

Application Name: Insurance Report Generation

Plan Names: Cash / Food / Medical / Employment

Plan Status: Approved / Denied / Terminated

Gender: Male / Fe-Male

Note: Insert citizens records into DB table manually with given permutations and combinations.

Plan Name:	<input type="text" value="- Select -"/>	Plan Status:	<input type="text" value="- Select -"/>	Gender :	<input type="text" value="- Select -"/>
Start Date:	<input type="text" value="-select-"/>	End Date:	<input type="text" value="- select -"/>	<input type="button" value="Search"/>	
<hr/>					
<search results will display here>					
<hr/>					
Export : <u>Excel</u> <u>Pdf</u>					

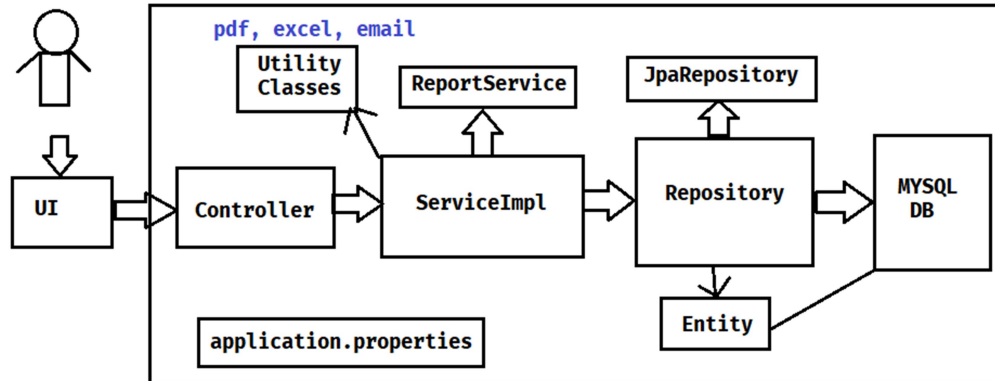
- 1) Plan Name dropdown values should come from database table
- 2) Plan Status drop down values should come from database table
- 3) It is dynamic search functionality. Based on Search criteria we need to fetch records from database and display in UI.
- 4) When users click on 'Search' without selecting any filter then we need to fetch all records from DB table and display in UI.
- 5) When user click on 'Excel' then complete database table data should be exported to excel file and send that excel file as an attachment to your email id.
- 6) When user click on 'PDF' then complete database table data should be exported to pdf file and send that pdf file as an attachment to your email id.

DB Table: CITIZEN PLANS INFO

CITIZEN_ID	NUMBER	PRIMARY KEY
CITIZEN_NAME	VARCHAR	
GENDER	VARCHAR	
PLAN_NAME	VARCHAR	
PLAN_STATUS	VARCHAR	
PLAN_START_DATE	DATE	
PLAN_END_DATE	DATE	
BENEFIT_AMOUNT	NUMBER	
DENIAL_REASON	VARCHAR	
TERMINATED_DATE	DATE	
TERMINATION_REASON	VARCHAR	

Java components

- 1) CitizenPlan.java (Entity Class)
- 2) CitizenPlanRepository.java (Interface)
- 3) ReportService.java (interface)
- 4) ReportServiceImpl.java (class)
- 5) PdfGenerator.java (class)
- 6) ExcelGenerator.java (class)
- 7) EmailUtils.java (class)
- 8) ReportController.java (class)
- 9) SearchRequest.java (class) - form binding/command/DTO
- 10) DataLoader.java (runner class)
- 11) index.jsp

Project Setup

- 1) Create starter project with required dependencies
- 2) Configure Data Source & View Resolver in Properties file
- 3) Create Entity class & repository interface
- 4) Create Runner to load data into DB table
- 5) Create Form Binding class
- 6) Create Service Interface with abstract methods
- 7) Create ServiceImpl class
- 8) Create Controller class
- 9) Create Utility Classes (PdfGenerator, ExcelGenerator & EmailUtils)
- 10) Create View Page (JSP) & add bootstrap css & bootstrap js (www.getbootstrap.com)
- 11) Run the application and access welcome page
- 12) Create Git Repo and Push Project into Git Repo