C# Developer Exercise: Train ticket machine

You are asked to write code to support the user interface of a train ticket machine.

You will not be writing any actual UI code, but are asked to develop a search algorithm to help the user entering the name of a station.

The machine has a touchscreen display which works as follows. As the user types each character of the station's name on the touchscreen, the display should update to show all valid choices for the next character and a list of possible matching stations.

The illustration below shows what is needed when 'D A R T' has been entered.

User input: **D A R T** __

A	B	C	D	E		**DARTFORD**
F	G	H	I	J		**DARTMOUTH**
K	L	**M**	N	O		
P	Q	R	S	T		
U	V	W	X	Y		
Z						

The requirements are:

- Typing a search string will return
    - All stations that start with the search string
    - All valid next characters for each matched station.
- Runtime speed is very important
- A space is a valid character when returning a list of next characters
- You don't need to go overboard with your station list in your tests. A small enough list of stations to adequately test each condition will suffice

Not required:

- A fast loading time is not required at start-up, runtime performance takes priority
- This will be run on a dedicated machine designed for the purpose
- The application will be used by a single user at a time. There's no need to code for concurrency
- No code is required for reading the stations from a data store. You may stub the station list or mock a station reader in your tests, whichever you feel represents the best real world solution.

Examples:

- Given the input 'DART' and a list of stations 'DARTFORD', 'DARTMOUTH', 'TOWER HILL', 'DERBY' the application should return next characters of 'F', 'M' and the stations 'DARTFORD', 'DARTMOUTH'.
- Given the input 'LIVERPOOL' and a list of stations 'LIVERPOOL', 'LIVERPOOL LIME STREET', 'PADDINGTON' the application should return next characters of ' ' and the stations 'LIVERPOOL', 'LIVERPOOL LIME STREET'
- Given the input 'KINGS CROSS' and a list of stations 'EUSTON', 'LONDON BRIDGE', 'VICTORIA' the application will return no next characters and no stations

We would like to see the following, most important first:

- Code quality (readable code; suitable unit tests; following OO design principals; use of appropriate language features)
- Good runtime performance (it is very important that it runs quickly. Start-up time is less important.)
- A complete solution, fulfilling all the requirements