Course: CS522 - Software Quality Assurance and Test Automation
Faculty: Prof. Henry Chang

Project: Three UML diagrams for JukeBox

Name: Haripriya A
ID: 19579

NORTHWESTERN POLYTECHNIC
UNIVERSITY

Table of Contents

Introduction

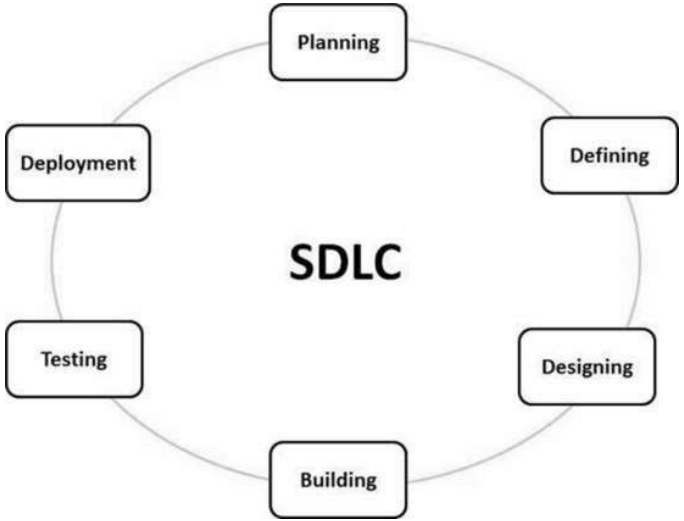
- Software Testing can be started from the Requirements Gathering phase and continued till the deployment of the software.
- For Software Quality Assurance and Test Automation one should have basic understanding of below
  - Software Development Life Cycle (SDLC).
  - Software programming using any programming language.

About Software Development Life Cycle

- Software Development Life Cycle depends on the development model that is being used.
  - In the Waterfall model, formal testing is conducted in the testing phase
  - In the incremental model, testing is performed at the end of every increment/iteration and the whole application is tested at the end.
- Testing is done in different forms at every phase of SDLC:
  - During the requirement gathering phase, the analysis and verification of requirements are also considered as testing.
  - Reviewing the design in the design phase with the intent to improve the design is also considered as testing.
  - Testing performed by a developer on completion of the code is also categorized as testing.

About Software Development Life Cycle

- The life cycle defines a methodology for improving the quality of software and the overall development process. Below is the SDLC Cycle stages
  - Stage 1: Planning and Requirement Analysis
  - Stage 2: Defining Requirements
  - Stage 3: Designing the product architecture
  - Stage 4: Building or Developing the Product
  - Stage 5: Testing the Product
  - Stage 6: Deployment in the Market and Maintenance



UML

- A good policy and process for the software development effort is necessary
- Mechanisms for capturing design are needed
  - Unified Modeling Language (UML).
- The UML is the collective brainchild of Grady Booch, Ivar Jacobson, and Jim Rumbaugh; three world renown object-oriented technologists who combined their efforts to create a standard method for capturing and visualizing object-oriented designs.
- UML has been one of those buzzwords in the software community that has found itself on the same hot list as Java, XML, and .NET. UML stands as a powerful and capable method for designing software. Unfortunately, it is vastly underutilized within the software development community.

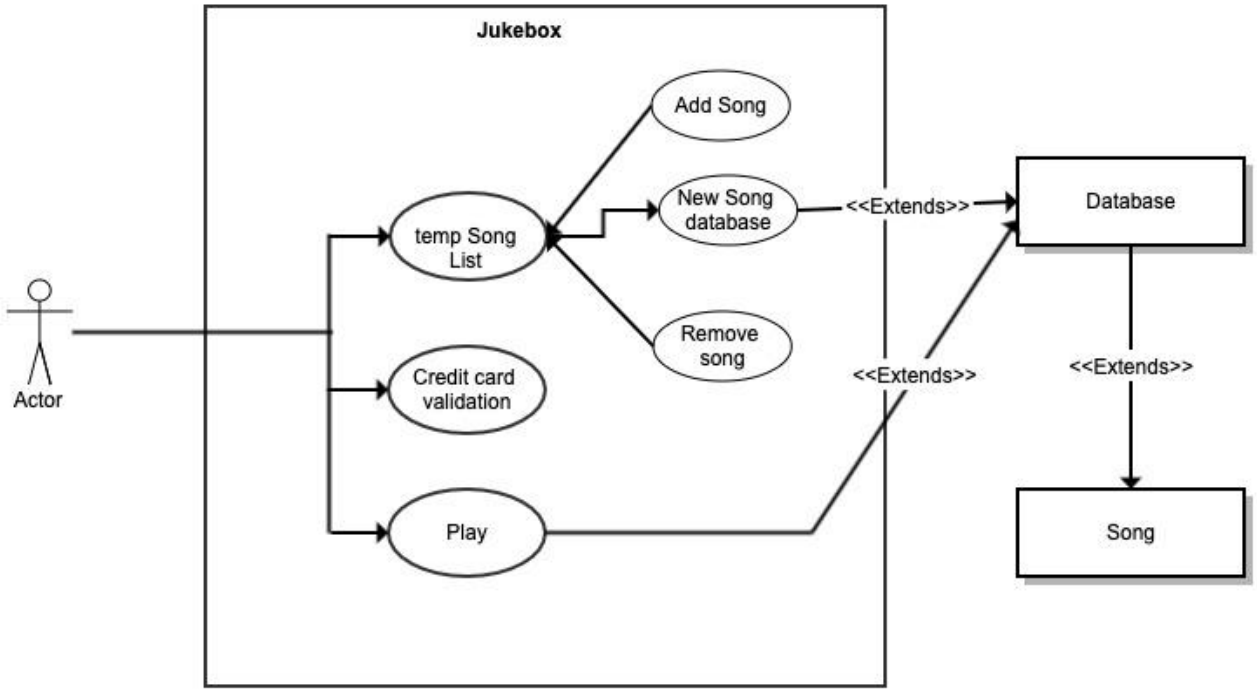
Implementation of JukeBox

- Planning: Our goal is to design a JukeBox that allows customers to select songs they want played or to submit a playlist that they have already created previously.
- Requirements: JukeBox spec
  - Allow customers to
    - select songs they want to play.
    - submit a playlist that they have already created previously.
  - To provide a mechanism for owners, record companies and artists to earn a profit. The Juke Box contains
    - A card swipe mechanism
- Design: UML Diagrams

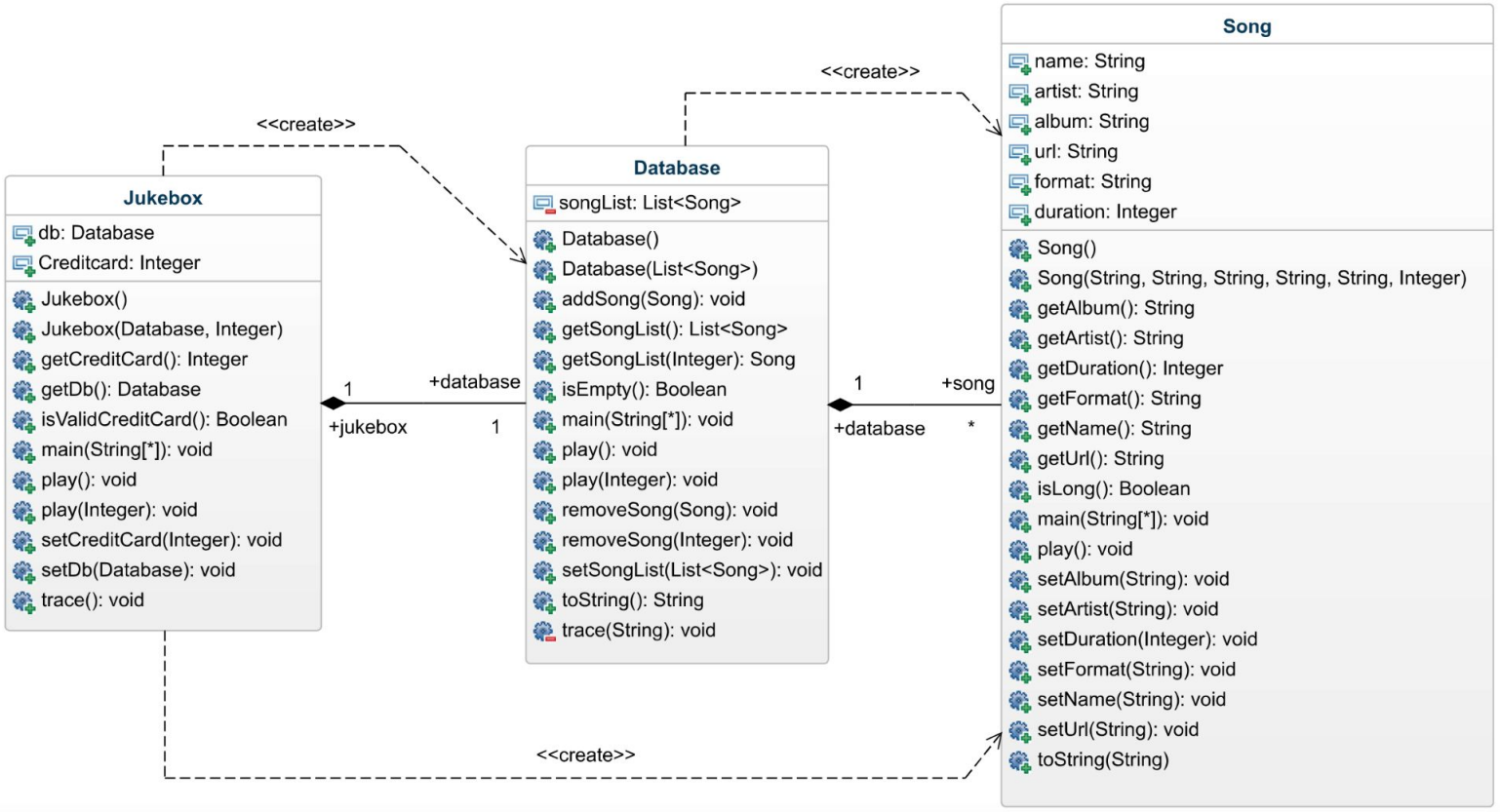
Jukebox SDLC

- Planning: Our goal is to design a JukeBox that allows customers to select songs they want played or to submit a playlist that they have already created previously.
- Requirements: JukeBox spec
  - Allow customers to
    - select songs they want to play.
    - submit a playlist that they have already created previously.
  - To provide a mechanism for owners, record companies and artists to earn a profit. The Juke Box contains
    - A card swipe mechanism
- Design: JukeBox UML

Use Case Diagram

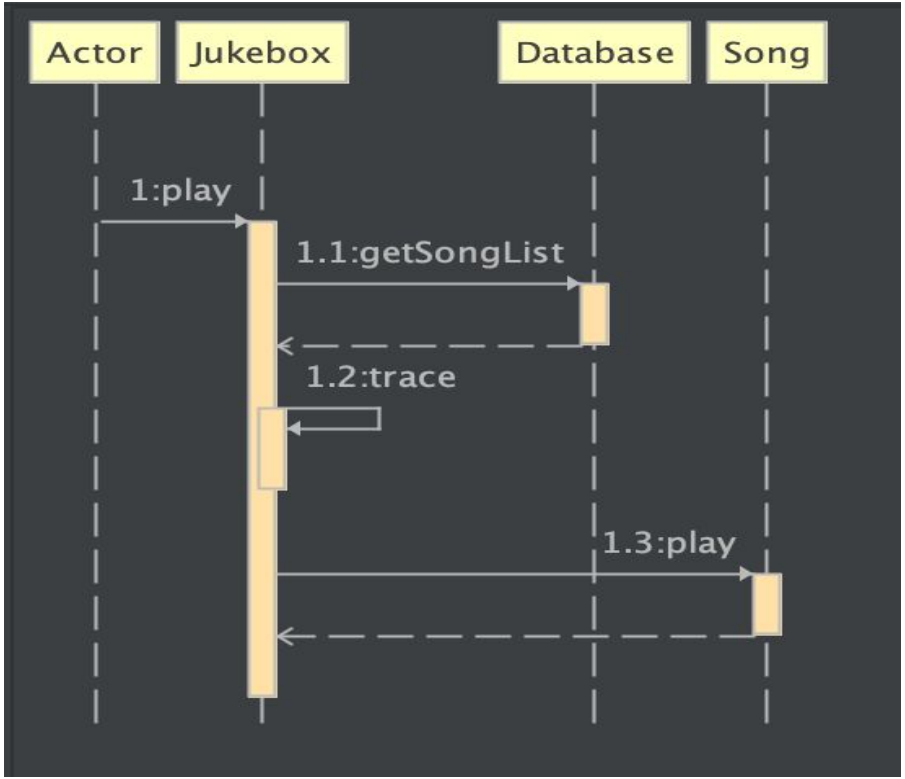


Class Diagram



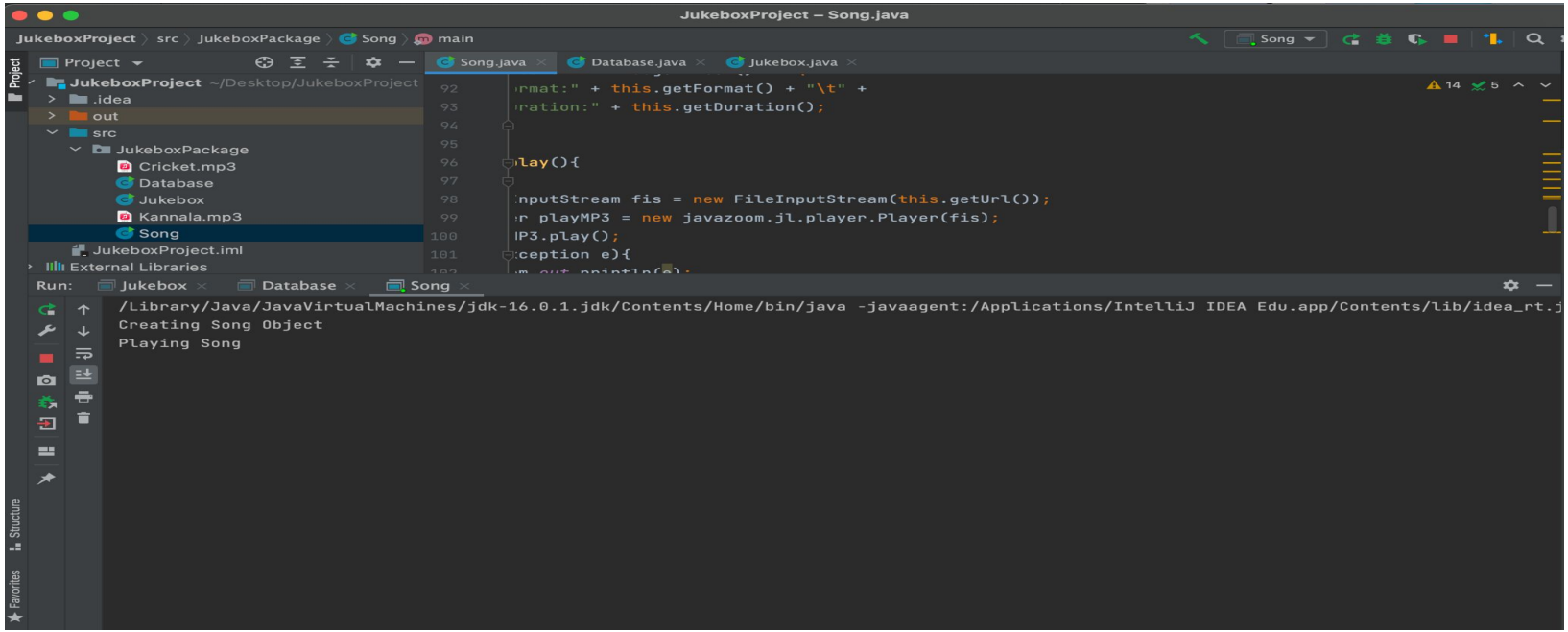
Sequence Diagram

A Sequence Diagram is a powerful visual aid for developers in understanding the interaction of objects within a system



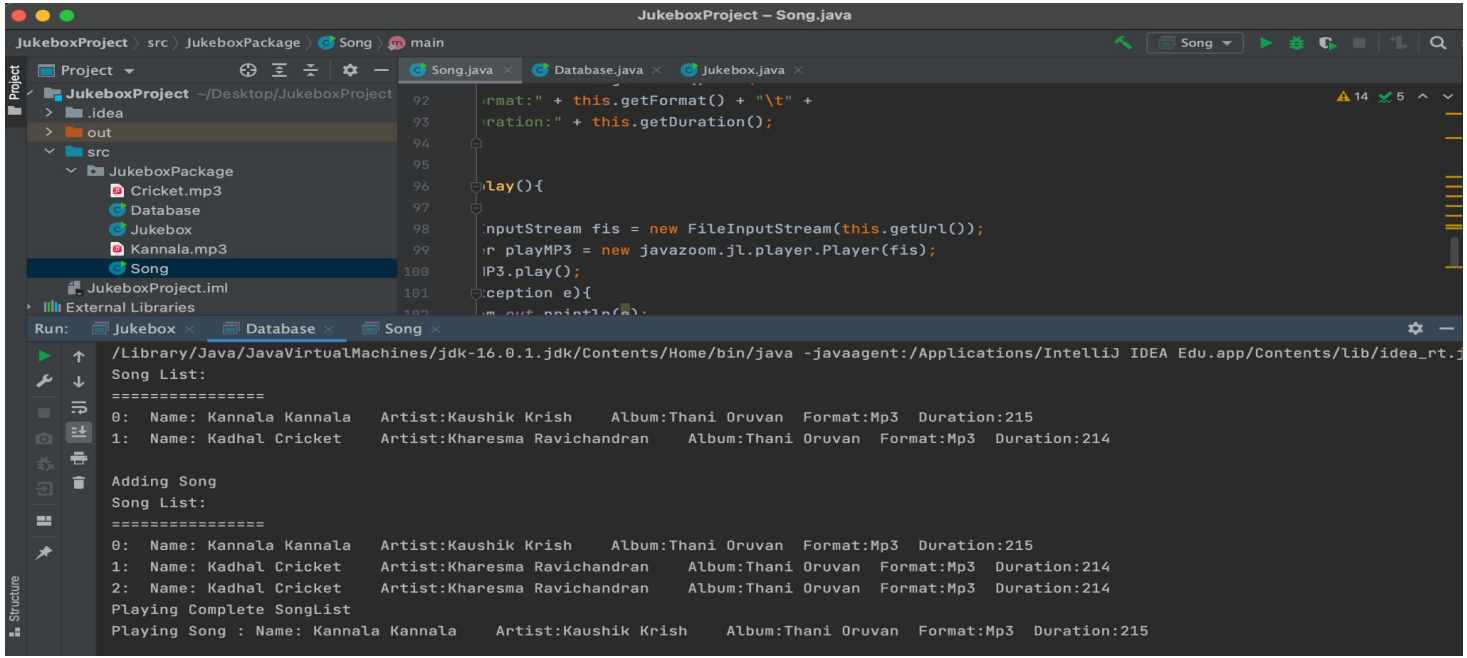
Program Output

- Song.java



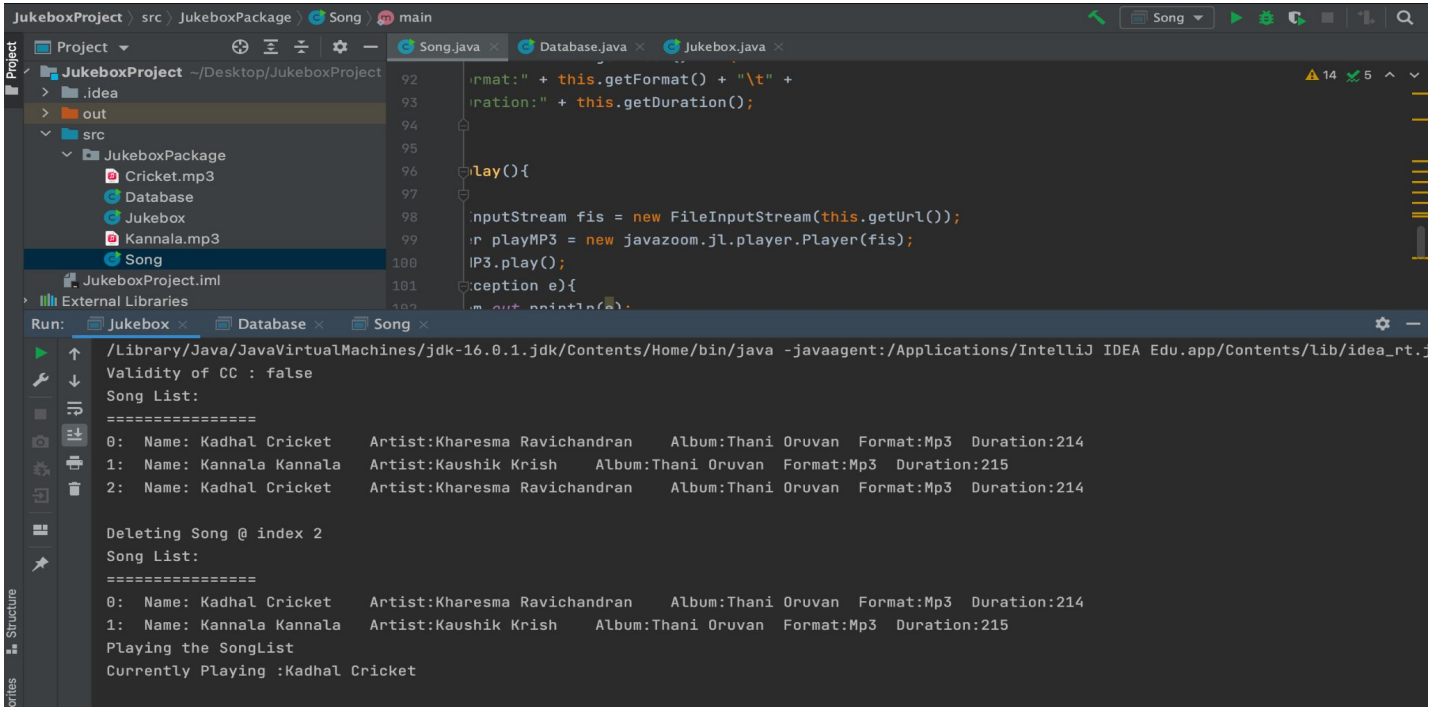
Program Output

- Database.java



Program Output

- Jukebox.java



Conclusion

- Hence the Jukebox UML - use case, class and sequence diagram has been shown in the above slides which is part of the Jukebox design.

References

- https://npu85.npu.edu/~henry/npu/classes/qa/sdlc_tutorialspoint/slide/index_slide.html
- https://npu85.npu.edu/~henry/npu/classes/qa/qa_tutorialspoint/slide/index_slide.html
- https://npu85.npu.edu/~henry/npu/classes/oo/uml_tutorial/slide/index_slide.html

Google slides URL :

https://docs.google.com/presentation/d/1jqi0RkhkyCa-nfRoOJWqFgxJqdIm9Gx9WZ8I-H7MYls/edit?usp=sharing

GitHub:
https://github.com/HaripriyaReddy880/Software-Quality-Assurance-and-Test-Automation/tree/main/Software%20Development%20Life%20Cycle%20(SDLC)/UML/JukeBox