

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

Specifications

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	

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 Planner

The MPP Planner app is being divided into 8 pages — All Request, Space, Power, SSU, FDF, Cable Tray, General.

Authentication Page

Requires
connection.php
admin.php
auth.php
terminate.php
authVerification.php

Before the Planner can modify or view any of the requests, he/she would need to login first.

The login form is inside `auth.php`. With the fields entered, it will be validation against the database using `authVerification.php` to check if the user is authorised for access. If authorised, the `admin.php` will be shown to the Planner. If an unauthorised user tries to access any page, the user will be shown an error page.

The Planner can end the session by logging out which then triggers `terminate.php` to end the session

All Request Page

Requires
filepath.php
admin.php

In the All Request page, you will find 6 different request type cards available for the Planner to choose from.

When the Planner clicks on the card itself, he/she will be directed to the selected request type.

Space, Power, SSU, FDF, Cable Tray, General Page

Requires
filepath2.php
updateInfo.php
updateInfoPower.php
updateInfoSsu.php
updateInfoFdf.php
updateInfoCableTray.php
updateInfoGeneral.php
updateInfoMmr.php
updateInfoAccess.php

In this page, you will see a dashboard positioned at the top of the page itself and all space requests being shown in a table below the dashboard.

The list of space requests is being sorted in descending order using the *request/d* in the table. Under the Action column, there are 3 types of actions that the Planner can use.

When the update button is clicked, the Planner will be directed to a page whereby the fields of that specific row can be edited.

When the view button is clicked, the Planner will be directed to a page whereby all the information of that specific row can be viewed at a glance.

When the decline button is clicked, the Planner will be directed to a page where the Planner needs to input a reason for declining the specific row

Update Space, Power, SSU, FDF, Cable Tray, General Page

Requires

connection.php
spaceRequests.php
powerRequests.php
ssuRequests.php
fdfRequests.php
cableTrayRequests.php
generalRequests.php
mmrRequests.php
accessRequests.php

In this page, you see some of fields that the Planner can update to provide more information for the Requestor to follow up. In each of the files, they contain different types of parameters that allows the Planner to make changes respectively.

View Space, Power, SSU, FDF, Cable Tray, General Page

Requires

connection.php
viewInfo.php
viewInfoPower.php
viewInfoSsu.php
viewInfoFdf.php
viewInfoCableTray.php
viewInfoGeneral.php
viewInfoMmr.php
viewInfoAccess.php

In this page, you will see an overview of the request itself with all the details furnished.

Delete Space, Power, SSU, FDF, Cable Tray, General Page

Requires

connection.php
deleteInfo.php
deleteInfoPower.php
deleteInfoSsu.php
deleteInfoFdf.php
deleteInfoCableTray.php
deleteInfoGeneral.php
deleteInfoMmr.php
deleteInfoAccess.php

In this page, the Requestor can decline the request with an appropriate reason.