Calculating daily costs

You are tasked with parsing data from an e-commerce store API. As part of processing the data there is a need for a function “**daily_cost**” which will calculate costs on a daily basis. The function should accept these parameters:

- A time period (**start_date**, **end_date**)
- A **list of time interval and cost pairs**
  - Time intervals can be one of: “daily, weekly, monthly”
  - Cost is a **positive** decimal

It should return an array of [date,cost] pairs for each day in the specified time period (start_date, end_date) inclusive of dates given. For a daily cost just add that cost for that day. For the weekly cost add 1/7 of the cost for each day and for the monthly cost add 1/Nth cost where N is the number of days in the current month.

You need to:

- Write a function **daily_costs** as described above
- Provide unit tests which check that the function works for
  - A single and for multiple individual periods
  - Mixed daily, weekly and monthly periods
  - Returns an error for wrong time interval given
  - Returns an error for wrong cost and/or wrong time period values

Example:

Below is an example of a time period of 3 days and two costs. A daily cost of 10 units and a weekly cost of 70 units. When run, the function returns a hash of each day and that days cost.

```
all_time_periods = %w(daily weekly monthly)

# time interval
start_date = Date.new(2019, 10, 1)
end_date = Date.new(2019, 10, 3)

time_period_costs = [
   {time_period: all_time_periods[0], # daily
    cost: 10.0
   },
   {time_period: all_time_periods[1], # weekly
    cost: 70.0
   }
]

>> daily_cost(start_date, end_date, time_period_costs)

[{:date=>Tue, 01 Oct 2019, :cost=>20.0}, {:date=>Wed, 02 Oct 2019, :cost=>20.0}, {:date=>Thu, 03 Oct 2019, :cost=>20.0}]
```