

## PROBLEM STATEMENT

## **PROBLEM**

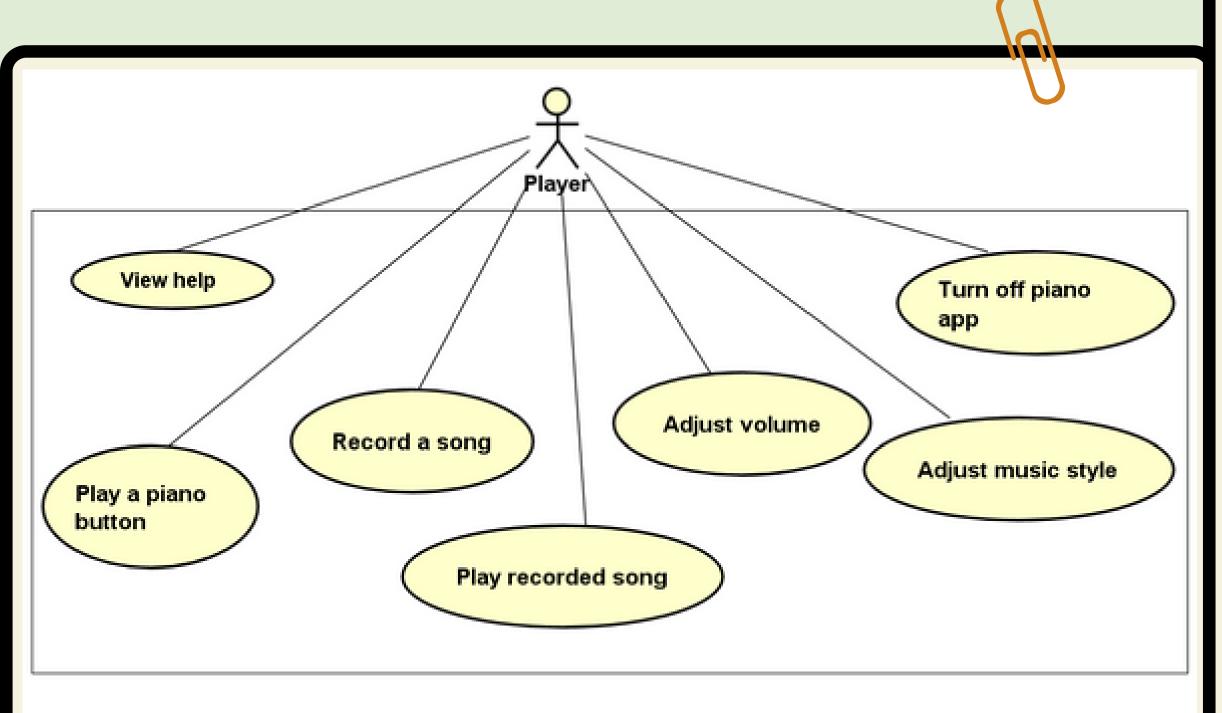
Implement an application that provides GUI for the user to virtually play an electronic piano.



## **DESIGN**

Design a UI with keyboard, volume, record, music style, ... that comfort users.

# USE CASE DIAGRAM



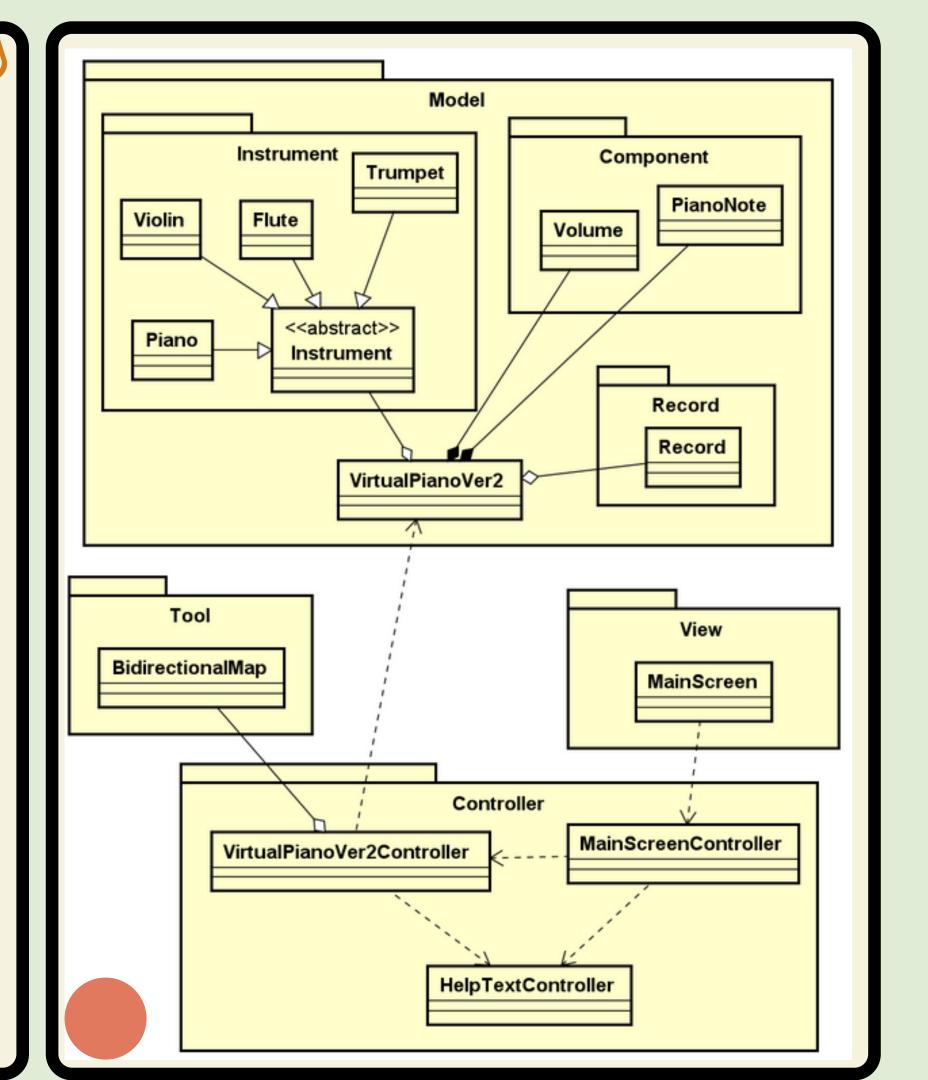
User can start to play by clicking, pressing or editing the song themself

- Play a note whenever user touches.
- Notice the user if a key on keyboard cannot play.
- Replay everything we've just played.
- Adjust volume, music style for different experiences.

User can view help menu.
User can confirm to exit
from the program.

# CLASS DIAGRAM

- Package "Model": store a complete piano
- Package "Controllers": store all the screen controllers.
- Package "View": store the "MainScreen" class of the application.
- Package "Tool": store key character and corresponding piano note,



#### Controller java.util HashMap MainScreenController + put(key : K, value : V) : V VirtualPianoVer2Controller piano : VirtualPianoVer2 + remove(key : K) : void helpButton, playButton: Button - piano : VirtualPianoVer2 - isRecording, assistanOn, songVisible : boolean + helpButtonPressed(ae : ActionEvent) : instrumentVisible, volumeVisible : boolean + playButtonPressed(ae : ActionEvent) : void buttonMap : BidirectionalMap<String,Button> - octave : byte enableKeys[]: boolean Tool + VirtualPianoVer2Controller(piano : VirtualPianoVer2) + helpButtonPressed(ae : ActionEvent) : void BidirectionalMap + backLabelClicked(me : MouseEvent) : void inversedMap : Map<V,K> + notePressed(e : KeyEvent) : void + noteReleased(e : KeyEvent) : void + getKey(value : V) : K + btnNoteMousePressed(me : MouseEvent) : void + remove(key : K) : void + btnNoteMouseReleased(me : MouseEvent) : void + put(key : K, value : V) : V + decreaseOctaveBtnClicked(me : KeyEvent) : void + valueSet() : Set<V> + increaseOctaveBtnClicked(me : KeyEvent) : void HelpTextController + assistantClicked(me : MouseEvent) : void + volumeBoxClicked(me : MouseEvent) : void - helpText : TextArea + instrumentBoxClicked(me : MouseEvent) : void - line : String + recordBoxClicked(me : MouseEvent) : void + initialize(): void + recordLabelClicked(me : MouseEvent) : void + showHelp(); void + saveRecord(me : MouseEvent) : void + replayPatternClicked(me : MouseEvent) : void + clearPatternClicked(me : MouseEvent) : void + btnPlayRecordPressed(ae : ActionEvent) : void + btnRemoveRecordPressed(ae : ActionEvent) : void + initialize(): void + updateRecordBox(rec : Record) : void + updateKeyLabel(text : String) : void + updateNoteDisplay(): void

#### Model VirtualPlanoVer2 volume : Volume notesMap : Map<String,PianoNote> records : ObservableList<Record> org.jfugue.realtime Instrument instrument : Instrument nbPianoNotes : int = 0 <<abstract>> RealtimePlayer Violin + VirtualPianoVer2() Instrument + startNote(note : Note) : void + VirtualPianoVer2(value : int) + getInstrument(): String name : String + stopNote(note : Note) : void + setUpPiano(volume : Volume) : void instrument id : int addPianoNote(note : PianoNote) : void Plane + Instrument() + getNotesMap(): Map<String,PianoNote> + Instrument(id : int) + getInstrument(): String + increaseVolume(): void + getInstrumentId(): int + decreaseVolume(): void + getInstrument() : String . + getVolume(): int Guitar + getRecords() : ObservableList<Record> + setInstrument(instrument : Instrument) : void + getInstrument(): String + getInstrument(): String + addRecord(rec : Record) : void Flute Trumpet + removeRecord(rec : int) : Record + play(rec : Record) : void + getInstrument(): String + getInstrument() : String + play(i : int) : void org.jfugue.theory Component Record Note PianoNote Volume + getOctave() : byte keyChar : String Record - ID: int value : int - name : String = "No Name" - coarse\_volume : byte pattern : StringBuilder + PianoNote() fine\_volume : byte + PianoNote(note : String, keyChar : String) + Record() + PianoNote(note : String, ID : int, keyChar : String) + Volume() + Record(pattern : String) + setOctave(octave : int) : void + Volume(value : int) + Record(name : String, pattern : String) + increaseOctave(): void + getValue(): int + getName() : String + decreaseOctave(): void + getCoarseVolume() : byte + getPattern() : StringBuilder + getOriginalString(): String + getFineVolume() : byte + getLength(): int + updateCoarseFineVolume() : void + getId(): int + toString(): String + getKeyChar(): String



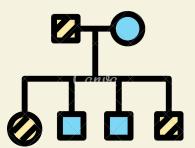
### Encapsulation

- Package Model, class
   VirtualPianoVer2
- Package Controller, class
   VirtualPianoVer2Controller
   use instance of
   BidirectionalMap



### Abstraction

 Package Instrument in package model, 5 class implement abstract method from abstract class Instrument



### Inheritance

- Package Model, class
   VirtualPianoVer2
- Package Controller, class
   VirtualPianoVer2Controller
   use instance of
   BidirectionalMap



### Polymorphism

 Package Instrument in package model, 5 class implement abstract method from abstract class Instrument Demo