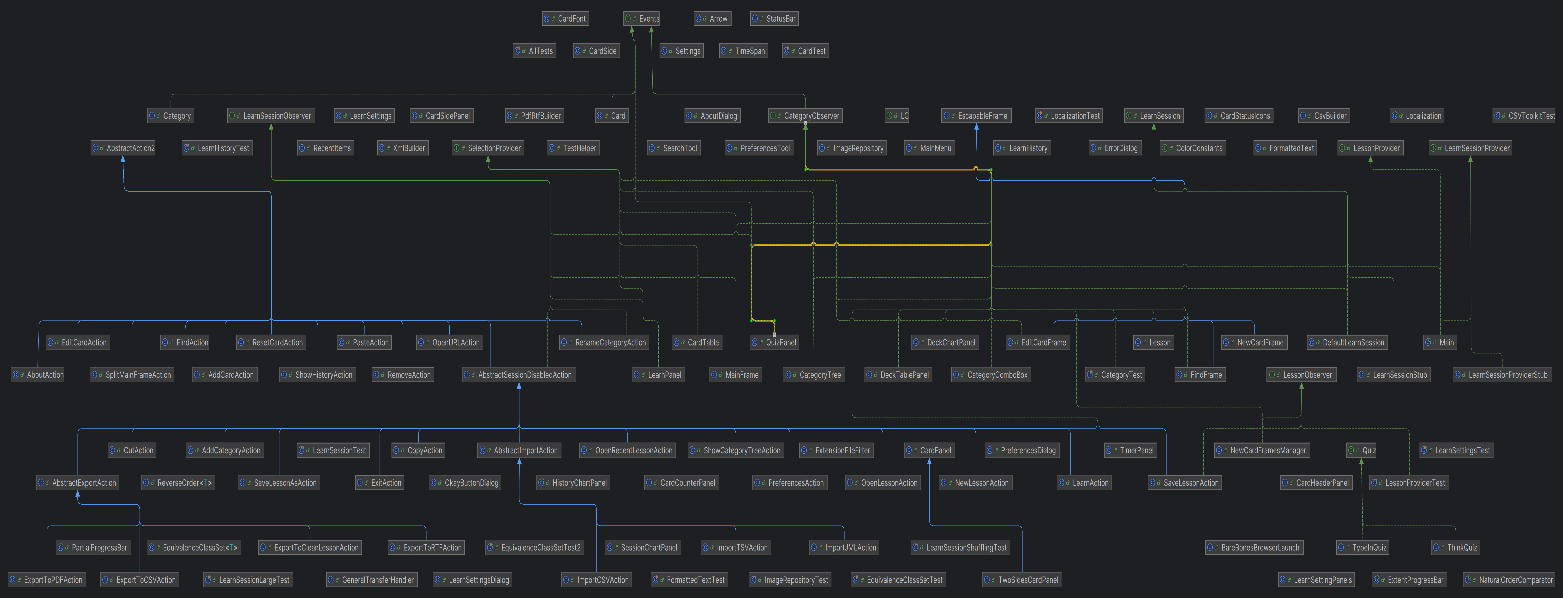
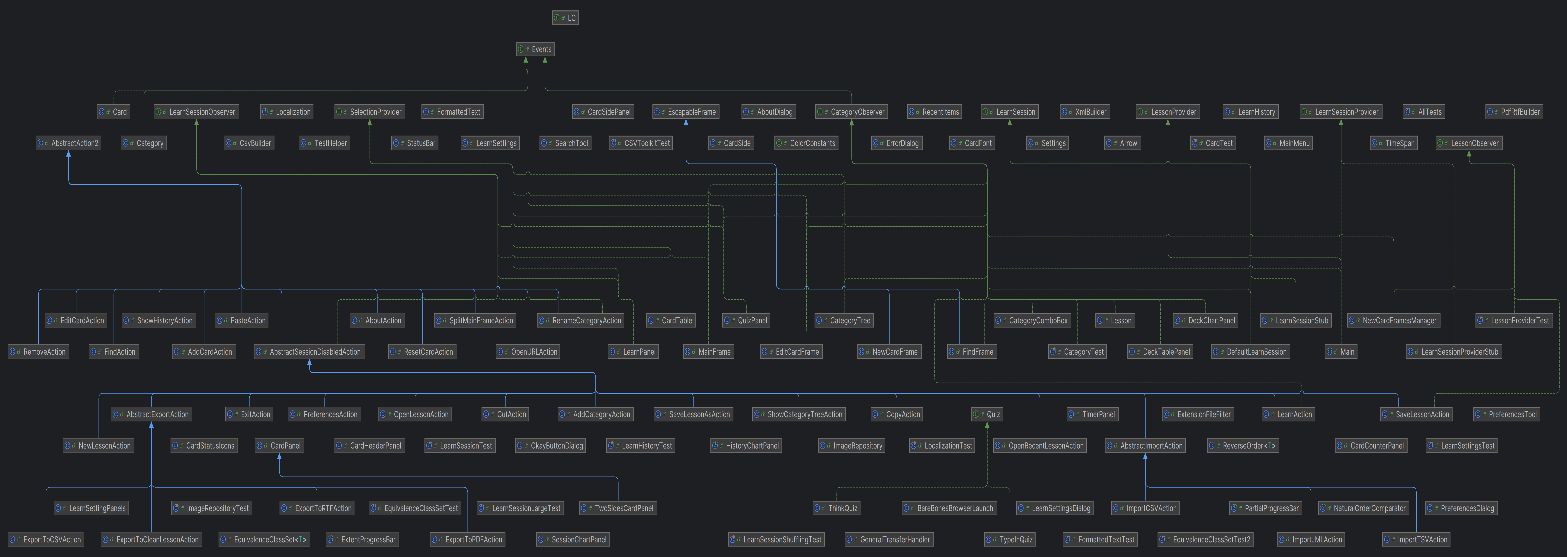
*The UML Diagram before the changes:*



*The UML Diagram after the changes:*



*The changes in the project:*

1. **Copy constructor change instead of clone in CardSide class**: public CardSide(CardSide other)
2. **appling copy constructor instead of clone**: public Card cloneWithoutProgress()
3. **todo move this back into decode**: private void decode(StyledDocument doc, String text, int offset) throws BadLocationException
4. **change list of images icon into a list of ids**: public List<String> getImageIDs(List<ImageIcon> images)

*Feature Implementation :*

import java.util.\*;

public class jMemorizeWithSRA {

// Define a class to represent flashcards

class Flashcard {

String question;

String answer;

int interval; // Time until next review

int repetitions; // Number of times reviewed

Date nextReviewDate; // Date of next review

public Flashcard(String question, String answer) {

this.question = question;

this.answer = answer;

this.interval = 1; // Initial interval

this.repetitions = 0; // Initial repetitions

this.nextReviewDate = new Date(); // Initial date

}

}

List<Flashcard> flashcards;

Random random;

public jMemorizeWithSRA() {

flashcards = new ArrayList<>();

random = new Random();

}

// Method to schedule the next review based on SM-2 algorithm

public void scheduleNextReview(Flashcard card, boolean correct) {

if (correct) {

card.repetitions++;

if (card.repetitions == 1) {

card.interval = 1;

} else if (card.repetitions == 2) {

card.interval = 6;

} else {

card.interval \*= 2.5; // Increase interval

}

} else {

card.repetitions = 0; // Reset repetitions if incorrect

card.interval = 1; // Restart interval

}

// Calculate next review date

long now = System.currentTimeMillis();

card.nextReviewDate = new Date(now + (long) (card.interval \* 24 \* 60 \* 60 \* 1000));

}

// Method to simulate reviewing a flashcard

public void reviewFlashcard(Flashcard card, boolean correct) {

scheduleNextReview(card, correct);

// Optionally, update flashcard in the database or GUI

}

// Method to simulate studying flashcards

public void study() {

for (Flashcard card : flashcards) {

// Simulate user interaction, e.g., displaying the question and receiving input

boolean correct = /\* user input \*/;

reviewFlashcard(card, correct);

}

}

public static void main(String[] args) {

jMemorizeWithSRA jMemorize = new jMemorizeWithSRA();

// Add flashcards to jMemorize

// e.g., jMemorize.flashcards.add(new Flashcard("Question", "Answer"));

// Simulate studying flashcards

jMemorize.study();

}

}