PERSONNEL ROSTERING SOLUTION

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Sakshi Sharma (151346)

Under the supervision of

Mr. Abhishek Garg

(Development Manager, SAP Innovative Business Solutions)

and

Mr. Shishir Agarwal

(Project Manager, SAP Innovative Business Solutions)

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **" Personnel Rostering Solution"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from February 2019 to May 2019 under the supervision of **Mr. Abhishek Garg** (Development Manager, SAP Innovative Business Solutions) and **Mr. Shishir Agarwal** (Project Manager, SAP Innovative Business Solutions).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Sakshi Sharma, 151346

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr. Abhishek Garg

Development Manager, SAP Innovative Business Solutions

Mr. Shishir Agarwal

Project Manager, SAP Innovative Business Solutions

Acknowledgement

The internship opportunity I had with SAP Labs India Pvt. Ltd. was a great chance for learning and professional development. Therefore, I consider myself a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me through this internship period.

I express my deepest thanks to Mr. Abhishek Garg, Development Manager, SAP Innovative Business Solutions for taking part in useful decision & giving necessary advice and guidance and arranged all facilities to make life easier. I choose this moment to acknowledge his contribution gratefully.

I would like to express my deepest gratitude and special thanks to Mr. Shishir Agarwal, Project Manager, SAP Innovative Business Solutions, who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my project at their esteemed organization and extending during the training.

It is my radiant sentiment to place on record my best regards, deepest sense of gratitude to the head of Department of Computer Science & Engineering and Information Technology, Prof. Dr. Satya Prakash Ghrera for giving me the opportunity to go through this internship. I would also like to thank all the team members of the project who were always there to constantly guide and help me throughout the project.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives.

Sincerely,

Sakshi Sharma

Table of Contents

List of Abbreviations

Abbreviation	**Explanation**
SAP ABAP	SAP Advanced Business Application Programming
FIT	Functional Integration Testing
GUI	Graphical User Interface
JS	JavaScript
JSON	JavaScript Object Notation
MIT	Module Intergration Testing
OData	Open Data Protocol
PRS	Personnel Rostering Solution
SAP HANA	SAP High-performance ANalytical Appliance
SAP HCM	SAP Human Capital Management
SAP SD	SAP Sales and Distribution
SAP S/4 HANA	SAP Business Suite 4 (generation) SAP HANA
SAPUI5	SAP User Interface for HTML5
UAT	User Acceptance Testing
UI	User Interface

List of Figures

List of Tables

Abstract

Workforce scheduling can be a difficult task to perform especially when it comes to providing services to multiple clients. To meet the service requirements, personnel are deployed daily for different shifts, customer sites, rules, contracts, and the list goes on and on. If all this work was to be done manually using excel based tools, then it would be a nightmare for anyone. So, the need of the hour is to have a flexible and scalable rostering solution.

As of now, there is no one-stop-shop rostering solution available in the market. All the solutions available are in bits and pieces. PRS (Personnel Rostering Solution) aims to fill that gap. The rostering solution must take care if performance issues. The rostering engine must be capable of processing high loads without degrading the speed aspect.

PRS caters to a wide range of functions. It can integrate with other systems to get clocking times of rostered personnel. Multiple individual software is used for various rostering related functions. PRS offers highly configurable, scalable and flexible system.

Chapter – 1

INTRODUCTION

1.1 Introduction

Personnel Rostering is the procedure of creating work schedules for the staff of an organisation in order to help the organisation to satisfy the demand for its services [1]. First of all, we need to determine personnel who meet the particular skill (or rank) as required by the customer according to the contract signed. They are then evaluated for various factors such as coverage, i.e. the days of the weeks that need to be covered and the different shifts and shift timings as per the contract, constraints (e.g. the person must be female), availability type (e.g. if the person is rostered, available, on off-shift, etc.), and much more. All the industrial rules related to the appropriate workplace agreements should be taken care of during rostering.

When it comes to the objectives of companies or organizations, then, the efficiency of personnel and productivity take the lead. The problem of personnel scheduling can be quite complicated when it comes to shifts for customers where shift personnel has to work throughout the week and there are non-routine working hours for some personnel. [2]

The personnel are divided into different plants or deployable business units for provided security services. The divisions are on the basis of armed and unarmed security services. Security personnel are deployed on various shifts and customers sites around the clock to meet the service requirement. While doing this, the multiple rules as stated in the customer contract must be adhered to and the various deployment-related guidelines must be followed. So, in order to accomplish are these tasks on a large scale, the personnel rostering solution must be flexible and scalable.

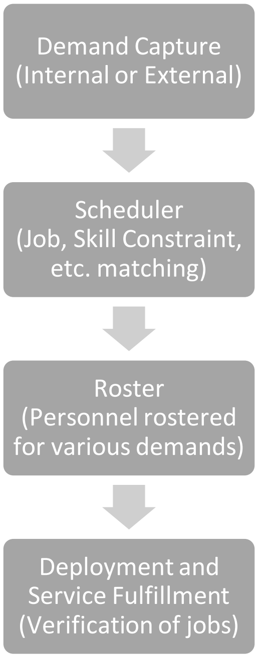

Demand Capture
(Internal or External)
Scheduler
(Job, Skill Constraint, etc. matching)
Roster
(Personnel rostered for various demands)
Deployment and Service Fulfillment
(Verification of jobs)

Figure 1.1: Rostering - PRS Process

The PRS Process can broadly be covered in the following steps:

i. **Contract:** Contracts created are used to capture requirements.

ii. **Demand capture and deployment items generation**: Enable demand managers to capture and automatically create internal demands that can be billed and deployed. Additionally, generate corresponding deployment items based on demand, shift planning and deployment attributes. The solution allows demand managers to create demands either from the deployable contracts or on an ad-hoc basis without a contract.

iii. **Roster generation and daily deployment**: Provide an automated solution and an easy to use tool to generate roster plans for 3 months and 1-month period upfront and assign personnel to demands. Provisions shall be made for manual adjustments.

iv. **Service fulfilment and reconciliation on actuals**: Capture service fulfilment in the PRS solution, integrating with the clocking Interface for actual attendance and customer confirmation interface of the system, which provides the confirmation of Actual times. The billable and payable hours are consolidated based on Actuals.

v. **Reports**: Backend-GUI and web browser-based reports shall be provided to different business users of the new PRS solution.

1.2 Problem Statement

Personnel Rostering is the task of distributing services among the workforce or personnel in order to provide services to satisfy the requirements of the customer. The resulting schedule or the roster may range over several weeks. The various criteria that must be satisfied by the roster are public holidays (depending upon coverage), off-days (e.g. at least one off day in 7 days), preferred personnel, etc.

Rostering becomes challenging when the number of personnel required and the number of shifts increases because of the shift rotation patterns that must be followed as per the coverage.

The issues with existing personnel rostering solutions are as follows:

i. **No complete visibility:**
Availability of various features is in bits and pieces. So, it becomes difficult to view each and everything on a single platform. That is why a One-Stop-Shop of the rostering solution must be created.

ii. **Performance issues:**
Rostering engines normally are slow to roster personnel because of processing load. With high loads, the available applications suffer from performance issues. This slows down business. Therefore, performance issues must be taken care of.

iii. **Island of systems:**
Multiple individual software is being used for various rostering related functions. So, the PRS should cater to a wide range of functions.

iv. **Rigid and non-scalable system:**
The available systems are not scalable and flexible. They suffer from scalability and rigidity issues. PRS should be highly configurable, scalable and flexible.

1.3 Objectives

The objective is to create a PRS (Personnel Rostering Solution) that overcomes all the business challenges faced by the rostering solutions currently existing in the market. The PRS should have the following features:

i. **Performant:** The system should not suffer from performance issues in case of high loads.

ii. **Scalability and configurability:** The system should be highly configurable system as per the needs of customers (like business rules maintenance, constraints, allowances, etc.).

iii. **Anywhere Access:** Web-based access with mobility.

iv. **Usability**: Live Dashboards to monitor the deployment should be available.

v. **Segregation of duty:** Role-based access. User should be able to view different applications and perform different actions. (like Sales, Deployment Manager, etc. on the basis of accessibility matrix).

1.4 Methodology

Software Development Methodology: Agile Scrum Methodology

For the development of software, agile methodology has been followed throughout the project. Scrum is a type of agile methodology in an iterative manner. [3] In this methodology, the roles and tasks are distributed among the people who are a part of the project.

Team Members:

i. Product Owner

ii. Members

iii. Scrum Master

Further, we have Project Manager, Quality Manager and architects for backend and frontend, depending upon the technologies.

The product owner is the one who represents the customer, understands the requirements of the customer and clarifies them to the whole team. The Scrum Master is like the coach throughout the development process of the project.

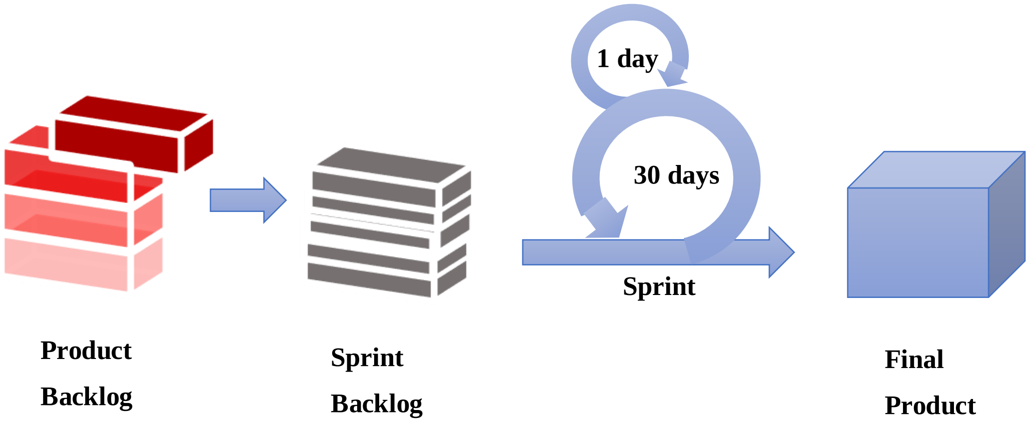


Figure 1.2: Overview - Scrum Process

The user requirements at a particular point of time are a part of the product backlog. They are arranged on the basis of business value or priority. The project duration is divided into sprints and at the start of each sprint, the sprint meeting takes place in which sprint backlog is discussed on which development will take place in the new sprint. The requirements are presented as user stories.

During the sprint, scrum meeting takes place every day which is a short meeting to discuss the progress of every member and to discuss blockers, if any.

Technologies Involved: SAPUI5, SAP ABAP, SAP HANA, NodeJS

Frontend – SAPUI5:

One of the objectives of PRS is mobility. The user should be able to access the applications anywhere and at any device. So, SAPUI5 makes that task easy from frontend point of view. SAPUI5 is a JavaScript-based framework which consists of libraries that help in building applications that support both mobile and desktop browsers. With the help of SAPUI5, we don’t have to maintain the different code base for mobile and desktop. Moreover, using oData services to render data is quite easy using SAPUI5.

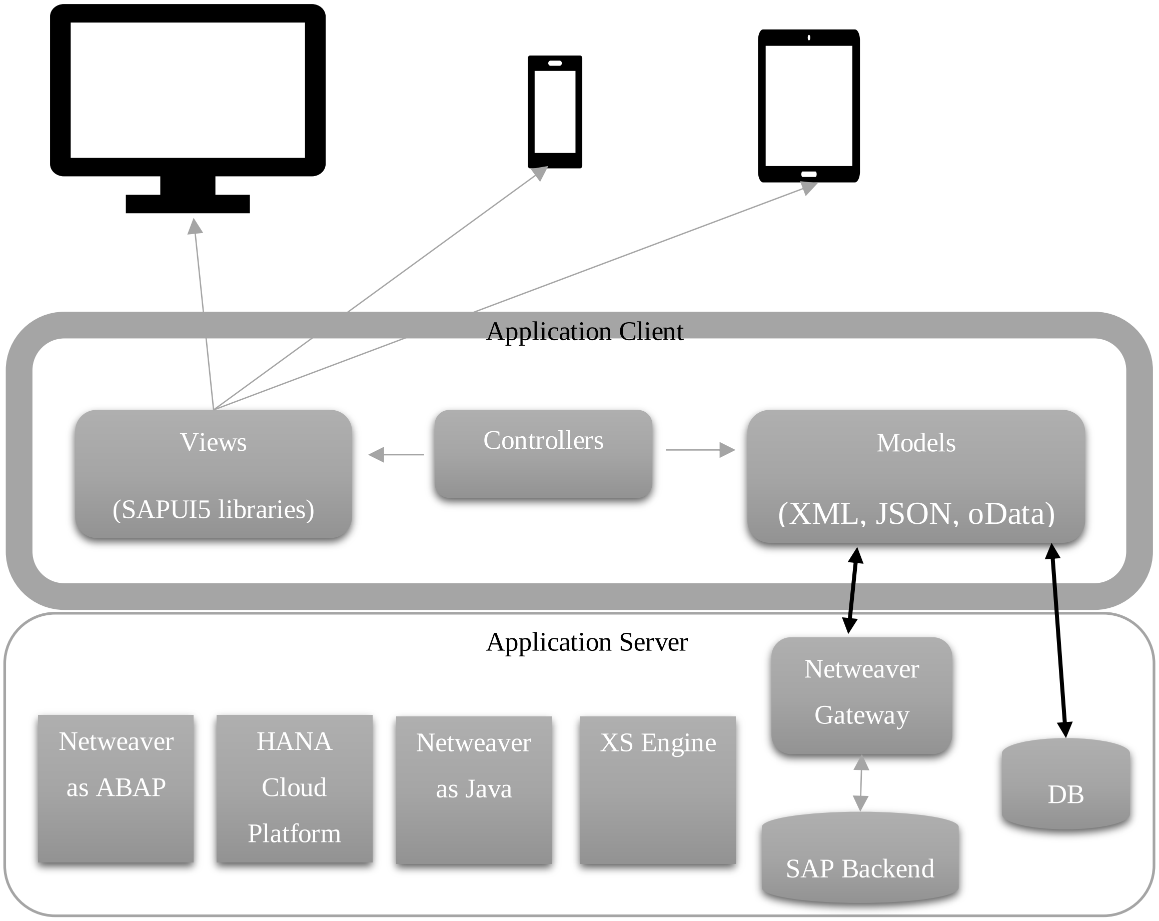


Figure 1.3: UI5 Architecture Overview

Model-View-Controller Concept:

SAPUI5 follows the model-view-controller concept. The aim is to separate information representation from user interaction. So, the UI code is divided into three parts:

i. Model: The model manages the application data.

ii. View: The view renders and defines the UI.

iii. Controller: The controller modifies the View and Model by reacting to events taking place in the view and during user interaction.

Controller:

The task of controller is reacting to events. So for that purpose, we define 4 types of events in the controller. Ideally, the controller files should be stored separately in the 'controller' folder and the file name will be something like '{controllerName}.controller.js'.

The four events defined in a controller are as follows:

i. onInit
ii. onBeforeRendering
iii. onAfterRendering
iv. onExit

SAP ABAP – Backend:

All the objects and coding part has been done using SAP ABAP.

SAP HANA – Cloud Platform:

SAP HANA is a kind of database that organizes all data in columns so that the data can be read from top to bottom. The focus is on columns that are needed in order to enable fast processing. The ambition of SAP HANA is to allow a combination of OLTP and OLAP patterns within one application. For intensive search operations and large number of columns, like PRS, Columnar Storage is preferred over Row Storage.

OData Testing Automation – Node.js:

For automating OData Testing through Postman, Node.js has been used. Using Node.js, the excel sheet containing request data can be converted to Postman collections.

Tools Used: Visual Studio Code, IMS, Jira Scrum Board, Postman

Code Editor – Visual Studio Code
IMS – for reporting/tracking bugs
Jira Scrum Board – Tool to task tracks in Scrum software development methodology
Postman – for OData testing

Version Control System: Git

Testing Phases: FIT, MIT, UAT

1.5 Organization

Chapter 2 contains information about Scope. Functional and Non-Functional Requirements will be covered in this chapter.

Chapter 3 contains information about System Development. The architecture and algorithm of PRS will be covered in this chapter.

Chapter 4 contains information about Performance Analysis. It contains details about the parameters on which performance of PRS has been analysed and the results obtained.

Chapter 5 contains the conclusion and future scope of PRS.

Chapter – 2

SCOPE

2.1 Scope Overview

PRS Scope is defined for each application. It covers the functional requirements of the project.

2.1.1 High-Level PRS Features

The PRS Process can broadly be covered in the following steps:

i. **Contract:** Contracts created are used to capture requirements.

ii. **Demand capture and deployment items generation**: Enable demand managers to capture and automatically create internal demands that can be billed and deployed. Additionally, generate corresponding deployment items based on demand, shift planning and deployment attributes. The solution allows demand managers to create demands either from the deployable contracts or on an ad-hoc basis without a contract.

iii. **Roster generation and daily deployment**: Provide an automated solution and an easy to use tool to generate roster plans for 3 months and 1-month period upfront and assign personnel to demands. Provisions shall be made for manual adjustments.

iv. **Service fulfilment and reconciliation on actuals**: Capture service fulfilment in the PRS solution, integrating with the clocking Interface for actual attendance and customer confirmation interface of the system, which provides the confirmation of Actual times. The billable and payable hours are consolidated based on Actuals.

v. **Reports**: Backend-GUI and web browser-based reports shall be provided to different business users of the new PRS solution.

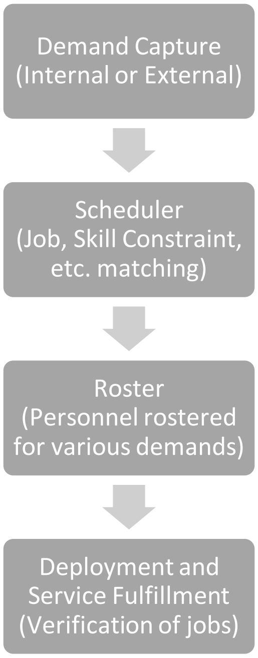

Demand Capture
(Internal or External)
Scheduler
(Job, Skill Constraint, etc. matching)
Roster
(Personnel rostered for various demands)
Deployment and Service Fulfillment
(Verification of jobs)

Figure 2.1: PRS – Overview

2.1.2 HCM Organizational Hierarchy

The deployment group is used to uniquely identify the number of deployable personnel. Below are key concepts regarding deployment group:

i. Currently, the Deployment group is assigned in a PRS custom table. For every organization unit, the Deployment group is assigned.

ii. Deployment Group needs to be assigned to each organization unit if it is being "PRS" applicable. So, if a person is not deployable, do not assign Deployment group to his organization unit.

iii. Ability to exclude non-deployable staff from deployable staff.

iv. Whenever an employee is transferred, his roster information should be deleted based on the transfer start date.

2.1.3 Contract Process

Below is the set of functional requirements:

i. The system shall provide the ability to create sales contract and provide all the contract related data at the header level.

ii. The system shall provide Sales users with Additional Interface to maintain specific Deployment Data for each Contract item - one-line item in demand will be converted into one demand in PRS.

iii. The system shall track Contract Deployment related changes to sync information.

iv. The system should be able to cater to the requirements for Constraints/ Rank/Deployment location of the personnel.

2.1.4 Coverage Process

Below is the set of functional requirements:

i. The system should be able to input coverage data corresponding to each contract line Item. The system should be able to enter personnel quantity and coverage days etc.

ii. System should be able to submit the contract and coverage data to web UI for Work Item and demand creation.

2.1.5 Demand Planning

PRS Solution shall be developed with the following features for Demand Planning:

i. The system shall be able to create new demands (Work items / Demand items) for Deployments and attaching the contract from PRS web UI.

ii. The system shall provide capability to user to maintain existing Demands.

iii. The system should be able to change the deployment location of the demand and to get input for deployment group for the demand.

iv. The system shall provide ability to maintain Constraints to a Demand.

v. The system shall provide the provision to add one or more constraints from the list of constraints.

vi. The system shall provide ability to maintain Allowances to a Demand.

vii. The system shall provide the provision to add one or more allowances from the list of allowances.

viii. The system shall allow Generate deployment from the PRS solution once demand planning is complete.

ix. The system shall allow to split Coverage Item into multiple Coverage items based on multiple shift patterns i.e. more than one shift pattern for a single demand item.

x. The system shall enable user to perform Shift Planning for each Demand item.

xi. The ad-hoc demands should be able to link to Contracts.

xii. The system shall allow to enter details regarding armoury and arms in relevant cases.

2.1.6 Generate deployment

Below are the key functional requirements:

Demand Generate deployment is a step which signifies that planning is complete and personnel are ready to be rostered. Below are the key functional requirements:

i. Once demand planning is complete, system should be able to perform action generate deployment.

2.1.7 Roster Requirement

Below are the key functional requirements:

i. System shall allow Deployment center to review generated one-month Roster Plan and make adjustments.

ii. System shall need to capture number of personnel through deployment group maintained in some level of hierarchy in the organization structure.

iii. Rostering of Demands shall be done based on pre-Defined Configurable Business Rules.

iv. System shall not consider debarred personnel at the time of automatic rostering.

2.1.8 Manage Roster

Below are the key functional requirements:

i. System shall be able to Search for personnel based on Constraints, availability Types, deployment Group and in a given time period.

ii. System shall allow adding or deleting Shifts in the Roster Plan and Daily Deployment.

iii. System shall allow splitting shift (changing shift times) manually in the Roster Plan and Daily Deployment.

iv. System shall be able to capture Unplanned absence of Personnel assigned to Shift.

2.1.9 Blacklisting / Whitelisting

Below are the key functional requirements:

i. System should be able to blacklist personnel based on Plant, contract or location.

ii. Blacklisted personnel will not be rostered automatically for the relevant plant/contract or demand, but they can always be deployed manually in case of shortfalls.

iii. Whitelisted personnel will always be given preference in automatic rostering for those contract/plant or demand

2.1.10 Deployment Dashboard & Maintain Tolerance

Below are the key functional requirements:

i. PRS solution provides a Dashboard which will be refreshed every minute. Dashboard will consist of Plan Shifts and Working Shifts.

ii. In addition, Live Dashboard will have calculated attributes for X hour window where X is configurable.

iii. Search Personnel: Live Dashboard will be searchable by personnel.

2.1.11 Job Confirmation

Below are the key functional requirements:

i. System shall provide a provision to confirm the job done by personnel.

ii. System shall provide review screen to enter billable and payable times.

iii. System shall provide a program to calculate Allowances based on Actuals and update Custom info type created for personnel Allowances.

iv. Ability at service confirmation to treat ‘confirmed’ checkbox as a level of approval for checking the times entered.

v. Service confirmation job should be scheduled such that it picks up complete only after a certain time.

2.1.12 Team Roster

Below are the key functional requirements:

i. System shall give the option to select the deployment group whose personnel’s roster information is to be shown.

ii. System shall show the list of employee with their roster details for a particular deployment group.

iii. System shall display employee’s roster, employee’s payable payable hours on daily basis, expected total working hours in the month, summary info, and detail per day (calendar view), shift timings, assigned locations, download roster.

iv. Must be easily viewable from smartphone/ table.

2.1.13 My Roster

i. Employee shall be able to view own roster, payable hours on daily basis, expected total working hours in the month, summary info, detail per day (calendar view), download roster.

ii. Employee shall be able to view total weekly hours per week (Monday-Sunday).

iii. Must be easily viewable from smartphone/ tablet.

2.2 Non-functional Requirements

2.2.1 Design and Implementation Constraints

These design and implementation constraints are to be considered with the functionality described in the above sub-section:

Design Decision and Constraints

i. PRS solution shall be deployed on S/4 HANA.

ii. PRS shall be using ABAP technology stack, with User Interface to be mainly developed using SAP UI5 technologies.

Common Requirement

i. STFK User Logins.

ii. Remote connection from home.

2.2.2 Supported Browsers/Devices

Desktop:

All PRS Fiori Apps (as per authorization)

Tablet:

i. My Roster

ii. Team Roster

iii. Manage Rosters

iv. Manage Deployment

v. Job Confirmation

Smartphone:

i. Manage Rosters

ii. My Roster

iii. Team Roster

2.2.3 Integration to HCM

2.2.3.1 Employee Constraints

Constraints under HR can be divided in 3 categories:

i. General

ii. Qualification

iii. Education

General:

Constraints such as Nationality, Driving License, Age, Gender, Race, etc. come under the general category.

Qualification:

i. These can be education and training related constraints and will be maintained in HCM system under qualifications Infotype.

ii. Qualification is plant based. So whatever qualifications are defined in HR system as relevant for a plant, only those will be available to choose.

iii. Any new qualification when added by HR and attached to the plant will become immediately available on refreshing the page.

iv. Qualifications have an expiry date too. So, when Roster checks these qualifications, it will match the demand coverage dates with qualification expiry date too.

Education:

i. Education constraints can also be defined in the system like – person should have this minimum educational qualification. Now for system, it means that any person who has that education or above, is eligible to be deployed.

Following employee personnel data is used as constraints to filter data as per the requirement.

i. Nationality: Used as “General” constraint

ii. Birth date (Age): Age of the employee is determined based on Birth date. Used as “General” constraint

iii. Gender: Used as “General” constraint

iv. Rank: Employees Rank / designation Used as constraint. Like Corporal /Lance Corporal

v. Race: Used as “General” constraint

vi. Work Experience: Used as “General” constraint

vii. Education: Used as “Education” constraint

viii. Qualification/ Skill: Used as “Qualification” constraint

2.2.3.2 Training and Event Management Process

i. Training mandatory for deployment. Upon the completion of successful completion of training, the qualification is updated.

ii. Qualification is used as a constraint to identify the right resource for deployment.

2.2.4 Quality Attributes

i. Performance

a. PRS solution extends flexibility to run business critical options in background.

b. Monthly rostering job shall be completed overnight for deployment group with size varying from 200-1000 jobs.

c. Debarment: Debarred personnel will be immediately removed from roster.

d. For a contract with 4 items with 100 personnel, deployment generation should finish approximately 2 minutes.

ii. **Robustness**

a. It shall be possible to detect any failure of Automatic Rostering process via Job monitoring and re-run/reschedule job manually in case of failure.

b. Any failure in automatic Rostering process shall cause complete data rollback so that data inconsistency won’t happen in system.

iii. **Usability**

a. Different colour coding shall be provided in deployment dashboard to see.

b. Relevant apps should be optimized for mobile usage.

iv. **Extensibility**

   a. Demand constraints are configurable, and it is possible to add new general/education/qualification constraints.

   b. Demand allowance – any allowance can be added as long as it follows the allowance rules configured in the system.

   c. Values of certain rules can be changed. No new rules can be added.

v. **Data Volume**

   a. PRS should work well with large volume of data.

   b. For example, a plant can have thousands of employee.

Chapter – 3

SYSTEM DEVELOPMENT

In this chapter, the high-level architecture of PRS will be covered along with the flowchart of all the apps to get an overview of the algorithms of PRS.

3.1 System Landscape and Architecture Overview

Below is the High-level Architecture and overall process flow for PRS.

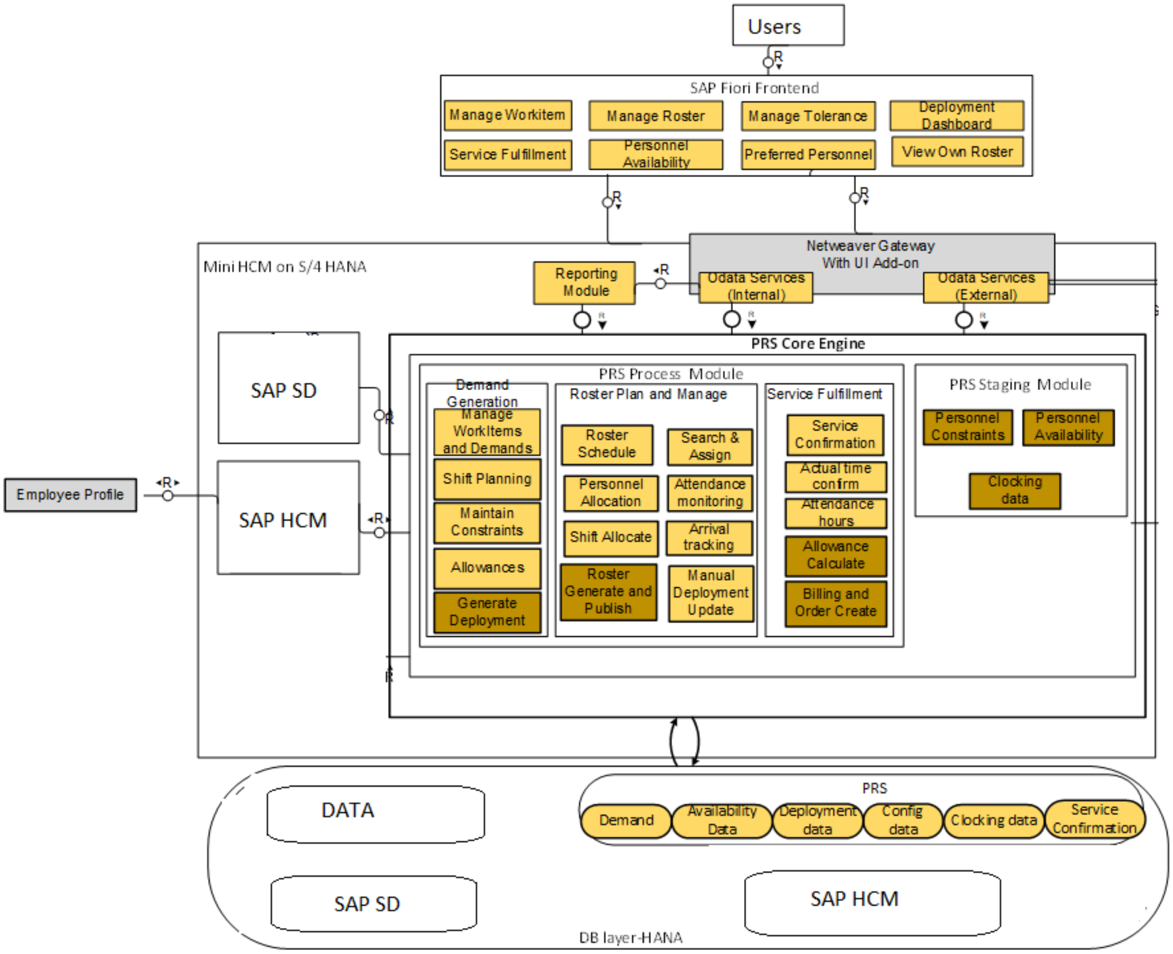


Figure 3.1: PRS Overall Architecture

The PRS solution shall be developed in S/4 HANA on-premise system. The system landscape used shall have development, quality and production system.

The key building blocks are described below:

i. PRS Frontend: Frontend User apps on SAP Fiori Launchpad accessing data via Odata Services invoking backend Business layer.
ii. Integration with Employee information: Employee information shall be captured in training and recruitment module.
iii. PRS backend: Backend Core solution is split into:
    a. Process Module:
        - Demand – Consisting of handling Work items, Demands and deployment generation
        - Roster – Consisting of Roster algorithms, Personnel assignment, Shift Planning, Publishing of Roster plans, Daily Deployment manage, Manual Updates, Arrival monitoring and tracking.
        - Service Fulfillment – Consisting of Actual, update Attendance hours and Allowance calculation.
    b. Staging Module:
        - Master Data staging – HCM Master data and Info Types will be staged into Constraint matrix and Availability data.
        - Data staging for Web Service interfaces – REST based Web services will be consumed from PRS backend for data.

3.2 UI Information Architecture

The user interface for the PRS solution consists of custom Fiori apps and SAPGUI screens. These are made available to business users in a role-based access.

3.2.1 User Interface

Below is list of user interfaces and their functionality.

	User Interface	Type	Functions
1	Manage Work Items	Custom Fiori app	Demand Planning
2	Manage Rosters	Custom Fiori app	Viewing Rosters, Making shift assignments
3	Manage Deployments	Custom Fiori app	Displays real-time deployment information, making shift assignments
4	Service Confirmation	Custom Fiori app	Displaying and entering clocking times, payable/billable times, job confirmation.
5	Blacklist/Whitelist	Custom Fiori app	Maintaining a black/white-listing of personnel at Client/Plant/Demand-level
6	Allowance Approval	Custom Fiori app	Approving allowances assigned against demands.
7	Personnel Availability	Custom Fiori app	Viewing availability of personnel of a Deployment Group
8	Unavailability Report	Custom Fiori app	Report listing unplanned absences.
9	My Roster	Custom Fiori app	Viewing own roster and payable hours for employee.
10	Team Roster	Custom Fiori app	Viewing roster and payable hours of a team member by team lead.

3.2.2 Overview of PRS Functionality

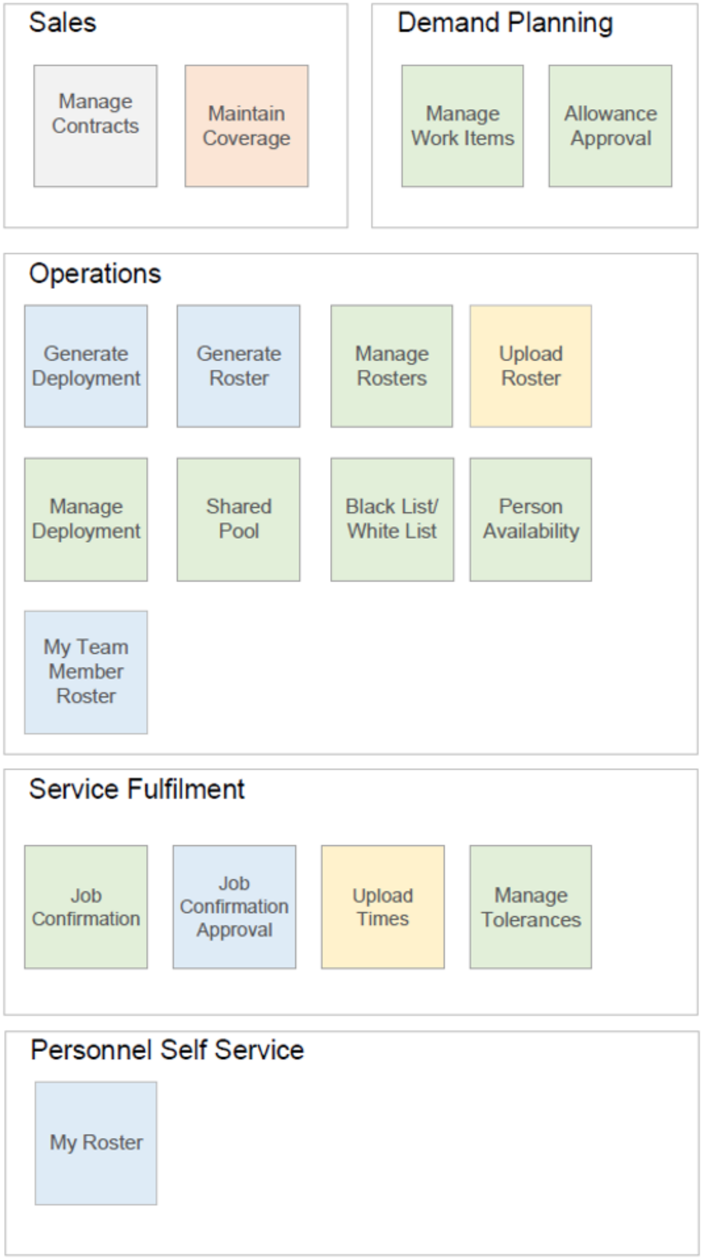


Figure 3.2: PRS Functionality

3.2.3 Manage Work Items

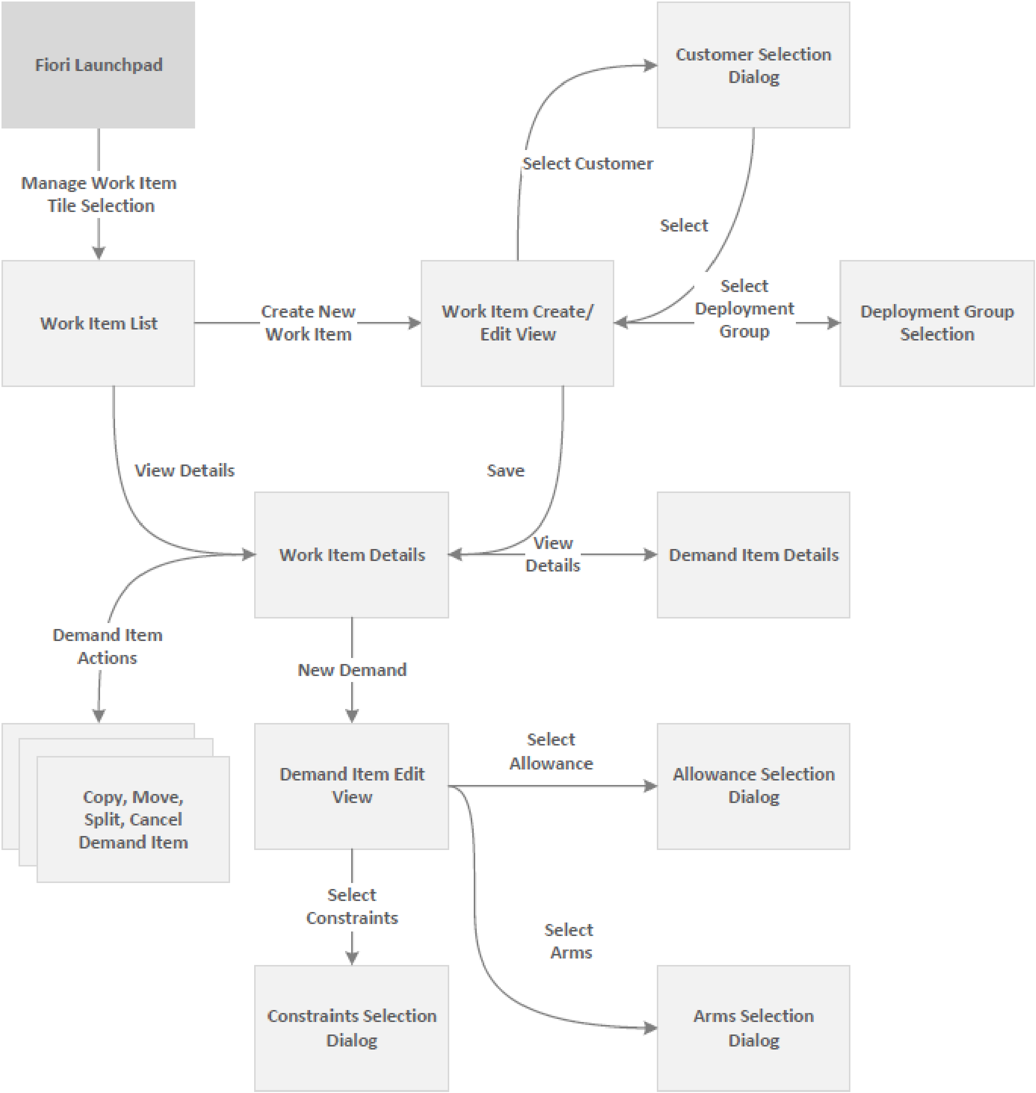


Figure 3.3: Manage Work Items

3.2.4 Generate Deployment

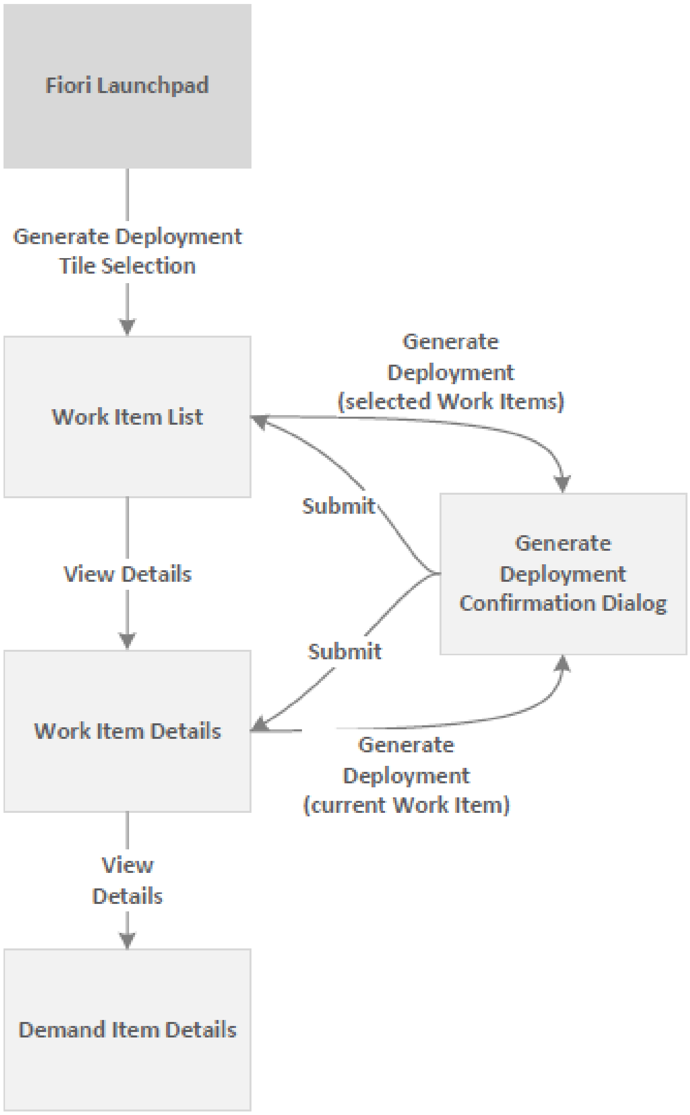

Fiori Launchpad
Generate Deployment Tile Selection
Work Item List
Generate Deployment (selected Work Items)
Submit
Generate Deployment Confirmation Dialog
View Details
Submit
Work Item Details
Generate Deployment (current Work Item)
View Details
Demand Item Details

Figure 3.4: Generate Deployment

3.2.5 Preferred Personnel

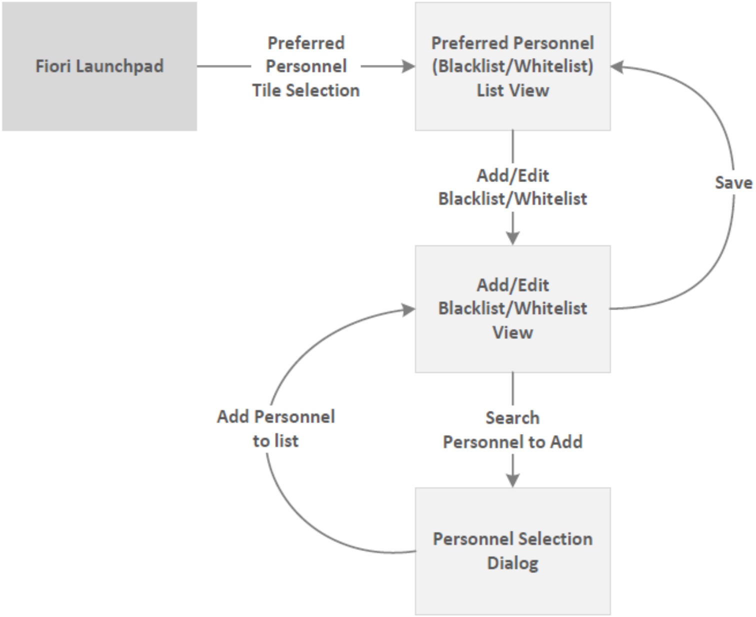

Fiori Launchpad
Preferred Personnel Tile Selection
Preferred Personnel (Blacklist/Whitelist) List View
Add/Edit Blacklist/Whitelist
Save
Add/Edit Blacklist/Whitelist View
Add Personnel to list
Search Personnel to Add
Personnel Selection Dialog

Figure 3.5: Preferred Personnel

3.2.6 Generate Roster

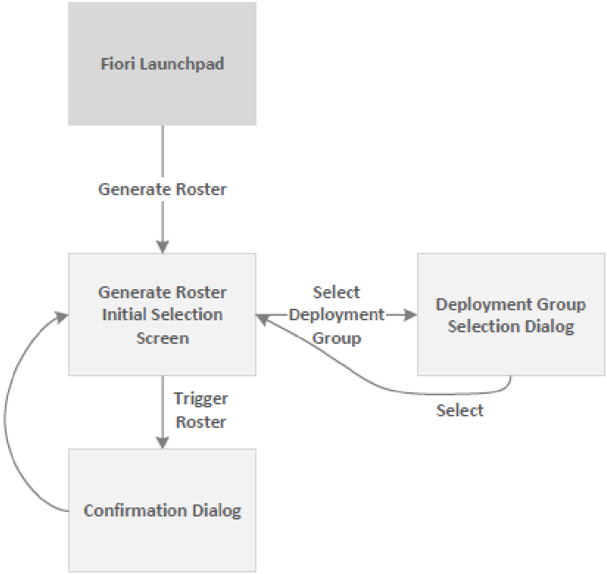


Figure 3.6: Generate Roster

3.2.7 Manage Roster

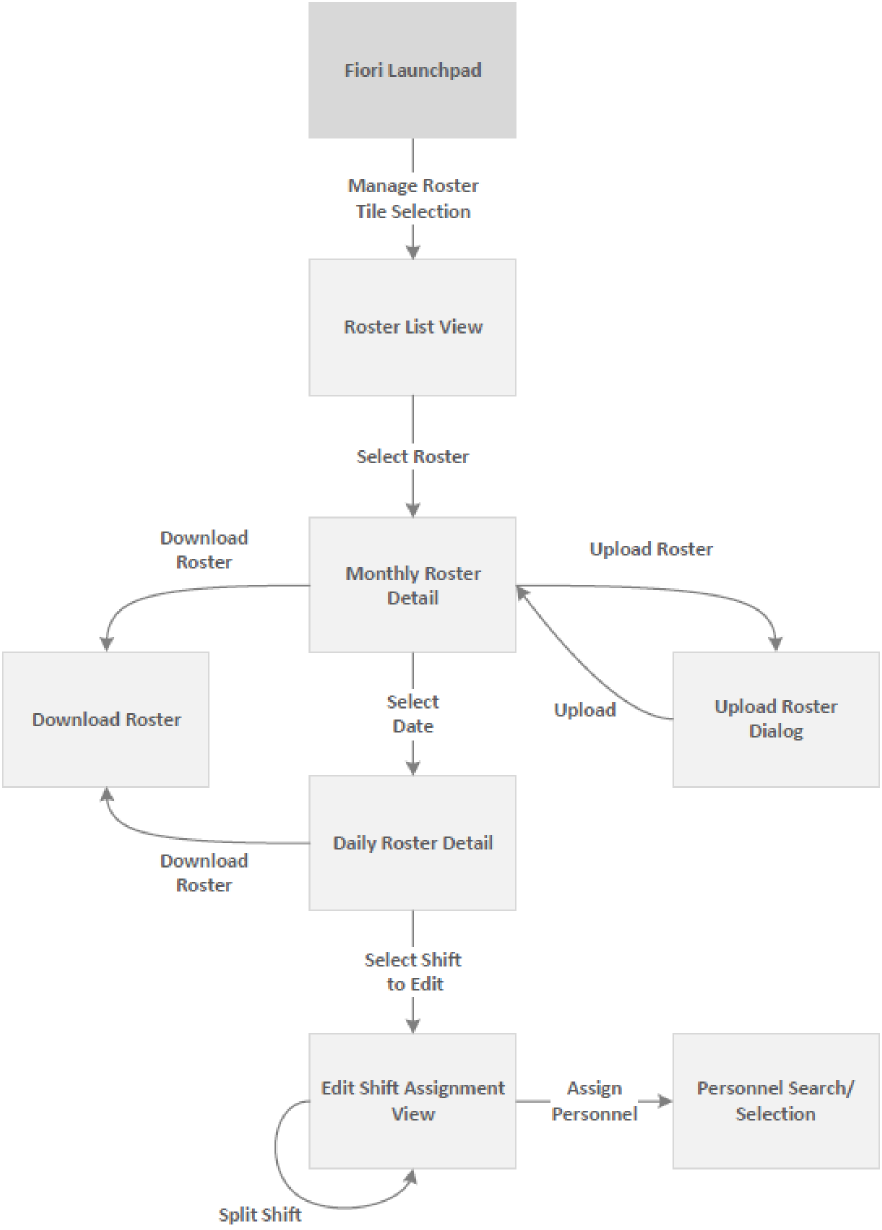


Figure 3.7: Manage Roster

3.2.8 Manage Deployment

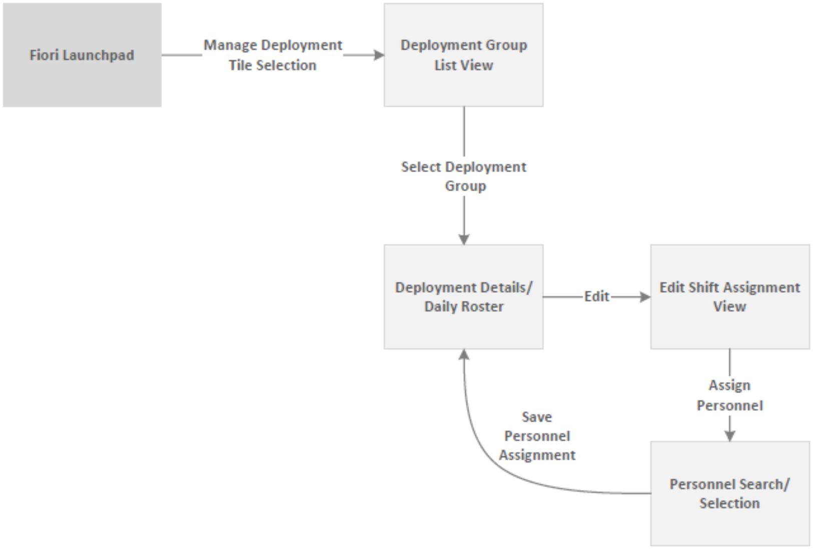

Fiori Launchpad
Manage Deployment Tile Selection
Deployment Group List View
Select Deployment Group
Deployment Details/ Daily Roster
Edit
Edit Shift Assignment View
Assign Personnel
Save Personnel Assignment
Personnel Search/ Selection

Figure 3.8: Manage Deployment

3.2.9 Job Confirmation

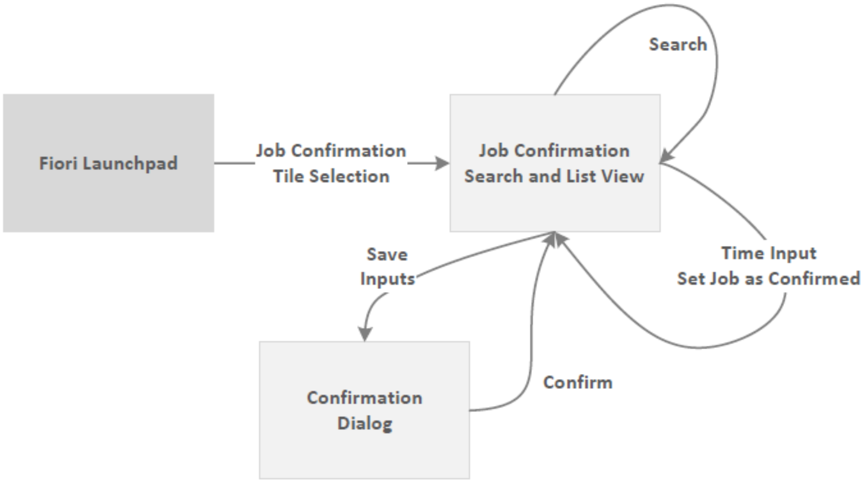


Figure 3.9: Job Confirmation

3.2.10 Job Confirmation Approval

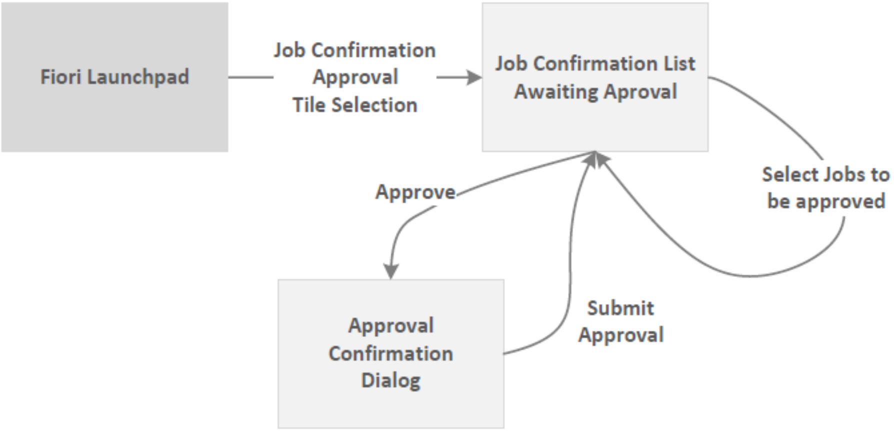

Fiori Launchpad
Job Confirmation Approval Tile Selection
Job Confirmation List Awaiting Aproval
Select Jobs to be approved
Approve
Approval Confirmation Dialog
Submit Approval

Figure 3.10: Job Confirmation Approval

3.2.11 Approve Allowances

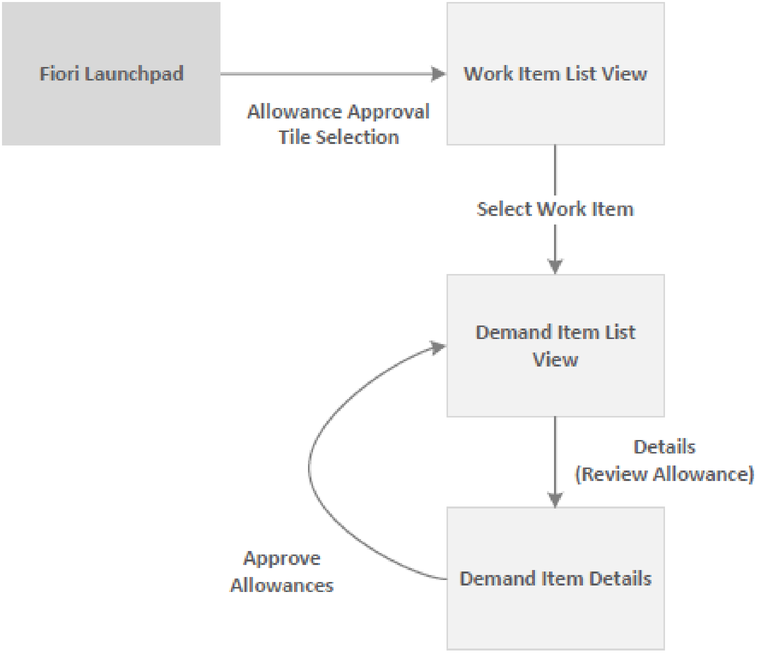


Figure 3.11: Approved Allowances

3.2.12 Personnel Availability

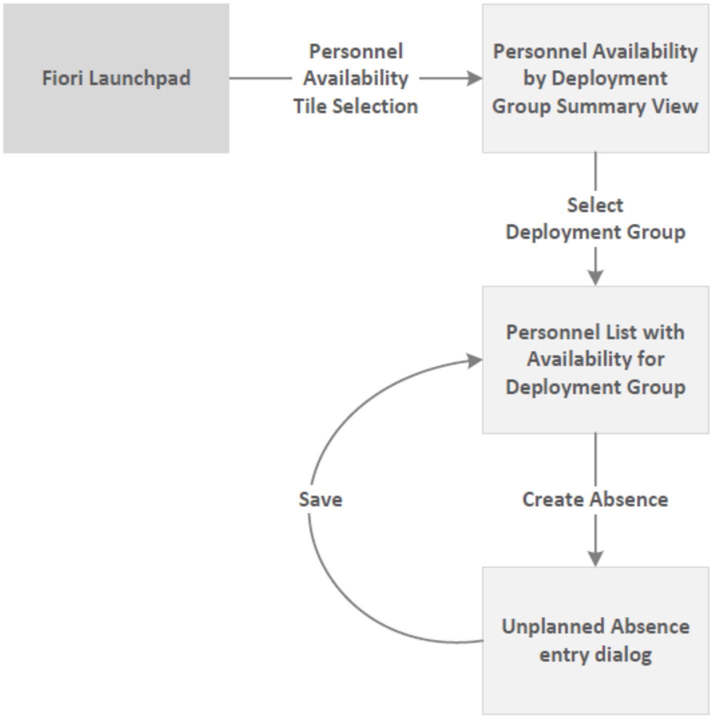

Fiori Launchpad
Personnel Availability Tile Selection
Personnel Availability by Deployment Group Summary View
Select Deployment Group
Personnel List with Availability for Deployment Group
Save
Create Absence
Unplanned Absence entry dialog

Figure 3.12: Personnel Availability

3.2.13 My Roster

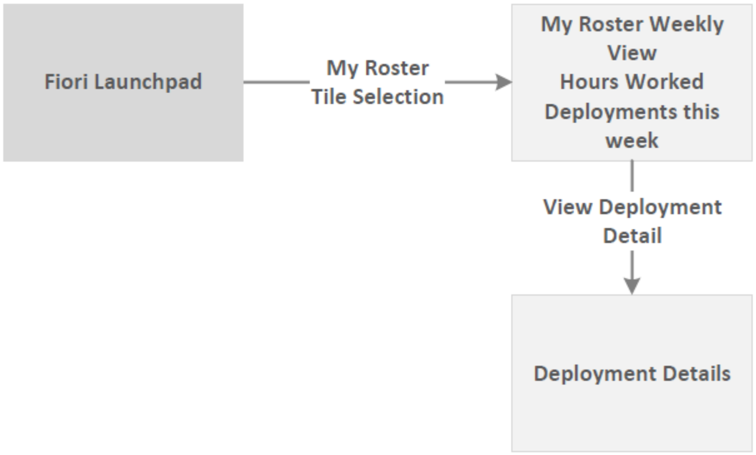


Figure 3.13: My Roster

3.2.14 Team Roster

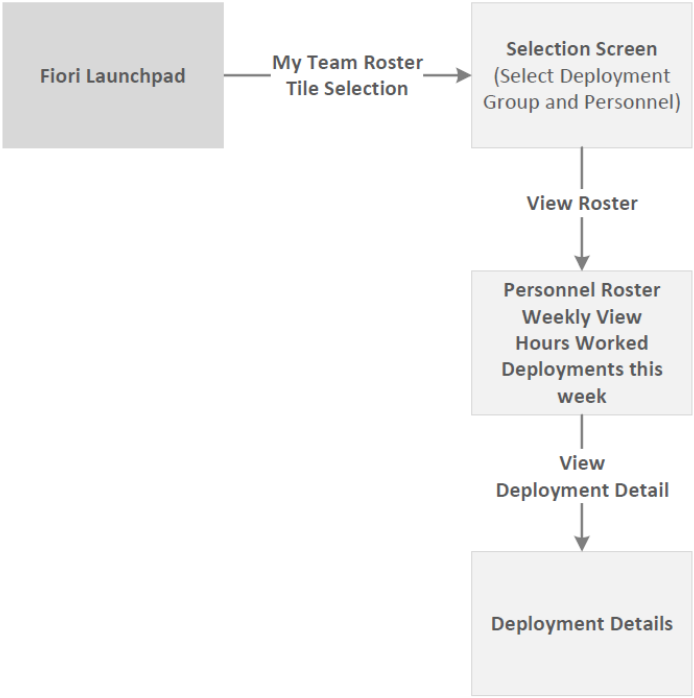

Fiori Launchpad
My Team Roster Tile Selection
Selection Screen (Select Deployment Group and Personnel)
View Roster
Personnel Roster Weekly View Hours Worked Deployments this week
View Deployment Detail
Deployment Details

Figure 3.14: Team Roster

3.3 Architecture Decisions

i. ABAP Object Oriented programming language should be used with standard ABAP features.

ii. User shall configure and maintain all the roles/authorizations.

iii. PRS background job frequency shall be defined by user.

iv. UI5 shall be the UI technology as far as possible.

3.4 OData Testing Automation

The excel workbook consisting of OData requests can directly be converted into Postman-readable JSON file. This JSON file can be imported as a collection in Postman. There are two different scripts for this purpose. The first script ‘main.js’ creates a single file for all the apps. The second script ‘main.1.js’ creates separate collections for each app.

3.4.1 OData Requests Format

The excel workbook consists of the OData requests. The workbook is divided into worksheets. First worksheet consists of two columns. The first column consists the list of applications. The second column consists of the path to respective apps.

Remaining worksheets consist of description of request and path of that request. In case of POST/ MERGE requests, Payload column will also be checked.

Currently, the script supports three types of requests:

i. GET
ii. POST
iii. MERGE

3.4.2 Process to convert Excel workbook to Postman Collection

Prerequisite: Node must be installed

1. Login to FioriLaunchpad
2. Get x-csrf-token, Cookie from network
3. Update the value of variables CSRFToken, AuthCookie in main.js and main.1.js
4. To get single Collection for all applications:
    - open terminal in VSCode.
    - node main.js > postman.json
5. To get different Collections for all applications:

    - open terminal in VSCode.
    - node main.1.js
    - different json files will be created in the Folder 'Collections'

6. Open postman
7. Click on Import
8. For single Collection: Select file postman.json.

   For different applications: Select folder Collections
9. Under the Collections tab, Run the Collection
10. The test cases will pass if Status Code: 200(GET), 202(POST), else fail.

3.4.3 Algorithm for OData Testing Automation

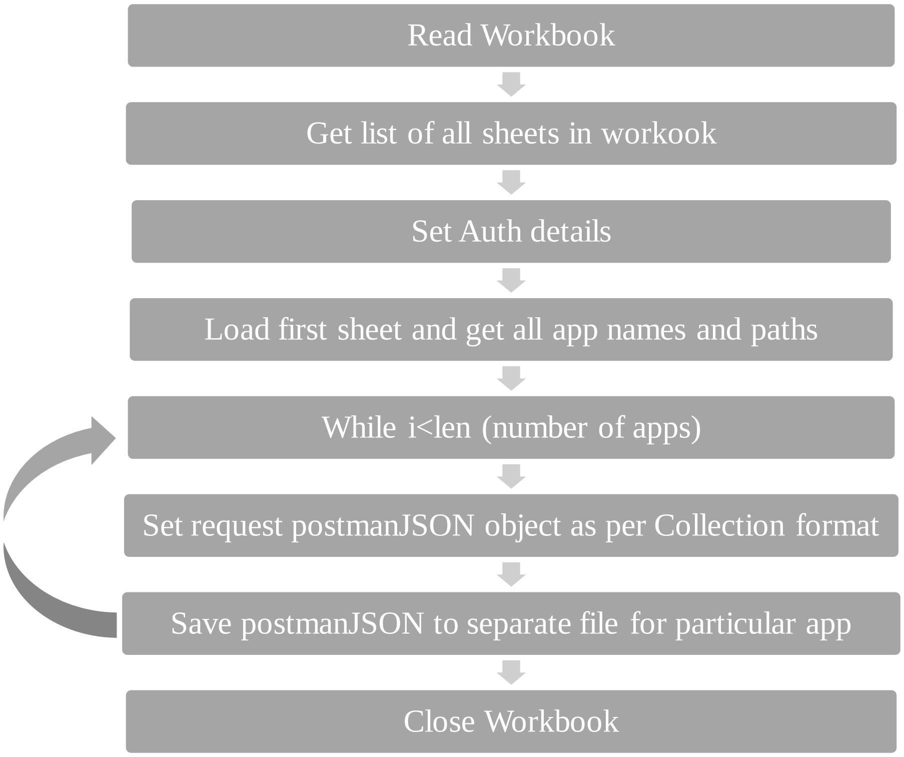
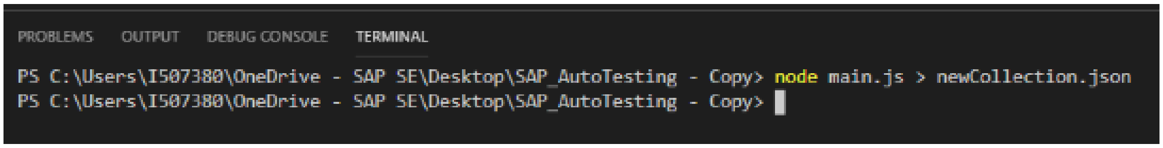
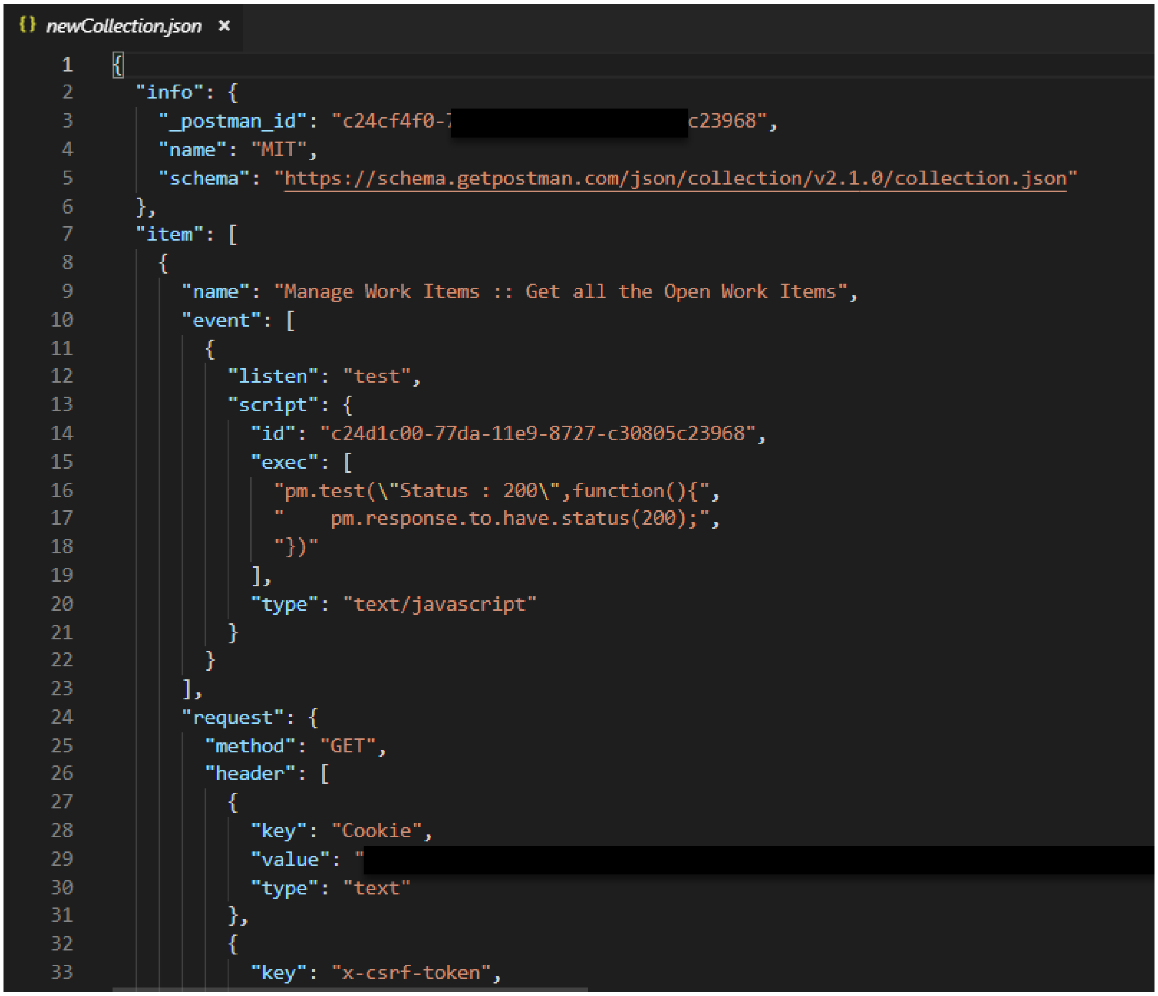

Read Workbook
Get list of all sheets in workook
Set Auth details
Load first sheet and get all app names and paths
While i<len (number of apps)
Set request postmanJSON object as per Collection format
Save postmanJSON to separate file for particular app
Close Workbook

Figure 3.15: OData Testing Automation Algorithm



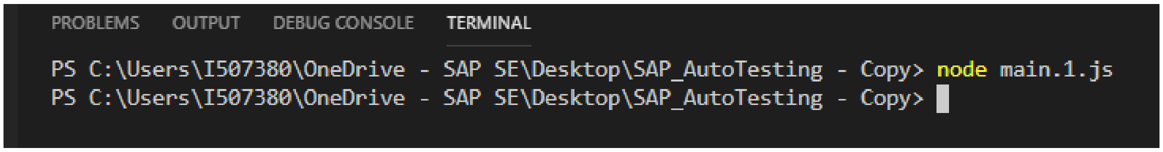
```
PS C:\Users\I507380\OneDrive - SAP SE\Desktop\SAP_AutoTesting - Copy> node main.js > newCollection.json
PS C:\Users\I507380\OneDrive - SAP SE\Desktop\SAP_AutoTesting - Copy> 
```

Figure 3.16: Creating single collection for all applications



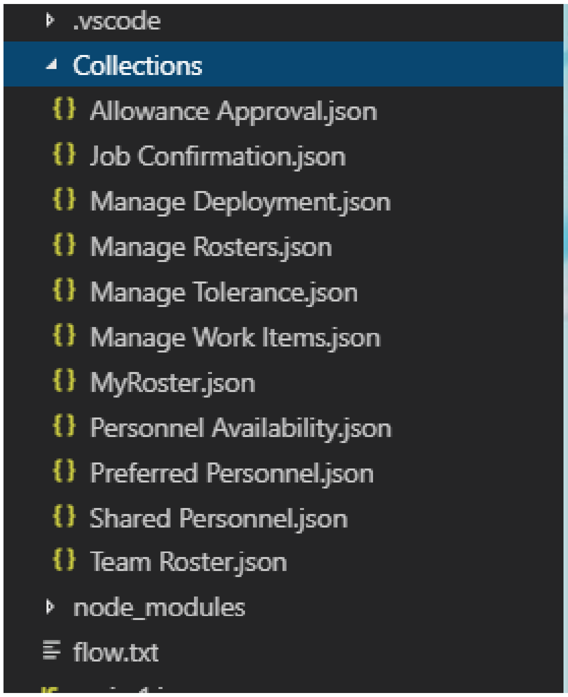
```
{
  "info": {
    "_postman_id": "c24cf4f0-[redacted]c23968",
    "name": "MIT",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
  },
  "item": [
    {
      "name": "Manage Work Items :: Get all the Open Work Items",
      "event": [
        {
          "listen": "test",
          "script": {
            "id": "c24d1c00-77da-11e9-8727-c30805c23968",
            "exec": [
              "pm.test(\"Status : 200\",function(){",
              "    pm.response.to.have.status(200);",
              "})"
            ],
            "type": "text/javascript"
          }
        }
      ],
      "request": {
        "method": "GET",
        "header": [
          {
            "key": "Cookie",
            "value": "[redacted]
            "type": "text"
          },
          {
            "key": "x-csrf-token",
```

Figure 3.17: Postman-readable JSON format



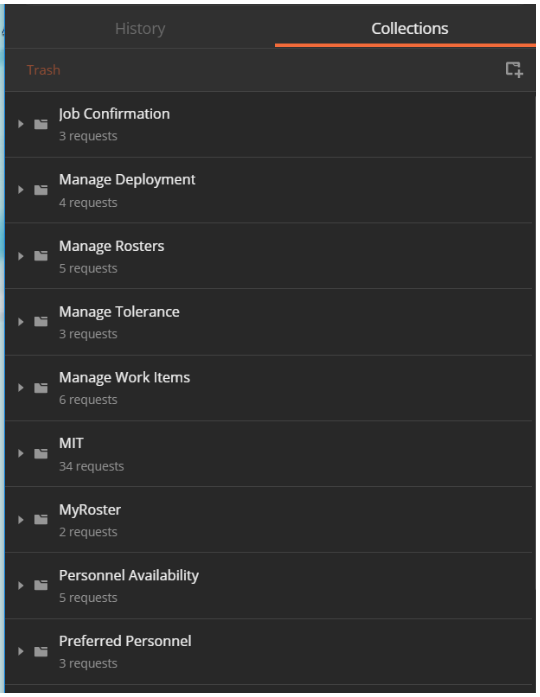
```
PS C:\Users\I507380\OneDrive - SAP SE\Desktop\SAP_AutoTesting - Copy> node main.1.js
PS C:\Users\I507380\OneDrive - SAP SE\Desktop\SAP_AutoTesting - Copy> 
```

Figure 3.18: Creating different collection for all applications



.vscode
Collections
Allowance Approval.json
Job Confirmation.json
Manage Deployment.json
Manage Rosters.json
Manage Tolerance.json
Manage Work Items.json
MyRoster.json
Personnel Availability.json
Preferred Personnel.json
Shared Personnel.json
Team Roster.json
node_modules
flow.txt

Figure 3.19: Different Postman-readable JSON files for different application



Job Confirmation
3 requests
Manage Deployment
4 requests
Manage Rosters
5 requests
Manage Tolerance
3 requests
Manage Work Items
6 requests
MIT
34 requests
MyRoster
2 requests
Personnel Availability
5 requests
Preferred Personnel
3 requests

Figure 3.20: JSON files imported as Collections

These collections can simply be run at a single click in Postman. This helps in saving time a lot. For example, if testing 34 requests needed 17 minutes (assuming 30 seconds for each request), then, using this script, it’ll just take 2 minutes because the user just has to change Cookie and x-csrf-token from network call, run the script and import the collection.

Chapter – 4

Performance Analysis

This chapter contains information about performance analysis of PRS. In order to analyse the performance of the project, various requirements have been defined. All these requirements are analysed to measure the performance of PRS.

Futhermore, PRS has gone through FIT during each sprint. After development, the project has gone through MIT and Regression Testing. Each feature and functionality of the project has been tested thoroughly to make sure that the quality of system is not compromised. After passing these testing phases, the project has also gone through UAT.

Requirement	**Requirement Description**	**Result**
1. General		
FIT	The various features of applications must be tested while development for functionality.	PASS
MIT	The modules are tested after integration to make sure that one module hasn't affected the other module.	PASS
Regression Tested	The testing is repeated again after fixing the bugs found in MIT.	PASS
UAT	The user itself tests PRS to make sure it works fine and stands up to their expectations.	PASS
2. Persistence Layer		
Performance-optimized accesses to persistence layer	This includes: - Appropriate indexes - Buffers and caches - No redundant accesses	PASS
3. Application Layer		
Parallel processing enabled	Scalable parallel processing and load balancing mechanisms have been employed in your product. Locks have no impact on parallel processing.	PASS

Linear dependency only	The CPU and memory required to process a given number of records depends linearly on the amount of processed data. Please test that there are only linear dependencies.	PASS
4. Communication		
Number of synchronous round trips	There shall be no more than 2 synchronous round trips between front end and application layer per user interaction step.	PASS
Bytes Transferred	Transferred bytes between frontend and application layer shall be kept at a minimum	PASS
End-to-End Dialog Response Time	Average end-to-end dialog response time per interaction step below 2 seconds. Only applicable for critical processes with user interaction. avg. End-to-end response time must be below 2 seconds.	PASS

Table 4.1: Performance Analysis Report

Chapter – 5

CONCLUSIONS

5.1 Conclusions

PRS is a one-stop-shop rostering solution that helps in deploying personnel automatically. This saves the manual tasks as the organization doesn't have to use excel-based tools. PRS simplies tasks of organization and therefore, helps them to save time and extra effort.

As per the Performance Analysis, the system doesn not suffer from performance issues in case of high loads. The system is highly configurable as per the needs of customers. Constraints, business rules, allowances, etc. can be modified as per the need of the customer. PRS can be accessed through the web anywhere and at any device. The live dashboards in Manage Deployment application helps in monitoring the deployment continuously. Also, roles have been made for different users as per the accessibility matrix. The roles decide the apps that will be visible to the user and what reports the user can run in the background.

5.2 Future Scope

The testing can be further automated using tools like Selenium in order to save time and extra effort.

The time taken in rostering can further be reduced.

References:

[1] A.T. Ernst, H. Jiang, M. Krishnamoorthy and D. Sier, “Staff scheduling and rostering: A review of applications, methods and models”, European Journal of Operational Research, CISRO Mathematical and Information Sciences, 2004 (pp. 3-4)

[2] Emir Hüseyin Özder, Evrencan Özcan andTamer Eren, Staff Task-Based Shift Scheduling Solution with an ANP and Goal Programming Method in a Natural Gas Combined Cycle Power Plant, MDPI, 2019 (pp. 1)

[3] Georgios Tsaramirsis and Abdullah Basahel, Scrum to manage humans and intelligent systems, IEEE, 2016 (pp. 1-4)