សកលវិទ្យាល័យ បៀលប្រាយ

**BUILD BRIGHT UNIVERSITY**

ប្រធានបទស្តីអំពីៈ

ប្រព័ន្ធគ្រប់គ្រងការលក់ប្រព័ន្ធ

គ្រប់​គ្រង​​លក្ខណៈ Web-based Application និងគេហទំព័រ

របស់​​ក្រុម​​ហ៊ុន ​Rabbit IT Solution

រៀបរៀងដោយៈ

១. និស្សិត​ ទៀប ពិសិដ្ឋ

២. និស្សិត ហេង យូស៊ួរ

៣. និស្សិត មឿង សេរីរតនៈ

៤. និស្សិត ប៉ុន ភក្តី

៥. និស្សិត ឡោ រ៉ាយុត

៦. និស្សិត ភី ផាន់នី

ដឹកនាំដោយៈ សាស្ត្រាចារ្យ ឡោ សូត្រ

សារណាបញ្ចប់ថ្នាក់បរិញ្ញាប័ត្រ

ផ្នែក វិទ្យាសាស្ត្រ និងបច្ចេកវិទ្យា របស់និស្សិតជំនាន់ទី ៨

ឯកទេស បច្ចេកវិទ្យាព័ត៌មាន និងបណ្តាញកុំព្យូទ័រ

ឆ្នាំសិក្សា ២០១២ - ២០១៦

សកលវិទ្យាល័យ បៀលប្រាយ

**BUILD BRIGHT UNIVERSITY**

ប្រធានបទស្តីអំពីៈ

ប្រព័ន្ធគ្រប់គ្រងការលក់ប្រព័ន្ធ

គ្រប់​គ្រង​​លក្ខណៈ Web-based Application និងគេហទំព័រ

របស់​​ក្រុម​​ហ៊ុន ​Rabbit IT Solution

រៀបរៀងដោយៈ

១. និស្សិត​ ទៀប ពិសិដ្ឋ

២. និស្សិត ហេង យូស៊ួរ

៣. និស្សិត មឿង សេរីរតនៈ

៤. និស្សិត ប៉ុន ភក្តី

៥. និស្សិត ឡោ រ៉ាយុត

៦. និស្សិត ភី ផាន់នី

សាស្ត្រាចារ្យដឹកនាំ់ៈ ឡោ សូត្រ

សារណាបញ្ចប់ថ្នាក់បរិញ្ញាប័ត្រ

ផ្នែក វិទ្យាសាស្ត្រ និងបច្ចេកវិទ្យា របស់និស្សិតជំនាន់ទី ៨

ឯកទេស បច្ចេកវិទ្យាព័ត៌មាន និងបណ្តាញកុំព្យូទ័រ

ឆ្នាំសិក្សា ២០១២ - ២០១៦

សមាសភាពគណៈកម្មការវាយតម្លៃការពារសារណា

១. .....................................................ហត្ថលេខា..............................ឋានៈប្រធាន

២. .....................................................ហត្ថលេខា..............................ឋានៈសមាជិក

៣. .....................................................ហត្ថលេខា..............................ឋានៈសមាជិក

ព្រឹទ្ធបុរស

សាកលវិទ្យាធិការ

សម្គាល់ៈ

* មហាវិទ្យាល័យ.......................................................................................ពុំមានយោបល់លើមតិដែលមានក្នុងសារណានេះទេ។
* បណ្តាមតិទាំងឡាយក្នុងសារណានេះចាត់ទុកជាមតិផ្ទាល់ខ្លួនរបស់អ្នកនិពន្ធ

# សេចក្តីអំណះអំណាង

ក្រុមរបស់យើងខ្ញុំសុំអះអាង និង​ធានាថាការសិក្សាស្រាវជ្រាវលើកិច្ចការរបស់យើងនេះខ្ញុំមានភាពដើមទាំងស្រុង។ ហើយកិច្ចការដែលក្រុមយើងខ្ញុំបានស្រាវជ្រាវនេះមិនត្រូវបានប្រគល់​ជូនទៅស្ថាប័នឧត្តមសិក្សា ឬស្ថាប័នស្រាវជ្រាវណាមួយផ្សេងទៀតឡើយដើម្បីទទួលយកសញ្ញាប័ត្រ ឬ​ឯកសារដ៏ទៃទៀត។ ​បើករណីណាដែលសាកលវិទ្យាល័យពិនិត្យឃើញថាករណីខាងលើនេះកើតឡើងមែននោះសារណារបស់យើង និង​​មិនមានលក្ខណៈគ្រប់គ្រាន់ដើម្បីទទួលយកសញ្ញាប័ត្រឡើយ ។

បាត់ដំបង ថ្ងៃទី.......ខែ........ឆ្នាំ២០១.....

ហត្ថលេខា និងឈ្មោះនិសិ្សត

.......................................

ទៀប ពិសិដ្ឋ

.......................................

ហេង យូស៊ួរ

.......................................

មឿង​ សេរីរតនៈ

.......................................

ប៉ុន ភក្តី

.......................................

ឡោ រ៉ាយុត

# អារម្ភកថា

ក្រុមនិស្សិតយើងខ្ញុំជានិស្សិតជំនាន់ទី៨ ឆ្នាំទី៤ នៃសាកលវិទ្យាល័យ បៀលប្រាយ សាខាខេត្តបាត់ដំបង។ ក្រុមនិស្សិតយើងខ្ញុំមានសេចក្តីសោមនស្សរីករាយជាខ្លាំងនៅក្នុងការរៀប​រៀង​ជាសៀវភៅនេះឡើងដែលមានការសហការដោយអស់ពីកម្លាំងកាយចិត្ត និងព្រមទាំងមានការជ្រុំជ្រែងពីសំណាក់​លោកសាស្រ្តាចារ្យ​​ ឡោ សូត្រ ទើបធ្វើឲ្យការដំណើរការនៃការចុះស្រាវជ្រាវនេះកាន់តែមានភាពប្រសើរឡើងជាខ្លាំង។

ដោយយល់ឃើញថាប្រទេសកម្ពុជាមានការរីកចំរើនស្ទើរគ្រប់វិស័យជាពិសេសនោះគឺលើវិស័យព័ត៌មានវិទ្យាដែលជាវិស័យមួយជាជំនួយជួយឲ្យប្រទេសកម្ពុជាយើងមានការរីកចម្រើន ទំនើប និងទាន់សម័យ។ ហេតុដូច្នេះហើយ ទើបក្រុមរបស់យើងខ្ញុំបានរៀបរៀងជាសៀវភៅនេះឡើង ដើម្បីជាគន្លឹះសម្រាប់ធ្វើការដោះស្រាយ និងជាចំណេះដឹងបន្ថែមដល់និស្សិតជំនាន់ក្រោយៗ​ សម្រាប់ធ្វើការសិក្សាស្រាវជ្រាវជាបន្តទៀតបន្ទាប់ពីជំនាន់របស់យើងខ្ញុំ​ ។

ទោះបីជាក្រុមរបស់យើងខ្ញុំ បានខិតខំព្យាយាមប្រិតប្រៀងយ៉ាងណាក៏ដោយក៏យើងខ្ញុំជឿជាក់ថានឹងនូវតែមានចន្លោះខ្វះខាតនៅចំណុចណាមួយជាមិនខាន។ ដូច្នេះ ក្រុមរបស់​យើងខ្ញុំសូមខន្តីអភ័យទោស ទុកជាមុនរាល់នូវចំនុចខ្វះខាតទាំងអស់នោះ ហើយក្រុមរបស់យើងខ្ញុំនឹងរងចាំទទួលនូវរាល់ការរិះគន់ទាំងឡាយណាដើម្បីធ្វើការស្ថាបនារាល់កំហុសជាថ្មី​ម្ដងទៀត ។

ជាចុងបញ្ចប់នេះ ក្រុមរបស់យើងខ្ញុំសូមថ្លែងអំណរគុណដល់ លោកសាស្រ្តាចារ្យ ឡោ សូត្រ ដែលបានផ្តល់នូវចំណេះដឹង និងគំនិតជាច្រើន និងព្រមទាំងជួយបង្ហាញផ្លូវដល់ក្រុមរបស់យើងខ្ញុំសម្រេចបានជោគជ័យនៅថ្ងៃខាងមុខ។ សូមថ្លែងអំណរគុណចំពោះគណៈគ្រប់គ្រងរបស់សាកលវិទ្យាល័យ បៀលប្រាយ​​ ដែលបានជួយជ្រុំជ្រែង និងជួយផ្តល់នូវជំនាញ បទពិសោធន៍ និងការអប់រំល្អៗដល់ក្រុមនិស្សិតយើងទាំងអស់គ្នា និង​សូមថ្លែងអំណរគុណដល់មិត្តអ្នកអាន ព្រមទាំងនិស្សិតជំនាន់ក្រោយទាំងអស់ដែលបានគាំទ្រនូវ​គំនិត​របស់ក្រុមយើងខ្ញុំ ។

បញ្ជីមាតិកា

ចំណងជើង ទំព័រ

[សេចក្តីអំណះអំណាង i](#_Toc451526633)

[អារម្ភកថា ii](#_Toc451526634)

[សេចក្តីថ្លែងអំណរគុណ viii](#_Toc451526635)

[មូលន័យសង្ខេបសារណា ix](#_Toc451526636)

[តារាង x](#_Toc451526637)

[តួលេខ xi](#_Toc451526638)

[រូបភាព xii](#_Toc451526639)

[បញ្ជីពាក្យអក្សរកាត់ xiv](#_Toc451526640)

[I. សេចក្តីផ្តើម 1](#_Toc451526641)

[១.១.​ លំនាំដើមនៃការស្រាវជ្រាវ 1](#_Toc451526642)

[១.២. ចំនោទបញ្ហា 2](#_Toc451526643)

[១.៣. គោលបំណងនៃការស្រាវជ្រាវ 2](#_Toc451526644)

[១.៣.១ គោលបំណងរួម 2](#_Toc451526645)

[១.៣.២ គោលបំណងជាក់លាក់ 2](#_Toc451526646)

[១.៤. ទំហំ​ និងដែនកំណត់នៃការស្រាវជ្រាវ 3](#_Toc451526647)

[១.៤.១ ទំហំនៃការស្រាវជ្រាវ 3](#_Toc451526648)

[១.៤.២​ ដែនកំណត់នៃការស្រាវជ្រាវ 3](#_Toc451526649)

[១.៥. សារៈសំខាន់នៃការស្រាវជ្រាវ 4](#_Toc451526650)

[១.៦. ការពន្យល់ពាក្យគន្លឹះ 4](#_Toc451526651)

[II. រំលឹកទ្រឹស្តី 7](#_Toc451526652)

[២.១. សញ្ញាណទូទៅនៃប្រព័ន្ធព័ត៌មាន 7](#_Toc451526653)

[២.១.១. បច្ចេកវិទ្យាព័ត៌មានជាអ្វី ? 7](#_Toc451526654)

[២.១.២ ប្រព័ន្ធជាអ្វី ? 7](#_Toc451526655)

[២.១.៣ ប្រព័ន្ធព័ត៌មានជាអ្វី ? 7](#_Toc451526656)

[២.១.៤​ មុខងារនៃប្រព័ន្ធព័ត៌មាន 7](#_Toc451526657)

[២.១.៥​ សមាសធាតុនៃប្រព័ន្ធព័ត៌មាន 8](#_Toc451526658)

[២.២. ប្រព័ន្ធគ្រប់គ្រងដេតាបេស (Database Management System) 9](#_Toc451526659)

[២.៣. Data Flow Diagram (DFD) 10](#_Toc451526660)

[២.៣.១. Data Flow Symbols 10](#_Toc451526661)

[២.៣.២. Context Diagram 11](#_Toc451526662)

[២.៣.៣. Diagram 0 12](#_Toc451526663)

[២.៣.៤. Lower-Level Diagram 12](#_Toc451526664)

[២.៣.៥. Strategies for Developing DFDs 12](#_Toc451526665)

[២.៣.៦. Data Dictionary 13](#_Toc451526666)

[២.៤ Data Design Terminology 13](#_Toc451526667)

[២.៤.១. Definitions 13](#_Toc451526668)

[២.៤.២. Key Fields 14](#_Toc451526669)

[២.៤.៣. Data Relationships 15](#_Toc451526670)

[២.៥. MongoDB 17](#_Toc451526671)

[២.៤. Meteor JS 20](#_Toc451526672)

[២.៦. វេបសាយ (Website) 21](#_Toc451526673)

[២.៦.១ Static Website 21](#_Toc451526674)

[២.៦.២ Dynamic Website 22](#_Toc451526675)

[២.៧. សេចក្តីណែនាំអំពីភាសាកូដ 23](#_Toc451526676)

[២.៧.១. ភាសា HTML និង HTML5 23](#_Toc451526677)

[២.៧.២. ភាសា CSS និង CSS3 24](#_Toc451526678)

[២.៧.៣ ភាសា JavaScript 25](#_Toc451526679)

[២.៧.៤. Jquery 26](#_Toc451526680)

[២.៧.៥ Ajax 26](#_Toc451526681)

[២.៧.៦ JSON 27](#_Toc451526682)

[២.៧.៧ Bootstrap 27](#_Toc451526683)

[III. វិធីសាស្ត្រនៃការស្រាវជ្រាវ 28](#_Toc451526684)

[៣.១. វិធីសាស្រ្តនៃការសិក្សា 28](#_Toc451526685)

[៣.២. ប្រភេទទិន្នន័យ និងបច្ចេកទេសប្រមូលទិន្នន័យ 28](#_Toc451526686)

[៣.២.១ ប្រភេទទិន្នន័យ 28](#_Toc451526687)

[៣.២.២ បច្ចេកទេសប្រមូលទិន្នន័យ 29](#_Toc451526688)

[៣.៣.១ ការវិភាគទិន្នន័យ 29](#_Toc451526689)

[៣.៤. ​វិធីសាស្រ្តរចនា និង អភិវឌ្ឍកម្មវិធី 30](#_Toc451526690)

[៣.៥. វិធីសាស្រ្តសាកល្បង 30](#_Toc451526691)

[IV. លទ្ធផលនៃការស្រាវជ្រាវ 32](#_Toc451526692)

[៤.១. ការវិភាគប្រព័ន្ធ 32](#_Toc451526693)

[៤.១.១ ប្រព័ន្ធបច្ចុប្បន្ន 32](#_Toc451526694)

[៤.១.២ ប្រព័ន្ធថ្មី 32](#_Toc451526695)

[៤.១.៣ Entity Relationship Diagram (E-R Diagram) 32](#_Toc451526696)

[៤.២. Data Description and Dictionary for Collection 33](#_Toc451526697)

[៤.២.១ Data Description 33](#_Toc451526698)

[៤.២.២ Data Dictionary for Collection 33](#_Toc451526699)

[៤.៣. ការសាងសង់ Web Application 40](#_Toc451526700)

[៤.៣.១ ទម្រង់នៃការបង្កើត Web Application 40](#_Toc451526701)

[៤.៣.២ .meteor 41](#_Toc451526702)

[៤.៣.៣ cpanel 41](#_Toc451526703)

[៤.៣.៤ rabbit 43](#_Toc451526704)

[៤.៤. ការបង្កើត User Interface 43](#_Toc451526705)

[៤.៤.១ ការរចនា Form 43](#_Toc451526706)

[V. ការពិភាក្សា 62](#_Toc451526707)

[៥.១. ប្រព័ន្ធចាស់ 62](#_Toc451526708)

[៥.២. ប្រព័ន្ធថ្មី 63](#_Toc451526709)

[៥.៣. សេចក្តីសំយោគ ៦៣](#_Toc451526710)

[VI. សេចក្តីសន្និដ្ឋាន និងអនុសាសន៍ ៦៤](#_Toc451526711)

[៦.១. សេចក្តីសន្និដ្ឋាន 64](#_Toc451526712)

[៦.២. អនុសាសន៍ 64](#_Toc451526713)

[ឯកសារយោង 65](#_Toc451526714)

[ឧបសម្ព័ន្ធ 66](#_Toc451526715)

# សេចក្តីថ្លែងអំណរគុណ

យើងខ្ញុំទាំងអស់គ្នាជានិស្សិតនៃសាកលវិទ្យាល័យ បៀលប្រាយ ផ្នែកវិទ្យាសាស្រ្ត ​​និ​ងបច្ចេកវិទ្យា ឯកទេស បច្ចេកវិទ្យាព័ត៌មាន កម្មវិធីគម្រោងរៀបចំធ្វើសារណានៅវគ្គទី១ ជំនាន់ទី៨ ឆ្នាំទី៤​ ដែលមានរាយគោត្តនាម និងនាមដូចខាងក្រោម៖

**១. ហេង យូស៊ួរ ២. ឡោ រ៉ាយុត ៣. ប៉ុន ភក្តី**

**៤. មឿង សេរីរតនៈ ៥. ទៀប ពិសិដ្ឋ**

សូមថ្លែងអំណរគុណចំពោះ

លោកសាកលវិទ្យាធិការ លោកសាកលវិទ្យាធិការរង និងលោកសាស្រ្តាចារ្យទាំងអស់នៃសាកលវិទ្យាល័យ បៀលប្រាយ ដែលបានផ្តួចផ្តើមគំនិតបង្កើតទីកន្លែងដែលសម្រាប់​ធ្វើការអប់រំ និងផ្តល់ចំណេះដឹងដល់យុវជន យុវនារី ឲ្យស្រូបយក​នូវចំណេះដឹងទាំងឡាយណាដែលធ្វើជាបំណងខ្លួនដែលយើងចង់បាន និងយកចំណេះដឹងទាំងអស់នោះ​​ទៅបម្រើសង្គមជាតិដែលធ្វើឲ្យសង្គមមានការរីកចំរើនកាន់តែខ្លាំង និងជាទំពាំងស្នងឬស្សីទៅថ្ងៃអនាគត ។

ជាពិសេស លោកឪពុក អ្នកម្តាយ​ ដែលបានផ្ដល់កំណើត និងចិញ្ចឹមបីបាច់ថែរក្សាយើងខ្ញុំយ៉ាងយកចិត្តទុកដាក់​ ព្រមទាំង លោកអ៊ុំ ពូ មីង បងប្អូនប្រុស ស្រីរបស់យើងខ្ញុំ និង មិត្តភក្តិរួមថ្នាក់ទាំងអស់ដែលបាន ជួយលើកទឹកចិត្ត និងផ្តល់ជាមតិយោបល់យ៉ាងមានប្រយោជន៍ក្នុងការចងក្រងសៀវភៅសារណាបទនេះ។

លោកគ្រូសាស្ត្រាចារ្យ ឡោ សូត្រ ដែលបានដឹកនាំក្រុមរបស់យើងខ្ញុំទាំងអស់គ្នាក្នុងការសរសេរសារណាបញ្ចប់ឆ្នាំសិក្សានេះ ហើយរូបលោកបានចំណាយពេលវេលាដ៏មានតម្លៃនៅក្នុងការកែសម្រួល អត្ថបទ និងបានផ្តល់នូវគំនិតសំខាន់ដល់សារណាក្រុមរបស់យើងខ្ញុំបានសម្រេចលទ្ធផល និងលេចចេញជារូបរាងឡើង ។

លោកគ្រូ អ្នកគ្រូ និងលោកគ្រូសាស្ត្រាចារ្យទាំងអស់ដែលមានវត្តមាន នៅក្នុងការបង្រៀនយើងខ្ញុំទាំងអស់គ្នា អស់ពីកម្លាំងកាយចិត្តរបស់លោកដោយមិនមានការត្អូញត្អែរបន្តិចឡើយ​​​ ។

ហើយជាចុងក្រោយ ក្រុមរបស់យើខ្ញុំសូមប្រសិទ្ធិពរជ័យ សិរីសួស្តី ជ័យមង្គល លោក លោកស្រី សូមជួបប្រទះនូវសេចក្តីសុខ សេចក្តីចម្រើន និងទទួលជោគជ័យគ្រប់ភារកិច្ច គ្រប់ក្រុមគ្រួសារ ។

# មូលន័យសង្ខេបសារណា

សព្វថ្ងៃនេះ គេសង្កេតឃើញថាប្រទេសកម្ពុជាយើងកំពុងតែមានការរីកចម្រើនស្ទើរតែគ្រប់ផ្នែក ជាពិសេសផ្នែកបច្ចេកវិទ្យា ដែលវាជាផ្នែកសំខាន់ដ៏ធំមួយនៅក្នុងការអភិវឌ្ឍន៍សង្គមឲ្យកាន់តែមានភាពរីកចម្រើនកាន់តែខ្លាំងជាងមុនមួយកម្រិតទៀត។ ជាក់ស្តែង ដូចជា​ នៅតាមគ្រឹះស្ថានសិក្សា ស្ថាប័នរដ្ឋ​ និងឯកជន ក្រសួង មន្ទីរ​​ ក្រុមហ៊ុន និងអង្គភាព​ ដ៏ទៃមួយចំនួនទៀតដែលបានប្រើ និងរក្សាទុកទិន្នន័យ និងធ្វើឲ្យមានការ​​ចំណេញពេលវេលា កាន់តែប្រសើរជាមុន និងមានភាពងាយស្រួលនៅក្នុងការគ្រប់គ្រង ។

ការសិក្សាស្រាវជ្រាវនេះគឺមានគោលដៅដើម្បីអភិវឌ្ឍកម្មវិធីប្រព័ន្ធគ្រប់គ្រងព័ត៌មានរបស់ក្រុមហ៊ុន Rabbit IT Solution។ ដោយសារតែការគ្រប់គ្រងដែលប្រើដោយការសរសេរកត់ត្រាដោយដៃដែលមានលក្ខណៈមិនទាន់សម័យ ហើយប្រឈមមុខនឹងហានិយភ័យខ្ពស់​។​ លើពីនេះ វាមានការលំបាកនៅក្នុងការរក្សាទុកទិន្នន័យ​ និងអាចមានការបាត់បង់ដោយប្រការណាមួយ​​ ការគ្រប់គ្រងទិន្នន័យអាចបានតិច និងធ្វើឲ្យមានការភាន់ច្រឡំច្រើននៅក្នុងការស្វែងរកទិន្នន័យនៅពេលណាមួយមិនខាន។ ដើម្បីឆ្លើយតបទៅនឹងតម្រូវការរបស់ក្រុមហ៊ុន Rabbit IT Solution ក្រុមរបស់យើងខ្ញុំបានសម្រេចបង្កើតនូវប្រព័ន្ធគ្រប់គ្រងទិន្នន័យថ្មីនេះឡើងដោយប្រើប្រាស់បច្ចេកវិទ្យា​ថ្មីៗ​ដូចជា Meteor JS Framework ជាមួយ MongoDB, HTML5, CSS3, JQUERY, JAVASCRIPT, AJAX និង JSON។ នៅក្នុងការស្រាវជ្រាវនេះ យើងនឹងធ្វើយ៉ាងណាដើម្បីធ្វើឲ្យការគ្រប់គ្រងកាន់តែមានភាពងាយស្រួល ដោយបានរៀបចំចងក្រង Database System ដែលអាចបញ្ចូលទិន្នន័យ លុប​ ស្វែងរក កែប្រែ មើលរបាយការណ៍ និងធ្វើការសង្ខេបទិន្នន័យទៅតាមតម្រូវការដែលយើងចង់បាន និងជួយសន្សំសំចៃពេលវេលាបានច្រើនជាងមុន។

ដោយយល់ឃើញនូវអត្ថប្រយោជន៍យ៉ាងច្រើនបែបនេះ របស់ប្រព័ន្ធបច្ចេកវិទ្យា រួមនឹង​តម្រូវការរបស់ក្រុមហ៊ុន Rabbit IT Solution កំពុងតែមានការមមាញឹកនៅក្នុងការគ្រប់គ្រងទៅលើការលក់ និងផ្នែកផ្សេងៗទៀត ដូច្នេះហើយជាការផ្តល់​ឱកាស និងជាបទពិសោធន៍ដ៏មានតម្លៃមួយដល់ក្រុមរបស់យើងខ្ញុំ នៅក្នុងការធ្វើការចុះសិក្សាស្រាវជ្រាវ​ និងបានបន្សល់ទុកនូវឯកសារសម្រាប់អ្នកសិក្សាស្រាវជ្រាវនៅជំនាន់ក្រោយៗថែមទៀតផង។

# តារាង

ចំណងជើង ​​ ទំព័រ

តារាង ១ 1

តារាង ២ 1

តារាង ៣ 1

# តួលេខ

ចំណងជើង ​​ ទំព័រ

តួលេខ ១ 1

តួលេខ ២ 1

# រូបភាព

ចំណងជើង ​​ ទំព័រ

រូបភាពទី ១ : មុខងារនៃប្រព័ន្ធព័ត៌មាន (Function of Information System) 1

រូបភាពទី ២ 1

រូបភាពទី ៣ 1

រូបភាពទី ៤ 1

រូបភាពទី ៥ 1

រូបភាពទី ៦ 1

រូបភាពទី ៧ 1

រូបភាពទី ៨ 1

រូបភាពទី ៩ 1

រូបភាពទី ១០ 1

រូបភាពទី ១១ 1

រូបភាពទី ១២ 1

រូបភាពទី ១៣ 1

រូបភាពទី ១៤ 1

រូបភាពទី ១៥ 1

រូបភាពទី ១៦ 1

រូបភាពទី ១៧ 1

រូបភាពទី ១៨ 1

រូបភាពទី ១៩ 1

រូបភាពទី ២០ 1

រូបភាពទី ២១ 1

រូបភាពទី ២២ 1

រូបភាពទី ២៣ 1

រូបភាពទី ២៤ 1

រូបភាពទី ២៥ 1

រូបភាពទី ២៦ 1

រូបភាពទី ២៧ 1

រូបភាពទី ២៨ 1

រូបភាពទី ២៩ 1

រូបភាពទី ៣០ 1

រូបភាពទី ៣១ 1

រូបភាពទី ៣២ 1

រូបភាពទី ៣៣ 1

រូបភាពទី ៣៤ 1

រូបភាពទី ៣៥ 1

រូបភាពទី ៣៦ 1

# បញ្ជីពាក្យអក្សរកាត់

AJAX : Asynchronous JavaScript and XML

CSS : Cascading Style Sheets

DBMS : Database management systems

HTML : HyperText Markup Language

# សេចក្តីផ្តើម

នាពេលបច្ចុប្បន្ននេះ យើងបានសងេ្កតឃើញថា វិស័យបច្ចេកវិទ្យា និងព័ត៌មានវិទ្យា (Information Technology) មានសន្ទុះយ៉ាងខ្លាំងក្លានៅក្នុងការចូល​រួមអភិវឌ្ឍន៍​ប្រទេស​នានា​នៅលើសកលលោកឲ្យមានការរីកចម្រើនទៅមុខជាលំដាប់។ ដោយ​សារតែវិស័យ​ព័ត៌មាន​វិទ្យា និងបច្ចេកវិទ្យា​ត្រូវបានយកទៅប្រើប្រាស់ស្ទើរតែគ្រប់វិស័យដូចជា ការគ្រប់គ្រងនៅ​ក្នុងវិស័យពាណិជ្ជកម្ម​​ វិស័យសេដ្ឋកិច្ច វិស័យសង្គមកិច្ច វិស័យកសិកម្ម ​​ វិស័យ​អប់រំ វិស័យទេសចរណ៍ និងការគ្រប់គ្រងបុគ្គលិកនៅតាមបណ្តាក្រសួង អង្គការ ក្រុមហ៊ុន និងការងារជាច្រើនផ្សេងទៀត។ ទន្ទឹមនឹងនេះផងដែរ ប្រទេសកម្ពុជាយើង នាពេលបច្ចុប្បន្ន រាជរដ្ឋាភិបាលក៏មានការយកចិត្តទុក​ដាក់ក្នុងការអភិវឌ្ឍន៍ និងលើកស្ទួយនូវវិស័យនេះផងដែរ ដើម្បីឈានទៅរកការរីកចម្រើន​ដូចប្រទេសនានានៅជុំវិញពិភព លោក។ ប្រសិនបើយើង​ក្រឡេក​មើលទៅក្រុមហ៊ុនមួយចំនួន យើងនឹងឃើញថាក្រុមហ៊ុនទាំងអស់នោះបាននិងកំពុងតែធ្វើការវិនិយោគទៅលើវិស័យនេះយ៉ាងខ្លាំងក្លា ។

ដោយសារមើលឃើញ សារៈប្រយោជន៍ជាច្រើនក្នុងការផ្លាស់ប្តូរព័ត៌មានទូទាំងពិភពលោក ដោយប្រើម៉ាស៊ីនកុំព្យូទ័រនៅក្នុងការបញ្ចូលព័ត៌មាន ​និង​អាចប្រើប្រាស់ដើម្បីគ្រប់​គ្រង ឬ រក្សាទុកទិន្នន័យមួយចំនួនដែលអាចជាជំនួយការងារ​ដល់មនុស្ស ដែលមាន​ការធ្វើ​ដោយដៃ ដោយយកម៉ាស៊ីនកុំព្យូទ័រមកជំនួសវិញ វាប្រើប្រាស់បានលឿន និងចំណេញ​ពេលវេលា ចំណេញថវិការ ចំណេញការងារ និងប្រកបដោយប្រសិទ្ធភាពខ្ពស់ ។

## ១.១.​ លំនាំដើមនៃការស្រាវជ្រាវ

ក្រុមហ៊ុន Rabbit IT Solution គឺជាក្រុមហ៊ុនមួយដែលផ្តល់សេវាកម្មជាច្រើនដូចជា ការលក់ប្រព័ន្ធ​គ្រប់​គ្រង​លក្ខណៈ Web-based Application និងគេហទំព័រ កុំព្យូទ័រ(Desktop, Laptop) គ្រឿងបន្លាស់ និងផ្តល់ការបង្រៀនជំនាញកុំព្យូទ័រដល់សិស្ស និស្សិតផងដែរ ហើយក្រុមហ៊ុននេះត្រូវបានបង្កើតឡើងក្នុងឆ្នាំ ២០១៤ ដែលមានទីតាំងស្ថិតនៅតាមបណ្តោយ​ផ្លូវជាតិ​លេខ៥​ ភូមិរំចេក៤ សង្កាត់រតនៈ ក្រុងបាត់ដំបង ខេត្តបាត់ដំបង (ទល់មុខសាខាពន្ធដារ ខេត្តបាត់ដំបង)​។ ដោយសារតែពីមួយថ្ងៃទៅមួយថ្ងៃ មានអតិថិជនមកទិញ ប្រព័ន្ធ​គ្រប់​គ្រង​លក្ខណៈ Web-based Application និងគេហទំព័រ កាន់តែច្រើនឡើង ហើយសព្វថ្ងៃនេះក្រុមហ៊ុនមិនទាន់មានប្រព័ន្ធគ្រប់គ្រងទិន្នន័យនៅឡើយទេ គឺកំពុងតែប្រើប្រាស់ការកត់ត្រាដោយដៃផង និងប្រើប្រាស់កម្មវិធី Microsoft Excel ផង ទើបបង្កអោយមានភាពលំបាកក្នុងការគ្រប់គ្រង និងការងារមានភាពយឺតយ៉ាវ។

ដោយយល់ឃើងថា ខាងក្រុមហ៊ុនមានភាពលំបាក និងកង្វះខាតច្រើន ដូច្នេះហើយ ទើបបានជាក្រុមនិស្សិតយើងខ្ញុំបានស្នើសុំទៅកាន់អ្នកគ្រប់គ្រងក្រុមហ៊ុន Rabbit IT Solution ដើម្បីចុះទៅធ្វើការស្រាវជ្រាវ និងប្រមូលទិន្នន័យរៀបចំបង្កើតនូវប្រព័ន្ធថ្មីមួយ ដែលទាក់ទងទៅនឹងការគ្រប់គ្រងការលក់ ដើម្បីអោយមានភាពងាយស្រួលជាងមុន។

## ១.២. ចំនោទបញ្ហា

ដោយសារតែ ក្រុមនិស្សិតយើងខ្ញុំសង្កេតឃើញថា ក្រុមហ៊ុនមានភាពលំបាកក្នុងការគ្រប់គ្រងលើផ្នែកផ្សេងៗ រួមមានដូចជា៖

* + ពិបាកក្នុងការសែ្វងរកព័ត៌មានរបស់អតិថិជន​
  + ​ព័ត៌មានអតិថិជនងាយបាត់បង់ និងគ្មានសុវត្តិភាព​
  + បិទបញ្ជីបា្រក់ចំណូលប្រចាំខែមិនមានភាពងាយស្រូល​
* ប្រាក់ចំណូលដែលទទួលបានមិនមានភាពច្បាស់លាស់
* ពិបាកក្នុងការតាមដានកាលបរិចេ្ឆទផុតកំណត់សេវាថែទាំរបស់អតិថិជននីមួយៗ
* ការចេញ Quotation និង​ Contract ត្រូវចំណាយពេលយូរ និងមានភាពសាំញ៉ាំ
* ពិបាកមើលរបាយកាណ៍នីមួយៗ
* ពិបាករកអតិថិជន​ដែលមិនទាន់បង់ប្រាក់​អស់​
* មិនមានភាពងាយស្រួលក្នុងការចេញ contract ដើម្បីទិញសេវាកម្ម​ថែទាំ​ សា​ជា​ថ្មី
* ចំណាយពេលវេលាយូរ​ និងលំហូរការងារយឺតយ៉ាវ ។

## ១.៣. គោលបំណងនៃការស្រាវជ្រាវ

### ១.៣.១ គោលបំណងរួម

គោលបំណងនៃការសិក្សាស្រាវជ្រាវនេះ គឺត្រូវបង្កើតឲ្យបាននូវប្រព័ន្ធគ្រប់គ្រងព័ត៌មាន​ដែល​មានលក្ខណៈជា Web​-based Application មួយសម្រាប់ប្រតិបត្តិការក្នុងការគ្រប់គ្រង ទិន្នន័យ​នៃការលក់ប្រព័ន្ធ​គ្រប់គ្រងលក្ខណៈជា Web-based Application និង​​ គេហទំព័រ​​ របស់ក្រុមហ៊ុន Rabbit IT Solution ។

### ១.៣.២ គោលបំណងជាក់លាក់

គោលបំណងជាក់លាក់នៃការសិក្សាស្រាវជ្រាវរួមមាន៖

* ស្វែងយល់ពីការគ្រប់គ្រងការលក់ និងបង្កើត Database​ System សម្រាប់ Rabbit IT​Solution ដោយប្រើប្រាស់ Meteor JS Framework និង MongoDB ។
* ស្វែងយល់ពីភាពងាយស្រួលនៃការមើល និងស្វែងរកទិន្នន័យ ហើយបង្កើត Interface Website ដោយប្រើប្រាស់ភាសា HTML5, CSS3, Jquery, JavaScript និង Bootstrap3 ។
* ធ្វើការភ្ជាប់ពី Interface Website ទៅកាន់ Database​ System ដោយប្រើប្រាស់ Meteor JS Framework ជាមួយកូដ JavaScript, Jquery និង Json ។
* ស្វែងយល់ពីការផ្តល់នូវសុវត្ថិភាពខ្ពស់ ក្នុងការរក្សាទុកទិន្នន័យ ការ Backup​ ទិន្នន័យ​​ទុក និងការ Restore ទិន្នន័យយកមកប្រើប្រាស់ឡើង​​​​​​​វិញ។
* ធ្វើអោយការមើលរបាយការណ៍ប្រចាំខែនីមួយៗកាន់តែមានភាពងាយស្រួល ។

## ១.៤. ទំហំ​ និងដែនកំណត់នៃការស្រាវជ្រាវ

### ១.៤.១ ទំហំនៃការស្រាវជ្រាវ

ការសិក្សាស្រាវជ្រាវនេះគឺ ក្រុមនិស្សិតយើងខ្ញុំលើកយកតែការគ្រប់គ្រងទិន្នន័យនៃការលក់ប្រព័ន្ធ​គ្រប់គ្រងដែលមានលក្ខណៈជា Web-based Application និងគេហទំព័រ របស់ក្រុមហ៊ុន Rabbit IT Solutionមកធ្វើការ​សិក្សា​ស្រាវជ្រាវ​តែប៉ុណ្ណោះ ។

### ១.៤.២​ ដែនកំណត់នៃការស្រាវជ្រាវ

ដោយយោងតាមពេល​វេលា​នៃ​ការស្រាវជ្រាវ និងធនធានមានកំណត់ ទើប​​តម្រូវអោយក្រុមនិស្សិតយើង​ខ្ញុំធ្វើការសិក្សា​ស្រាវជ្រាវ​​ដោយផ្តោត និងយកចិត្តទុក​ដាក់​​ទៅលើការគ្រប់គ្រងការលក់ប្រព័ន្ធគ្រប់គ្រងលក្ខណៈជា Web-based Application និងគេហទំព័រ តែប៉ុណ្ណោះដែលផ្តោតទៅលើចំនុចមួយចំនួនដូចខាងក្រោម៖

* គ្រប់គ្រងការលក់ប្រព័ន្ធ (Web-based Application)
* គ្រប់គ្រងការបង្កើតគេហទំព័រ (Website Design)
* គ្រប់គ្រងអតិថិជន
* គ្រប់គ្រងភ្នាក់ងារនាំអតិថិជន (Agent)
* គ្រប់គ្រងប្រព័ន្ធថ្មី និងប្រព័ន្ធដែលមានស្រាប់
* គ្រប់គ្រងការចុះកិច្ចសន្យា (Contract)
* គ្រប់គ្រងការចេញ Quotation
* គ្រប់គ្រងការចេញ System Feature
* គ្រប់គ្រងការបង់ប្រាក់
* គ្រប់គ្រងការទិញសេវាកម្ម (service)
* ការមើលរបាយការណ៍ Quotation
* ការមើលរបាយការណ៍ ភ្នាក់ងារនាំអតិថិជន (Agent)
* ការមើលរបាយការណ៍ កិច្ចសន្យា
* ការមើលរបាយការណ៍ ទិញទីស្នាការរបស់អតិថិជន (Office)
* ការមើលរបាយការណ៍ ទិញសេវាកម្មថែទាំ
* ការមើលរបាយការណ៍ បង់ប្រាក់របស់អតិថិជន (customer)
* ការមើលរបាយការណ៍ ប្រាក់សមតុល្យចុងគ្រា
* ការមើលរបាយការណ៍ អតិថិជនដែលមិនទាន់បង់ប្រាក់អស់
* ការមើលរបាយការណ៍ ទិញសេវាកម្មថែទាំសារជាថ្មី ។

ហើយនៅក្នុងការរៀបចំបង្កើតប្រព័ន្ធនេះឡើង ក្រុមនិស្សិតយើងខ្ញុំបានកំណត់យកនូវកម្មវិធីមួយ​ចំនួន​ដើម្បីប្រើប្រាស់មានដូចជា៖

* RoboMongo ជាកម្មវិធីសម្រាប់មើល កែប្រែ និងលុបទិន្នទ័យ​ដែលបាន​កត់ត្រា។
* Webstorm និង Atom ដែលជា IDE និង Code Editor សម្រាប់សរសេរ​កូដផេ្សង​ៗ​ ដើម្បីអោយបង្ហាញទិន្នន័យនៅលើ Browser។
* Chrome និង Firefox ជា Browser សម្រាប់បង្ហាញទិន្នន័យ និងរបាយការណ៍​នានាដើម្បីមើល ឬបោះពុម្ភ។

## ១.៥. សារៈសំខាន់នៃការស្រាវជ្រាវ

ការសិក្សាស្រាវជ្រាវក្នុងការបង្កើតប្រព័ន្ធគ្រប់គ្រងការលក់ប្រព័ន្ធ​គ្រប់​គ្រង​លក្ខណៈ Web-based Application និងគេហទំព័រ មានអត្ថប្រយោជន៍ជាច្រើនដែលបង្កភាពងាយស្រួលជាងកាតប្រើដោយដៃ ដោយមានការគ្រប់គ្រងការលក់ប្រព័ន្ធ គ្រប់គ្រងប្រាក់ចំនូល គ្រប់គ្រងអតិថិជន ដើម្បីអោយក្រុមហ៊ុនកាន់តែមានភាពរីកចំរើនជាងមុន និងមានភាពងាយស្រួលក្នុងការគ្រប់គ្រងអាជីវកម្មប្រកបដោយប្រសិទ្ធភាពខ្ពស់។

## ១.៦. ការពន្យល់ពាក្យគន្លឹះ

ដើម្បីឲ្យមានភាពស្រួល និងឲ្យកាន់តែយល់ច្បាស់ពីអត្ថន័យនៃពាក្យមួយចំនួនដែលពាក់ព័ន្ធ ក្រុមសិក្សាស្រាវជ្រាវយើងខ្ញុំសូមពន្យល់ និងបកស្រាយដូចខាងក្រោម៖

* ប្រព័ន្ធ៖ ការចងក្រង ការប្រមូលផ្ដុំនូវរបស់អ្វីមួយ។ ក្នុងន័យនេះប្រព័ន្ធសំដៅព័ត៌មាន​ និងការចំណាយចំណូលរបស់ក្រុមហ៊ុន ។
* គ្រប់គ្រង៖ គឺជាដំណើរការនៃការធ្វើផែនការ ការរៀបចំ ការដឹកនាំ និងការត្រួតពិនិត្យ​ ធនធានដែលមានស្រាប់ដើម្បីសម្រេចគោលបំណង។ ក្នុងន័យនេះការគ្រប់គ្រងមាន័យថា ជាការរៀបចំប្រព័ន្ធព័ត៌មាន​ និងការបង់ប្រាក់របស់ក្រុមហ៊ុន ឲ្យមានរបៀបរៀបរយ និងច្បាស់លាស់។
* Web-based Application ៖ គឺជាកម្មវិធីដែលគេសរសេរឡើងអោយដំណើរនៅលើ Web Browser ។
* គេហទំព័រ (Website)៖ **គឺជាការតភ្ជាប់ចូលគ្នារវាង** webpages **ជាច្រើនដែលផ្ទុកនូវ អត្ថបទ រូបភាព វីដេអូ និងឯកសារផ្សេងៗទៀត ។ ហើយ** Website បានផ្ដល់អត្ថប្រយោជន៍យ៉ាងច្រើនដល់យើង ដូចជា ធ្វើឲ្យងាយស្រួលក្នុងការស្រាវជ្រាវស្វែងរកទិន្នន័យ និងឯកសារផ្សេងៗ យ៉ាងឆាប់រហ័ស មិនថាអ្នកណា នៅទីកន្លែងណា នៅពេលណាក៏បាន សុទ្ធតែអាចចូលមើល Website បាន ។
* Collection ៖ ជាបណ្តុំនៃព័ត៌មាន (information) ដែលមានលក្ខណៈជា Object ហើយមិនប្រើបា្រស់ Relationship ដូ​ច SQL Server ទេ តែវាអាចជំនួសដោយការ Embed Object ដែលផ្តល់ភាពងាយស្រួលក្នុងការ គ្រប់គ្រង ស្វែងរកទិន្នន័យ ផ្សេងៗ ដែលបានរក្សាទុក។
* Embed Object ៖ ជាការទាញយក Document (បើក្នុង SQL ជា Record) មួយទៅដាក់ក្នុង Document នៃ Collection (បើក្នុង SQL ជា Table) មួយទៀតជាលក្ខណៈ Object ឬ Array នៃ Object ។
* RoboMongo​ ៖ គឺជាកម្មវិធី open source shell-centric cross-platform សម្រាប់គ្រប់គ្រង MongoDB (Ex: Admin GUI) ។ វា embed JavaScript engine ណាដែលដូចគ្នាដោយប្រើប្រាស់ MongoDB 2.2 mongo shell​ ហើយវាមានមុខងារ auto complete សម្រាប់ objects ទាំងអស់ដែល JavaScript ស្គាល់។ វាក៏បានបញ្ចូល auto complete សម្រាប់ databases, collections, និងសូម្បីតែ document objects។
* IDE ៖ (Integrated Development Environment) គឺជាកម្មវិធីមួយដែលផ្តល់សេវាកម្មទូលំទូលាយសម្រាប់អ្នកអភិវឌ្ឍន៍កម្មវិធី(Developer) មានភាពងាយក្នុងការអភិវឌ្ឍន៍កម្មវិធីផ្សេងៗ ដោយវាបានបង្កើនផលិតភាពរបស់អ្នកអភិវឌ្ឍន៍កម្មវិធីពីព្រោះ IDE ខ្លះបានដាក់បញ្ជូលមុខងារមួយចំនួនដូចជា compiler, interpreter, modifier, deploying និងdebugger ជាដើម។ ហើយ IDE មួយចំនួនធំត្រូវបានគេបង្កើតឡើងសម្រាប់ programming language ណាមួយជាក់លាក់តែម្តង។ ឧទាហរណ៍ ៖ Webstorm, Phpstorm, Eclipse, ActiveState Komodo, IntelliJ IDEA, MyEclipse, Oracle JDeveloper, NetBeans, Microsoft Visual Studio និងXcode ។
* Code Editor ៖ (Source Code Editor) គឺជាកម្មវិធីដែលគេបានរចនាឡើងសម្រាប់កែ Source Code នៃកម្មវិធីណាមួយ។ វាអាចជាប្រភេទកម្មវិធី standalone ឬអាចជាកម្មវិធី ដែល build-in ជាមួយ IDE ឬ web browser។ ហើយវាជាកម្មវិធីមូលដ្ឋានបំផុតសម្រាប់អ្នកអភិវឌ្ឍន៍កម្មវិធីក្នុងការសរសេរ និងកែប្រែ Source Code។ Ex: Sublime, Notepad++, Atom
* Interface ៖ គឺជាផ្ទៃបង្ហាញសម្រាប់អ្នកប្រើបា្រស់មានភាពងាយស្រួលប្រើ​ បា្រស់។
* Source code ៖ មានន័យថាបណ្តុំនៃកូដជាច្រើនលាយឡំគ្នា។
* Front-End​ ៖ គឺជាអ្វីៗដែលជាប់ទាក់ទងនឹងអ្វីដែលអ្នកប្រើបានមើលឃើញដោយ​​រួមបញ្ចូលទាំងការរចនា និងភាសាកូដមួយចំនួនដូចជា HTML, CSS និង JavaScript ត្រូវបានគ្រប់គ្រងដោយ browser ។
* Back-End ៖ ឬអាចហៅថា “server-side” សំដៅទៅលើអ្វីគ្រប់យ៉ាងដែលអ្នកប្រើមិនអាចមើលឃើញនៅក្នុង browser មាន databases និង servers ។ ជាធម្មតាមនុស្សដែលធ្វើការនៅផ្នែក back-end គេហៅថាអ្នកសរសេរកម្មវិធី ឬ ជា developers ។ Back-end developers ភាគច្រើនមានការព្រួយបារម្ភពីសុវត្ថិភាព, រចនាសម្ព័ន្ធ និងការគ្រប់គ្រងផ្សេងៗ ។
* Open-Source ៖ គឺជា software ដែលកូដប្រភពដើមត្រូវបានធ្វើ និងដាក់ឱ្យប្រើប្រាស់ដោយសេរី និងអ្នកដ៏ទៃអាចយកកូដទៅបែងចែក និងកែប្រែបាន។
* Cross-platform ៖ គឺជាប្រភេទនៃ computer software ឬ computing methods និង concepts ដែលអាចចូលរួមចំណែក ឫដំណើរការនៅលើ multiple​computer platforms​ ។ Cross-platform software អាចចែកចេញជាពីរប្រភេទ៖ ទីមួយវាទាមទារ individual building ឬ compilation សម្រាប់ platform នីមួយៗដែលវា supports និង ទីពីរអាចជាកម្មវិធីដែលដំណើរការ directly នៅលើ platform ផ្សេងៗ ដោយមិនចាំបាច់ special preparation ។

ឧទាហរណ៍៖ កម្មវិធីប្រភេទ cross-platform ​ អាចដំណើរការនៅលើ Microsoft Windows x86 architecture, Linux x86 architecture និង Mac OS X ឬ PowerPC​ or x86 based Apple Macintosh systems.

* Database ៖ គឺជាការប្រមូលរៀបចំទិន្នន័យរបស់ collection។ ហើយក៏ជា collection នៃ schemas, tables, queries, reports, views និង objects ដទៃទៀត។
* Entity ៖​ អាចជាមនុស្ស ក្រុមហ៊ុន ការិយាល័យ ឬស្ថាប័ននានា ផ្តល់ទិន្នន័យដល់ប្រព័ន្ធ ឬទទួលលទ្ធផលពីប្រព័ន្ធ ។ ហើយវាត្រូវបានចែកចេញជាពីរគឺ Weak Entity និងStrong Entity។
* Servers ៖……………………
* Client ៖……………………

# រំលឹកទ្រឹស្តី

## ២.១. សញ្ញាណទូទៅនៃប្រព័ន្ធព័ត៌មាន

### ២.១.១. បច្ចេកវិទ្យាព័ត៌មានជាអ្វី ?

បច្ចេកវិទ្យាព័ត៌មាន (Information Technology) គឺជាការរួមផ្សំឡើងរវាងបច្ចេកវិទ្យា កុំព្យូទ័រ (Software and Hardware) ជាមួយនិងបច្ចេកវិទ្យាទូរគមនាគមន៍ (Telecommunication Technology សម្រាប់ចែករំលែក Data, Image, Video, ជាដើម តាមរយៈ Network) ដើម្បីជួយនូវប្រតិបត្តិការមុខជំនួញ (Business Operation) និងជួយសម្រួលដល់អ្នកគ្រប់គ្រងក្នុងការធ្វើការសម្រេចចិត្ត។

### ២.១.២ ប្រព័ន្ធជាអ្វី ?

ប្រព័ន្ធ គឺជារចនាសម្ព័ន្ធមួយដែលផ្សំឡើងពីផ្នែកផ្សេងៗ​ ដែលមានប្រាស្រ័យទាក់ទងរវាងគ្នានិងគ្នា ដើម្បីសម្រេចចិត្តលើគោលដៅណាមួយ។

### ២.១.៣ ប្រព័ន្ធព័ត៌មានជាអ្វី ?

ប្រព័ន្ធព័ត៌មាន គឺជាការរៀបចំ People, Data, Process, Hardware និង Software ដែលអាចជួយបង្កើនប្រតិបត្តិការប្រចាំថ្ងៃ ព្រមទាំងជួយដោះស្រាយបញ្ហា​ និង​ការសម្រេចចិត្តដល់អ្នកគ្រប់គ្រង និងអ្នកប្រើប្រាស់ដ៏ទៃទៀតក្នុងក្រុមហ៊ុនបានទៀតផង។

ឧទាហរណ៍៖ School Management System, Accounting System, Loan System……។

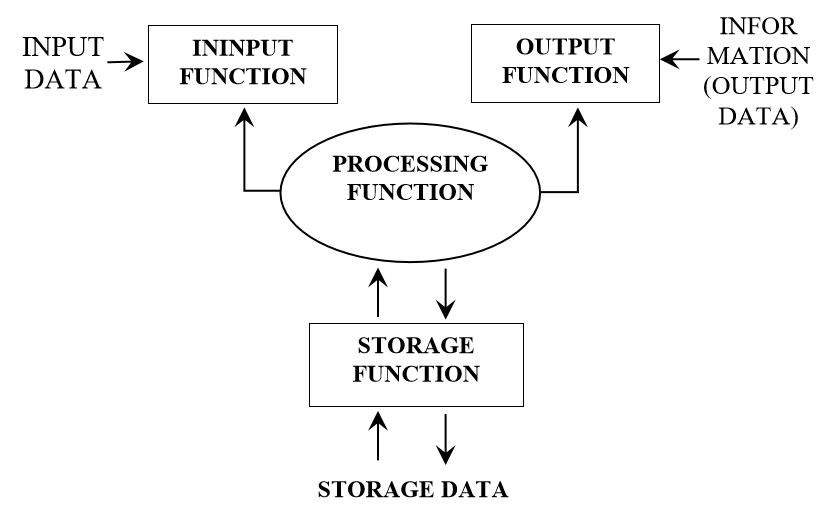
### ២.១.៤​ មុខងារនៃប្រព័ន្ធព័ត៌មាន

ក្នុងប្រព័ន្ធព័ត៌មានមានមុខងារសំខាន់ៗ បួនគឺ៖

* Input Function: ជាការទទួលខុសត្រូវយកនូវការបញ្ចូលទិន្នន័យពីខាងក្រៅប្រព័ន្ធប្រមូលផ្តុំចូលទៅក្នុងប្រព័ន្ធ។ ឧទាហរណ៍៖ ឃីបដ (Keyboard), ​ ម៉ៅ(Mouse), ម៉ៃក្រូហ្វូន (Microphone), ស្គេនន័រ (Scanner), បាកូដស្គេនន័រ (Barcode Scanner) ...។
* Storage Function: សម្រាប់រក្សាទុកនូវទិន្នន័យ (Backup) ដែលបានបញ្ចូល និងទាញយកទិន្នន័យ (Restore) នោះមកវិញនៅពេលដែលប្រព័ន្ធត្រូវការ។ ឧទាហរណ៍៖ ហាតឌីស (Hard Disk), ផ្លាស់ដ្រាយ (Flash Drive), ស៊ីឌី​ ឬ ឌីវីឌី (CD or DVD) ។
* Process Function: សម្រាប់គណនា និង Manipulate នូវទិន្នន័យដែលបាន​

បញ្ចូល។ ឧទាហរណ៍៖ រ៉េម (RAM), ស៊ីភីយូ (CPU) ...... ។

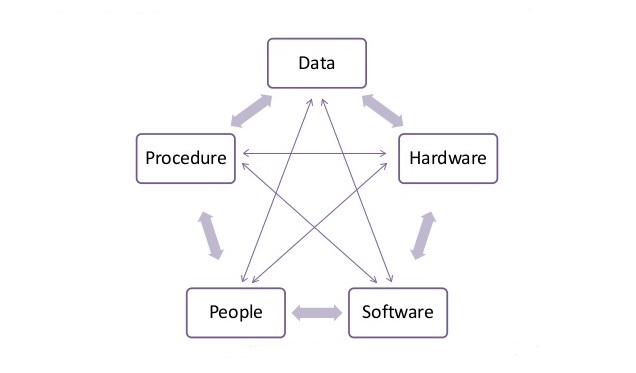
* Output Function: ជាអ្នកផ្តល់នូវព័ត៌មានផ្សេងៗ ដែលត្រូវបានគេហៅថា Information ។ ឧទាហរណ៍៖ ម៉ូនីទ័រ (Monitor), ស្ពីកខ័រ (Speaker), អ៊ែរហ្វូន(Earphone), ព្រីនធ័រ (Printer)។

តួលេខ ២.១ មុខងារនៃប្រព័ន្ធព័ត៌មាន(Function of Information System)

### ២.១.៥​ សមាសធាតុនៃប្រព័ន្ធព័ត៌មាន

សមាសធាតុនៃប្រព័ន្ធព័ត៌មានត្រូវបានគេចែកចេញជា ៥​ សំខាន់គឺ Hardware, Software, Data, People និង Process។ Hardware និង Software គឺជាបច្ចេកវិទ្យាព័ត៌មាន (Information Technology) នៃប្រព័ន្ធ ហើយ People និង Process គឺជាធនធានមនុស្ស (Human Resources) នៃប្រព័ន្ធ។ ចំណែក Data ជាអ្នកចងភ្ជាប់ទំនាក់ទំនងរវាង បច្ចេកវិទ្យាព័ត៌មាន និង ធនធានមនុស្ស។

* Hardware: គឺជាសមាសភាគដំបូងនៃប្រព័ន្ធព័ត៌មានដែលរួមមានដូចជា កុំព្យូទ័រ(Computer) បរិក្ខារទំនាក់ទំនង (Communication Equipment) និងឧបករណ៍(Devices) ផ្សេងទៀតហើយវាត្រូវបានគេដំណើរការ និងបញ្ជារដោយ Software។
* Software: ជាសមាសភាគមួយផ្សេងទៀតនៃប្រព័ន្ធព័ត៌មានដែលផ្ទុកនូងឃ្លាបញ្ជាឬ ការណែនាំប្រាប់ Hardware នូវអ្វីដែលត្រូវធ្វើ។ កុំព្យូទ័រ(Computer) និងបរិក្ខារទំនាក់ទំនង​ (Communication Equipment) មិនអាចដំណើរការដោយគ្មាន Software បានទេ មានន័យថា រាល់គ្រប់ Hardware ត្រូវតែមានឃ្លាបញ្ជាដើម្បីប្រាប់វានូវអ្វីត្រូវធ្វើ។ Software ត្រូវបានគេបែងចែកជា២ប្រភេទគឺ៖
* System Software: ជាប្រព័ន្ធសម្រាប់គ្រប់គ្រង ឬបញ្ជាទៅលើ Hardware ដោយផ្ទាល់។ System Software មានដូចជា៖ MS-DOS, Microsoft Windows, Mac OSX, Linux (Fedora, Ubuntu), Android, IOS…។
* Application Software: គឺជា Software ដែលដំណើរការនៅលើ System Software សម្រាប់ប្រើប្រាស់ក្នុងការបំពេញកាងារផ្សេងៗ មានដូចជា៖ Microsoft Office Word, Excel, Power Point, Webstorm, Google Chrome, Photoshop, Final Cut Pro, Cinema 4D…។
* Data: ជាទិន្នន័យដែលរក្សាទុកនៅលើ Hardware និងដំណើការដោយ Software។ ទិន្នន័យគឺជាវត្ថុធាតុដើមសម្រាប់បំលែងទៅជាព័ត៌មានដែលមានប្រយោជន៍។
* People: ប្រព័ន្ធព័ត៌មានមិនអាចដំណើរការបានដោយផ្ទាល់ខ្លួនវាបានទេ ដូចនេះវាត្រូវការមនុស្ស (People) ដែលជាអ្នកប្រើប្រាស់ដើម្បីធ្វើឱ្យវាដំណើរការ ហើយការងាររបស់មនុស្សគឺ បញ្ចូលទិន្នន័យ និង​ ទាញយកទិន្នន័យដែលបានបញ្ចូល។
* Process: ឬអាចហៅម៉្យាងទៀតថា Procedure ដែលជាឃ្លាបញ្ជា (Instructions) ប្រាប់មនុស្ស ​(People) ពីរបៀបនៃការប្រើប្រាស់ប្រព័ន្ធ។

តួលេខ ២.២ សមាសធាតុនៃប្រព័ន្ធព័ត៌មាន

## ២.២. ប្រព័ន្ធគ្រប់គ្រងដេតាបេស (Database Management System)

Database Management System (DBMS) **គឺជាកម្មវិធីដែលត្រូវបានប្រើប្រាស់សម្រាប់ បង្កើត គ្រប់គ្រង និងរក្សាទុក** Database **ព្រមទាំងផ្ដល់នូវការត្រួតពិនិត្យទៅលើដំណើរការរបស់** Database**។** DBMS **មានជាច្រើនប្រភេទដូចជា** Microsoft Access, Microsoft SQL Server, MongoDB, Visual Forpro, MySQL **និង** Oracle **(ប៊ូឈុន, ២០១០)។**

អត្ថប្រយោជន៍នៃប្រព័ន្ធគ្រប់គ្រងទិន្នន័យ DBMS៖

* **អាចគ្រប់គ្រងទិន្នន័យដ៏ច្រើន**
* **កំណត់សិទ្ធិក្នុងការ** Access **ចូលមើលទិន្នន័យ**
* **ផ្ដល់កន្លែងស្ថិតស្ថេរសម្រាប់** Program Objects **និង** Data Structures
* **ផ្ដល់នូវ** User Interfaces **ជាច្រើន**
* **អាចជំនួសឲ្យទំនាក់ទំនងដ៏ស្មុកស្មាញរវាងទិន្នន័យ**
* **អាចទប់ស្កាត់កុំឲ្យអ្នកប្រើប្រាស់** **បញ្ចូលទិន្នន័យដែលមិនត្រឹមត្រូវ** (Enforcing Integrity Constraints)
* **អាចឲ្យ** User **ធ្វើការ** Backup and Recovery (Elmasri & Navathe, 2000)**។**

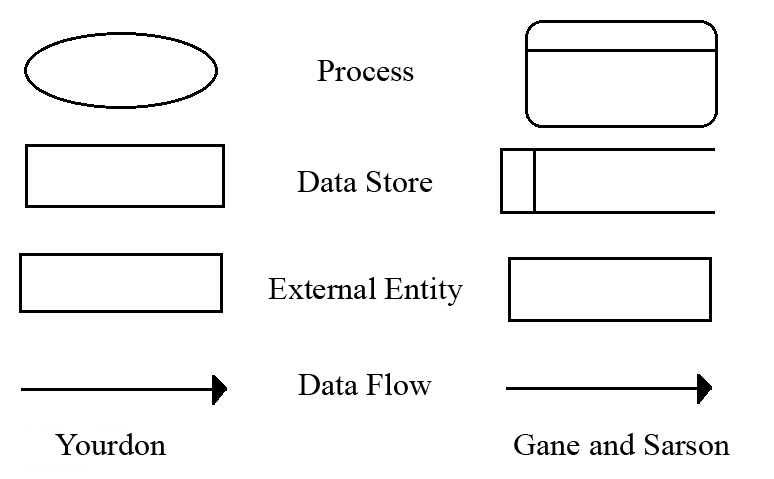
## ២.៣. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) បង្ហាញពីទិន្នន័យធ្វើដំណើរការនៅក្នុងប្រព័ន្ធព័ត៌មាន ប៉ុន្តែមិនបានបង្ហាញពី Program Logic រឺជំហ៊ានរបស់ដំណើរការទេ។ DFD បង្ហាញពី Logical Model មានន័យថាវាបង្ហាញពីអ្វីដែលប្រព័ន្ធដំណើរការតែមិនបង្ហាញថាតើប្រព័ន្ធដំណើរការដោយរបៀបណាទេ។

### ២.៣.១. Data Flow Symbols

DFDS ប្រើប្រាស់និមិត្តសញ្ញាមូលដ្ឋានចំនួន ៤ ដើម្បីបង្ហាញពី Processes, Data Flows, Data Stores និង External Entities ។ មានប្រភេទរូបតំណាងជាច្រើនចំពោះ DFD Symbols ប៉ុន្តែពួកគេនៅតែថែរក្សាអត្ថន័យដូចគ្នា ។

* Processes ទទួល Input Data បង្កើត Output ដែល Content, Form រឺ ទាំងពីរមានលក្ខណៈខុសពីមុន។ ហើយនិម្មិតសញ្ញារបស់ Process គឺរង្វង់មូល ហើយឈ្មោះ Process ស្ថិតនៅក្នុងរង្វង់មូលនោះតែម្តង។ ឈ្មោះ Process សម្រាប់កំណត់ Function ជាក់លាក់ណាមួយ និងផ្ទុកនូវកិរិយាស័ព្ទ (និងគុណនាម ប្រសិនបើចាំបាច់) តាមក្រោយដោយនាម ។ ឧទាហរណ៍ៈ ឈ្មោះ Process ដូចជា Calculate Commission, Verify order and Fill Order ។
* Data Flow គឺជាផ្លូវទិន្នន័យដែលផ្លាស់ទីពីផ្នែកណាមួយនៃប្រព័ន្ធព័ត៌មានទៅកាន់ផ្នែកមួយផ្សេងទៀត ហើយ Data Flow នៅក្នុង DFD បង្ហាញពី Data Item មួយរឺច្រើន។ ឧទាហរណ៍ៈ Data Flow​ បង្ហាញពី Single Data Item ដូចជា Student ID number រឺក៏បង្ហាញពីសំនុំទិន្នន័យដូចជាៈ Student ID number, Name and Registration date ចំពោះការចុះឈ្មោះចូលរៀន។ និម្មិតសញ្ញាសម្រាប់ Data Flow គឺបន្ទាត់ដែលមានចុងព្រួញមួយ រឺពីរ ហើយឈ្មោះData Flow បង្ហាញនៅខាងលើ​ រឺខាងក្រោមបន្ទាត់។​ ឧទាហរណ៍ៈ ឈ្មោះ Data Flow ដូចជាៈ Deposit, Order, Student Grade and Commission ។
* Data Store or Data Repository ប្រើប្រាស់ក្នុង DFDសម្រាប់បង្ហាញពីស្ថានការណ៍ដែលប្រព័ន្ធដែលប្រព័ន្ធត្រូវរក្សាទុកទិន្នន័យនោះ ព្រោះ Process មួយរឺច្រើន ផ្សេងទៀតត្រូវការប្រើប្រាស់វានៅពេលក្រោយ​​ ហើយចំពោះរយៈរក្សាទុកទិន្នន័យមិនសំខាន់តែអាច១នាទី, ១ម៉ោង រឺ ១ខែ ជាដើម។ ឧទាហរណ៍ៈ គ្រូត្រូវរក្សាទុកពិន្ទុ Assignmentសិស្ស ដូចនេះនៅពេលប្រលងឆមាសគេអាច គណនាពិន្ទុលទ្ធផលបាន។ និម្មិតសញ្ញាសម្រាប់ Data Store គឺជាចតុកោណកែងដែលចំហរនៅផ្នែកខាងឆ្វេង និងខាងស្តាំហើយឈ្មោះ Data Storeបង្ហាញនៅចន្លោះចតុកោណកែងនោះ។ឧទាហរណ៍៖ ឈ្មោះ Data Store​ ដូចជា students, Products, Employees and Daily payments។
* External Entity គឺជាមនុស្ស, ​Department, Outside Organization និងប្រព័ន្ធព័ត៌មានផ្សេងទៀតដែលផ្តល់ទិន្នន័យទៅកាន់ប្រព័ន្ធ​រឺទទួលលទ្ធផលពីប្រព័ន្ធ។ ឧទាហរណ៍ៈ Customer បញ្ចូល Order​ គឺជា External Entity ពីព្រោះ Customer ផ្តល់ទិន្នន័យទៅកាន់ប្រព័ន្ធ។ និម្មិតសញ្ញាសម្រាប់ External Entity គឺជាចតុកោណកែង ហើយឈ្មោះ External Entity បង្ហាញនៅចន្លោះចតុកោណកែងនោះ។​ ឧទាហរណ៍៖ ឈ្មោះ External Entity ​ដូចជា Customer, Student, Employee, Warehouse and Bank។

តួលេខ ២.៣ និមិត្តសញ្ញារបស់ DFD

### ២.៣.២. Context Diagram

យើងប្រើប្រាស់ Interviews, Questionnaires and Techniques ផ្សេងៗសម្រាប់ប្រមួល Facts ចំពោះប្រព័ន្ធ និងសិក្សាស្វែងយល់អំពីរបៀបទំនាក់ទំនងរវាងមនុស្សជាច្រើន, Departments, Data and Processes ក្នុងកំឡុងពេល Requirement Modeling រួចមកហើយឥឡូវយើងត្រៀមជាស្រេចដើម្បីបង្កើត Graphic Model សម្រាប់ប្រព័ន្ធព័ត៌មានដោយសំអាងទៅលើលទ្ធផលនៃ Fact-Finding។

ជំហ៊ានដំបូងក្នុងការកសាងសំនុំ DFD គឺយើងធ្វើការគូរ Context Diagrams (Context Diagram គឺជា Top-Level View នៃប្រព័ន្ធព័ត៌មានដែលបង្ហាញពីព្រំដែនរបស់ប្រព័ន្ធ)។ ដើម្បីគូរ Context Diagram យើងចាប់ផ្តើមគូរ Process តែមួយគត់នៅទីតាំងកណ្តាលទំព័រ ហើយនិមិត្តសញ្ញានេះបង្ហាញពីប្រព័ន្ធព័ត៌មានទាំងមូល និងយើងកំនត់វាជា Process លេខ 0 (សូន្យ) ។ បន្ទាប់មកដាក់ External Entities នៅជុំវិញបរិវេណទំព័រនោះ និងប្រើ Data Flow ដើម្បីភ្ជាប់ External Entities ទៅកាន់ Process។ យើងមិនបានបង្ហាញអំពី Data Store ណាមួយនៅក្នុង Context Diagram ទេពីព្រោះ Data Store វាស្ថិតនៅខាងក្នុងប្រព័ន្ធ។

រាល់ការគូរ DFD​​ ត្រូវគោរពទៅតាមគោលការណ៍ដូចខាងក្រោមៈ

* គ្រប់ Context Diagram ត្រូវស្ថិតនៅក្នុងមួយទំព័រ
* ឈ្មោះ Process នៅក្នុង Context Diagram លើសពីនេះទៅទៀត គួរតែជាឈ្មោះប្រព័ន្ធព័ត៌មាន
* ប្រើប្រាស់ឈ្មោះតែមួយ Unique Name ចំពោះនិម្មិតសញ្ញាទាំងអស់
* កុំគូរពីលើខ្សែរបន្ទាត់គ្នា Don’t Cross Line
* ប្រើប្រាស់ Unique Reference Number ចំពោះ Process Symbol

និមួយៗ។

### ២.៣.៣. Diagram 0

Context Diagram ផ្តល់នូវទស្សនៈទូទៅចំពោះប្រព័ន្ធព័ត៌មាន និងផ្ទុកនិម្មិតសញ្ញា Process តែមួយប៉ុណ្ណោះ ហេតុដូច្នេះដើម្បីបង្ហាញព័ត៌មានលំអិតខាងក្នុង Process នោះយើងត្រូវតែបង្កើត DFD diagram 0 ។ Diagram 0 (លេខសូន្យមិនមែនតួរអក្សរអូទេ) ពង្រីកContext Diagram អោយកាន់តែធំងាយស្រួលមើល និងបង្ហាញពី Process, Data Flows និង Data stores ចំបងៗហើយវាក៏បង្ហាញពី External Entities and Data Flows ម្តងទៀតផងដែរ។

នៅពេលដែលយើងពង្រីក Context Diagram ទៅកាន់ Diagram 0 យើងត្រូវតែរក្សារាល់ Connections ចេញ និងចូលពី Process ។

### ២.៣.៤. Lower-Level Diagram

Lower-Level Diagram (គេអាចហៅបានថា Child Diagram) ប្រើប្រាស់សម្រាប់បង្ហាញព័ត៌មានលំអិត ហើយ Diagrams ទាំងនោះត្រូវតែ Leveling and Balancing​។ Leveling គឺជាដំណើរការគូរស៊េរី Diagram កើនឡើងជាបន្តបន្ទាប់រហូតទាល់តែដល់កំរិតព័ត៌មានលំអិតចង់បានហើយ Balancing ថែរក្សា Consistency ក្នុងចំណោម Diagrams ទាំងអស់នោះដោយរាប់បញ្ចូលទាំង Input and Output data flows, Data definition and Process definitions​។

### ២.៣.៥. Strategies for Developing DFDs

យុទ្ធសាស្រ្តសម្រាប់បង្កើត DFDs មានចំនួន​ ២ គឺ Top-Down model and Button-Up model។ Top-Down model គឺជា Model មួយដែលដំណើរការពីលើចុះមកក្រោមដោយជាដំបូងអ្នកវិភាគត្រូវបង្កើត Context Diagram ជាមុនសិនបន្ទាប់មក Diagram 0 និង Child Diagrams សម្រាប់ Diagram 0​ ។ល។​ទោះបីជាយ៉ាងណាក៏ដោយ មានយុទ្ធសាស្ត្រមួយទៀតផ្ទុយពី Top-Down model គឺ Button-Up model ដែលដំណើរការពីក្រោមឡើងលើដោយជាដំបូងអ្នកវិភាគត្រូវកំនត់ Function Primitives (Function Primitives គឺជា Process ដែលមាន Function តែមួយគត់ ហើយមិនអាចបំបែកតទៅទៀត), Data Stores, External Entities and Data Flows ជាមុនសិន បន្ទាប់មកផ្តុំ Process ជាមួយនិមិ្មតសញ្ញាពាក់ព័ន្ធផ្សេងទៀតដើម្បីបង្កើត Lowest-Level Diagram។ បន្ទាប់មកទៀតផ្តុំ Diagrams នោះដើម្បីបង្កើត Higher-Level Diagram រហូតទាល់តែដល់ Context Diagram។

### ២.៣.៦. Data Dictionary

សំនុំ DFDs បានបង្កើត Logical Model សម្រាប់ប្រព័ន្ធប៉ុន្តែព័ត៌មានលំអិតក្នុង DFDs ត្រូវបានកត់ត្រាដាច់ដោយឡេកពី DFDs ទៅកាន់ Data Dictionaryដែលគឺជាសមាសភាពទី២ នៃ Structured Analysis។

Data Dictionary or Data Repository​ គឺជាកន្លែងកណ្តាល (Central Storehouse) សម្រាប់រក្សាព័ត៌មានអំពីទិន្នន័យប្រព័ន្ធ។ អ្នកវិភាគប្រើប្រាស់ Data Dictionary ​ដើម្បីប្រមូល, កត់ត្រាទុក និងរៀបចំ Facts ជាក់លាក់អំពីប្រព័ន្ធដោយរួមបញ្ចូលទាំង Contents នៃ Data Flows, Data Store, External Entities and Process ហើយ Data Dictionary ក៏កំនត់ និងរៀបរាប់រាល់ពេល Data Element ទាំងអស់ព្រមទាំងការរួមបញ្ចុលគ្នារវាង Data Element ដែលមានន័យ។ Data Elements or Data Item or Filed គឺជាបំណេកតូចបំផុតនៃទិន្នន័យដែលមានអត្ថន័យនៅក្នុងប្រព័ន្ធព័ត៌មាន។ ឧទាហរណ៍ៈ Data Elements មានដូចជា Student Grade, Address, Salary, Social Security number and Province។ Data Elements ជាច្រើនរួមបញ្ចូលគ្នាបង្កើតចេញជា Data Structure or Record ហេតុដូចនេះ Record គឺជាការច្របាច់បញ្ចូលគ្នាមានន័យរវាង Data Element ពាក់ព័ន្ធ។ ឧទាហរណ៍ៈ​ Student Record ​មាន Data Elements ដូចជា Student ID Number, Name, Sex, Date of Birth and Address ជាដើម។

## ២.៤ Data Design Terminology

អ្នកវិភាគប្រព័ន្ធ ត្រូវជ្រើសរើសវិធីសាស្ត្រមួយ ដើម្បីចាប់ផ្តើមបង្កើត Data Management System ដោយប្រើប្រាស់ Data Design Concepts ​ហើយជំហ៊ានដំបូងត្រូវស្វែងយល់ពី Data Design Terminology ជាមុនសិន។

### ២.៤.១. Definitions

Entity: សំដៅទៅលើមនុស្ស, ទីកន្លែង, វត្ថុ រឺព្រឹត្តិការណ៍សម្រាប់ប្រមូល និងរក្សាទុកទិន្នន័យ។ ឧទាហរណ៍ៈ Sales System ផ្ទុក Entities មានឈ្មោះថា Customer, Product, Order, and Supplier។ នៅពេលយើងរៀបចំបង្កើត DFD​ កំឡុងពេល Systems Analysis Phase យើងបានធ្វើការកំនត់ Entities and Data Stores ផ្សេងៗរួចហើយ ក្នុងពេលនេះយើងនិងពិចារណាទៅលើទំនាក់ទំនង (Relationship) រវាង Entities ទាំងនោះ។

Field: Field អាចហៅថា Attribute ដែលគឺជាចរិកលក្ខណៈរឺ ហេតុការណ៍តែមួយ (Single Characteristic or Fact) នៃ Entity។ នៅក្នុងឧទាហរណ៍ខាងលើ Customer នីមួយៗដូចជា Customer មាន Fields ជាច្រើនប្រើសម្រាប់រក្សាទុកទិន្នន័យចំពោះ Customer នីមួយៗដូចជា Customer’s Name, Sex and Phone។ Common Field គឺជា Attribute ដែលស្ថិតក្នុង Entity លើសពីមួយហើយជាទូទៅ Common Field ត្រូវបានគេប្រើប្រាស់ដើម្បី Entities មានទំនាក់ទំនងជាមួយគ្នា។

Record: Record អាចហៅថា Tuple គឺជាសំនុំនៃ Fields មានទំនាក់ទំនងគ្នាប្រើសម្រាប់រៀបរាប់ពី One Instance ឬសមាជិក Entity ដូចជា One Customer, One Order or One Product។ Record មួយអាចមាន Field​ មួយឬច្រើនអាស្រ័យទៅតាមតំរូវការព័ត៌មានចង់បាន។

នៅក្នុង File-Oriented System សំនុំ Records ទាក់ទងគ្នាជាច្រើនផ្តុំចូលគ្នាបង្កើតបានជា File​ សម្រាប់ផ្ទុកទិន្នន័យចំពោះមនុស្ស, ទីកន្លែង,​ វត្ថុ រឺព្រឹត្តិការណ៍ដូចទៅនិងអ្វីដែលយើងបានសិក្សា។ ឧទាហរណ៍៖ ប្រសិនបើ Inventory System គ្រប់គ្រង 500 Products ដូច្នេះ គេនិយាយថា Product File អាចត្រូវបានគេរៀបចំអោយមានរចនាសម្ត័ន្ធក្នុងទំរង់មួយដើម្បីល្បឿនដំណើរការនៅក្នុង File-Oriented System ។

នៅក្នុង Database Environment សំនុំ Records ទាក់ទងគ្នាជាច្រើនផ្តុំចូលគ្នាបង្កើតជា Table សម្រាប់រក្សាទុកទិន្នន័យចំពោះ​ Entity ជាក់លាក់ណាមួយ។ ជាទូទៅ Table បង្ហាញក្នុងទំរងជា Two-Dimensional Structures ដែលផ្ទុកនូវ Vertical Colum បង្ហាញពី Fields និង Horizontal Rows បង្ហាញពី Records។ ឧទាហរណ៍ៈ Sale database System ប្រហែលជាមាន Tables ច្រើនដូចជាៈ Customer, Order, Product and Supplier។

### ២.៤.២. Key Fields

ក្នុងកំឡុងពេល System Design Phase យើងប្រើប្រាស់ Key Field ដើម្បីរៀបចំប្រើប្រាស់ (access) និងថែរក្សា Data Structure​។ Keys ចែកចេញជា ៣ ប្រភេទគឺ Primary Key, Candidate Keys and Foreign Keys។

Primary Keys: គឺជា File មួយរឺបណ្តុំ Fields ដែលធ្វើការកំណត់អត្តសញ្ញាណតែឯកឯង និងតូចបំផុតចំពោះសមាជិក Entity។ ឧទាហរណ៍៖ នៅក្នុង Customer Table មាន Field Customer Number គឺជា Primary Key ពីព្រោះគ្មាន Customers ណាដែលមាន Customer Number ដូចគ្នាទេ។ Primary Key ត្រូវតែផ្ទុកព័ត៌មានតិចបំផុតដោយជ្រើសរើស Field ណាដែលគេត្រូវការកដើម្បីកំនត់អត្តសញ្ញាណតែប៉ុណ្ណោះ។

Primary Keys អាចកើតពីការផ្សំចូលគ្នារវាង Fields ចាប់ពី២ឡើងទៅ។ ឧទាហរណ៍ៈ ប្រសិនបើសិស្សចុះឈ្មោះវគ្គសិក្សាចំនួន ៤នាក់នោះ Student Number និងបង្ហាញដល់ទៅ ៤ Records នៅក្នុង Registration System ។ បើសិស្សចុះឈ្មោះវគ្គសិក្សាចំនួន ២០នាក់នោះ ២០ Separate Records និងលេចឡើងចំពោះ Course Number នោះជាមិនខានដោយ Record មួយសម្រាប់សិស្សនីមួយៗដែលបានចុះឈ្មោះ។

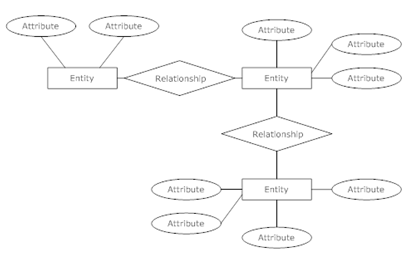
Candidate Key: នៅក្នុងករណីមួយចំនួនយើងអាចជ្រើសរើសយក Field មួយរឺបណ្តុំ Fields ជាច្រើនដើម្បីកំនត់ជា Primary Key។ Fields​ ទាំងឡាយណាដែលអាចដើរតួនាទីជា Primary Key គេហៅថា Candidate Key។ ឧទាហរណ៍៖ ប្រសិនបើ Employee នីមួយៗមាន Unique Employee Number ដូច្នេះយើងអាចកំនត់ Employee Number or Social Security Number ធ្វើជា Primary Key។ ដោយសារតែនៅក្នុង Table មួយមាន Primary Key តែមួយគត់ដូច្នេះយើងត្រូវជ្រើសរើស Field ណាដែលផ្ទុកទិន្នន័យតិចបំផុត និងងាយស្រួលប្រើប្រាស់លើសពីនេះទៀត Field ណាដែលផ្ទុកទិន្នន័យតិចបំផុត និងងាយស្រួលប្រើប្រាស់លើសពីនេះទៀត Field ណាមិនត្រូវបានជ្រើសរើសជា Primary Key គេហៅថា Nonkey Field។

Foreign Key: គូរកត់សំគាល់ថា Common Field លេចឡើងនៅក្នុង Table ជាច្រើនជាមួយ និងត្រូវបានគេប្រើប្រាស់ដើម្បីបង្កើតទំនាក់ទំនង (Relationship or Link) រវាង Tables ទាំងនោះ។ Foreign Key គឺជា Field ស្ថិតនៅក្នុង Table មួយដែលត្រូវផ្គូរផ្គងជាមួយតំលៃ Primary Key នៅក្នុង Table ​មួយផ្សេងទៀតដើម្បីបង្កើតទំនាក់ទំនងរវាង Table ទាំងពីរ។ Foreign Key មិនចាំបាច់មានឯកឯង (Unique) ដូច Primary key ទេ។

### ២.៤.៣. Data Relationships

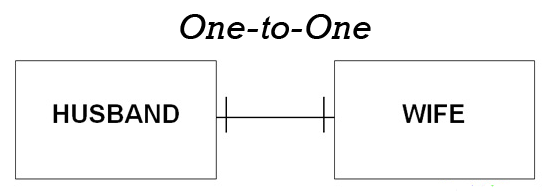
Relationship គឺជាការតភ្ជាប់ (Logical Link) រវាង Entities។ ឧទាហរណ៍ៈ Relationship កើតមានរវាង Entities Product and Warehouse ពីព្រោះ Products រក្សាទុកនៅក្នុងWarehouse។

Entity-Relationship Diagrams (ERD) គឺជាការបង្ហាញជាលក្ខណៈតួលេខចំពោះប្រព័ន្ធព័ត៌មានដោយបង្ហាញពីទំនាក់ទំនងរវាង Entities ទាំងនោះ។ រាល់ Entity ទាំងអស់បង្ហាញជារាងចតុកោណកែង (Rectangle) ព្រមទាំងឈ្មោះជា Singular Noun and Diamond បង្ហាញពី Relationship ព្រមទាំងឈ្មោះបង្ហាញជា Active verb។ ឧទាហរណ៍ៈ រូបខាងក្រោមបង្ហាញពីវេជ្ជបណ្ខិត ប៉ុន្តែជារឿយៗគេច្រើនតែប្រើប្រាស់ Active verb ពីព្រោះវាមានលក្ខណៈស្តង់ដារ។ Entity-Relationship Diagram មិនបានពណ៌នាពី Data or Information Flow ទេហើយ ERD​ គ្មានប្រើសញ្ញាក្បាលព្រួញទេលើសពីនេះទៀត យើងអាចរៀបចំទីតាំង Entities នៅខាងលើ រឺខាងឆ្វេងតាមបំណងរបស់យើងដោយលៃយ៉ាងណាអោយងាយមើល។​

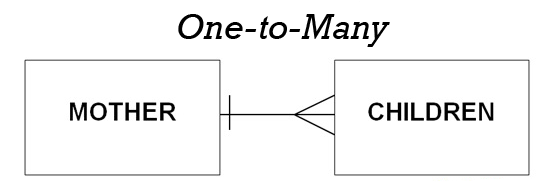
តួលេខ ២.៤ Entity Relationship Diagram

Relationship Diagram ចែកជា​៣ ប្រភេទចំបងគឺៈ

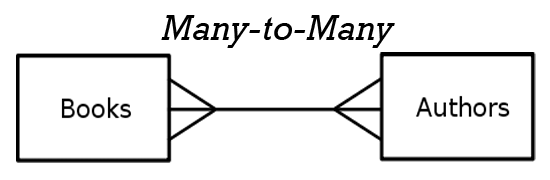
* One-to-one Relationship: សរសេរអក្សរកាត់ជា 1:1 ដែលកើតឡើងនៅពេល Record នីមួយៗនៅក្នុង Entity ទី១ទាក់ទងជាមួយ Record តែមួយគត់នៅក្នុង Entity ទី២​ ហើយ Record នីមួយៗនៅក្នុង Entity ទី២ទាក់ទងជាមួយ Record តែមួយគត់នៅក្នុង Entity ទី១។ DFDs ចំពោះ 1:1 Entity Relationship បង្ហាញនៅរួបរាងខាងក្រោម។ តំលៃលេខ១ គេដាក់ក្បែរបន្ទាត់ទាំងពីរដែលតភ្ជាប់ពី Rectangle ទៅកាន់ Diamond ដើម្បីបញ្ជាក់ 1:1 Relationship។

តួលេខ ២.៥ One to One Relationship

* One-to-Many Relationship: សរសេរអក្សរជា 1:M ដែលកើតឡើងនៅពេល Record មួយនៅក្នុង Entity ទី១ទាក់ទង Record ជាច្រើននៅក្នុង Entity ទី២ហើយ Record នីមួយៗនៅក្នុង Entity ទី២ទាក់ទងជាមួយ Record តែមួយគត់នៅក្នុង Entity​ ទី១។

តួលេខ ២.៦ One to Many Relationship

* Many-to-Many Relationship: សរសេរអក្សរកាត់ជា M:N ដែលកើតឡើងនៅពេលRecord មួយនៅក្នុង Entity​ ទី១ទាក់ទង Record ជាច្រើននៅក្នុង Entity ទី២ ហើយ Record មួយនៅក្នុង Entity ទី២ ទាក់ទងជាមួយ Record ជាច្រើននៅ Entity ក្នុងទី១។
* M:N Relationship ​មានលក្ខណៈខុសប្លែកពី 1:1 or 1:M Relationship ពីព្រោះព្រឹត្តិការណ៍សម្រាប់ភ្ជាប់ Entities ទាំងពីរនេះជារឿយៗក្លាយទៅជា Entity ទី៣ ដែលគេអោយឈ្មោះថា Associative Entity ដែលមានសំនុំ Attributes ផ្ទាល់ខ្លួនរបស់វា។ នៅរូបខាងក្រោមយើងសង្កេតឃើយថា Enroll In Relationship បង្ហាញពី Relationship Entity ដែលកត់ត្រារាល់ Instances ចំពោះ Student ជាក់លាក់ណាមួយចុះឈ្មោះចូលសិក្សា Course ណាមួយ។ ដូចគ្នាផងដែរ Reserves Seat on Relationship បង្ហាញពី Reservation Entity ដែលកត់ត្រារាល់ Instances ចំពោះ Passenger ជាក់លាក់ណាមួយ។

តួលេខ ២.៧ Many to Many Relationship

ERD ពេញលេញនៅក្នុងរួបរាងក្រោមបង្ហាញរាល់ Entities and Relationships ទាំងអស់កើតមានក្នុងប្រព័ន្ធដោយមាន Entities ចំនួន៥ គឺ SALES REP, CUSTOMER, ORDER. PRODUCT and WAREHOUSE ព្រមទាំង Relationships រវាង SALES REP និង PRODUCT គឺជា One-to-Many Relationship។ ទោះបីជាយ៉ាងណាក៏ដោយនៅក្នុងក្រុមហ៊ុនមួយចំនួនមាន SALES REP ជាច្រើនបំរើចំពោះ CUSTOMER តែម្នាក់ដូច្នេះនៅក្នុងករណីនេះយើងត្រូវបំបែកវាជា Many-to-Many Relationship។

* Normalization: យោងទៅតាមសៀវភៅ Database System ដោយដកស្រងអំពីទ្រឹស្តី Normalization ដែលជាទ្រឹស្តីសម្រាប់បំលែង Table អោយស្ថិតនៅលក្ខណៈធម្មតា។ ក្នុងនោះមានបីដំណាក់កាលក្នុងការបំលែង Table​ ស្ថិតក្នុងលក្ខណៈធម្មតា គឺ First Normalization, Second Normalization និង Third Normalization ដើម្បីធានាឱ្យបាននូវការមិនបាត់បង់ទិន្នន័យនិង ជំរុះឱ្យអស់នូវ Redundancy Data។
* First Normal Form: Entity មួយជា First Normal Form លុះត្រាតែមិនទាន់មាន Attribute ដែលមានភាពច្រំដែលសម្រាប់ Entity មួយ។
* Second Normal Form: Entity មួយជា Second Normal Form បើសិនជាគ្រប់ Non-Key attribute គឺពឹងផ្អែកលើ Full Key Attributeជាមួយ Normal Form ដែលនៅក្នុងFirst Normal Form យើងបានដកចេញគ្រប់ Relationship group of attribute។
* Third Normal Form: Entity មួយជា Third Normal Form បើសិនជា Entity នោះត្រូវបាន Normalized រូចនៅក្នុង Second Normal Form ហើយគ្រប់ Non-Key Columns ដែលពឹងផ្អែកជាមួយ non-key Columns ដ៏ទៃទៀងបានផ្លាស់ប្តូរ។

Normalization គឺជាញឹកញាប់ត្រូវបានអនុវត្តន៍នៅក្នុងជំហ៊ាននីមួយ ដែលត្រូវគ្នាទៅនិង Normal Form។ Normal Form គឺជាសមាសភាពទំនាក់ទំនង ដែលអាចកំណត់បានដោយដាក់ក្នុងទំរង់ធម្មតាដែល ចាត់ទុកថាអាស្រ័យទៅនិងគ្នាទៅវិញទៅមក រវាង Attribute ដែលជាប់ទំនាក់ទំនងគ្នា។ ជំហ៊ាននៃ Normalization ក្នុងការបំលែង Table ដែលមាន Repeating Group។

## ២.៥. MongoDB

MongoDB គឺជា cross-platform document-oriented database។ ហើយវាត្រូវបានគេស្គាល់ថាជាប្រភេទ NoSQL database, MongoDB បានជៀសវាងការប្រើប្រាស់ traditional table-based relational database structure ក្នុង favor នៃ JSON-like documents ជាមួយនឹង dynamic schemas (MongoDB មាន format ជា BSON) ធ្វើឱ្យការរួមបញ្ចូលនៃ data ក្នុង certain types នៃ applications មានភាពងាយស្រួល និងលឿនជាងមុន។ MongoDB អាចគាំទ្រការ search by field, range queries, regular expression searches. Queries can return specific field នៃ documents និងអាចបញ្ចូល user-defined JavaScript functions។

MongoDB បានចេញផ្សាយក្រោមការរួមបញ្ចូលគ្នានៃអាជ្ញាប័ណ្ណសាធារណៈរបស់ GNU ទូទៅ និងការ Affero និងអាជ្ញាប័ណ្ណរបស់ Apache ហើយ MongoDB គឺជាកម្មវិធីគិតថ្លៃនិងជា open-source software ។ វាត្រូវបានបង្កើតឡើងជាលើកដំបូងដោយក្រុមហ៊ុន Software ឈ្មោះថា MongoDB Inc នៅខែតុលាឆ្នាំ ២០០៧ ទុកជា component នៃ planned platform ជាមួយ service product ហើយក្រុមហ៊ុននេះបានផ្លាស់ប្តូរទៅជា open source development model នៅក្នុងឆ្នាំ ២០០៩ ដោយមានការគាំទ្រពី MongoDB ទាំងពាណិជ្ជកម្ម និងសេវាកម្មផ្សេងទៀត។ ចាប់តាំងពីពេលនោះមក MongoDB ត្រូវបានអនុម័តថាជា backend software ដោយគេហទំព័រ និងសេវាកម្ម major ដូចជា Craigslist, eBay, និង Foursquare ដទៃទៀត។ ក្នុងខែកក្កដាឆ្នាំ ២០១៥ MongoDB គឺជាប្រភេទ database management system ដែលមានប្រជាប្រិយភាពបំផុតទីបួន និងជា document stores ដែលពេញនិយមបំផុត។

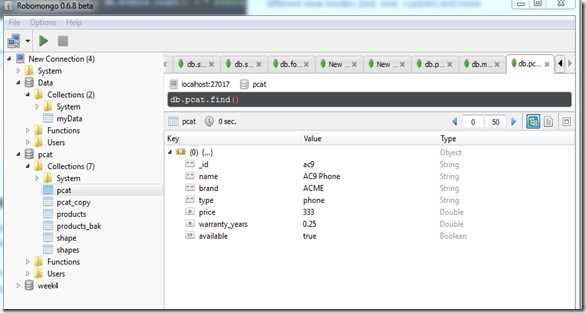
|  |  |
| --- | --- |
| [MongoDB Logo.png](https://en.wikipedia.org/wiki/File:MongoDB_Logo.png) | |
| [**Developer(s)**](https://en.wikipedia.org/wiki/Software_developer)**:** | [MongoDB Inc.](https://en.wikipedia.org/wiki/MongoDB_Inc.) |
| **Initial release:** | 2009 |
| [**Stable release**](https://en.wikipedia.org/wiki/Software_release_life_cycle)**:** | 3.0.5 / 28 July 2015 |
| **Development status:** | Active |
| **Written in:** | C++, JavaScript, [C](https://en.wikipedia.org/wiki/C_(programming_language)) |
| [**Operating system**](https://en.wikipedia.org/wiki/Operating_system)**:** | Cross-platform |
| **Available in:** | English |
| [**Type**](https://en.wikipedia.org/wiki/List_of_software_categories)**:** | [Document-Oriented Database](https://en.wikipedia.org/wiki/Document-oriented_database) |
| [**License**](https://en.wikipedia.org/wiki/Software_license)**:** | GNU AGPL v3.0 (drivers: Apache license) |
| **Website:** | [www.mongodb.org](https://www.mongodb.org/) |

* ភាសាកូដដែលគាំទ្រ

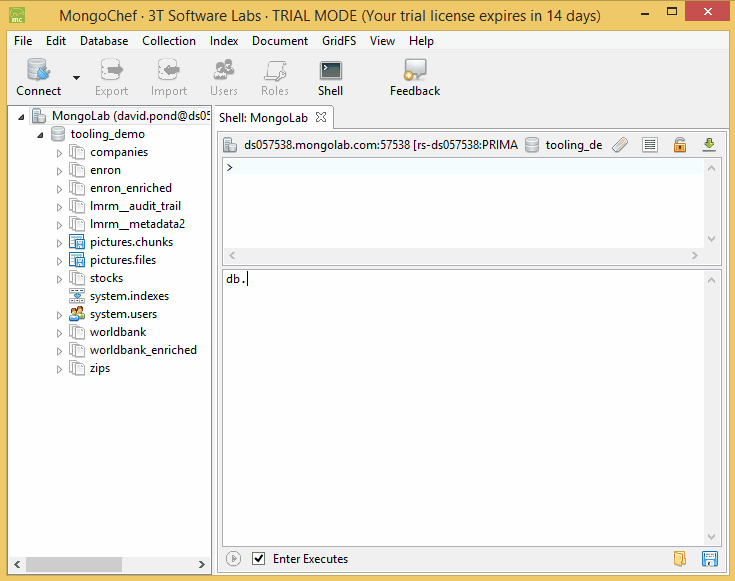
MongoDB មាន official driver សម្រាប់ភាសាកូដជាច្រើនដែលពេញនិយម ហើយវាក៏មាន unofficial ឬ community-supported drivers សម្រាប់ភាសាកូដ និង frameworks ដ៏ទៃទៀតដែរ។

* Management និង graphical front-ends

រាល់ Record ផ្សេងៗនៅក្នុង MongoDB ភាគច្រើនត្រូវបានបញ្ចូលដោយប្រើប្រាស់ command line tools ដូចជា mongo shell ជាដើមព្រោះ MongoDB មិនមាន GUI-style administrative interface ទេតែយើងអាចជំនួសដោយ third-party software មានដូចជា Robomongo, Mongochef ផងដែរ។

រូបភាពទី ២.១ ផ្ទាំងកម្មវិធី Robomongo

រូបភាពទី ២.២ ផ្ទាំងកម្មវិធី Mongochef



* ការផ្តល់អាជ្ញាប័ណ្ណ និងការគាំទ្រ

MongoDB ដាក់ឲ្យប្រើប្រាស់ឥតគិតថ្លៃស្ថិតនៅក្រោម GNU Affero General Public License ហើយ language drivers ស្ថិតនៅក្រោម ​ Apache License។

## ២.៤. Meteor JS

Meteor ឬ Meteor JS គឺជា open-source JavaScript web application framework ដែលសរសេរដោយប្រើប្រាស់ Node.js។ Meteor អនុញ្ញាតអោយប្រើប្រាស់ជា cross-platform (Web​ Browser, Android, iOS)។ វាប្រើប្រាស់រួមបញ្ចូលជាមួយ MongoDB និងប្រើ Distributed Data Protocol និង publish–subscribe pattern ដើម្បីបង្កើតការផ្លាស់ប្តូរទិន្នន័យដោយស្វ័យប្រវត្តិទៅកាន់ clients ដោយមិនចាំបាច់ទាមទារអោយ developer សរសេរកូដដើម្បី synchronization (មិនចាំបាច់ Reload Page)។ នៅលើ client Meteor អាស្រ័យទៅលើ jQuery និងអាចត្រូវបានប្រើជាមួយ JavaScript UI widget library ផងដែរ។

ប្រវត្តិនៃ Meteor JS

Meteor ត្រូវបានដាក់ឲ្យប្រើប្រាស់ដំបូងក្នុងខែ ធ្នូ ឆ្នាំ ២០១១ ក្រោមឈ្មោះថា Skybreak ។​ ក្នុងខែតុលា ឆ្នាំ ២០១៤, Meteor Development Group acquired Y Combinator alum FathomDB មានគោលដៅពង្រីកការគាំទ្រមូលដ្ឋាននៃ Meteor's database ។

Meteor JS Command Line Tool

* meteor help

សម្រាប់បង្ហាញរាល់ meteor  commands ទូទៅទាំងអស់។ Running meteor help <command> វានឹងបង្ហាញជំនួយ detail អំពី meteor <command> នោះ។

* meteor create <project name>

សម្រាប់បង្កើត Meteor project ថ្មី (ត្រូវទាញ file ពី internet)​។

* meteor ឬ meteor run

សម្រាប់ run current project ។ ចូលទៅកាន់ directory របស់ project រួច Run command meteor ហើយវាយ [http://localhost:3000](http://localhost:3000/)​ នៅលើ Browser ។

* meteor debug

សម្រាប់ Run meteor project ជាមួយនឹង Node Inspector attached ប៉ុន្តែផ្អាកដំណើរការរបស់ server ដើម្បី debugging។ ដូច្នេះយើងអាចមើលនិង debugging server code ពីមួយជួរទៅមួយជួរ។

* meteor deploy <site>

សម្រាប់ Bundle project  របស់យើង និង deploy វាទៅកាន់ <site>។ Meteor បានផ្តល់ការ hosting ដោយឥតគិតថ្លៃប្រសិនបើយើង deploy ទៅកាន់ <your app>.meteor.com ដោយឈ្មោះ <your app> ដែលដាក់នោះ មិនត្រូវបានប្រើប្រាស់ដោយនរណាម្នាក់ផ្សេងទៀត។

* meteor update

សម្រាប់ update Meteor installation របស់យើងទៅកាន់ latest released version តែបើយើងប្រើ command នេះនៅក្នុង directory របស់ project នោះវានឹង update រាល់ packages ដែល compatible មានប្រើនៅក្នុង project របស់យើងទៅកាន់ latest versions។

* meteor add <package name>

សម្រាប់ Add package មួយឬច្រើន ចូលទៅកាន់ Meteor project របស់យើង. ដើម្បីឆែកមើល available packages ត្រូវប្រើ command meteor search។

* meteor remove <package name>

សម្រាប់ Remove package ចាស់ៗដែលបាន added ចូល Meteor project របស់យើង. ដើម្បីឆែកមើល list របស់ packages ដែល project កំពុងប្រើប្រាស់ត្រូវប្រើ command​ meteor list ។

* meteor mongo

សម្រាប់បើក MongoDB shell ដើម្បី viewing ឬក៏ manipulating collections stored នៅក្នុង database។ សម្គាល់៖ ត្រូវប្រាកដថាយើង​បាន run server របស់ current project (នៅក្នុង terminal window​ ដ៏ទៃ) ដើម្បីអាចឲ្យ meteor mongo  connect ទៅកាន់ project database ។

* meteor reset

ជា command ប្រើសម្រាប់ Reset project របស់យើងទៅកាន់ fresh state ហើយវាលុបទិន្នន័យទាំងអស់។

## ២.៦. វេបសាយ (Website)

Website **គឺជាការភ្ជាប់ចូលគ្នារវាង** webpages **ជាច្រើនដែលផ្ទុកនូវ អត្ថបទ រូបភាព វីដេអូ និងឯកសារផ្សេងៗទៀត** (Wikipedia[a], 2012)**។** Website បានផ្ដល់អត្តប្រយោជន៍យ៉ាងច្រើនដល់យើង ដូចជា ធ្វើឲ្យងាយស្រួលក្នុងការស្រាវជ្រាវស្វែងរកទិន្នន័យ និងឯកសារផ្សេងៗ យ៉ាងឆាប់រហ័ស មិនថាអ្នកណា នៅទីកន្លែងណា នៅពេលណាក៏បាន សុទ្ធតែអាចចូលមើល Website បាន (Drug Information Enquiry Group21, 2003)។

### ២.៦.១ Static Website

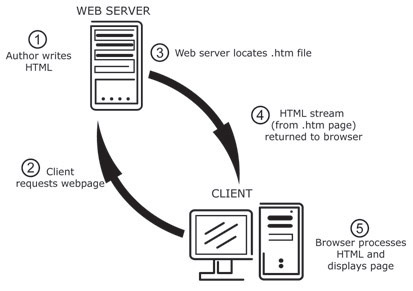
Static website គឺជា Website ដែលផ្ទុកទៅដោយ Web pages ជាច្រើនហើយ Web pages ទាំងនោះជាប្រភេទ fixed content ដោយ page នីមួយៗគឺសរសេរ code ដោយប្រើប្រាស់ HTML ហើយវាបង្ហាញព័ត៌មានដូចគ្នាទៅកាន់អ្នកទស្សនាទាំងអស់។ Static sites គឺជាប្រភេទនៃ website ដែលងាយស្រួលក្នុងការបង្កើត ដោយគ្រាន់តែសរសេរកូដ HTML ប៉ុន្មានទំព័ររួច publishing ទៅកាន់ Web server។

គុណសម្បត្តិនៃ Static Websites

* ការបង្កើតមានភាពងាយស្រួល និងឆាប់រហ័ស
* **ដំណើរការលឿនក្នុងការ Loading pages**
* **មិនចាំបាច់ត្រូវការ hosting server ពិសេស**

គុណវិបត្តិនៃ Static Websites

* **ពិបាកក្នុងការកែប្រែទិន្នន័យ និងថែទាំ**
* តម្រូវឱ្យមានអ្នកជំនាញ**ក្នុងការកែប្រែទិន្នន័យ**
* Website មិនសូវជាមានទិន្នន័យមានប្រយោជន៍សម្រាប់អ្នកប្រើប្រាស់
* **មាន**មុខងារមានប្រយោជន៍តិចតួច ឬក៏អាចគ្មាន

រូបភាពទី ២.៣ ដំណើរការរបស់ Static Websites

### ២.៦.២ Dynamic Website

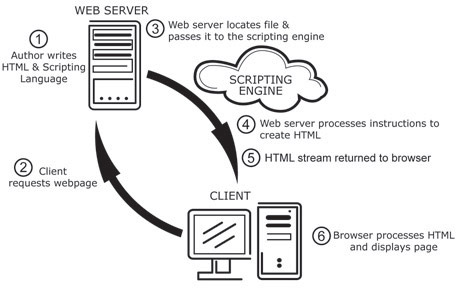
Dynamic Website គឺជាប្រភេទ Website ដែលមានសមត្ថភាពក្នុងការដាក់ content ផ្សេងពីគ្នាសម្រាប់អ្នកទស្សនាផ្សេងពីគ្នាដោយចេញពីប្រភពកូដតែមួយដោយប្រើប្រាស់កូដដូចជា PHP, ASP.Net, C# ជាដើម។ វាអាចបង្ហាញ content ផ្សេងពីគ្នាអាស្រ័យទៅតាម operating system ឬ browser ដែល visitor បានប្រើប្រាស់។ Dynamic Website មិនចាំបាច់ត្រូវតែល្អជាង static Website ទេគឺវាគ្រាន់តែបម្រើទៅតាមគោលបំណងផ្សេងគ្នាប៉ុណ្ណោះ។

គុណសម្បត្តិនៃ Dynamic websites

* មាន function ច្រើនក្នុងការ Develop
* **មាន user interface សម្រាប់ update ទិន្នន័យ**
* ងាយស្រួលក្នុងការ **update ទិន្នន័យ**
* **មាន Search Engine Optimization**
* **ដំណើរការលឿនក្នុងការ Loading pages (for some CMS’s)**
* អាច update ព័ត៌មានរាល់ថ្ងៃ
* អាចធ្វើការជាប្រព័ន្ធក្នុងការបញ្ចូលទិន្នន័យ

គុណវិបត្តិនៃ Dynamic Websites

* ការបង្កើតមានភាពពិបាក និងចំនាយពេលយូរ
* **ទាមទារការចេះភាសាកូដច្រើន**
* ការ Hosting ត្រូវចំនាយថវិការច្រើន

រូបភាពទី ២.៤ ៖ ដំណើរការរបស់ Dynamic Websites

## ២.៧. សេចក្តីណែនាំអំពីភាសាកូដ

### ២.៧.១. ភាសា HTML និង HTML5

#### **២.៧.១.១. HTML**

HTML (Hypertext Markup Language) ត្រូវបានគេប្រើប្រាស់យ៉ាងទូលំទូលាយសម្រាប់ការសរសេរគេហទំព័រ។ Hypertext សំដៅទៅលើមធ្យោបាយសម្រាប់ឲ្យ Web Pages (HTML Documents) មានទំនាក់ទនងគ្នាទៅវិញទៅមក។​​ Markup Language គឺជាភាសាសម្រាប់ពិពណ៌នាពីរបៀបដែល HTML ដំណើរការ។ វាជាអ្នករៀបចំអត្ថបទ ឬឯកសារដោយប្រើប្រាស់ Tags ដែលប្រើសម្រាប់ប្រាប់ Web Browser ថាត្រូវបង្ហាញអត្ថបទ ឬឯកសារទាំងនោះដោយរបៀបណា (Tutorialspoint[a], 2014)។ HTML គឺជាមធ្យោបាយមួយសម្រាប់កំណត់ Contents របស់គេហទំព័រដោយប្រើ tag ដែលកើតឡើងដោយប្រើសញ្ញា Pointy Brackets (< >) ។ HTML ត្រូវបានធ្វើឲ្យមានលក្ខណៈ ជាស្តង់ដារ លើកដំបូងនៅក្នុងឆ្នាំ 1993 ហើយវាជាប្រេងឥន្ធនៈសម្រាប់បញ្ឆេះ វើលវ៉ាយវ៉េប (World Wide Web) (Fulton & Fulton, 2011) ។

អត្តប្រយោជន៍នៃការប្រើប្រាស់ HTML

* HTML **មានភាពងាយស្រួលយល់ និងងាយស្រួលប្រើប្រាស់**
* HTMLSupport **គ្រប់** browsers
* HTML អាចប្រើដោយមិនចាំបាច់បង់ប្រាក់ (free)
* HTML Support ច្រើនកម្មវិធីដែលប្រើប្រាស់សម្រាប់សរសេរវា ឧទាហរណ៍ Sublime Text, Atom, Notepad, Notepad++ ................
* HTML ធ្វើឲ្យ Search Engine ងាយស្រួលរកឃើញ
* HTML មិនប្រកាន់អក្សរធំតូចទេ(VTech SEO, 2012)។

#### **២.៧.១.២. HTML5**

HTML5 គឺជា Version ថ្មីរបស់ Hypertext Markup Language សម្រាប់ Websites ដែលបង្កើតដោយ World Wide Web Consortium (W3C)។ HTML5 ត្រូវបានបង្កើតឡើងដំបូងនៅឆ្នាំ ២០០៨ រហូតដល់ ២០១១ ទើបគេចាប់ផ្តើមសរសេរ និង​ប្រើបាស់វា ប៉ុន្តែវាមិនទាន់ support គ្រប់ browsers ទេ។ HTML5 បានបន្ថែម Tag មួយចំនួនដូចជា: article, aside, header, footer, nav, section, audio, video, embed និង canvas(1st Web Designer, 2014) ។

### ២.៧.២. ភាសា CSS និង CSS3

#### **២.៧.២.១ CSS**

CSS (Cascading Style Sheets) គឺជាភាសា ដែលបានរចនាឡើងសម្រាប់ការពិពណ៌នាអំពី រូបរាងនៃឯកសារដែលសរសេរក្នុងភាសា Markup Language ដូចជា HTML ។ ជាមួយនឹង CSS អ្នកអាចត្រួតពិនិត្យពណ៌នៃអត្ថបទ រចនាបថ ពុម្ពអក្សរ គម្លាតរវាងកថាខណ្ឌ ពណ៌ ឬ រូបភាព Background ដែលត្រូវប្រើ និងប្រភេទ Effects ផ្សេងៗទៀត។

ប្រវត្តិរបស់ CSS ៖

* ក្នុងឆ្នាំ 1994 ត្រូវបានស្នើឡើងដោយ Håkon Wium Lie,
* នៅក្នុង ខែធ្នូ ឆ្នាំ 1996 CSS Level 1 ត្រូវបាន published (បោះពុម្ភផ្សាយ)
* ក្នុងឆ្នាំ 1998 CSS2 បានគេ Recommend ឲ្យប្រើប្រាស់ (ហើយ CSS 2.1 បង្កើតឡើង សម្រាប់កែកំហុស របស់ CSS 2) ។
* បច្ចុប្បន្ន CSS 3 (Pouncey & York, 2011)។

អត្តប្រយោជន៍នៃការប្រើប្រាស់ CSS ៖

* CSS **ធ្វើឲ្យចំនាញពេលវេលា៖ យើងអាចសរសេរ** CSS **មួយហើយយកទៅ ប្រើប្រាស់សម្រាប់** HTML **ជាច្រើន** Pages**។ យើងអាចកំណត់** style **(រចនាបថ)សម្រាប់** HTMLElement **(ធាតុរបស់** HTML**) នីមួយៗហើយយកទៅប្រើប្រាស់សម្រាប់គ្រប់** Element **ទាំងឡាយណាដែលយើងចង់បាន។**
* CSS **ធ្វើឲ្យគេហទំព័រដើរលឿនៈ ការប្រើប្រាស់**CSS **ធ្វើឲ្យយើងសរសេរកូដតិចជាងការ ដែលជាហេតុនាំឲ្យគេហទំព័រដើរលឿន។**
* **ងាយស្រួលគ្រប់គ្រងៈ ប្រសិនបើយើងចង់កែ** Style **(រចនាបថ) នៅគ្រប់ទំព័រ របស់គេហទំព័រ យើងមិនចាំបាច់ កែគ្រប់** HTML Pages **ទាំងអស់ទេ គឺយើងគ្រាន់តែកែលើ** CSSfile **របស់យើងតែប៉ុណ្ណោះ។**
* CSS **ធ្វើឲ្យ** HTML **មាន** Style **ដ៏ល្អប្រសើរជាង** Style **ដែលសរសេរដោយ** HTML Attributes**។**
* CSS **អាចធ្វើឲ្យ** Support **នៅ** Device **(ឧបករណ៍) ជាច្រើនប្រភេទ។**
* CSS **មានលក្ខណៈស្តង់ដារទូទៅនៃការបង្កើតគេហទំព័រ** (Tutorialspoint[b], 2014). **។**

#### **២.៧.២.២ CSS3**

CSS3 **គឺជា** Version **ថ្មីរបស់** CSS​ **ដែលវាបានបន្ថែមនូវលក្ខណៈ និងមុខងារថ្មីៗ​** (New Features) **ដើម្បីជួយដោះស្រាយនូវបញ្ហាផ្សេងៗ ដោយមិនចំបាច់ប្រើប្រាស់** non-semantic markup, script **ដែលស្មុកស្មាញ** (Complex Scripting) **ឬការបន្ថែមរូបភាពទេ។**

**កាលពីមុនដើម្បីអាចបង្កើតជា** Gradients, Shadows​ **និង** Rounded Corners **អ្នករចនា គេហទំព័រត្រូវប្រើប្រាស់តិចនិចជាច្រើនពេលខ្លះត្រូវបន្ថែម** HTML elements **ពេលខ្លះត្រូវ បន្ថែមរូបភាព។**

* CSS3 **អាចឲ្យយើងអាចបន្ថែម​ការរចនាបែបនេះបាន​ដោយងាយស្រួល** ( Site Point, 2014 )**។**

### ២.៧.៣ ភាសា JavaScript

JavaScript គឺជា Client-Side Scripting Language មួយដែលផ្ដល់ភាពងាយស្រួលក្នុងការបង្កើត Online Application ដោយផ្ទាល់សម្រាប់ធ្វើការ Link ទៅកាន់ Objects ផ្សេងៗ និង Resource នៅលើ Client និង Servers ផងដែរ។ JavaScript ជាភាសាមួយប្រភេទដែលគេបង្កើតឡើងសម្រាប់ជួយសម្រួលក្នុងការបង្កើតវ៉ិបសាយ (Website) ឲ្យមានភាពរស់រវើក។ ដូចដែលយើងធ្លាប់បានឃើញថានៅលើវ៉ិបសាយខ្លះមានប្រមាណវិធីគណនា និងខ្លះទៀតមានភាពរស់រវើកដូចជាការ កែប្រែលក្ខណៈរបស់ Browser មានការកំនត់ឲ្យអក្សរត់លើ Status Bar ប្ដូរពណ៌របារ សរសេរម៉ោង ប្រក្រតិទិនជាដើម។ JavaScript ក៏ត្រូវបានអាចគេប្រើប្រាស់សម្រាប់ធ្វើការបង្កើតជា Game សម្រាប់ជាការកំសាន្ត ដែលមានលក្ខណៈជា Dynamic Game និង Static Game (ទៀង វុទ្ធីដែន, ២០០៨).។

អត្តប្រយោជន៍នៃការប្រើប្រាស់ JavaScript ៖

* ធ្វើការជាមួយ Server តិចតួចៈ យើងអាចធ្វើការបញ្ជាក់ការបញ្ជូលទិន្នន័យរបស់អ្នកប្រើប្រាស់ (Validate User Input) ដោយមិនចំបាច់បញ្ជូនទៅកាន់ Server ដែលជាហេតុ Server ដំណើរការលឿន។
* មានប្រតិកម្ម (Response or Feedback) មកកាន់អ្នកប្រើប្រាស់ ឬអ្នកទស្សនាគេហទំព័រភ្លាមៗ​ នៅពេលគេ​ចង់អ្វីធ្វើមួយ (User Request)។
* អាចបង្កើននូវអន្តរកម្ម និងភាពទាក់ទាញរវាងអ្នកប្រើប្រាស់ និងគេហទំព័ររបស់យើងដោយការប្រើប្រាស់ Slider, Drag-and-Drop Components និង event ផ្សេងៗ (Tutorial spoint[c], 2014)។

### ២.៧.៤. Jquery

jQuery **គឺជា** library **ដែលមានលក្ខណៈ តូច លឿន និងមានគ្រប់** feature **របស់** Javascript​ **។ វាធ្វើឲ្យ** HTML document traversal manipulation, event handling (**ការគ្រប់គ្រងលើ** event), animation **និង** Ajax **មានលក្ខណៈងាយស្រួលប្រើប្រាស់ក្នុងគ្រប់** Browsers។ jQuery **បានផ្លាស់ប្ដូររបៀបដែលមនុស្សរាប់លាននាក់ សរសេរភាសា** Javascript **ដែលធ្វើឲ្យមានភាពខ្លី និងងាយស្រួលជាងមុន (**jQuery, 2014**)។**

អត្តប្រយោជន៍នៃការប្រើប្រាស់ Jquery៖

* **ងាយស្រួលប្រើប្រាស់បើធៀបជាមួយ** JavaScript **វាមានភាពងាយស្រួលព្រោះវាត្រូវការសរសេរកូដខ្លីដើម្បីទទួលបានលក្ខណៈ ឬ ដំណើរការដូចគ្នា។**
* Large library: JQuery **មាននូវ** functions​ **ជាច្រើនបើប្រៀបធៀបទៅនឹង** library **របស់** JavaScript​ **ផ្សេងទៀត។**
* Strong open source community:
* **មានឯកសារ និង​មេរៀនច្រើនដែលធ្វើឲ្យងាយស្រួលក្នុងសិក្សា**
* Ajax support (JScripters, 2014)**។**

### ២.៧.៥ Ajax

Ajax (Asynchronous JavaScript and XML) គឺជាទិចនិចក្នុងការបង្កើត Web pages ដែលមានល្បឿនលឿន និង​មានលក្ខណៈ Dynamic។ Ajax អនុញ្ញាតឱ្យយើង update ផ្នែកមួយចំនួននៃ web pages ដោយមិនចាំបាច់ reloading pages ទាំងមូលឡើយដែលធ្វើឲ្យ Web pages ដើរលឿន។ Web Applications ដែលប្រើ AJAX មាន Google Maps, Gmail, Youtube, **និង** Facebook tabs (w3schools, 2014)។

អត្តប្រយោជន៍នៃការប្រើប្រាស់ Ajax៖

* Ajax **ងាយស្រួលក្នុងការសិក្សា និងងាយស្រួលយល់។**
* Ajax **កាត់បន្ថយការចំនាយពេលវេលា និងបង្កើនល្បឿនក្នុងការបញ្ជូនទិន្នន័យទាំងការ** request **និងការ** response**.**
* Ajax **ក្នុង** web page **ទាំងមូលយើងអាចឲ្យវាធ្វើការបញ្ជូនទិន្នន័យតែមួយផ្នែកតូចក្នុងពេលតែមួយ។**
* **យើងអាចប្រើប្រាស់** Web Browser Scripting language **ដូចជា** JavaScript **ដើម្បីបញ្ជូន** XML **និងទិន្នន័យផ្សេងទៀត និងប្រើប្រាស់** Web server **ដោយប្រើប្រាស់** HTTP**។**
* Ajax **អនុញ្ញាតឲ្យប្រើប្រាស់សមត្ថភាពក្នុងការ** asynchronous calls **ទៅកាន់** Web Server**។ វាអនុញ្ញាតឲ្យ** client browser **អាចជៀសវាងការរង់ចាំទិន្នន័យទាំងអស់មកដល់មុននឹងអាចឲ្យអ្នកប្រើប្រាស់អាចធ្វើការងារមួយផ្សេងទៀត។**
* Ajax **អាចឲ្យយើងធ្វើការបញ្ជាក់ការក្នុងបញ្ជូលទិន្នន័យរបស់អ្នកប្រើប្រាស់ (**Validation User input) **ដែលជាអត្ថប្រយោជន៍ដ៏សំខាន់មួយ។**
* Ajax **មិនចំបាច់** Reload Page **ទេដែលធ្វើឲ្យយើងប្រីប្រាស់** bandwidth**(ចំនួនទិន្នន័យដែលអាច** send **ពី** Computer **មួយទៅ** Computer **មួយ) របស់** Server **តិច** (Way2tutorial, 2014)**។**

### ២.៧.៦ JSON

JSON (JavaScript Object Notation) គឺជាទម្រង់នៃការផ្លាស់ប្តូរទិន្នន័យទៅវិញទៅមកមានលក្ខណៈស្រាល ឬមានទំហំតូច (lightweight data-interchange format)។ វាមានភាពងាយស្រួលសម្រាប់មនុស្សក្នុងការអាន​ និងសរសេរហើយងាយស្រួលសម្រាប់ម៉ាស៊ីនក្នុងការចម្លង និងបង្កើតទិន្នន័យ។

JSON មានពីទម្រង់គឺ៖

* ជាបន្ដុំនៃឈ្មោះ និងតម្លៃ (name/value pairs) ដែលភាសាកូដដ៏ទៃទៀតស្គាល់ថាជា object, record, struct, hash table, associative array…
* ជាបន្ដុំនៃតម្លៃដែលភាសាកូដដ៏ទៃទៀតស្គាល់ថាជា Array, sequence (JSON, 2014)។

អត្តប្រយោជន៍នៃការប្រើប្រាស់ JSON៖

JSON គឺជាសមាសភាពដែលមានទំហំតូច និង​មិនត្រូវការពេលច្រើនក្នុងការ execution។ វា support data structure ដែលប្រើក្នុងភាសាកូដទំនើបៗ (modern languages)។ ក្នុងភាសា JavaScript យើងមិនអាចទាញតម្លៃពី files ផ្សេង ឧទាហរណ៍ដូចជា Java។ តែនៅក្នុងការ JSon យើងអាចទាញតម្លៃពីគ្រប់ទីកន្លែង។ វា support គ្រប់ភាសាកូដ (CandidJava, 2014)។

### ២.៧.៧ Bootstrap

Bootstrap គឺជា CSS **និង** JavaScript Framework ដ៏ល្បីល្បាញមួយដែលប្រើប្រាស់សម្រាប់បង្កើត website និង mobile project ដែលមានលក្ខណៈ responsive និងស្រស់ស្អាត។ Bootstrap ធ្វើឲ្យការបង្កើត front-end web មានភាពឆាប់រហ័ស និងមានភាពងាយស្រួល។ Bootstrap គឺជា open source ដែលត្រូវបាន host ​ បង្កើត និងថែទាំដោយ GitHub (Bootstrap, 2014)។

2. វិធីសាស្ត្រនៃការស្រាវជ្រាវ

## ៣.១. វិធីសាស្រ្តនៃការសិក្សា

ស្ទើរតែគ្រប់ការងារទាំងអស់មុននឹងឈានទៅរកជោគជ័យបានគឺ តម្រូវ និងទាមទារឲ្យយើងស្វែងរកនូវវិធីសាស្រ្តថាតើធ្វើដូចម្តេចដើម្បីដោះស្រាយនូវការងារនេះឲ្យបានសម្រេច។

## ៣.២. ប្រភេទទិន្នន័យ និងបច្ចេកទេសប្រមូលទិន្នន័យ

### ៣.២.១ ប្រភេទទិន្នន័យ

ក្រោយពីក្រុមនិសិ្សតយើងខ្ញុំបានចុះទៅធ្វើការប្រមូលទិន្នន័យរួចមក យើង​ខ្ញុំ​​បានកំណត់យកទិន្នន័យជាពីរប្រភេទគឺ ទិន្នន័យចម្បង និងទិន្នន័យ​​បន្ទាប់បន្សំ។

#### **៣.២.១.១ ទិន្នន័យចំបង**

គឺជាប្រភេទទិន្នន័យដែលសំខាន់​ ហើយវាឆ្លើយតបតាមតម្រូវការរបស់ប្រធាន​បទ​នៃ​សារណាហើយជាប្រភេទទិន្នន័យដែលទទួលបានពី៖

* ការប្រមូលយកឯកសារដែលពាក់ព័ន្ធនឹងកម្មវិធីដែលកំពុងប្រើប្រាស់
* ការធ្វើការសម្ភាសន៍ជាមួយអ្នកគ្រប់គ្រង ឬបុគ្គលិកផ្សេងៗ
* ការចុះធ្វើការអង្កេតដោយផ្ទាល់

#### **៣.២.១.២ ទិន្នន័យបន្ទាប់បន្សំ**

គឺជាប្រភេទទិន្នន័យដែលបានចំលងទុកធ្វើជាឯកសារសម្រាប់យកមកប្រើប្រាស់នៅ​​ពេលក្រោយ។​ រាល់គ្រប់ទិន្នន័យទាំងអស់ ដែលប្រមូលយកមកធ្វើការវិភាគ និងបង្ហាញមាន​ប្រភព​មកពី៖

* ឯកសារផ្សេងៗក្នុងបណ្ណាល័យរបស់សាកលវិទ្យាល័យបៀលប្រាយ សាខា​ខេត្ត បាត់ដំបង
* ការស្រាវជ្រាវតាមបណ្តាញអ៊ីនធើណិត
* ឯកសារដែលទាក់ទងនឹងការរំលឹកទ្រឹស្តីដែលទទួលបានមកពីសៀវភៅកិច្ចការ និងលោកគ្រូសាស្ត្រាចារ្យ ។

### ៣.២.២ បច្ចេកទេសប្រមូលទិន្នន័យ

**ក្នុងការប្រមូលទិន្នន័យពីក្រុមហ៊ុន Rabbit IT Solution ក្រុមនិស្សិតយើងខ្ញុំបានជ្រើសរើសនូវ​វិធី​សាស្រ្តបីប្រភេទ​​ គឺ៖**

**Interview: ក្រុមនិស្សិតយើងខ្ញុំធ្វើការណាត់ជាមួយនឹងលោកប្រធានក្រុមហ៊ុនដើម្បី​សាកសួរព័ត៌មាន ការសម្ភាសន៍ជាមួយនឹងអ្នកលក់ ១នាក់ ការសម្ភាសន៍ជាមួយនឹងអ្នកទីផ្សារ ចំនួន១នាក់ និងសម្ភាសន៍ជាមួយនឹងអ្នកប្រើប្រាស់ផ្ទាល់ចំនួន២ នាក់។**

**Observation: ក្រុមនិស្សិតយើងខ្ញុំ​ ធ្វើការចុះអង្កេតដោយផ្ទាល់នៅក្រុមហ៊ុន Rabbit IT Solution ពីការសរសេរ និងលក់ប្រព័ន្ធ ហើយពិនិត្យមើលនូវ​ការរៀបចំ​ឯកសារ​ផ្សេងៗនៅក្នុងក្រុមហ៊ុន ។**

**Document Review: ក្រុមនិស្សិតយើងខ្ញុំ ធ្វើការស្នើសុំនូវឯកសារដែលពាក់ព័ន្ធនឹងក្រុមហ៊ុន ក្នុងនោះរួមមាន Software ជាពិសេស កម្មវិធី​ដែលកំពុង​ប្រើប្រាស់​បច្ចុប្បន្ន ។**

៣.៣. ការវិភាគ និងការរៀបចំទិន្នន័យ

### ៣.៣.១ ការវិភាគទិន្នន័យ

នៅក្នុងការស្រាវជ្រាវលើប្រធានបទ​ **“**ប្រព័ន្ធគ្រប់គ្រងការលក់ប្រព័ន្ធគ្រប់គ្រង​​

លក្ខណៈ Web-based Application និងគេហទំព័រ របស់​ក្រុមហ៊ុន Rabbit IT Solution**”** នេះ ក្រុមនិស្សិតយើងខ្ញុំបានជ្រើសរើសនូវវិធីសាស្រ្ត Structure Analysis ក្នុងការវិភាគទិន្នន័យដែលហៅថា Systems Development Life Cycle (SDLC) ដែលមាន ៥ ដំណាក់កាល គឺ៖

តួលេខ ៣.១ Systems Development Life Cycle

**ដំណាក់កាលទី ១**

System Planning

**ដំណាក់កាលទី ២**

System Analysis

**ដំណាក់កាលទី ៣**

System Design

**ដំណាក់កាលទី ៤**

System Implementation

**ដំណាក់កាលទី ៥**

System Operation and Support

* System Planning គឺជាដំណាក់កាលនៃការរៀបចំគម្រោងក្នុងការប្រមូលទិន្នន័យ និងការកំណត់លក្ខណៈពិសេសរបស់ប្រព័ន្ធ ។
* System Analysis គឺជាដំណាក់កាលនៃការវិភាគទិន្នន័យទៅលើដំណើរការប្រតិបត្តិការរបស់ប្រព័ន្ធគ្រប់គ្រង ។
* System Design គឺជាដំណាក់កាលនៃការចាប់ផ្តើមបង្កើតនូវទំរង់នៃប្រព័ន្ធដើម្បី​បំពេញទៅតាមតម្រូវការ ។
* System Implementation គឺជាដំណាក់កាលនៃការចាប់ផ្តើមសរសេរកូដដើម្បីសាកល្បងស្វែងរកនូវកំហុសទាំងឡាយ មុននឹងតំឡើងប្រព័ន្ធនេះឲ្យ​អ្នកប្រើប្រាស់ ។
* System Operation and Support គឺជាដំណាក់កាលនៃការចាប់ផ្តើមនូវដំណើរ​ការ​របស់ប្រព័ន្ធដែលត្រូវបានប្រើដោយអ្នកប្រើប្រាស់ផ្ទាល់ និងបំរើសេវាថែទាំបន្ថែមទៅលើការប្រើប្រាស់ប្រព័ន្ធ ។

៣.៣.២. ការរៀបចំទិន្នន័យ

ក្នុងការរៀបចំទិន្នន័យ ក្រុមនិស្សិតយើងខ្ញុំបានជ្រើសរើសនូវវិធីសាស្រ្ត ២ យ៉ាង៖

* Entity Relationship គឺជាការភ្ជាប់ទំនាក់ទំនងរវាង​ Entity និង Entity ។
* Data Flow Diagram (DFD) គឺជា Diagram មួយសម្រាប់បង្ហាញពីដំណើរការ​ របស់ប្រព័ន្ធគ្រប់គ្រងព័ត៌មាន ប៉ុន្តែវាមិនបានបង្ហាញពីជំហានដំណើរការទេ ។

## ៣.៤. ​វិធីសាស្រ្តរចនា និង អភិវឌ្ឍកម្មវិធី

* ប្រើប្រាស់កម្មវិធី Atom និង Webstorm 10 សម្រាប់ការសរសេរ Code នៃភាសា HTML, CSS, Jquery និង JavaScript ។
* ប្រើប្រាស់ MongoDB សម្រាប់ Store Collection។
* ប្រើប្រាស់កម្មវិធី RoboMongo សម្រាប់ពិនិត្យ និងកែ MongoDB Collection
* ប្រើប្រាស់ Web Browser ដូចជា Mozila Firefox, Internet Explorer, Opera, Google Chrome សម្រាប់ view​ HTML, CSS, JQuery និង JavaScript ។
* ប្រើប្រាស់ Adobe Photoshop CS6 សម្រាប់រចនារូបភាពផ្សេងៗ

## ៣.៥. វិធីសាស្រ្តសាកល្បង

ដើម្បីឲ្យកម្មវិធីដែលបានបង្កើតនេះត្រឹមត្រូវ និង អាចយកទៅប្រើប្រាស់បានយើងត្រូវធ្វើ ការសាកល្បង​។ ហើយការសាកល្បងនេះមានដូចជា៖

* ចូលទៅកាន់ប្រព័ន្ធតាមរយៈកុំព្យូទ័រផ្សេង និងប្រើប្រាស់ប្រភេទ Browser ​

ផ្សេងៗគ្នា

* សាកល្បងបញ្ចូលទិន្នន័យទៅក្នុង Collection
* សាកល្បងទាញយកទិន្នន័យពី Collection
* សាកល្បងកែប្រែ និង​លុបទិន្នន័យពី Collection
* សាកល្បង Backup និង Restore ទិន្នន័យដែលមាននៅលើប្រព័ន្ធ
* និងសាកល្បងមុខងារផ្សេងទៀតរបស់ប្រព័ន្ធនេះ។

# លទ្ធផលនៃការស្រាវជ្រាវ

## ៤.១. ការវិភាគប្រព័ន្ធ

### ៤.១.១ ប្រព័ន្ធបច្ចុប្បន្ន

ទោះបីជាបច្ចុប្បន្នក្រុមហ៊ុន Rabbit IT Solution មានការស្គាល់ និងគាំទ្រច្រើនពីសំណាក់ក្រុមហ៊ុននានា ក៏ប៉ុន្តែក្រុមហ៊ុនកំពុងតែនៅប្រើប្រាស់ប្រព័ន្ធការងារដែលបំពេញដោយដៃនៅឡើយ ហេតុដូច្នេះហើយទើបការងាររបស់ក្រុមហ៊ុននៅតែមានភាពយឺតយ៉ាវ និងប្រទះនូវការលំបាកជាច្រើនដែលទាក់តងនឹងការគ្រប់គ្រងទិន្នន័យ។

ដោយមើលឃើញថាក្រុមហ៊ុននៅមានចំនុចខ្វះខាត និងមានផលវិបាកជាច្រើននៅក្នុងការគ្រប់គ្រងទិន្នន័យ ទើបក្រុមយើងខ្ញុំបានស្នើសុំទៅអ្នកគ្រប់គ្រងរបស់ក្រុមហ៊ុន Rabbit IT Solution ដើម្បីចុះទៅសិក្សាស្រាវជ្រាវ និងរៀនចំប្រព័ន្ធគ្រប់គ្រងថ្មីមួយដែលជាភាពងាយស្រួលដល់គ្រប់គ្រងក្នុងការគ្រប់គ្រងទៅលើការងារផ្សេងៗនៅក្នុងក្រុមហ៊ុនប្រកប​ដោយប្រសិទ្ឋភាពជាងមុន។

### ៤.១.២ ប្រព័ន្ធថ្មី

ការប្រើប្រាស់ប្រព័ន្ធថ្មីនេះតម្រូវយើងឲ្យប្រើប្រាស់ និងរក្សាទុកទិន្នន័យនៅលើប្រព័ន្ធកុំព្យូទ័រ ដោយវាបានកាត់បន្ថយនូវការបំពេញការងារដោយដៃមួយចំនួនធំ កាត់បន្ថយនូវចំនួនបុគ្គលិកច្រើន និងហានិភ័យផ្សេងៗដែលទាក់ទងនឹងសុវត្តិភាពទិន្នន័យបានយ៉ាងច្រើន។

តួលេខ ៤.១ ដ្យាក្រាម Context សម្រាប់ប្រព័ន្ធថ្មី

(Context Diagram for Proposed System)

អតិថិជន

អ្នកគ្រប់គ្រង

ដ្យាក្រាមលំហូរសម្រាប់ប្រព័ន្ធដាតាបេស

(Data Flow Diagram for Database System)

ដ្យាក្រាមទំនាក់ទំនង Entity សម្រាប់ប្រព័ន្ធគ្រប់គ្រងទិន្នន័យ

(Entity Relationship Diagram for Proposed System)

### ៤.១.៣ Entity Relationship Diagram (E-R Diagram)

## ៤.២. Data Description and Dictionary for Collection

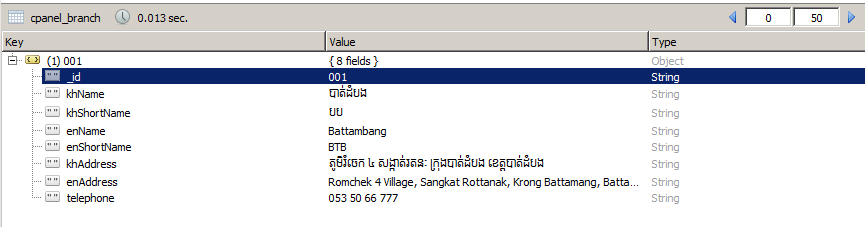
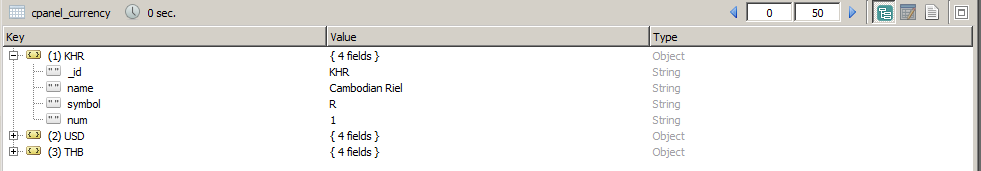
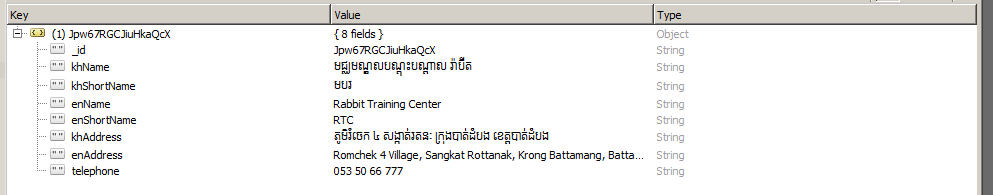
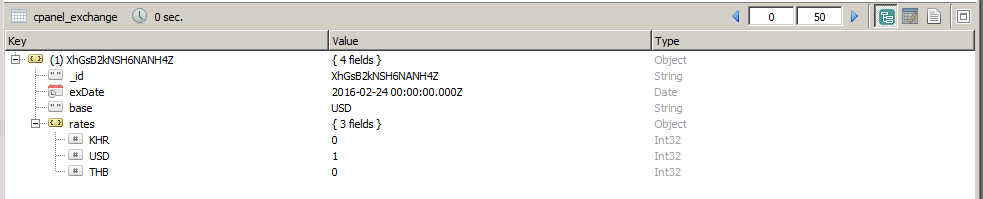
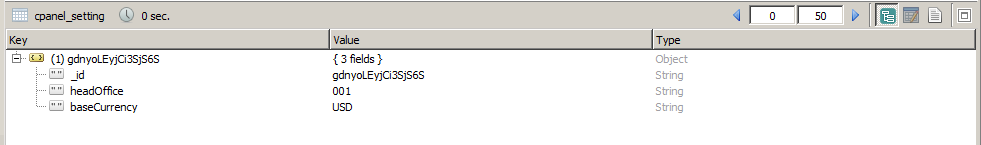
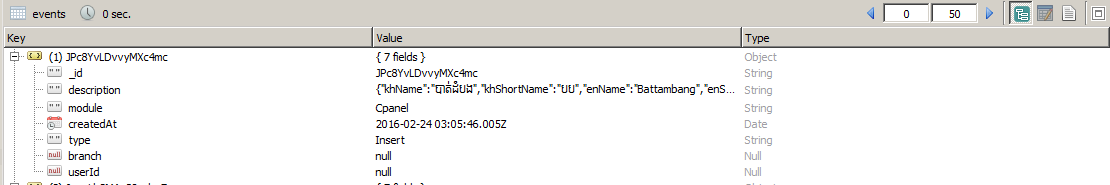
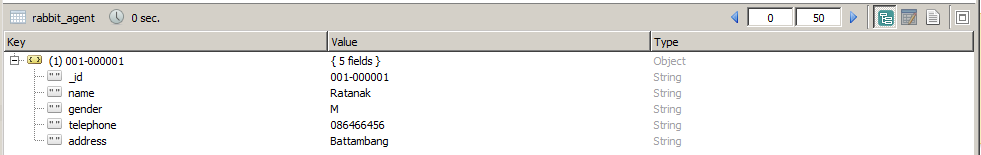
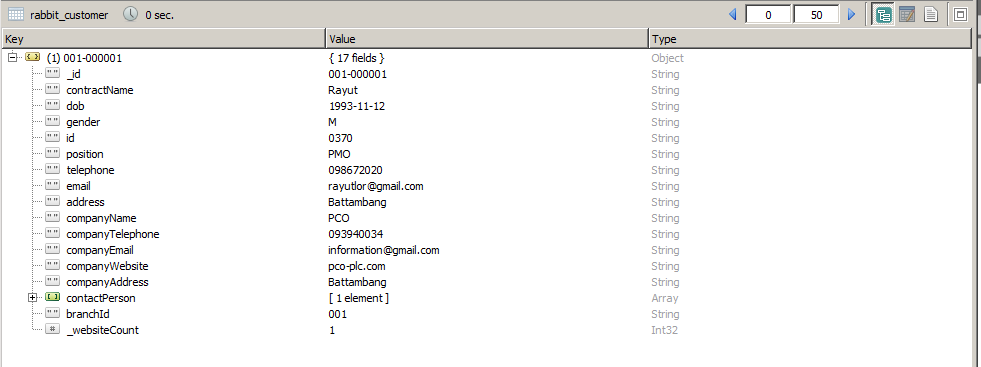
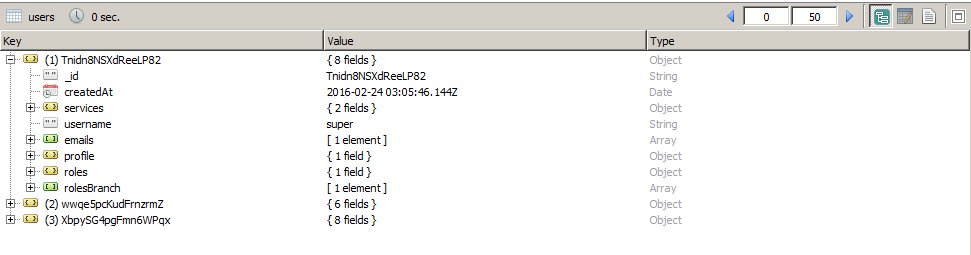
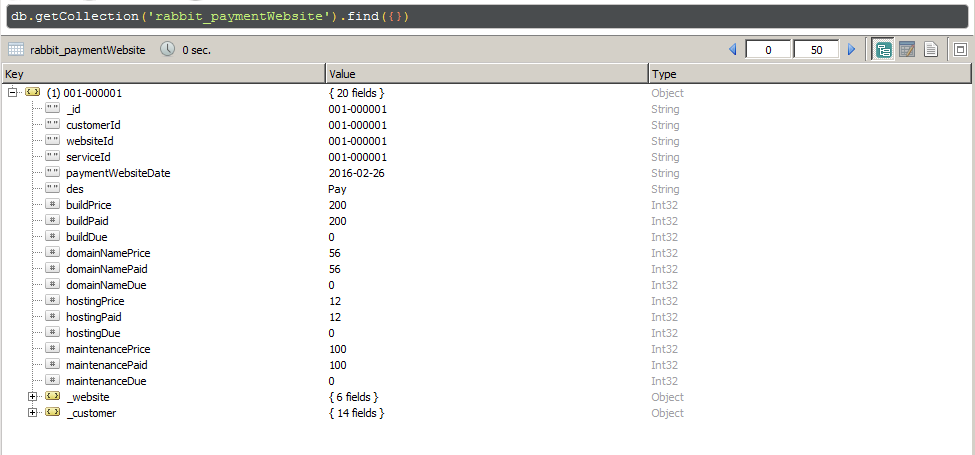
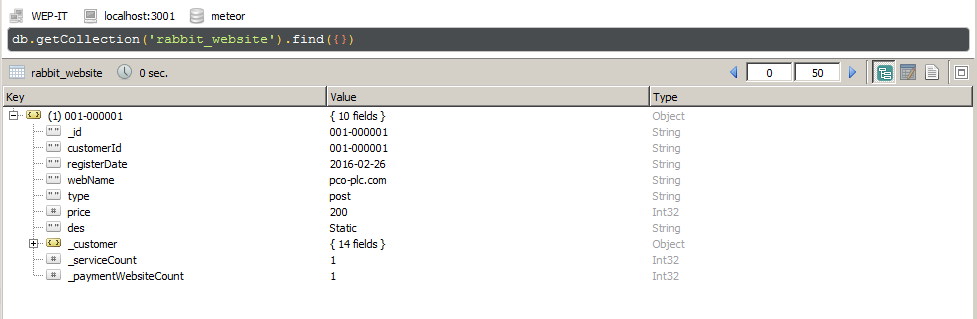
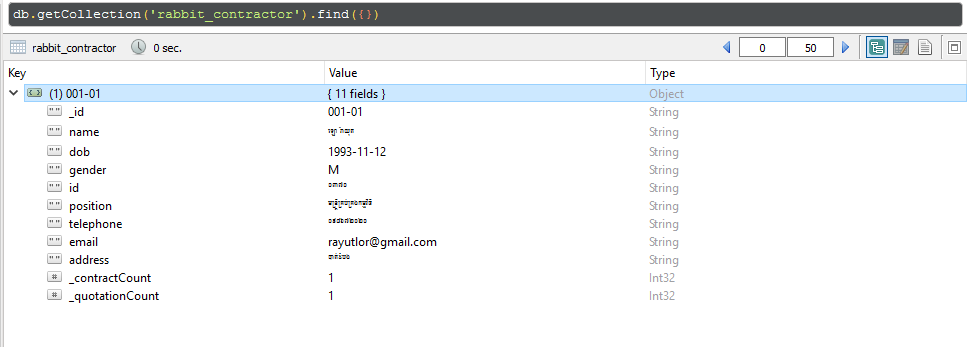
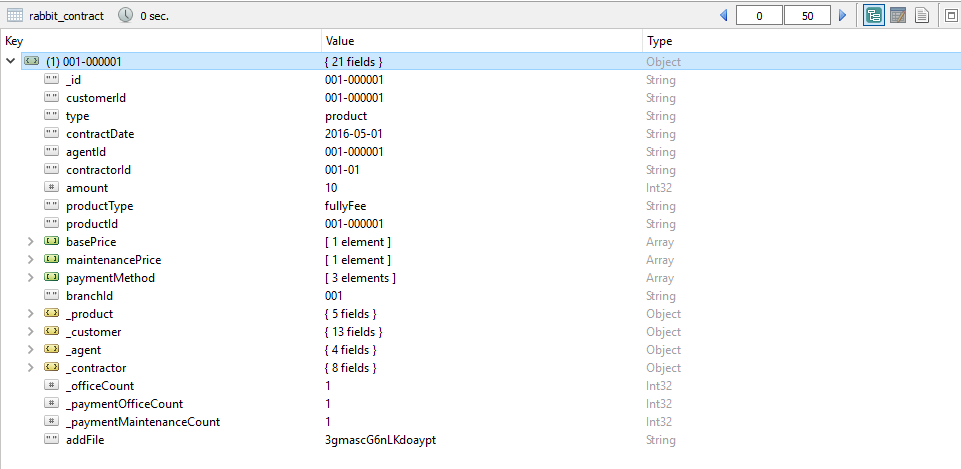
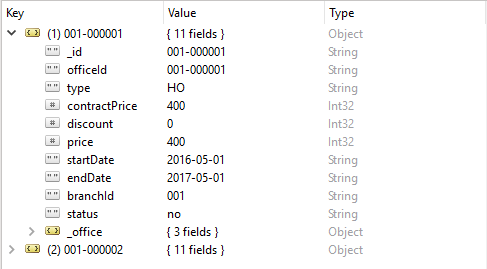
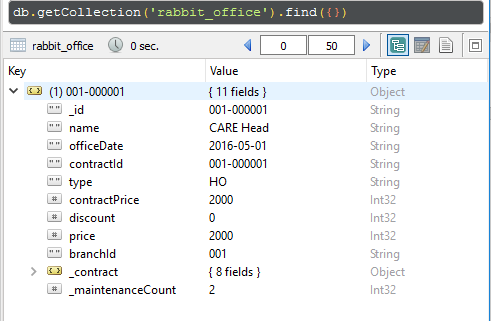
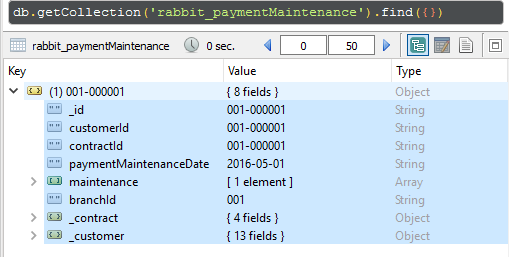
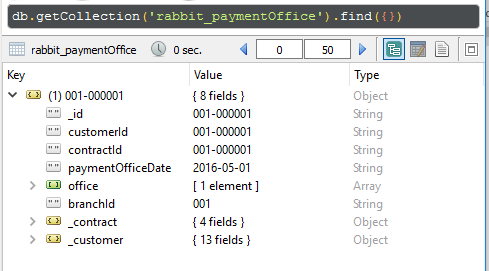
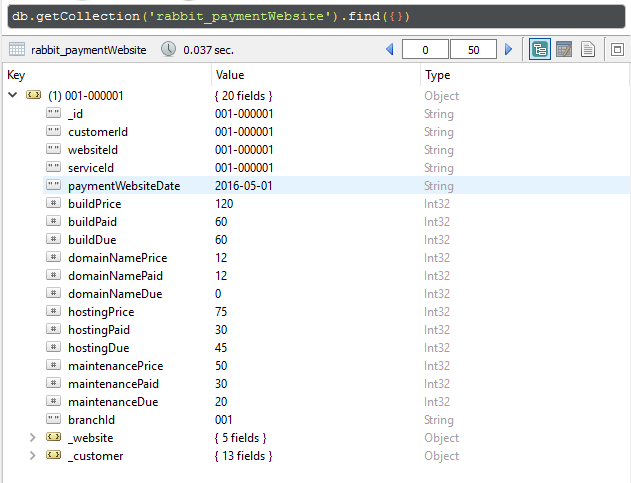
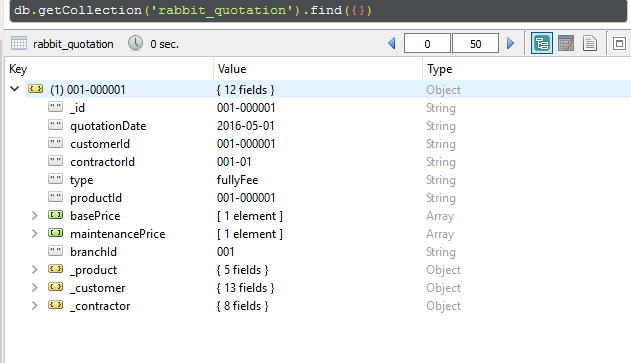
### ៤.២.១ Data Description

តារាង ៤.១ Data Desription

|  |  |  |
| --- | --- | --- |
| No | Data Store | Discription |
| 1 | cpanel\_branch​ | សម្រាប់ផ្ទុកព័ត៍មានរបស់សាខា |
| 2 | cpanel\_company | សម្រាប់ផ្ទុកព័ត៍មានរបស់ក្រុមហ៊ុន |
| 3 | cpanel\_currency | សម្រាប់ផ្ទុកព័ត៍មានប្រភេទប្រាក់ |
| 4 | cpanel\_exchange | សម្រាប់ផ្ទុកព័ត៍មានការប្តូរប្រាក់ |
| 5 | cpanel\_setting | សម្រាប់ផ្ទុកព័ត៍មាន របស់អ្នក​ប្រើប្រាស់ |
| 6 | events | សម្រាប់ផ្ទុកព័ត៍មាន​ ព្រឹត្តិការណ៍ដែលបានធ្វើ |
| 8 | rabbit\_product | សម្រាប់ផ្ទុកព័ត៍មានផលិតផលរបស់ក្រុមមហ៊ុន |
| 9 | rabbit\_website | សម្រាប់ផ្ទុកព័ត៍មានវេបសាយ |
| 10 | role | សម្រាប់ផ្ទុកព័ត៍មានកំណត់សិទ្ធទៅឲ្យអ្នកប្រើប្រាស់ |
| 11 | users | សម្រាប់ផ្ទុកព័ត៍មានរបស់អ្នកប្រើប្រាស់ |
| 12 | rabbit\_agent | សម្រាប់ផ្ទុកពត៌មានអ្នកដែលនាំមក |
| 13 | rabbit\_contract | សម្រាប់ធ្វើការចុះកិច្ចសន្យា |
| 14 | rabbit\_contractor | សម្រាប់អ្នកដែលចុះកិច្ចសន្យា |
| 15 | rabbit\_maintenance | សម្រាប់ការថែទាំ |
| 16 | rabbit\_office | សម្រាប់ផ្ទុកពត៌មានដែលទាក់ទងនិងការលក់ប្រព័ន្ធ |
| 17 | rabbit\_paymentOffice | សម្រាប់ផ្ទុកពត៌មានដែលបានធ្វើការបង់ប្រាលក់ទៅតាមការិយាល័យនីមួយៗ |
| 18 | rabbit\_paymentMaintenance | សម្រាប់ផ្ទុកពត៌មានការបង់ប្រាក់ថែរទាំប្រព័ន្ធ |
| 19 | rabbit\_paymentWebsite | សម្រាប់ផ្ទុកពត៌មានដែលបានធ្វើការបង់ប្រាក់ទៅលើការទិញវេបសាយ |
| 20 | rabbit\_quotation | សម្រាប់ផ្ទុកពត៌មានដែលទាក់ទងទៅនិងការសាក់សួរពត៌មាន |
| 21 | rabbit\_service | សម្រាប់ផ្ទុកពត៌មានដែលទាក់ទងទៅនឹងសេវាកម្ម |

### ៤.២.២ Data Dictionary for Collection

ប្រព័ន្ធគ្រប់គ្រងដែលក្រុមស្រាវជ្រាវយើងខ្ញុំបានបង្កើតឡើងចាំបាច់ត្រូវបង្កើត Collection ដូចខាងក្រោម៖

* cpanel\_branch សម្រាប់ផ្ទុកព័ត៌មានរបស់សាខា
* cpanel\_currency សម្រាប់ផ្ទុកព័ត៌មានរបស់ប្រភេទរូបិយប័ណ្ណ
* cpanel\_company សម្រាប់ផ្ទុកព័ត៌មានរបស់ក្រុមហ៊ុន
* cpanel\_exchange សម្រាប់ផ្ទុកព័ត៌មានរបស់ការបញ្ចូលអត្រាប្តូរប្រាក់
* cpanel\_setting សម្រាប់ផ្ទុកព័ត៌មាននៃការកំណត់ផ្សេងៗ
* event សម្រាប់ផ្ទុកព័ត៌មាននៃព្រឹតិ្តការណ៍ដែលអ្នកប្រើប្រាស់ធ្លាប់បានធ្វើ (User)
* role សម្រាប់ផ្ទុកព័ត៌មាននៃការកំណត់សិទ្ធរបស់អ្នកប្រើប្រាស់ (User)
* rabbit\_agent សម្រាប់ផ្ទុកព័ត៌មានរបស់ភ្នាក់ងារនាំអតិថិជន (Agent)
* rabbit\_customer សម្រាប់ផ្ទុកព័ត៌មានរបស់អតិថិជន
* users សម្រាប់ផ្ទុកព័ត៌មានរបស់អ្នកប្រើប្រាស់ប្រព័ន្ធ (User)
* rabbit\_paymentWebsite សម្រាប់ផ្ទុកព័ត៌មានការបង់ប្រាក់របស់ Website
* rabbit\_website សម្រាប់ផ្ទុកព័ត៌មានរបស់ website
* rabbit\_contract សម្រាប់ផ្ទុកព័ត៌មានការធ្វើការចុះកិច្ចសន្យា
* rabbit\_contractor សម្រាប់ផ្ទុកព័ត៌មានការធ្វើការចុះកិច្ចសន្យារវាងអ្នកទិញ និងអ្នកលក់
* rabbit\_mantenance សម្រាប់ផ្ទុកព័ត៌មានការថែទាំប្រព័ន្ធ
* rabbit\_office សម្រាប់ផ្ទុកពត៌មានដែលទាក់ទងនិងការលក់ប្រព័ន្ធ
* rabbit\_paymentMaintenance សម្រាប់ផ្ទុកពត៌មានការបង់ប្រាក់ថែរទាំប្រព័ន្ធ
* rabbit\_paymentOffice សម្រាប់ផ្ទុកពត៌មានដែលបានធ្វើការបង់ប្រាលក់ទៅតាមការិយាល័យនីមួយៗ
* rabbit\_paymentWebsite សម្រាប់ផ្ទុកពត៌មានដែលបានធ្វើការបង់ប្រាក់ទៅលើការទិញវេបសាយ
* rabbit\_quotation សម្រាប់ផ្ទុកពត៌មានដែលទាក់ទងទៅនិងការសាក់សួរពត៌មាន
* rabbit\_service សម្រាប់ផ្ទុកពត៌មានដែលទាក់ទងទៅនឹងសេវាកម្ម

## ៤.៣. ការសាងសង់ Web Application

### ៤.៣.១ ទម្រង់នៃការបង្កើត Web Application

ក្នុងការបង្កើត Web Application យើងខ្ញុំមាន Structure ដូចខាងក្រោម៖

* .meteor: ជាថតឯកសារ (Folder) ផ្ទុកនូវ source code files សំខាន់ៗរបស់ framework សម្រាប់ដំណើរការ project។
* cpanel: ជាថតឯកសារ (Folder) ផ្ទុកនូវ source code file សំខាន់ៗ និង​ ជា​ Main Package សម្រាប់គ្រប់គ្រង Setting ផេ្សងៗទៀត ដូចជា៖ Exchange, Branch, Company, User, Setting, Backup, Restore
* rabbit: ជាថតឯកសារ (Folder) ផ្ទុកនូវ source code file សំខាន់ៗ សម្រាប់ Project របស់យើង។
* packages: សម្រាប់ផ្ទុកនូវ packages ចាំបាច់ផ្សេងៗដែលប្រើប្រាស់ជាមួយ project (Package ដែលយើងបង្កើតលើ Local) ។
* public: សម្រាប់ផ្ទុកនូវ File ផ្សេងៗសម្រាប់ហៅយកទៅប្រើប្រាស់មានដូចជា៖ រូបភាព ពុម្ភអក្សរ ...​ (សម្រាប់ប្រើបានទាំង cpanel និង rabbit) ។

### ៤.៣.២ .meteor

នៅក្នុងថតឯកសារ (folder) .meteor មានថតឯកសារ (folder) ផ្សេងៗទៀតដូចជា៖ Local, Platforms, Release​​ និង Version ផងដែរ។

local: ជា folder សម្រាប់ផ្ទុក Database និង package ដែលបានដំឡើងហើយ ។

packages: គឺជា File ផ្ទុករាល់ឈ្មោះ Packages ដែលប្រើបា្រស់​ក្នុង Project (ទាំង online និង offline) ។

platforms: ​​ គឺជា file ផ្ទុកព៍តមានជំនាន់របស់ Platforms ដែលបានប្រើប្រាស់លើ Project ។ ឧទាហរណ៍: Windows, Linux, Mac OSX, Android និង IOS ។

release: គឺជា file ផ្ទុកព៍តមានជំនាន់របស់ Framework ។

version: គឺជា file ផ្ទុកព៍តមានជំនាន់របស់ package ដែលប្រើបា្រស់ក្នុង project ។

### ៤.៣.៣ cpanel

នៅក្នុងថតឯកសារ (folder) cpanel មានផ្ទុកនៅថតឯកសារ (folder) ផ្សេងៗទៀតដូចជា៖ client, common, server។

#### **៤.៣.៣.១ client**

សម្រាប់ផ្ទុកនូវ File ដែលត្រូវប្រើនៅលើ client។ ហើយវាមានផ្ទុកថតឯកសារផ្សេងៗទៀតដូចជា៖ app, compatibility, css, templates។

* app: នៅក្នុងថតឯកសារ (folder) app មានផ្ទុកនៅថតឯកសារ (folder) ផ្សេងៗទៀតដូចជា៖ config, libs, list, startup, theme។
* libs: សម្រាប់ផ្ទុក function ផ្សេងៗដែល Configure ទៅលើ plug in (អាចដំណើរការបានតែលើ Client) ។
* lists: សម្រាប់ផ្ទុក function ប្រើបា្រស់ជាមួយ select box
* startup: សម្រាប់ផ្ទុក function មួយចំនួនដែលទាមទារអោយដំណើរការមុនពេល Project ដំណើរការចប់ ។
* theme: សម្រាប់ផ្ទុក function ដើម្បីអោយ Project ទាំងមូលស្គាល់ file css។
* compatibility: សម្រាប់ផ្ទុករាល់ plug in ទាំងអស់ដែលមិនទាន់មានជា Package ។
* stylesheet: ផ្ទុកនៅរាល់ប្រភេទ file css
* routers: ផ្ទុកនៅរាល់ប្រភេទ file js សម្រាប់តំណរភ្ចាប់ (link) ពីទំព័រមួយទៅទំព័រមួយទៀត។
* templates: ផ្ទុកនៅរាល់ប្រភេទ file html​ និង js សម្រាប់បង្ហាញ​ form interface ។

#### **៤.៣.៣.២​ common**

common សម្រាប់ផ្ទុកនូវ File ដែលត្រូវប្រើទាំងនៅលើ client និង server។ ហើយក្នុងនោះវាមានថតឯកសារមួយចំនួនទៀតដូចជា៖ collections, lib និង tabulars ។

* collection: ផ្ទុកនៅ file database (collection)។
* lib: សម្រាប់ផ្ទុក function ផ្សេងៗដែល Configure ទៅលើ plug in (អាចដំណើរការបានទាំងលើ Client និង Server) ។
* tabulars: ផ្ទុកនៅ file tables ដែលត្រូវទាញយកទៅប្រើប្រាស់ជាមួយនឹង template។

#### **៤.៣.៣.៣​ server**

server សម្រាប់ផ្ទុកនូវ File ដែលត្រូវប្រើនៅលើ server។

* app: ​ ផ្ទុក function ផ្សេងៗដែលសម្រាប់កំណត់សិទ្ធឲ្យ User ។
* collection\_hook: សម្រាប់ផ្ទុករាល់ File js ប្រើបា្រស់សម្រាប់មុនពេល Insert Remove ឬក៏ Update ចូលទៅ Database (collection)។
* methods: សម្រាប់ផ្ទុក function ផ្សេងៗដែលសម្រាប់ហៅទៅប្រើនៅលើClient ។
* publisher: ផ្ទុក function សម្រាប់ publish data ទៅ client ។
* startup/fixtures: សម្រាប់ផ្ទុក រាល់ File js ប្រើបា្រស់សម្រាប់ Insert Data ចូលទៅ Database Automatic ។

### ៤.៣.៤ rabbit

នៅក្នុងថតឯកសារ (folder) rabbit នេះមានផ្ទុកនៅថតឯកសារ (folder) ផ្សេងៗដូចទៅនឹង cpanel ដែរតែវាសម្រាប់ប្រើតែក្នុង rabbit នេះប៉ុណ្ណោះ។

## ៤.៤. ការបង្កើត User Interface

មុននឹងឈានដល់ដំណើរការរបស់ប្រព័ន្ធគ្រប់គ្រងរបស់ក្រុមហ៊ុន​​ Rabbit IT Solution ក្រុមយើងខ្ញុំបានធ្វើការរៀបចំកម្មវិធីដែលមានលក្ខណៈជា Internal Form ដែលមាន Main Form និង Child Form ជាច្រើនដែលមានមុខងារផ្សេងៗគ្នា។

ក្នុងការ Design User Interface យើងអាចចែកចេញជាពីរផ្នែកគឺ៖

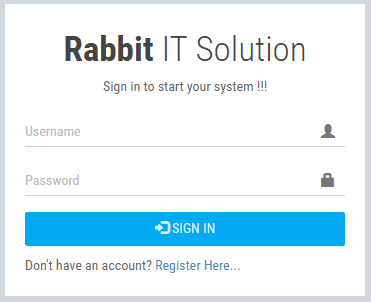
* **Form Design:** ជា Interface រវាង User និង Database ដើម្បីឲ្យការគ្រប់គ្រងទិន្នន័យដែលមានក្នុង Database នេះបានដោយងាយស្រួល។
* **Report Design:** ជា Interface ដែល User អាចប្រើប្រាស់សម្រាប់មើល និងទាញយករបាយការណ៍ ដើម្បីទទួលបានលទ្ធផល និងត្រូវបានគេបោះពុម្ភមកក្រៅ។

### ៤.៤.១ ការរចនា Form

1. Form Login

មុននឹងអាចចូលទៅប្រើប្រាស់បានយើងត្រូវធ្វើការ login ជាមុនសិន។

រូបភាព ៤.១ Login Form

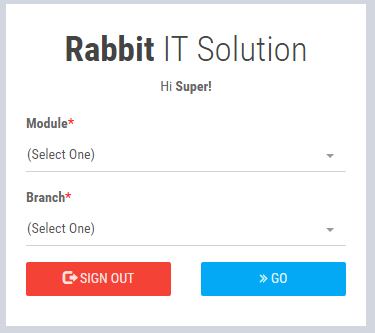
****

រូបភាព ៤.១ Register Form

1. Module Form

បន្ទាប់ពីបានបញ្ចូលឈ្មោះនិង ពាក្យសម្ងាត់បានត្រឹមត្រូវហើយ វានិងបង្ហាញផ្ទាំងមួយទៀតដើម្បីឲ្យយើងរើស Module និងសាខា។

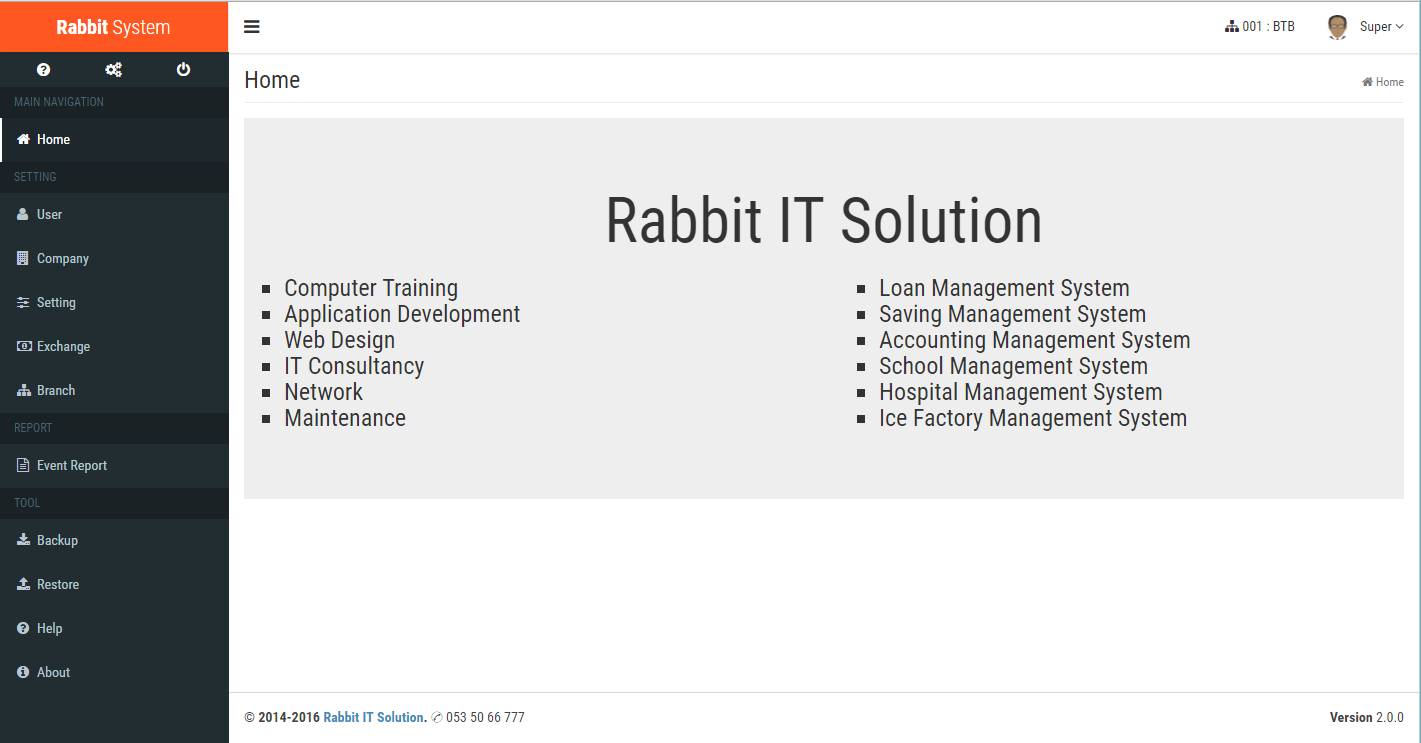
រូបភាព ៤.៣ Module Form



1. Main Page

បន្ទាប់ពីរើស Module និងសាខា បានត្រឹមត្រូវហើយ នោះយើងអាចប្រើប្រាស់ប្រព័ន្ធនេះបានទៅតាមសាខា និង Module ដែលបានរើស។ ហើយវានឹងប​ង្ហាញ Main Page ដែលមានទំរង់ដូចខាងក្រោមៈ

រូបភាព ៤.៤ Main Page

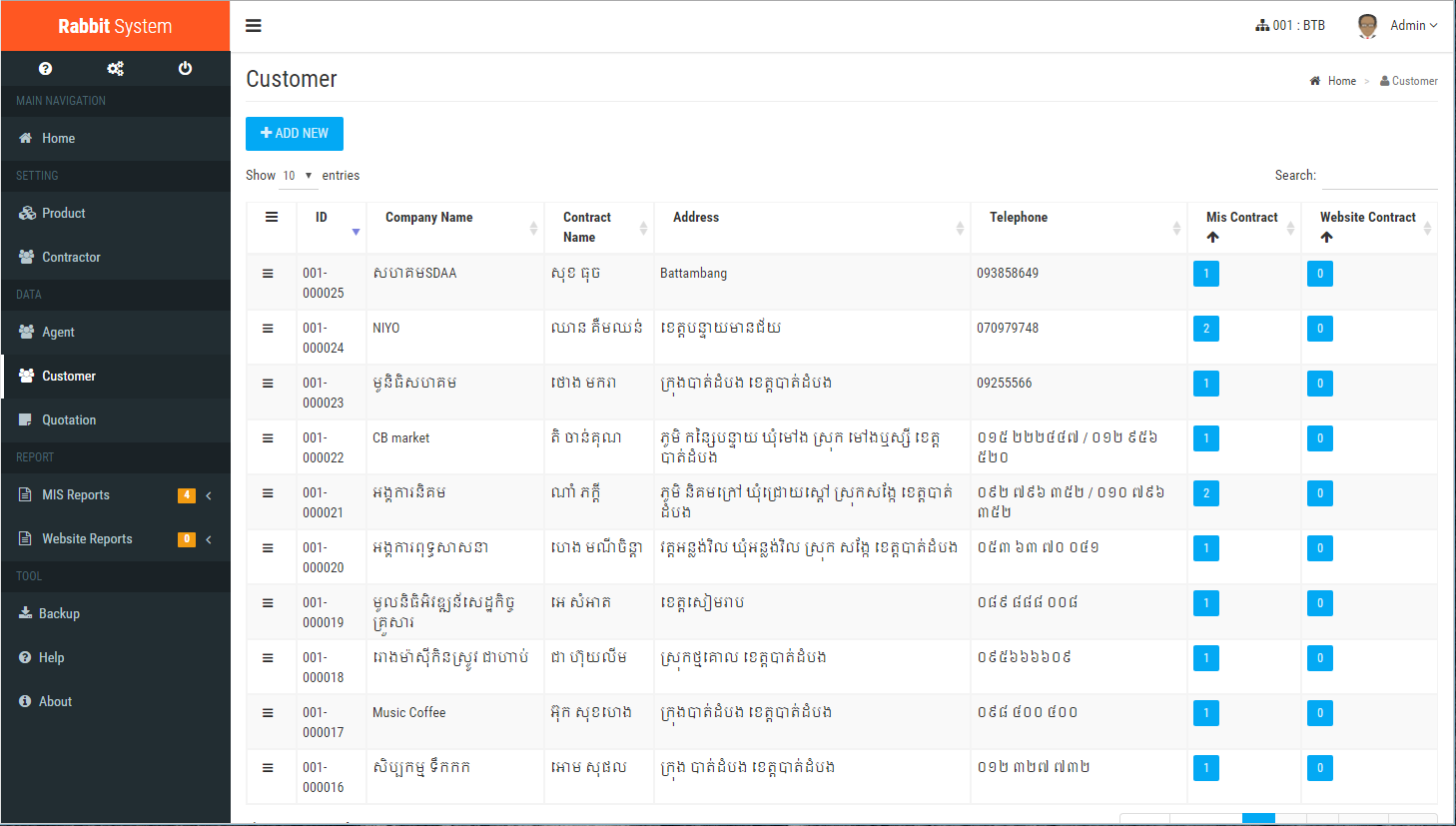


Page នេះសម្រាប់ដំណើរការរាល់សកម្មភាពរបស់ ប្រព័ន្ធទាំងមូលតែម្តងហើយ ក្នុង Page នេះអ្នកប្រើប្រាស់អាចប្រើបាននូវ Page និងForm ផ្សេងៗទៀតទៅតាមតំរូវការ។ ដែលនូវផ្នែកខាងឆ្វេង (Sidebar) បង្ហាញពីមឺនុយ, ផ្នែកខាងស្តាំសម្រាប់បង្ហាញ Content និងផ្នែកខាងលើ (Navbar) បង្ហាញពី ឈ្មោះអ្នកកំពុងប្រើប្រាស់ (User) និងសាខា (Branch) ។

1. Customer

ជា Page សម្រាប់បង្ហាញពីព័ត៌មានរបស់អតិថិជន។

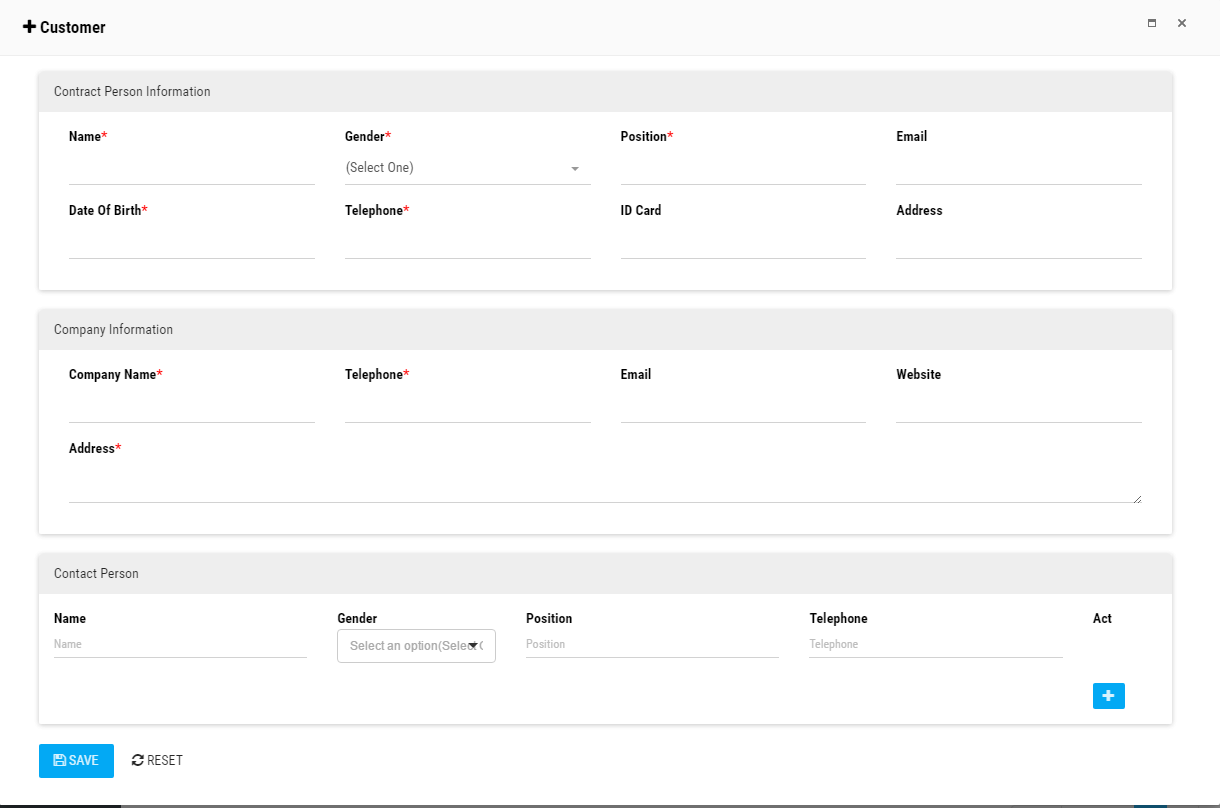
រូបភាព ៤.៥ Customer



* 1. Add Customer Form

Form នេះសម្រាប់ធ្វើការបញ្ចូលនូវព័ត៌មានអតិថិជនថ្មី។

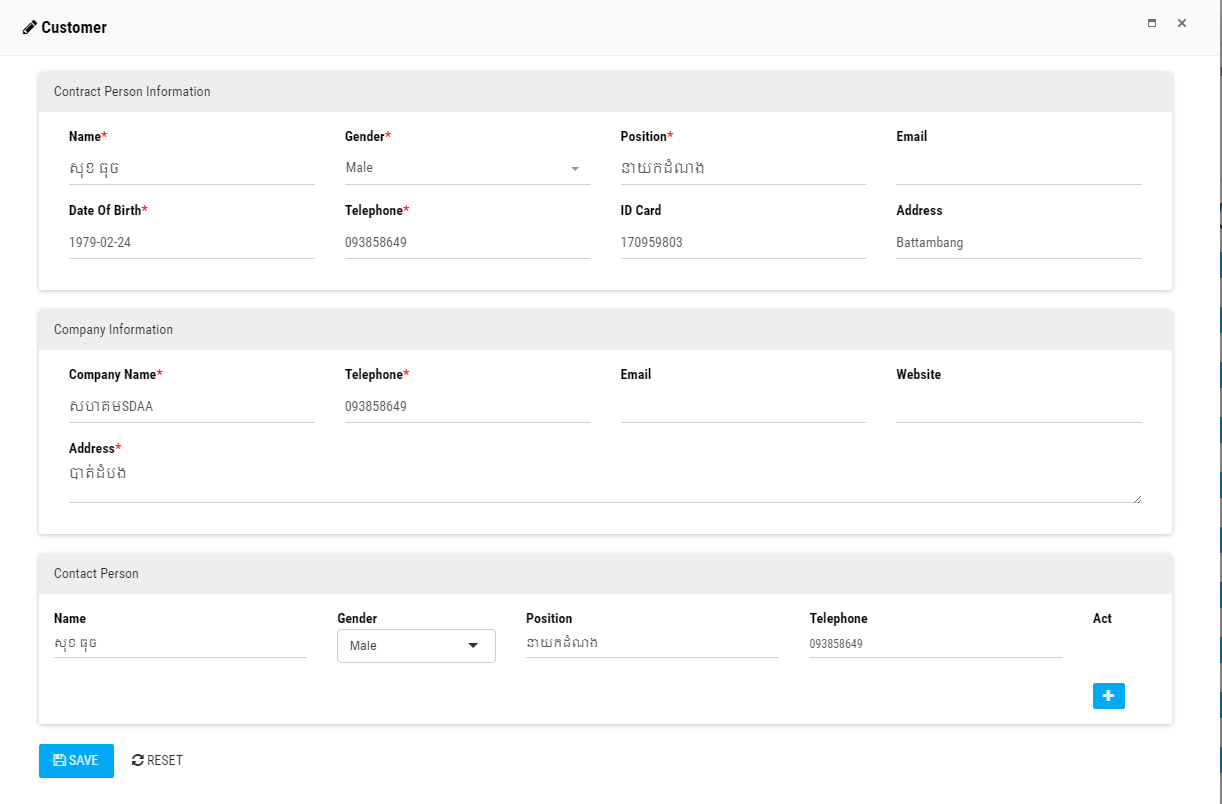
រូបភាព ៤.៦ Add Customer Form

****

* 1. Edit Customer Form

Form នេះសម្រាប់ធ្វើការកែប្រែនូវព័ត៌មានអតិថិជនដែលធ្លាប់បាន Add

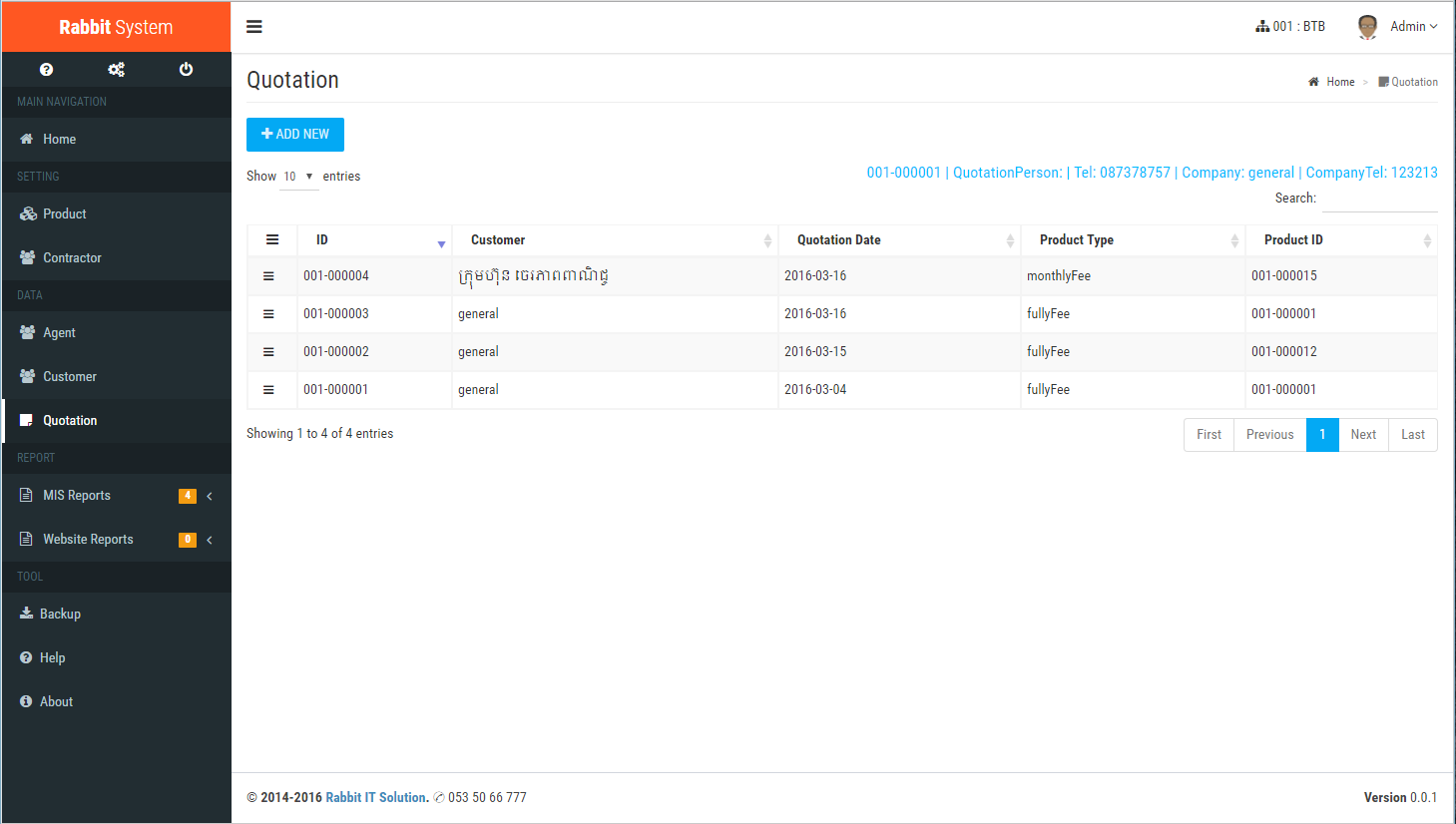
រូបភាព ៤.៧ Edit Customer Form



1. Quotation

Page នេះសម្រាប់បង្ហាញ ។

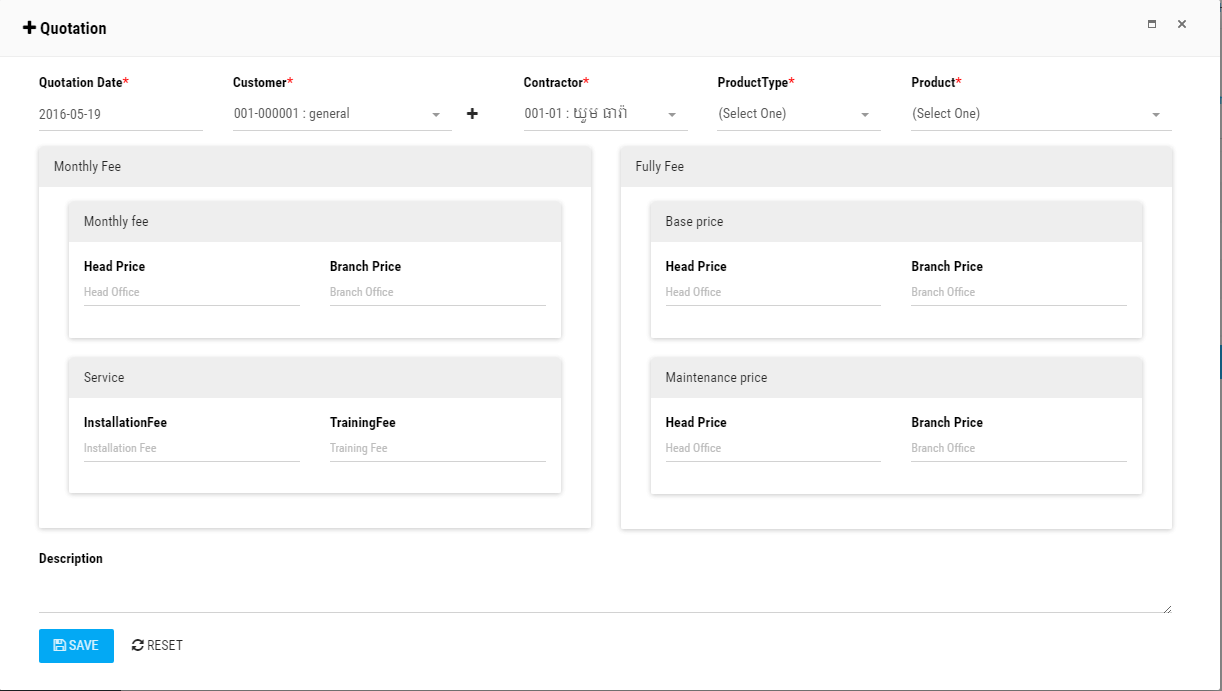
រូបភាព ៤.៨ Quotation



* 1. Add Quotation Form

Form នេះសម្រាប់ធ្វើការបញ្ចូលនូវព័ត៌មានអតិថិជនថ្មី។

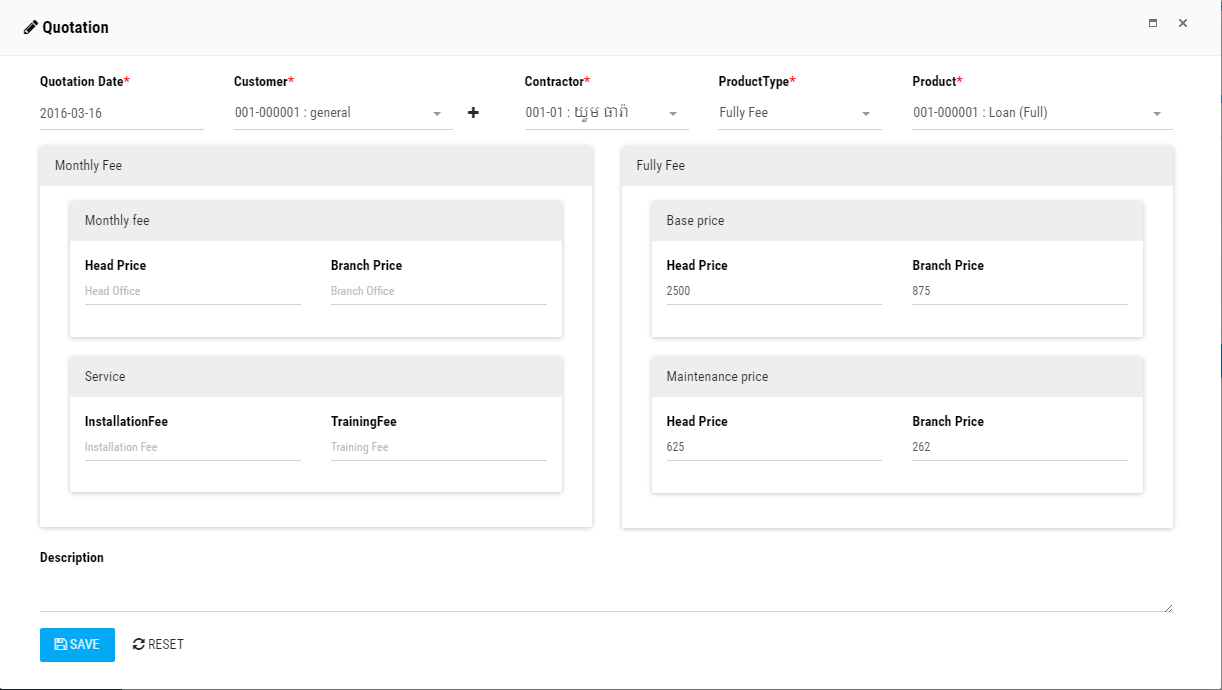
រូបភាព ៤.៩ Add New Quotation



* 1. Edit Quotation Form

Form នេះសម្រាប់ធ្វើការកែប្រែនូវព័ត៌មានអតិថិជនដែលធ្លាប់បាន Add

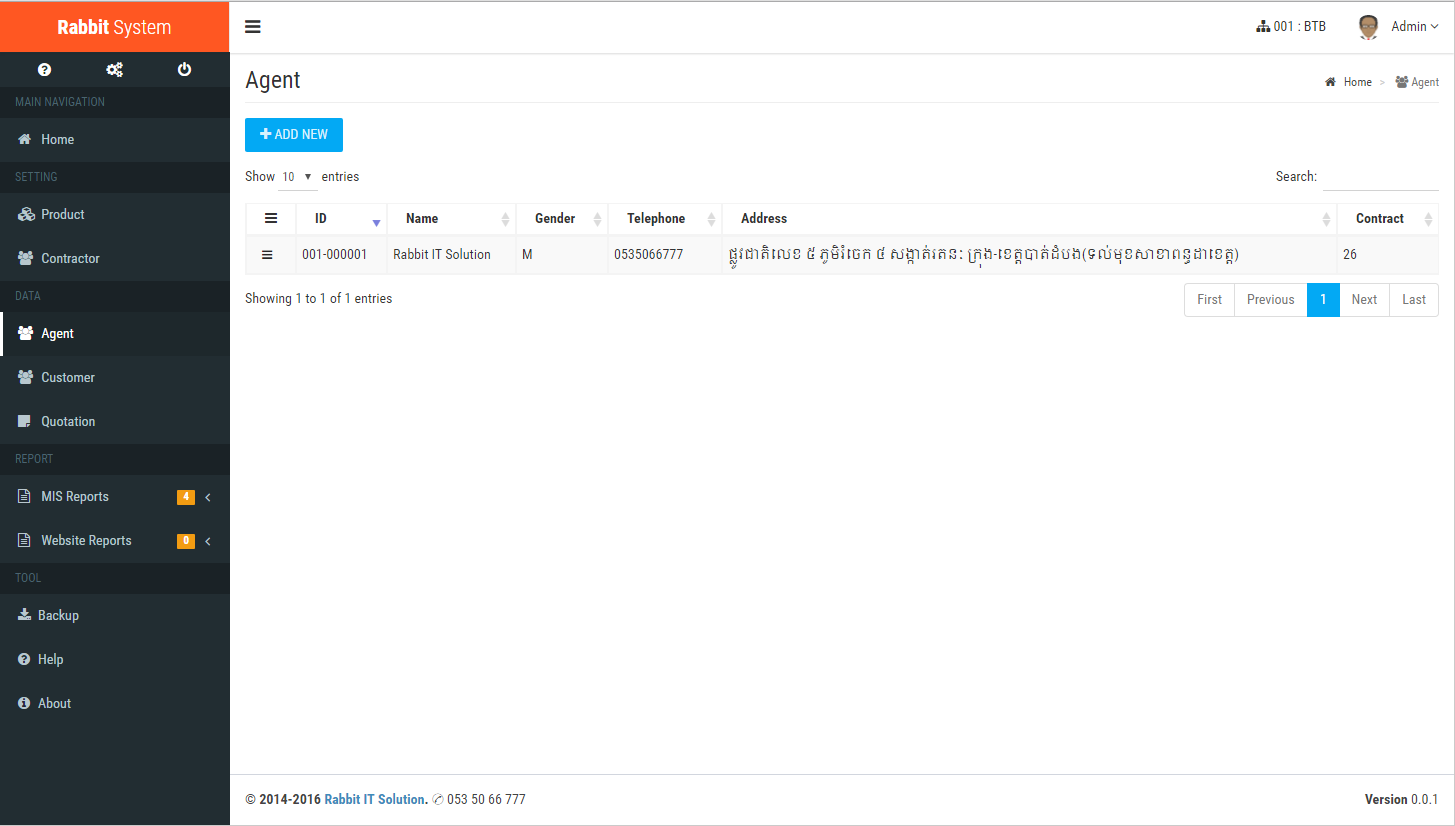
រូបភាព ៤.១០ Edit Quotation



1. Agent

Page នេះសម្រាប់បង្ហាញពីព័ត៌មានរបស់អ្នកនាំអតិថិជន។

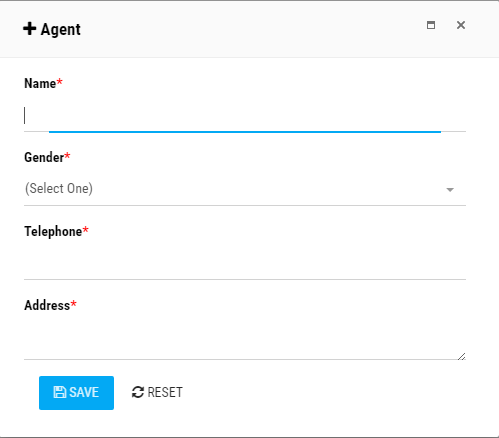
រូបភាព ៤.១១ Agent

****

* 1. Add Agent Form

Form នេះសម្រាប់ធ្វើការបញ្ចូលនូវព័ត៌មានអ្នកនាំអតិថិជនថ្មី។

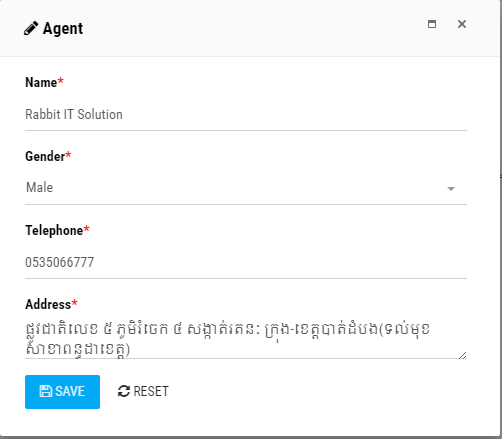
រូបភាព ៤.១២ Add New Agent



* 1. Edit Agent Form

Form នេះសម្រាប់ធ្វើការកែប្រែនូវព័ត៌មានអ្នកនាំអតិថិជនដែលធ្លាប់បាន Add។

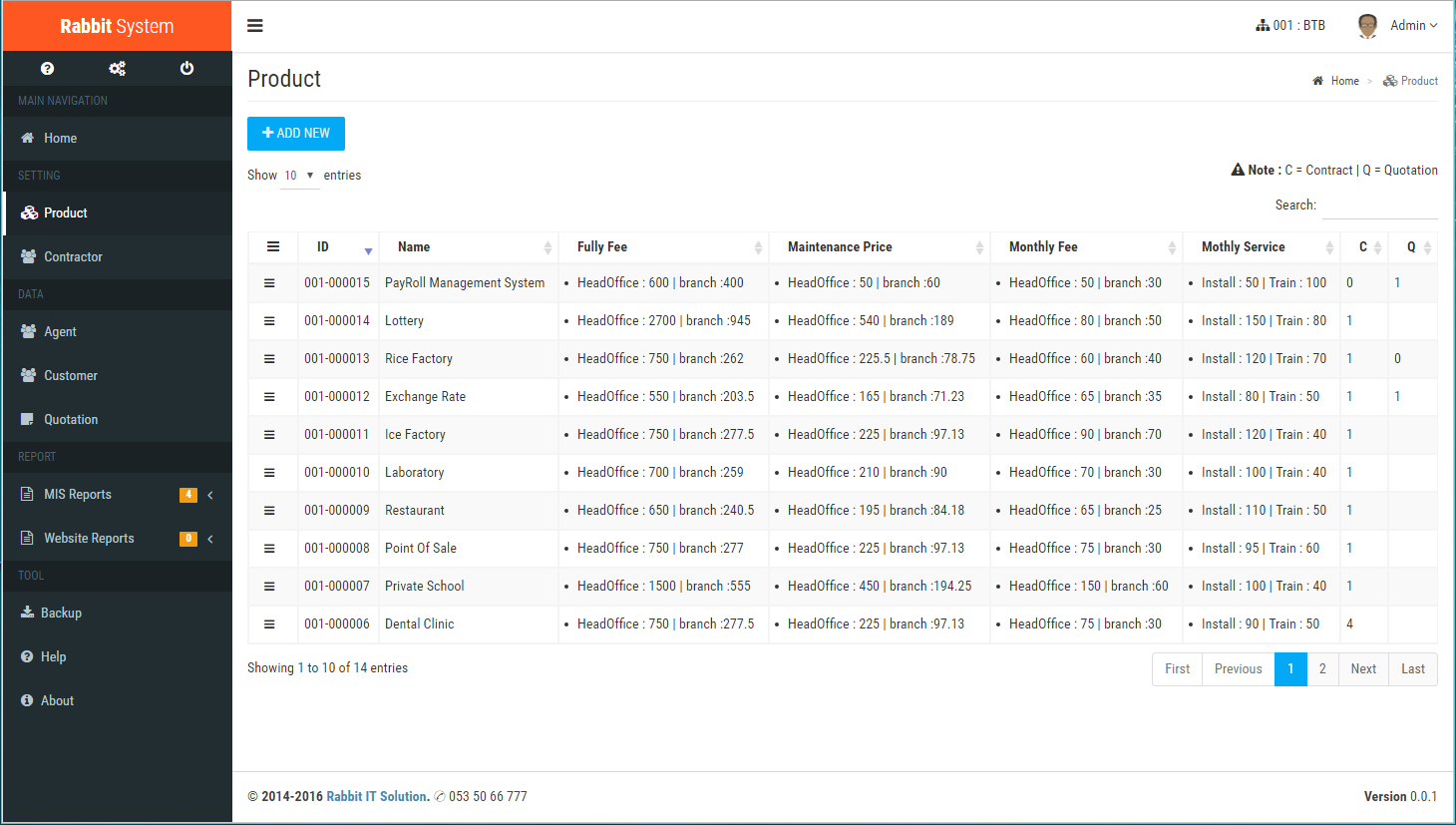
រូបភាព ៤.១៣ Edit Agent



1. Product

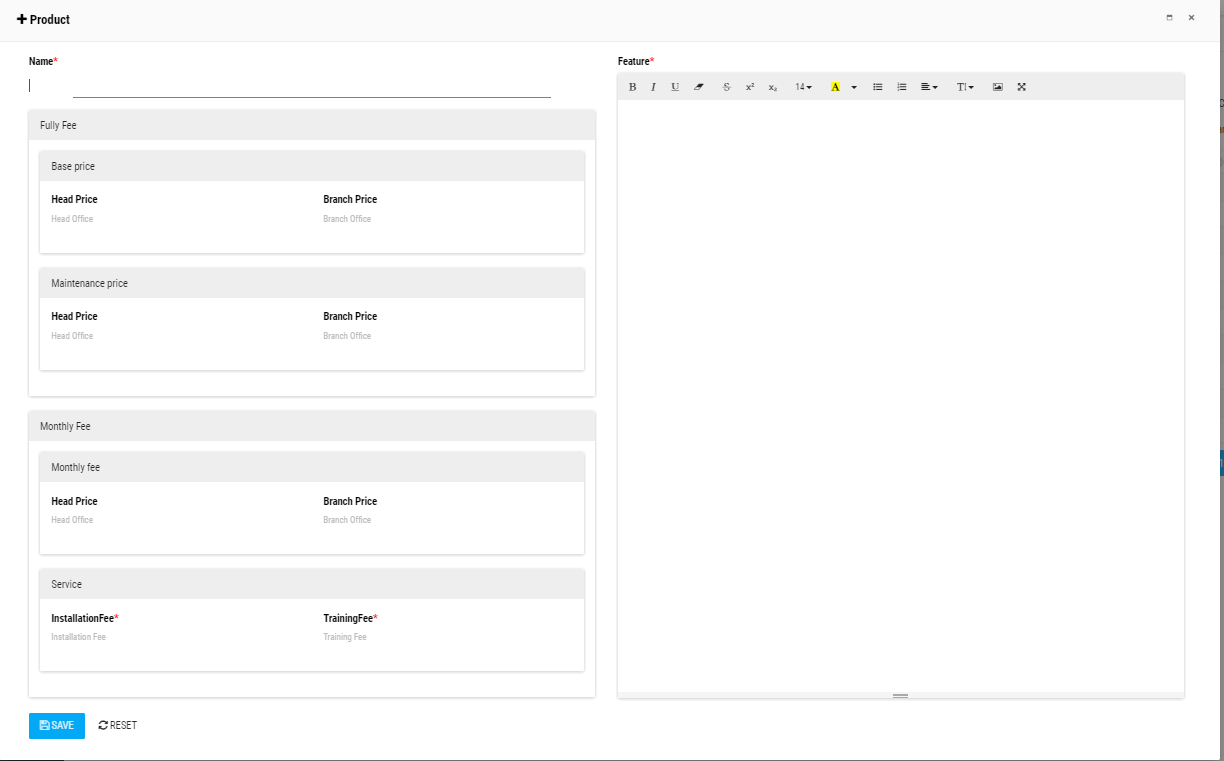
Page នេះប្រើសម្រាប់បង្ហាញពីផលិតផលដែលក្រុមហ៊ុនមាន។

រូបភាព ៤.១៤ Product



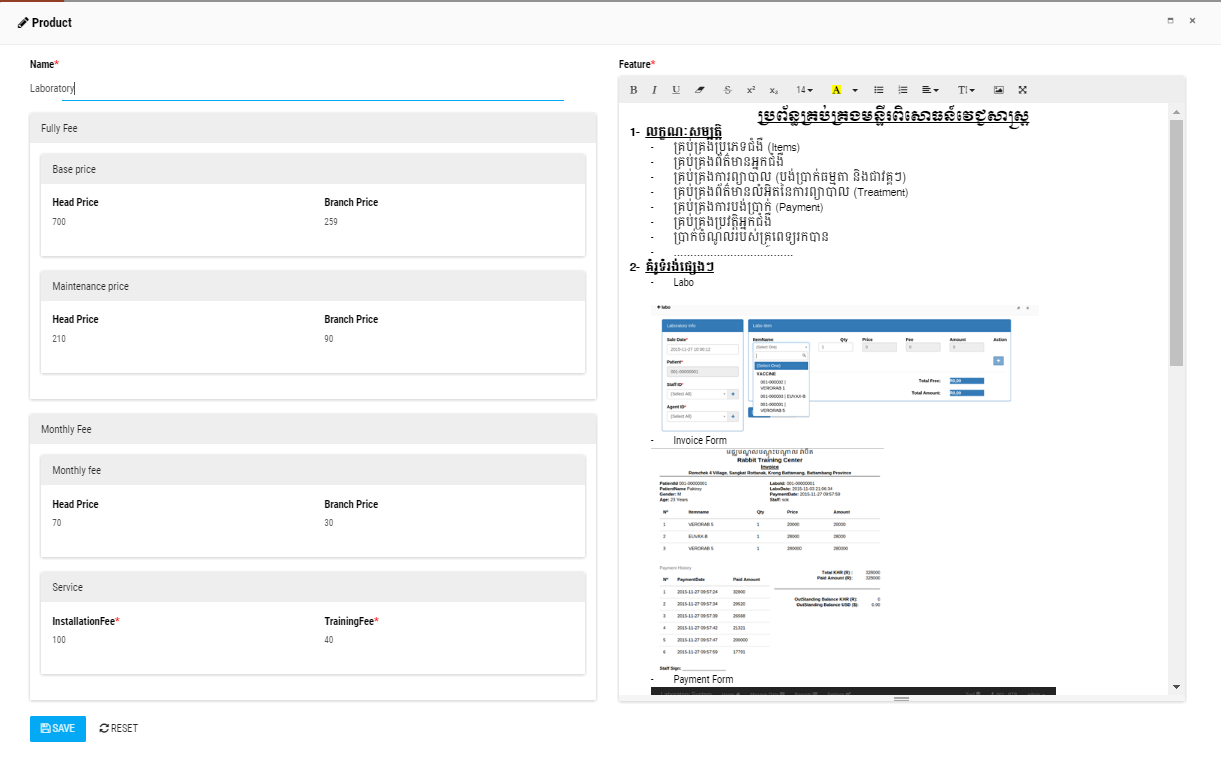
* + 1. Add Product form

រូបភាព ៤.១៥ Add New Product



* + 1. Edit Product form

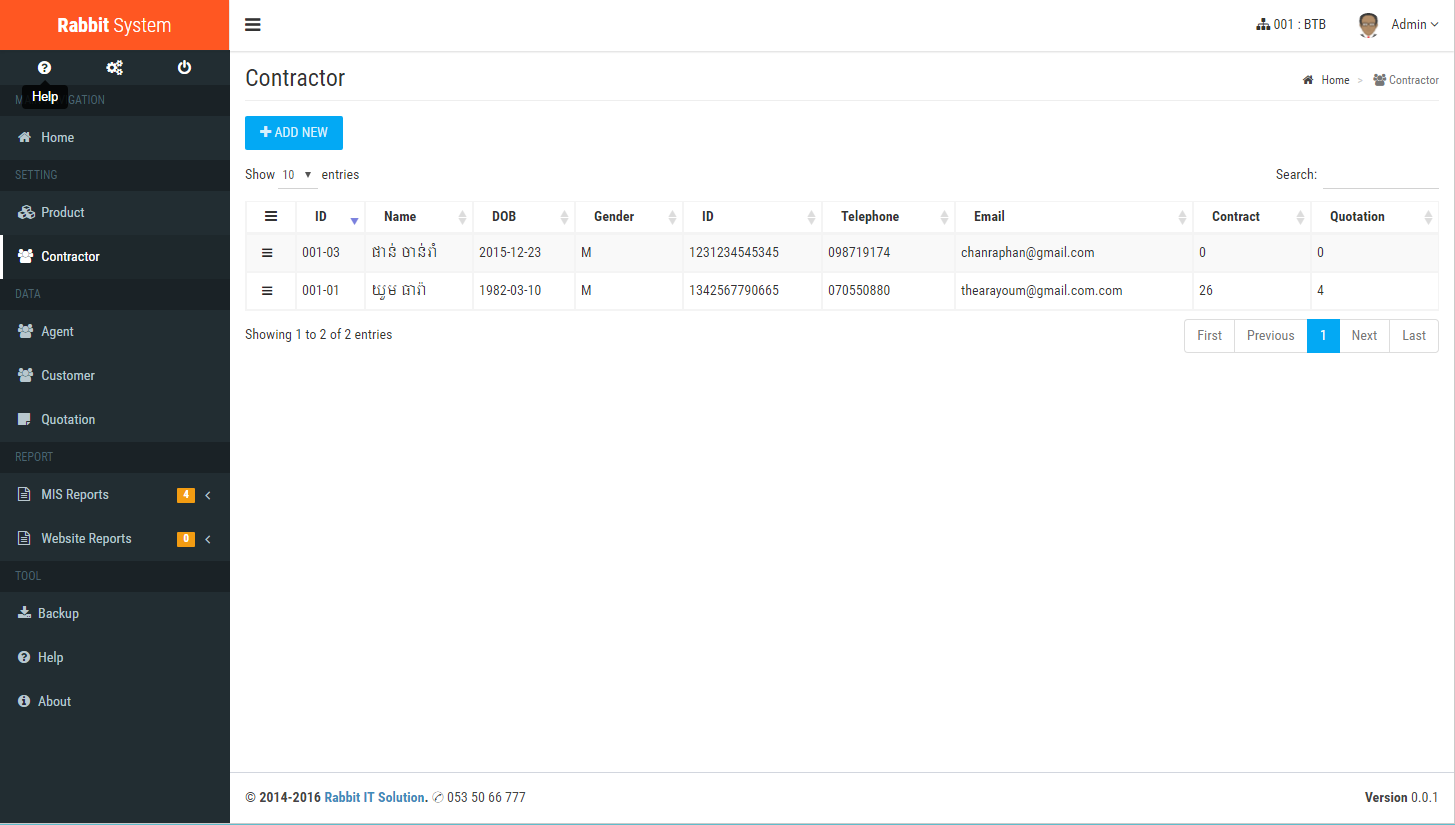
រូបភាព ៤.១៦ Edit Product



1. Contractor

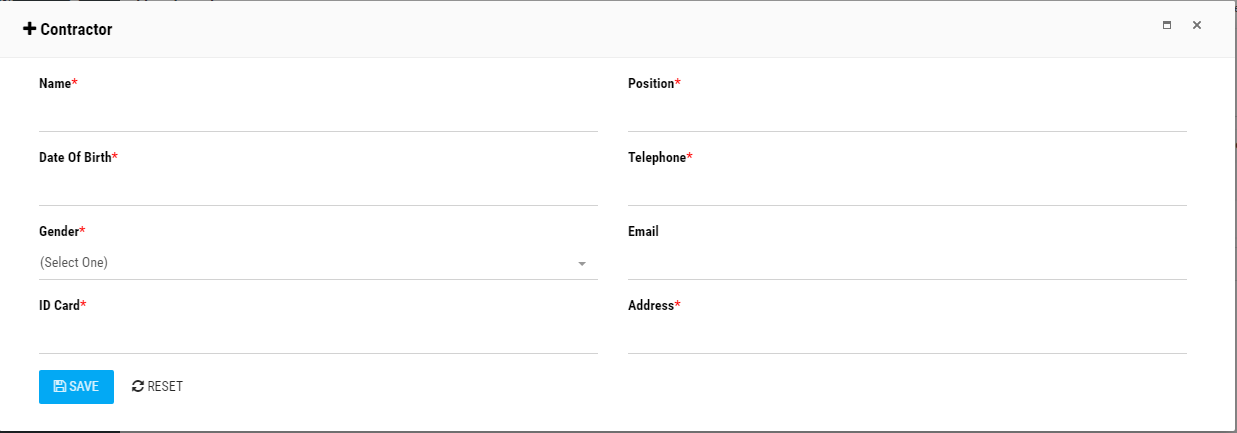
Page នេះប្រើសម្រាប់បង្ហាញពីព័ត៌មានអ្នកចុះកិច្ចសន្យា។

រូបភាព ៤.១៧ Contractor



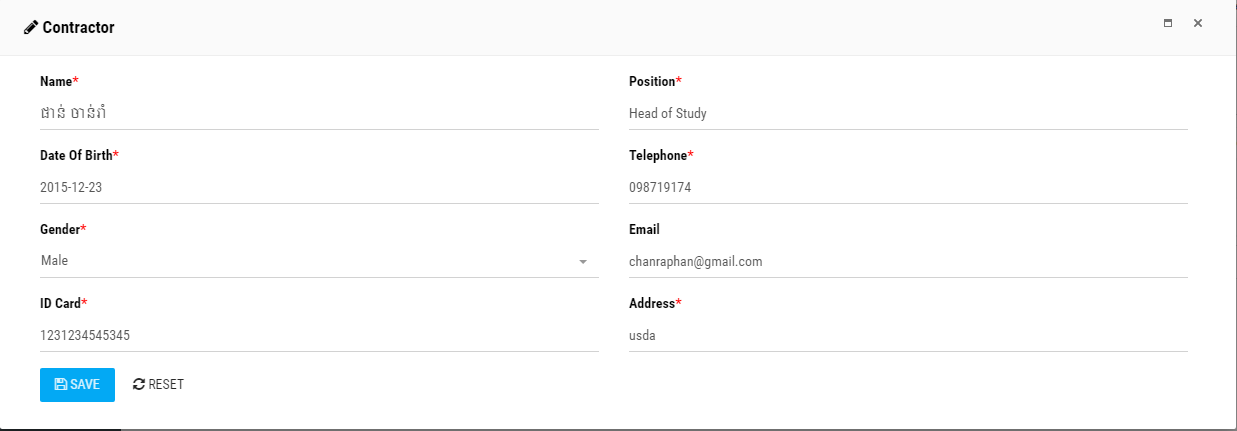
1. Add Contractor form

រូបភាព ៤.១៨ Add New Conctrator



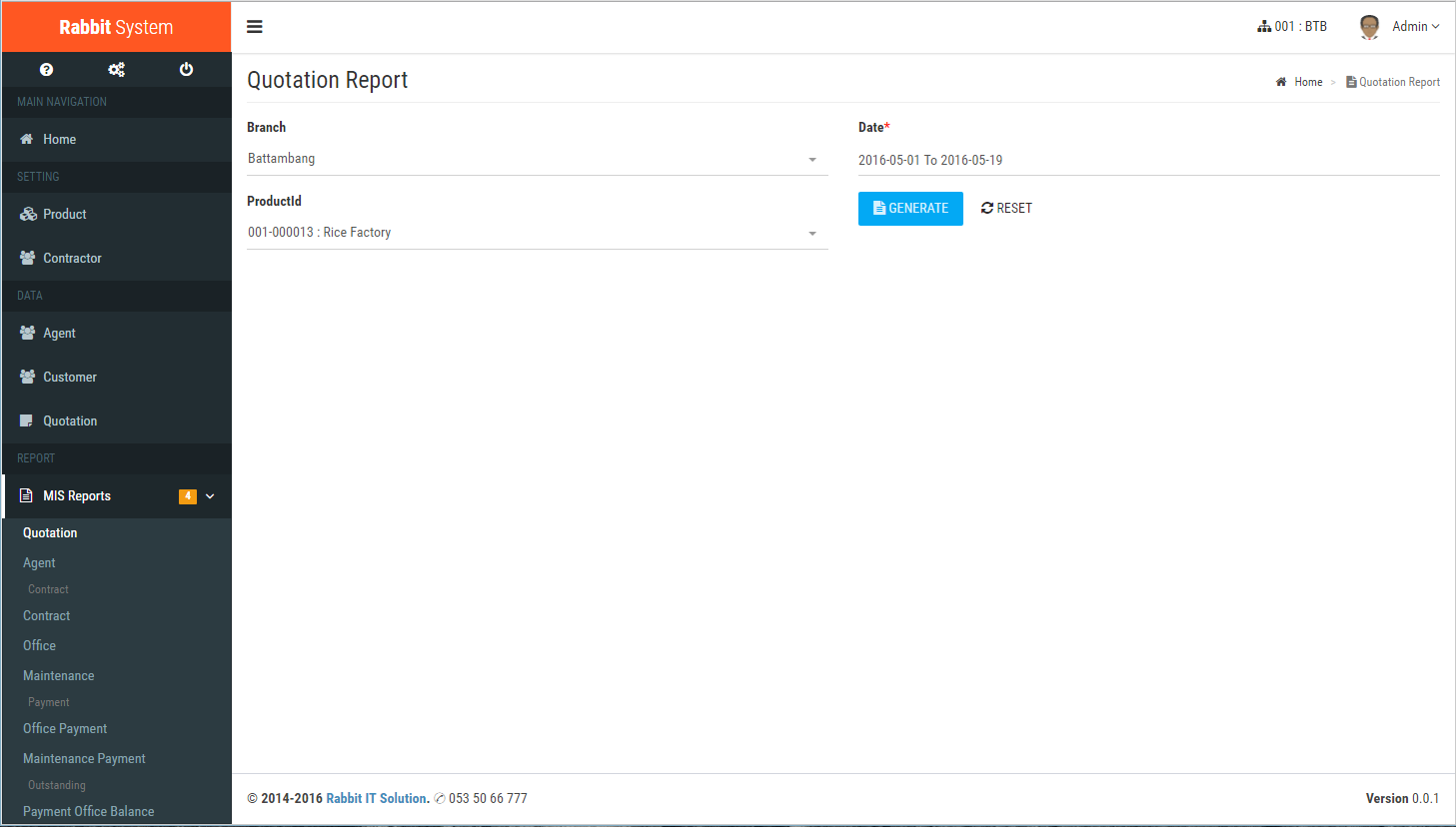
1. Edit Contractor form

រូបភាព ៤.១៩ Edit Contractor



1. MIS Report
   1. Quotation

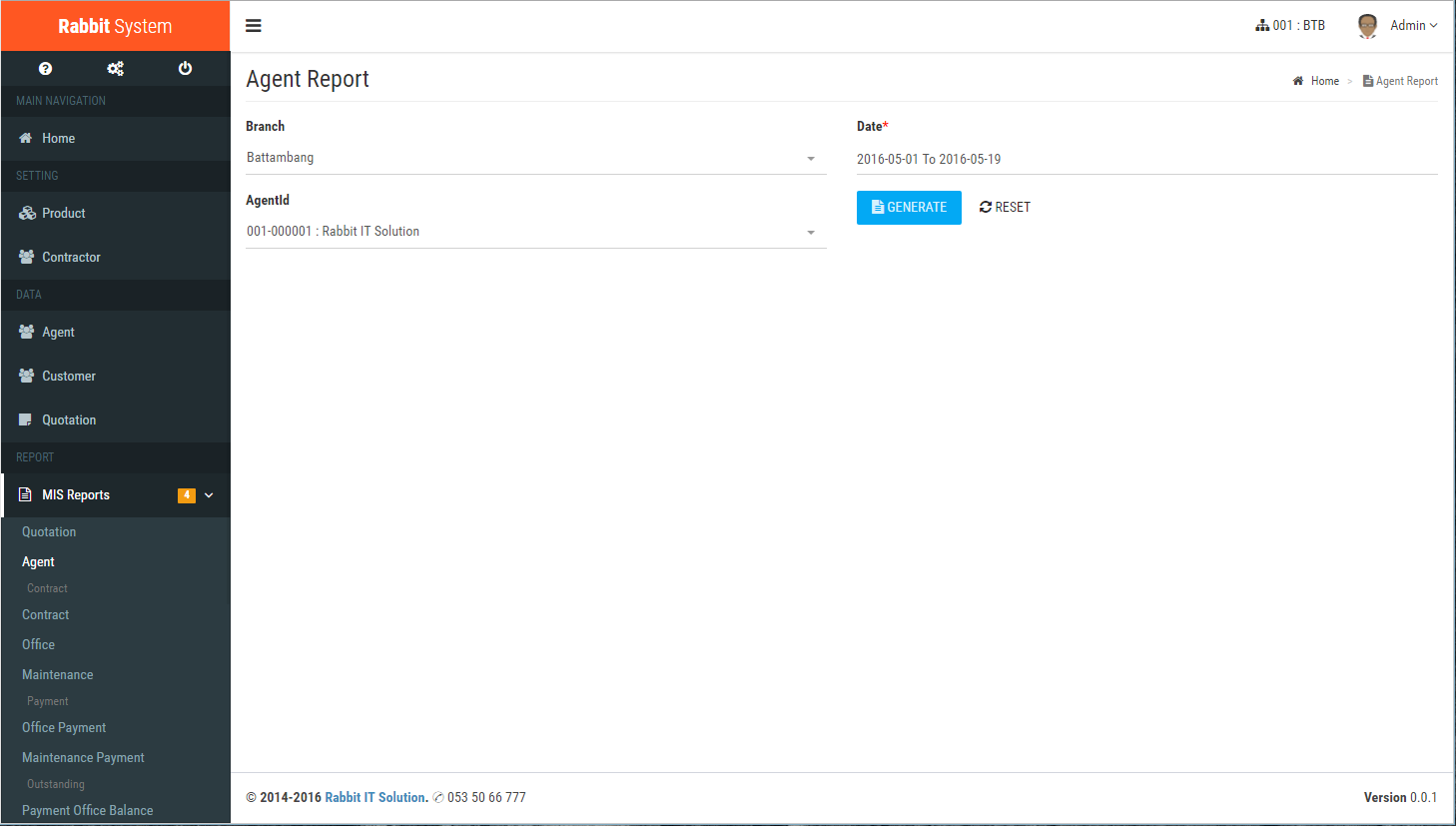
រូបភាព ៤.២០ Quotation Report



* 1. Agent

ជារបាយការណ៍របស់អ្នកនាំអតិថិជន។

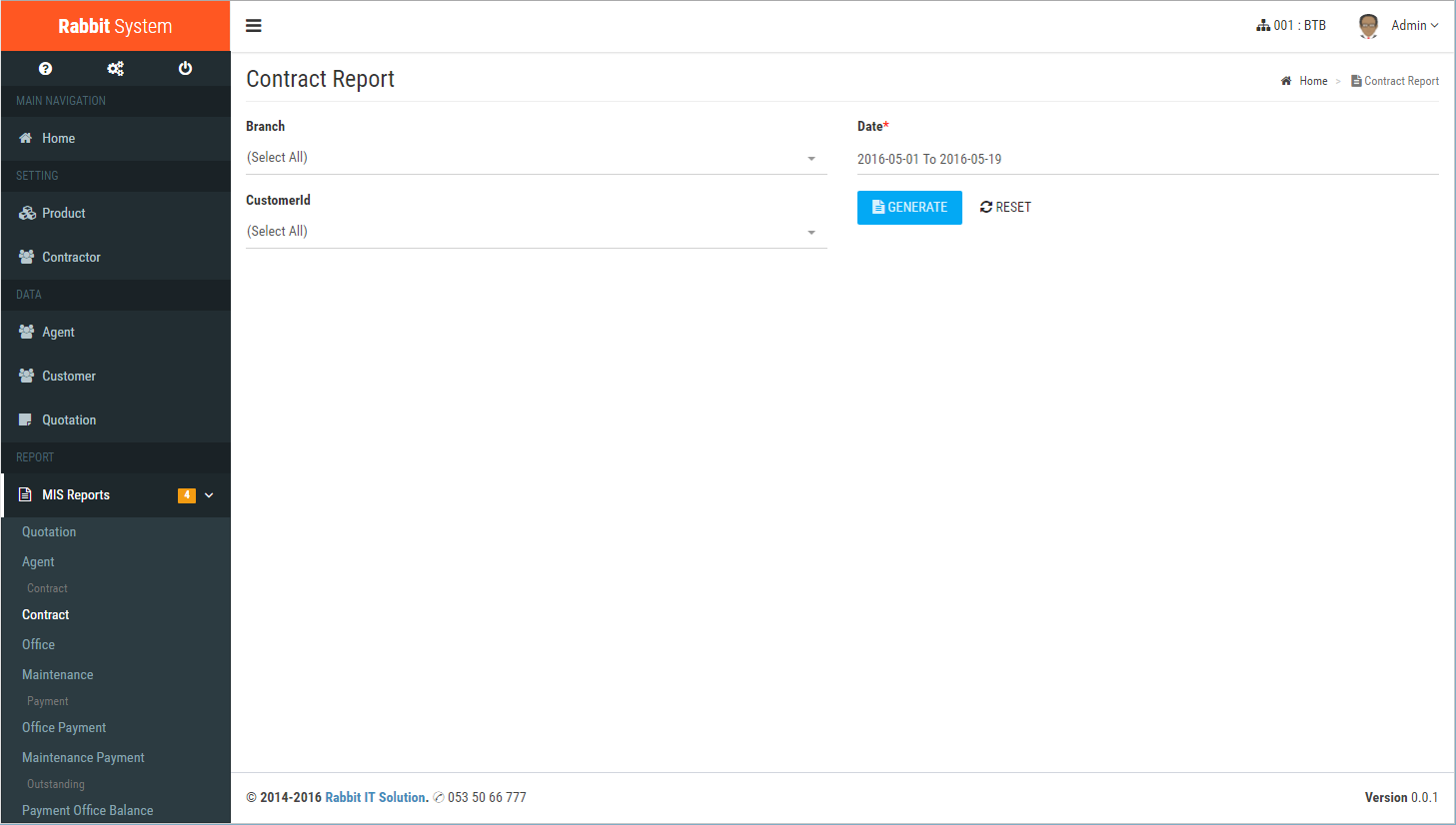
រូបភាព ៤.២១ Agent Report



* 1. Contract

ជារបាយការណ៍នៃការចុះកិច្ចសន្យា។

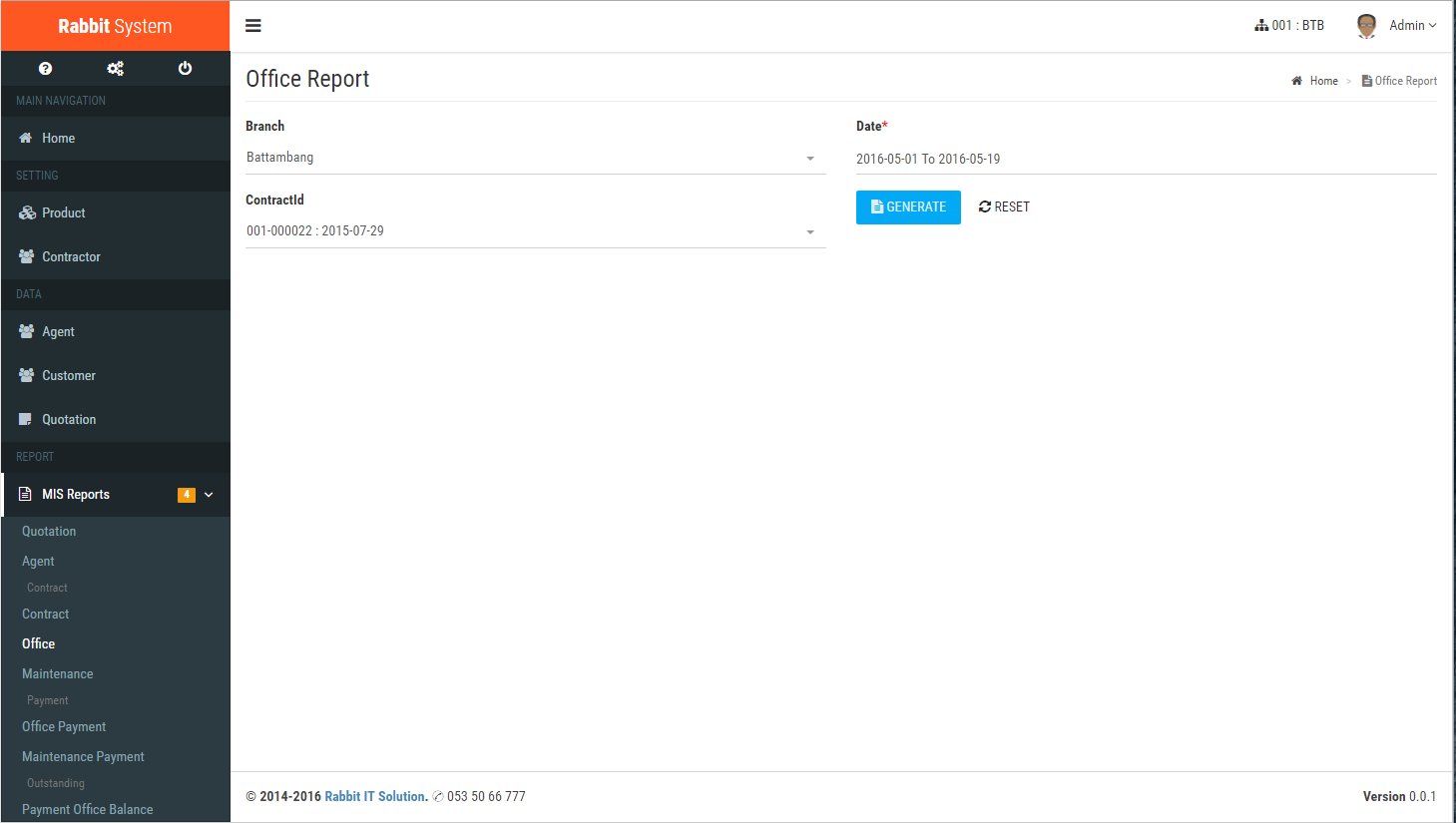
រូបភាព ៤.២២ Contract Report



* 1. Office

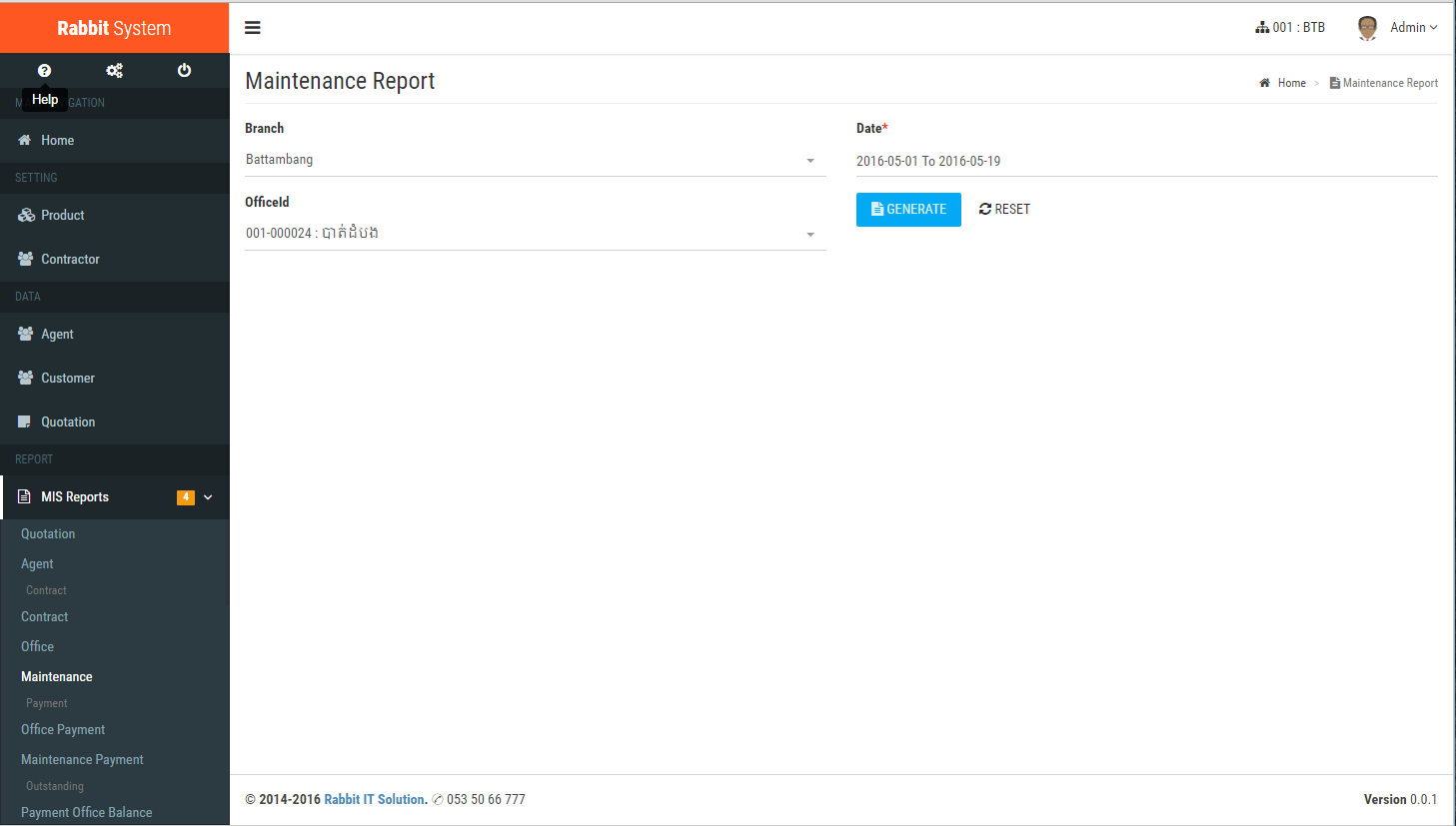
ជារបាយការណ៍នៃការទិញ Office

រូបភាព ៤.២៣ Office Report



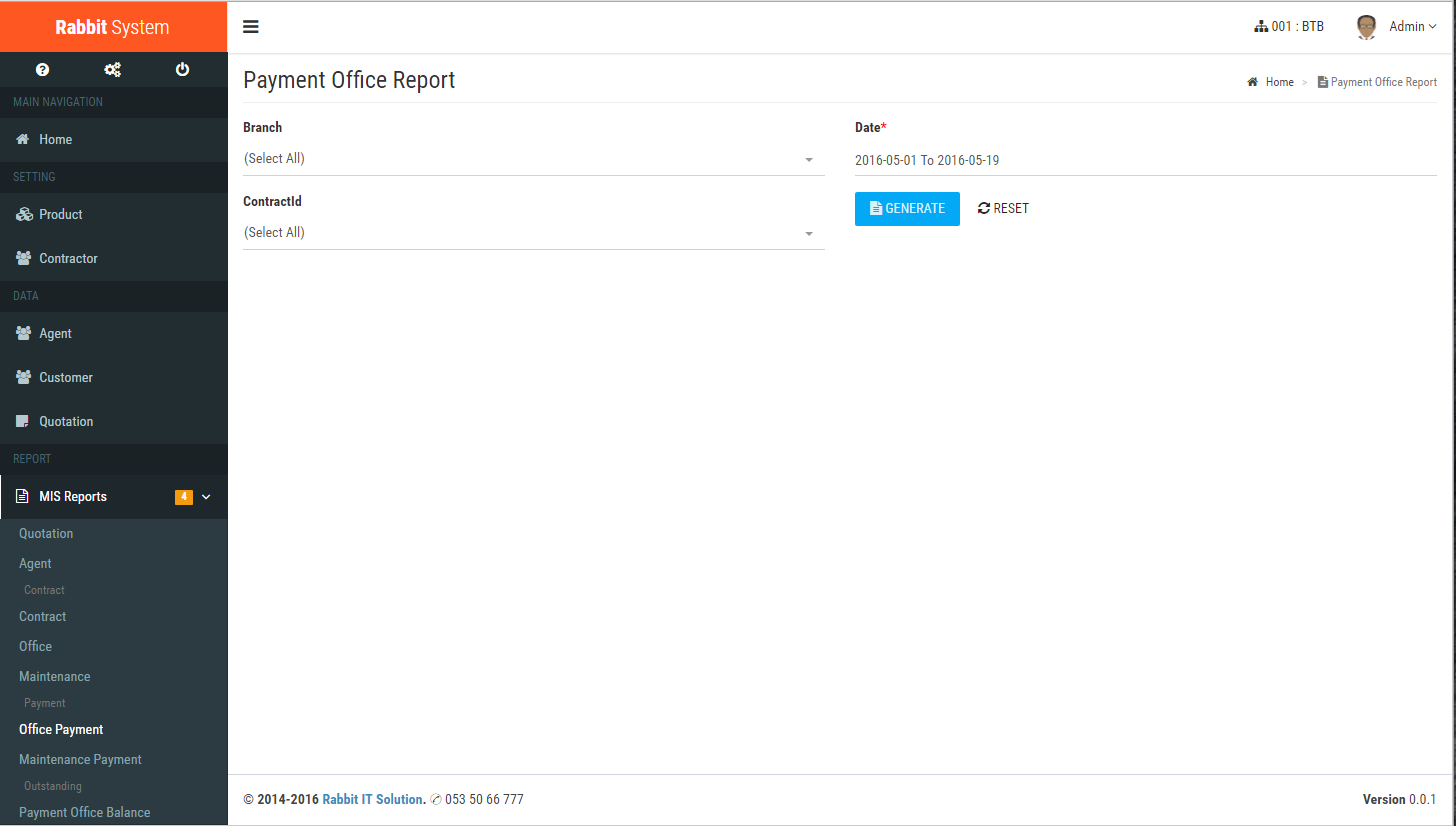
* 1. Maintenance

ជារបាយការណ៍នៃការទិញការថែទាំ។



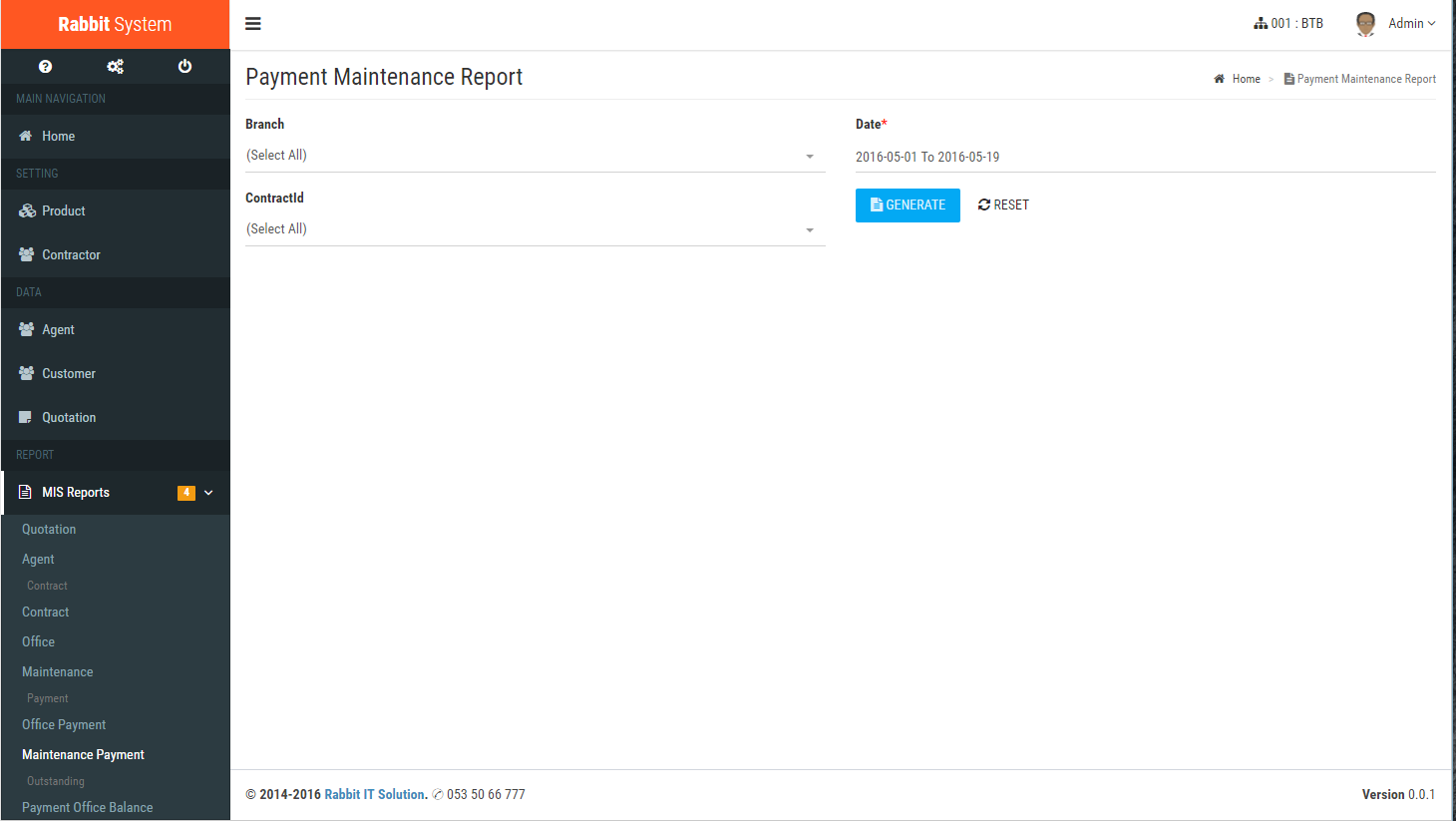
* 1. Office Payment

ជារបាយការណ៍នៃការបង់ប្រាក់ទៅលើការទិញ Office។



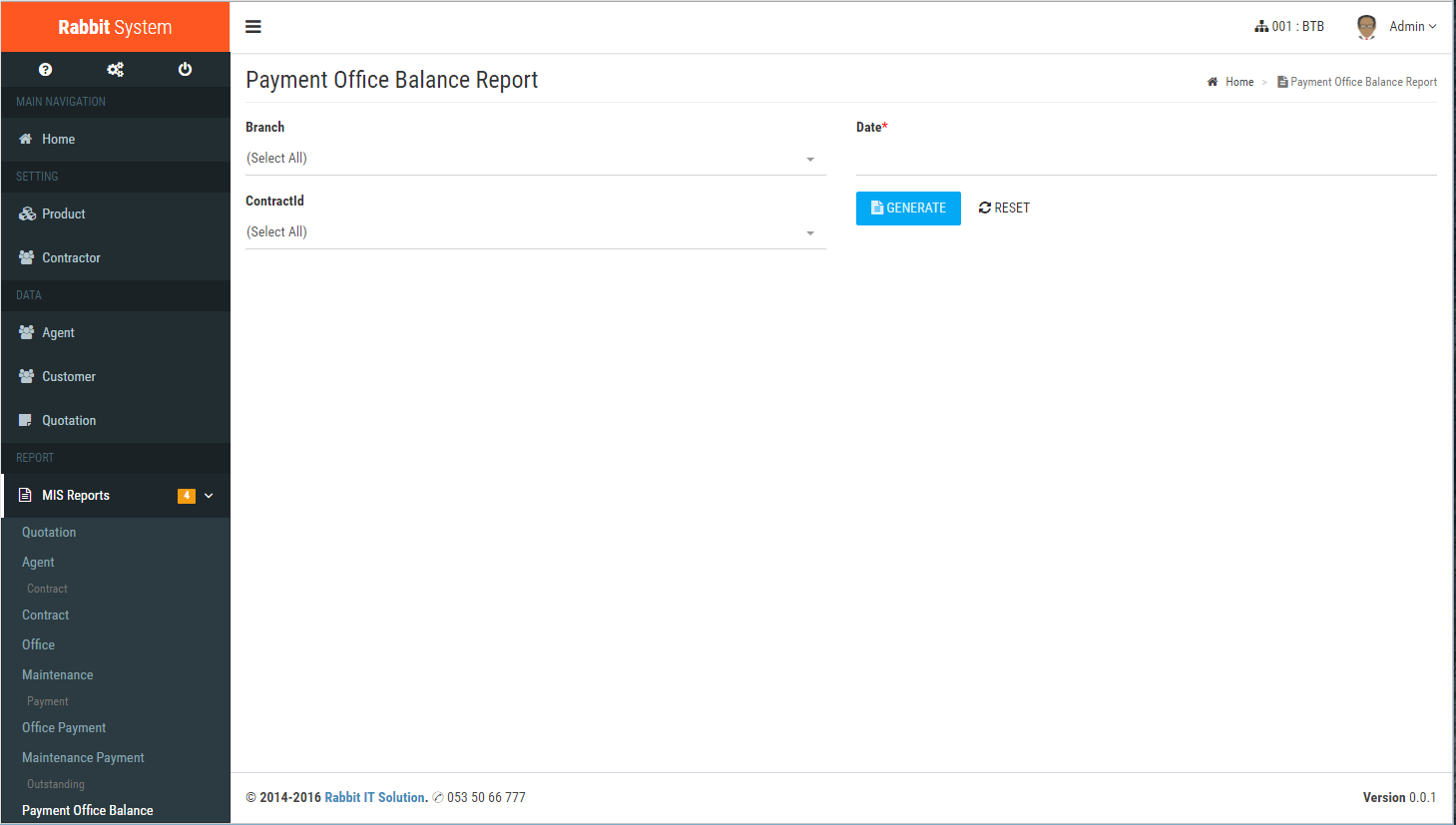
* 1. Maintenance Payment

ជារបាយការណ៍នៃការបង់ប្រាក់ទៅលើការទិញការថែទាំ។



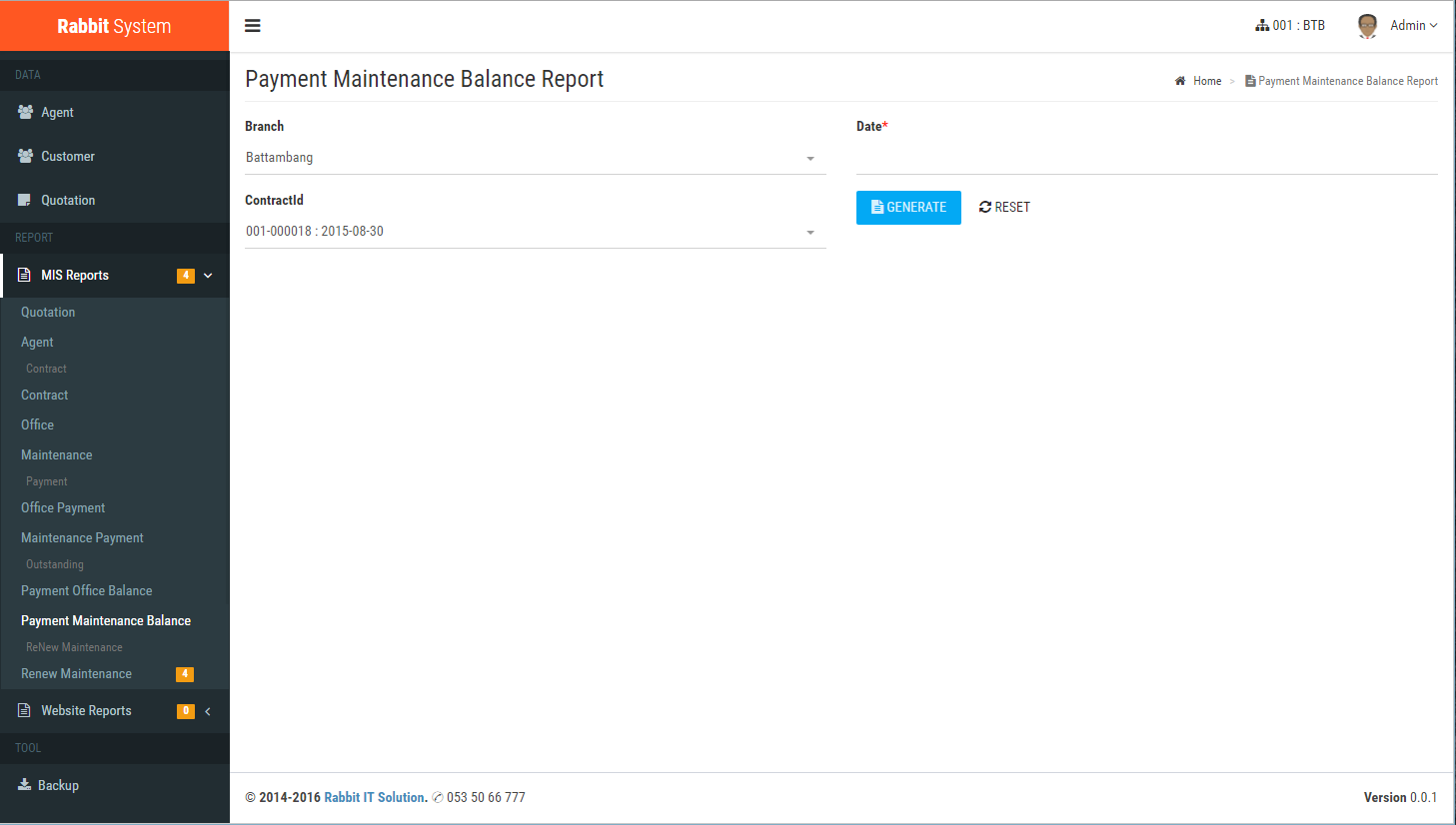
* 1. Payment Office Balance

ជារបាយការណ៍នៃ



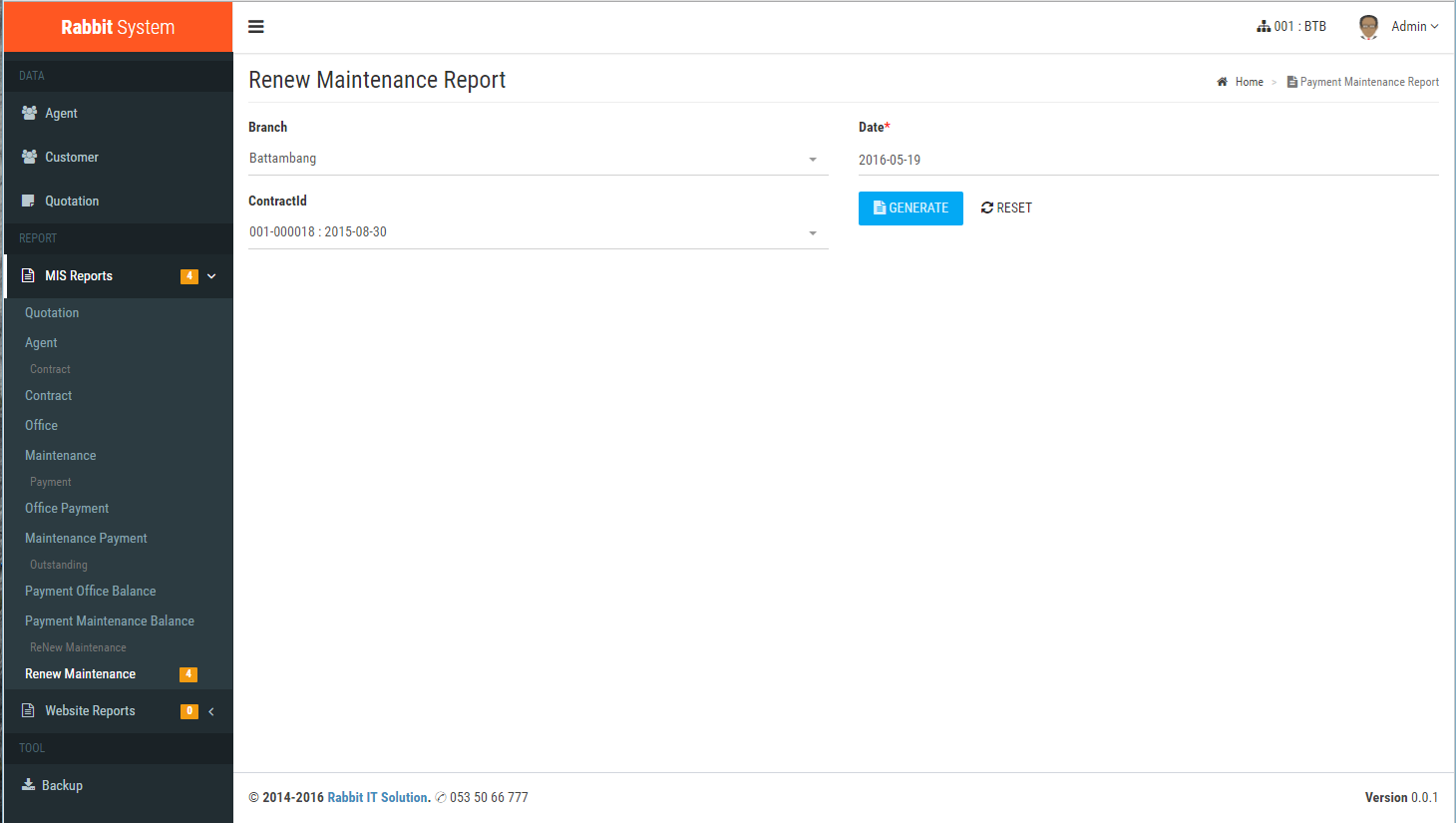
* 1. Payment Maintenance Balance

ជារបាយការណ៍នៃ



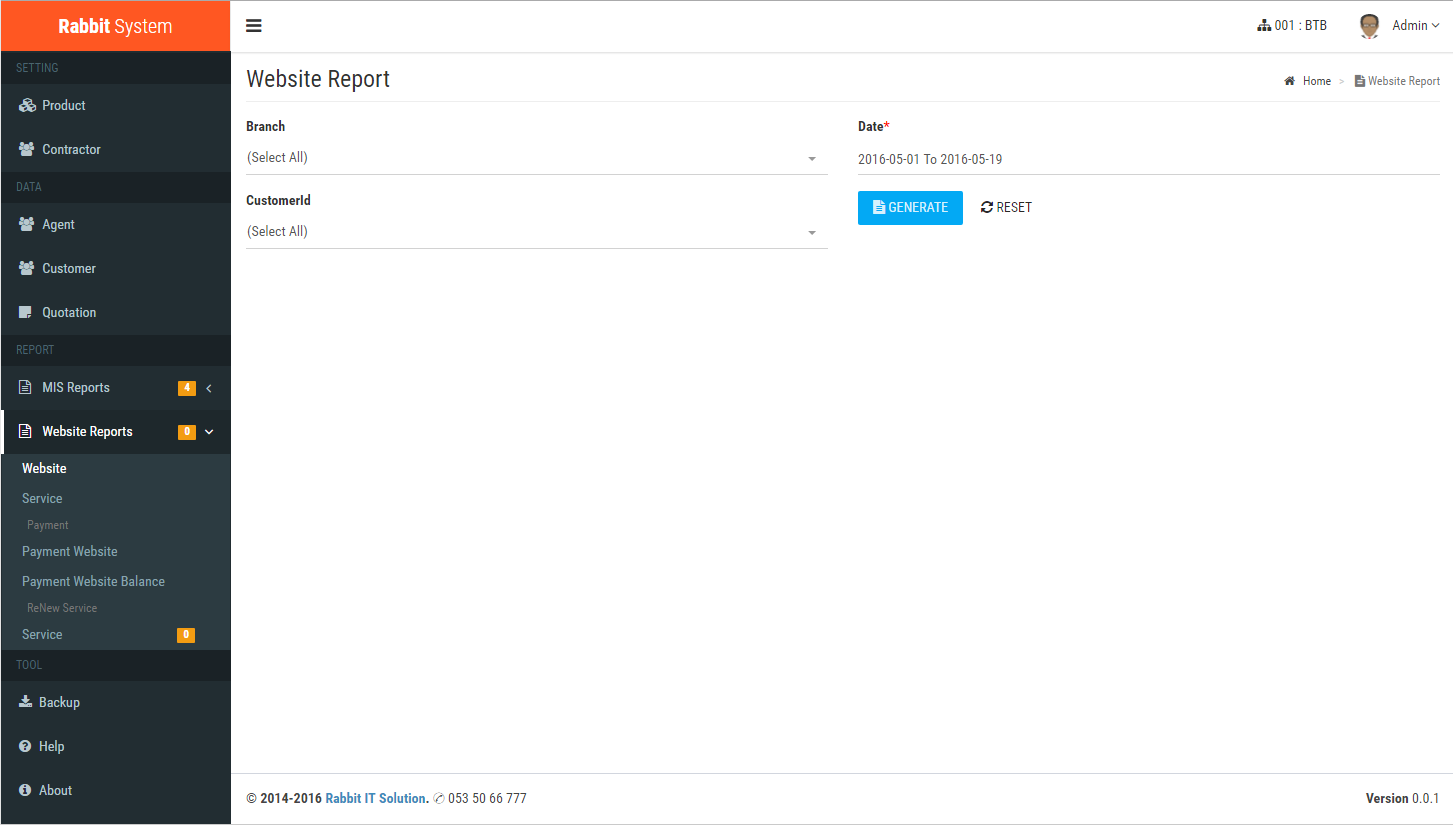
* 1. Renew Maintenance

ជារបាយការណ៍នៃការបន្តការទិញការថែទាំ។



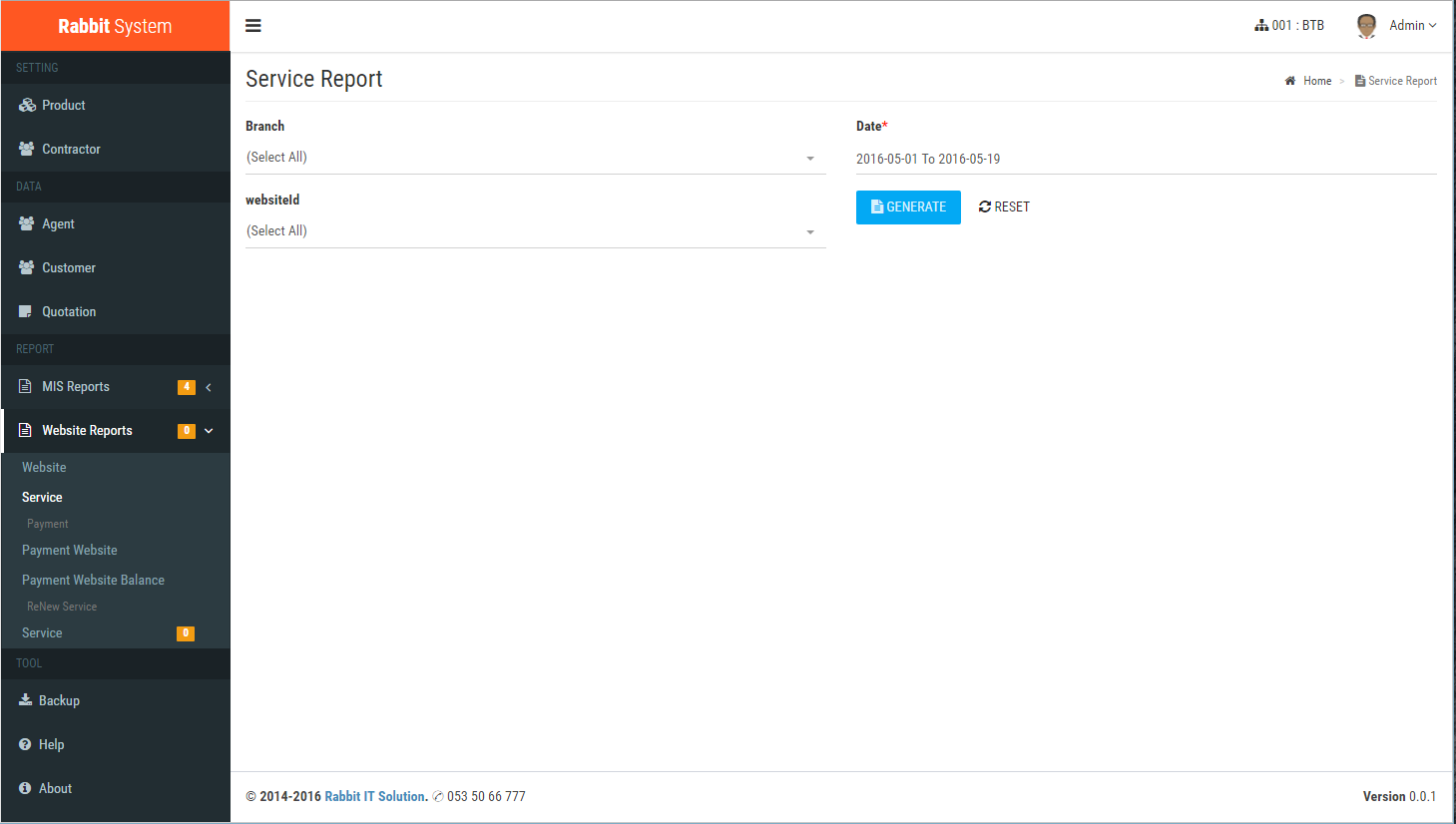
1. Website Report
   1. Website

ជារបាយការណ៍នៃការទិញវេបសាយ។



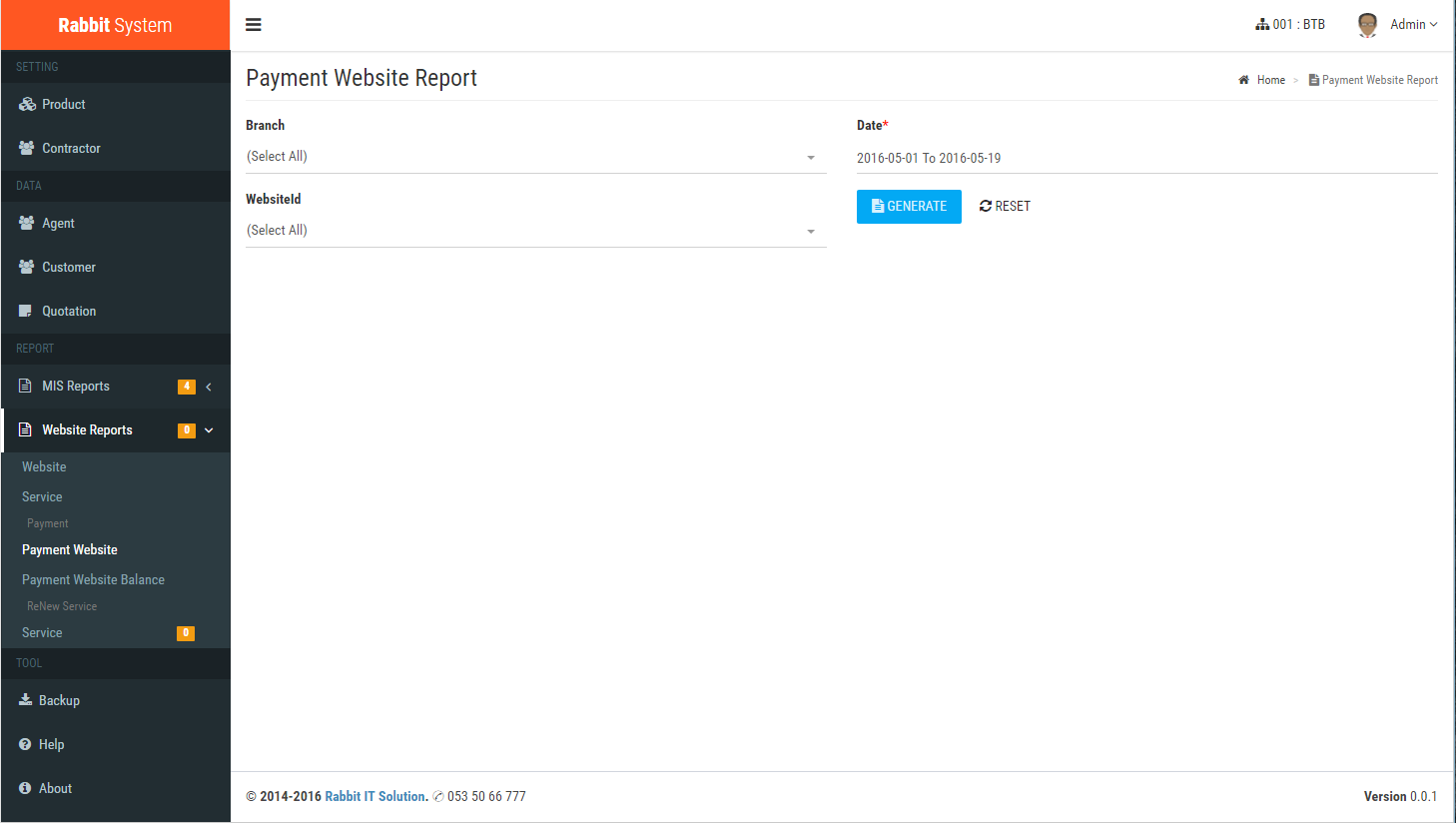
* 1. Service

ជារបាយការណ៍នៃការទិញសេវាកម្មរបស់វេបសាយ (Domain Name, Hosting)។



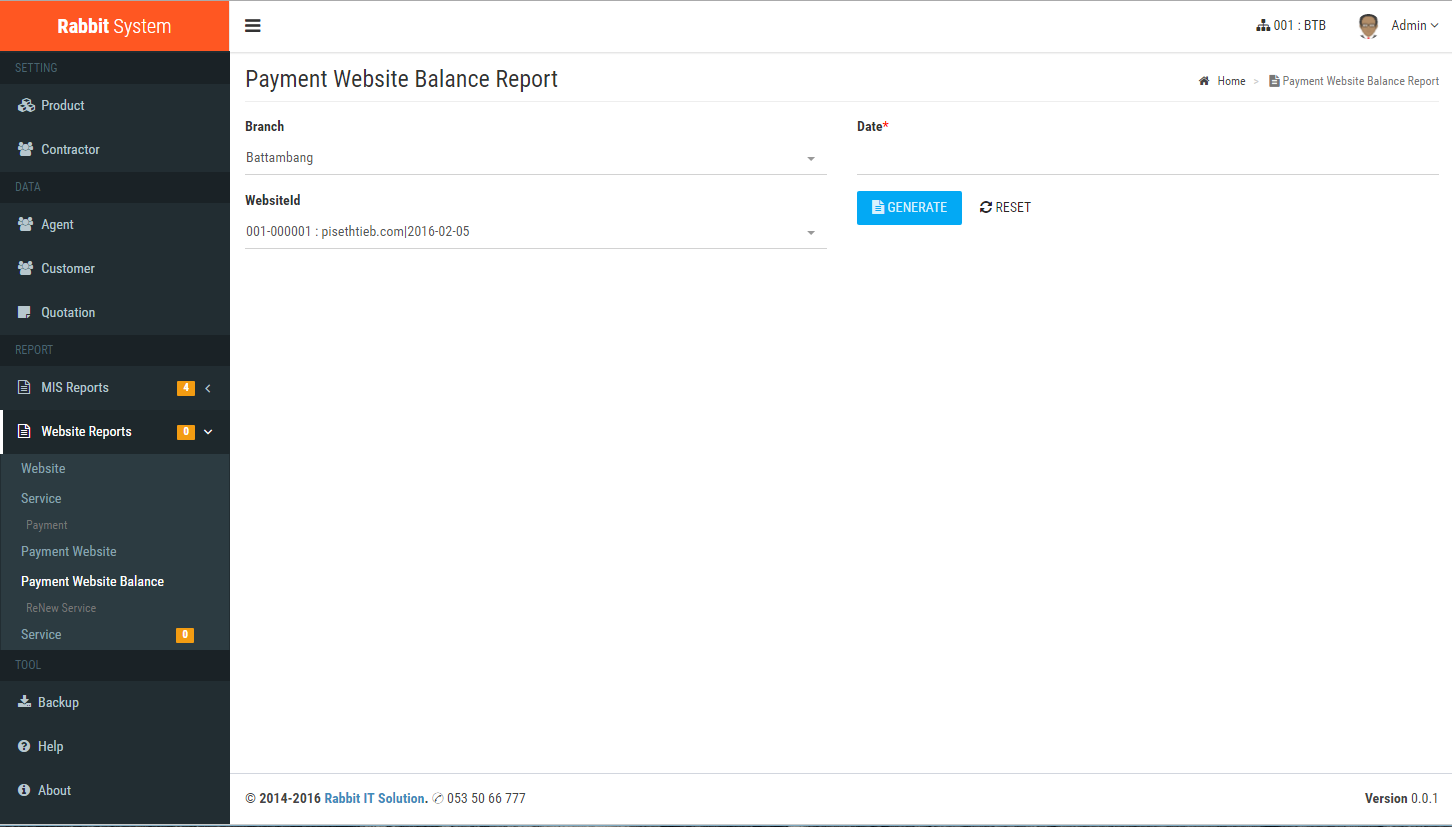
* 1. Payment Website

ជារបាយការណ៍នៃការបង់ប្រាក់ទៅលើការទិញវេបសាយ។



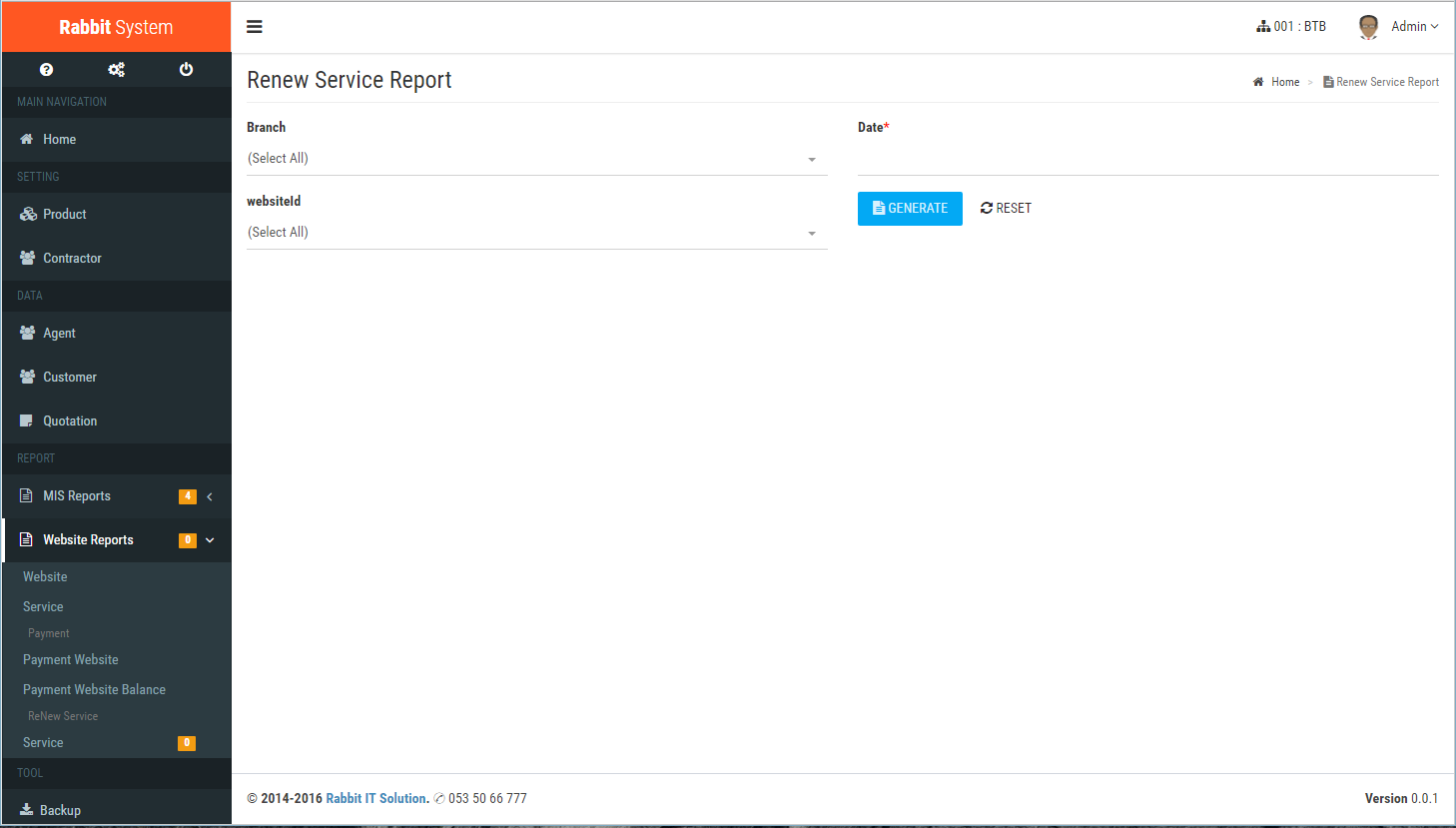
* 1. Payment Website Balance

ជារបាយការណ៍នៃការ



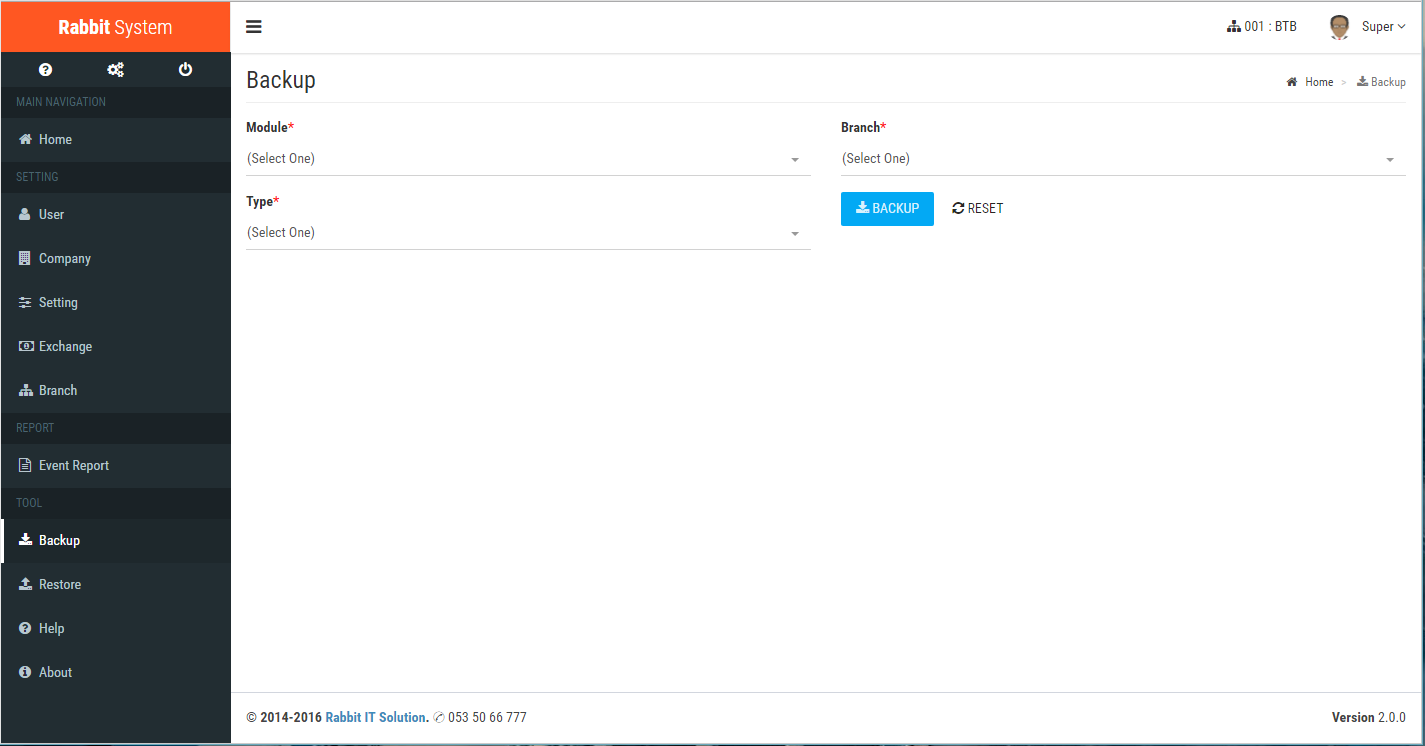
* 1. Renew Service

ជារបាយការណ៍នៃការបន្តការទិញសេវាកម្ម។

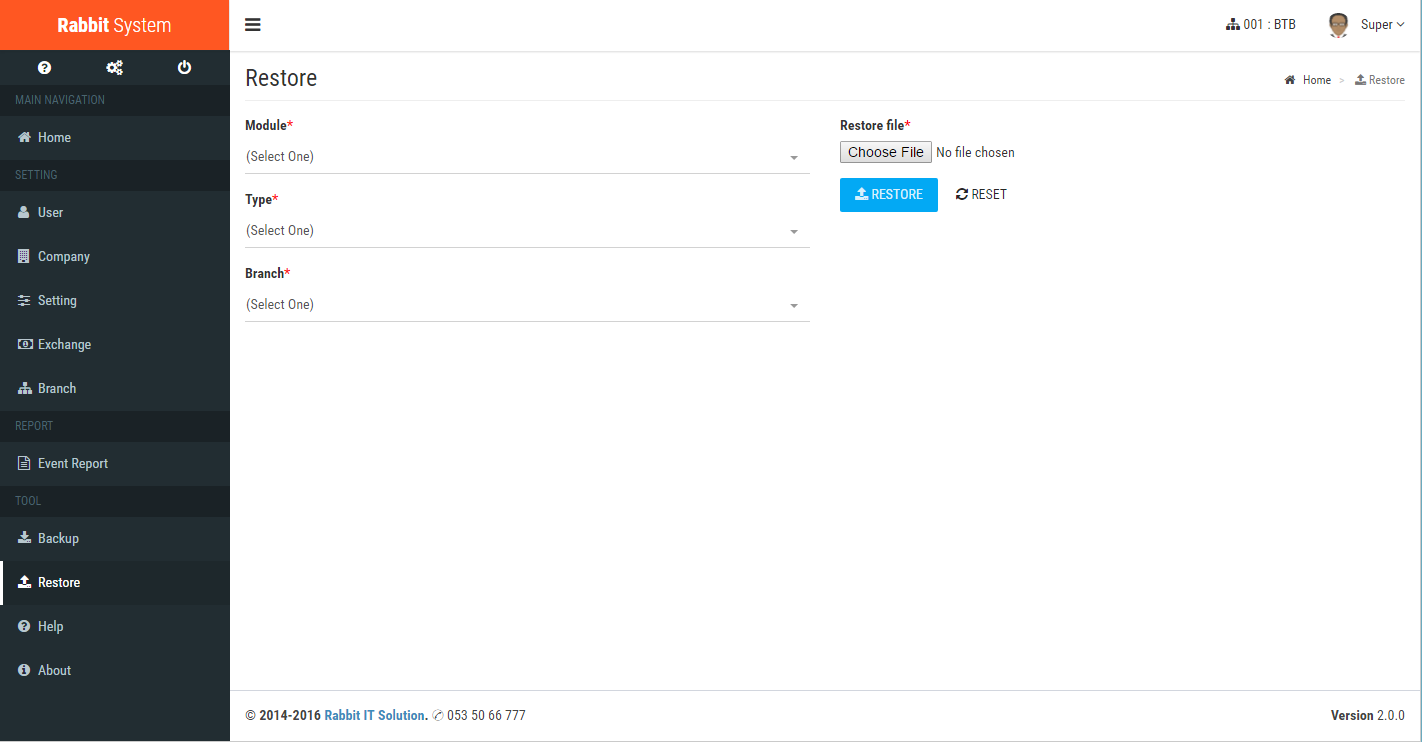


1. Backup

សម្រាប់ធ្វើការ Backup ទិន្នន័យចាស់ៗទុក។



1. Restore



1. Help

ជា Page សម្រាប់បង្ហាញពីរបៀបនៃការប្រើប្រាស់ប្រព័ន្ធ។

1. About

សម្រាប់បង្ហាញពីព័ត៌មាន ក្រុមហ៊ុន, ក្រុមការងារ និងព័ត៌មានរបស់ប្រព័ន្ធ។

៤.៥ **តំរូវការ Hardware និង Software​**

**ប្រព័ន្ធគ្រប់គ្រងនេះបង្កើតមានលក្ខណៈជា** Web Application **ដូចនេះមានតំរូវការ** Hardware​ **និង** Software **ទាបបំផុតសម្រាប់ដំណើរការប្រព័ន្ធនេះត្រូវមានដូចខាងក្រោម៖**

**+ Hardware**

- CPU: 1GHz

- RAM: 256MB

- Monitor

- Keyboard

- Mouse

- Hard Disk: 40GB

**+ Software**

**- ប្រភេទ** OS **អ្វីក៏បាន (**Windows, Linux, Mac OS**)**

**- ប្រភេទ** Browser **អ្វីក៏បាន (**Chrome, Firefox….)

# ការពិភាក្សា

បន្ទាប់ពីការចុះអង្កេត និងសាកសួរព័ត៌មានកន្លងមកក្រុមយើងខ្ញុំឃើញថាប្រព័ន្ធគ្រប់គ្រងនៅក្រុមហ៊ុន Rabbit IT Solution មានការលំបាកព្រោះប្រព័ន្ធនៃការគ្រប់គ្រងចាស់ដែលធ្វើឡើងដោយដៃទើបបណ្តាលឲ្យការងារខ្វះប្រសិទ្ធភាពដែលបញ្ហាទាំងអស់មានដូចជា៖

## ៥.១. ប្រព័ន្ធចាស់

* ការគ្រប់គ្រងការលក់៖ ការបញ្ចូលទិន្នន័យមានភាពយឺតយ៉ាវ និងពិបាកស្វែងរក
* ការគ្រប់គ្រងផលិតផល៖ មិនដឹងថាផលិតផលណាខ្លះបានលក់ចេញក្នុងមួយខែៗ
* ការគ្រប់គ្រងអតិថិជន៖ ពិបាកក្នុងការស្វែងរកឈ្មោះក្រុមហ៊ុន អាសយដ្ឋាន និងចំនួនទឹកប្រាក់ដែលបង់មិនទាន់អស់
* ការគ្រប់គ្រងទិន្នន័យ៖ គ្មានសុវត្ថិភាព ងាយបាត់បង់ និងអាចមានបញ្ហាផ្សេងៗគ្រប់ពេល
* ការគ្រប់គ្រងប្រាក់ចំនូល៖ ពិបាករៀបចំប្រាក់ចំណូលដែលទទួលបាន
* ការគ្រប់គ្រងភ្នាក់ងារនាំអតិថិជន(Agent)៖ ពិបាកក្នុងការស្វែងរកឈ្មោះ និងប្រាក់កម្រៃដែលធ្លាប់ផ្តល់ជូន​
* ការគ្រប់គ្រងរបាយការណ៍៖ ពេលរៀបចំរបាយការណ៍ម្តងៗត្រូវចំនាយពេលយូរ, របាយការណ៍គ្មានរបៀបរៀបរយ។

## ៥.២. ប្រព័ន្ធថ្មី

ក្នុងការសិក្សាបង្កើតប្រព័ន្ធថ្មីដែលអ្នកប្រើប្រាស់ត្រូវប្រតិបត្តិការក្នុងការគ្រប់គ្រងការងារក្នុងនោះក្រុមយើងខ្ញុំបានធ្វើការរៀបចំបង្កើតដែលធ្វើឲ្យមាន៖

* គ្រប់គ្រងការលក់
* គ្រប់គ្រងផលិតផល
* គ្រប់គ្រងព័ត៌មានអតិថិជន
* គ្រប់គ្រងទិន្នន័យ
* គ្រប់គ្រងប្រាក់ចំនូល
* គ្រប់គ្រងព័ត៌មានភ្នាក់ងារនាំអតិថិជន(Agent)
* មានភាពងាយស្រួលក្នុងការមើលរបាយការណ៍

## ៥.៣. សេចក្តីសំយោគ

ការប្រើប្រាស់ប្រព័ន្ធចាស់បានបង្ហាញឲ្យឃើញថា តែងតែមានការលំបាកក្នុងការគ្រប់គ្រងការងារ ដូច្នេះដើម្បីកាត់បន្ថយបញ្ហាទាំងនេះក្រុមនិស្សិតយើងខ្ញុំបានធ្វើការកែប្រែនូវប្រព័ន្ធចាស់ដោយធ្វើការផ្លាស់ប្តូរទៅជាការងារដែលគ្រប់គ្រងដោយកុំព្យូទ័រ ព្រោះថាប្រព័ន្ធថ្មីមានសមត្ថភាពខ្ពស់ លឿនរហ័ស ចំនេញពេលវេលា ចំនេញថវិកា និងអាចជឿទុកចិត្តបានក្នុងការគណនា ក៏ដូចជាការផ្ទុកទិន្នន័យក្នុងចំនួនច្រើនដែលប្រកបដោយសុវត្ថិភាពខ្ពស់ ហើយនឹងព្រមទាំងផ្តល់នូវភាពងាយស្រួលដល់ការគ្រប់គ្រងដែលប្រកប​ដោយប្រសិទ្ធិភាព។

# សេចក្តីសន្និដ្ឋាន និងអនុសាសន៍

## ៦.១. សេចក្តីសន្និដ្ឋាន

សរុបសេចក្តីមក យោងតាមលទ្ធផលដែលបានមកពីការសាកល្បងប្រើប្រាស់ប្រព័ន្ធថ្មីនេះ និងការវិភាគទៅលើដំណើរការរបស់ប្រព័ន្ធថ្មីមួយនះបានយ៉ាងល្អិតល្អន់រួចមក យើង​​អាច​សន្និដ្ឋានបានថា ប្រព័ន្ធនេះអាចប្រើប្រាស់បានយ៉ាងមានប្រសិទ្ធិភាព រហ័ស និង​អាចជឿជាក់បាន ក្នុងការគ្រប់គ្រងការលក់ប្រព័ន្ធ​គ្រប់​គ្រង​លក្ខណៈ Web-based Application និងគេហទំព័រ។ ខ្ញុំបាទសង្ឃឹមថា “គ្រប់គ្រងការលក់ប្រព័ន្ធ​គ្រប់​គ្រង​លក្ខណៈ Web-based Application និងគេហទំព័រ” ដែលក្រុមយើងខ្ញុំបាទបានបង្កើតឡើងថ្មីនេះ អាចជួយសម្រល់ដល់បុគ្គលិក របស់ក្រុមហ៊ុន Rabbit IT Solution ក្នុងការគ្រប់គ្រងទិន្នន័យរបស់ការលក់ ប្រកបដោយប្រសិទ្ធិភាពខ្ពស់ និងមានភាពងាយស្រួលក្នុងការបំពេញភារកិច្ចរបស់ពួកគាត់។

## ៦.២. អនុសាសន៍

ប្រព័ន្ធគ្រប់គ្រងការលក់ប្រព័ន្ធ​គ្រប់​គ្រង​លក្ខណៈ Web-based Application និង គេហទំព័រថ្វីបើមានអត្ថប្រយោជន៍ជាច្រើន​សម្រាប់ក្រុមហ៊ុនក៏ពិតមែន ប៉ុន្តែយើងក៏អាច កែប្រែបន្ថែមនៅពេលអនាគតផងដែរ ពីព្រោះបច្ចេកវិទ្យាទំនើបៗក្នុងយុគសម័យនេះ មានការរីកចម្រើនយ៉ាងខ្លាំងហើយ មានក្រុមហ៊ុនជាច្រើន បានបង្កើតនូវ Smart Devices ជាច្រើនដូចជា Smart Phone, Tablet ជាដើម ដែលមានភាពងាយស្រួល ក្នុងការប្រើប្រាស់។ ហេតុដូចនេះ និស្សិតជំនាន់ក្រោយអាចកែប្រែប្រព័ន្ធនេះ ដោយបន្ថែម​លទ្ធភាពឲ្យប្រព័ន្ធនេះអាចភា្ជប់ទៅកាន់ Smart Devices​ ទាំងឡាយ ដោយបង្កើតជា API រឺ Web service សម្រាប់ឲ្យ Smart Devices អាចផ្ដោះប្តូរទិន្នន័យជាមួយប្រព័ន្ធនេះបាន។

# 

ឯកសារយោង

* <https://docs.mongodb.org/manual/>
* <https://www.meteor.com/>
* http://blog.digitaltutors.com/whats-difference-front-end-back-end/
* <https://opensource.com/resources/what-open-source>
* <https://en.wikipedia.org/wiki/Integrated_development_environment>
* <https://en.wikipedia.org/wiki/Source_code>
* <https://en.wikipedia.org/wiki/Source_code_editor>
* https://robomongo.org/

# ឧបសម្ព័ន្ធ

|  |
| --- |
| * **Product** * **Product Router (product.js)**   **var *subs*** = **new** SubsManager(); **rabbitRoutes**.route(**'/product'**, {  **name**: **'rabbit.product'**,  **title**: **'Product'**,  action: **function** (params, queryParams) {  **Layout**.main(**'rabbit\_product'**);  },  **breadcrumb**: {**title**: **'Product'**,  **icon**: **'cubes'**,  **parent**: **'rabbit.home'** } });   * **Product Collections (product.js)****Rabbit**.**Collection**.**Product** = **new** Mongo.Collection(**"rabbit\_product"**);**Rabbit**.**Schema**.**Product** = **new** SimpleSchema({  **name**: {  **type**: String,  **label**: **'Name'**,  **unique**: **true**,  **max**: 250  },  **monthlyFee**: {  **type**: Array,**minCount**: 1,  **maxCount**: 1  },  **'monthlyFee.$'**: {  **type**: Object,  **decimal**: **true** },  **'monthlyFee.$.headOffice'**: {  **type**: Number,  **decimal**: **true** },  **'monthlyFee.$.branch'**: {  **type**: Number,  **decimal**: **true** },  **installationFee**: {  **type**: Number,  **label**: **'InstallationFee'**,  **decimal**: **true** },  **trainingFee**: {  **type**: Number,  **label**: **'TrainingFee'**,  **decimal**: **true** },  **basePrice**: {  **type**: Array,**minCount**: 1,  **maxCount**: 1  },  **'basePrice.$'**: {  **type**: Object  },  **'basePrice.$.headOffice'**: {  **type**: Number,  **decimal**: **true** },  **'basePrice.$.branch'**: {  **type**: Number,  **decimal**: **true** },  **maintenancePrice**: {  **type**: Array,  **minCount**: 1,  **maxCount**: 1  },  **'maintenancePrice.$'**: {  **type**: Object  },  **'maintenancePrice.$.headOffice'**: {  **type**: Number,  **decimal**: **true** },  **'maintenancePrice.$.branch'**: {  **type**: Number,  **decimal**: **true** },  **feature**: {  **type**: String,  **label**: **'Feature'**,  **autoform**: {  **afFieldInput**: {  **type**: **'summernote'**,  **class**: **'editor'**,*// optional* **settings**: {  **height**: 790,  **toolbar**: [  [**'style'**, [**'bold'**, **'italic'**, **'underline'**, **'clear'**]],  [**'font'**, [**'strikethrough'**, **'superscript'**, **'subscript'**]],  [**'fontsize'**, [**'fontsize'**]],  [**'color'**, [**'color'**]],  [**'para'**, [**'ul'**, **'ol'**, **'paragraph'**]],  [**'height'**, [**'height'**]],  [**'picture'**, [**'picture'**, **'fullscreen'**]]  ]  }*// summernote options goes here* }  }  },  **branchId**: {  **type**: String,  **label**: **"Branch"** } }); **Rabbit**.**Collection**.**Product**.attachSchema(**Rabbit**.**Schema**.**Product**); * **Product Tabulars (product.js)**   **Rabbit**.**TabularTable**.**Product** = **new Tabular**.Table({  **name**: **"rabbit\_productList"**,  **collection**: **Rabbit**.**Collection**.**Product**,  **pagingType**: **"full\_numbers"**,  **autoWidth**: **false**,  **columnDefs**: [  {**"width"**: **"12px"**, **"targets"**: 0}  ],  **order**: [[**'1'**, **'desc'**]],  **columns**: [  {**title**: **'<i class="fa fa-bars"></i>'**, **tmpl**: Meteor.***isClient*** && ***Template***.**rabbit\_productAction**},  {**data**: **"\_id"**, **title**: **"ID"**},  {**data**: **"name"**, **title**: **"Name"**},  {**data**: **"basePrice"**, **title**: **"Fully Fee"**,  render: **function** (val, type, doc) {  **var** str = **""**;  **if** (val != **null**) {  val.forEach(**function** (o) {  str += **"<li>HeadOffice : "** + o.**headOffice** + **" | branch :"** + o.**branch** + **"</li>"**;  });  }  **return** str  }  },  {**data**: **"maintenancePrice"**, **title**: **"Maintenance Price"**,  render: **function** (val, type, doc) {  **var** str = **""**;  **if** (val != **null**) {  val.forEach(**function** (o) {  str += **"<li>HeadOffice : "** + o.**headOffice** + **" | branch :"** + o.**branch** + **"</li>"**;  });  }  **return** str  }   },  {**data**: **"monthlyFee"**, **title**: **"Monthly Fee"**,  render: **function** (val, type, doc) {  **var** str = **""**;  **if** (val != **null**) {  val.forEach(**function** (o) {  str += **"<li>HeadOffice : "** + o.**headOffice** + **" | branch :"** + o.**branch** + **"</li>"**;  });  }  **return** str  }  },  {**data**: **""**, **title**: **"Mothly Service"**,  render: **function** (val, type, doc) {  **return "<li> Install : "** + doc.**installationFee** + **" | "** + **" Train : "** + doc.**trainingFee** + **"</li>"**;  }  },  {**data**: **"\_contractCunt"**, **title**: **"C"**},  {**data**: **"\_quotationCount"**, **title**: **"Q"**}  ],  **extraFields**: [**'feature'**, **'installationFee'**, **'trainingFee'**] });   * **Product Template (product.html)**   *<!--Index-->* <**template** name=**"rabbit\_product"**>  <**p**><**button** type=**"button"** class=**"btn btn-primary js-insert"**>{{**fa "plus"**}} Add New</**button**></**p**>  <**p** class=**"pull-right"**>  <**b**>{{**fa "exclamation-triangle"**}} Note :</**b**> C = Contract | Q = Quotation  </**p**>   {{> **tabular table**=**Rabbit**.**TabularTable**.**Product selector**=**selector class**=**"table table-striped table-bordered table-condensed table-hover-pointer"**}} </**template**>  *<!--Action-->* <**template** name=**"rabbit\_productAction"**>  {{#**tabularAction**}}  <**li**><**a** href=**"#"** class=**"js-update"**>{{**fa "pencil"**}} Update</**a**></**li**>  <**li**><**a** href=**"#"** class=**"js-remove"**>{{**fa "remove"**}} Remove</**a**></**li**>  <**li**><**a** href=**"#"** class=**"js-show"**>{{**fa "eye"**}} Show</**a**></**li**>  <**b**>  <**li** role=**"separator"** class=**"divider"**></**li**>  </**b**>  <**li**><**a** href=**"/rabbit/product/print/**{{**this**.**\_id**}}**"** target=**"\_blank"** class=**"feature"**>{{**fa "print"**}} Print  Feature</**a**>  </**li**>  {{/**tabularAction**}} </**template**> *<!--Insert-->* <**template** name=**"rabbit\_productInsert"**>  {{#**autoForm collection**=**Rabbit**.**Collection**.**Product id**=**"rabbit\_productInsert" type**=**"insert"**}}  <**div** class=**"row"**>  <**div** class=**"col-md-6"**>  {{> **afQuickField name**=**"name"**}}  <**div** class=**""**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Fully Fee</**div**>  <**div** class=**"panel-body"**>  {{> **afArrayField name**=**"basePrice" id**=**"basePrice" template**=**"basePriceArrayField" subtemplate**=**"ProductBasePriceObjectField" label**=false}}  {{> **afArrayField name**=**"maintenancePrice" id**=**"maintenancePrice" template**=**"basePriceArrayField" subtemplate**=**"maintenancePriceObjectField"**}}  </**div**>  </**div**>  </**div**>  <**div** class=**""**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Monthly Fee</**div**>  <**div** class=**"panel-body"**>  {{> **afArrayField name**=**"monthlyFee" id**=**"monthlyFee" template**=**"basePriceArrayField" label**=false **subtemplate**=**"productMonthlyFeeObjectField"**}}  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Service</**div**>  <**div** class=**"panel-body"**>  <**div** class=**"col-sm-6-custom-left"**>  {{> **afQuickField name**=**"installationFee" class**=**"input-sm installationFee" placeholder**=**'Installation Fee'**}}  </**div**>  <**div** class=**"col-sm-6-custom-right"**>  {{> **afQuickField name**=**"trainingFee" class**=**"input-sm trainingFee" placeholder**=**"Training Fee"**}}  </**div**>  </**div**>  </**div**>  </**div**>  </**div**>  </**div**>  </**div**>  <**div** class=**"col-md-6"**>  {{> **afQuickField name**=**"feature"**}}  </**div**>  </**div**>  <**button** type=**"submit"** class=**"btn btn-primary"**>{{**fa "floppy-o"**}} Save</**button**>  <**button** type=**"reset"** class=**"btn btn-default"**>{{**fa "refresh"**}} Reset</**button**>  {{/**autoForm**}} </**template**>  *<!--Update-->* <**template** name=**"rabbit\_productUpdate"**>  *<!--Built-in template sub ready by all-->* {{#**if Template**.**subscriptionsReady**}}  {{#**autoForm collection**=**Rabbit**.**Collection**.**Product id**=**"rabbit\_productUpdate" doc**=**this type**=**"update"**}}  <**div** class=**"row"**>  <**div** class=**"col-md-6"**>  {{> **afQuickField name**=**"name"**}}  <**div** class=**""**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Fully Fee</**div**>  <**div** class=**"panel-body"**>  {{> **afArrayField name**=**"basePrice" id**=**"basePrice" template**=**"basePriceArrayField" subtemplate**=**"ProductBasePriceObjectField" label**=false}}  {{> **afArrayField name**=**"maintenancePrice" id**=**"maintenancePrice" template**=**"basePriceArrayField" subtemplate**=**"maintenancePriceObjectField"**}}  </**div**>  </**div**>  </**div**>  <**div** class=**""**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Monthly Fee</**div**>  <**div** class=**"panel-body"**>  {{> **afArrayField name**=**"monthlyFee" id**=**"monthlyFee" template**=**"basePriceArrayField" label**=false **subtemplate**=**"productMonthlyFeeObjectField"**}}  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Service</**div**>  <**div** class=**"panel-body"**>  <**div** class=**"col-sm-6-custom-left"**>  {{> **afQuickField name**=**"installationFee" class**=**"input-sm installationFee" placeholder**=**'Installation Fee'**}}  </**div**>  <**div** class=**"col-sm-6-custom-right"**>  {{> **afQuickField name**=**"trainingFee" class**=**"input-sm trainingFee" placeholder**=**"Training Fee"**}}  </**div**>  </**div**>  </**div**>  </**div**>  </**div**>  </**div**>  </**div**>  <**div** class=**"col-md-6"**>  {{> **afQuickField name**=**"feature"**}}  </**div**>  </**div**>  <**button** type=**"submit"** class=**"btn btn-primary"**>{{**fa "floppy-o"**}} Save</**button**>  <**button** type=**"reset"** class=**"btn btn-default"**>{{**fa "refresh"**}} Reset</**button**>  {{/**autoForm**}}  {{**else**}}  {{> **loading**}}  {{/**if**}} </**template**>  *<!--Show-->* <**template** name=**"rabbit\_productShow"**>  *<!--Product template sub ready by all-->* {{#**with data**}}  <**dl** class=**"dl-horizontal"**>  <**dt**>ID:</**dt**>  <**dd**>{{**\_id**}}</**dd**>  <**dt**>Name:</**dt**>  <**dd**>{{**name**}}</**dd**>  <**dt**>Product BasePrice:</**dt**>  <**dd**>{{{**base**}}}</**dd**>  <**dt**>Maintenance Price:</**dt**>  <**dd**>{{{**maintenance**}}}</**dd**>  <**dt**>Monthly Fee Price:</**dt**>  <**dd**>{{{**monthlyFee**}}}</**dd**>  <**dt**>InstallationFee:</**dt**>  <**dd**>{{{**installationFee**}}}</**dd**>  <**dt**>Training Fee:</**dt**>  <**dd**>{{{**trainingFee**}}}</**dd**>  <**dt**>Feature:</**dt**>  <**dd**>{{{**feature**}}}</**dd**>  </**dl**>  {{/**with**}} </**template**>  *<!--Custom object field-->* <**template** name=**"ProductBasePriceObjectField"**>  <**div** class=**"col-sm-6"**>  {{> **afQuickField name**=**current**.**headOffice id**=**'basePriceHeadOffice' template**=**"bootstrap3" label**=false **class**=**"input-sm" placeholder**=**"Head Office"**}}  </**div**>  <**div** class=**"col-sm-6"**>  {{> **afQuickField name**=**current**.**branch id**=**'basePriceBranch' template**=**"bootstrap3" label**=false **class**=**"input-sm" placeholder**=**"Branch Office"**}}  </**div**> </**template**> <**template** name=**"maintenancePriceObjectField"**>  <**div** class=**"col-sm-6"**>  {{> **afQuickField name**=**current**.**headOffice template**=**"bootstrap3" id**=**'MaintenaceHeadOffice' label**=false **class**=**"input-sm" placeholder**=**"Head Office"**}}  </**div**>  <**div** class=**"col-sm-6"**>  {{> **afQuickField name**=**current**.**branch id**=**'MaintenaceBranch' template**=**"bootstrap3" label**=false **class**=**"input-sm" placeholder**=**"Branch Office"**}}  </**div**> </**template**> <**template** name=**"productMonthlyFeeObjectField"**>  <**div** class=**"col-sm-6"**>  {{> **afQuickField name**=**current**.**headOffice template**=**"bootstrap3" id**=**'monthlyFeeBranch' label**=false **class**=**"input-sm" placeholder**=**"Head Office"**}}  </**div**>  <**div** class=**"col-sm-6"**>  {{> **afQuickField name**=**current**.**branch id**=**'monthlyFeeBranch' template**=**"bootstrap3" label**=false **class**=**"input-sm" placeholder**=**"Branch Office"**}}  </**div**> </**template**>   * **Product Template (product.js)**   **var *indexTpl*** = ***Template***.**rabbit\_product**,  ***insertTpl*** = ***Template***.**rabbit\_productInsert**,  ***updateTpl*** = ***Template***.**rabbit\_productUpdate**,  ***showTpl*** = ***Template***.**rabbit\_productShow**;  ***indexTpl***.onCreated(**function** () {  *// SEO* **SEO**.set({  **title**: **'Product'**,  **description**: **'Description for this page'** });  *// Create new alertify* createNewAlertify([**"product"**], {**size**: **'lg'**});  createNewAlertify([**"productShow"**], {**size**: **'lg'**});  createNewAlertify([**"locationAddon"**], {**transition**: **'zoom'**, **size**: **'lg'**}); });  ***indexTpl***.helpers({  selector: **function** () {  **return** {**branchId**: ***Session***.get(**'currentBranch'**)};  } });  ***indexTpl***.events({  **'click .js-insert'**: **function** (e, t) {  **alertify**.**product**(fa(**"plus"**, **"Product"**), renderTemplate(***insertTpl***));  },  **'click .js-update'**: **function** (e, t) {  **alertify**.**product**(fa(**"pencil"**, **"Product"**), renderTemplate(***updateTpl***, **this**));  },  **'click .js-remove'**: **function** (e, t) {  **var** id = **this**.**\_id**;  **let** self = **this**;  **var** arr = [  {**collection**: **'Rabbit.Collection.Contract'**, **selector**: {**productId**: id}},  {**collection**: **'Rabbit.Collection.Quotation'**, **selector**: {**productId**: id}}  ];  Meteor.*call*(**'isRelationExist'**, arr, **function** (error, result) {  **if** (result) {  **alertify**.**message**(self.**\_id** + **'|'** + self.**name** + **' is in used !'**);  **return false** }  **else** {  **alertify**.**confirm**(  fa(**"remove"**, **"Product"**),**"Are you sure to delete ["** + self.**\_id** + **"] ?"**,**function** () {  **Rabbit**.**Collection**.**Product**.remove(self.**\_id**, **function**(error) {  **if** (error){  **alertify**.**error**(error.**message**);  } **else** {  **alertify**.**success**(**"Success"**);  }  });  },  **null** );  }  } ); }, **'click .js-show'**: **function** (e, t) {  **alertify**.productShow(fa(**"eye"**, **"Product"**),  renderTemplate(***showTpl***, **this**));  } });  */\*\*  \* Insert  \*/* insertTpl.onRendered(**function** () {  configOnRender(); }); insertTpl.events({  **'keypress #basePriceHeadOffice,#basePriceBranch,**  **#MaintenaceHeadOffice,#MaintenaceBranch,#monthlyFeeHeadOffice,#monthlyFeeBranch,.trainingFee,.installationFee'**: **function** (evt) {  **var** charCode = (evt.which) ? evt.which : evt.keyCode;  **if** ($(evt.currentTarget).val().indexOf(**'.'**) != -1){  **if** (charCode == 46) {  **return false**;  }  }  **return** !(charCode != 46 && charCode > 31 && (charCode < 48 || charCode > 57));  }, });   */\*\*  \* Update  \*/* updateTpl.onCreated(**function** () {  **this**.subscribe(**'rabbit\_product'**, **this**.data.\_id); }); updateTpl.onRendered(**function** () {  Meteor.setTimeout(**function** () {  configOnRender();  }, 200); }); updateTpl.events({  **'keypress #basePriceHeadOffice,#basePriceBranch,**  **#MaintenaceHeadOffice,#MaintenaceBranch,#monthlyFeeHeadOffice,#monthlyFeeBranch,.trainingFee,.installationFee'**: **function** (evt) {  **var** charCode = (evt.which) ? evt.which : evt.keyCode;  **if** ($(evt.currentTarget).val().indexOf(**'.'**) != -1){  **if** (charCode == 46) {  **return false**;  }  }  **return** !(charCode != 46 && charCode > 31 && (charCode < 48 || charCode > 57));  } });  updateTpl.helpers({  data: **function** () {  **var** data= Rabbit.Collection.Product.findOne(**this**.\_id);  **return** data;  } });   */\*\*  \* Show  \*/* showTpl.onCreated(**function** () {  **this**.subscribe(**'rabbit\_product'**, **this**.data.\_id); });  showTpl.helpers({  data: **function** () {  **var** data = Rabbit.Collection.Product.findOne(**this**.\_id);  **return** data;  },  base: **function** () {  **var** str = **"<table class='table table-bordered'>**  **<thead>"** + **"<tr>"** + **"<th>Head Office</th>"** + **"<th>Branch</th>"** + **"</tr>"** + **"</thead><tbody>"**;  **this**.basePrice.forEach(**function** (o) {  str += **'<tr>'** + **'<td>'** + o.headOffice + **'</td>'** + **'<td>'** + o.branch + **'</td>'** + **'</tr>'** });  str += **"</tbody></table>"**;  **return new** Spacebars.SafeString(str);  },  maintenance: **function** () {  **var** str = **"<table class='table table-bordered'>**  **<thead>"** + **"<tr>"** + **"<th>Head Office</th>"** + **"<th>Branch</th>"** + **"</tr>"** + **"</thead><tbody>"**;  **this**.maintenancePrice.forEach(**function** (o) {  str += **'<tr>'** + **'<td>'** + o.headOffice + **'</td>'** + **'<td>'** + o.branch + **'</td>'** + **'</tr>'** });  str += **"</tbody></table>"**;  **return new** Spacebars.SafeString(str);  },  monthlyFee: **function** () {  **var** str = **"<table class='table table-bordered'>**  **<thead>"** + **"<tr>"** + **"<th>Head Office</th>"** + **"<th>Branch</th>"** + **"</tr>"** + **"</thead><tbody>"**;  **this**.monthlyFee.forEach(**function** (o) {  str += **'<tr>'** + **'<td>'** + o.headOffice + **'</td>'** + **'<td>'** + o.branch + **'</td>'** + **'</tr>'** });  str += **"</tbody></table>"**;  **return new** Spacebars.SafeString(str);  } });   */\*\*  \* Hook  \*/* AutoForm.hooks({  *// Product* rabbit\_productInsert: {  before: {  insert: **function** (doc) {  doc.branchId = Session.get(**'currentBranch'**);  **var** prefix = doc.branchId + **'-'**;  Meteor.call(**'rabbit'**, prefix);  **return** doc;  }  },  onSuccess: **function** (formType, result) {  alertify.success(**'Success'**);  },  onError: **function** (formType, error) {  alertify.error(error.message);  }  },  rabbit\_productUpdate: {  onSuccess: **function** (formType, result) {  alertify.product().close();  alertify.success(**'Success'**);  },  onError: **function** (formType, error) {  alertify.error(error.message);  }  } });  *// Config date picker* **var** configOnRender = **function** () {  *// date* **var** dob = $(**'[name="dob"]'**);  DateTimePicker.date(dob); };  **+ Contractor**   * **Contractor Router (contractor.js)**   **var *subs*** = **new** SubsManager(); **rabbitRoutes**.route(**'/contractor'**, {  name: **'rabbit.contractor'**,  title: **'Contractor'**,  action: **function** (params, queryParams) {  Layout.main(**'rabbit\_contractor'**);  },  breadcrumb: {title: **'Contractor'**,  icon: **'users'**,  parent: **'rabbit.home'** } });   * **Contractor Collections (contractor.js)**   **Rabbit**.**Collection**.**Contractor** = **new** Mongo.Collection (**"rabbit\_contractor"**);  *// Schema* **Rabbit**.**Schema**.**Contractor** = **new** SimpleSchema({  **name**: {  **type**: String,  **label**: **"Name"** },  **gender**: {  **type**: String,  **label**: **"Gender"**,  **autoform**: {  **type**: **"select2"**,  options: **function** () {  **return Rabbit**.**List**.gender();  },  **afFieldInput**: {  **select2Options**: {  **theme**: **"bootstrap"** }  }  }  },  **dob**: {  **type**: String,  **abel**: **"Date Of Birth"** },  **id**: {  **type**: String,  **label**: **'ID Card'** },  **address**: {  **type**: String,  **label**: **'Address'** },  **telephone**: {  **type**: String,  **label**: **"Telephone"** },  **email**: {  **type**: String,  **label**: **'Email'**,  **optional**: **true**,  **regEx**: SimpleSchema.**RegEx**.**Email** },  **position**: {  **type**: String,  **label**: **"Position"** } }) ; *// Attach schema* **Rabbit**.**Collection**.**Contractor**.attachSchema(**Rabbit**.**Schema**.**Contractor**);   * **Contractor Tabulars (contractor.js)**   **Rabbit**.**TabularTable**.**Contractor** = **new Tabular**.Table({  **name**: **"rabbit\_saleHeadOfficeList"**,  **collection**: **Rabbit**.**Collection**.**Contractor**,  **pagingType**: **"full\_numbers"**,  **autoWidth**: **false**,  **columnDefs**: [  {**"width"**: **"12px"**, **"targets"**: 0}  ],  **order**: [[**'1'**, **'desc'**]],  **columns**: [  {**title**: **'<i class="fa fa-bars"></i>'**, **tmpl**: Meteor.***isClient*** && ***Template***.**rabbit\_contractorAction**},  {**data**: **"\_id"**, **title**: **"ID"**},  {**data**: **"name"**, **title**: **"Name"**},  {**data**: **"dob"**, **title**: **"DOB"**},  {**data**: **"gender"**, **title**: **"Gender"**},  {**data**: **"id"**, **title**: **"ID"**},  {**data**: **"telephone"**, **title**: **"Telephone"**},  {**data**: **"email"**, **title**: **"Email"**},  {**data**: **"\_contractCount"**, **title**: **"Contract"**},  {**data**: **"\_quotationCount"**, **title**: **"Quotation"**}  ],  **extraFields**: [**'position'**, **'contractorDate'**, **'basePrice'**, **'maintenancePrice'**, **'paymentMethod'**, **'type'**, **'testing'**, **'maintenanceFee'**, **'des'**, **'\_product'**, **'\_customer'**, **'addFile'**, **'address'**] });   * **Contractor Template (contractor.html)**   <**template** name=**"rabbit\_contractor"**>  <**p**><**button** type=**"button"** class=**"btn btn-primary js-insert"**>{{**fa "plus"**}} Add New</**button**></**p**>  {{> **tabular table**=**Rabbit**.**TabularTable**.**Contractor class**=**"table table-striped table-bordered table-condensed table-hover-pointer"**}} </**template**>  *<!--Action-->* <**template** name=**"rabbit\_contractorAction"**>  {{#**tabularAction**}}  <**li**><**a** href=**"#"** class=**"js-update"**>{{**fa "pencil"**}} Update</**a**></**li**>  <**li**><**a** href=**"#"** class=**"js-remove"**>{{**fa "remove"**}} Remove</**a**></**li**>  <**li**><**a** href=**"#"** class=**"js-show"**>{{**fa "eye"**}} Show</**a**></**li**>  {{/**tabularAction**}} </**template**>  *<!--Insert-->* <**template** name=**"rabbit\_contractorInsert"**>  {{#**autoForm collection**=**Rabbit**.**Collection**.**Contractor id**=**"rabbit\_contractorInsert" type**=**"insert"**}}  <**div** class=**"row"**>  <**div** class=**"col-md-6"**>  {{> **afQuickField name**=**"name"**}} {{> **afQuickField name**=**"dob"**}}  {{> **afQuickField name**=**"gender"**}} {{> **afQuickField name**=**"id"**}}  </**div**>  <**div** class=**"col-md-6"**>  {{> **afQuickField name**=**"position"**}}  {{> **afQuickField name**=**"telephone"**}} {{> **afQuickField name**=**"email"**}} {{> **afQuickField name**=**"address"**}}  </**div**>  <**div** class=**"col-md-12"**>  <**button** type=**"submit"** class=**"btn btn-primary"**>{{**fa "floppy-o"**}} Save</**button**>  <**button** type=**"reset"** class=**"btn btn-default"**>{{**fa "refresh"**}} Reset</**button**>  </**div**>  </**div**>  {{/**autoForm**}} </**template**>  *<!--Update-->* <**template** name=**"rabbit\_contractorUpdate"**>  *<!--Built-in template sub ready by all-->* {{#**if Template**.**subscriptionsReady**}} {{#**autoForm collection**=**Rabbit**.**Collection**.**Contractor id**=**"rabbit\_contractorUpdate" doc**=**this type**=**"update"**}}  <**div** class=**"row"**>  <**div** class=**"col-md-6"**>  {{> **afQuickField name**=**"name"**}} {{> **afQuickField name**=**"dob"**}}  {{> **afQuickField name**=**"gender"**}} {{> **afQuickField name**=**"id"**}}  </**div**>  <**div** class=**"col-md-6"**>  {{> **afQuickField name**=**"position"**}} {{> **afQuickField name**=**"telephone"**}} {{> **afQuickField name**=**"email"**}}  {{> **afQuickField name**=**"address"**}}  </**div**>  <**div** class=**"col-md-12"**>  <**button** type=**"submit"** class=**"btn btn-primary"**>{{**fa "floppy-o"**}} Save</**button**>  <**button** type=**"reset"** class=**"btn btn-default"**>{{**fa "refresh"**}} Reset</**button**>  </**div**>  </**div**>  {{/**autoForm**}}  {{**else**}}  {{> **loading**}}  {{/**if**}}  </**template**>  *<!--Show-->* <**template** name=**"rabbit\_contractorShow"**>  *<!--Contractor template sub ready by all-->* {{#**if data**}}  {{#**with data**}}  <**dl** class=**"dl-horizontal"**>  <**dt**>ID:</**dt**>  <**dd**>{{**name**}}</**dd**>  <**dt**>Gender:</**dt**>  <**dd**>{{**gender**}}</**dd**>  <**dt**>Date Of Birth:</**dt**>  <**dd**>{{**dob**}}</**dd**>  <**dt**>ID:</**dt**>  <**dd**>{{**id**}}</**dd**>  <**dt**>Position:</**dt**>  <**dd**>{{**position**}}</**dd**>  <**dt**>Telephone:</**dt**>  <**dd**>{{**telephone**}}</**dd**>  <**dt**>Email:</**dt**>  <**dd**>{{**email**}}</**dd**>  <**dt**>Address:</**dt**>  <**dd**>{{**address**}}</**dd**>  </**dl**>  {{/**with**}}  {{**else**}}  {{> **loading**}}  {{/**if**}} </**template**>   * **Contractor Template (contractor.js)**   **var *indexTpl*** = ***Template***.**rabbit\_contractor**,  ***insertTpl*** = ***Template***.**rabbit\_contractorInsert**,  ***updateTpl*** = ***Template***.**rabbit\_contractorUpdate**,  ***showTpl*** = ***Template***.**rabbit\_contractorShow**;  *//locationAddOnTpl = Template.rabbit\_locationAddOnContractor;   /\*\*  \* Index  \*/* ***indexTpl***.onCreated(**function** () {  *// SEO* **SEO**.set({  **title**: **'Contractor'**,  **description**: **'Description for this page'** });   *// Create new alertify* createNewAlertify([**"contractor"**], {**size**: **'lg'**});  createNewAlertify([**"contractorShow"**], {**size**: **'sm'**});  createNewAlertify([**"locationAddon"**], {**transition**: **'zoom'**, **size**: **'lg'**}); });  ***indexTpl***.onRendered(**function** () {  *//* });  ***indexTpl***.helpers({  selector: **function** () {  **return** {**branchId**: ***Session***.get(**'currentBranch'**)};  } });  ***indexTpl***.events({  **'click .js-insert'**: **function** (e, t) {  **alertify**.**contractor**(fa(**"plus"**, **"Contractor"**), renderTemplate(***insertTpl***));  },  **'click .js-update'**: **function** (e, t) {  **alertify**.**contractor**(fa(**"pencil"**, **"Contractor"**), renderTemplate(***updateTpl***, **this**));  },  **'click .js-remove'**: **function** (e, t) {  **var** id = **this**.**\_id**;  **let** self = **this**;  **var** arr = [  {**collection**: **'Rabbit.Collection.Contract'**, **selector**: {**contractorId**: id}},  {**collection**: **'Rabbit.Collection.Quotation'**, **selector**: {**contractorId**: id}}  ];  Meteor.*call*(**'isRelationExist'**, arr, **function** (error, result) {  **if** (result) {  **alertify**.**message**(self.**\_id** + **'|'** + self.**name** + **' is in used !'**);  **return false** } **else** {  **alertify**.**confirm**(  fa(**"remove"**, **"Contractor"**),  **"Are you sure to delete ["** + self.**\_id** + **"] ?"**,  **function** () {  **Rabbit**.**Collection**.**Contractor**.remove(self.**\_id**, **function** (error) {  **if** (error) {  **alertify**.error(error.**message**);  } **else** {  **alertify**.success(**"Success"**);  }  });  },  **null** );  }  }  );  },  **'click .js-show'**: **function** (e, t) {  **alertify**.contractorShow(fa(**"eye"**, **"Contractor"**), renderTemplate(***showTpl***, **this**));  },  **'click .contractAction'**: **function** () {  **FlowRouter**.go(**'rabbit.contract'**, {  **contractorId**: **this**.**\_id** })   }  });  */\*\*  \* Insert  \*/* ***insertTpl***.onRendered(**function** () {  *configOnRender*(); });   */\*\*  \* Update  \*/* ***updateTpl***.onCreated(**function** () {  **this**.subscribe(**'rabbit\_contractor'**, **this**.data.\_id); });  updateTpl.onRendered(**function** () {  Meteor.setTimeout(**function** () {  configOnRender();  }, 200); });  updateTpl.helpers({  data: **function** () {  **var** data = Rabbit.Collection.Contractor.findOne(**this**.\_id);  **return** data;  } });   */\*\*  \* Show  \*/* showTpl.onCreated(**function** () {  **this**.subscribe(**'rabbit\_contractor'**, **this**.data.\_id); });  showTpl.helpers({  data: **function** () {  **var** data = Rabbit.Collection.Contractor.findOne(**this**.\_id);  **return** data;   },  contactPerson: **function** () {  **var** str = **"<table class='table table-bordered'><thead>"** +  **"<tr>"** +  **"<th>Name</th>"** +  **"<th>Gender</th>"** +  **"<th>Position</th>"** +  **"<th>Telephone</th>"** +  **"</tr>"** +  **"</thead><tbody>"**;  **this**.contactPerson.forEach(**function** (o) {  str += **'<tr>'** +  **'<td>'** + o.name + **'</td>'** +  **'<td>'** + o.gender + **'</td>'** +  **'<td>'** + o.position + **'</td>'** +  **'<td>'** + o.tel + **'</td>'** +  **'</tr>'** });  str += **"</tbody></table>"**;  **return new** Spacebars.SafeString(str);  } });   */\*\*  \* Hook  \*/* AutoForm.hooks({  *// Contractor* rabbit\_contractorInsert: {  before: {  insert: **function** (doc) {  doc.branchId = Session.get(**'currentBranch'**);  **var** prefix = doc.branchId + **'-'**;  Meteor.call(**'rabbit'**, prefix);  **return** doc;  }  },  onSuccess: **function** (formType, result) {  alertify.success(**'Success'**);  },  onError: **function** (formType, error) {  alertify.error(error.message);  }  },  rabbit\_contractorUpdate: {  onSuccess: **function** (formType, result) {  alertify.contractor().close();  alertify.success(**'Success'**);  },  onError: **function** (formType, error) {  alertify.error(error.message);  }  } });  *// Config date picker* **var** configOnRender = **function** () {  **debugger**;  *// date* **var** dob = $(**'[name="dob"]'**);  DateTimePicker.date(dob); };  **+ Agent**   * **Agent Router (agent.js)**   **var *subs*** = **new** SubsManager();  **rabbitRoutes**.route(**'/agent'**, {  **name**: **'rabbit.agent'**,  **title**: **'Agent'**,  subscriptions: **function** (params, queryParams) {},  action: **function** (params, queryParams) {  **Layout**.main(**'rabbit\_agent'**);  },  **breadcrumb**: {**title**: **'Agent'**,  **icon**: **'users'**,  **parent**: **'rabbit.home'** } });   * **Agent Collections (agent.js)**   **Rabbit**.**Collection**.**Agent**= **new** Mongo.Collection(**"rabbit\_agent"**);  *// Schema* **Rabbit**.**Schema**.**Agent**= **new *SimpleSchema***({  **name**: {  **type**: String,  **label**: **"Name"** },  **gender**: {  **type**: String,  **label**: **"Gender"**,  **autoform**: {  **type**: **"select2"**,  options: **function** () {  **return Rabbit**.**List**.gender();  },  **afFieldInput**: {  **select2Options**: {  **theme**: **"bootstrap"** }  }  }  },  **address**: {  **type**: String,  **label**: **'Address'**,  **autoform**: {  **afFieldInput**: {  **type**: **"textarea"** }  }  },  **telephone**: {  **type**: String,  **label**: **"Telephone"** },  **branchId**: {  **type**: String,  **label**: **"Branch"** } }) ; *// Attach schema* **Rabbit**.**Collection**.**Agent**.attachSchema(**Rabbit**.**Schema**.**Agent**);   * **Agent Tabulars (agent.js)**   **Rabbit**.**TabularTable**.**Agent** = **new Tabular**.Table({  **name**: **"rabbit\_agent"**,  **collection**: **Rabbit**.**Collection**.**Agent**,  **pagingType**: **"full\_numbers"**,  **autoWidth**: **false**,  **columnDefs**: [  {**"width"**: **"12px"**, **"targets"**: 0}  ],  **order**: [[**'1'**, **'desc'**]],  **columns**: [  {**title**: **'<i class="fa fa-bars"></i>'**, **tmpl**: Meteor.***isClient*** && ***Template***.**rabbit\_contractorAction**},  {**data**: **"\_id"**, **title**: **"ID"**},  {**data**: **"name"**, **title**: **"Name"**},  *//{data: "dob", title: "DOB"},* {**data**: **"gender"**, **title**: **"Gender"**},  *//{data: "id", title: "ID"},* {**data**: **"telephone"**, **title**: **"Telephone"**},  {**data**: **"address"**, **title**: **"Address"**},  *//{data: "email", title: "Email"},* {**data**: **"\_agentCount"**, **title**: **"Contract"**}    ],  **extraFields**: [**'position'**, **'contractorDate'**, **'basePrice'**, **'maintenancePrice'**, **'paymentMethod'**, **'type'**, **'testing'**, **'maintenanceFee'**, **'des'**, **'\_product'**, **'\_customer'**, **'addFile'**] });   * **Agent Template (agent.html)**   <**template** name=**"rabbit\_agent"** xmlns=**"http://www.w3.org/1999/html"**>  <**p**>  <**button** type=**"button"** class=**"btn btn-primary js-insert"**>{{**fa "plus"**}} Add New</**button**>  </**p**>  {{> **tabular table**=**Rabbit**.**TabularTable**.**Agent class**=**"table table-striped table-bordered table-condensed table-hover-pointer"**}} </**template**>  *<!--Action-->* <**template** name=**"rabbit\_agentAction"**>  {{#**tabularAction**}}  <**li**><**a** href=**"#"** class=**"js-update"**>{{**fa "pencil"**}} Update</**a**></**li**>  <**li**><**a** href=**"#"** class=**"js-remove"**>{{**fa "remove"**}} Remove</**a**></**li**>  <**li**><**a** href=**"#"** class=**"js-show"**>{{**fa "eye"**}} Show</**a**></**li**>   {{/**tabularAction**}} </**template**>  *<!--Insert-->* <**template** name=**"rabbit\_agentInsert"**>  {{#**autoForm collection**=**Rabbit**.**Collection**.**Agent id**=**"rabbit\_agentInsert" type**=**"insert"**}}  <**div** class=**"row"**>  {{> **afQuickField name**=**"name"**}}  *<!--*{{> **afQuickField name**=**"dob"**}}*-->* {{> **afQuickField name**=**"gender"**}}  *<!--*{{> **afQuickField name**=**"id"**}}*-->  <!--*{{> **afQuickField name**=**"position"**}}*-->* {{> **afQuickField name**=**"telephone"**}}  *<!--*{{> **afQuickField name**=**"email"**}}*-->* {{> **afQuickField name**=**"address"**}}  </**div**>   <**button** type=**"submit"** class=**"btn btn-primary"**>{{**fa "floppy-o"**}} Save</**button**>  <**button** type=**"reset"** class=**"btn btn-default"**>{{**fa "refresh"**}} Reset</**button**>  {{/**autoForm**}} </**template**>  *<!--Update-->* <**template** name=**"rabbit\_agentUpdate"**>  *<!--Built-in template sub ready by all-->* {{#**if Template**.**subscriptionsReady**}}  {{#**autoForm collection**=**Rabbit**.**Collection**.**Agent id**=**"rabbit\_agentUpdate" doc**=**this type**=**"update"**}}  <**div** class=**"row"**>  {{> **afQuickField name**=**"name"**}}  *<!--*{{> **afQuickField name**=**"dob"**}}*-->* {{> **afQuickField name**=**"gender"**}}  *<!--*{{> **afQuickField name**=**"id"**}}*-->  <!--*{{> **afQuickField name**=**"position"**}}*-->* {{> **afQuickField name**=**"telephone"**}}  *<!--*{{> **afQuickField name**=**"email"**}}*-->* {{> **afQuickField name**=**"address"**}}  <**button** type=**"submit"** class=**"btn btn-primary"**>{{**fa "floppy-o"**}} Save</**button**>  <**button** type=**"reset"** class=**"btn btn-default"**>{{**fa "refresh"**}} Reset</**button**>  </**div**>  {{/**autoForm**}}  {{**else**}}  {{> **loading**}}  {{/**if**}} </**template**>  *<!--Show-->* <**template** name=**"rabbit\_agentShow"**>  *<!--Agenttemplate sub ready by all-->* <**dl** class=**"dl-horizontal"**>  <**dt**>ID:</**dt**>  <**dd**>{{**\_id**}}</**dd**>  <**dt**>Name:</**dt**>  <**dd**>{{**name**}}</**dd**>  <**dt**>Gender:</**dt**>  <**dd**>{{**gender**}}</**dd**>  <**dt**>Telephone:</**dt**>  <**dd**>{{**telephone**}}</**dd**>  <**dt**>Address:</**dt**>  <**dd**>{{**address**}}</**dd**>  </**dl**> </**template**>   * **Agent Template (agent.js)**   **var *indexTpl*** = ***Template***.**rabbit\_agent**,  ***insertTpl*** = ***Template***.**rabbit\_agentInsert**,  ***updateTpl*** = ***Template***.**rabbit\_agentUpdate**,  ***showTpl*** = ***Template***.**rabbit\_agentShow**;   */\*\*  \* Index  \*/* ***indexTpl***.onCreated(**function** () {  *// SEO* **SEO**.set({  **title**: **'agent'**,  **description**: **'Description for this page'** });   *// Create new alertify* createNewAlertify([**"agent"**], {**size**: **'sm'**});  createNewAlertify([**"agentShow"**]);  createNewAlertify([**"locationAddon"**], {**transition**: **'zoom'**, **size**: **'lg'**}); });  ***indexTpl***.helpers({  selector: **function** () {  **return** {**branchId**: ***Session***.get(**'currentBranch'**)};  } });  ***indexTpl***.events({  **'click .js-insert'**: **function** (e, t) {  **alertify**.agent(fa(**"plus"**, **"Agent"**), renderTemplate(***insertTpl***));  },  **'click .js-update'**: **function** (e, t) {  **alertify**.agent(fa(**"pencil"**, **"Agent"**), renderTemplate(***updateTpl***, **this**));  },   **'click .js-remove'**: **function** (e, t) {  **var** id = **this**.**\_id**;  **let** self = **this**;  **var** arr = [  {**collection**: **'Rabbit.Collection.Contract'**, **selector**: {**agentId**: id}},  {**collection**: **'Rabbit.Collection.Quotation'**, **selector**: {**agentId**: id}}  ];  Meteor.*call*(**'isRelationExist'**, arr, **function** (error, result) {  **if** (result) {   **alertify**.**message**(self.**\_id** + **'|'** + self.**name** + **' is in used !'**);  **return false** } **else** {  **alertify**.**confirm**(  fa(**"remove"**, **"Agent"**),  **"Are you sure to delete ["** + self.**\_id** + **"] ?"**,  **function** () {  **Rabbit**.**Collection**.**Agent**.remove(self.**\_id**, **function** (error) {  **if** (error) {  **alertify**.error(error.**message**);  } **else** {  **alertify**.success(**"Success"**);  }  });  },  **null** );  }  }  );},    **'click .js-show'**: **function** (e, t) {  **alertify**.agentShow(fa(**"eye"**, **"Agent"**), renderTemplate(***showTpl***, **this**));  }   });  */\*\*  \* Insert  \*/* ***insertTpl***.onRendered(**function** () {  *configOnRender*(); });   */\*\*  \* Update  \*/* ***updateTpl***.onCreated(**function** () {  *//this.subscribe('rabbit\_agent', this.data.\_id);* });  ***updateTpl***.onRendered(**function** () {  *configOnRender*(); });   */\*\*  \* Hook  \*/* **AutoForm**.hooks({  *// agent* **rabbit\_agentInsert**: {  **before**: {  insert: **function** (doc) {  doc.**branchId** = ***Session***.get(**'currentBranch'**);  **var** prefix = doc.**branchId** + **'-'**;  Meteor.*call*(**'rabbit'**, prefix);  **return** doc;  }  },  onSuccess: **function** (formType, result) {  **alertify**.success(**'Success'**);  },  onError: **function** (formType, error) {  **alertify**.error(error.**message**);  }  },  **rabbit\_agentUpdate**: {  onSuccess: **function** (formType, result) {  **alertify**.agent().close();  **alertify**.success(**'Success'**);  },  onError: **function** (formType, error) {  **alertify**.error(error.**message**);  }  } });  *// Config date picker* **var** *configOnRender* = **function** () {  **debugger**;  *// date* **var** dob = $(**'[name="dob"]'**);  **DateTimePicker**.date(dob); };  **+ Customer**   * **Customer Router (customer.js)**   **var *subs*** = **new** SubsManager();  **rabbitRoutes**.route(**'/customer'**, {  **name**: **'rabbit.customer'**,  **title**: **'Customer'**,  action: **function** (params, queryParams) {  **Layout**.main(**'rabbit\_customer'**);  },  **breadcrumb**: {**title**: **'Customer'**,  **icon**: **'user'**,  **parent**: **'rabbit.home'** } });   * **Customer Collections (customer.js)**   **Rabbit**.**Collection**.**Customer** = **new** Mongo.Collection(**"rabbit\_customer"**);  *// Schema* **Rabbit**.**Schema**.**Customer** = **new *SimpleSchema***({  **contractName**: {  **type**: String,  **label**: **"Name"** },  **gender**: {  **type**: String,  **label**: **'Gender'**,  **autoform**: {  **type**: **"select2"**,  options: **function**() {  **return Rabbit**.**List**.gender();  },  **afFieldInput**: {  **select2Options**: {  **theme**: **"bootstrap"** }  }  }  },  **dob**: {  **type**: String,  **label**: **'Date Of Birth'** },  **id**: {  **type**: String,  **label**: **'ID Card'**,  **optional**: **true** },  **position**: {  **type**: String,  **label**: **"Position"** },  **address**: {  **type**: String,  **label**: **'Address'**, **optional**: **true** },  **telephone**: {  **type**: String,  **label**: **'Telephone'** },  **email**: {  **type**: String,  **label**: **'Email'**,  **regEx**: ***SimpleSchema***.**RegEx**.**Email**,  **optional**: **true** },    **companyName**: {  **type**: String,  **label**: **"Company Name"**,  **max**: 200  },  **companyAddress**: {  **type**: String,  **label**: **"Address"**,  **autoform**: {  **afFieldInput**: {  **type**: **"textarea"** }  }  },  **companyTelephone**: {  **type**: String,  **label**: **"Telephone"** },  **companyEmail**: {  **type**: String,  **label**: **"Email"**,  **regEx**: ***SimpleSchema***.**RegEx**.**Email**,  **optional**: **true** },  **companyWebsite**: {  **type**: String,  **label**: **"Website"**,  **optional**: **true** },  **contactPerson**: {  **type**: Array,  **minCount**: 1,  **maxCount**: 5,  **optional**: **true**,  **label**: **"Contact Person"** },  **'contactPerson.$'**: {  **type**: Object  },  **'contactPerson.$.name'**: {  **type**: String  },  **'contactPerson.$.gender'**: {  **type**: String,  **autoform**: {  **type**: **"selectize"**,  options: **function**() {  **return Rabbit**.**List**.gender();  }  }  },  **'contactPerson.$.position'**: {  **type**: String  },  **'contactPerson.$.tel'**: {  **type**: String  },  **branchId**: {  **type**: String,  **label**: **"Branch"** } }); **Rabbit**.**Collection**.**Customer**.attachSchema(**Rabbit**.**Schema**.**Customer**);   * **Customer Tabulars (customer.js)**   **Rabbit**.**TabularTable**.**Customer** = **new Tabular**.Table({  **name**: **"rabbit\_customerList"**,  **collection**: **Rabbit**.**Collection**.**Customer**,  **pagingType**: **"full\_numbers"**,  **autoWidth**: **false**,  **columnDefs**: [  {**"width"**: **"12px"**, **"targets"**: 0}  ],  **order**: [[**'1'**, **'desc'**]],  **columns**: [  {**title**: **'<i class="fa fa-bars"></i>'**, **tmpl**: Meteor.***isClient*** && ***Template***.**rabbit\_customerAction**},  {**data**: **"\_id"**, **title**: **"ID"**},  {**data**: **"companyName"**, **title**: **"Company Name"**},  {**data**: **"contractName"**, **title**: **"Contract Name"**},  {**data**: **"address"**, **title**: **"Address"**},  {**data**: **"telephone"**, **title**: **"Telephone"**},   {  **data**: **"\_contractCount"**,  **title**: **"Mis Contract <i class='fa fa-arrow-up'></i>"**,  **tmpl**: Meteor.***isClient*** && ***Template***.**rabbit\_contractLinkAction** },  {  **data**: **"\_websiteCount"**,  **title**: **"Website Contract <i class='fa fa-arrow-up'></i>"**,  **tmpl**: Meteor.***isClient*** && ***Template***.**rabbit\_websiteLinkAction** }], **extraFields**: [**'contactPerson'**, **'contractName'**, **'age'**, **'telephone'**, **'dob'**, **'id'**, **'position'**, **'gender'**, **'email'**,  **'companyName'**, **'companyAddress'**, **'companyWebsite'**, **'companyTelephone'**, **'companyEmail'**] });   * **Customer Template (customer.html)**   *<!--Index-->* <**template** name=**"rabbit\_customer"**>  <**p**>  <**button** type=**"button"** class=**"btn btn-primary js-insert"**>{{**fa "plus"**}} Add New</**button**>  </**p**>  {{> **tabular table**=**Rabbit**.**TabularTable**.**Customer selector**=**selector class**=**"table table-striped table-bordered table-condensed table-hover-pointer"**}} </**template**>  *<!--Action-->* <**template** name=**"rabbit\_customerAction"**>  {{#**tabularAction**}}  <**li**><**a** href=**"#"** class=**"js-update"**>{{**fa "pencil"**}} Update</**a**></**li**>  <**li**><**a** href=**"#"** class=**"js-remove"**>{{**fa "remove"**}} Remove</**a**></**li**>  <**li**><**a** href=**"#"** class=**"js-show"**>{{**fa "eye"**}} Show</**a**></**li**>  {{/**tabularAction**}} </**template**>  *<!--Insert-->* <**template** name=**"rabbit\_customerInsert"**>  {{#**autoForm collection**=**Rabbit**.**Collection**.**Customer id**=**"rabbit\_customerInsert" type**=**"insert"**}}  <**div** class=**"row"**>  <**div** class=**"col-md-12"**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Contract Person Information</**div**>  <**div** class=**"panel-body"**>   <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"contractName"**}} {{> **afQuickField name**=**"dob"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"gender"**}} {{> **afQuickField name**=**"telephone"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"position"**}} {{> **afQuickField name**=**"id"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"email"**}} {{> **afQuickField name**=**"address"**}}  </**div**>  </**div**>  </**div**>  </**div**>   <**div** class=**"col-md-12"**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Company Information</**div**>  <**div** class=**"panel-body"**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"companyName"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"companyTelephone"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"companyEmail"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"companyWebsite"**}}  </**div**>  <**div** class=**"col-md-12"**>  {{> **afQuickField name**=**"companyAddress"**}}  </**div**>  </**div**>  </**div**>  </**div**>  <**div** class=**"col-md-12"**>  {{> **afArrayField name**=**"contactPerson" id**=**"contactPerson" template**=**"customArrayField" subtemplate**=**"customObjectField"**}}  <**button** type=**"submit"** class=**"btn btn-primary"**>{{**fa "floppy-o"**}} Save</**button**>  <**button** type=**"reset"** class=**"btn btn-default"**>{{**fa "refresh"**}} Reset</**button**>  </**div**>  </**div**>  {{/**autoForm**}} </**template**>  *<!--Update-->* <**template** name=**"rabbit\_customerUpdate"**>  *<!--Built-in template sub ready by all-->* {{#**if Template**.**subscriptionsReady**}} {{#**autoForm collection**=**Rabbit**.**Collection**.**Customer id**=**"rabbit\_customerUpdate" doc**=**this type**=**"update"**}}  <**div** class=**"row"**>  <**div** class=**"col-md-12"**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Contract Person Information</**div**>  <**div** class=**"panel-body"**>   <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"contractName"**}} {{> **afQuickField name**=**"dob"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"gender"**}} {{> **afQuickField name**=**"telephone"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"position"**}} {{> **afQuickField name**=**"id"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"email"**}} {{> **afQuickField name**=**"address"**}}  </**div**>  </**div**>  </**div**>  </**div**>   <**div** class=**"col-md-12"**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Company Information</**div**>  <**div** class=**"panel-body"**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"companyName"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"companyTelephone"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"companyEmail"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"companyWebsite"**}}  </**div**>  <**div** class=**"col-md-12"**>  {{> **afQuickField name**=**"companyAddress"**}}  </**div**>  </**div**>  </**div**>  </**div**>  <**div** class=**"col-md-12"**>  {{> **afArrayField name**=**"contactPerson" id**=**"contactPerson" template**=**"customArrayField" subtemplate**=**"customObjectField"**}}  <**button** type=**"submit"** class=**"btn btn-primary"**>{{**fa "floppy-o"**}} Save</**button**>  <**button** type=**"reset"** class=**"btn btn-default"**>{{**fa "refresh"**}} Reset</**button**>  </**div**>  </**div**>  {{/**autoForm**}} {{**else**}} {{> **loading**}} {{/**if**}} </**template**>  *<!--Show-->* <**template** name=**"rabbit\_customerShow"**>  *<!--Customer template sub ready by all-->* {{#**if data**}}  {{#**with data**}}  <**dl** class=**"dl-horizontal"**>  <**dt**>ID:</**dt**>  <**dd**>{{**\_id**}}</**dd**>  <**dt**>Name:</**dt**>  <**dd**>{{**contractName**}}</**dd**>  <**dt**>Gender:</**dt**>  <**dd**>{{**gender**}}</**dd**>  <**dt**>Date Of Birth:</**dt**>  <**dd**>{{**dob**}}</**dd**>  <**dt**>ID Card:</**dt**>  <**dd**>{{**id**}}</**dd**>  <**dt**>Position:</**dt**>  <**dd**>{{**position**}}</**dd**>  <**dt**>Telephone:</**dt**>  <**dd**>{{**telephone**}}</**dd**>  <**dt**>Email:</**dt**>  <**dd**>{{**email**}}</**dd**>  <**dt**>Address:</**dt**>  <**dd**>{{**address**}}</**dd**>  <**dt**>Company Name:</**dt**>  <**dd**>{{**companyName**}}</**dd**>  <**dt**>Company Telephone:</**dt**>  <**dd**>{{**telephone**}}</**dd**>  <**dt**>Company Email:</**dt**>  <**dd**>{{**companyEmail**}}</**dd**>  <**dt**>Company Website:</**dt**>  <**dd**>{{**companyWebsite**}}</**dd**>  <**dt**>Company Address:</**dt**>  <**dd**>{{**companyAddress**}}</**dd**>  <**dt**>Contact Person:</**dt**>  <**dd**>{{**contactPerson**}}</**dd**>  </**dl**>  {{/**with**}}  {{**else**}}  {{> **loading**}}  {{/**if**}} </**template**>  *<!--Custom array field-->* <**template** name=**"afArrayField\_customArrayField"**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>{{**afFieldLabelText name**=**this**.**atts**.**name**}}</**div**>  <**div** class=**"panel-body"**>  <**div** class=**"row"**>  <**div** class=**"col-md-3"**>  <**p**><**b**>Name</**b**></**p**>  </**div**>  <**div** class=**"col-md-2"**>  <**p**><**b**>Gender</**b**></**p**>  </**div**>  <**div** class=**"col-md-3"**>  <**p**><**b**>Position</**b**></**p**>  </**div**>  <**div** class=**"col-md-3"**>  <**p**><**b**>Telephone</**b**></**p**>  </**div**>  <**div** class=**"col-md-1"**>  <**p**><**b**>Act</**b**></**p**>  </**div**>  </**div**>  {{#**if afFieldIsInvalid name**=**this**.**atts**.**name**}}  <**div** class=**"has-error"**>  <**span** class=**"help-block"**>{{{**afFieldMessage name**=**this**.**atts**.**name**}}}</**span**>  </**div**>  {{/**if**}} {{#**afEachArrayItem name**=**this**.**atts**.**name minCount**=**this**.**atts**.**minCount maxCount**=**this**.**atts**.**maxCount**}}  <**div** class=**"row item-list"**>  {{> **UI**.**dynamic template**=**..**/**atts**.**subtemplate**}}  <**div** class=**"col-sm-1"**>  {{#**if afArrayFieldHasMoreThanMinimum name**=**..**/**atts**.**name minCount**=**..**/**atts**.**minCount maxCount**=**..**/**atts**.**maxCount**}}  <**button** type=**"button"** class=**"btn btn-sm btn-danger autoform-remove-item btnRemove"**>  <**span** class=**"glyphicon glyphicon-minus"**></**span**>  </**button**>  {{/**if**}}  </**div**>  </**div**>  {{/**afEachArrayItem**}} {{#**if afArrayFieldHasLessThanMaximum name**=**this**.**atts**.**name minCount**=**this**.**atts**.**minCount maxCount**=**this**.**atts**.**maxCount**}}  <**div** class=**"row"**>  <**div** class=**"col-sm-1 col-md-offset-11"**>  <**button** type=**"button"** class=**"btn btn-primary autoform-add-item btn-sm btnAdd"** data-autoform-field=**"**{{**this**.**atts**.**name**}}**"** data-autoform-minCount=**"**{{**this**.**atts**.**minCount**}}**"** data-autoform-maxCount=**"**{{**this**.**atts**.**maxCount**}}**"**>  <**span** class=**"glyphicon glyphicon-plus"**></**span**>  </**button**>  </**div**>  </**div**>  {{/**if**}}  </**div**>  </**div**> </**template**>  *<!--Custom object field-->* <**template** name=**"customObjectField"**>  <**div** class=**"col-sm-3"**>  {{> **afQuickField name**=**current**.**name template**=**"bootstrap3" label**=false **class**=**"input-sm" placeholder**=**"Name"**}}  </**div**>  <**div** class=**"col-sm-2"**>  {{> **afQuickField name**=**current**.**gender template**=**"bootstrap3" label**=false **class**=**"input-sm"**}}  </**div**>  <**div** class=**"col-sm-3"**>  {{> **afQuickField name**=**current**.**position template**=**"bootstrap3" label**=false **class**=**"input-sm" placeholder**=**"Position"**}}  </**div**>  <**div** class=**"col-sm-3"**>  {{> **afQuickField name**=**current**.**tel template**=**"bootstrap3" label**=false **class**=**"input-sm" placeholder**=**"Telephone"**}}  </**div**> </**template**> <**template** name=**"rabbit\_contractLinkAction"**>  <**button** type=**"button"** class=**"btn btn-primary btn-sm contractAction"**>  {{#**if \_contractCount**}} {{**\_contractCount**}} {{**else**}} 0 {{/**if**}}  </**button**> </**template**> <**template** name=**"rabbit\_websiteLinkAction"**>  <**button** type=**"button"** class=**"btn btn-primary btn-sm websiteAction"**>  {{#**if \_websiteCount**}} {{**\_websiteCount**}} {{**else**}} 0 {{/**if**}}  </**button**> </**template**>   * **Customer Template (customer.js)**   **var *indexTpl*** = ***Template***.**rabbit\_customer**,  ***insertTpl*** = ***Template***.**rabbit\_customerInsert**,  ***updateTpl*** = ***Template***.**rabbit\_customerUpdate**,  ***showTpl*** = ***Template***.**rabbit\_customerShow**; */\*\*  \* Index  \*/* ***indexTpl***.onCreated(**function** () {  *// SEO* **SEO**.set({  **title**: **'Customer'**,  **description**: **'Description for this page'** });   *// Create new alertify* createNewAlertify([**"customer"**], {**size**: **'lg'**});  createNewAlertify([**"customerShow"**], {**size**: **'lg'**});  createNewAlertify([**"locationAddon"**], {**transition**: **'zoom'**, **size**: **'lg'**}); });  ***indexTpl***.helpers({  selector: **function** () {  **return** {**branchId**: ***Session***.get(**'currentBranch'**)};  } });  ***indexTpl***.events({  **'click .js-insert'**: **function** (e, t) {  **alertify**.customer(fa(**"plus"**, **"Customer"**), renderTemplate(***insertTpl***));  },  **'click .js-update'**: **function** (e, t) {  **alertify**.customer(fa(**"pencil"**, **"Customer"**), renderTemplate(***updateTpl***, **this**));  },  **'click .js-remove'**: **function** (e, t) {  **var** self = **this**;  **var** id = **this**.**\_id**;  **var** arr = [  {**collection**: **'Rabbit.Collection.Contract'**, **selector**: {**customerId**: id}},  {**collection**: **'Rabbit.Collection.Website'**, **selector**: {**customerId**: id}}  ];  Meteor.*call*(**'isRelationExist'**, arr, **function** (error, result) {  **if** (result) {   **alertify**.**message**(self.**\_id** + **'|'** + self.companyName + **' is in used !'**);  **return false** } **else** {  **alertify**.**confirm**(  fa(**"remove"**, **"Customer"**),  **"Are you sure to delete ["** + self.**\_id** + **"] ?"**,  **function** () {  **Rabbit**.**Collection**.**Customer**.remove(self.**\_id**, **function** (error) {  **if** (error) {  **alertify**.error(error.**message**);  } **else** {  **alertify**.success(**"Success"**);  }  });  },  **null** );  }  }  ); },  **'click .js-show'**: **function** (e, t) {  **alertify**.customerShow(fa(**"eye"**, **"Customer"**), renderTemplate(***showTpl***, **this**));  }  ,  **'click .contractAction'**: **function** () {  **FlowRouter**.go(**'rabbit.contract'**, {  **customerId**: **this**.**\_id** })   }  ,  **'click .websiteAction'**: **function** () {  **FlowRouter**.go(**'rabbit.website'**, {  **customerId**: **this**.**\_id** })  } }) ;  */\*\*  \* Insert  \*/* ***insertTpl***.onRendered(**function** () {  *configOnRender*(); });  */\*\*  \* Update  \*/* ***updateTpl***.onCreated(**function** () {  *//this.subscribe('rabbit\_customer', this.data.\_id);* });  ***updateTpl***.onRendered(**function** () {  Meteor.*setTimeout*(**function** () {  *configOnRender*();  }, 200); });   */\*\*  \* Show  \*/* ***showTpl***.onCreated(**function** () {  **this**.subscribe(**'rabbit\_customer'**, **this**.data.**\_id**); });  ***showTpl***.helpers({  data: **function** () {  **var** data = **Rabbit**.**Collection**.**Customer**.findOne(**this**.**\_id**);  **return** data;   },  contactPerson: **function** () {  **var** str = **"<table class='table table-bordered'><thead>"** +  **"<tr>"** +  **"<th>Name</th>"** +  **"<th>Gender</th>"** +  **"<th>Position</th>"** +  **"<th>Telephone</th>"** +  **"</tr>"** +  **"</thead><tbody>"**;  **this**.contactPerson.forEach(**function** (o) {  str += **'<tr>'** +  **'<td>'** + o.**name** + **'</td>'** +  **'<td>'** + o.gender + **'</td>'** +  **'<td>'** + o.**position** + **'</td>'** +  **'<td>'** + o.**tel** + **'</td>'** +  **'</tr>'** });  str += **"</tbody></table>"**;  **return new Spacebars**.SafeString(str);  } });   */\*\*  \* Hook  \*/* **AutoForm**.hooks({  *// Customer* **rabbit\_customerInsert**: {  **before**: {  insert: **function** (doc) {  doc.**branchId** = ***Session***.get(**'currentBranch'**);  **var** prefix = doc.**branchId** + **'-'**;  Meteor.*call*(**'rabbit'**, prefix);  **return** doc;  }  },  onSuccess: **function** (formType, result) {  **alertify**.success(**'Success'**);  },  onError: **function** (formType, error) {  **alertify**.error(error.**message**);  }  },  **rabbit\_customerUpdate**: {  onSuccess: **function** (formType, result) {  **alertify**.customer().close();  **alertify**.success(**'Success'**);  },  onError: **function** (formType, error) {  **alertify**.error(error.**message**);  }  } });  *// Config date picker* **var** *configOnRender* = **function** () {  *// date* **debugger**;  **var** dob = $(**'[name="dob"]'**);  **DateTimePicker**.date(dob); };  **+ Quotation**   * **Quotation Router (quotarion.js)**   **var *subs*** = **new** SubsManager();  **rabbitRoutes**.route(**'/quotation'**, {  **name**: **'rabbit.quotation'**,  **title**: **'Quotation'**,  subscriptions: **function** (params, queryParams) {  *// Customer* **this**.register(**'rabbit\_customer'**, ***subs***.subscribe(**'rabbit\_customer'**, ***Session***.get(**'currentBranch'**)));  **this**.register(**'rabbit\_product'**, ***subs***.subscribe(**'rabbit\_product'**));  **this**.register(**'rabbit\_contractor'**, ***subs***.subscribe(**'rabbit\_contractor'**));  },  action: **function** (params, queryParams) {  **Layout**.main(**'rabbit\_quotation'**);  },  **breadcrumb**: {  **params**: [**'customerId'**],  *//queryParams: ['show', 'color'],* **title**: **'Quotation'**,  **icon**:**'sticky-note'**,  **parent**: **'rabbit.home'** } });   * **Quotation Collection (quotarion.js)**   **Rabbit**.**Collection**.**Quotation** = **new** Mongo.Collection(**"rabbit\_quotation"**);  *// Schema* **Rabbit**.**Schema**.**Quotation** = **new *SimpleSchema***({  **quotationDate**: {  **type**: String,  **label**: **"Quotation Date"**,  defaultValue: **function** () {  **var** currentDate = **moment**(**ReactiveMethod**.call(**"currentDate"**), **'YYYY-MM-DD'**).**format**(**'YYYY-MM-DD'**);  **return** currentDate;  }  },  **customerId**: {  **type**: String,  **autoform**: {  **type**: **'select2'**,  options(){  **return Rabbit**.**List**.customer();  },  **afFieldInput**: {  **select2Options**: {  **theme**: **"bootstrap"** }  }  }  },  **productId**: {  **type**: String,  **label**: **"Product"**,  **autoform**: {  **type**: **'select2'**,  options(){  **return Rabbit**.**List**.product();  },  **afFieldInput**: {  **select2Options**: {  **theme**: **"bootstrap"** }  }  }  },  **monthlyFee**: {  **type**: Array,  *//label: "Branch Price",* **minCount**: 1,  **maxCount**: 1,  **optional**: **true** },  **'monthlyFee.$'**: {  **type**: Object,  **decimal**: **true** },  **'monthlyFee.$.headOffice'**: {  **type**: Number,  **decimal**: **true** },  **'monthlyFee.$.branch'**: {  **type**: Number,  **decimal**: **true** },  **type**: {  **type**: String,  **label**: **'ProductType'**,  **autoform**: {  **type**: **'select2'**,  options(){  **return Rabbit**.**List**.type();  },  **afFieldInput**: {  **select2Options**: {  **theme**: **"bootstrap"** }  }  }  },  **installationFee**: {  **type**: Number,  **label**: **'InstallationFee'**,  **decimal**: **true**,  **optional**: **true** },  **trainingFee**: {  **type**: Number,  **label**: **'TrainingFee'**,  **decimal**: **true**,  **optional**: **true** },  **basePrice**: {  **type**: Array,  *//label: "Branch Price",* **minCount**: 1,  **maxCount**: 1,  **optional**: **true** },  **'basePrice.$'**: {  **type**: Object  },  **'basePrice.$.headOffice'**: {  **type**: Number,  **decimal**: **true** },  **'basePrice.$.branch'**: {  **type**: Number,  **decimal**: **true** },  **maintenancePrice**: {  **type**: Array,  **minCount**: 1,  **maxCount**: 1,  **optional**: **true** },  **'maintenancePrice.$'**: {  **type**: Object  },  **'maintenancePrice.$.headOffice'**: {  **type**: Number,  **decimal**: **true** },  **'maintenancePrice.$.branch'**: {  **type**: Number,  **decimal**: **true** },   **branchId**: {  **type**: String,  **label**: **"Branch"** }  ,  **des**: {  **type**: String,  **label**: **"Description"**,  **autoform**: {  **afFieldInput**: {  **type**: **"textarea"** }  },  **optional**: **true** },  **contractorId**: {  **type**: String,  **label**: **"Contractor"**,  **autoform**: {  **type**: **'select2'**,  options(){  **return Rabbit**.**List**.contractors();  },  **afFieldInput**: {  **select2Options**: {  **theme**: **"bootstrap"** }  }  }  }   }) ; *// Attach schema* **Rabbit**.**Collection**.**Quotation**.attachSchema(**Rabbit**.**Schema**.**Quotation**);   * **Quotation Tabulars (quotarion.js)**   **Rabbit**.**TabularTable**.**Quotation** = **new Tabular**.Table({  **name**: **"rabbit\_quotationLists"**,  **collection**: **Rabbit**.**Collection**.**Quotation**,  **pagingType**: **"full\_numbers"**,  **autoWidth**: **false**,  **columnDefs**: [  {**"width"**: **"12px"**, **"targets"**: 0}  ],  **order**: [[**'1'**, **'desc'**]],  **columns**: [  {**title**: **'<i class="fa fa-bars"></i>'**, **tmpl**: Meteor.***isClient*** && ***Template***.**rabbit\_quotationAction**},  {**data**: **"\_id"**, **title**: **"ID"**},  {**data**: **"\_customer.companyName"**, **title**: **"Customer"**},  {  **data**: **"quotationDate"**, **title**: **"Quotation Date"**,  render: **function** (val, type, doc) {  **return moment**(val).**format**(**'YYYY-MM-DD'**);  }  },  {**data**: **"type"**, **title**: **"Product Type"**},  {**data**: **"productId"**, **title**: **"Product ID"**}   ],  **extraFields**: [**'customerId'**, **'quotationDate'**, **'basePrice'**, **'maintenancePrice'**, **'paymentMethod'**, **'type'**, **'testing'**, **'maintenanceFee'**, **'des'**, **'\_product'**, **'\_customer'**, **'addFile'**, **'contractorId'**, **'monthlyFee'**, **'installationFee'**, **'trainingFee'**] });   * **Quotation Template (quotarion.html)**   <**template** name=**"rabbit\_quotation"**>  <**p**>  <**button** type=**"button"** class=**"btn btn-primary js-insert"**>{{**fa "plus"**}} Add New</**button**>  </**p**>  <**div** class=**"pull-right"** style=**"**display: **inline"**>  <**small**>  {{#**with customer**}}  <**span** class=**"jsInfo"**>{{**\_id**}} | QuotationPerson: {{**quotationName**}} | Tel: {{**telephone**}}  | Company: {{**companyName**}} |  CompanyTel: {{**companyTelephone**}}</**span**>  {{/**with**}}  </**small**>  </**div**>  {{> **tabular table**=**Rabbit**.**TabularTable**.**Quotation selector**=**selector class**=**"table table-striped table-bordered table-condensed table-hover-pointer"**}} </**template**>  *<!--Action-->* <**template** name=**"rabbit\_quotationAction"**>  {{#**tabularAction**}}  <**li**><**a** href=**"#"** class=**"js-update"**>{{**fa "pencil"**}} Update</**a**></**li**>  <**li**><**a** href=**"#"** class=**"js-remove"**>{{**fa "remove"**}} Remove</**a**></**li**>  <**li**><**a** href=**"#"** class=**"js-show"**>{{**fa "eye"**}} Show</**a**></**li**>  <**b**>  <**li** role=**"separator"** class=**"divider"**></**li**>  </**b**>  <**li**><**a** href=**"/rabbit/quotation/printQuotation/**{{**this**.**\_id**}}**"** target=**"\_blank"**>{{**fa "print"**}} Print Quotation</**a**>  </**li**>  {{/**tabularAction**}} </**template**>  *<!--Insert-->* <**template** name=**"rabbit\_quotationInsert"**>  {{#**autoForm collection**=**Rabbit**.**Collection**.**Quotation id**=**"rabbit\_quotationInsert" type**=**"insert" doc**=**quotationData**}}  <**div** class=**"row"**>  <**div** class=**"col-md-2"**>  {{> **afQuickField name**=**"quotationDate"**}}  </**div**>  <**div** class=**"col-md-3"**>  <**div** class=**"form-group**{{#**if afFieldIsInvalid name**=**'customerId'**}} **has-error**{{/**if**}}**"** data-required=**"true"**>  <**label** class=**"control-label"** for=**"customerId"**>Customer</**label**>  *<!--Customer Add On-->* <**div** class=**"input-group select2-bootstrap-append"**>  {{> **afFieldInput name**=**'customerId'**}}  <**span** class=**"input-group-btn"**>  <**a** href=**"#"** class=**"customerAddon btn btn-default"**>  {{**fa "plus"**}}  </**a**>  </**span**>  </**div**>  </**div**>  {{#**if afFieldIsInvalid name**=**'customerId'**}}  <**span** class=**"help-block"**>{{**afFieldMessage name**=**'customerId'**}}</**span**> {{/**if**}}  </**div**>  <**div** class=**"col-md-2"**>  {{> **afQuickField name**=**"contractorId"**}}  </**div**>  <**div** class=**"col-md-2"**>  {{> **afQuickField name**=**"type" class**=**"type"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"productId" class**=**'productId'**}}  </**div**>   <**div** class=**"col-md-6 monthly"**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Monthly Fee</**div**>  <**div** class=**"panel-body"**>  <**div** class=**"col-md-12"**>  {{> **afArrayField name**=**"monthlyFee" id**=**"monthlyFee" template**=**"basePriceArrayField" label**=false **subtemplate**=**"quotationMonthlyFeeObjectField"**}}  </**div**>  <**div** class=**"col-md-12"**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Service</**div**>  <**div** class=**"panel-body"**>  <**div** class=**"col-sm-6-custom-left"**>  {{> **afQuickField name**=**"installationFee" class**=**"input-sm installationFee" placeholder**=**"Installation Fee"**}}  </**div**>  <**div** class=**"col-sm-6-custom-right"**>  {{> **afQuickField name**=**"trainingFee" class**=**"input-sm trainingFee" placeholder**=**"Training Fee"**}}  </**div**>  </**div**>  </**div**>  </**div**>  </**div**>  </**div**>  </**div**>  <**div** class=**"col-md-6 fully"**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Fully Fee</**div**>  <**div** class=**"panel-body"**>  <**div** class=**"col-md-12"**>  {{> **afArrayField name**=**"basePrice" template**=**"quotationBasePriceArrayField" subtemplate**=**"quotationBasePriceObjectField" id**=**"basePrice"**}}  </**div**>  <**div** class=**"col-md-12"**>  {{> **afArrayField name**=**"maintenancePrice" id**=**"maintenancePrice" template**=**"basePriceArrayField" subtemplate**=**"quotationMaintenancePriceObjectField"** }}  </**div**>  </**div**>  </**div**>  </**div**>  <**div** class=**"col-md-12"**>  {{> **afQuickField name**=**"des"**}}  </**div**>  </**div**>  <**button** type=**"submit"** class=**"btn btn-primary"**>{{**fa "floppy-o"**}} Save</**button**>  <**button** type=**"reset"** class=**"btn btn-default"**>{{**fa "refresh"**}} Reset</**button**>  {{/**autoForm**}} </**template**>  *<!--Update-->* <**template** name=**"rabbit\_quotationUpdate"**>  *<!--Built-in template sub ready by all-->* {{#**if Template**.**subscriptionsReady**}} {{#**autoForm collection**=**Rabbit**.**Collection**.**Quotation id**=**"rabbit\_quotationUpdate" doc**=**this type**=**"update"**}}  <**div** class=**"row"**>  <**div** class=**"col-md-2"**>  {{> **afQuickField name**=**"quotationDate"**}}  </**div**>  <**div** class=**"col-md-3"**>  <**div** class=**"form-group**{{#**if afFieldIsInvalid name**=**'customerId'**}} **has-error**{{/**if**}}**"** data-required=**"true"**>  <**label** class=**"control-label"** for=**"customerId"**>Customer</**label**>  *<!--Customer Add On-->* <**div** class=**"input-group select2-bootstrap-append"**>  {{> **afFieldInput name**=**'customerId'**}}  <**span** class=**"input-group-btn"**>  <**a** href=**"#"** class=**"customerAddon btn btn-default"**>  {{**fa "plus"**}}  </**a**>  </**span**>  </**div**>  </**div**>  {{#**if afFieldIsInvalid name**=**'customerId'**}}  <**span** class=**"help-block"**>{{**afFieldMessage name**=**'customerId'**}}</**span**> {{/**if**}}  </**div**>  <**div** class=**"col-md-2"**>  {{> **afQuickField name**=**"contractorId"**}}  </**div**>  <**div** class=**"col-md-2"**>  {{> **afQuickField name**=**"type" class**=**"type"**}}  </**div**>  <**div** class=**"col-md-3"**>  {{> **afQuickField name**=**"productId" class**=**'productId'**}}  </**div**>   <**div** class=**"col-md-6 monthly"**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Monthly Fee</**div**>  <**div** class=**"panel-body"**>  <**div** class=**"col-md-12"**>  {{> **afArrayField name**=**"monthlyFee" id**=**"monthlyFee" template**=**"basePriceArrayField" label**=false **subtemplate**=**"quotationMonthlyFeeObjectField"**}}  </**div**>  <**div** class=**"col-md-12"**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Service</**div**>  <**div** class=**"panel-body"**>  <**div** class=**"col-sm-6-custom-left"**>  {{> **afQuickField name**=**"installationFee" class**=**"input-sm installationFee" placeholder**=**"Installation Fee"**}}  </**div**>  <**div** class=**"col-sm-6-custom-right"**>  {{> **afQuickField name**=**"trainingFee" class**=**"input-sm trainingFee" placeholder**=**"Training Fee"**}}  </**div**>  </**div**>  </**div**>  </**div**>  </**div**>  </**div**>  </**div**>  <**div** class=**"col-md-6 fully"**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>Fully Fee</**div**>  <**div** class=**"panel-body"**>  <**div** class=**"col-md-12"**>  {{> **afArrayField name**=**"basePrice" template**=**"quotationBasePriceArrayField" subtemplate**=**"quotationBasePriceObjectField" id**=**"basePrice"**}}  </**div**>  <**div** class=**"col-md-12"**>  {{> **afArrayField name**=**"maintenancePrice" id**=**"maintenancePrice" template**=**"basePriceArrayField" subtemplate**=**"quotationMaintenancePriceObjectField"** }}  </**div**>  </**div**>  </**div**>  </**div**>  <**div** class=**"col-md-12"**>  {{> **afQuickField name**=**"des"**}}  </**div**>  </**div**>  <**button** type=**"submit"** class=**"btn btn-primary"**>{{**fa "floppy-o"**}} Save</**button**>  <**button** type=**"reset"** class=**"btn btn-default"**>{{**fa "refresh"**}} Reset</**button**>  {{/**autoForm**}} {{**else**}} {{> **loading**}} {{/**if**}} </**template**>  *<!--Show-->* <**template** name=**"rabbit\_quotationShow"**>  *<!--Quotation template sub ready by all-->* {{#**if data**}}  {{#**with data**}}  <**dl** class=**"dl-horizontal"**>  <**dt**>Customer:</**dt**>  <**dd**>{{**customerId**}}</**dd**>  <**dt**>Quotation Date:</**dt**>  <**dd**>{{**quotationDate**}}</**dd**>  <**dt**>Product:</**dt**>  <**dd**>{{**\_product**.**name**}}</**dd**>  <**dt**>Base Price:</**dt**>  <**dd**>{{**base**}}</**dd**>  <**dt**>Maintenance Price:</**dt**>  <**dd**>{{**maintenance**}}</**dd**>  <**dt**>Monthly Fee Price:</**dt**>  <**dd**>{{{**monthlyFee**}}}</**dd**>  <**dt**>InstallationFee:</**dt**>  <**dd**>{{**installationFee**}}</**dd**>  <**dt**>TrainingFee:</**dt**>  <**dd**>{{**trainingFee**}}</**dd**>  <**dt**>Description:</**dt**>  <**dd**>{{**des**}}</**dd**>  </**dl**>  {{/**with**}}  {{**else**}}  {{> **loading**}}  {{/**if**}} </**template**>  <**template** name=**"afArrayField\_quotationBasePriceArrayField"**>  <**div** class=**"panel panel-default"**>  <**div** class=**"panel-heading"**>{{**afFieldLabelText name**=**this**.**atts**.**name**}}</**div**>  <**div** class=**"panel-body"**>  <**div** class=**"col-sm-6-custom"**>  <**p**><**b**>Head Price</**b**></**p**>  </**div**>  <**div** class=**"col-sm-6"**>  <**p**><**b**>Branch Price</**b**></**p**>  </**div**>  {{#**if afFieldIsInvalid name**=**this**.**atts**.**name**}}  <**div** class=**"has-error"**>  <**span** class=**"help-block"**>{{{**afFieldMessage name**=**this**.**atts**.**name**}}}</**span**>  </**div**>  {{/**if**}} {{#**afEachArrayItem name**=**this**.**atts**.**name minCount**=**this**.**atts**.**minCount maxCount**=**this**.**atts**.**maxCount**}}  <**div** class=**"row item-list"**>  {{> **UI**.**dynamic template**=**..**/**atts**.**subtemplate**}}   <**div** class=**"col-sm-1"**>  {{#**if afArrayFieldHasMoreThanMinimum name**=**..**/**atts**.**name minCount**=**..**/**atts**.**minCount maxCount**=**..**/**atts**.**maxCount**}}  <**button** type=**"button"** class=**"btn btn-sm btn-danger autoform-remove-item btnRemove"**>  <**span** class=**"glyphicon glyphicon-minus"**></**span**>  </**button**>  {{/**if**}}  </**div**>  </**div**>  {{/**afEachArrayItem**}} {{#**if afArrayFieldHasLessThanMaximum name**=**this**.**atts**.**name minCount**=**this**.**atts**.**minCount maxCount**=**this**.**atts**.**maxCount**}}  <**div** class=**"row"**>  <**div** class=**"col-sm-1 pull-right"**>  <**button** type=**"button"** class=**"btn btn-primary autoform-add-item btn-sm btnAdd"** data-autoform-field=**"**{{**this**.**atts**.**name**}}**"** data-autoform-minCount=**"**{{**this**.**atts**.**minCount**}}**"** data-autoform-maxCount=**"**{{**this**.**atts**.**maxCount**}}**"**>  <**span** class=**"glyphicon glyphicon-plus"**></**span**>  </**button**>  </**div**>  </**div**>  {{/**if**}}  </**div**>  </**div**> </**template**>   <**template** name=**"quotationBasePriceObjectField"**>  <**div** class=**"col-sm-6"**>  {{> **afQuickField name**=**current**.**headOffice id**=**'basePriceHeadOffice' template**=**"bootstrap3" label**=false **class**=**"input-sm" placeholder**=**"Head Office"**}}  </**div**>  <**div** class=**"col-sm-6"**>  {{> **afQuickField name**=**current**.**branch id**=**'basePriceBranch' template**=**"bootstrap3" label**=false **class**=**"input-sm" placeholder**=**"Branch Office"**}}  </**div**> </**template**>  <**template** name=**"quotationMaintenancePriceObjectField"**>  <**div** class=**"col-sm-6"**>  {{> **afQuickField name**=**current**.**headOffice template**=**"bootstrap3" id**=**'MaintenaceHeadOffice' label**=false **class**=**"input-sm" placeholder**=**"Head Office"**}}  </**div**>  <**div** class=**"col-sm-6"**>  {{> **afQuickField name**=**current**.**branch id**=**'MaintenaceBranch' template**=**"bootstrap3" label**=false **class**=**"input-sm" placeholder**=**"Branch Office"**}}  </**div**> </**template**>  <**template** name=**"quotationMonthlyFeeObjectField"**>  <**div** class=**"col-sm-6"**>  {{> **afQuickField name**=**current**.**headOffice template**=**"bootstrap3" id**=**'monthlyFeeHeadOffice' label**=false **class**=**"input-sm" placeholder**=**"Head Office"**}}  </**div**>  <**div** class=**"col-sm-6"**>  {{> **afQuickField name**=**current**.**branch id**=**'monthlyFeeBranch' template**=**"bootstrap3" label**=false **class**=**"input-sm" placeholder**=**"Branch Office"**}}  </**div**> </**template**>   * **Quotation Template (quotarion.js)**   **var *indexTpl*** = ***Template***.**rabbit\_quotation**,  ***insertTpl*** = ***Template***.**rabbit\_quotationInsert**,  ***updateTpl*** = ***Template***.**rabbit\_quotationUpdate**,  ***showTpl*** = ***Template***.**rabbit\_quotationShow**;  *//locationAddOnTpl = Template.rabbit\_locationAddOnQuotation; /\*\*  \* Index  \*/* ***indexTpl***.onCreated(**function** () {  *// SEO* **SEO**.set({  **title**: **'Quotation'**,  **description**: **'Description for this page'** });   *// Create new alertify* createNewAlertify([**"quotation"**], {**size**: **'lg'**});  createNewAlertify([**"quotationShow"**]);  createNewAlertify([**"locationAddon"**], {**transition**: **'zoom'**, **size**: **'lg'**}); }); ***indexTpl***.helpers({  selector: **function** () {  **let** id = **FlowRouter**.getParam(**'customerId'**);  *//return FlowRouter.getParam('customerId');* **return** {**customerId**: id}  },  customer: **function** () {  *//let id = FlowRouter.getParam('customerId');* **let** customer = **Rabbit**.**Collection**.**Customer**.findOne({**\_id**: **'001-000001'**});  **return** customer   } }); ***indexTpl***.events({  **'click .js-insert'**: **function** (e, t) {  **alertify**.quotation(fa(**"plus"**, **"Quotation"**), renderTemplate(***insertTpl***));  },  **'click .js-update'**: **function** (e, t) {  **alertify**.quotation(fa(**"pencil"**, **"Quotation"**), renderTemplate(***updateTpl***, **this**));  },  **'click .js-remove'**: **function** (e, t) {  **var** self = **this**;   **alertify**.**confirm**(  fa(**"remove"**, **"Quotation"**),  **"Are you sure to delete ["** + self.**\_id** + **"] ?"**,  **function** () {  **Rabbit**.**Collection**.**Quotation**.remove(self.**\_id**, **function** (error) {  **if** (error) {  **alertify**.error(error.**message**);  } **else** {  **alertify**.success(**"Success"**);  }  });  },  **null** );  },  **'click .js-show'**: **function** (e, t) {  **alertify**.quotationShow(fa(**"eye"**, **"Quotation"**), renderTemplate(***showTpl***, **this**));  },  **'click .officeAction'**: **function** () {  **FlowRouter**.go(**'rabbit.office'**, {  **customerId**: **this**.**customerId**, **quotationId**: **this**.**\_id** });  } });  */\*\*  \* Insert  \*/* ***insertTpl***.onCreated(**function** () {    Meteor.*subscribe*(**'rabbit\_customer'**); }); ***insertTpl***.onRendered(**function** () {  *//Meteor.typeahead.inject();* Meteor.*setTimeout*(**function** () {  $(**'#basePriceHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#basePriceBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'.productId'**).attr(**'disabled'**, **"disabled"**);  $(**'.installationFee'**).attr(**'disabled'**, **"disabled"**);  $(**'.trainingFee'**).attr(**'disabled'**, **"disabled"**);  *configOnRender*();  }, 100); }); ***insertTpl***.helpers({  quotationData(){  **debugger**;  **return** {  **quotationDate**: **moment**(**ReactiveMethod**.call(**"currentDate"**), **'YYYY-MM-DD'**).**format**(**'YYYY-MM-DD'**),  **customerId**: **Rabbit**.**Collection**.**Customer**.findOne({**\_id**: **'001-000001'**}).**\_id**,  **contractorId**: **Rabbit**.**Collection**.**Contractor**.findOne({**\_id**: **'001-01'**}).**\_id** }   },  search: **function** (query, sync, callback) {  Meteor.*call*(**'searchCustomer'**, query, {}, **function** (err, res) {  **if** (err) {  **console**.log(err);  **return**;  }  callback(res);  });  },  selected: **function** (event, suggestion, dataSetName) {  *// event - the jQuery event object  // suggestion - the suggestion object  // datasetName - the name of the dataset the suggestion belongs to  // TODO your event handler here* **var** id = suggestion.**\_id**;  **var** selector = {**\_id**: id};  **var** data = getValidatedValues();  **if** (data.**valid**) {  checkBeforeAddOrUpdate(selector, data);  } **else** {  **alertify**.**warning**(data.**message**);  }  $(**'#product-barcode'**).**focus**();  }, }); ***insertTpl***.events({  **'click .customerAddon'**: **function** () {  **alertify**.quotationShow(fa(**"plus"**, **"Customer"**), renderTemplate(***Template***.**rabbit\_customerInsert**)).**maximize**()  },  **'change .productId'**: **function** (e, t) {  **let** productId = $(e.currentTarget).val();  **let** type = $(**'.type'**).val();   **let** product = Rabbit.Collection.Product.findOne({\_id: productId});  **if** (type == **'fullyFee'**) {  **if** (product) {  $(**'#basePriceHeadOffice'**).val(product.basePrice[0].headOffice);  $(**'#basePriceBranch'**).val(product.basePrice[0].branch);  $(**'#MaintenaceHeadOffice'**).val(product.maintenancePrice[0].headOffice);  $(**'#MaintenaceBranch'**).val(product.maintenancePrice[0].branch);  $(**'#monthlyFeeHeadOffice'**).val(**''**);  $(**'#monthlyFeeBranch'**).val(**''**);  } **else if** (productId == **""**) {  $(**'#basePriceHeadOffice'**).val(**""**);  $(**'#basePriceBranch'**).val(**""**);  $(**'#MaintenaceHeadOffice'**).val(**""**);  $(**'#MaintenaceBranch'**).val(**""**);  $(**'#monthlyFeeHeadOffice'**).val(**''**);  $(**'#monthlyFeeBranch'**).val(**''**);  $(**'.installationFee'**).val(**''**);  $(**'.trainingFee'**).val(**''**);  }  } **else if** (type == **"monthlyFee"**) {  **if** (product) {  $(**'#monthlyFeeHeadOffice'**).val(product.monthlyFee[0].headOffice);  $(**'#monthlyFeeBranch'**).val(product.monthlyFee[0].branch);  $(**'.trainingFee'**).val(product.trainingFee);  $(**'.installationFee'**).val(product.installationFee);  $(**'#basePriceHeadOffice'**).val(**""**);  $(**'#basePriceBranch'**).val(**""**);  $(**'#MaintenaceHeadOffice'**).val(**""**);  $(**'#MaintenaceBranch'**).val(**""**);  } **else if** (productId == **""**) {  $(**'#basePriceHeadOffice'**).val(**""**);  $(**'#basePriceBranch'**).val(**""**);  $(**'#MaintenaceHeadOffice'**).val(**""**);  $(**'#MaintenaceBranch'**).val(**""**);  $(**'#monthlyFeeHeadOffice'**).val(**''**);  $(**'#monthlyFeeBranch'**).val(**''**);  $(**'.installationFee'**).val(**''**);  $(**'.trainingFee'**).val(**''**);  }  }  },  **'keypress #basePriceHeadOffice,#basePriceBranch,#MaintenaceHeadOffice,#MaintenaceBranch,#monthlyFeeHeadOffice,#monthlyFeeBranch,.trainingFee,.installationFee'**: **function** (evt) {  **var** charCode = (evt.which) ? evt.which : evt.keyCode;  **if** ($(evt.currentTarget).val().indexOf(**'.'**) != -1) {  **if** (charCode == 46) {  **return false**;  }  }  **return** !(charCode != 46 && charCode > 31 && (charCode < 48 || charCode > 57));  },  **"change .type"**: **function** (e, t) {  **let** type = $(**'.type'**).val()  **if** (type == **'fullyFee'**) {  $(**'#basePriceHeadOffice'**).val(**""**);  $(**'#basePriceBranch'**).val(**""**);  $(**'#MaintenaceHeadOffice'**).val(**""**);  $(**'#MaintenaceBranch'**).val(**""**);  $(**'#monthlyFeeHeadOffice'**).val(**''**);  $(**'#monthlyFeeBranch'**).val(**''**);  $(**'.installationFee'**).val(**''**);  $(**'.trainingFee'**).val(**''**);  $(**'#basePriceHeadOffice'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'#basePriceBranch'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceHeadOffice'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceBranch'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'.installationFee'**).attr(**'disabled'**, **"disabled"**);  $(**'.trainingFee'**).attr(**'disabled'**, **"disabled"**);  $(**'.productId'**).removeAttr(**'disabled'**, **"disabled"**);  } **else if** (type == **"monthlyFee"**) {  $(**'#basePriceHeadOffice'**).val(**""**);  $(**'#basePriceBranch'**).val(**""**);  $(**'#MaintenaceHeadOffice'**).val(**""**);  $(**'#MaintenaceBranch'**).val(**""**);  $(**'#monthlyFeeHeadOffice'**).val(**''**);  $(**'#monthlyFeeBranch'**).val(**''**);  $(**'.installationFee'**).val(**''**);  $(**'.trainingFee'**).val(**''**);  $(**'#basePriceHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#basePriceBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeHeadOffice'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeBranch'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'.trainingFee'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'.installationFee'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'.productId'**).removeAttr(**'disabled'**, **"disabled"**);  } **else** {  $(**'#basePriceHeadOffice'**).val(**""**);  $(**'#basePriceBranch'**).val(**""**);  $(**'#MaintenaceHeadOffice'**).val(**""**);  $(**'#MaintenaceBranch'**).val(**""**);  $(**'#monthlyFeeHeadOffice'**).val(**''**);  $(**'#monthlyFeeBranch'**).val(**''**);  $(**'.installationFee'**).val(**''**);  $(**'.trainingFee'**).val(**''**);  $(**'#basePriceHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#basePriceBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'.productId'**).attr(**'disabled'**, **"disabled"**);  $(**'.installationFee'**).attr(**'disabled'**, **"disabled"**);  $(**'.trainingFee'**).attr(**'disabled'**, **"disabled"**);  }  } });  */\*\*  \* Update  \*/ //updateTpl.onCreated(function () { // this.subscribe('rabbit\_quotation', this.data.\_id); //});* updateTpl.onRendered(**function** () {  Meteor.setTimeout(**function** () {  $(**'#basePriceHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#basePriceBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'.productId'**).attr(**'disabled'**, **"disabled"**);  $(**'.trainingFee'**).attr(**'disabled'**, **"disabled"**);  $(**'.installationFee'**).attr(**'disabled'**, **"disabled"**);  configOnRender();  }, 100); });  updateTpl.events({  **'click .customerAddon'**: **function** () {  alertify.quotationShow(fa(**"plus"**, **"Customer"**), renderTemplate(Template.rabbit\_customerInsert)).maximize()  },  **'change .productId'**: **function** (e, t) {  **let** productId = $(e.currentTarget).val();  **let** type = $(**'.type'**).val();   **let** product = Rabbit.Collection.Product.findOne({\_id: productId});  **if** (type == **'fullyFee'**) {  **if** (product) {  $(**'#basePriceHeadOffice'**).val(product.basePrice[0].headOffice);  $(**'#basePriceBranch'**).val(product.basePrice[0].branch);  $(**'#MaintenaceHeadOffice'**).val(product.maintenancePrice[0].headOffice);  $(**'#MaintenaceBranch'**).val(product.maintenancePrice[0].branch);  $(**'#monthlyFeeHeadOffice'**).val(**''**);  $(**'#monthlyFeeBranch'**).val(**''**);  } **else if** (productId == **""**) {  $(**'#basePriceHeadOffice'**).val(**""**);  $(**'#basePriceBranch'**).val(**""**);  $(**'#MaintenaceHeadOffice'**).val(**""**);  $(**'#MaintenaceBranch'**).val(**""**);  $(**'#monthlyFeeHeadOffice'**).val(**''**);  $(**'#monthlyFeeBranch'**).val(**''**);  $(**'.installationFee'**).val(**''**);  $(**'.trainingFee'**).val(**''**);  }  } **else if** (type == **"monthlyFee"**) {  **if** (product) {  $(**'#monthlyFeeHeadOffice'**).val(product.monthlyFee[0].headOffice);  $(**'#monthlyFeeBranch'**).val(product.monthlyFee[0].branch);  $(**'.trainingFee'**).val(product.trainingFee);  $(**'.installationFee'**).val(product.installationFee);  $(**'#basePriceHeadOffice'**).val(**""**);  $(**'#basePriceBranch'**).val(**""**);  $(**'#MaintenaceHeadOffice'**).val(**""**);  $(**'#MaintenaceBranch'**).val(**""**);  } **else if** (productId == **""**) {  $(**'#basePriceHeadOffice'**).val(**""**);  $(**'#basePriceBranch'**).val(**""**);  $(**'#MaintenaceHeadOffice'**).val(**""**);  $(**'#MaintenaceBranch'**).val(**""**);  $(**'#monthlyFeeHeadOffice'**).val(**''**);  $(**'#monthlyFeeBranch'**).val(**''**);  $(**'.installationFee'**).val(**''**);  $(**'.trainingFee'**).val(**''**);  }  }  },  **'keypress #basePriceHeadOffice,#basePriceBranch,#MaintenaceHeadOffice,#MaintenaceBranch,#monthlyFeeHeadOffice,#monthlyFeeBranch,.trainingFee,.installationFee'**: **function** (evt) {  **var** charCode = (evt.which) ? evt.which : evt.keyCode;  **if** ($(evt.currentTarget).val().indexOf(**'.'**) != -1) {  **if** (charCode == 46) {  **return false**;  }  }  **return** !(charCode != 46 && charCode > 31 && (charCode < 48 || charCode > 57));  },  **"change .type"**: **function** (e, t) {  **let** type = $(**'.type'**).val()  **if** (type == **'fullyFee'**) {  $(**'#basePriceHeadOffice'**).val(**""**);  $(**'#basePriceBranch'**).val(**""**);  $(**'#MaintenaceHeadOffice'**).val(**""**);  $(**'#MaintenaceBranch'**).val(**""**);  $(**'#monthlyFeeHeadOffice'**).val(**''**);  $(**'#monthlyFeeBranch'**).val(**''**);  $(**'.installationFee'**).val(**''**);  $(**'.trainingFee'**).val(**''**);  $(**'#basePriceHeadOffice'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'#basePriceBranch'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceHeadOffice'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceBranch'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'.installationFee'**).attr(**'disabled'**, **"disabled"**);  $(**'.trainingFee'**).attr(**'disabled'**, **"disabled"**);  $(**'.productId'**).removeAttr(**'disabled'**, **"disabled"**);  } **else if** (type == **"monthlyFee"**) {  $(**'#basePriceHeadOffice'**).val(**""**);  $(**'#basePriceBranch'**).val(**""**);  $(**'#MaintenaceHeadOffice'**).val(**""**);  $(**'#MaintenaceBranch'**).val(**""**);  $(**'#monthlyFeeHeadOffice'**).val(**''**);  $(**'#monthlyFeeBranch'**).val(**''**);  $(**'.installationFee'**).val(**''**);  $(**'.trainingFee'**).val(**''**);  $(**'#basePriceHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#basePriceBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeHeadOffice'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeBranch'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'.trainingFee'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'.installationFee'**).removeAttr(**'disabled'**, **"disabled"**);  $(**'.productId'**).removeAttr(**'disabled'**, **"disabled"**);  } **else** {  $(**'#basePriceHeadOffice'**).val(**""**);  $(**'#basePriceBranch'**).val(**""**);  $(**'#MaintenaceHeadOffice'**).val(**""**);  $(**'#MaintenaceBranch'**).val(**""**);  $(**'#monthlyFeeHeadOffice'**).val(**''**);  $(**'#monthlyFeeBranch'**).val(**''**);  $(**'.installationFee'**).val(**''**);  $(**'.trainingFee'**).val(**''**);  $(**'#basePriceHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#basePriceBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#MaintenaceBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeHeadOffice'**).attr(**'disabled'**, **"disabled"**);  $(**'#monthlyFeeBranch'**).attr(**'disabled'**, **"disabled"**);  $(**'.productId'**).attr(**'disabled'**, **"disabled"**);  $(**'.installationFee'**).attr(**'disabled'**, **"disabled"**);  $(**'.trainingFee'**).attr(**'disabled'**, **"disabled"**);  }  } }); */\*\*  \* Show  \*/* showTpl.onCreated(**function** () {  **this**.subscribe(**'rabbit\_quotation'**, **this**.data.\_id); }); showTpl.helpers({  data: **function** () {  **var** data = Rabbit.Collection.Quotation.findOne(**this**.\_id);  **return** data;  },  base: **function** () {  **var** str = **"<table class='table table-bordered'><thead>"** +  **"<tr>"** +  **"<th>Head Office</th>"** +  **"<th>Branch</th>"** +  **"</tr>"** +  **"</thead><tbody>"**;  **this**.basePrice.forEach(**function** (o) {  str += **'<tr>'** +  **'<td>'** + o.headOffice + **'</td>'** +  **'<td>'** + o.branch + **'</td>'** +  **'</tr>'** });  str += **"</tbody></table>"**;  **return new** Spacebars.SafeString(str);  },  maintenance: **function** () {  **var** str = **"<table class='table table-bordered'><thead>"** +  **"<tr>"** +  **"<th>Head Office</th>"** +  **"<th>Branch</th>"** +  **"</tr>"** +  **"</thead><tbody>"**;  **this**.maintenancePrice.forEach(**function** (o) {  str += **'<tr>'** +  **'<td>'** + o.headOffice + **'</td>'** +  **'<td>'** + o.branch + **'</td>'** +  **'</tr>'** });  str += **"</tbody></table>"**;  **return new** Spacebars.SafeString(str);  },  monthlyFee: **function** () {  **var** str = **"<table class='table table-bordered'><thead>"** +  **"<tr>"** +  **"<th>Head Office</th>"** +  **"<th>Branch</th>"** +  **"</tr>"** +  **"</thead><tbody>"**;  **this**.monthlyFee.forEach(**function** (o) {  str += **'<tr>'** +  **'<td>'** + o.headOffice + **'</td>'** +  **'<td>'** + o.branch + **'</td>'** +  **'</tr>'** });  str += **"</tbody></table>"**;  **return new** Spacebars.SafeString(str);  } }); */\*\*  \* Hook  \*/* AutoForm.hooks({  *// Quotation* rabbit\_quotationInsert: {  before: {  insert: **function** (doc) {  doc.branchId = Session.get(**'currentBranch'**);  **var** prefix = doc.branchId + **'-'**;  Meteor.call(**'rabbit'**, prefix);  **return** doc;  }  },  onSuccess: **function** (formType, result) {  alertify.quotation().close();  alertify.success(**'Success'**);  },  onError: **function** (formType, error) {  alertify.error(error.message);  }  },  rabbit\_quotationUpdate: {  onSuccess: **function** (formType, result) {  alertify.quotation().close();  alertify.success(**'Success'**);  },  onError: **function** (formType, error) {  alertify.error(error.message);  }  } }); *// Config date picker* **var** configOnRender = **function** () {  *// date* **var** quotationDate = $(**'[name="quotationDate"]'**);  DateTimePicker.date(quotationDate); }; |

កម្រងសំណួរសម្រាប់ធ្វើការសម្ភាសន៍

1. តើសព្វថ្ងៃក្រុមហ៊ុនរបស់លោកមានការលក់ប្រព័ន្ធអ្វីខ្លះ?
2. តើមានសាខាចំនួនប៉ុន្មាន និងមានទីតាំងនៅទីណាខ្លះ?
3. តើក្រុមហ៊ុនលោកបានបង្កើតឡើងក្នុងឆ្នាំណា? ដោយអ្នកណា?
4. តើសព្វថ្ងៃក្រុមហ៊ុនរបស់លោកមានប្រព័ន្ធទិន្នន័យដើម្បីគ្រប់គ្រងការលក់ហើយឬនៅ?
5. បើមានតើប្រព័ន្ធនោះជាអ្វី? ហើយមានលក្ខណះ និងដំណើរការយ៉ាងដូចម្តេច?
6. ប្រសិនបើគ្មាន តើលោកមានចំនាប់អារម្ភណ៍លើប្រព័ន្ធទិន្នន័យដែរឬទេ?
7. តើលោកចង់បានអត្ថប្រយោជន៍អ្វីខ្លះពីប្រព័ន្ធគ្រប់គ្រងថ្មីនេះ?
8. តើអ្នកណាខ្លះនឹងប្រើប្រាស់ប្រព័ន្ធនេះ? ហើយអ្នកទាំងនោះគួរកំណត់សិទ្ធបែបណា និងឲ្យប្រើប្រាស់បានអ្វីខ្លះ?
9. តើក្រុមហ៊ុនលោកមានបុគ្គលិកចំនួនប៉ុន្មានអ្នក?
10. តើក្រុមហ៊ុនលោកមានលេខទូស័ព្ទទំនាក់ទំនងអ្វីខ្លះ?
11. តើអថិតិជនរបស់លោកជាអតិថិជនប្រភេទអ្វីខ្លះ?
12. ជាចុងក្រោយតើលោកមានទស្សនៈ និងអារម្មណ៍បែបណាចំពោះប្រព័ន្ធដែលក្រុមយើងខ្ញុំបង្កើតជូនក្រុមហ៊ុនរបស់លោក?