Hotel Management System for Delonix Regia

Nur Sabrina Bte Osman (1703087E)
Gerald Soh WeiJie (1703393G)
Su Swe Zin Nyan (1702107B)

Background

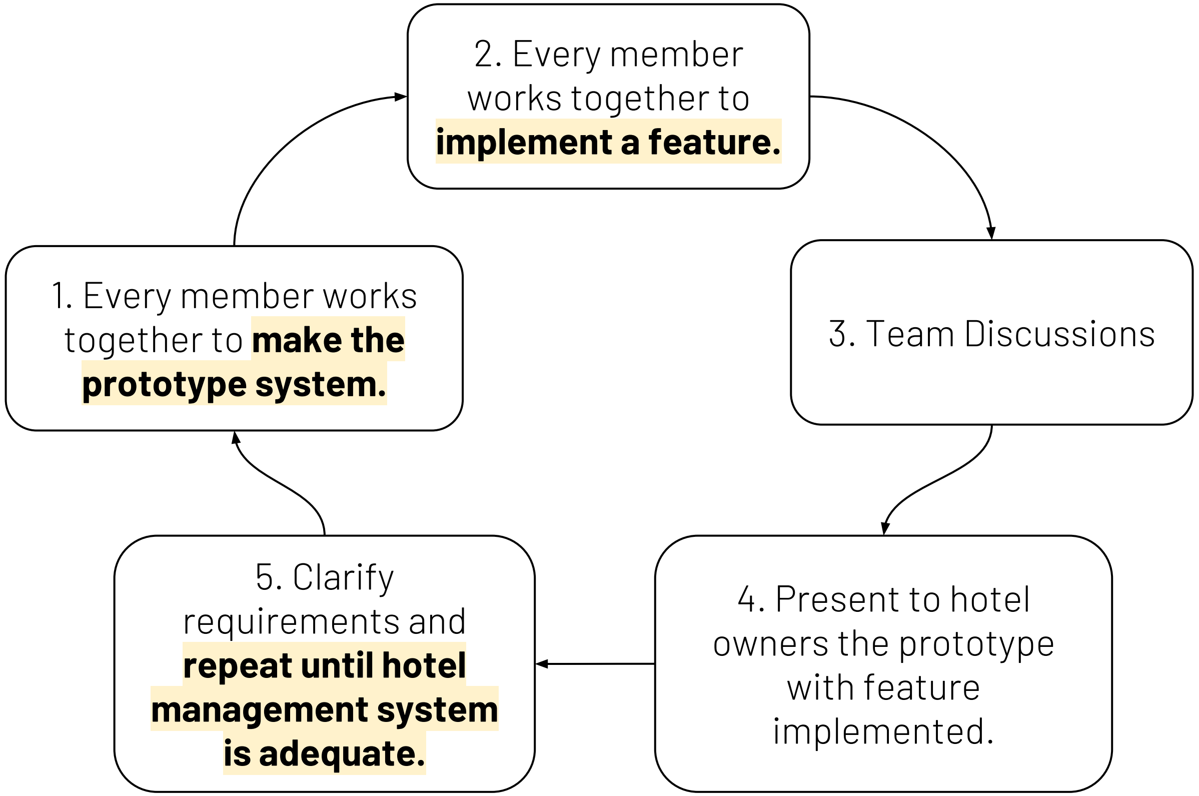
About

Location

Problem

Content

Approach & Methodology


1. Every member works together to **make the prototype system.**
2. Every member works together to **implement a feature.**
3. Team Discussions
4. Present to hotel owners the prototype with feature implemented.
5. Clarify requirements and **repeat until hotel management system is adequate.**

Evolutionary Prototype Model

Modules to be Developed

1. Front-desk
2. Housekeeping
3. Transport
4. Review

Time taken to complete each phase

Planning & Design	+ 21 days (15/10/18 - 13/11/18)
Development of Application	+ 58 days (13/11/18 - 1/2/19)
Total project time	= 80 days

Constraints

- At least 2 hrs for discussion
- Little room for delays or changes
- Only 17 weeks to complete project

- Lack of suitable venues for meetings
- Hotel is a far distance from office

- School schedules may interfere with internship projects
- Members may have commitments outside of Patheon Systems

Resources & Budget

Software Requirement Specifications

Changes From Phase 1

- Review Module removed
- Report Module to be developed

1. FRONT DESK

System Functions: Front-Desk

1. Creating reservation for the guest
2. Checking in a guest
3. Checking out a guest

2. TRANSPORT

By Nur Sabrina Bte Osman

System Functions: Transport

1. Create a Transport Booking
2. Update Transport Booking
3. View All Transport Bookings

3. HOUSEKEEPING

By Su Swe Zin Nyan

System Functions: Housekeeping

1. Update room cleanliness
2. Update facilities status
3. To update which staff is appointed to which room
4. Notify staff of room that requires attention

4. REPORT

By Gerald Soh WeiJie

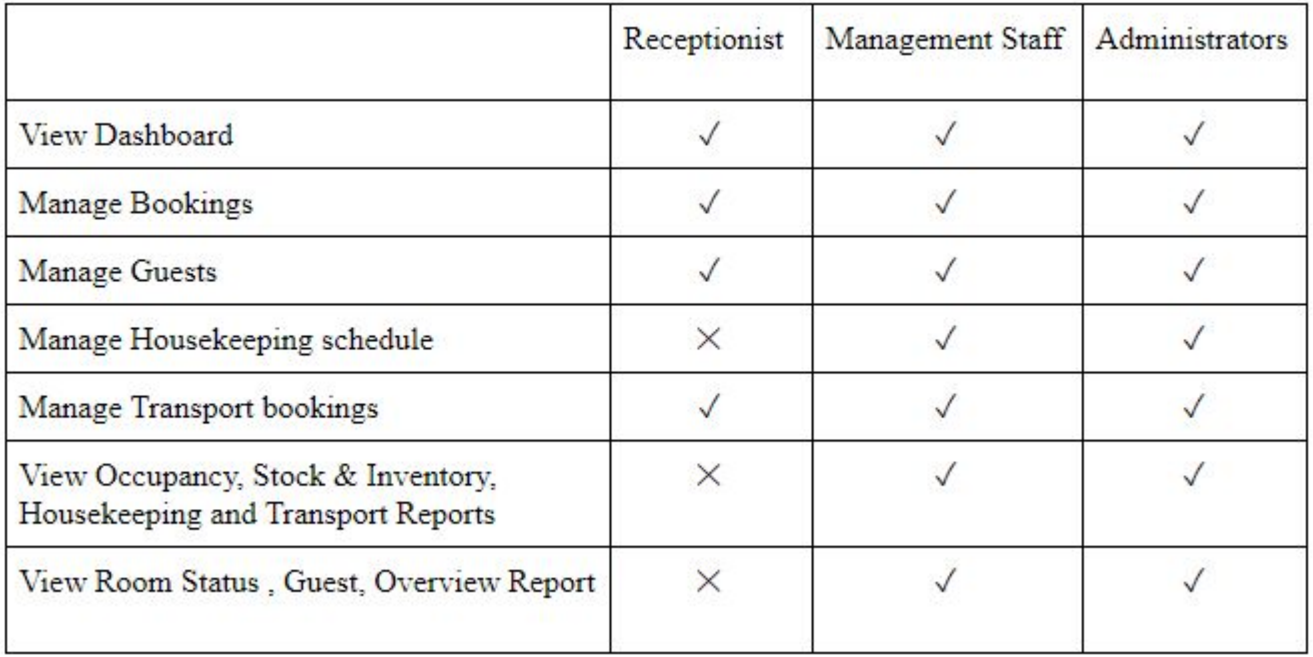
System Functions: Reports

1. Generate Room Status Report
2. Generate Guest Report
3. Generate Overview Report
4. Generate Room Occupancy Report
5. Generate Housekeeping Report
6. Generate Stock Report
7. Generate Transport Report

User Characteristics

1. Receptionist
2. Housekeepers
3. Management Staff
4. System Administrator

Authorization



	Receptionist	Management Staff	Administrators
View Dashboard	✓	✓	✓
Manage Bookings	✓	✓	✓
Manage Guests	✓	✓	✓
Manage Housekeeping schedule	✗	✓	✓
Manage Transport bookings	✓	✓	✓
View Occupancy, Stock & Inventory, Housekeeping and Transport Reports	✗	✓	✓
View Room Status , Guest, Overview Report	✗	✓	✓

Reports

Software Requirement Specifications

- Functional Requirements
- Data Requirements

Software Design Specifications

- Architecture Design
- UI Design
- Program Design

Revised Constraints

Operating System
The users should be able to use the hotel system through operating systems from Windows 7 and newer.

Due Date
Project is to be due by 10 February 2019.

Budget
Budget for project is set at $70,000

Data Expiry
Data held by the Hotel will expire in 5 years.

Software Test Specifications

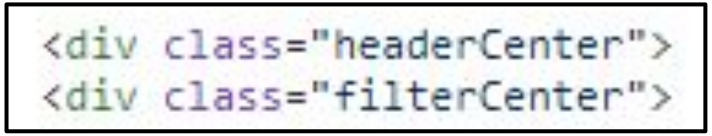
Changes From Phase 2

- Developer responsible for Front Desk Module resigned
- Test specifications on Front desk module could not be created
- Facilities status is cancelled out because Mr. Wang says it can be used as duty type.

Module Development

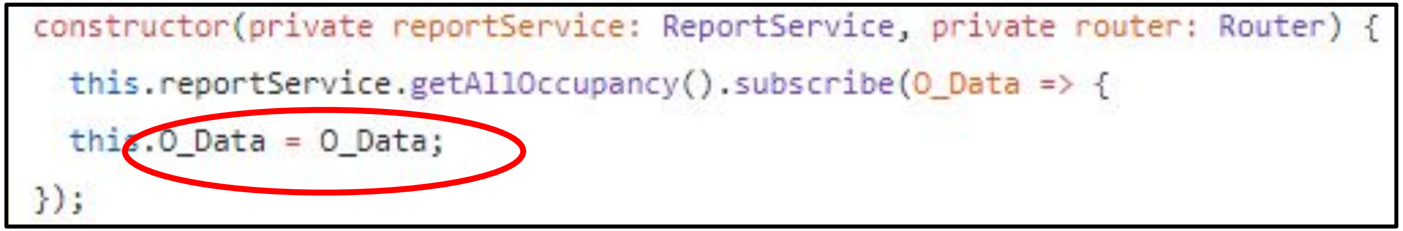
Report Module

- Naming convention: camelCase



```
<div class="headerCenter">
<div class="filterCenter">
```

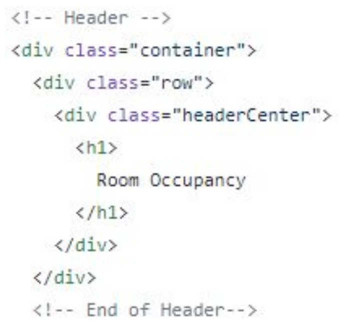
- Code Layout: End-of-Line Style



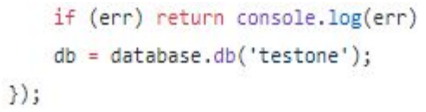
```
constructor(private reportService: ReportService, private router: Router) {
  this.reportService.getAllOccupancy().subscribe(O_Data => {
  this.O_Data = O_Data;
});
```

Report Module

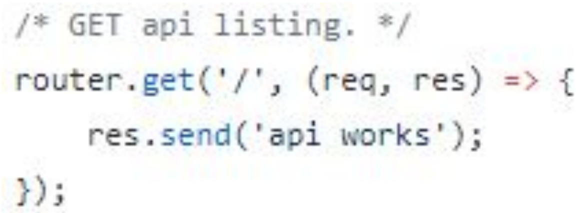
- Commenting



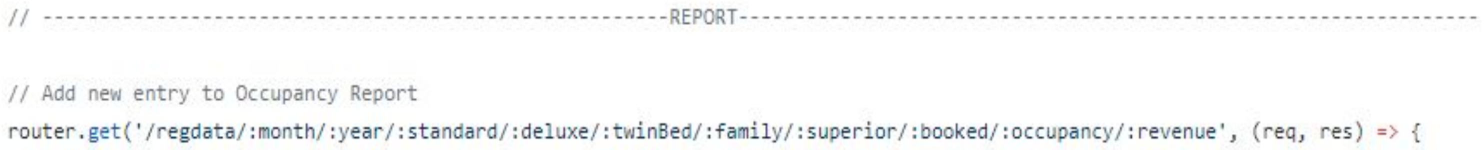
```
<!-- Header -->
<div class="container">
  <div class="row">
    <div class="headerCenter">
      <h1>
        Room Occupancy
      </h1>
    </div>
  </div>
  <!-- End of Header-->
```



```
    if (err) return console.log(err)
    db = database.db('testone');
});
```



```
/* GET api listing. */
router.get('/', (req, res) => {
    res.send('api works');
});
```



```
// ------------------------------------------------------------REPORT---------------------------------------------------------------

// Add new entry to Occupancy Report
router.get('/regdata/:month/:year/:standard/:deluxe/:twinBed/:family/:superior/:booked/:occupancy/:revenue', (req, res) => {
```

Transport Module

Good Programming Style

- Comments
- Code Layout
- Naming Conventions

Transport Module

Comments

- Inline & Block Comments
- Uses for Comments
  - Description for Codes
  - Debugging

```
 // get all transport bookings
 getAllTransport() {
   return
this.http.get<any[]>('./api/transport');
 }
```

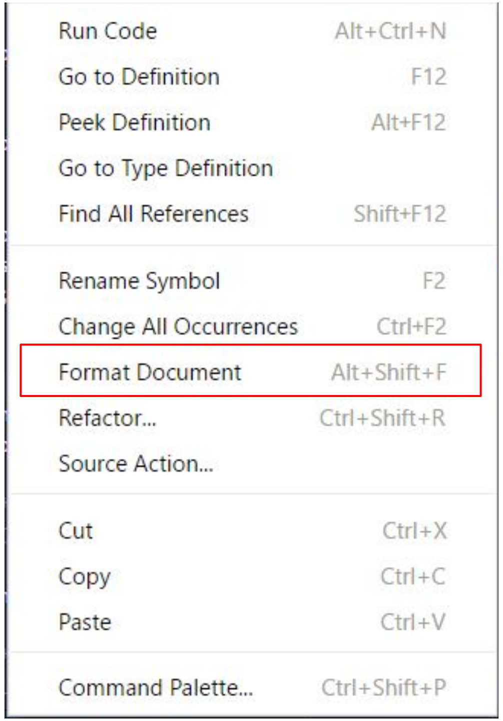
```
/* Get port from environment and store in Express. */
const port = process.env.PORT || '3000';
app.set('port', port);
```

Transport Module

Code Layout

- Braces Placement
  - End-of-Line
- Format Document Tool

```
 ngOnInit() {

this.transportService.getAllTransport()
.subscribe(transport => {
     this.transport = transport;
   });
 }
```



Transport Module

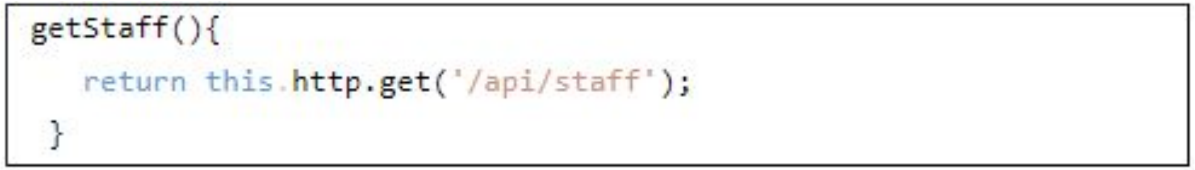
Name Conventions

- Lower Camel Case

```
// get all transport bookings
getAllTransport() {
  return this.http.get<any[]>('./api/transport');
}
```

Housekeeping Module

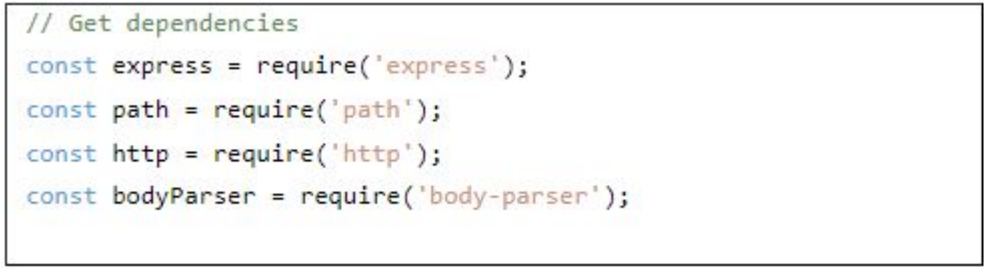
- **lowerCamelCase**
- insertrequest
- insertRequest
- Understand codes faster and easier
- Lesser/no mistakes made
- Lesser/no errors made
- Simple and descriptive



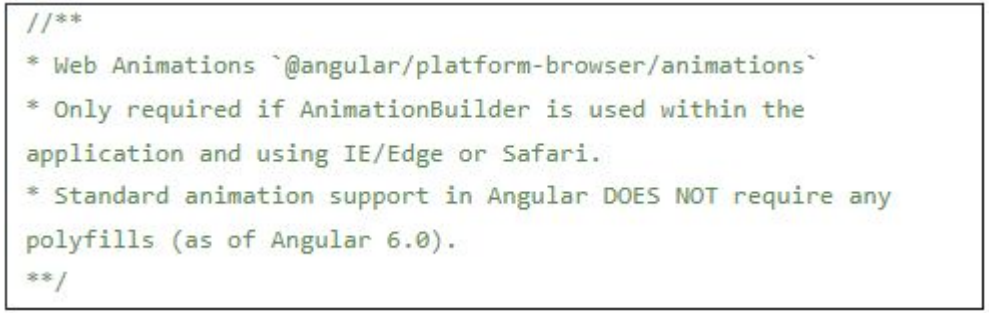
```
getStaff(){
   return this.http.get('/api/staff');
 }
```

Housekeeping Module

- Comments
- Aid javascript
- Explain what is going on
- Help programmer remember
- Remove bits of code from execution
- Single -line comments
- " // "
- Multi - line comments
- " */ "



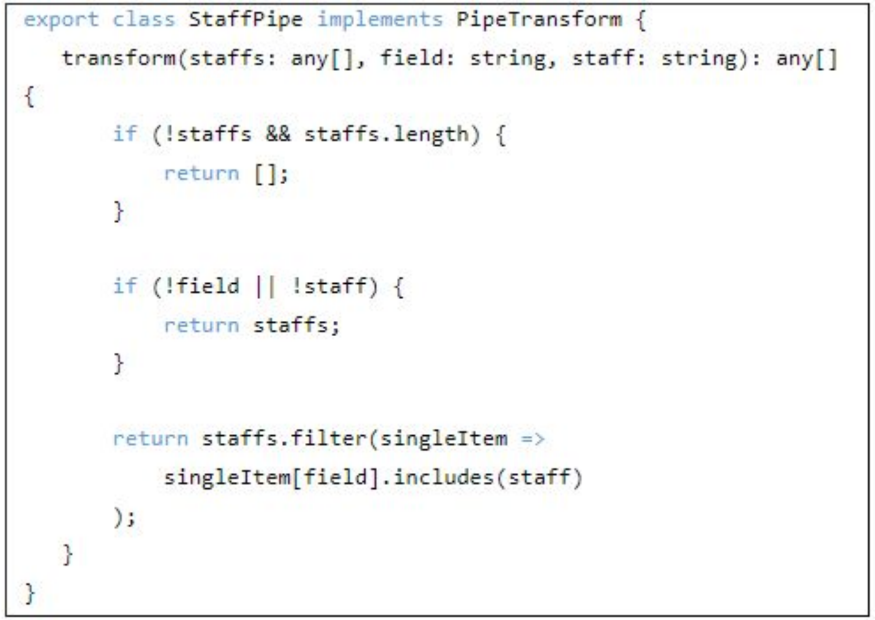
```
// Get dependencies
const express = require('express');
const path = require('path');
const http = require('http');
const bodyParser = require('body-parser');
```



```
//**
* Web Animations `@angular/platform-browser/animations`
* Only required if AnimationBuilder is used within the
application and using IE/Edge or Safari.
* Standard animation support in Angular DOES NOT require any
polyfills (as of Angular 6.0).
**/
```

Housekeeping Module

- Formatting
  - Read code easier
  - Elimates “diff noise”
  - More comfortable pairing
  - Sharing code bases
  - Standardize codes
  - Make it easier to read code with comfort
  - Lesser mistakes made

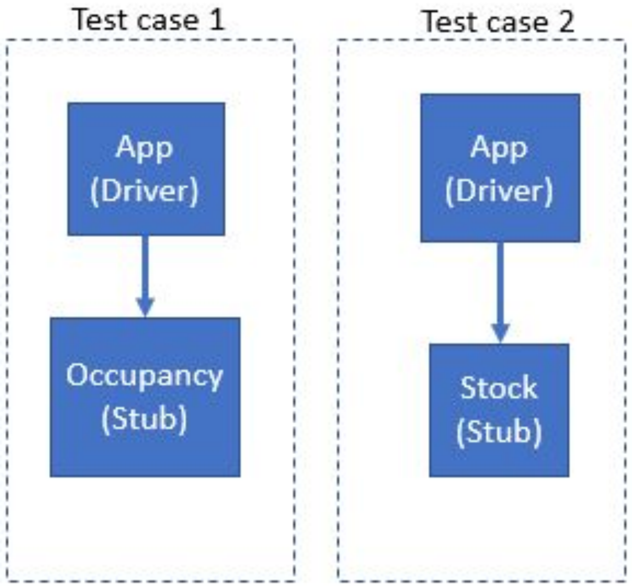


```
export class StaffPipe implements PipeTransform {
   transform(staffs: any[], field: string, staff: string): any[]
{
        if (!staffs && staffs.length) {
            return [];
        }

        if (!field || !staff) {
            return staffs;
        }

        return staffs.filter(singleItem =>
            singleItem[field].includes(staff)
        );
    }
}
```

Unit Testing

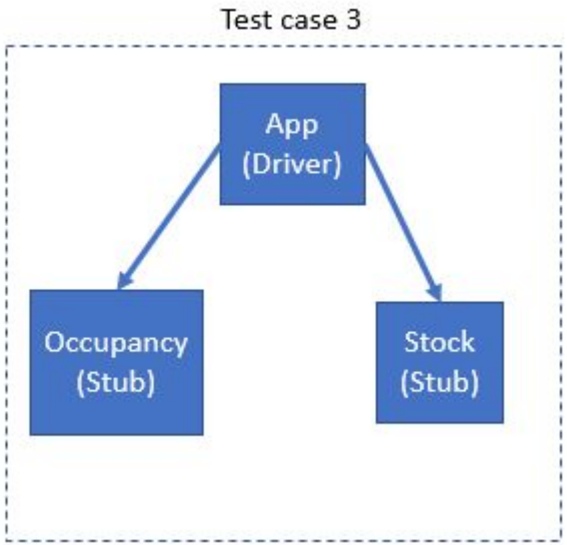
Report Module

- Unit testing



Report Module

- Integration testing



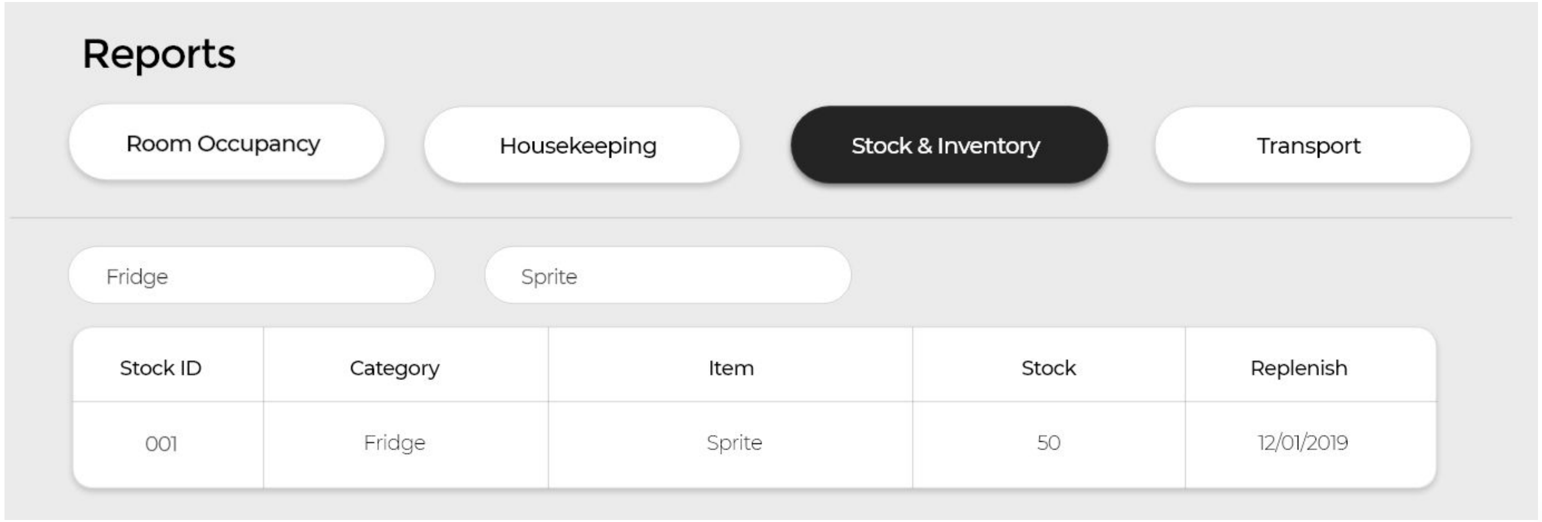
Report Module

- Normal, Abnormal and Illegal Data Types
- Input Field: Filter

Normal Data	Illegal and Abnormal Data
Reports will be shown	Reports will not be shown

Report Module

- Normal Data Type

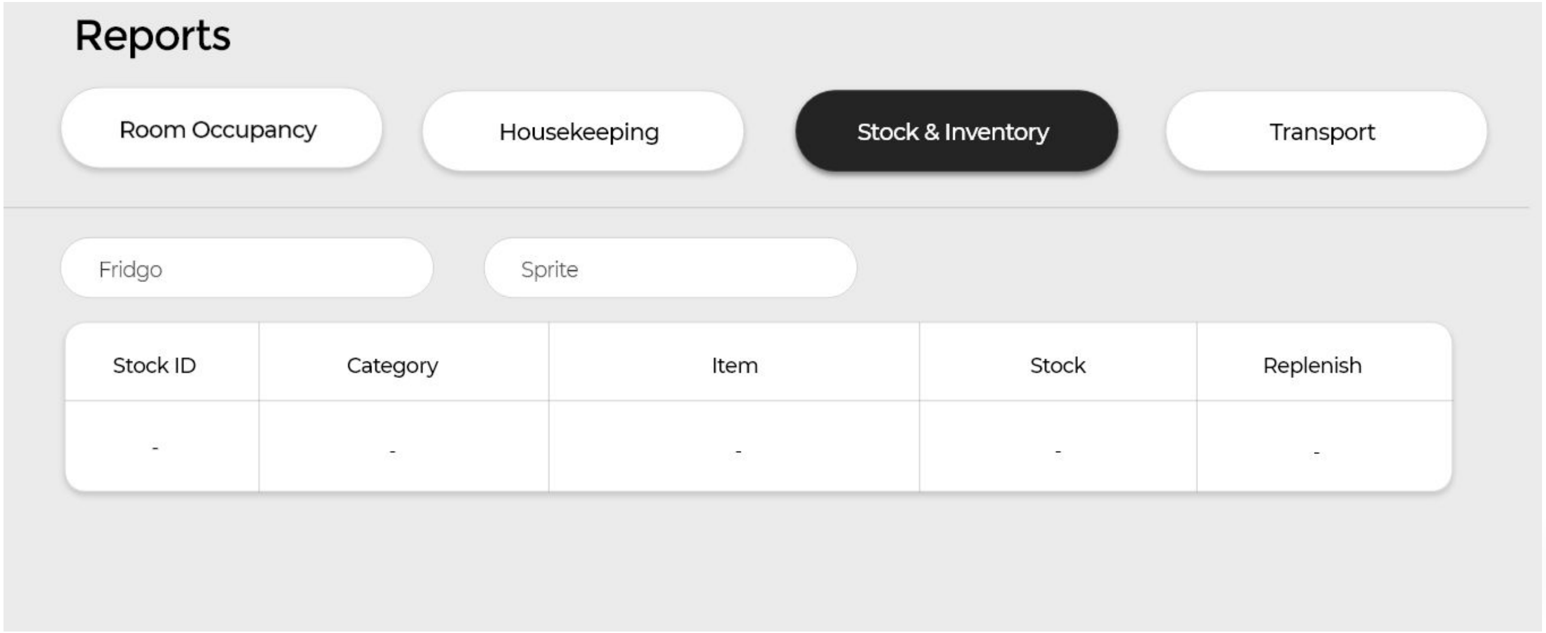


Reports

Stock ID	Category	Item	Stock	Replenish
001	Fridge	Sprite	50	12/01/2019

Report Module

- Illegal & Abnormal Data Type



Reports

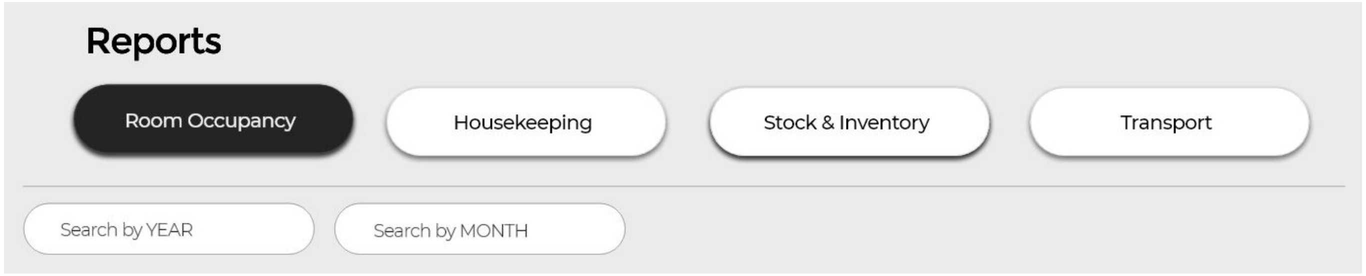
Fridgo

Sprite

Stock ID	Category	Item	Stock	Replenish
-	-	-	-	-

Report Module

- Equivalence Partitioning



	Valid	
	2016 | 2017 | 2018 | 2019	

Report Module

- Equivalence Partitioning

Invalid	Valid	Invalid
~2015	2016 | 2017 | 2018 | 2019	2020~

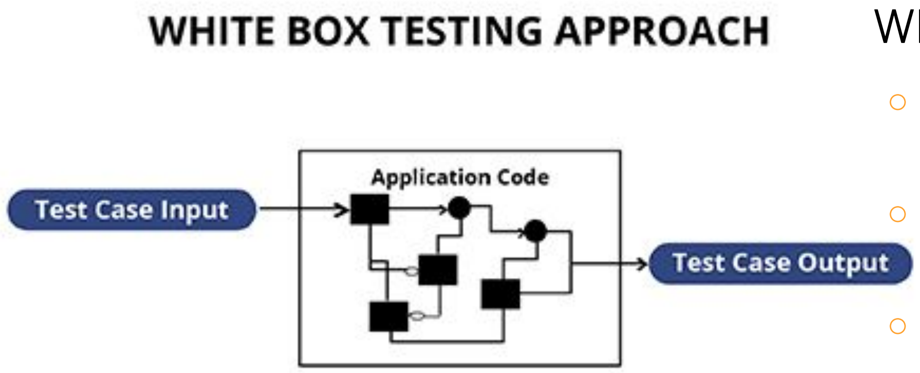
Invalid	Valid	Invalid
~2015	2016 | 2017 | 2018 | 2019	2020~
First partition	Second partition	Third partition

Report Module

- Equivalence Partitioning and Boundary value analysis

Invalid	Valid	Invalid
2015	2016 2019	2020
First partition	Second partition	Third partition

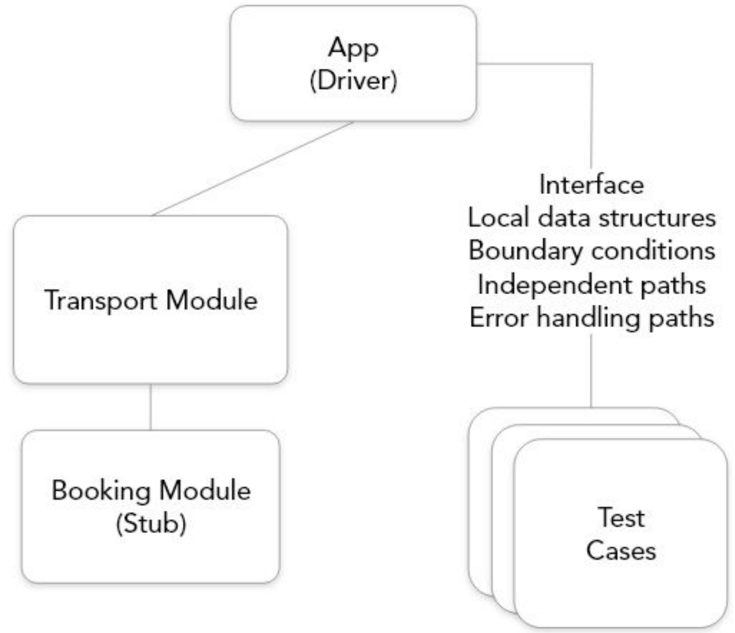
Transport Module



White-Box Testing

- Used in Unit, Integration & System Testing
- Can be commenced at early stages of development
- Tested by Developer
- To test internal structure of system

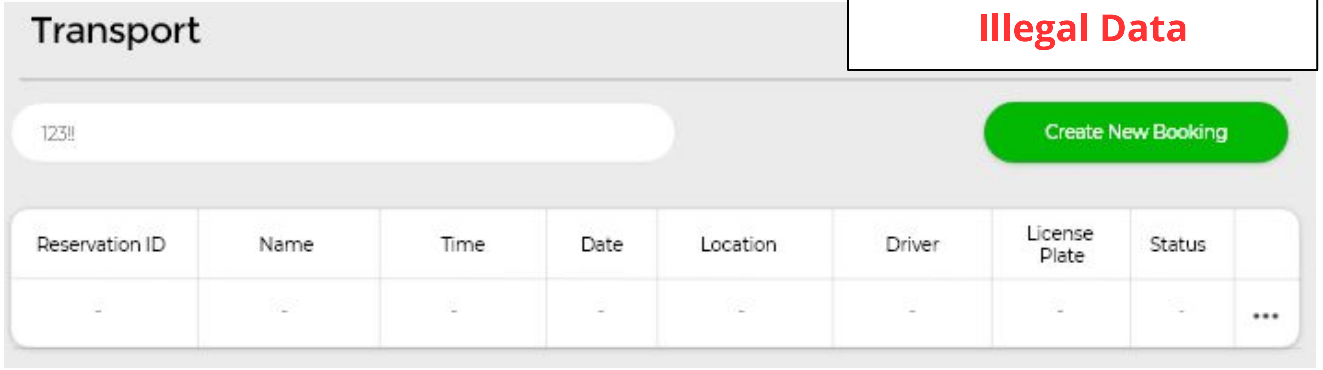
Transport Module



Stubs & Drivers

- Tests to be done on incomplete or unavailable modules
- Temporary replacements to undeveloped modules

Transport Module



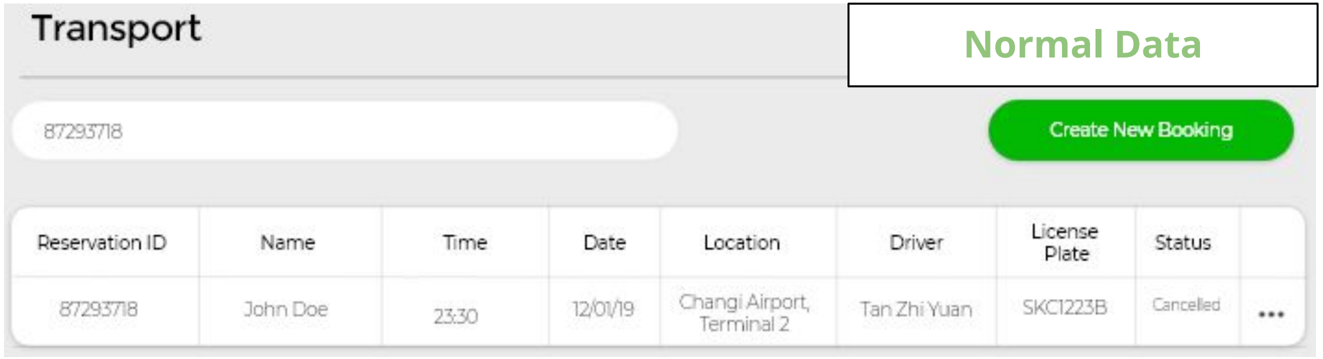
Illegal Data

Transport

123!!

Create New Booking

Reservation ID	Name	Time	Date	Location	Driver	License Plate	Status	
-	-	-	-	-	-	-	-	•••



Normal Data

Transport

87293718

Create New Booking

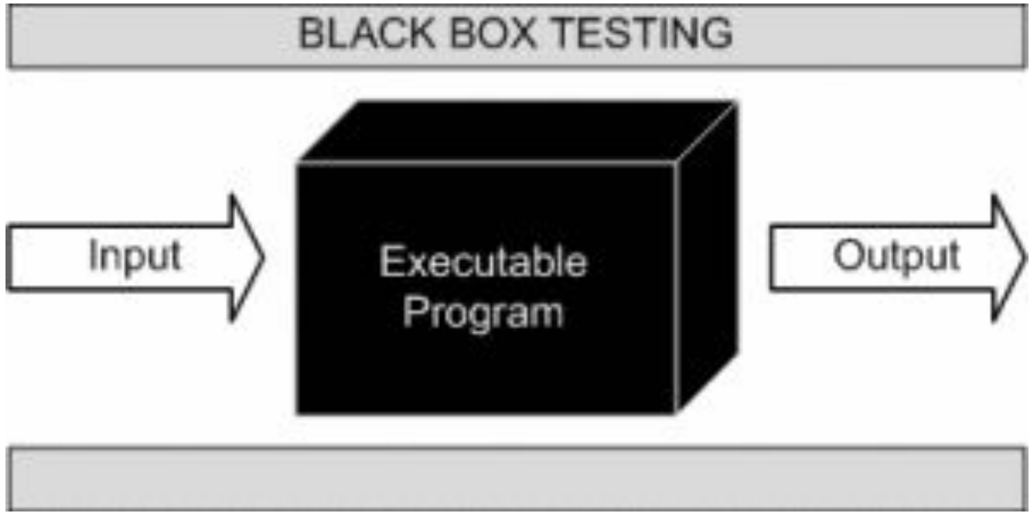
Reservation ID	Name	Time	Date	Location	Driver	License Plate	Status	
87293718	John Doe	23:30	12/01/19	Changi Airport, Terminal 2	Tan Zhi Yuan	SKC1223B	Cancelled	•••

Housekeeping Module

- Unit Testing
- First level of testing
- Ensures individual components are functional and work like designed to
- Makes debugging easier
- Unit Testing Method is performed by using White-Box Testing method
- Three unit testing task
    - Unit Test Plan
    - Unit Test Cases
    - Unit Test
- Tests are periodically ran
- The sooner the problems are shown, the better it is

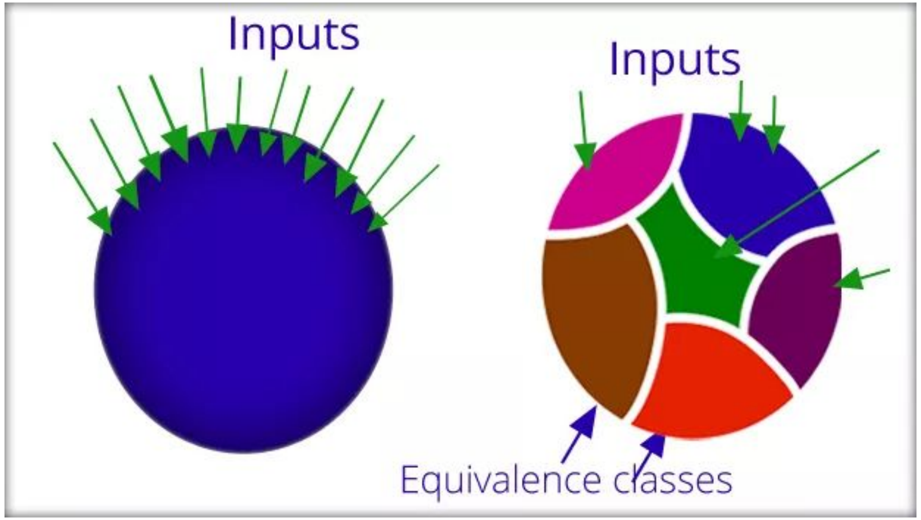
Housekeeping Module

- Black-Box (Functionally) Testing



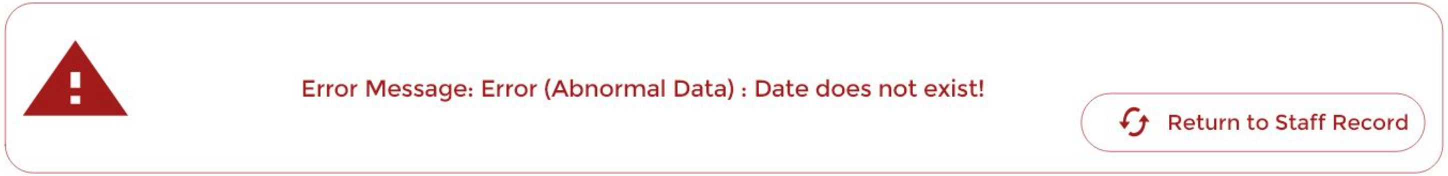
Housekeeping Module

- Attempts to find errors such as incorrect or missing functions
- Interface errors
- Errors in data structures
- Behaviour or performances errors
- **Equivalence Partitioning**

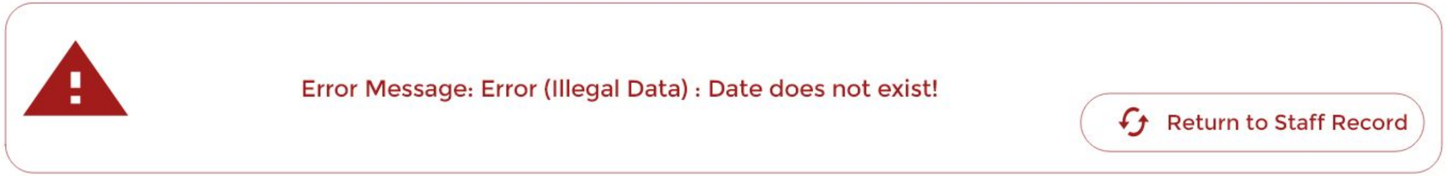


Housekeeping Module

Abnormal Data	Illegal Data
Cannot be of a date that does not exist	Date of birth cannot be a future date
2nd January 2019	34th January or 30th February

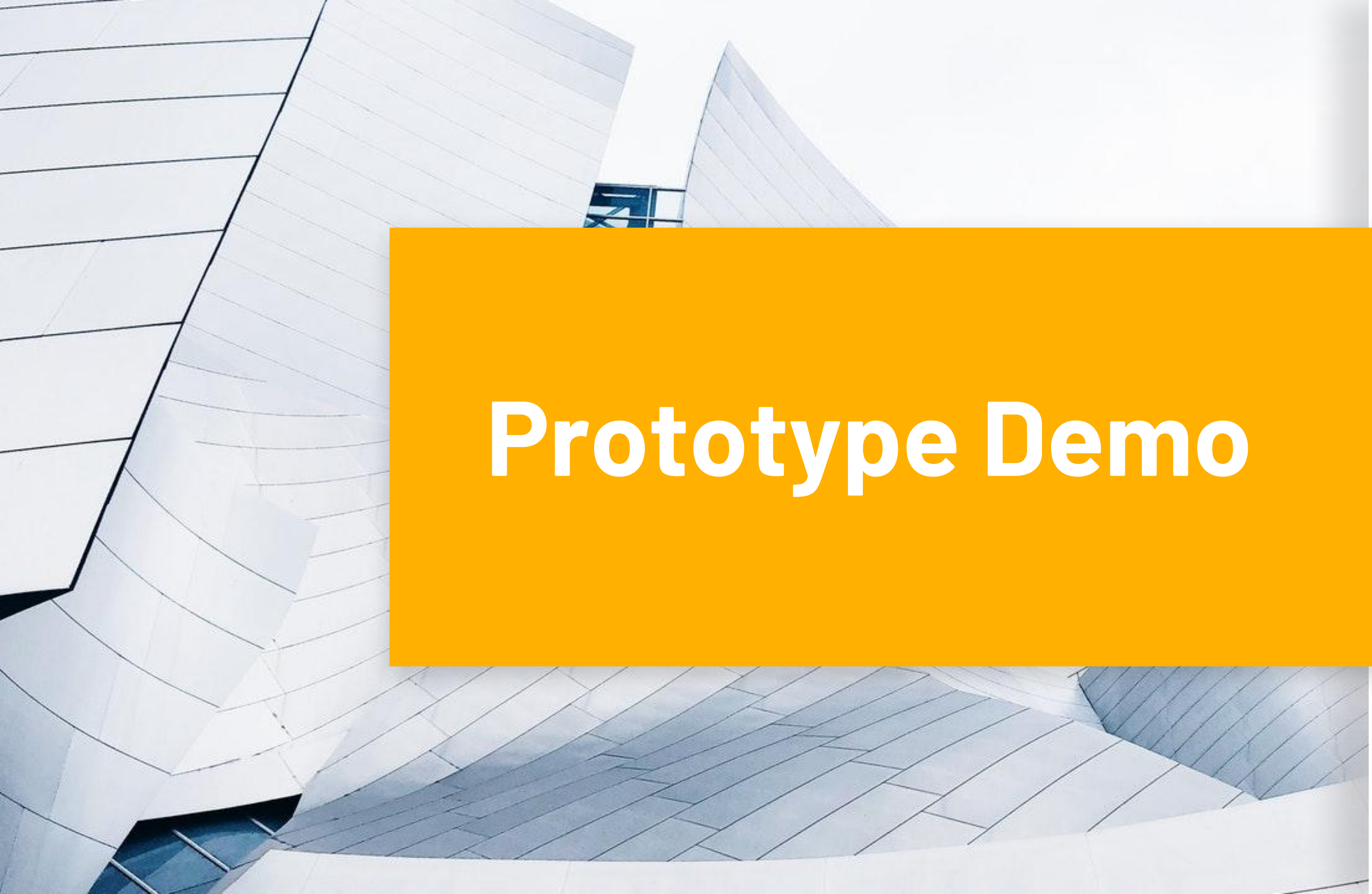


Error Message: Error (Abnormal Data) : Date does not exist!



Error Message: Error (Illegal Data) : Date does not exist!

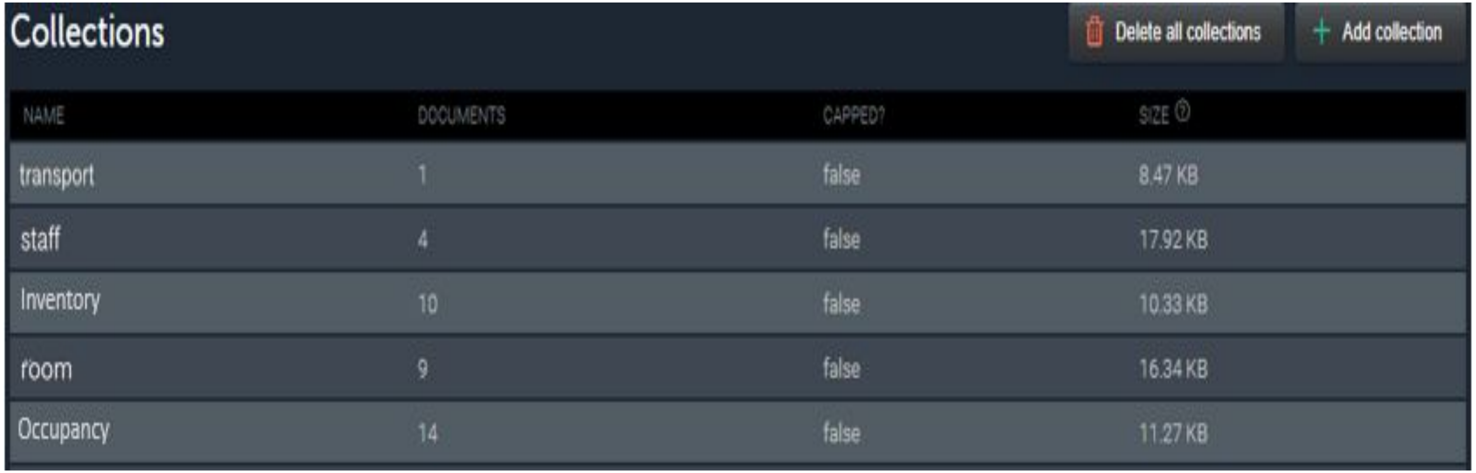
System Integration



Prototype Demo

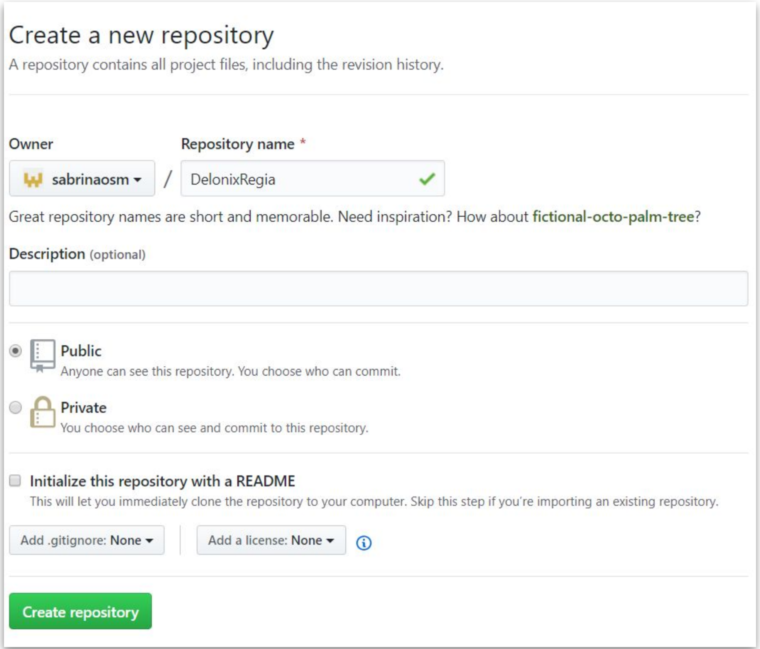
Thank you!

Database



NAME	DOCUMENTS	CAPPED?	SIZE
transport	1	false	8.47 KB
staff	4	false	17.92 KB
Inventory	10	false	10.33 KB
room	9	false	16.34 KB
Occupancy	14	false	11.27 KB

Integration with GitHub



Create a new repository

A repository contains all project files, including the revision history.

Owner: sabrinaosm / Repository name *: DelonixRegia

Great repository names are short and memorable. Need inspiration? How about fictional-octo-palm-tree?

Description (optional)

(●) Public
Anyone can see this repository. You choose who can commit.

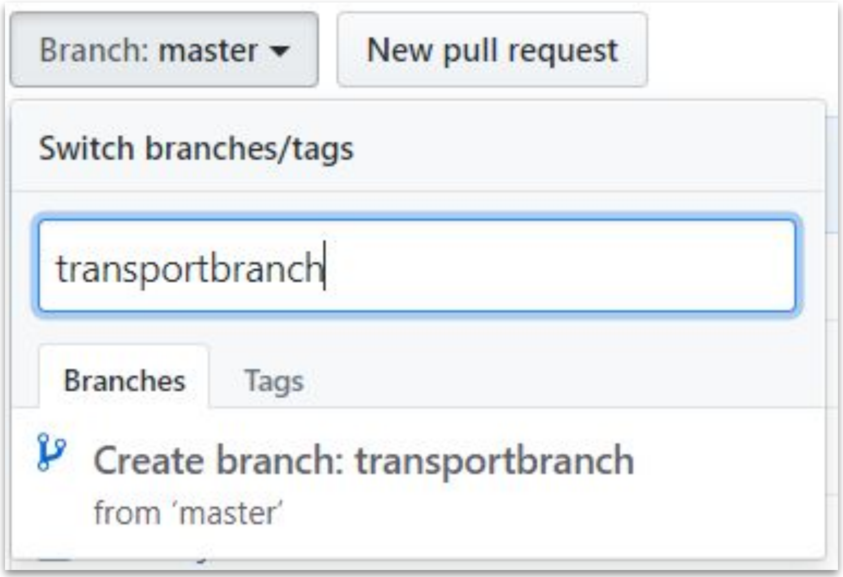
( ) Private
You choose who can see and commit to this repository.

[ ] Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

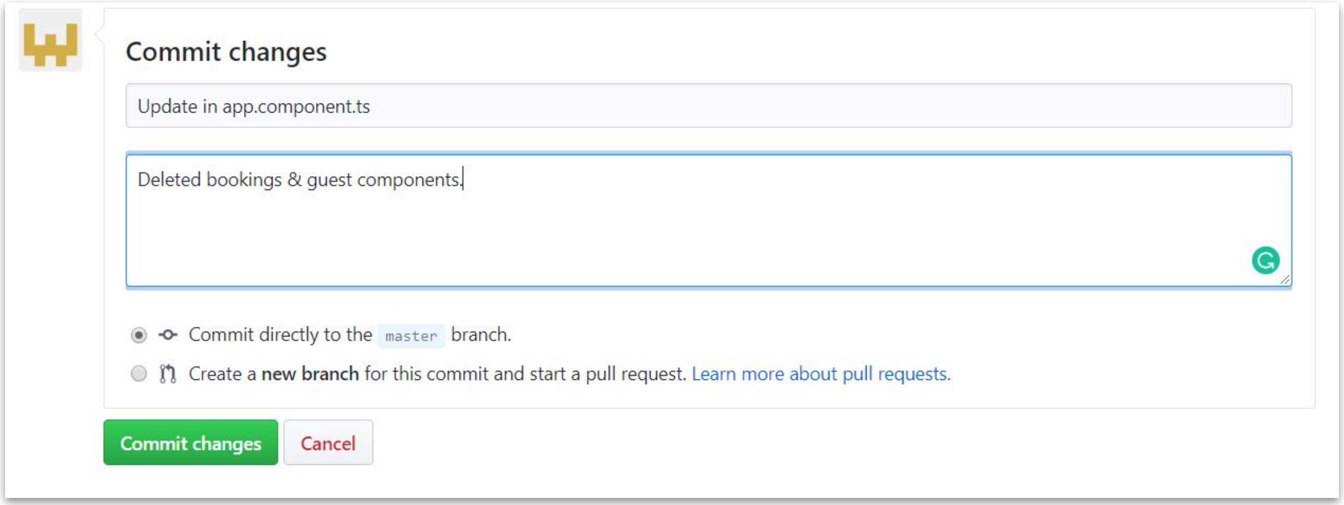
1) Creating a Repository in GitHub

Integration with GitHub



2) Creating a branch for updates & experiments

Integration with GitHub


Commit changes

Update in app.component.ts

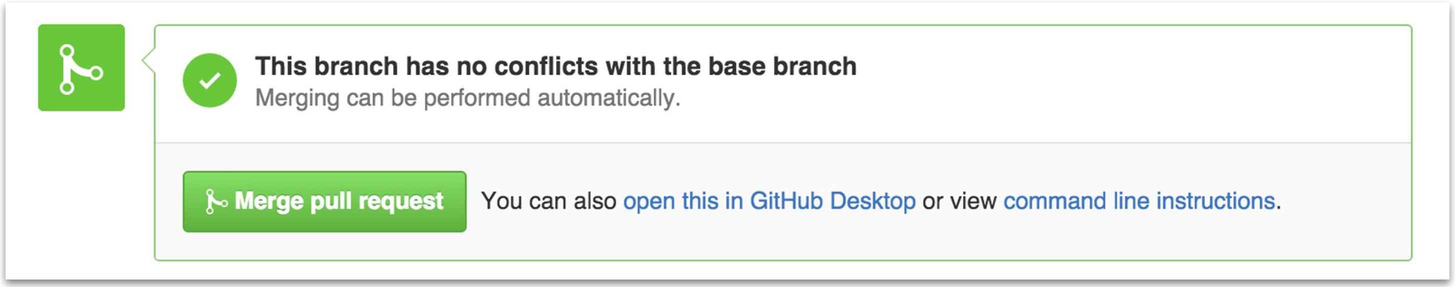
Deleted bookings & guest components.

[✓ Commit directly to the `master` branch.]

Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

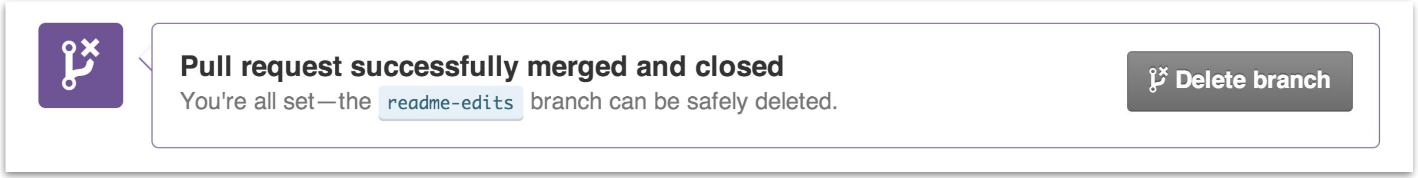
3) Make changes to a file & push them to GitHub as commits

Integration with GitHub



This branch has no conflicts with the base branch
Merging can be performed automatically.

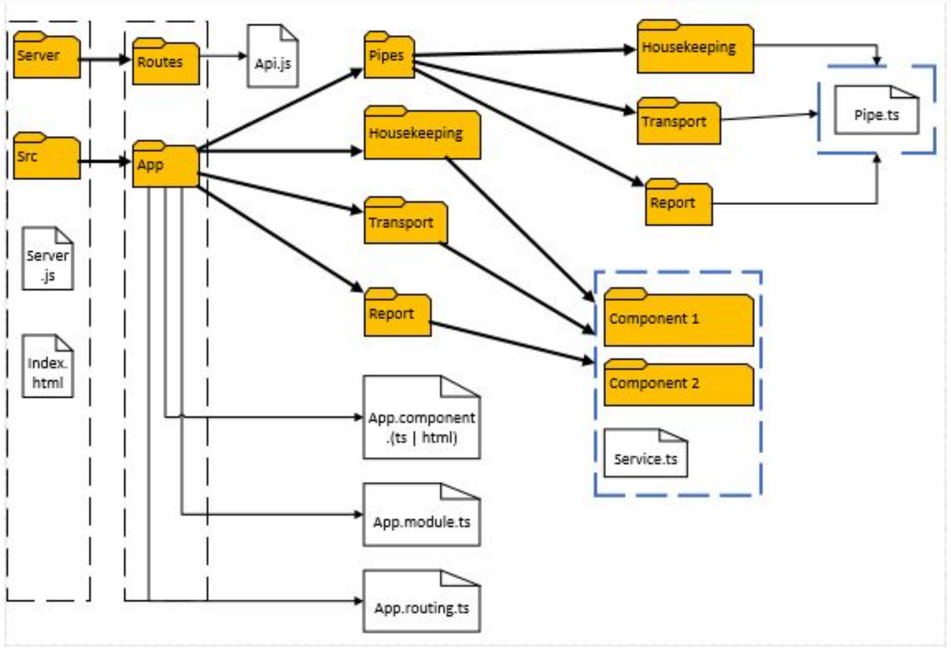
You can also open this in GitHub Desktop or view command line instructions.



Pull request successfully merged and closed
You're all set—the `readme-edits` branch can be safely deleted.

4) Open & merge request

Integration with GitHub



GitHub File Directory