Version History:

- 1. Introduction

4. Key Contributors

1. Moderator functionalities

2. Admin user functionalities

2. Product Functionalities

3. Technology Stack

6. Integration with APIs

4. Module List

14th March 2019

Date

Table of Contents: 1. Purpose 2. Product Overview 3. Browser compatibility

Author(s)

Shetty

Abdul Rehman, Shashank S

Comments

Initial Architecture for Admin Portal

Version

1.0

- 7. Modular Design 8. Apply Standards and Best Practices
 - **Introduction** 1. Purpose: To create a moderator tool, where moderators and administrators can monitor the content created on

5. High Level Architecture Design

- the HCA Inspire app. 2. Product Overview: An admin portal, where administrators & moderators can monitor posts and their users. This will allow them to decide whether to keep or discard a post that's been flagged as inappropriate by some
- users. The users can choose to generate reports based on the action, reason type, dates (or) request the data for a specific person. This application can also be used for searching a particular user and see the list of posts added by that user. In addition to this, administrators can add or remove a user from the system and also modify role of a user.

#

- 3. Browser Compatibility: 1. Chrome + 4 prior versions 2. Firefox current version + 4 prior versions
- 3. Safari current version + 2 prior versions
- 4. IE11 and above 4. Key Contributors:
 - Abdul Rehman 1 2 Shashank S Shetty

Author

- **Product Functionalities**
- **Moderator Functionalities:** • Onboarding (Sign In / Sign Out) View Task List
- Perform Action: Keep/Discard posts Generate Reports Search Users Admin User Functionalities: • Onboarding (Sign In / Sign Out) View Task List

START

Display list of posts

View a post

Read through

the message

Is offensive?

SUBMIT

END

-YES

REMOVE

Select reason

to discard

Add comments

View reporters

Close (X)

NO-

ALLOW

Select reason

for keeping

Add comments

Role

Software Engineer

Associate Software Engineer

Technology Stack

Generate Reports

• Update Users' Roles

Search Users

Remove Users

• React - JS library for building user interfaces in a Single Page Application(SPA).

Perform action: Keep/Discard posts

• React Router - library for mimicking navigation in SPA. • Redux - State management library for storing and manipulating data.

• <u>Jest</u> - Unit Testing library.

• <u>Istanbul</u> - Unit test coverage reporter.

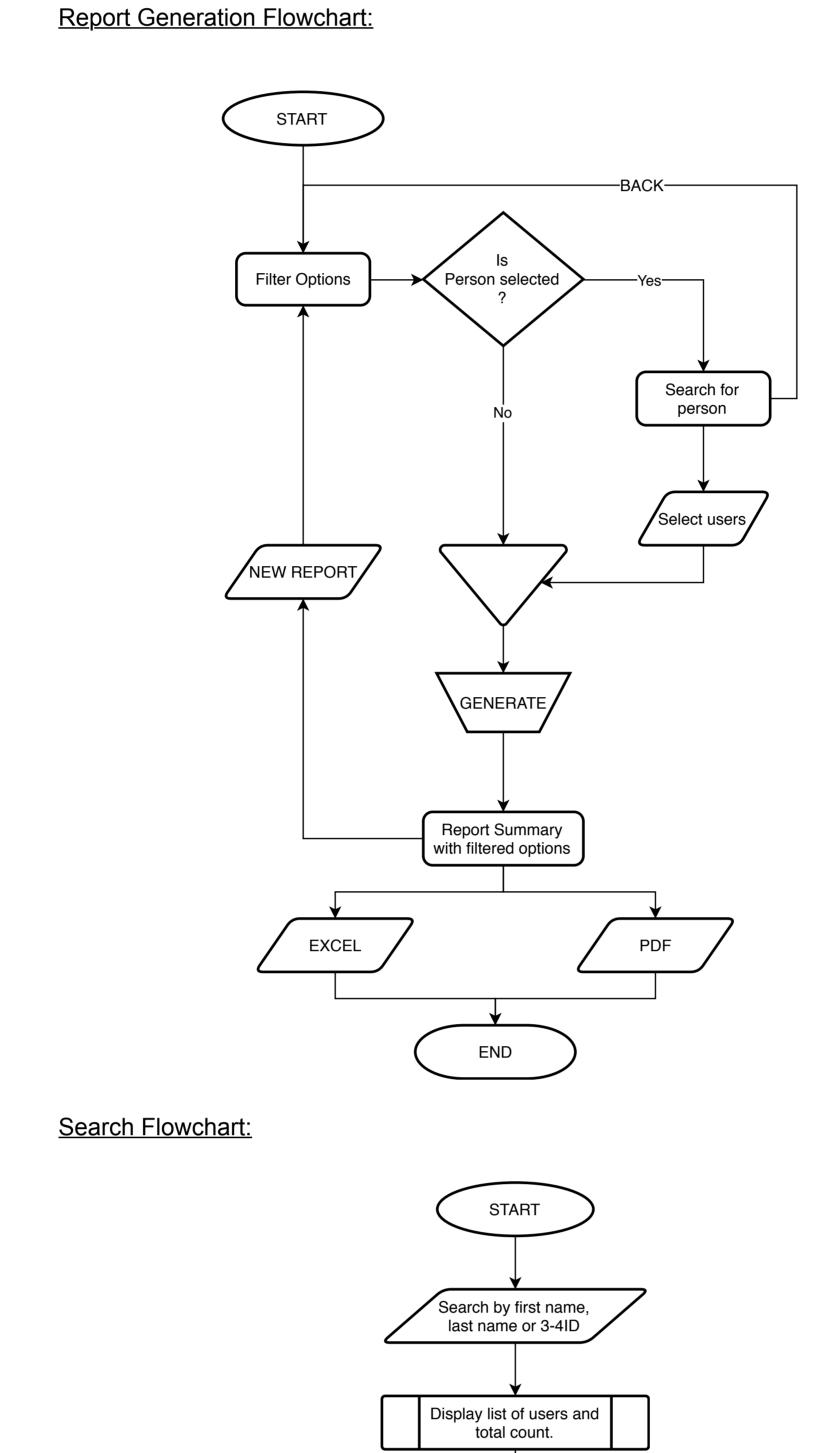
- <u>Docker</u> run software packages called containers. • Story Book - User interface development environment and playground for UI components. • <u>SCSS</u> - CSS preprocessor.
- Redux Saga Asynchronous API call event manager.
- **Export Reports** User Search **User Management User Profiles**

Module List

Task List

User Auth

- **Modular Design Tasklist Flowchart:**



Click on user field

View user's personal details and post.

END

START

Navigate to next/previous pages

Select a user

END

TableHeader

+ headerSortBy(sortBy: string,

asc: Boolean): void

TaskList

- notificationCount: Integer - filterBy: String - viewGrid: Boolean # data: List - showCount: Integer - hasMoreData: Boolean

+ allowPost(id: ID): Boolean + removePost(id: ID): Boolean + fetchData(filterBy: String, currentCount: Integer, showCount: Integer): void - updateFilterBy(filter: String): List - toggleGridView(): void

Display list of users and

user details

CANCEL

Remove user

confirm removal

Yes, REMOVE

Show deleted alert message

ListView

header: List - sortableHeader: List

+ loadMoreData(): List

TableRow

- readMore: Boolean

+ loadMoreData(): List

- toggleViewMore(): void

- readMore: Boolean

+ loadMoreData(): List

- toggleViewMore(): void

Search

- search: String

+ getSearchedText(): String

TaskList Class Diagram:

GridView

+ loadMoreData(): List

Card

- showReporter: Boolean

- toggleShowReporter(): void

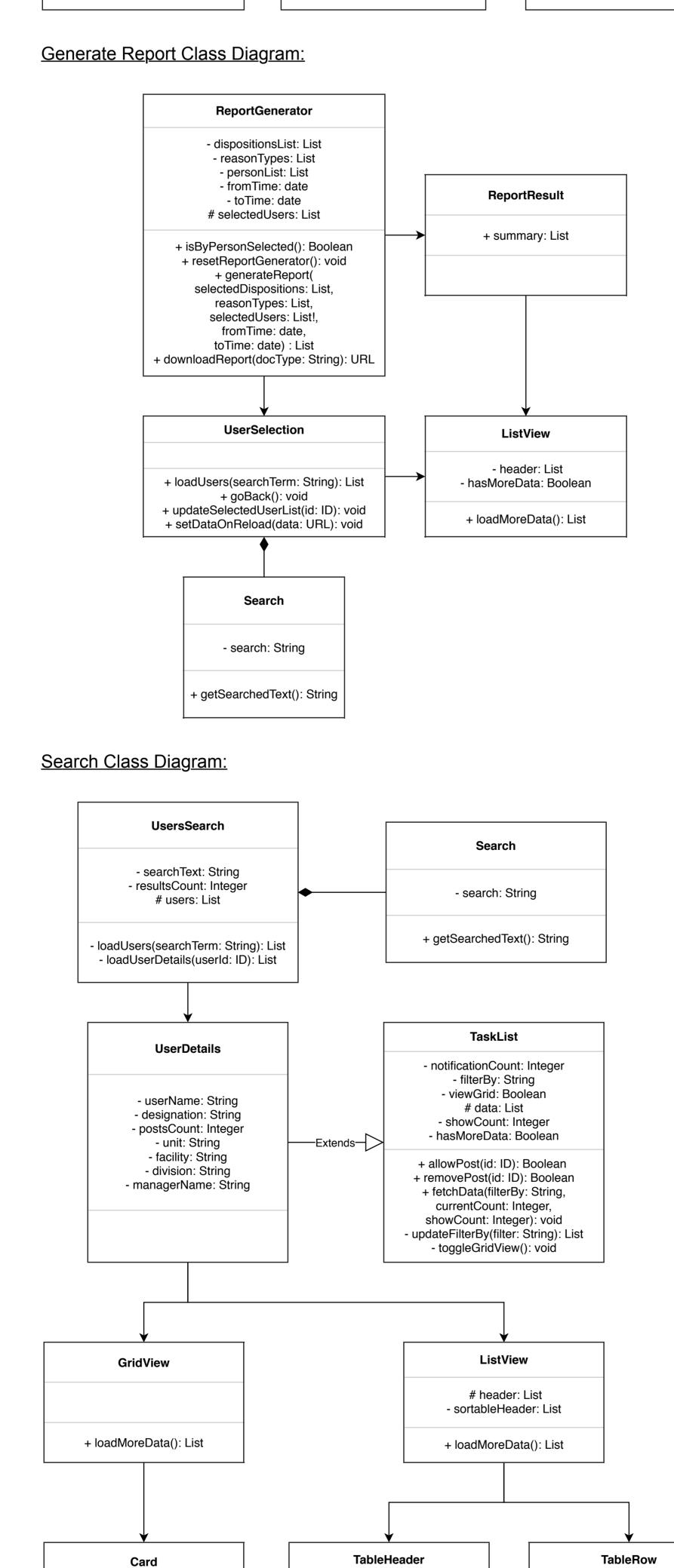
Admins & Moderators Flowchart:

Search by first name,

last name or 3-4ID

Update user's role

SAVE **CHANGES**



+ headerSortBy(sortBy: string,

asc: Boolean): void

ModeratorsAndAdmins

- maxPages : Integer - currentPage : Integer # data : List - showUsersCount: Integer -#updatedRoleList: List

+ searchUsers(search: String): List

+ saveChanges(): void

+ fetchData(currentPage: Integer, showUsersCount: Integer, sortBy: String!, asc: Boolean!, search: String!): void + updateUserRole(id: ID, actualValue: String, updatedValue: String): void + removeUser(id: ID): void

PaginationTable

header: List # sortBy: String # asc: Boolean - sortableHeader: List

+ loadMoreData(sortBy: string, asc: Boolean): void + navigateTo(step: Integer): void

TableRow

- readMore: Boolean

+ loadMoreData(): List - toggleViewMore(): void

TableHeader

Apply Standards and Best Practices

• ESLint and Prettier for Linting.

• Airbnb Style Guide for best coding practices.

+ headerSortBy(sortBy: string,

asc: Boolean): void

- showReporter: Boolean - reviewStatus: String - reviewDate: Date

- reviewerName: String

- toggleShowReporter(): void

Alert

- message: String

+ showAlert(): String

Admin and Moderators Class Diagram:

Use Case: Login View reported posts Approve/Reject posts Generate Reports <<extends>> Save Reports as PDF/EXCEL Moderator Administrator Search HCA Employee **Search Moderators** and Admins Update user's role Remove user

Log out