# Design Document

## **Class Diagram**

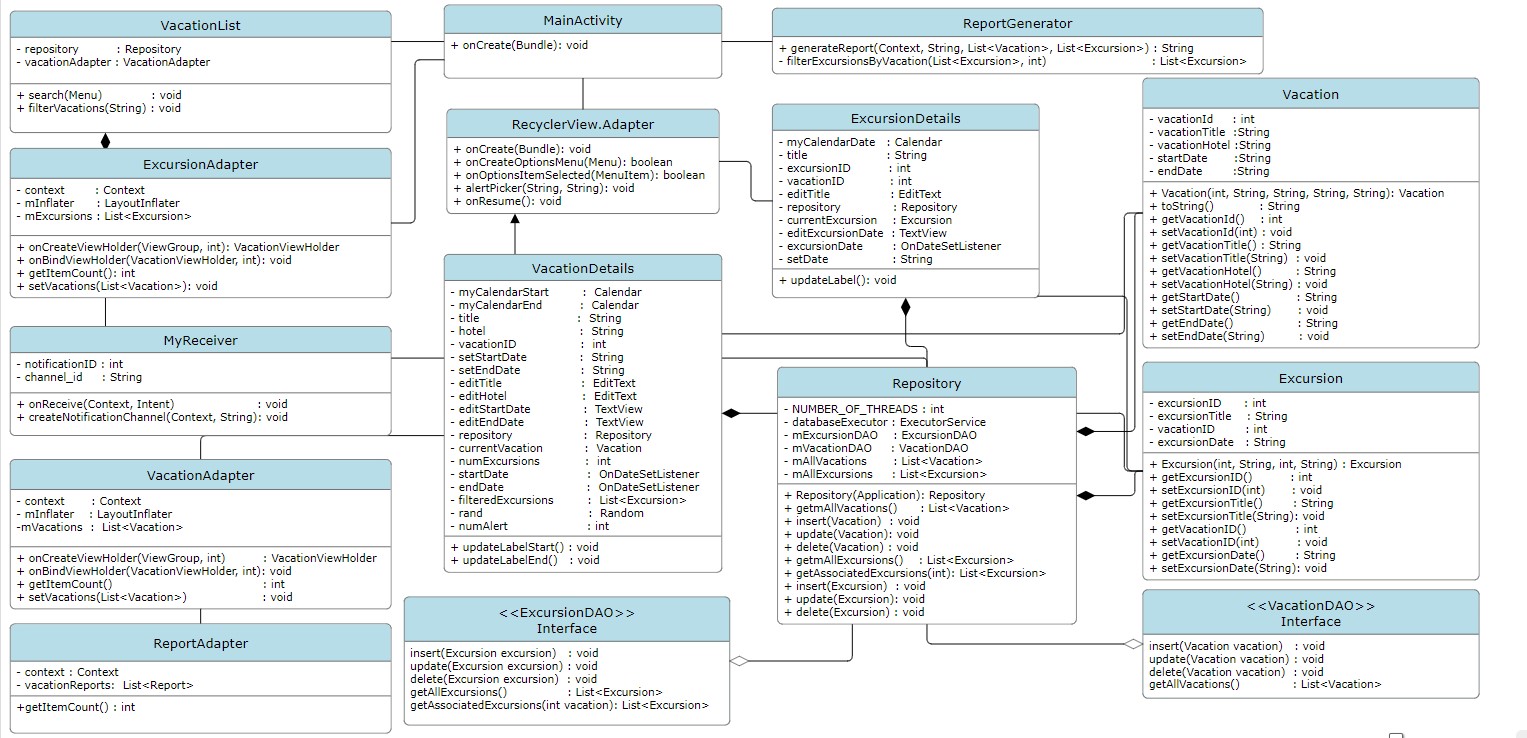


Figure 1: Class diagram

This simple app design is centered around several key classes, each responsible for different parts of the application. For simplicity, I have only mentioned the classes which has been affected by the assignment tasks.

* VacationList
  + Responsible for displaying the list of vacations. It uses a VacationAdapter to bind vacation data to the UI.
* MainActivity:
  + The entry point of the PA app that initializes the app and displays the vacation list.
* ReportActivity:
  + Renders the Report page UI
* VacationAdapter, ExcursionAdapter, and ReportAdapter:
  + An adapter class that binds vacation data to a RecyclerView in the UI.
* VacationDetails and ExcursionDetails:
  + Display detailed information about a vacation or excursion and provide methods for editing and updating them.
* ReportGenerator:
  + This class is responsible for generating reports based on the available vacations and excursions. The report is a simple row column table structure. It makes use of the Report Class
* Repository:
  + This is data access layer, which manages access to both vacation and excursion data through DAOs. It makes use of VacationDatabaseBuilder class under the hood.

## **Design Diagram**

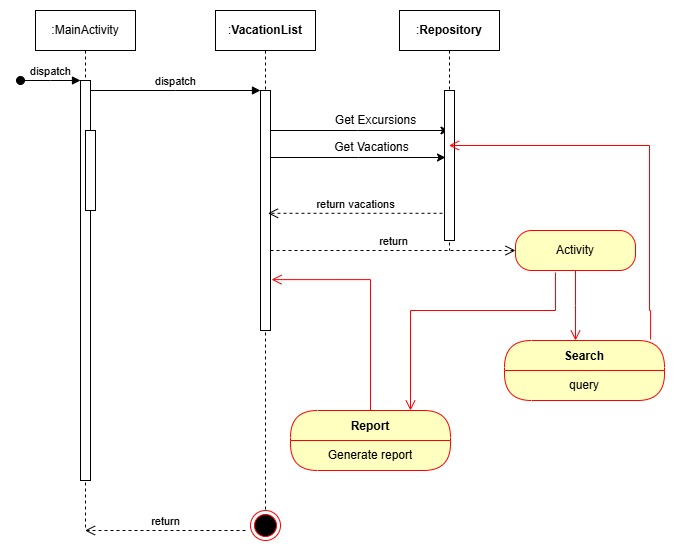


Figure 2: Simple design flow diagram of the app's functionality

This is a simple flow activity diagram which shows how the internal interactions happens between the system components. User opens the app which triggers Main activity to load the home screen. Form the home-screen, use clicks the vacations button and gets directed to the Vacation List page. In the vacation list page, use can search or generate report.

## **ERD Diagram**

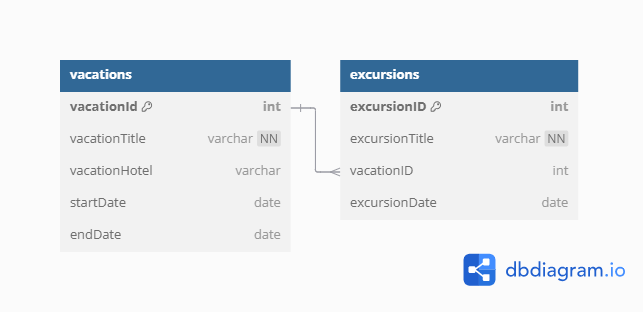


Figure 3: ERD diagram

This ERD diagram represents a simple schema of the PA app database layer. It has two relations that interact to store the data.

* Vacation Table: Stores vacation details including:
  + vacationId,
  + title,
  + hotel,
  + startDate, and
  + endDate.
* Excursion Table: Stores excursion details including:
  + excursionID,
  + title,
  + vacationID, and
  + excursionDate.

There is a one-to-many relationship between Vacation and Excursion Table. That is, vacation can have multiple excursions associated with it, forming a one-to-many relationship.