```java
package st;

import java.util.ArrayList;

public class EntryMap {

    private ArrayList<Entry> entries;

    private HashSet<Entry> uniqueEntries;

    public EntryMap(){
        entries = new ArrayList<>();
        uniqueEntries = new HashSet<>();
    }

    public void store(String pattern, String value, Boolean caseSensitive) throws RuntimeException{
        if (caseSensitive == null){
            caseSensitive = Boolean.FALSE;
        }
        Entry entry = new Entry(pattern, value, caseSensitive);
        if (!isEntryValid(entry)){
            throw new RuntimeException();
        }

        if (isEntryUnique(entry)){
            addEntry(entry);
        }
    }

    private Boolean isEntryValid(Entry entry){
        if (entry.getPattern()== null)
            return Boolean.FALSE;
        if (entry.getPattern().isEmpty())
            return Boolean.FALSE;
        if (entry.getValue() == null)
            return Boolean.FALSE;
        return Boolean.TRUE;
    }

    private Boolean isEntryUnique(Entry entry){
        return !uniqueEntries.contains(entry);
    }

    private void addEntry(Entry entry){
        entries.add(entry);
        uniqueEntries.add(entry);
    }

    public ArrayList<Entry> getEntries() {
        return entries;
    }

```

```java
    class Entry {
        String pattern;
        String value;
        Boolean caseSensitive;

        public Entry(String pattern, String value, Boolean caseSensitive) {
            this.pattern = pattern;
            this.value = value;
            this.caseSensitive = caseSensitive;
        }

        public String getPattern() {
            return pattern;
        }

        public String getValue() {
            return value;
        }

        public Boolean getCaseSensitive() {
            return caseSensitive;
        }

        @Override
        public boolean equals(Object o) {
            if (this == o) return true;
            if (o == null || getClass() != o.getClass()) return false;

            Entry entry = (Entry) o;

            if (!getPattern().equals(entry.getPattern())) return false;
            if (!getValue().equals(entry.getValue())) return false;
            return getCaseSensitive() != null ? getCaseSensitive().equals(entry.getCaseSensitive()) : entry.getCaseSensitive() == null;
        }

        @Override
        public int hashCode() {
            int result = getPattern().hashCode();
            result = 31 * result + getValue().hashCode();
            result = 31 * result + (getCaseSensitive() != null ? getCaseSensitive().hashCode() : 0);
            return result;
        }
    }

}

```