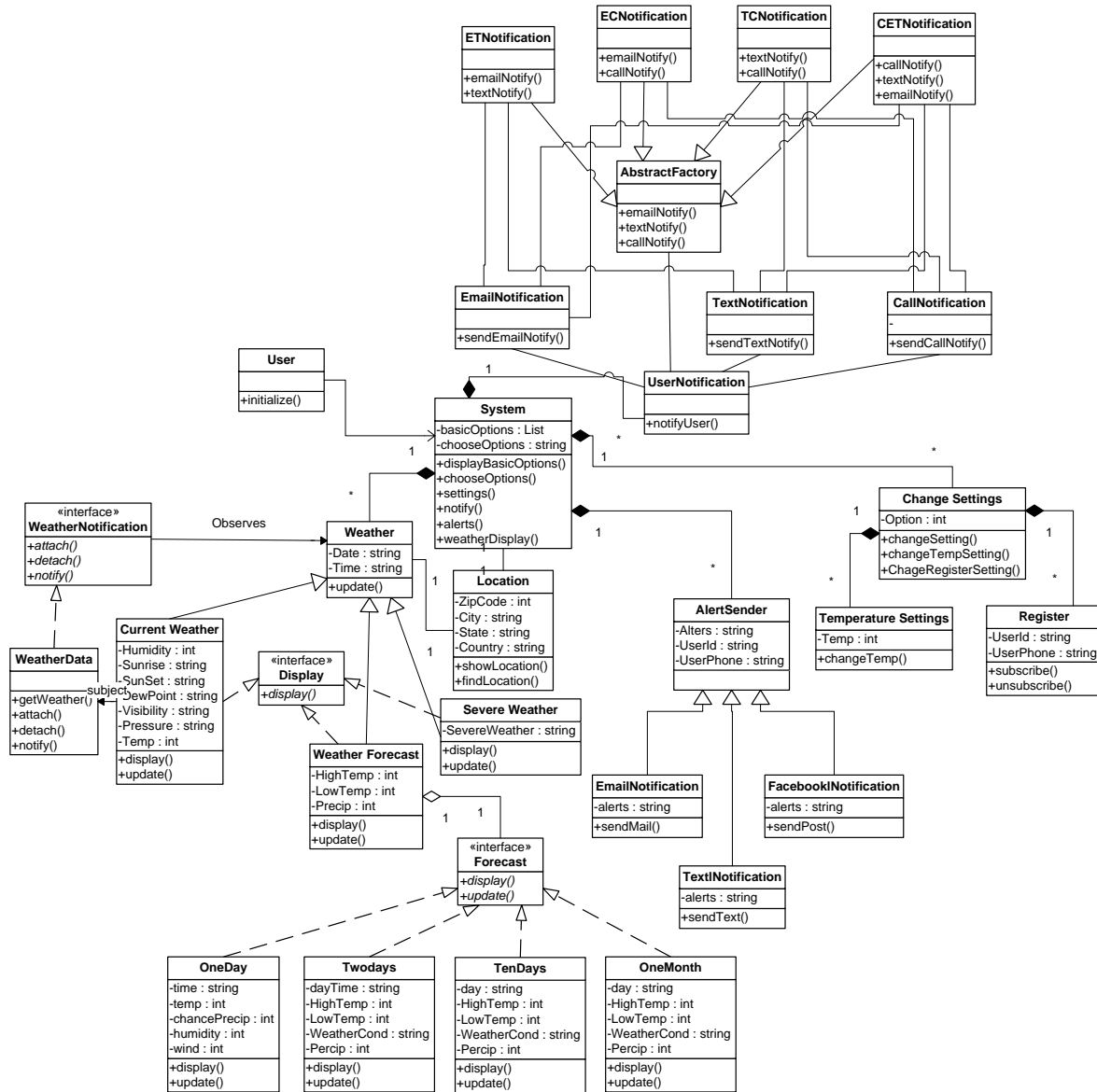


6.3. Sequence Diagram for Change temperature settings



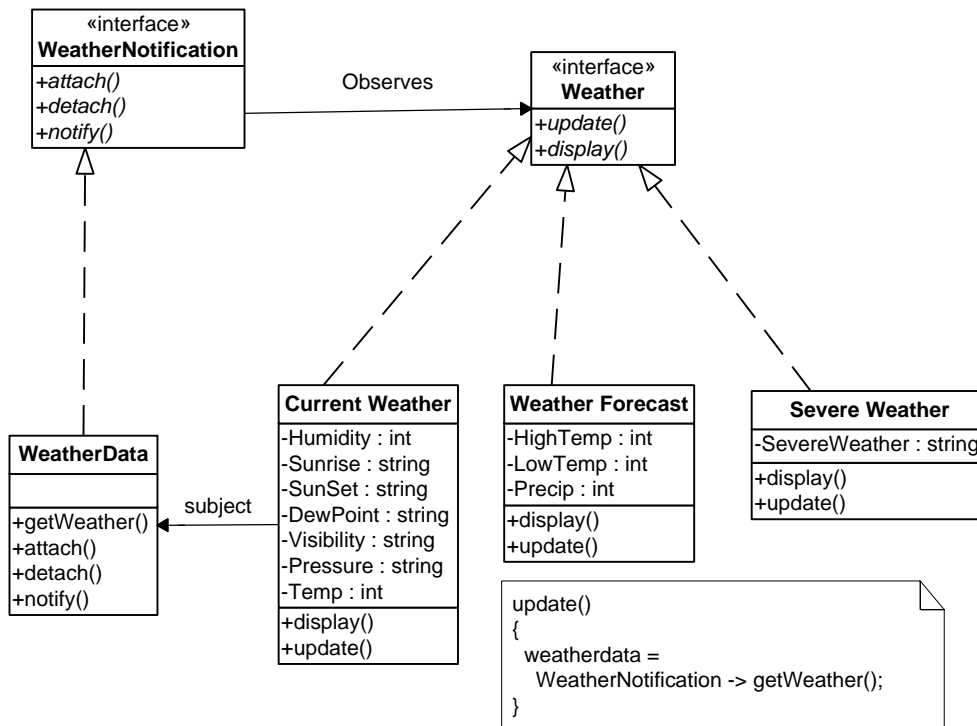
7. Design Model

7.1. Class Diagram



8. Design Patterns

8.1. Observer Pattern



Type: Behavioral Pattern

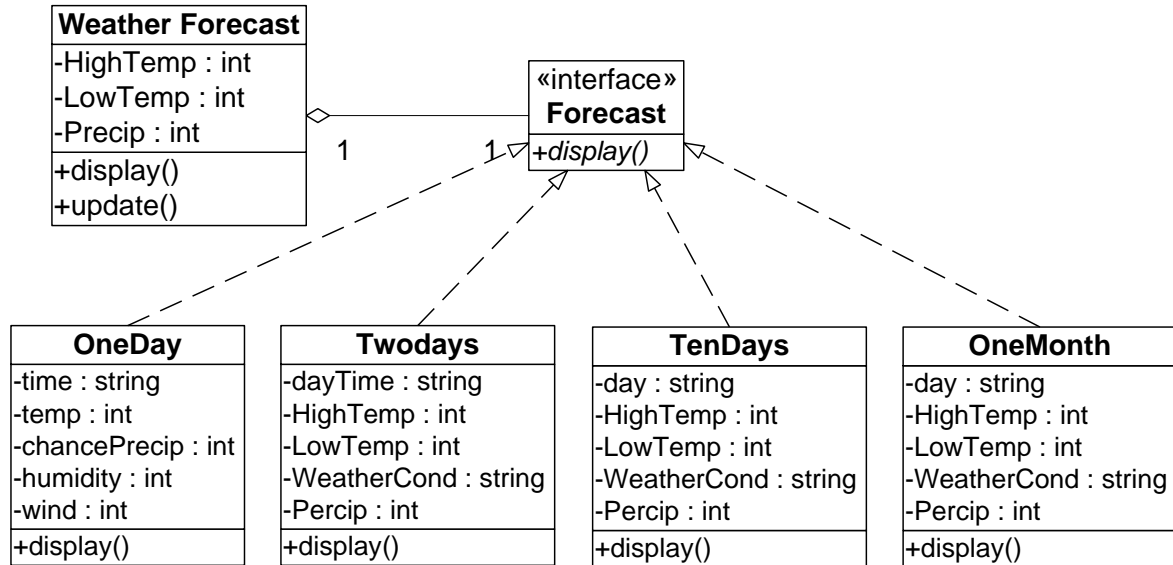
Goal:

- Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.
- Encapsulate the core components in a Subject abstraction, and the variable components in an Observer hierarchy.

Functionality:

- Application has the weather functionality, which has `getWeather()` method that fetch the weather data from the National Weather Service.
- We need to implement 3 display elements, Current Weather, Weather Forecast or Severe Weather. These displays must be updated whenever new weather data is available.

8.2. Strategy Pattern



Type: Behavioral Pattern

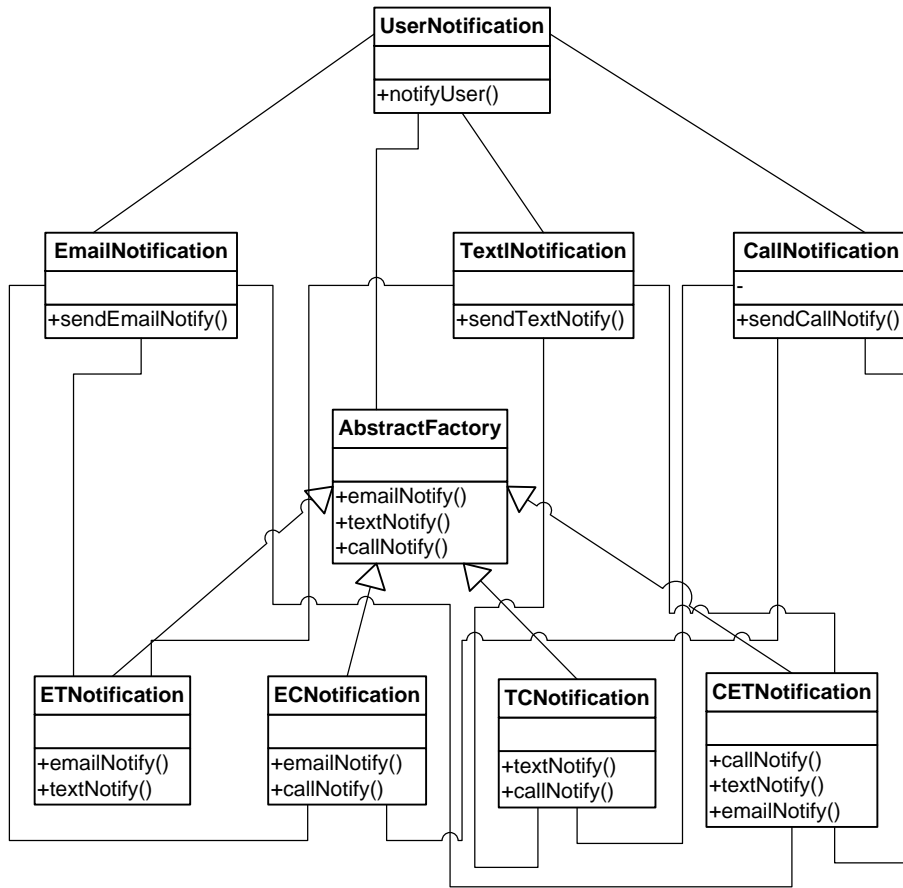
Goal:

- Define a family of algorithms, encapsulate each one, and make them interchangeable. Strategy lets the algorithm vary independently from the clients that use it.
- Capture the abstraction in an interface; bury implementation details in derived classes.

Functionality:

- Application requirements state that user should be able to view the weather forecast for 24 hours, 2 days, 10 days or a month.
- Weather forecast class will display the weather forecast details as per user requirement by choosing the appropriate algorithm from Oneday, Twodays, Tendays and Onemonth.

8.3. Abstract Factory



Type: Creational Pattern

Goal:

- Provide an interface for creating families of related or dependent objects without specifying their concrete classes.
- A hierarchy that encapsulates: many possible “platforms”, and the construction of a suite of “products”.

Functionality:

- In case of severe weather alerts, system should send notifications to subscribed users.
- User notification sends different types of notifications depending upon the types subscribed by the user.
- All notifications should have certain properties in common so that they can work with user notification. These specifications for the notifications are the abstract factories.

- So as per the user requirements, each factory will act and notify the user by the way user subscribed.