**Master Planner Portal**

Design Documentation

**Team**

|  |  |
| --- | --- |
| Manager | Dew Oh  dewohyy@singtel.com |
| Advisor | Jialong Leong  jialong.leong@singtel.com  Wah Wah Tun  wahwah.tun@singtel.com  Andy Maung  andy.maung@singtel.com |
| Development | Chester Yee  chester.yee@singtel.com |
| Deployment | Howard Wong  howard.wong@singtel.com  Rick Wang  rick.wang@singtel.com |

|  |  |  |
| --- | --- | --- |
| OS | CentOS | 7.x |
| Database | MySQL | 8.0.18 |
|  | PHP | 7.3.9 |
|  | Apache | 2.4.41 |
| Technology | HTML |  |
|  | CSS |  |
|  | JS |  |
|  | SQL |  |
| Libraries / Frameworks | jQuery |  |
|  | Bootstrap |  |
|  | Popper |  |
|  | FontAwesome |  |
| Infrastructure | NFVi |  |
|  | SMTP |  |
|  | DNS |  |

**Specifications**

**Request Structure**

All request types comes in a set of 3 files:

**[requestType]Form.php**

**[requestType]FormProcess.php**

**[requestType]FormStatus.php**

**[requestType]ChangeToInstallationInProgress.php**

**[requestType]ChangeToCompleted.php**

[requestType] denotes the type of request — Space, Power, FDF, SSU, Cable Tray, General and MMR

**[requestType]Form.php**

This file contains the structure of the form itself.

For space and power requests, the dynamic form is being divided into 3 fieldsets*.* The first fieldset requires the Requestor to input their personal information followed by the number of racks in the second fieldset which then generates the number of rack fields in the third fieldset based on value that the Requestor has selected in the second fieldset.

For SSU, FDF, Cable Tray, General and MMR requests, the dynamic form is being divided into 2 fieldsets*.* The first fieldset requires the Requestor to input their personal information followed by the technical information in the second fieldset.

**[requestType]FormProcess.php**

This file processes the data for the backend.

Data from the fields are posted to the database.

**[requestType]FormStatus.php**

This file retrieves the data from the database and shows the Requestor the information that is required.

The status page is being divided into 3 main sections.

The left panel shows the general information of the request itself.

The right pane shows the Service Level Agreement (SLA) timeline component and actions component. In the SLA component. The timeline component shows the respective states with the timing once the state has been activated. There is also a countdown timer at the situated at the footer of the component itself that shows the number of days left to change to a different state.

The actions component shows state specific messages upon a change in state. For states like *Assigned* and *Installation in Progress*, the Requestor would be required to complete a set of tasks firsts before the change of state is permitted. For the other states, the Planner would be the one modifying that is subjected to approval and satisfaction of verification.

The metrics component shows the key information for the request itself.

**[requestType]ChangeToInstallationInProgress.php**

This file updates the status of the request to Installation in Progress.

**[requestType]ChangeToCompleted.php**

This file updates the status of the request to Completed.

Requestor is required to upload a set of images when they trigger the change in state to Completed

MPP Requestor

The MPP requestor app is being divided into 3 pages — *Request*, *Status*, *Guide*.

Request Page

In the *Request* page, you will find 3 cards containing a series of request types available for the Requestor to choose from.

When the Requestor clicks on the card itself, they will be directed to the specific form of the particular request. The only exception is when the Requestor clicks on the *Others* card, a dropdown list will be shown with a list of other requests made available.

Space Request

Requires

filepath.php

spaceForm.php

spaceFormProcess.php

The dynamic form is being into 3 *fieldsets.* The first fieldset requires the Requestor to input their personal information followed by the number of racks in the second fieldset which then generates the number of rack fields in the third fieldset based on value that the Requestor has selected in the second fieldset.

After each request is submitted, a *requestId* will be generated for the Requestor to track its status and progress of the request itself.

**filepath.php**

Contains the credentials for login into database

**spaceForm.php**

Contains the dynamic form with 3 fieldsets, each with their respective fields.

For the first fieldset, the fields include the following: Requestor Name, Requestor Email, Requestor Department and Requestor Reason.

For the second fieldset, the only field is the Number of Racks.

For the third fieldset, the fields include the following: Power Type, Rack Type, Installation Date (Start), Completion Date (End), Preferred Exchange, Preferred Room, Power Consumption, Breaker Quantity[\*], Breaker Size[\*], Rack Length[\*], Rack Breadth[\*], Rack Height[\*], Rack Weight[\*]

[\*] denotes the number of rack selected in the second fieldset varying from 1 to 5.

Space Request

| Field | Id |
| --- | --- |
| Requestor Name | requestorName |
| Requestor Email | requestorEmail |
| Requestor Department | requestorDepartment |
| Requestor Reason | requestorReason |

| Field | Id |
| --- | --- |
| Requestor Input | |
| Power Type | powerType |
| Rack Type | rackType |
| Installation Date (Start) | startDate |
| Completion Date (End) | endDate |
| Preferred Exchange | exchange |
| Preferred Room | room |
| Power Consumption[\*] | powerConsumption[\*] |
| Breaker Quantity[\*] | breakerQuantity[\*] |
| Breaker Size[\*] | breakerSize[\*] |
| Rack Length[\*] | rackSizeLength[\*] |
| Rack Breadth[\*] | rackSizeBreadth[\*] |
| Rack Height[\*] | rackSizeHeight[\*] |
| Rack Weight[\*] | rackSizeWeight[\*] |
| Upload Images | requestorFileUpload |
| Background services | requestTimestamp |
| requestStatus |
| requestStatusAcknowledged |
| requestStatusAssigned |
| requestStatusInProgress |
| requestStatusCompleted |
| requestStatusClosed |
| Planner Input | |
| Rack Location | rackLocation[\*] |
| Sub PDU Feed[\*]A1 | subPdu[\*]FeedA1 |
| Sub PDU Feed[\*]A2 | subPdu[\*]FeedA2 |
| Sub PDU Feed[\*]B1 | subPdu[\*]FeedB1 |
| Sub PDU Feed[\*]B2 | subPdu[\*]FeedB2 |
| Breaker Name[\*]Feed A1 | breakerName[\*]FeedA1 |
| Breaker Name[\*]Feed A2 | breakerName[\*]FeedA2 |
| Breaker Name[\*]Feed B1 | breakerName[\*]FeedB1 |
| Breaker Name[\*]Feed B2 | breakerName[\*]FeedB2 |
| Upload Floorplan | adminFileUpload |
| Background services | remarks |

Power Request

Requires

filepath.php

powerForm.php

powerFormProcess.php

The dynamic form is being into 3 *fieldsets.* The first fieldset requires the Requestor to input their personal information followed by the number of racks in the second fieldset which then generates the number of rack fields in the third fieldset based on value that the Requestor has selected in the second fieldset.

After each request is submitted, a *requestId* will be generated for the Requestor to track its status and progress of the request itself.

**filepath.php**

Contains the credentials for login into database

**powerForm.php**

Contains the dynamic form with 3 fieldsets, each with their respective fields.

For the first fieldset, the fields include the following: Requestor Name, Requestor Email, Requestor Department and Requestor Reason.

For the second fieldset, the only field is the Number of Racks.

For the third fieldset, the fields include the following: Power Type, Installation Date (Start), Completion Date (End), Exchange, Room, Rack Location, Power Consumption[\*], Breaker Quantity[\*], Breaker Size[\*]

[\*] denotes the number of rack selected in the second fieldset varying from 1 to 5.

Power Request

| Field | Id |
| --- | --- |
| Requestor Name | requestorName |
| Requestor Email | requestorEmail |
| Requestor Department | requestorDepartment |
| Requestor Reason | requestorReason |

| Field | Id |
| --- | --- |
| Requestor Input | |
| Power Type | powerType |
| Installation Date (Start) | startDate |
| Completion Date (End) | endDate |
| Exchange | exchange |
| Room | room |
| Power Consumption[\*] | powerConsumption[\*] |
| Breaker Quantity[\*] | breakerQuantity[\*] |
| Breaker Size[\*] | breakerSize[\*] |
| Rack Location[\*] | rackLocation[\*] |
| Upload Images | requestorFileUpload |
| Background services | requestTimestamp |
| requestStatus |
| requestStatusAcknowledged |
| requestStatusAssigned |
| requestStatusInProgress |
| requestStatusCompleted |
| requestStatusClosed |
| Planner Input | |
| Sub PDU Feed[\*]A1 | subPdu[\*]FeedA1 |
| Sub PDU Feed[\*]A2 | subPdu[\*]FeedA2 |
| Sub PDU Feed[\*]B1 | subPdu[\*]FeedB1 |
| Sub PDU Feed[\*]B2 | subPdu[\*]FeedB2 |
| Breaker Name[\*]Feed A1 | breakerName[\*]FeedA1 |
| Breaker Name[\*]Feed A2 | breakerName[\*]FeedA2 |
| Breaker Name[\*]Feed B1 | breakerName[\*]FeedB1 |
| Breaker Name[\*]Feed B2 | breakerName[\*]FeedB2 |
| Upload Floorplan | adminFileUpload |
| Background services | remarks |

SSU Request

Requires

filepath.php

ssuForm.php

ssuFormProcess.php

The dynamic form is being divided into 2 *fieldsets.* The first fieldset requires the Requestor to input their personal information followed by the technical information.

After each request is submitted, a *requestId* will be generated for the Requestor to track its status and progress of the request itself.

**filepath.php**

Contains the credentials for login into database

**ssuForm.php**

Contains the dynamic form with 2 fieldsets, each with their respective fields.

For the first fieldset, the fields include the following: Requestor Name, Requestor Email, Requestor Department and Requestor Reason.

For the second fieldset, the fields include the following: Number of Ports, Transmission Type, Interfacing Type, Installation Date (Start), Completion Date (End), Exchange, Room.

SSU Request

| Field | Id |
| --- | --- |
| Requestor Name | requestorName |
| Requestor Email | requestorEmail |
| Requestor Department | requestorDepartment |
| Requestor Reason | requestorReason |

| Field | Id |
| --- | --- |
| Requestor Input | |
| Number of Ports | numberOfPorts |
| Transmission Type | transmissionType |
| Interfacing Type | interfacingType |
| Installation Date (Start) | startDate |
| Completion Date (End) | endDate |
| Exchange | exchange |
| Room | room |
| Upload Images | requestorFileUpload |
| Background services | requestTimestamp |
| requestStatus |
| requestStatusAcknowledged |
| requestStatusAssigned |
| requestStatusInProgress |
| requestStatusCompleted |
| requestStatusClosed |
| Planner Input | |
| Background services | remarks |

FDF Request

Requires

filepath.php

fdfForm.php

fdfFormProcess.php

The dynamic form is being divided into 2 *fieldsets.* The first fieldset requires the Requestor to input their personal information followed by the technical information.

After each request is submitted, a *requestId* will be generated for the Requestor to track its status and progress of the request itself.

**filepath.php**

Contains the credentials for login into database

**fdfForm.php**

Contains the dynamic form with 2 fieldsets, each with their respective fields.

For the first fieldset, the fields include the following: Requestor Name, Requestor Email, Requestor Department and Requestor Reason.

For the second fieldset, the fields include the following: Number of Ports, Number of Cable Ties, Installation Date (Start), Completion Date (End), Exchange, Room.

FDF Request

| Field | Id |
| --- | --- |
| Requestor Name | requestorName |
| Requestor Email | requestorEmail |
| Requestor Department | requestorDepartment |
| Requestor Reason | requestorReason |

| Field | Id |
| --- | --- |
| Requestor Input | |
| Number of Ports | numberOfPorts |
| Number of Cable Ties | numberOfCableTies |
| Installation Date (Start) | startDate |
| Completion Date (End) | endDate |
| Exchange | exchange |
| Room | room |
| Upload Images | requestorFileUpload |
| Background services | requestTimestamp |
| requestStatus |
| requestStatusAcknowledged |
| requestStatusAssigned |
| requestStatusInProgress |
| requestStatusCompleted |
| requestStatusClosed |
| Planner Input | |
| Background services | remarks |

Cable Tray Request

Requires

filepath.php

cableTrayForm.php

cableTrayFormProcess.php

The dynamic form is being divided into 2 *fieldsets.* The first fieldset requires the Requestor to input their personal information followed by the technical information.

After each request is submitted, a *requestId* will be generated for the Requestor to track its status and progress of the request itself.

**filepath.php**

Contains the credentials for login into database

**cableTrayForm.php**

Contains the dynamic form with 2 fieldsets, each with their respective fields.

For the first fieldset, the fields include the following: Requestor Name, Requestor Email, Requestor Department and Requestor Reason.

For the second fieldset, the fields include the following: Rack Location, FDF Rack Location, Installation Date (Start), Completion Date (End), Exchange, Room.

Cable Tray Request

| Field | Id |
| --- | --- |
| Requestor Name | requestorName |
| Requestor Email | requestorEmail |
| Requestor Department | requestorDepartment |
| Requestor Reason | requestorReason |

| Field | Id |
| --- | --- |
| Requestor Input | |
| Rack Location | rackLocation |
| FDF Rack Location | fdfRackLocation |
| Installation Date (Start) | startDate |
| Completion Date (End) | endDate |
| Exchange | exchange |
| Room | room |
| Upload Images | requestorFileUpload |
| Background services | requestTimestamp |
| requestStatus |
| requestStatusAcknowledged |
| requestStatusAssigned |
| requestStatusInProgress |
| requestStatusCompleted |
| requestStatusClosed |
| Planner Input | |
| Background services | remarks |

General Request

Requires

filepath.php

generalForm.php

generalFormProcess.php

The dynamic form is being divided into 2 *fieldsets.* The first fieldset requires the Requestor to input their personal information followed by the technical information.

After each request is submitted, a *requestId* will be generated for the Requestor to track its status and progress of the request itself.

**filepath.php**

Contains the credentials for login into database

**generalForm.php**

Contains the dynamic form with 2 fieldsets, each with their respective fields.

For the first fieldset, the fields include the following: Requestor Name, Requestor Email, Requestor Department and Requestor Reason.

For the second fieldset, the fields include the following: Query.

General Request

| Field | Id |
| --- | --- |
| Requestor Name | requestorName |
| Requestor Email | requestorEmail |
| Requestor Department | requestorDepartment |
| Requestor Reason | requestorReason |

| Field | Id |
| --- | --- |
| Requestor Input | |
| Query | query |
| Background services | requestTimestamp |
| requestStatus |
| requestStatusAcknowledged |
| requestStatusAssigned |
| requestStatusInProgress |
| requestStatusCompleted |
| requestStatusClosed |
| Planner Input | |
| Background services | remarks |

MMR Request

Requires

filepath.php

mmrForm.php

mmrFormProcess.php

The dynamic form is being divided into 2 *fieldsets.* The first fieldset requires the Requestor to input their personal information followed by the technical information.

After each request is submitted, a *requestId* will be generated for the Requestor to track its status and progress of the request itself.

filepath.php

**mmrForm.php**

Contains the dynamic form with 2 fieldsets, each with their respective fields.

For the first fieldset, the fields include the following: Requestor Name, Requestor Email, Requestor Department and Requestor Reason.

For the second fieldset, the fields include the following: Query.

**filepath.php**

Contains the credentials for login into database

MMR Request

| Field | Id |
| --- | --- |
| Requestor Name | requestorName |
| Requestor Email | requestorEmail |
| Requestor Department | requestorDepartment |
| Requestor Reason | requestorReason |

| Field | Id |
| --- | --- |
| Requestor Input | |
| Request | request |
| Background services | requestTimestamp |
| requestStatus |
| requestStatusAcknowledged |
| requestStatusAssigned |
| requestStatusInProgress |
| requestStatusCompleted |
| requestStatusClosed |
| Planner Input | |
| Background services | remarks |

Status Page

Requires

filepath.php

spaceFormStatus.php

powerFormStatus.php

ssuFormStatus.php

fdfFormStatus.php

cableTrayFormStatus.php

generalFormStatus.php

changeToCompleted.php

changeToInstallationInProgress.php

mmrFormStatus.php

accesssFormStatus.php

In the *Status* page, the Requestor is able to input their *requestId* into the field to retrieve the relevant information.

If the *requestId* is found, all the relevant information will be displayed for the Requestor.

The status page is being divided into 3 main sections.

The left panel shows the general information of the request itself.

The right pane shows the Service Level Agreement (SLA) timeline component and actions component. In the SLA component. The timeline component shows the respective states with the timing once the state has been activated. There is also a countdown timer at the situated at the footer of the component itself that shows the number of days left to change to a different state.

The actions component shows state specific messages upon a change in state. For states like *Assigned* and *Installation in Progress*, the Requestor would be required to complete a set of tasks firsts before the change of state is permitted. For the other states, the Planner would be the one modifying that is subjected to approval and satisfaction of verification.

The metrics component shows the key information for the request itself.

**changeToInstallationInProgress.php**

This file allows for the change in state from Installation in Progress to Completed.

The Requestor needs to make some changes in the FNT Command app before clicking on the button that triggers the change in state

**changeToCompleted.php**

This file allows for the change in state from Installation in Progress to Completed.

The Requestor needs to upload a few images for verification purposes before clicking on the button that triggers the change in state

Guide Page

Requires

guide.php

In the *Guide* page, the Requestor is able to find a list of Frequently Asked Questions (FAQ) as well as some other guidelines that the Requestor should be made aware of when using the app itself.