**Data Flow Diagram and Structure Chart**

**Team Members (C2\_6):**

Neeraj Kumar– 57

Anish Sahu– 56

Aarav Bhandari– 47

Anup Pai– 45

**DFD on our project named Song Recommendation System:**







**Data Dictionary for above DFD:**

userid : integer

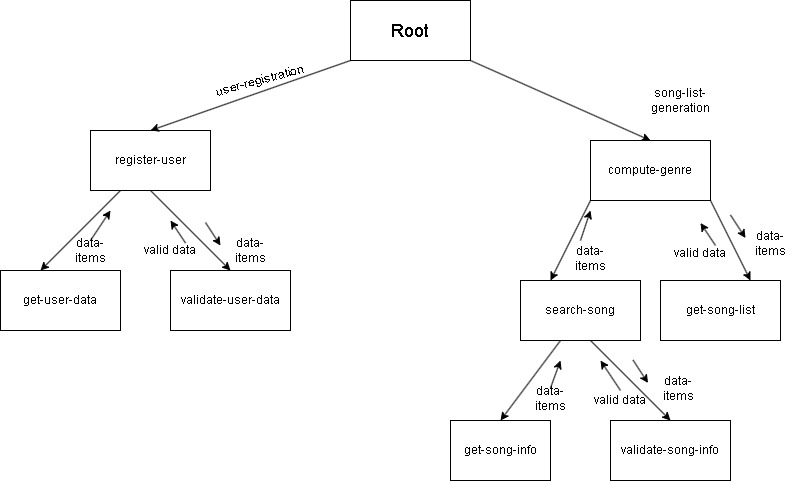
user-details : {userid+name}

song-details : {songid+songname+genre}

generate-playlist: {song-details}\*

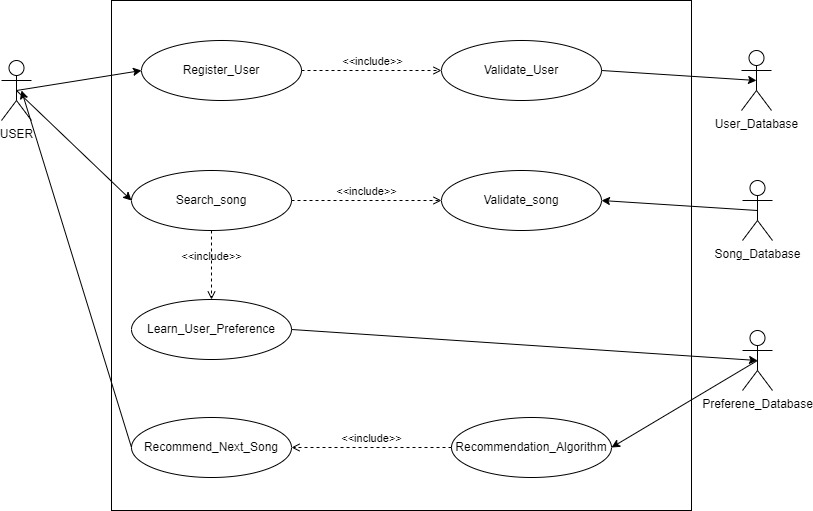
get-song-info: song-details

**Structure Chart on our project named Song Recommendation System:**



**Object Oriented Design:**

**Use Case diagram on Song Recommendation System:**



**Text description:**

**U1: register-user:** Using this use case, the user can register himself by providing the necessary details.

**Scenario 1: Mainline sequence**

1. user: select register user option

3 System: display prompt to enter First name, Last name, password and email

2. User: enter the necessary values

4 System: display the generated id and the message that the User has successfully been registered.

**Scenario 2: At step 4 of mainline sequence**

4 : System: displays the message that the User has already registered.

**Scenario 3: At step 4 of mainline sequence**

4 : System: displays message that some input information have not been entered. The system displays a prompt to enter the missing values.

**U2: Search Song: This use case allows the user to search for songs in the system**.

**Scenario 1: Mainline Sequence**

1. User: Selects the Search Song option.

2. System: Displays a prompt to enter song title, artist, or genre.

3. User: Enters search criteria.

4. System: Searches for songs matching the criteria and displays the results.

5. User: Selects a song from the results to view more details.

**Scenario 2: At step 4 of mainline sequence**

4. System: Displays a message indicating that no songs were found matching the criteria.

**Scenario 3: At step 2 of mainline sequence**

4. System: Displays a message indicating that the input is invalid and prompts the user to enter valid search criteria.

**U3: Learn User Preference:** This use case captures user preferences based on their song searches to enhance recommendations.

**Scenario 1: Mainline Sequence**

1. User: Searches for a song using the Search Song use case.

2. System: Records the searched song and user’s selection.

3. System: Updates user preferences based on the search and selection data.

**Scenario 2: At step 2 of mainline sequence**

2. System: Displays a message indicating that no preferences could be learned from the search (e.g., if no songs were selected).

**Scenario 3: At step 3 of mainline sequence**

3.System: Displays an error message indicating that there was a problem recording preferences.

**U4: Recommend Song:** This use case provides song recommendations based on learned user preferences.

**Scenario 1: Mainline Sequence**

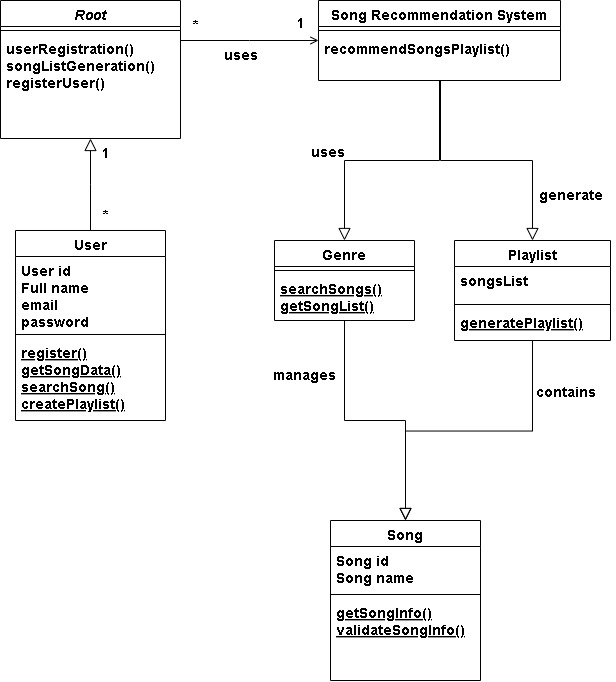
1. User: Selects the Recommend Song option.

2. System: Analyzes recorded user preferences.

3. System: Displays a list of recommended songs based on preferences.

4. User: Selects a recommended song to listen to.

**Class Diagram on Song Recommendation System:**



**DFD on Case Study 14 named Newspaper Agency Automation Software:**







**Data Dictionary for above DFD:**

CN : integer

address : name+house#+street#+city+pin

customer-details : address+{address}\*

valid-customer : CN+customer-details

pay-amount : integer

newspaper-details : newpapername + pay-amount

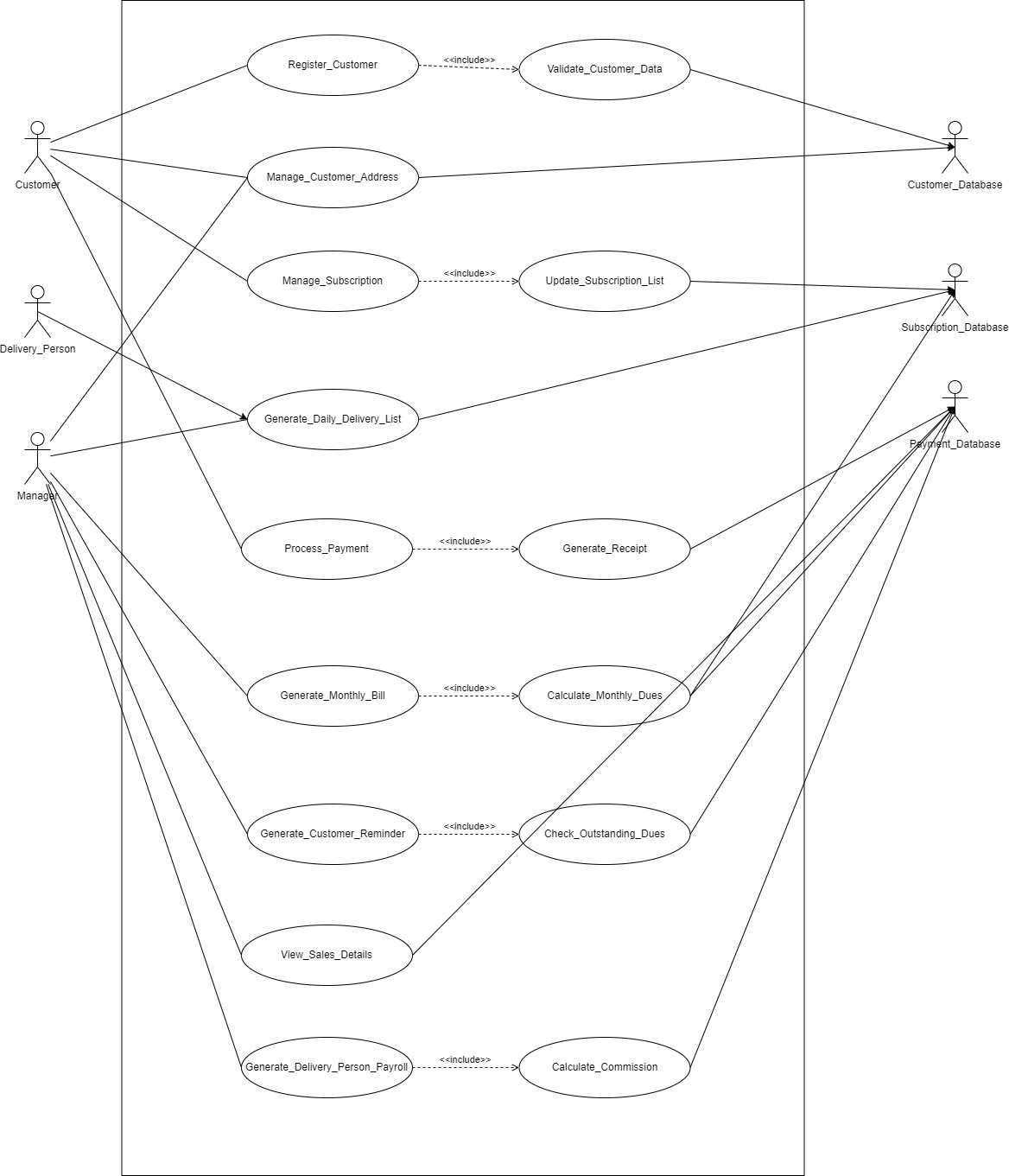
subscription-details : newspaper-details + valid-customer

**Structure Chart on Case Study 14 named Newspaper Agency Automation Software:**



**Object Oriented Design:**

**Use Case diagram:**



**Text description:**

**U1: register-customer:** A new customer can register for the newspaper delivery service by providing necessary details.

**Scenario 1: Mainline sequence**

User: Selects register customer option

System: Displays prompt to enter Name, Address, Contact information, and initial subscription preferences

User: Enters the necessary values

System: Validates the customer data

System: Creates a new customer record in the database

System: Displays the generated customer ID and a message that the User has successfully been registered

**Scenario 2: At step 4 of mainline sequence**

4. System: Displays a message that the User has already registered with the provided information

**Scenario 3: At step 4 of mainline sequence**

4. System: Displays an error message that some input information have not been entered. The system displays a prompt to enter the missing values.

**U2: Manage Subscription:** Allows the customer to modify their existing subscriptions.

**Scenario 1: Mainline Sequence**

User: Selects the Manage Subscription option

System: Displays current subscription details

User: Makes desired changes (add/remove publications, change frequency, suspend delivery)

System: Validates the changes

System: Updates the subscription list

System: Confirms the changes to the user

**Scenario 2: At step 4 of mainline sequence**

4. System: Displays a message indicating that the requested changes are invalid (e.g., trying to remove a non-existent subscription)

**Scenario 3: At step 3 of mainline sequence**

3. User: Requests to cancel all subscriptions

4. System: Displays a confirmation prompt for cancellation

5. User: Confirms cancellation

6. System: Processes the cancellation and updates records

**U3: Generate Daily Delivery List:** Creates optimized daily delivery lists for each delivery person.

**Scenario 1: Mainline Sequence**

System: Initiates daily delivery list generation (automatic or manager-triggered)

System: Retrieves active subscriptions from the database

System: Organizes deliveries by area and optimizes routes

System: Generates personalized delivery lists for each delivery person

System: Makes the lists available for delivery persons to access

**Scenario 2: At step 2 of mainline sequence**

2. System: Detects suspended subscriptions and excludes them from the delivery list

**Scenario 3: At step 3 of mainline sequence**

3. System: Encounters an error in route optimization

4. System: Generates non-optimized routes and flags for manual review

**U4: Generate Monthly Bill:** Creates monthly bills for customers based on their subscriptions and delivery history.

**Scenario 1: Mainline Sequence**

System: Initiates monthly billing process

System: Retrieves customer subscription and delivery data

System: Calculates monthly dues for each customer

System: Generates itemized bills

System: Makes bills available for delivery or electronic distribution

**Scenario 2: At step 3 of mainline sequence**

3. System: Detects partial month subscription or delivery suspension

4. System: Pro-rates charges for partial month or suspended period

5. System: Includes explanation of pro-rated charges on the bill

**Scenario 3: At step 2 of mainline sequence**

2. System: Detects customers with outstanding dues

3. System: Adds outstanding dues to the current bill with a reminder notice

**Class Diagram:**

