1. សេចក្តីផ្តើម

## ​១.១​.​ លំនាំដើម

បច្ចុប្បន្ននេះនៅទូទាំងពិភពលោកកំពុងតែមានការរីកចំរើន លើគ្រប់វិស័យទាំងផ្នែក កសិកម្ម ឧស្សាហ៍កម្ម ពាណិជ្ជកម្ម ហើយជាពិសេស បច្ចេកវិទ្យាព័ត៌មានក៏មានសន្ទុះគួរឲ្យចាប់អារម្មណ៍ខ្លាំងផងដែរដែលវាអាចជួយដោះស្រាយបញ្ហាបានយ៉ាងឆាប់ រហ័ស និង បានច្រើនជាងមុនថែមទៀត។ ហើយនៅក្នុងនេះដែរយើងបានដឹងហើយថាប្រទេសកម្ពុជា ជាប្រទេសមួយដែលបានឆ្លងកាត់នូវសង្រ្គាមអស់រយៈពេលជាច្រើនឆ្នាំមក ហើយទើបនឹងបានបញ្ចប់នៅក្នុងរយៈពេលជាង​៣០ឆ្នាំចុងក្រោយនេះ ប៉ុន្តែយើង សង្កេតឃើញថាប្រទេសកម្ពុជាក៏ជាប្រទេសមួយស្ថិតក្នុងការអភិវឌ្ឍលើគ្រប់វិស័យ មិនថាវិស័យបច្ចេកវិទ្យាព័ត៌មានដែលមនុស្សនៅលើពិភពលោកយើងនេះតែងតែខិតខំស្វែងរក អ្វីដែល ចំលែកៗអ្វីដែលថ្មីៗអ្វីដែលល្អៗ ទាំងផ្នែកបរិភោគ ឧបករណ៍សំភារៈគ្រឿងអេឡិចត្រូនិច និង​ ផ្នែកបច្ចេកវិទ្យាវិទ្យាសាស្រ្ត ។ល។ គួរឲ្យចាប់អារម្មណ៍យ៉ាងខ្លាំងផងដែលវាអាចជួយ ដោះស្រាយបញ្ហាបានយ៉ាងច្រើន និង​បានលឿនជាងមុនថែមទៀត។​ នៅក្នុងនោះមានក្រុមហ៊ុនជាច្រើនដែលបានប្រើប្រាស់នូវបច្ចេកវិទ្យាព័ត៌មាន ដើម្បីជំរុញឲ្យពាណិជ្ជកម្ម របស់ពួកគេទទួលបានជោគជ័យ​ និង​ ប្រកបដោយប្រសិទ្ធិភាព។

ដោយបានឃើញបញ្ហាជាច្រើនដែលបានកើតមានលើការរៀបរាប់ខាងលើ​ ទើបក្រុមនិស្សិតយើងខ្ញុំដែលសិក្សានៅឆ្នាំទី ៤ ជំនាន់ទី៧ ក្រុម ss នៃសាកលវិទ្យាល័យបៀលប្រាយសាខា​ ខេត្តបាត់ដំបង បានមានឱកាសចុះកម្មសិក្សាស្រាវជ្រាវ លើប្រធានបទ ប្រពន្ធ័គ្រប់គ្រង​ការលក់របស់ អាហារដ្ឋានឡាយឡាយខេត្តបាត់ដំបង មកសិក្សាស្រាវជ្រាវដើម្បីបង្កើតឲ្យមានការគ្រប់គ្រង កាន់តែមានភាពច្បាស់លាស់ និង​ ធ្វើឲ្យការងាររីកចំរើនប្រកបដោយប្រសិទ្ធិភាព សុវត្ថិភាព និង​រហ័សថែមទៀត ដើម្បីចងក្រងជាសៀវភៅទុកជា ឯកសារជំនួយដល់ការគ្រប់គ្រងនៅក្រុមហ៊ុន ឬ អង្គភាពផ្សេងៗទៀតជាពិសេសវាជាសារណាមួយ សំរាប់ក្រុមនិស្សិតយើងខ្ញុំក្នុងការបង្ហាញពីអ្វី ដែលបានខិតខំស្រាវជ្រាវអស់រយៈពេល ៤ឆ្នាំមកនេះដែល បានខិតខំចំណាយទាំងពេលវេលា កំលាំងកាយ ថវិកា ដើម្បីទទួលយកនូវចំណេះដឹងថ្មីសំរាប់ចូលរួមកសាងប្រទេសជាតិ កសាងគ្រួសារ កសាងខ្លួនឯងផងដែរនាពេលអនាគតបន្ទាប់ពីបញ្ចប់ការសិក្សាថ្នាក់បរិញ្ញាបត្រ។

## ១.២. ចំណោទបញ្ហា

​​​​​​​​​​​​​​​​​ ក្រុមយើងខ្ញុំបានសំរេចចិត្តចុះកម្មសិក្សាស្រាវជ្រាវនៅ អាហារដ្ឋាន ឡាយឡាយ ដោយផ្ទាល់ដើម្បីស្វែងយល់ពីប្រព័ន្ធ។

ក្នុងការចុះកម្មសិក្សានេះក្រុមយើងខ្ញុំបានសង្កេតឃើញបញ្ហាមួយចំនួនដូចជា៖

* ពិបាកក្នុងការគ្រប់គ្រងទិន្នន័យ
* ពិបាកក្នុងការទូរទាត់តម្លៃអាហារជាមួយអតិថិជន
* ងាយនិងភាន់ច្រលំក្នុងការទូរទាត់ប្រាក់
* ពុំមានរបាយការណ៍ប្រចាំថ្ងៃ ប្រចាំខែ​​ និង ប្រចាំឆ្នាំ
* ចំណាយក្រដាស់ច្រើនក្នុងការកត់ត្រា
* មានការច្រឡំនៃតំលៃម្ហូបដែលបានលក់ឲ្យអតិថិជនបញ្ហាទាំងនេះហើយ ដែលធ្វើឲ្យអាហារដ្ឋានមានការលំបាកក្នុងការលក់និងគ្រប់គ្រង។

ជួនជាពេលដែលក្រុមនិស្សិតយើងខ្ញុំ ត្រូវសរសេរនិក្ខេបបទដើម្បី បញ្ចប់ឆ្នាំសិក្សាថ្នាក់បរិញ្ញាបត្រផ្នែកព័ត៌មានវិទ្យា និង​ដើម្បីដាក់បង្ហាញពីស្នាដៃដែលក្រុមនិស្សិតយើងខ្ញុំបានសិក្សាអស់រយៈពេល ៤ឆ្នាំកន្លងមកនេះ គឺយើងបានសិក្សាច្រើនទៅលើការប្រើប្រាស់នូវ​ឮ បច្ចេកវិទ្យាព័ត៌មាន និង​កសាងប្រព័ន្ធ​សំរាប់ធ្វើការគ្រប់គ្រង។ ដើម្បីបញ្ចប់ថាការសិក្សារបស់យើងពិតជាអាចធ្វើទៅបាន។

តើអនុវត្តន៍នូវវិធីសាស្រ្តនៃការគ្រប់គ្រងដោយប្រើប្រាស់កុំព្យូទ័រមានលទ្ធភាពដោះស្រាយបញ្ហាខាងលើនេះបានយ៉ាងណាដែរ? ចុះវាអាចជួយអោយអ្នកគ្រប់គ្រងទទួលបាន អ្វីខ្លះពីផលប្រយោជន៍នៃការគ្រប់គ្រងដោយប្រើប្រាស់បច្ចេកវិទ្យាព័ត៌មានវិទ្យា?ហើយមាន គោលបំណងអ្វី?

## ១.៣. គោលបំណងនៃការស្រាវជ្រាវ

ក្នុងការចុះកម្មសិក្សាស្រាវជ្រាវនៅ អាហារដ្ឋាន ឡាយឡាយ នេះក្រុមនិស្សិតយើង ខ្ញុំធ្វើឡើងក្នុងគោលបំណងជួយដោះស្រាយរាលបញ្ហាដែលកើតមានឡើងនៅពេលដែល មានអតិថិជនមកចូលពិសារភោជនីអាហារ និងភេសជ្ជៈទៅតាមចំនង់និងចំណូលចិត្តរបស់ពួកគេដែលចង់បាន។ ដើម្បីឆ្លើយតបទៅតាមតំរូវការនេះអោយទាន់ពេលវេលារបស់អតិថិជន ក្រុមនិស្សិតយើងខ្ញុំក៏សំរេចចិត្តបង្កើត ប្រព័ន្ធថ្មីមួយដែលមានឈ្មោះថា**”ប្រព័ន្ធគ្រប់គ្រងការលក់របស់អាហារដ្ឋាន ឡាយឡាយ ខេត្តបាត់ដំបង”** ។ ប្រព័ន្ធថ្មីនេះអាចជួយទៅដល់ការគ្រប់គ្រងឲ្យមានសភាពងាយស្រួលនិង មានប្រសិទ្ធិភាពខ្ពស់ជាងការធ្វើដោយដៃ។ ដើម្បីដំណើរការប្រព័ន្ធនេះទៅលុះត្រាតែយើងត្រូវបានបច្ចេកវិទ្យាថ្មីមកកសាង។ បច្ចេកវិទ្យាថ្មីដែលយើងយកមកកសាងនេះគឺ យើងជ្រើសរើស យកកម្មវិធីមួយចំនួនដែលមានដូចជា កម្មវិធី វីស៊ួលស្ទិតយូរ(Visual Studio 2013),អេសឃ្យូរអិល​ស័រវ័រអ័រធូរ(SQL Server R2) និង គ្រីសស្ទាលរីផត(C) សំរាប់បញ្ចូល(Insert)​ បង្ហាញ(Report & Interface) និងផ្ទុកទិន្នន័យ(Database)។

ដូច្នេះគោលបំណងនៃការបង្កើតប្រព័ន្ធថ្មី គឺដើម្បីផ្តល់ប្រយោជន៍មួយចំនួនដូចខាងក្រោម៖

* បង្កើននូវភាពងាយស្រួលក្នុងការគ្រប់គ្រងនៅក្នុងអាហារដ្ឋាន
* គ្រប់គ្រងទិន្នន័យនៃការលក់់
* បំពេញតាមតំរូវការរបស់អតិថិជន ពេលទូរទាត់ប្រាក់
* កាត់បន្ថយការប្រើប្រាស់ក្រដាស់
* ចំណេញពេលវេលាក្នុងការស្វែងរកវិក័យបត្រ័ចាស់
* បង្កើនល្បឿននៃការងារ
* កាត់បន្ថយភាពស្មុគស្មាញក្នុងការគ្រប់គ្រង
* បង្កើននូវសុវត្ថិភាពនៃការគ្រប់គ្រង

ក្រុមនិស្សិតយើងសង្ឃឹមថា ប្រព័ន្ធថ្មីដែលកសាងរួច​ និងមានលក្ខណៈស្អាតហើយគួរអោយចាប់អារម្មណ៍ក្នុងការប្រើប្រាស់ថែមទាំងងាយស្រួលប្រើទៀតផង។

## ១.៤.ទំហំ និង ដែនកំណត់នៃការស្រាវជ្រាវ

ជាក់ស្តែងមនុស្សម្នាក់ៗមុននិងធ្វើអ្វីមួយតែងតែមានការគិតទុកជាមុនដើម្បីជៀសវាងនូវកំហុសដែលកើតមានឡើងពីព្រោះពីមួយជំនាន់ទៅមួយជំនាន់សុទ្ធសឹងតែជួបប្រទះនូវបញ្ហា​មួយចំនួនដែលធ្វើអោយពួកគេមាន បទពិសោធន៍ផ្សេងៗពីគ្នាក្នុងការជៀសវាងពី ហានិភ័យ។ ដូចនេះសំរាប់ក្រុមសិក្សាស្រាវជ្រាវ យើងខ្ញុំដែលបានលើកយកប្រធានបទ ប្រព័ន្ធគ្រប់គ្រង​ ការលក់របស់អាហារដ្ឋាន ឡាយឡាយ សំរាប់បញ្ចប់ថ្នាក់បរិញ្ញបត្រដែលមានរយៈពេលត្រឹមតែ ៤ខែបានកំណត់នូវទំហំ និងដែនកំណត់របស់ប្រព័ន្ធដែលមានជា អាទិ៍ដូចខាងក្រោម៖

* ការគ្រប់គ្រងតំលៃអាហារនិងភេសជ្ជៈក្នុងអាហារដ្ឋាន
* ការគ្រប់គ្រងព័ត៌មានការលក់ចេញ
* មានការគណនាច្បាស់លាស់រាល់មុខម្ហូបដែលបានលក់ចេញ
* ការបង្កើតជាវិក័យបត្រជូនអតិថិជន(Report)
* គ្រប់គ្រងរបាយ៍ការណ៍ប្រចាំថ្ងៃ ប្រចាំខែ ប្រចាំឆ្នាំ

1. រំលឹកទ្រឹស្តី

## ២.១ អ្វីទៅជាបច្ចេកវិទ្យាព័ត៌មានវិទ្យា?

បច្ចកវិទ្យាព័ត៌មានវិទ្យា(Information System) ជាសំណុំនៃធាតុ​រឺ សមាសធាតុពាក់ព័ន្ធគ្នាជាច្រើនដែលប្រមូលផ្តុំ(Input), រៀបចំចងក្រង(Process), និងចែកចាយ(Output) ទិន្នន័យនិង ព័ត៌មានហើយផ្តល់នូវផលត្រឡប់មកវិញដើម្បីសំរេចគោលបំណង។

## ២.២. សញ្ញានទូទៅនៃប្រព័ន្ធព័ត៌មាន?

### ២.២.១. អ្វីជាព័ត៌មាន (Information)?

ព័ត៌មាន (Information)គឺជា ការប្រមូលផ្តុំ នៃវត្ថុធាតុជាច្រើនតាមមធ្យោ បាយណាមួយដែលធ្វើអោយពួកវាមានតំលៃបន្ថែមលើសពីតំលៃដើមរបស់ពួកវា។

### ២.២.២.​ អ្វីទៅជាទិន្នន័យ (Data)?

ទិន្នន័យ (Data) គឺជាអ្វីៗដែលយើងត្រូវការដកស្រង់ ប្រមូល ចងក្រង ដែលពុំទាន់មានន័យសម្រាប់ការស្រាវជ្រាវទេ មានន័យថាពុំទាន់វិភាគ។។ទិន្នន័យត្រូវបញ្ចូលទៅក្នុងប្រព័ន្ធ រក្សាទុកក្នុងប្រព័ន្ធ​ និង ដំណើរការដោយប្រព័ន្ធ។

ទិន្នន័យចែកចេញជាពីរគឺទិន្នន័យចំបង និងទិន្នន័យបន្ទាប់បន្សំ។

* ទិន្នន័យចំបងគឺជាទិន្នន័យដែលអ្នកស្រាវជ្រាវ (researchers) ធ្វើការប្រមូលដោយផ្ទាល់ពីតាមក្រុមហ៊ុនឬពីកន្លែងដែលចុះស្រាវជ្រាវ។
* ទិន្នន័យបន្ទាប់បន្សំគឺជា ទិន្នន័យដែលអ្នកស្រាវជ្រាវ (researchers) ធ្វើការប្រមូលពីឯកសារដែលបានសិក្សា សៀវភៅ កិច្ចការដែលបានបោះពុម្ព កំណត់ត្រា អត្ថបទនានា ដែលការប្រមូលទនិន្នន័យនេះ ធ្វើឡើងមុនការចុះស្រាវជ្រាវ និងមាន ទំនាក់ទំនងនឹងប្រធានបទ។

### ២.២.៣.​ អ្វីទៅជាប្រព័ន្ធ (System)?

ប្រព័ន្ធ (System) ​គឺជាការប្រមូលផ្តុំនូវសមាសធាតុ ដែលមានទំនាក់ទំនងគ្នាជាសកម្មភាព ដើម្បីបំពេញគោលបំណងច្បាស់លាស់អ្វីមួយ។ ប្រព័ន្ធត្រូវមានសមាសធាតុដែលមានទំនាក់ទំនង (Related Component) និងគោលបំណង(Goal)។

### ២.២.៤.​ អ្វីទៅជាប្រព័ន្ធព័ត៌មាន (Information System)?

​ប្រព័ន្ធព័ត៌មាន(Information System)មិនមែនជាវត្ថុតែមួយទេ ប៉ុន្តែវាជាបណ្តុំនៃវត្ថុដែលធ្វើការជាមួយគ្នាយ៉ាងជិតស្និត ហើយវត្ថុទាំងនោះហៅថា Components នៃប្រព័ន្ធដែលរាប់បញ្ចូល Equipment ដូចជា Computer, Instruction for equipment, Facts រក្សាទុកនៅក្នុងប្រព័ន្ធ People ដើម្បីប្រតិបត្តិប្រព័ន្ធនិង Procedures Edl People ត្រូវគោរព

តាម។សមាសភាពភាគច្រើននោះ ធ្វើការជាមួយគ្នាដើម្បី ផ្តល់ព័ត៌មានប្រើប្រាស់សំរាប់ប្រតិបត្តិ ការប្រចាំថ្ងៃ (Daily Operation) និងគ្រប់គ្រង (Management) ចំពោះ Organization។​ និយាយអោយខ្លីហើយច្បាស់ ប្រព័ន្ធព័ត៌មានគឺជាបណ្តុំសមាសភាពដែលធ្វើការជាមួយគ្នាដើម្បី ផ្តល់ព័ត៌មានជួយប្រតិបត្តិ ​និង​ គ្រប់គ្រង Organization។

### **២.២.៤.១.​ការគ្រប់គ្រងប្រព័ន្ធពត៌មាន**(**Management Information System**)

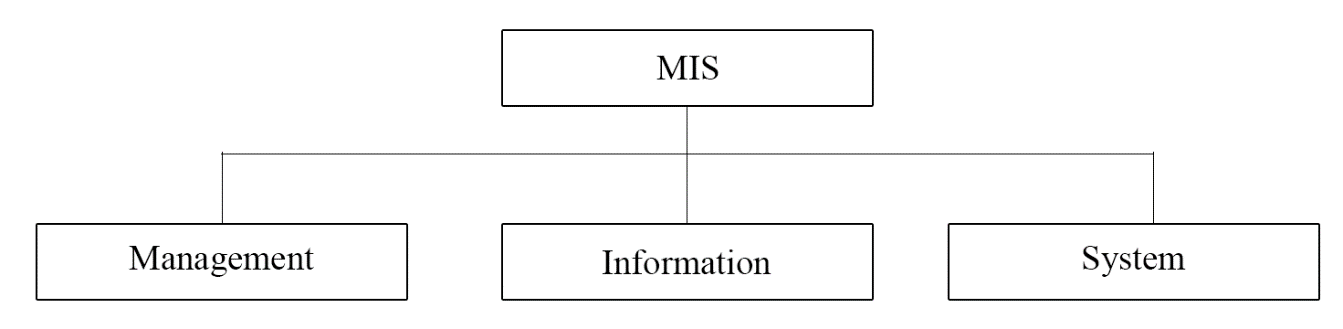
តើប្រព័ន្ធគ្រប់គ្រងព័ត៌មាន(MIS)ជាអ្វី?

MIS គឺជាប្រព័ន្ធព័ត៌មានដែលប្រើសំរាប់ផ្តល់ព័ត៌មានដល់អ្នកគ្រប់គ្រង ដើម្បី ធ្វើការសំរេចចិត្ត។​ក៏ព្រោះតែកុំព្យូទ័រមាន Tools ជាច្រើនសំរាប់ដំណើរការ Data ក៏ដោយ ក៏ទស្សនៈអ្នកគ្រប់គ្រងជាច្រើនបានយល់ឃើញថា វាជាសមាសភាពដ៏សំខាន់មួយ នៅក្នុង ប្រព័ន្ធព័ត៌មាន។ មុខងារពិតប្រាកដរបស់វាគឺដើម្បី ផ្តល់ព័ត៌មានសំរាប់សំរេចចិត្ត ការរៀបចំ ផែនការហើយ និងត្រួតពិនិត្យមើលប្រតិបត្តិការណ៍។ (Kenneth C. Laudon, Jane P. Laudon 2002 )

MIS មិនថ្មីទេគ្រាន់តែ ការគណនារបស់វាថ្មីប៉ុណ្ណោះ។ ពីមុន បច្ចេកទេស MIS របស់កុំព្យូទ័រមាន រួចមកហើយក្នុងការផ្គត់ផ្គង់ អ្នកគ្រប់គ្រង់ដោយវាគ្រាន់តែបន្ថែមភាពសំខាន់មួយរឺ ច្រើនដូចជាលឿនការងារ និងពង្រីកទំហំនៃទិន្នន័យ ដែលអនុញ្ញាតឲ្យពិនិត្យក្នុងការពិចារណាមួយបានច្រើនដង។​

បច្ចុប្បន្ននេះនៅ​ លើពិភពលោកស្ទើរតែគ្មានជំនួញណាដែលមិនបានយកបច្ចេកវិទ្យាប្រព័ន្ធព័ត៌មានប្រើប្រាស់ដើម្បីធ្វើអោយឱកាសជំនួញរបស់ពួកគេមានការឈាន ទៅមុខគេ ជានិច្ច។ តាមទស្សនៈរបស់ អ្នកជំនួញបានពោលថាពិភពលោកថ្មីនៃប្រព័ន្ធព័ត៌មានគឺ ពឹង ផ្អែកទៅលើយើងទាំងអស់គ្នា ដែលជាអ្នកប្រើប្រាស់រឺ គ្រប់គ្រង។

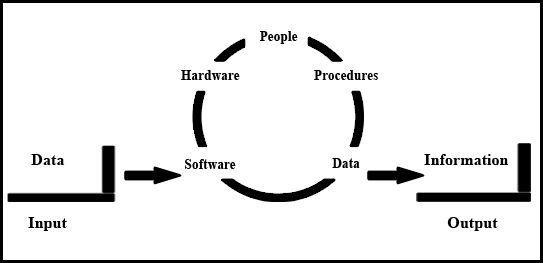
តួលេខ ២.១ ប្រព័ន្ធគ្រប់គ្រងព័ត៌មាន



### **២.២.៤.២.​ សមាសធាតុនៃប្រព័ន្ធព័ត៌មាន**

សមាសធាតុនៃប្រព័ន្ធព័ត៌មានមានចែកចេញជា ៥ សំខាន់គឺ Hardware, Software, Data, People និង Processes។ Hardware និងSoftware គឺជា Information Technology នៃប្រព័ន្ធហើយProcesses និងPeopleគឺជាHuman Resources នៃប្រព័ន្ធ។ Data ចងភ្ជាប់ទំនាក់ទំនងរវាង Information Technology និង Human Resources។

តួលេខ ២.២ សមាសធាតុនៃប្រព័ន្ធព័ត៌មាន



Hardware: សមាសភាពដំបូងនៃប្រព័ន្ធព័ត៌មានគឺ Hardware ដែលផ្ទុកនូវ Computers, Communication equipment និង Device ផ្សេងទៀតប្រើប្រាស់ក្នុងប្រព័ន្ធព័ត៌មាន។ ប្រព័ន្ធព័ត៌មានភាគច្រើន ប្រើប្រាស់ Computers ជាច្រើនប្រភេទរួមបញ្ជូលគ្នាហើយ Computers ទាំងអស់នោះភ្ជាប់គ្នាដោយប្រើប្រាស់ Communication equipment ដូចជា Wires ជាដើម។ Devices ផ្សេងទៀតមានដូចជា Digital Camera, Scanners…។

Software: គឺជាសមាសភាពផ្សេងទៀតនៃប្រព័ន្ធព័ត៌មាន ដែលផ្ទុកនូវ Instruction ប្រាប់ទៅHardware នូវអ្វីដែលត្រូវធ្វើ។ Computers និង Communication equipment មិនអាចដំណើរដោយគ្មាន Software នោះទេ មានន័យថារាល់​​ Hardware ត្រូវតែមាន Instruction ដើម្បីប្រាប់នូវអ្វីដែលត្រូវធ្វើ។​ ប្រព័ន្ធព័ត៌មានត្រូវការ Software ជាច្រើនប្រភេទដូចជាSystem Computer និង Application Software ។ System Software ផ្តល់សមត្ថភាពដែលអាចអោយ Computer ដំណើរការបានដូចជា Operating System, Device Driver និង Utilitiesដែលដំណើរការងារដូចជាបំលែងទិន្នន័យទៅជាទំរងផ្សេងៗ Virus Protection, Sort, Merge, Print, Copy និង Backup Utilities​។ Application Software ផ្ទុកកម្មវិធី ទ្រទ្រង់ អ្នកប្រើប្រាស់ ឬដំណើរការមុខងារនៅក្នុងប្រព័ន្ធ និង ធ្វើអោយក្រុមហ៊ុនមានលទ្ធភាពដំណើរការ Business Function ។ អ្នកប្រើប្រាស់បង្កើនផលិតភាពជាមួយ Tools ដូចជា Spreadsheets, Word Processors និង Database Management System (DBMS)​។

Data: សមាសភាពទី៣នៃប្រព័ន្ធមានគឺ ទិន្នន័យដែលរក្សាទុកលើ Hardware និងដំណើរការ (Process) ដោយ Software​​ ។ Data Component នៃប្រព័ន្ធព័ត៌មានផ្ទុកទិន្នន័យ ទាំងអស់រក្សានៅក្នុង Computers ចំពោះប្រព័ន្ធ និងប្រើប្រាស់ដោយSoftware នៃប្រព័ន្ធ។

People: គោលបំណងជាចំបងរបស់ប្រព័ន្ធព័ត៌មាន គឺផ្តល់នូវព័ត៌មានមានតំលៃទៅ ដល់ Manager និង ប្រើប្រាស់ទាំងអស់់នៅក្រុមហ៊ុនហើយល្បឿនដំណើរការព័ត៌មានត្រូវតែល្បឿន និងគួរអោយទុកចិត្តបាន។ ប្រព័ន្ធព័ត៌មានមិនមានដំណើរការដោយខ្លួនវាផ្ទាល់ទេ ដូចនេះវាត្រូវការ People ដើម្បីធ្វើអោយវាដំណើរការទូទៅ People ត្រូវបញ្ចូល (Input Data) ទៅកាន់ប្រព័ន្ធទទួល Output Data ធ្វើអោយ Hardware ដំណើរការជាមួយគ្នា និងចាប់ផ្តើម ដំណើរការ Software អ្នកប្រើប្រាស់ (User រឺ អាចហៅថា End User) រួមមាន Employees, Customers, Vendors​ និងអ្នកដ៏ទៃទៀតដែលពាក់ព័ន្ធជាមួយប្រព័ន្ធព័ត៌មាន ដែលចែកចេញជា ២ប្រភេទគឺ អ្នកប្រើប្រាស់ខាងក្នុង (Internal User) មាន Managers, Technicians, Sale Reps និងអ្នកធ្វើការនៅក្នុងការិយាល័យ និងអ្នកប្រើប្រាស់ខាងក្រៅ (External User) មាន Customers ដែលតាមដានការកុម្ម៉ងទិញនៅលើ Website, Suppliers ដែលប្រើប្រាស់ប្រព័ន្ធរបស់អតិថិជនដើម្បីគ្រោងលើការផលិតអោយបានត្រឹមត្រូវ និង Employers ដែល Log onទៅកាន់ Internet​របស់ក្រុមហ៊ុនពីផ្ទះដើម្បីពិនិត្យមើល E-mail message​។

Processes: សមាសភាពចុងក្រោយនៃប្រព័ន្ធព័ត៌មានគឺ Process រឺអាចហៅម៉្យាងទៀតថា Procedure ដែលគឺ Instruction ប្រាប់ People ត្រូវប្រើប្រាស់ដោយរបៀបណា​។ ដូចទៅនឹង Hardware មិនអាចដំណើរការបើគ្មាន Software, People មិនដឹងអ្វីត្រូវធ្វើប្រសិនបើគ្មាន Processes សំរាប់ធ្វើតាមទេ​។

## ២.៣. អ្វីទៅជា Database System?

Database System គឺជាប្រព័ន្ធមួយប្រើប្រាស់សំរាប់រៀបចំទិន្នន័យដែលមានទំនាក់ទំនងគ្នាជាច្រើន និងមានភាពស៊ាំញ៉ាំដើម្បី សំរេចនូវគោលបំណង ជាក់លាក់មួយ។ Database System ត្រូវបានគេប្រើប្រាស់សំរាប់ Store Data, Manipulate (ការគណនា) និង Retrieve Data គ្រប់ប្រភេទរួមមានៈ Business, Health, Education, Library,….។ ឧទាហរណ៍ Customer Management System, Online Shopping, Book Management,….។

## ២.៤. Database Management System (DBMS)

DBMS ភាគច្រើនផ្តល់នូវ Query language ឬ Report ដែលអាចអោយ User ធ្វើការសាកសួរ និងទទួលបានព័ត៌មានទាំងនោះស្ទើរតែភ្លាមៗលើ Terminal របស់ពួកគេ ដោយមិនចាំបាច់ទាមទារអោយ Programmers សរសេរ Software ទាញព័ត៌មានទាំងនោះពី Database ទេ។ ឧទាហរណ៍ៈ Branch Manager អាចបង្ហាញរាល់ Flats ទាំងអស់ដែលមាន Monthly Rent លើសពី $500ដោយឃ្លាបញ្ជ SQLលើ Terminal ដូចខាងក្រោមៈ

SELECT\*FROM Property\_for\_Rent

WHERE type=Flat AND rent>500

DBMS គឺជា System Software ពិសេសដែលប្រើដើម្បីបង្កើតគ្រប់គ្រង និងថែទាំ Database។ វាមានសម្មតិភាពបែងចែក Business Application នីមួយៗដើម្បីទាញយក Data (to Extract Data) ដែលប្រើដោយមិនចាំបាច់បង្កើតfile ដាច់ដោយលែកផ្តល់ និយមន័យក្នុង Program របស់ខ្លួនក្នុងកុំព្យូទ័រមានន័យថា DBMS ជា Software ដែលសំរាប់គ្រប់គ្រង Database។ DBMS ដើរតួរជា Interface Application Program និង Physical Data File។ នៅពេល Application Program ត្រូវការ Data, DBMS ស្វែងរកក្នុង Database និងតំណាងឱ្យ Data ទៅកាន់ Application Program នោះ។

Database គឺជា Object ស្មុគស្មាញសំរាប់ផ្ទុកព័ត៌មានដែលមានរចនាសម្ព័ន្ធ ហើយព័ត៌មានទាំង នោះត្រូវបានគេរៀបចំ និង​ រក្សាទុកក្នុងមធ្យោបាយមួយដែលអាចអោយការទាញយក ទិន្នន័យមាន ភាពលឿន និង​ មានប្រសិទ្ធិភាព។

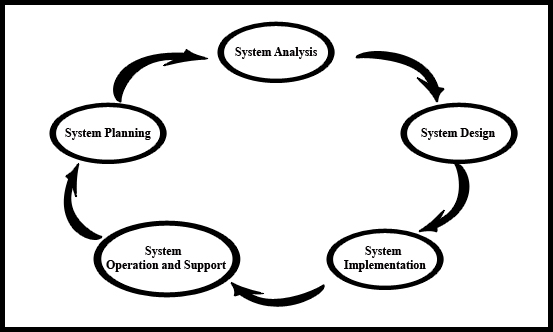
SQL Server 2008R2 (Structured Query Language ) គឺជា Server DBMS និង​ Relationship DBMS ដែលជាផលិតផលរបស់ក្រុមហ៊ុន Microsoft ។ Microsoft SQL Server មានសមាសធាតុផ្សំមួយចំនួនដូចជា Relationship Database Engine, Relationship, Data Transformation Service (DTS), Analysis Service, Meta Data service និង English Query។

## ២.៥ វដ្តជិវិតអភិវឌ្ឍន៍ប្រព័ន្ធ(The Systems Development Life Cycle)

Structured Analysis ប្រើបា្រស់វិធីដែលគេហៅថា Systems Development Life Cycle (SDLC) ដើម្បីរៀបចំ និងគ្រប់គ្រងដំណើរអភិវឌ្ឍន៍ប្រព័ន្ធហើយ SDLC រៀបចំពីសកម្មភាព (Activities) និងមុខងារ​(Functions)ដែលអ្នកបង្កើតប្រព័ន្ធជាទូទៅដំនើរការ។

SDLC ចែកចេញជា​​ ៥​ ដំណាក់កាលដែលដំណើរការជាបន្តបន្ទាប់ ហើយមាន​​លក្ខណៈសារចុះឡើង នៅពេលមានការកែតំរូវដូចខាងក្រោមៈ​

តួលេខ ២.៣ វដ្ដេជីវិតអភិវឌ្ឍន៍ប្រព័ន្ធ (The System Development Life Cycle)



### ២.៥.១. Systems Planning

System Planning ជាទូទៅចាប់ផ្តើមដោយធ្វើសំណើរសុំជាផ្លូវការទៅកាន់ IT Department ហៅថា Systems Request ដែលរៀបរាប់អំពីបញ្ហារឺ ក៏ចង់កែប្រែទៅលើប្រព័ន្ធព័ត៌មាន រឺ Business Process ។ System Request អាចចេញពី Top Manager, Department Head រឺ IT Department ខ្លួនឯងផ្ទាល់ ហើយសំណើរសុំនោះអាចសំខាន់ណាស់ព្រោះវាពាក់ព័ន្ធដល់ការបង្កើតប្រព័ន្ធព័ត៌មានថ្មី រឺជំនួសប្រព័ន្ធចាស់កំពុងប្រើប្រាស់ដែលមិនអាចដោះស្រាយតំរូវការបច្ចុប្បន្ន​បាន ផ្ទុយទៅវិញសំណើរសុំមានលក្ខណៈតូចតាច ដែលពាក់ព័ន្ធជាមួយការសុំ Feature ថ្មីរឺ កែប្រែ User Interface ។

### ២.៥.២. Systems Analysis

គោលបំណងនៃ System Planning Phase គឺយល់អោយច្បាស់នូវ Business Requirements និងបង្កើត Logical Model នៃប្រព័ន្ធថ្មី។ ជំហ៊ានដំបូងគឺ Requirements Modeling ដែលយើងកំនត់ និងរៀបរាប់ពី Business Processes ហើយ Requirement Modeling បន្តដំណើរការស៊ីបអង្កេតដែលបានចាប់ផ្តើមតាំងពី System Planning ជាបន្តទៀត និងពាក់ព័ន្ធ Fact-Finding Techniques មួយចំនួនដូចជា កិច្ចសំភាសន៍ (Interview), ការស្ទង់(Surveys), ការសង្កេត(Observation) និងការជ្រើសរើសសំណាក (Sampling) ជាដើម​​ ។

បន្ទាប់មក កិច្ចការជាបន្តទៀតគឺ Data Modeling, Process Modeling និង Object Modeling​ដែលយើងប្រើប្រាស់សំរាប់បង្កើត Logical Model of Business Processes ដែលប្រព័ន្ធត្រូវទ្រទ្រង់ហើយ End Product នៃ System Analysis Phase គឺជា System requirements Document ដែលរៀបរាប់ពី Management and User Requirements, ផែនការនិង តំលៃផ្សេងទៀត (Alterative Plans and Costs) និងអនុសាសន៍របស់អ្នកវិភាគប្រព័ន្ធ។

### ២.៥.៣. Systems Design

គោលបំណងនៃ Systems Design Phase គឺបង្កើត Blueprint សំរាប់ប្រព័ន្ធថ្មីដែលបំពេញរាល់ Documented Requirements។ កំឡុងពេល Systems Design យើងធ្វើការកំនត់រាល់ Outputs, Input, Interface and Processes ចាំបាច់ទាំងអស់លើសពីនេះទៀតយើងកសាង Internal and External Controls ដោយរាប់បញ្ចូលទាំង Computer-based or Manual Feature ដើម្បីធានាថា ប្រព័ន្ធគួរអោយជឿជាក់, ត្រឹមត្រូវ, អាចថែទាំបាន និងសុវត្ថិភាព។

End Product នៃ System Design គឺជា Systems Design Specification ហើយយើងត្រូវបង្ហាញ Systems Design Specification ជូនទៅ Managers និង Users ដើម្បីត្រួតពិនិត្យ និងយល់ព្រម។ ការរួមបញ្ចូលរបស់ Users and Managers មានសារៈសំខាន់ណាស់ដើម្បីជៀសវាងការភាន់ច្រឡំណាមួយ ចំពោះអ្វីដែលប្រព័ន្ធថ្មីនិងធ្វើ, តើវាធ្វើដោយរបៀបណា។

### ២.៥.៤. Systems Implementation

នៅក្នុងដំណាក់កាល Systems Implementation យើងធ្វើការបង្កើតប្រព័ន្ធថ្មីអាចលេចចេញជារូបរាង ដោយមិនខ្វល់ថាតើអ្នកអភិវឌ្ឍន៍ប្រើប្រាស់ Structured Analysis or O-O Methods ព្រោះទំរង់របស់វាដូចគ្នាគឺចាប់ផ្តើមសរសេរកម្មវិធី, សាកល្បង, រក្សាឯកសារ និង តំឡើងប្រព័ន្ធ។ ប្រសិនបើប្រព័ន្ធនោះទិញពីក្រុមហ៊ុនដ៏ទៃ ពេលនោះអ្នកវិភាគប្រព័ន្ធដំណើរការ Modification and Completely Functioning and Documented Information Systems។

សរុបមកជំហ៊ាននេះគឺប្រព័ន្ធត្រៀមជាស្រេចក្នុងការប្រើប្រាស់ដោយរួមបញ្ចូលទាំងការបំលែងទិន្នន័យទៅជា Files ក្នុងប្រព័ន្ធថ្មី, បណ្តុះបណ្តាល User and Transition ទៅកាន់ប្រព័ន្ធថ្មី។ Systems Implementation រូមបញ្ចូលផងដែរនូវការប៉ាន់ប្រមាណ (Assessment) គេហៅថា System Evaluation ដើម្បីកំនត់ថាតើប្រព័ន្ធដំណើរការដោយត្រឹមត្រូវដែរឬទេ និងថាតើចំណាយ និងកំរ៉ៃស្ថិតក្នុងការរំពឹងទុកដែររឺទេ។

## ២.៦ Data Flow Diagram(DFD)

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

### ២.៦.១. Data Flow Symbols

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

Process Symbol: Processes ទទួល Input Data បង្កើត Output ដែល Content, Form រឺ ទាំងពីរមានលក្ខណៈខុសពីមុន។

និម្មិតសញ្ញាសំរាប់ Process គឺជារង្វង់មូល ហើយឈ្មោះ Process ស្ថិតនៅក្នុងរង្វង់មូលនោះតែម្តង។ ឈ្មោះ Process សំរាប់កំណត់ Function ជាក់លាក់ណាមួយ និងផ្ទុកនូវកិរិយាស័ព្ទ(និងគុណនាម ប្រសិនបើចាំបាច់) តាមក្រោយដោយនាម ។ ឧទាហរណ៍ៈ ឈ្មោះ Process ដូចជា Calculate Commission, Verify order and Fill Order ។

Data Flow Symbol: 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។

និម្មិតសញ្ញាសំរាប់ Data Flow គឺបន្ទាត់ដែលមានចុងព្រួញមួយ រឺពីរ ហើយឈ្មោះData Flow បង្ហាញនៅខាងលើ​ រឺខាងក្រោមបន្ទាត់។​ ឧទាហរណ៍ៈ ឈ្មោះ Data Flow ដូចជាៈ Deposit, Order, Student Grade and Commission ។

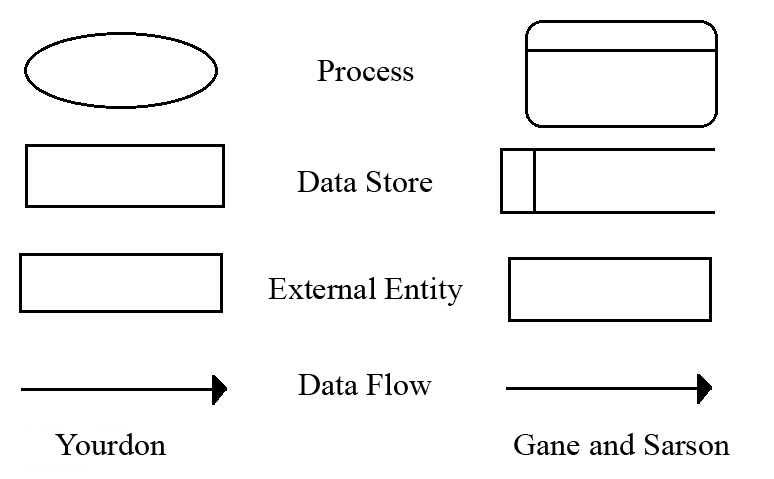
Data Store Symbol: Data Store or Data Repository ប្រើប្រាស់ក្នុង DFDសំរាប់បង្ហាញពីស្ថានការណ៍ដែលប្រព័ន្ធដែលប្រព័ន្ធត្រូវរក្សាទុកទិន្នន័យនោះ ព្រោះ Process មួយរឺច្រើន ផ្សេងទៀតត្រូវការប្រើប្រាស់វានៅពេលក្រោយ​​ ហើយចំពោះរយៈរក្សាទុកទិន្នន័យមិនសំខាន់តែអាច១នាទី, ១ម៉ោង រឺ ១ខែ ជាដើម។ ឧទាហរណ៍ៈ គ្រូត្រូវរក្សាទុកពិន្ទុ Assignmentសិស្ស ដូចនេះនៅពេលប្រលងឆមាសគេអាច គណនាពិន្ទុលទ្ធផលបាន។

និម្មិតសញ្ញាសំរាប់ Data Store គឺជាចតុកោណកែងដែលចំហរនៅផ្នែកខាងឆ្វេង និងខាងស្តាំហើយឈ្មោះ Data Storeបង្ហាញនៅចន្លោះចតុកោណកែងនោះ។ឧទាហរណ៍ៈ ឈ្មោះ Data Store​ ដូចជា students, Products, Employees and Daily payments។

External Entity Symbol: External Entity គឺជាមនុស្ស,​Department, Outside Organization និងប្រព័ន្ធព័ត៌មានផ្សេងទៀតដែលផ្តល់ទិន្នន័យទៅកាន់ប្រព័ន្ធ​រឺទទួលលទ្ធផលពីប្រព័ន្ធ។ ឧទាហរណ៍ៈ Customer បញ្ចូល Order​ គឺជា External Entity ពីព្រោះ Customer ផ្តល់ទិន្នន័យទៅកាន់ប្រព័ន្ធ។

និម្មិតសញ្ញាសំរាប់ External Entity គឺជាចតុកោណកែង ហើយឈ្មោះ External Entity បង្ហាញនៅចន្លោះចតុកោណកែងនោះ។​ ឧទាហរណ៍ៈ ឈ្មោះ External Entity ​ដូចជា Customer, Student, Employee, Warehouse and Bank។

តួលេខ ២.៤ និមិត្តសញ្ញារបស់ DFD



### ២.៦.២. Context Diagram

យើងប្រើប្រាស់ Interviews, Questionnaires and Techniques ផ្សេងៗសំរាប់ប្រមួល Facts ចំពោះប្រព័ន្ធនិងសិក្សាស្វែងយល់អំពីរបៀបទំនាក់ទំនងរវាងមនុស្សជាច្រើន, Departments, Data and Processes ក្នុងកំឡុងពេល Requirement Modeling រួចមកហើយឥឡូវយើងត្រៀមជាស្រេចដើម្បីបង្កើត Graphic Model សំរាប់ប្រព័ន្ធព័ត៌មានដោយសំអាងទៅលើលទ្ធផលនៃ 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។

### ២.៦.៣. 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 ។

### ២.៦.៤. Lower-Level Diagram

Lower-Level Diagram (គេអាចហៅបានថា Child Diagram) ប្រើប្រាស់សំរាប់បង្ហាញព័ត៌មានលំអិត 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​។

### ២.៦.៥. Strategies for Developing DFDs

យុទ្ធសាស្រ្តសំរាប់បង្កើត 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 សំរាប់ 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។

### ២.៦.៦. Data Dictionary

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

Data Dictionary or Data Repository​ គឺជាកន្លែងកណ្តាល (Central Storehouse) សំរាប់រក្សាព័ត៌មានអំពីទិន្នន័យប្រព័ន្ធ។ 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។

## ២.៧ Data Design Terminology

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

### ២.៧.១. Definitions

Entity: សំដៅទៅលើមនុស្ស, ទីកន្លែង, វត្ថុ រឺព្រឹត្តិការណ៍សំរាប់ប្រមូល 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 ជាច្រើនប្រើសំរាប់រក្សាទុកទិន្នន័យចំពោះ 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 មានទំនាក់ទំនងគ្នាប្រើសំរាប់រៀបរាប់ពី One Instance​ រឺសមាជិក Entity ដូចជា One Customer, One Order or One Product។ Record មួយអាចមាន Field​ មួយរឺច្រើនអាស្រ័យទៅតាមតំរូវការព័ត៌មានចង់បាន។

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

នៅក្នុង Database Environment សំនុំ Records ទាក់ទងគ្នាជាច្រើនផ្តុំចូលគ្នាបង្កើតជា Table សំរាប់រក្សាទុកទិន្នន័យចំពោះ​ 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។

### ២.៧.២. 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។ ឧទាហរណ៍ៈ 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 មួយសំរាប់សិស្សនីមួយៗដែលបានចុះឈ្មោះ។

Candidate Key: នៅក្នុងករណីមួយចំនួនយើងអាចជ្រើសរើសយក Field មួយរឺបណ្តុំ Fields ជាច្រើនដើម្បីកំនត់ជា Primary Key។ Fields​ ទាំងឡាយណាដែលអាចដើរតួនាទីជា Primary Key គេហៅថា Candidate Key។ ឧទាហរណ៍ៈ 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។

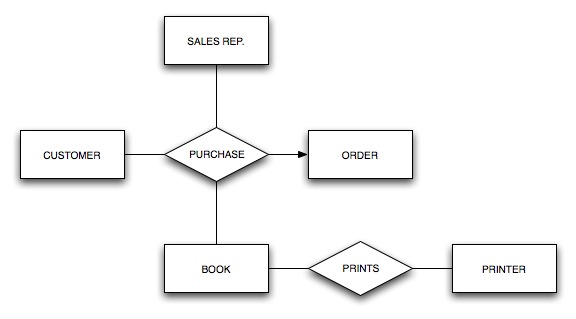
Foreign Key: គូរកត់សំគាល់ថា Common Field លេចឡើងនៅក្នុង Table ជាច្រើនជាមួយ និងត្រូវបានគេប្រើប្រាស់ដើម្បីបង្កើតទំនាក់ទំនង (Relationship or Link) រវាង Tables ទាំងនោះ។ Foreign Key គឺជា Field ស្ថិតនៅក្នុង Table មួយដែលត្រូវផ្គូរផ្គងជាមួយតំលៃ Primary Key នៅក្នុង Table ​មួយផ្សេងទៀតដើម្បីបង្កើតទំនាក់ទំនងរវាង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

### ២.៧.៣. 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។​

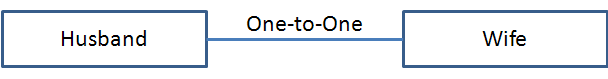
តួលេខ ២.៥ Entity Relationship Diagram



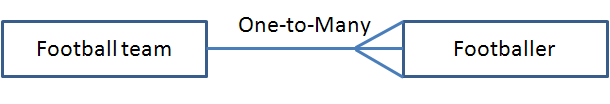
Relationship Diagram ចែកជា​៣ ប្រភេទចំបងគឺៈ

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 ២.៦ One to One Relationship



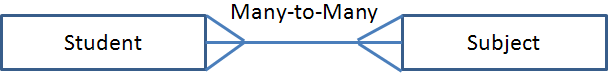
One-to-Many Relationship: សរសេរអក្សរជា 1:M ដែលកើតឡើងនៅពេល Record មួយនៅក្នុង Entity ទី១ទាក់ទង Record ជាច្រើននៅក្នុង Entity ទី២ហើយ Record នីមួយៗនៅក្នុង Entity ទី២ទាក់ទងជាមួយ Record តែមួយគត់នៅក្នុង Entity​ ទី១។

តួលេខ ២.៧ 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 ពីព្រោះព្រឹត្តិការណ៍សំរាប់ភ្ជាប់ 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 ២.៨ Many to Many Relationship



ERD ពេញលេញនៅក្នុងរួបរាងក្រោម ​បង្ហាញរាល់ 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។

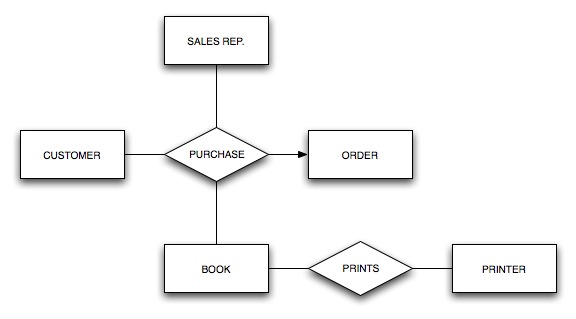
* Normalization

យោងទៅតាមសៀវភៅ Database System ដោយដកស្រងអំពីទ្រឹស្តី Normalization ដែលជាទ្រឹស្តីសំរាប់បំលែង 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 ដែលមានភាពច្រំដែលសំរាប់ 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:

តួលេខ ២.៩ Entity Diagram Relationship



1. វិធីសាស្រ្តស្រាវជ្រាវ

ក្នុងការនីមួយៗដើម្បីឈានរកភាព​ ជោគជ័យទាមទារអោយគេស្វែងរកនូវវិធីសាស្រ្តថាតើធ្វើដូចម្តេច ដើម្បីអោយការងារនោះអាចសំរេចបាន។​ អាស្រ័យហេតុនេះតាម​

រយៈការចុះស្រាវជ្រាវដើម្បីបង្កើតជាប្រព័ន្ធគ្រប់គ្រង ទិន្នន័យការលក់នៅអាហារដ្ឋានឡាយឡាយខេត្តបាត់ដំបងដែលក្រុមយើងខ្ញុំ បានចុះទៅសិក្សាផ្ទាល់យើងបានប្រើប្រាស់វិធីសាស្រ្តដូចជា៖

## ៣.១ វិធីសាស្រ្តកំណត់គំរូតាង

ក្នុងការសិក្សានេះយើងបានធ្វើការស្រាវជ្រាវនៅ អាហារដ្ឋានឡាយឡាយ ដែលមានទីតាំងនៅខេត្តបាត់ដំបង យើងក៏បានប្រើរយៈពេល៤ ខែ ដើម្បីធ្វើការស្រាវជ្រាវប្រមូលទិន្នន័យ និងសកម្មភាពជាច្រើន។ យើងបានបង្កើតជាដ្យាក្រាមដែលបង្ហាញពីវិធីស្រាវជ្រាវក្នុងការបង្កើតប្រព័ន្ធនេះដែលមានដូចខាងក្រោម៖

តួលេខ ៣.១ វិធីសាស្រ្តកំណត់គំរូតាង

ជំហានទី១៖ ចុះសាកសួរព័ត៌មានដោយធ្វើការសម្ភាសន៍អំពីដំនើរការរបស់ប្រព័ន្ធ និងប្រមូលឯកសារដែលពាក់ព័ន្ធនឹងប្រព័ន្ធ

ជំហានទី២៖ វិភាគ និងធ្វើការរៀបចំប្រព័ន្ធ

ជំហានទី៣៖ វិភាគលើចំនុចល្អ និងការខ្វះខាតលើប្រព័ន្ធ

ជំហានទី៤៖​ ធ្វើការកែតំរូវ និង អនុវត្តន៍ប្រព័ន្ធ

ជំហានទី៥៖​ វិភាគ និងគូររូបភាព

ជំហានទី៦៖​ សរសេរកូដ

ក្រុមយើងខ្ញុំបានសិក្សាទៅ លើគំរោងការងារដើម្បីកំនត់ លើចំនុចធំដែល ជាគោលដៅរបស់ប្រព័ន្ធដែលនិងយកទៅធ្វើការ ស្រាវជ្រាវអោយបានល្អិតល្អន់និង ម្យ៉ាងទៀតដើម្បី រកឯកសារទាំងឡាយដែលមានលក្ខណៈប្រហាក់ប្រហែលមកធ្វើការស្រាវជ្រាវផងដែរ។

ជំហានទី១ ៖ ការចុះសំភាសន៍និង ប្រមូលទិន្នន័យនៅអាហារដ្ឋានឡាយឡាយ យកព័ត៌មានយ៉ាងលំអិតពីអាហារដ្ឋាន ដោយការរៀបចំសំនួរជាស្រេច ដើម្បីចង់ដឹងពីដំណើរការ លក់សព្វថ្ងៃមានលក្ខណៈយ៉ាងណា។ ហើយក្រោយពីបានសំភាសន៍ជាមួយអ្នកគ្រប់គ្រង និង​បុគ្គលិករូចរាល់ហើយ យើងបានប្រមូលឯកសារមួយចំនួនដែលពាក់ព័ន្ធ និង​ការប្រើប្រាស់ ប្រព័ន្ធគ្រប់គ្រងការលក់របស់យើង។

ជំហាន់ទី២ ៖ វិភាគ និង ការរៀបចំប្រព័ន្ធ

ក្រោយពីបានសំភាសន៍ និង ប្រមូលបានឯកសារមួយចំនួនមក យើងបានយកទិន្នន័យ ទាំងនោះមកធ្វើការវិភាគប្រសិន បើមានចំនុចណាមួយមិនទាន់ច្បាស់លាស់ រឺ នៅ ខ្វះខាតចំ​នុចណាមួយនោះយើងនិងទៅ សាកសួរម្តងទៀតហើយ ប្រមូលឯកសារបន្ថែម ទៀត។​ ម្យ៉ាងទៀតក្រុមការងារយើង ខ្ញុំបានចុះទៅអង្កេតជាមួយ និង ផ្នែកលក់តែម្តង ដើម្បី ងាយស្រួលយល់អោយកាន់តែច្បាស់ទៅ លើដំណើរការនៃប្រព័ន្ធនេះដែល បង្ករអោយមាន ភាពងាយស្រួលដល់ការរៀបចំនិង គ្រប់គ្រងទៅលើប្រព័ន្ធថ្មីអោយមានប្រសិទ្ធភាព។

ជំហាន់ទី៣ ៖ វិភាគលើចំនុចល្អ​ និងចំនុចខ្វះខាត់របស់ប្រព័ន្ធ

-​ចំនុចល្អរបស់ប្រព័ន្ធៈ ទិន្នន័យអាចរក្សាទុកបានយូរប្រកបដោយសុវត្ថិភាព មានភាពងាយស្រួល ក្នុងការកែប្រែទិន្នន័យចំណេញពេលវេលា បានច្រើន ដែលជាជំនួយក្នុងការ បង្កើនល្បឿនការងារ ជួយការកាត់ បន្ថយការ ភាន់ច្រលំក្នុងការកត់ត្រាទិន្នន័យផ្សេងៗ ទំរង់ក្នុងការបញ្ចូលទិន្នន័យ អនុវត្តន៍ដោយកម្មវិធី Microsoft Visual Studio 2013 និង ការរក្សាទុកទិន្នន័យអនុវត្តន៍ដោយកម្មវិធីMicrosoft SQL Server2008ដែលមានលក្ខណៈ ជា Database System ដែលធ្វើអោយការស្វែងរកទិន្នន័យ​ មានភាពរហ័ស ដែលជួយសំរាលការងារបានយ៉ាងច្រើន​ ។

-ចំនុចខ្វះខាតរបស់ប្រព័ន្ធ​ : យើងយល់ឃើញថាប្រព័ន្ធថ្មីនេះមិនទាន់ មានលក្ខណះល្អពេញលេញនៅឡើយទេ ដោយសារការសិក្សា និងចំណេះដឹងនៅមានកំរិតនៅឡើយ​ រីឯពេលវេលាក៏មានកំណត់ទៀតនោះ ការខ្វះខាតប្រាកដជាមានចំណុចមួយចំនួនដែលត្រូវធ្វើ និង​ ការកែប្រែបន្ថែម ទៀតដើម្បីអោយ ប្រព័ន្ធកាន់​ តែមានប្រសិទ្ធភាពឡើង ។

ជំហានទី៤ ៖ ធ្វើការកែតំរូវ​ និងអនុវត្តន៍ប្រព័ន្ធ

-ការបង្កើតវិក័យប័ត្រលក់ និងរបាយការណ៍ផ្សេងៗ​ ដែលធ្វើដោយក្រដាស់ និងកត់ត្រាដោយដៃ នោះត្រូវប្តូរទៅ​អនុវត្តន៍នៅក្នុងកម្មវិធី Crystal Report 10 ព្រោះមានលក្ខណៈងាយស្រូលហើយឆាប់​ រហ័ស ។

​ -បង្ហាត់បង្រៀនអ្នកប្រើប្រាស់អោយយល់ពីរបៀបប្រើប្រាស់ ប្រព័ន្ធដើម្បីងាយស្រួលដោះស្រាយពេលមានបញ្ហា​ ដែលទាក់ទងទៅ​ នឹងដំណើរការរបស់ប្រព័ន្ធ​។  
 ជំហានទី៥​៖វិភាគនិងគូរតួលេខ ក្នុងជំហាននេះយើងខ្ញុំបានរៀប​ចំប្រព័ន្ធ​ អោយមានលក្ខណះ Database ដោយធ្វើការបង្កើតជាTable, Relationship,

Form, and​ ​Report។  
 ជំហានទី៦​ ៖ សរសេរកូដ

បន្ទាប់ពីបានអនុវត្តន៍​ ជំហានទាំងអស់អោយមាន ភាពច្បាស់លាស់រួចរាល់មកយើងបានចាប់ ផ្តើមសរសេរកូដ ដើម្បីអោយក្លាយជាប្រព័ន្ធ ថ្មីមួយមានលក្ខណះពេញលេញយកទៅប្រើប្រាស់បានដែលប្រកបដោយសុវត្ថិភាពខ្ពស់ក្នុងការរក្សាទិន្នន័យ​ ។

## ៣.២​ វិធីសាស្រ្តប្រមូលទិន្នន័យ

៣.២.១​ប្រភេទនៃទិន្នន័យ ដើម្បីអោយការសិក្សា ស្រាវជ្រាវនេះទទួល បានលទ្ធផលល្អគឺទាមទារ យ៉ាងខ្លាំងនូវពត៌មាននិងឯកសារ​ មួយចំនួនដែលមានការទាក់ទង និងប្រធានបទនៃការសរសេរសារណារនេះ។  
​​ ប្រភេទទិន្នន័យដែលក្រុមយើងខ្ញុំបានប្រមូល និងស្រាវជ្រាវ យកមកត្រូវបានបែងចែកជា២ប្រភេទគឺ៖

-ទិន្នន័យចំបង​(Primary Data)ជាប្រភេទទិន្នន័យសំខាន់ ដែលត្រូវមានជា​ ចាំបាច់នៅពេលប្រមូលទិន្នន័យ ហើយយើងត្រូវមាន វាសំរាប់បង្កើតប្រព័ន្ធថ្មី ។​ ទិន្នន័យនេះអាចទទួលបាន ដោយចុះទៅ ធ្វើការសាកសួរដោយ ផ្ទាល់ពី សំណាក់អ្នកគ្រប់គ្រង និងបុគ្គលិក ផ្សេងៗទៀត ដែលទាកទងនឹងប្រព័ន្ធហើយកត់ត្រាបាននូវចំនុចមួយចំនួនដែលជាបញ្ហាដែលត្រូវដោះស្រាយ។

-ទិន្នន័យបន្ទាប់បន្សំ (Secondary Data) ជាប្រភេទទិន្នន័យដែលប្រើសំរាប់ផ្ទៀងផ្ទាត់កែតំរូវនឹងបំពេញនូវសេចក្តីខ្វះខាតរបស់ទិន្នន័យចំបង។ទិន្នន័យ ប្រភេទ នេះយើងទទួលបានពីការប្រមូលឯកសារដែលពាក់ព័ន្ធនឹងអាហារដ្ឋានដូចជាវិក័យប័ត្រការលក់ (Invoice) ការត្រួតពិនិត្យទំនិញនៅសល់ក្នុងស្តុក​ ​ការកត់ត្រាទិន្នន័យការលក់និងរបាយការណ៍ការលក់៖

​​ - ឯកសារដែលបានមកពីការស្រាវជ្រាវពី បណ្ណាល័យ

- ឯកសារដែលបានមកពីការសិស្សានៅសាកលវិទ្យាល័យបៀលប្រាយ​

​​​​ - សៀវភៅដែលទាក់ទងនិងប្រធានបទ។

### ៣.២.២ វិធីសាស្រ្តប្រមូលទិន្នន័យ

ទិន្នន័យគឺជាធាតុមួយយ៉ាងសំខាន់ក្នុងការបង្កើតប្រព័ន្ធដែលមិនអាចខ្វះបាន។ដូចនេះដើម្បីប្រមូលទិន្នន័យអោយបានគ្រប់គ្រាន់ក្រុមយើងខ្ញុំបានប្រើនូវវិធីសាស្រ្តដូចជាៈ

-តាមការសំភាសន៍ៈយើងបានធ្វើការសំភាសន៍និងសាកសួរដោយផ្ទាល់ពីសំណាក់លោកដែលជាអ្នកគ្រប់គ្រងនិងបុគ្កលិកភោជនីយដ្ឋានអំពីការដំណើរការនៃប្រព័ន្ធចាស់នឹងតំរូវការប្រព័ន្ធថ្មី។

-ការស្រាវជ្រាវទៅលើឯកសារ ឯកសារមុនៗនិងថ្មីដែលមានការពាក់ព័ន្ធយើងបានសិក្សាស្រាវជ្រាវទៅលើឯកសារ​ ដែលបានមកពីភោជនីយដ្ឋាន ពីការសិក្សាកន្លងមកនិងសៀវភៅឯកសារថ្មីៗ ដើម្បីធ្វើកត់ត្រាថតចំលងយកនូវទិន្នន័យទាំងឡាយណាដែលមានការព័ន្ធទៅនឹងប្រធានបទនៃការស្រាវជ្រាវនោះ។

## ៣.៣ វិធីសាស្រ្តវិភាគទិន្នន័យ

បន្ទាប់ពីបាន​ ប្រមូលទិន្នន័យរួចរាល់ហើយយើងខ្ញុំបានយកទិន្នន័យទាំងនោះមក ​ វិភាគយ៉ាងល្អិតល្អន់បង្កើតបានជារូបរាងនៃប្រព័ន្ធដោយយោងទៅលើទ្រឹស្តីការសិក្សាស្រាវបន្ថែមនិងការប្រៀបធៀបរវាងចរន្តការងារ ដែលកំពង់ដំណើរការសព្វថ្ងៃនឹងប្រព័ន្ធថ្មីដែលត្រូវបង្កើតឡើង។ក្នុងការវិភាគនេះយើងយកចំនុចល្អ និងភាពខ្វះខាតរបស់ប្រព័ន្ធយក មកសិក្សាដើម្បីស្វែងរកចំណុចខ្វះខាតរបស់វា ទើបបង្កើតបានជារូបរាងប្រព័ន្ធពេញលេញមួយដែលអាចយកទៅប្រើប្រាស់បាន។

## ៣.៤ រចនាសម្ព័ន្ធនៃការសិក្សា

​​​ នៅក្នុងការរចនាសម្ព័ន្ធនៃការស្រាវជ្រាវនេះយើងបានបែងចែកដូចតទៅៈ

- ជ្រើសរើសប្រធានបទ

-​ ប្រមូលទិន្នន័យ និង​ព័ត៌មានក្នុងការបង្កើតប្រព័ន្ធ

- ស្វែងរកឯកសារស្រាវជ្រាវពីប្រភពផ្សេងៗ

1. លទ្ធផលនៃការស្រាវជ្រាវ

ក្រោពីបានធ្វើការសិក្សារស្រាវជ្រាវមក យើងទទួលបាននៅលទ្ធផលដែលមាន នៅក្នុងជំពូកនេះមានដូចតទៅ :

## **៤.១**ស្ថានភាពទូទៅនៃភោជនីដ្ឋានឡាយឡាយ

ភោជនីដ្ឋានឡាយឡាយ មានទីតាំងនៅក្នុងទីក្រុងបាត់ដំបងនេះតែម្តងដែលស្ថិតនៅក្រោយ **សណ្ឋាគារ ខេមរា** ។ ភោជនីដ្ឋានបានចាប់ផ្តើមបើក​ដំណើរការអារជីវកម្ម របស់ខ្លួននៅក្នុងឆ្នាំ 2012 មកម្លេះ ដែលមានលក្ខណះជាភោជនីដ្ឋានខ្នាតមធ្យមហើយការងាររបស់ភោជនីដ្ឋាន គឺផ្តោតទៅលើ លក់ម្ហូប និង លក់ភេជ្ជគ្រប់ប្រភេទ ។ ភោជីដ្ឋាននេះសព្វថ្ងៃមាន​នាក់គ្រប់គ្រង១នាក់ និង បុគ្គលិកទាំងអស់ ៧ នាក់ រួមទាំងសន្តិសុខផងដែល ។ អ្នកគ្រប់គ្រងមានតួនាទីគ្រប់គ្រង ទៅលើដំណើរការ ផ្សេងរបស់ភោជនីដ្ឋាន ជាពិសេសផ្នែកលក់ដែលជា ការងារដ៏ចំបងមានដូចជា : ស្តុក , របាយការណ័ការលក់ និង​ រៀបចំឯកសារផ្សេង រី​ ឯបុគ្គលិកមានតួនាទីលក់នូវម្ហូប និង ភេជ្ជះទៅតាមការកម្មង់របស់អតិថិជន ។​

## **៤**.**២ ស្ថានភាពទូទៅនៃប្រពន្ធ័ចាស់**

តាមរយះការចុះសិក្សារប្រពន្ធ័ផ្ទាល់យើងបានដឹងពីដំណើរការរបស់ប្រពន្ធ័យ៉ាងច្បាស់នូវស្ថានភាពដែលកំពុងដំណើរការនាពេលបច្ចុប្បន្នដែលមានលក្ខណះដូចជា :

\_ ក្រោយពីបានពិនិត្យស្តុកភេសជ្ជះ និង របស់ផ្សេងៗ អ្នកគ្រប់គ្រង ត្រូ​វរៀបចំដោយ របាយការណ៍ទៅអោយប្រធាន ដើម្បីទិញនូវអ្វីដែលបានលក់អស់​ ។

\_ បន្ទាប់ពីការរៀបចំឯកសារដូចជា វិក័យបត្រ័លក់ (Invoice) ។ របាយការណ័ផ្សេងៗ ត្រូវបានធ្វើឡើងដោយដៃ ។ បន្ទាប់ពីការរៀបចំឯកសារ ហើយការកត់ត្រាទិន្ន័យលក់ ត្រូវធ្វើឡើងដោយដៃផ្ទាល់ដដែល ។

# **៤.៣**. **ការបង្កើតប្រពន្ធ័ថ្មី**

ក្នុងការសិក្សារស្រាវជ្រាវបង្កើតប្រពន្ធ័ថ្មីនេះ ក្រុមយើងខ្ញុំបានអនុវត្តន៏ទៅតាមជំហ៊ានវដ្ត ជីវិតនៃការបង្កើតប្រពន្ធ័ ( SDLC ) ដូចដែលបានរៀបរាប់ ក្នុងជំពូករំលឹកទ្រឹស្តីដើម្បីផ្សា ភ្ជាប់ទ្រឹស្តីទៅនិងការអនុវត្តន៏ផ្ទាល់ដែលមានវីធីសាស្រ្តដូចជា :

### ៤.៣.១ការរៀបចំគម្រោងប្រពន្ធ័ (​ System Planning )

ក្នុងការបង្កើតប្រព័ន្ធនេះ ក្រុមយើងខ្ញុំបានបង្កើតជាតារាង គំរោងនៃការសិក្សារស្រាវជ្រាវ ដើម្បីធ្វើការបែកចែក និង គ្រប់គ្រងការងារ ដែលត្រូវទៅតាមជំហ៊ាន នីមួយៗដែលមានដូចជា ខាងក្រោម :

តារាង ៤.១ ការរៀបគម្រោងប្រព័ន្ធ

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Name | Start | Finish |
| 1 | Restaurant Management System | 01/01/2015 | 31/04/2015 |
| 2 | Planning : | 05/01/2015 | 10/01/2015 |
| 3 | Analysis: | 10/01/2015 | 31/02/2015 |
| 4 | Design: | 01/03/2015 | 20/03/2015 |
| 5 | Implementation: | 21/04/2015 | 31/04/2015 |

### ៤.៣.២ការវិភាគប្រព័ន្ធ(System Analysis)

### **៤.៣.២.១.​ ការវិភាគប្រពន្ធ័ការងារ និង ការកែប្រែ**

ប្រពន្ធ័បច្ចុប្បន្ននេះមិនមាន លក្ខណះជា Database System ទេ​ដោយ ការងារមួយចំនួនធំនៅប្រព្រឹត្តទៅដោយការកត់ត្រាដោយដៃនៅឡើយហើយការងារមួយចំនួនទៀតធធ្វើឡើងនៅកម្មវិធី Microsoft World​ ។ ដោយយល់ឃើញការលំបាកនេះ យើងបានបង្កើតជាប្រពន្ធ័ថ្មីនេះតែម្តងព្រោះវាមានលក្ខណះជា Database System​ ងាយស្រួលគ្រប់គ្រង និង​ ចំណេញពេលវេលាទៀតផង ។ ការបង្កើតប្រពន្ធ័ថ្មីនេះឡើងយើងបានប្រមូលយក នូវឯកសាររបស់ភោជនីដ្ឋានបានមួយចំនួនយក​ មកផ្គួបផ្សំ និង ការចុះពិនិត្យការងារដែលកំពុងដំណើរការបច្ចុប្បន្ន យើងបានបង្កើតបានជាប្រពន្ធ័ សំរាប់គ្រប់គ្រងការលក់នៅភោជនីដ្ឋានផ្ទាល់ ដោយកែប្រែពីការងារកត់ត្រាដោយដៃមកធ្វើនៅ កុំព្យូទ័តវិញ ។

ចំពោះប្រពន្ធ័គ្រប់គ្រង ការលក់ថ្មីនេះមាននាទីគ្រប់គ្រងនៅរាល់ដំណើរការនៃការប្រែប្រួល នៅក្នុងស្តុកក៏ដូចជា ជួយសំរួលក្នុងការផ្នែកលក់បានមួយផ្នែកធំផងដែល ដូចជា : ម្ហូប រឺភេសជ្ជះដែលកបានលក់ និង ភេសជ្ជះដែលមាននៅក្នុងស្តុកដើម្បីងាយស្រួលក្នុងការ ធ្វើរបាយការណ៏លក់ ។ ដើម្បីងាយគ្រប់គ្រង ប្រពន្ធ័ដែលមានលក្ខណះជា​ Database ត្រូវរៀបចំបែងចែកមុខម្ហូប និង​ ភេសជ្ជះតាមប្រភេទនីមួយៗ បង្កើតជា Table សំរាប់ផ្ទុកទិន្ន័យ និង រៀបចំរបាយការណ៍មុខម្ហូប ភេសជ្ជះដែលបានលក់ ។ បន្ទាប់ពីបានសិក្សាលើប្រពន្ធ័ ការងារសព្វថ្ងៃឃើញថា មិនមានលក្ខណះល្អទាំងស្រុងនោះទេ ទាំងការផ្ទុកទិន្ន័យ ( Store Data ) ទាំងការរកទិន្ន័យ ( Search Data ) ជាដើម្បីសុទ្ធតែមានការខ្វះខាតយ៉ាងច្រើន ។ ដូចនេះប្រពន្ធ័ថ្មីដែលក្រុមយើងខ្ញុំគ្រោងហ្នឹងធ្វើនេះ គឺយើងជ្រើសយកកម្មវិធី Microsoft SQL Server 2008 សំរាប់ផ្ទុកទិន្ន័យ និង កម្មវិធី Microsoft Visual studio 2013 សំរាប់រចនា ទំរង់ប្រើប្រាស់ដើម្បីបញ្ជូលទិន្ន័យដែល និង​មានផលប្រយោជន៍ដូចជា៖

​ -​ ការផ្ទុកទិន្នន័យមានទំហំធំ និងមានសុវត្ថិភាពខ្ពស់

- ស្វែងរកទិន្នន័យបានលឿនដោយបានបង្កើតនូវលក្ខខ័ណ្ខស្វែងរក ​ ទិន្នន័យ (Search) ដែលមានច្រើនពិបាកនិងរកបានឆាប់រហ័ស

* ​​​ការទទួលព័ត៌មានក៏បានរហ័សដែលដោយយើងបានរៀបចំរបាយការណ៍ជា ស្រេចចំពោះតំរូវការរបស់អ្នកប្រើប្រាស់
* ចំណេញពេលវេលាក្នុងការធ្វើវិក័យប័ត្រជូនអតិថិជន។

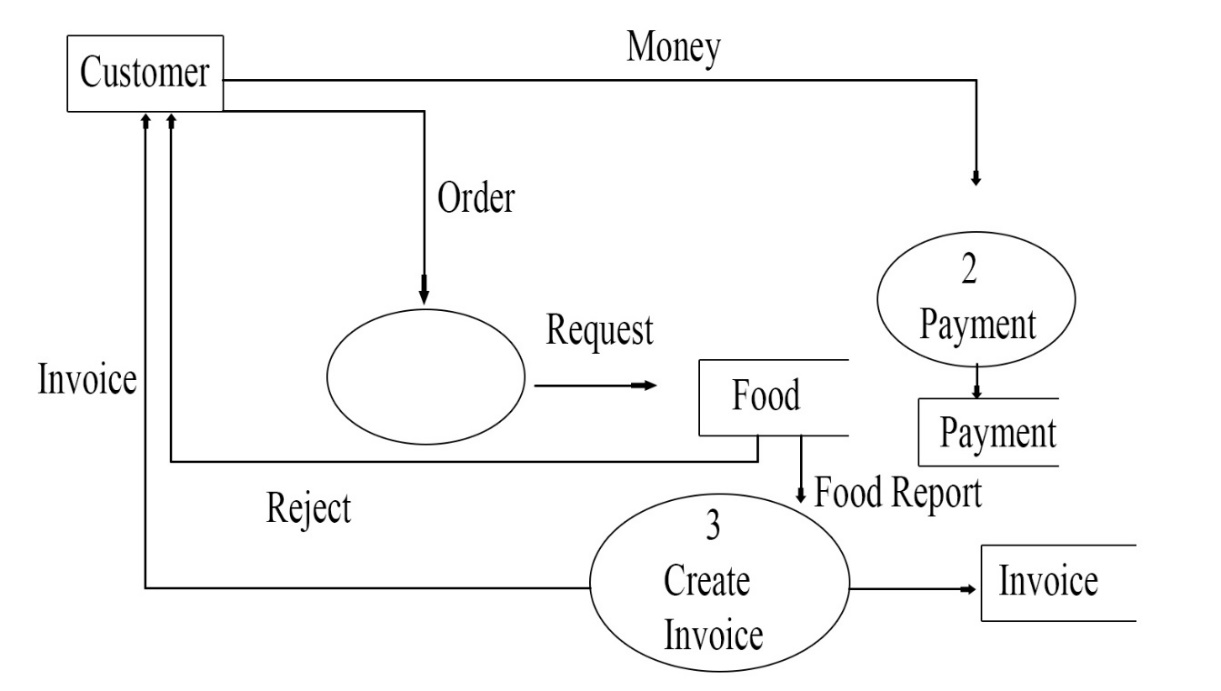
### **៤.៣.២.២ Context Diagram**

តួលេខ ៤.២ Context Diagram

****

### **៤.៣.២.៣ Data Flow Diagram(DFD)**

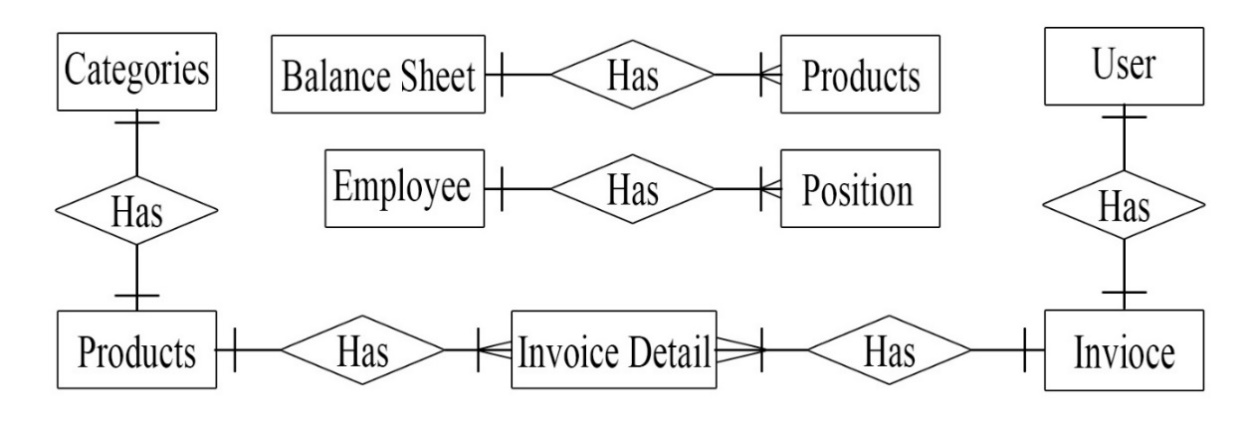
តួលេខ ៤.៣ Data Flow Diagram

****

### ៤.៣.៣ ការរៀបចំរចនាប្រព័ន្ធ**(System Design)**

### **៤.៣.៣.១ System Design**

តួលេខ ៤.៤ System Design

**៤.៣.៣.១ Entity Relationship Diagram (ERD)**

Foods ោថដ

Balance Sheet

ថដ

Food Typeោថដ

Position ោថដ

Staff ោថដ

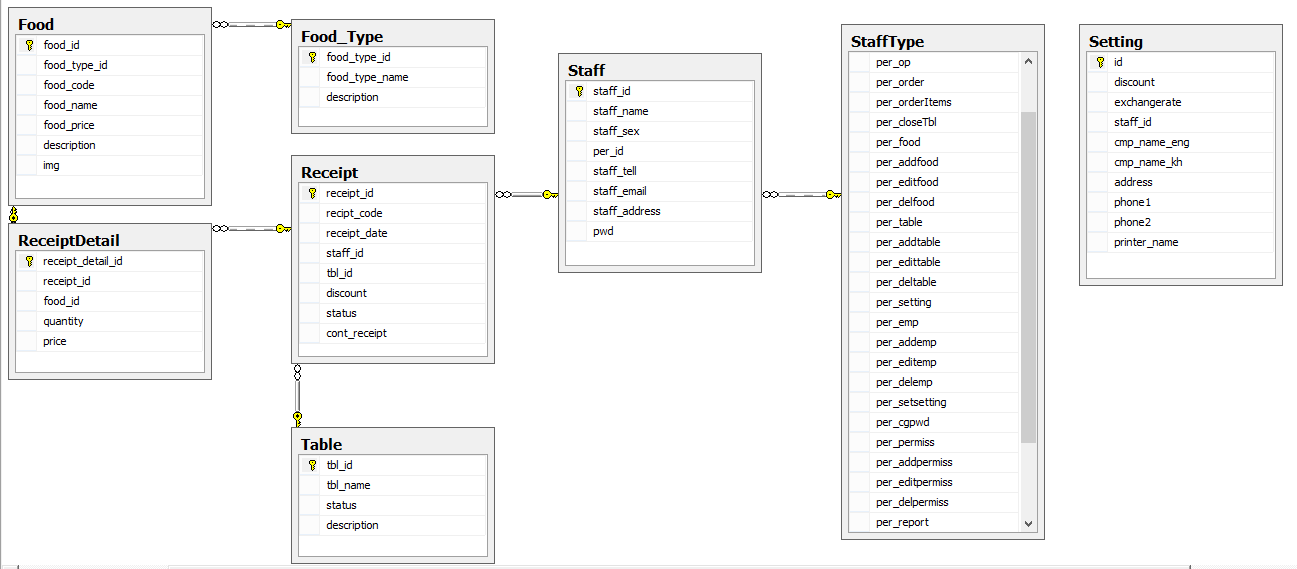
ថដ

Receipt Detail

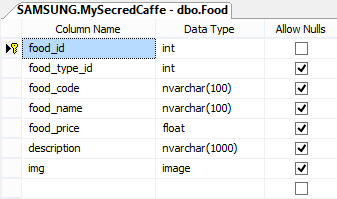
Receiptោថដ

Foods ោថដ

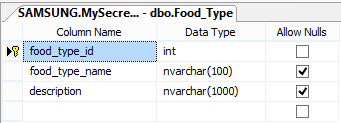
### **៤.៣.៣.២ Table Diagram**

****តួលេខ ៤.៥ Table Diagram

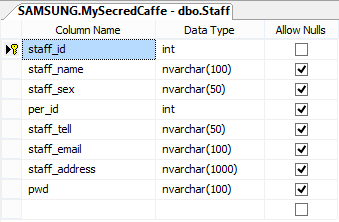
### **៤.៣.៣.៣ Data Dictionary (DD)**

1. Table Food

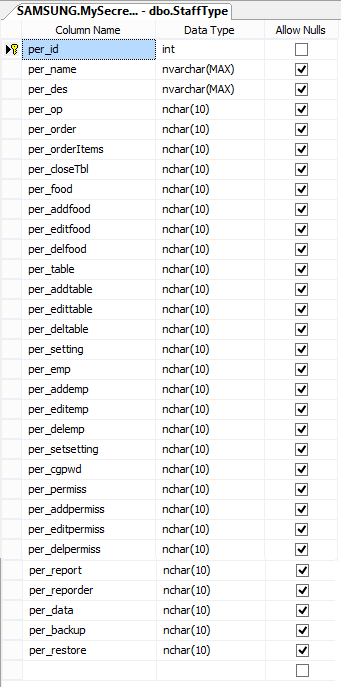
2.Table Food Type

****

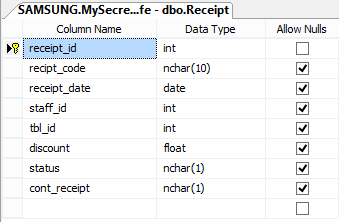
3. Table Staff



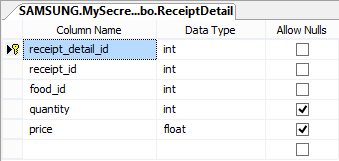
4. Table Staff Detail



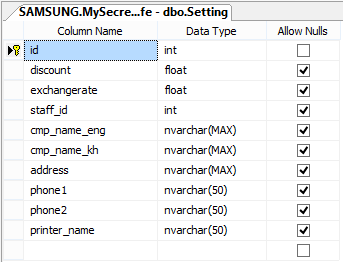
5. Table Receipt

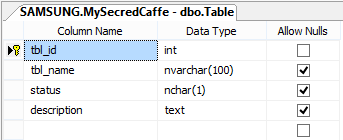


6. Table Receipt Detail

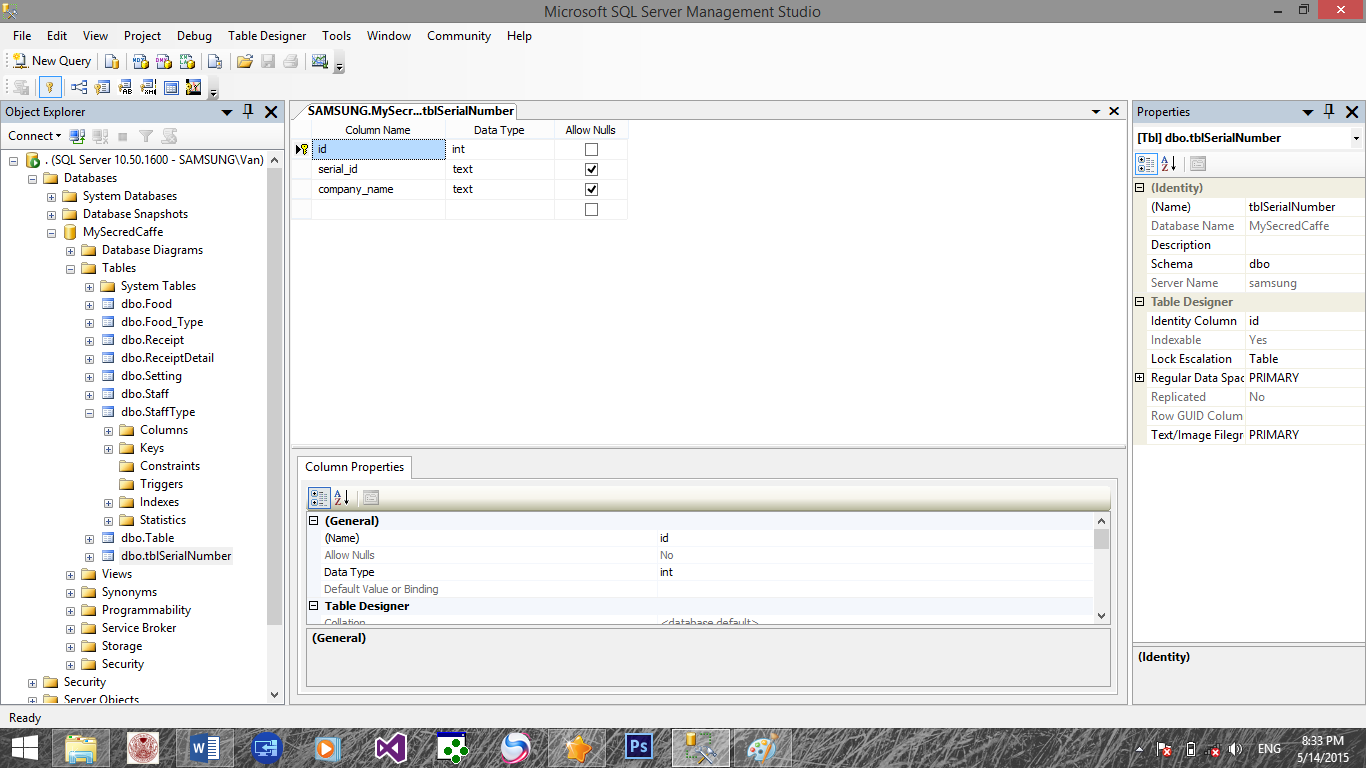


7. Table Setting



 8. Table Tables

9. Table Serial

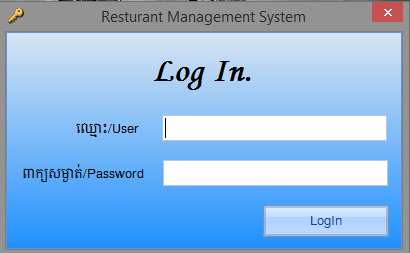


**៤.៣.៣.៤ ការគ្រោងបង្កើត User Interface**

ក្នុងកម្មវិធីរបស់ប្រព័ន្ធគ្រប់គ្រងការលក់នេះយើងបានរៀបចំបង្កើតអោយ មានលក្ខណៈជាទំរង Interface ជាច្រើនដោយបែងចែកមុខងារនីមួយៗដែលត្រូវ បានប្រើក្នុងការគ្រប់គ្រងការលក់តែម្តង។​ ដើម្បីអាចដំណើរការកម្មវិធីនេះយើងចាំបាច់ត្រូវមានៈ

1. Form Login

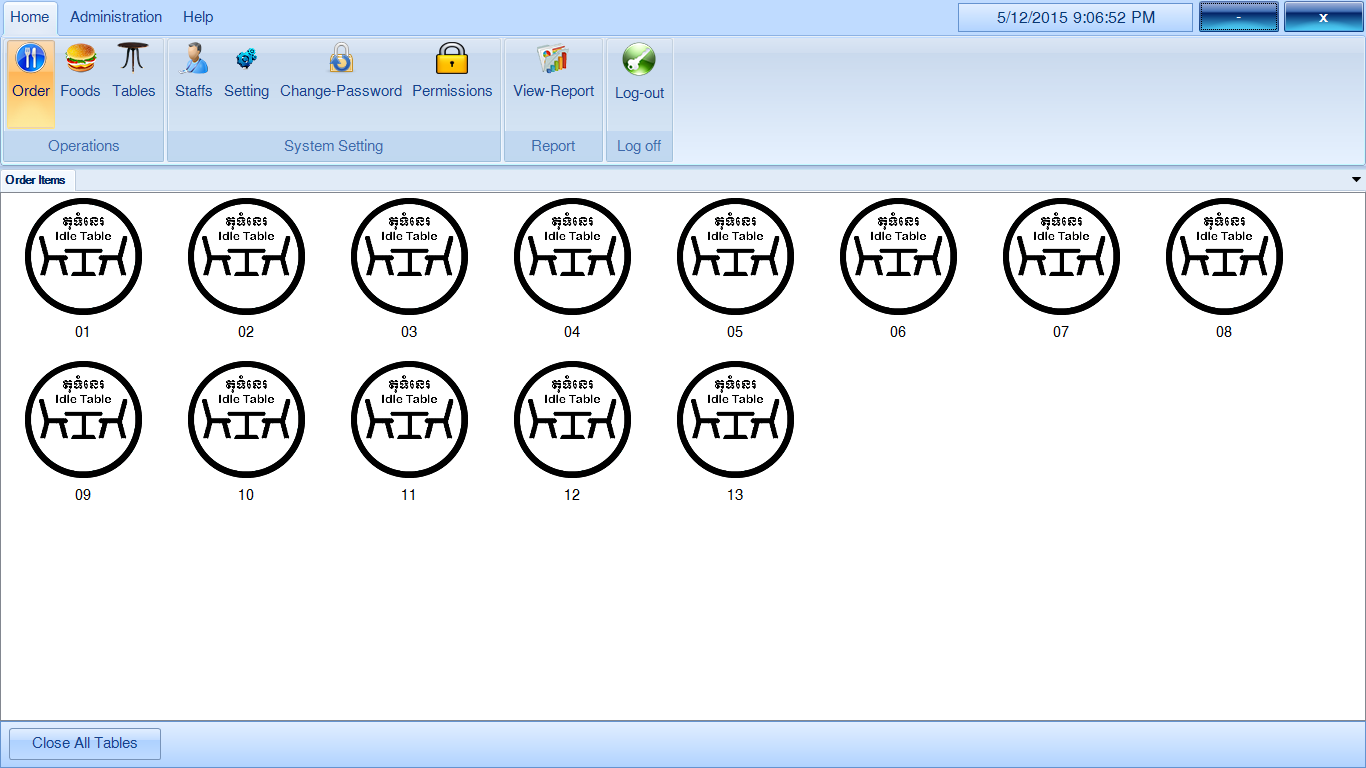
រូបភាព ៤.១ Form Login



បន្ទាប់ពីបានបញ្ចូលឈ្មោះនិង ពាក្យសម្ងាត់បានត្រឺមត្រូវហើយ លោកអ្នកអាច ប្រើប្រាស់កម្មវិធីនេះបាន ហើយវានិង ប​ង្ហាញ Main Form មានទំរងដូចខាងក្រោមៈ

1. Main Form

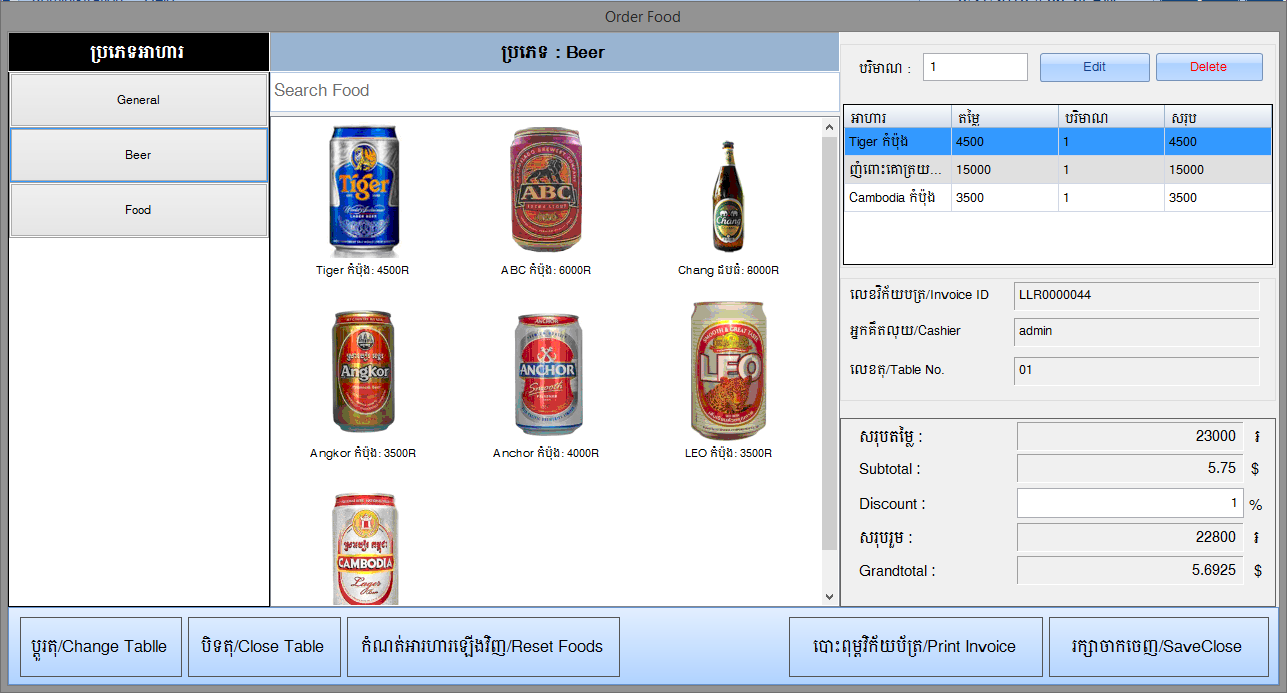
រូបភាព ៤.២ Form Main



Form នេះសំរាប់ដំណើរការរាល់សកម្មភាពរបស់ ប្រព័ន្ធទាំងមូលតែម្តងហើយ ក្នុងForm នេះអ្នកប្រើប្រាស់អាចប្រើបាននូវ Form ផ្សេងៗទៀតទៅតាមតំរូវការ។

1. Form Order Food

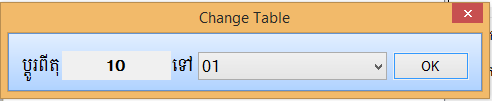
រូបភាព ៤.៣ Form Order Food



ដើម្បីដំណើរ Form នេះUser ត្រូវ Right Click លើតុណាមួយដែលមានការកម្មង់ហើយជ្រើសរើសយក Add New។

1. Form Change Table

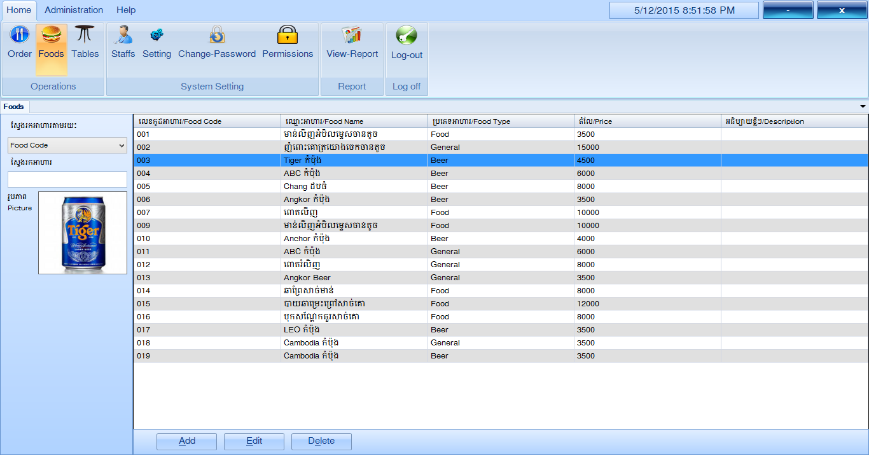
រូបភាព ៤.៤ Form Change Table



Form នេះសំរាប់ធ្វើការផ្លាស់ប្តូរតុទៅតាមតំរូវការរបស់អតិថិជន។

1. Form Food

រូបភាព ៤.៥ Form Food



Form នេះសំរាប់បង្ហាញឈ្មោះមុខម្ហូប និងភេសជ្ជៈ ។

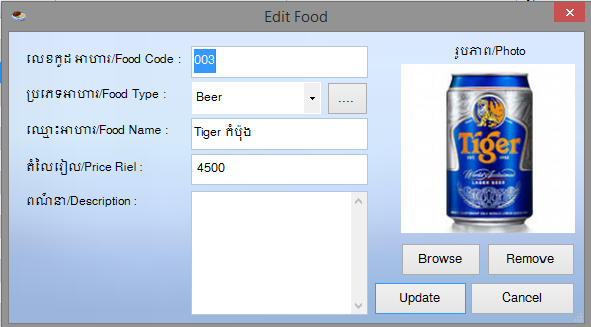
1. Form Add Food

រូបភាព ៤.៦ Form Add Food



Form នេះសំរាប់បន្ថែមឈ្មោះមុខម្ហូប និងភេសជ្ជៈចូលទៅក្នុង Form Food

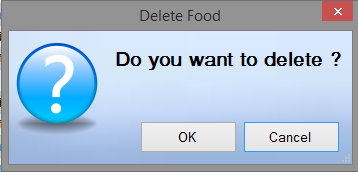
1. Form Edit Food

រូបភាព ៤.៧ Form Edit Food

Form នេះប្រើសំរាប់កែប្រែមុខម្ហូបរឺភេសជ្ជៈផ្សេងៗ។

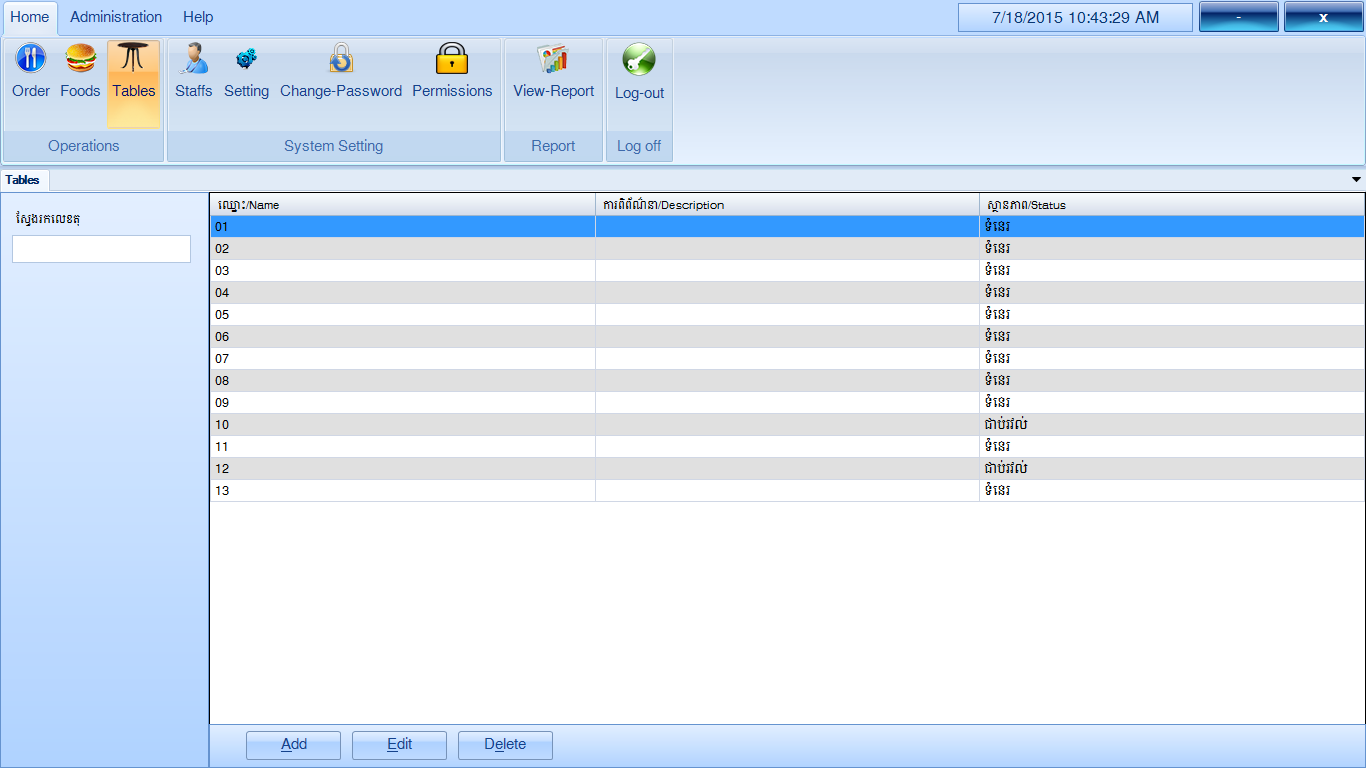
1. Form Delete Foods

រូបភាព ៤.៨ Form Delete Food



Form នេះប្រើសំរាប់លុបមុខម្ហូបរឺភេសជ្ជៈពីតារាងមុខម្ហូបក្នុងប្រព័ន្ធ។

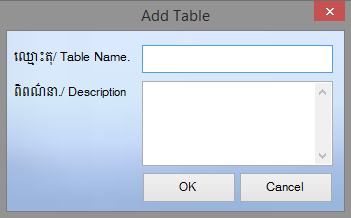
1. Form Table

រូបភាព ៤.៩ Form Table

Form នេះបង្ហាញពីចំនួនតុ, ស្ថានភាពតុ និង​លេខរបស់តុក្នុងហាង។ Form នេះយើងអាចបន្ថែម កែប្រែ និងលុបចំនួនតុ។

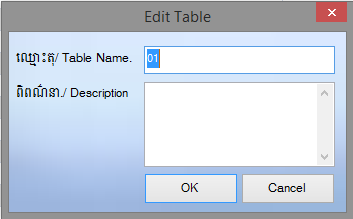
1. Form Add Table

រូបភាព ៤.១០ Form Add Table



Form នេះសំរាប់បន្ថែមចំនួនតុក្នុងហាង។

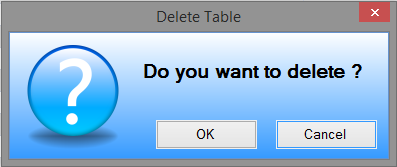
1. Form Edit Table

រូបភាព ៤.១១ Form Edit Table

Form​នេះប្រើសំរាប់កែប្រែលេខតុនិង ព័ត៌មានរបស់តុ។

1. Form Delete Table

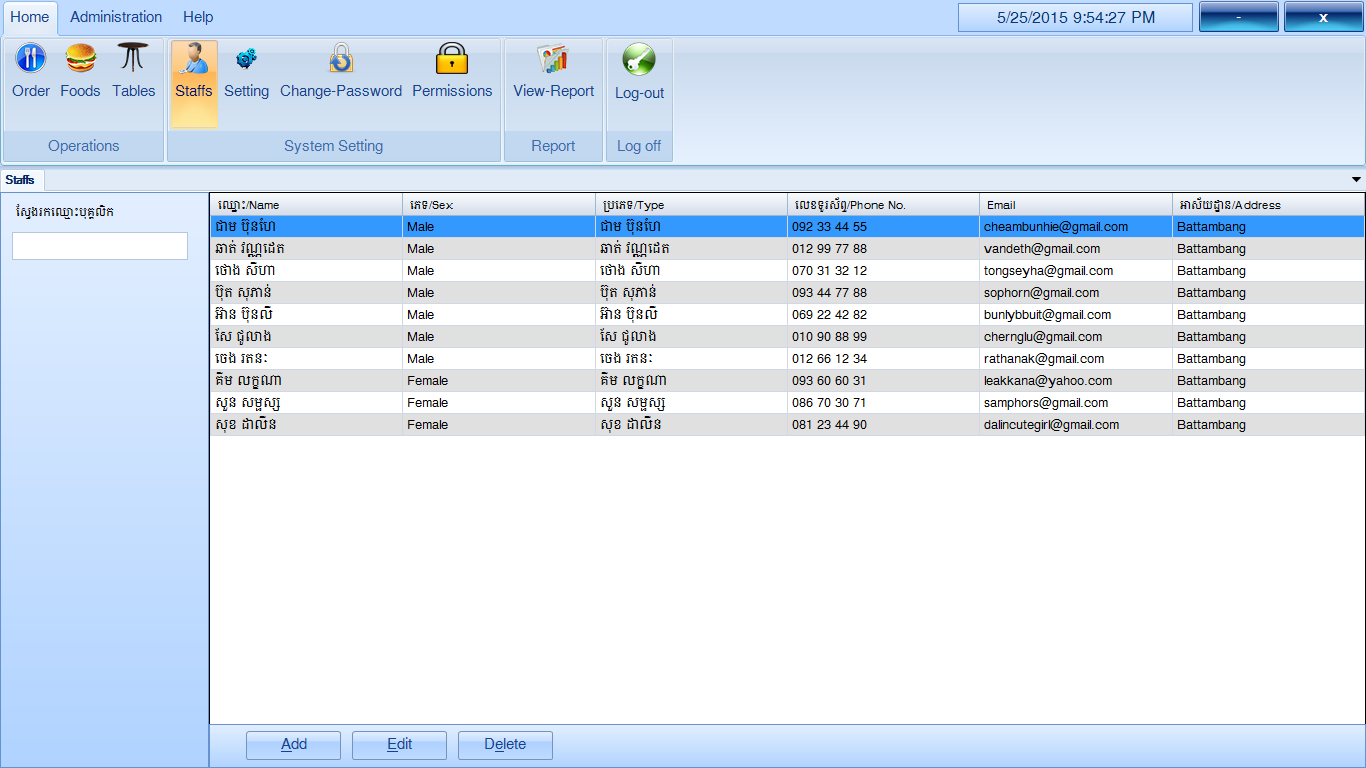
រូបភាព ៤.១២ Form Delete Table



Form នេះប្រើសំរាប់លុបតុចេញពី Form Table។

1. Form Staff

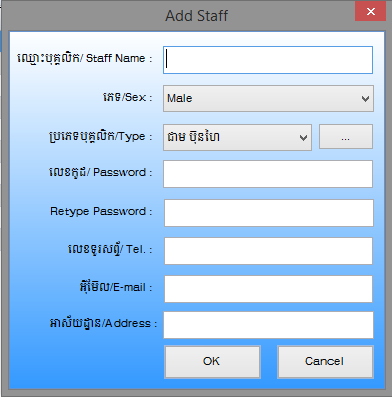
រូបភាព ៤.១៣ Form​ Staff



Form នេះប្រើសំរាប់បង្ហាញព័ត៌មានរបស់បុគ្គលិក។

1. Form Add Staff

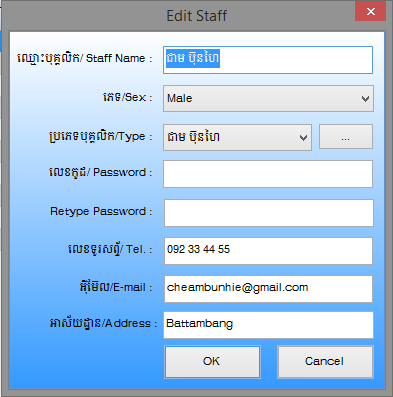
រូបភាព ៤.១៤ Form Add Staff



Formនេះប្រើសំរាប់បញ្ចូលព័ត៌មានរបស់បុគ្គលិកនិងជ្រើសរើសសិទ្ធរបស់បុគ្គលិកនីមួយៗ

1. Form Edit Staff

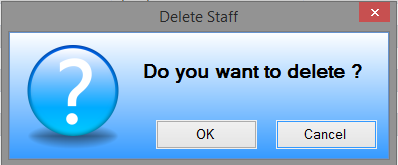
រូបភាព ៤.១៥ Form Edit Staff



Form នេះប្រើសំរាប់កែប្រែព័ត៌មានរបស់បុគ្គលិនដែលបានបញ្ចូលហើយ។

1. Form Delete Staff

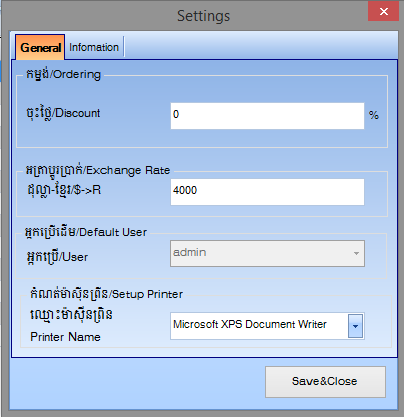
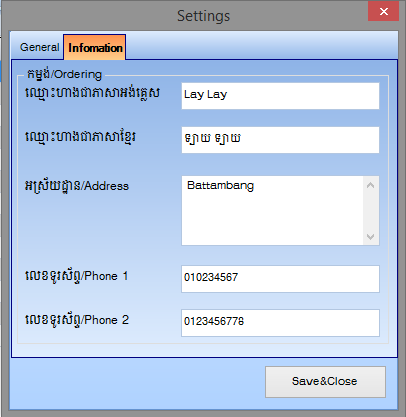
រូបភាព ៤.១៦ Form Delete Staff



Form នេះប្រើសំរាប់លុបនូវព័ត៌មានរបស់បុគ្គលិក។

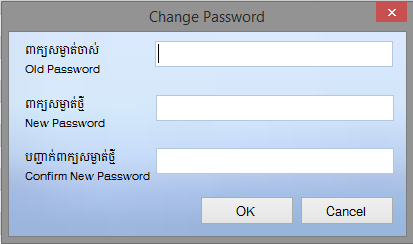
1. Form Setting

រូបភាព ៤.១៧ Form Setting



Form នេះប្រើសំរាប់កំណត់ការបញ្ចុះតំលៃ អត្រាប្តូរប្រាក់និង ព័ត៌មានរបស់ប្រព័ន្ធ។

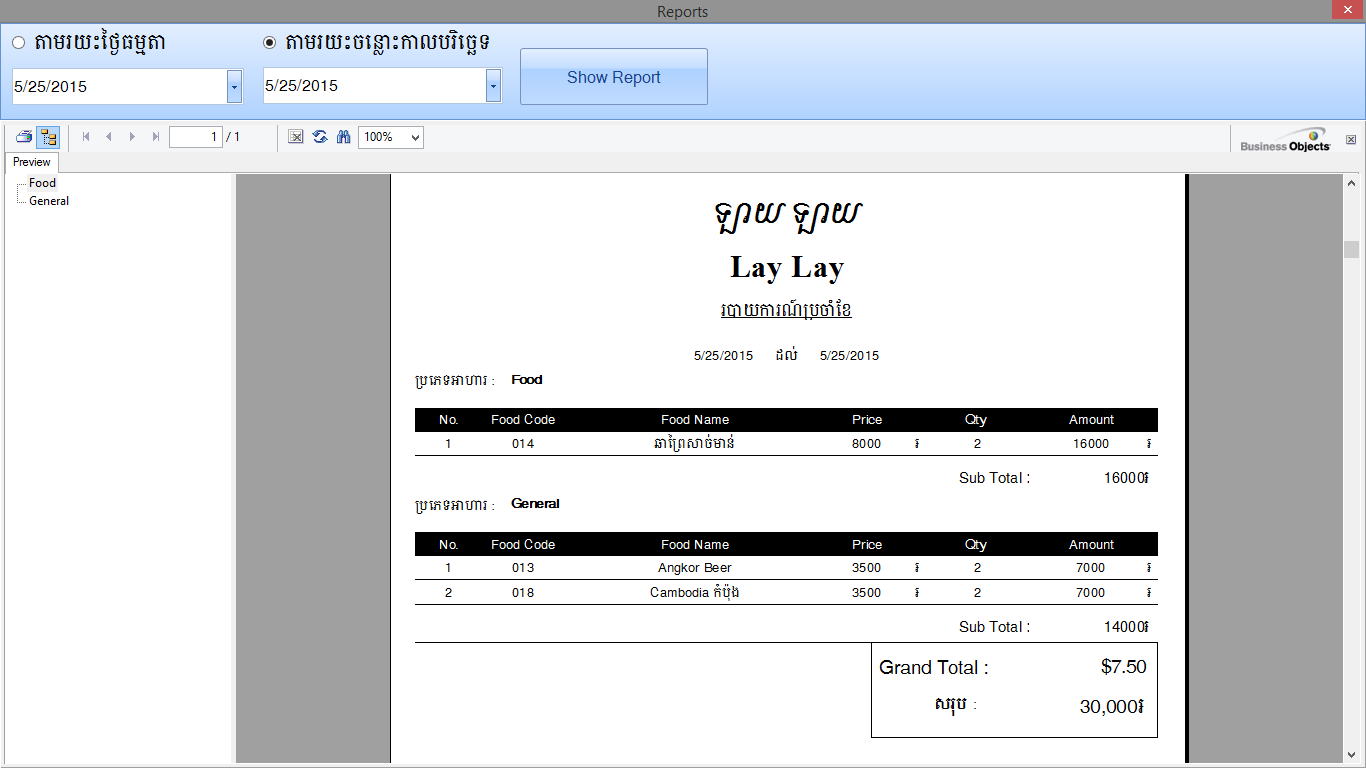
1. Form Change Password

រូបភាព ៤.១៨ Form Change Password

Form នេះប្រើសំរាប់ធ្វើការផ្លាស់ប្តូរលេខសម្ងាត់ថ្មីរបស់ប្រព័ន្ធ។

1. Form View Report

រូបភាព ៤.១៩ Form View Report



Form នេះប្រើសំរាប់បង្ហាញនូវរបាយការណ៍ដែលបានលក់របស់អាហារដ្ឋាន។

## **៤.៤**. **តំរូវការ Hardware និង Software​**

**ប្រព័ន្ធគ្រប់គ្រងនេះបង្កើតឡើងដោយអាចប្រើបានជាលក្ខណៈ**Desktop Application **ដូចនេះមានតំរូវការ** Hardware​ **និង** Software **ទាបបំផុតសំរាប់ដំណើរការប្រព័ន្ធនេះត្រូវមានដូចខាងក្រោម៖**

- CPU 1GHz

- RAM 256MB

- VGA Monitor

- Keyboard

- Mouse

- Hard Disk 40GB

- Microsoft windows 7 Professional

1. **ការពិភាក្សា**

ជាការឆ្លើយតបទៅ និង គំរោងនៃការបញ្ចប់កាសិក្សាផ្នែកព័ត៌មានវិទ្យានៅសាកលវិទ្យាល័យ បៀលប្រាយ ក្រុមនិស្សិតបានជ្រើសរើសប្រធានបទមួយដោយសិក្សាទៅលើប្រព័ន្ធការលក់ ម្ហូបអាហារនិង ភេសជ្ជះ នៅអាហារដ្ឋានឡាយឡាយ មកធ្វើជាប្រធានបទនៅក្នុងការសរសេរនិក្ខេបបទ ដើម្បី បញ្ចប់ ការសិក្សា ។ យើងខ្ញុំបានបំលែងប្រព័ន្ធការលក់ធ្វើដោយដៃមកធ្វើលើប្រព័ន្ធ Computer វិញ ។

ក្រោយពីក្រុមនិសិ្សតយើងខ្ញុំ បានចុះស្រាវជ្រាវប្រមូលពត៏មាន និង ទិន្នន័យមួយចំនួនដែលទាក់ទង និង ការលក់នៅអាហារដ្ឋាននេះហើយ យើងបានយកទិន្នន័យទាំងអស់នេះទៅធ្វើការវិភាគ ដើម្បីបង្កើតបាន Database ថ្មីមួយដែលប្រព័ន្ធស្តីពីការលក់ ។ យើងខ្ញុំបានធ្វើការប្រើប្រាស់ Microsoft SQL Server 2008 បង្កើត Database (Back-End) នៅក្នុង Personal Computer (PC) ម៉្យាងវិញទៀត យើងបានជ្រើសរើសយក Microsoft Visual Studio 2013(Front End) ក្នុងការបង្កើត From សរសេរ Code ។

## **៥.១**.គុណសម្បត្តិរបស់ប្រព័ន្ធថ្មី

\_ ប្រព័ន្ធថ្មីមានសុវត្តិភាពខ្ពស់

\_ អាចបោះពុម្ពជារបាយការណ៏ (Report) និងវិក័យប័ត្រ (Invoice)

បានស្រួល

\_ Userងាយស្រួលប្រើប្រាស់

\_ ការគ្រប់គ្រងទិន្នន័យមានលក្ខណះជា Database

\_ មិនមានភាពភាន់ច្រលំក្នុងការគិតលុយ

\_ បានចូលរួមចំណែកមួយផ្នែកក្នុងកាអភិវឌ្ឍន៍វិស័យព័ត៌មានវិទ្យា

## **៥.២**.គុណវិបត្តិរបស់ប្រព័ន្ធថ្មី

\_ ត្រូវការពេលេវេលាស្រាវជ្រាវច្រើនក្នុងកាបង្កើតប្រព័ន្ធ

­\_ ប្រព័ន្ធនេះប្រើកម្មវិធីច្រើនមានកាពិបាកក្នុងការស្រាវជ្រាវ

\_ មិនបានផ្តោតទៅលើការប្រើប្រាស់ជាលក្ខណះបណ្តាញ

(Network)

1. **សេចក្តីសន្និដ្ឋាន និងអនុសាសន៍**

## **៦.១**.សេចក្តីសន្និដ្ឋាន

នេះជាប្រព័ន្ធថ្មីដែល នៅក្នុងអាហារដ្ឋានឡាយឡាយ មិនធ្លាប់ប្រើពីមុនមានទាល់តែសោះដូចនេះហើយក្រោយពីការបង្កើតប្រព័ន្ធនេះរួចហើយ យើងអាចសនិ្នដ្ឋានបានថាវាមានភាពងាយស្រួលជាងប្រព័ន្ធចាស់ព្រោះ :

-ងាយស្រួលក្នុងការគ្រប់គ្រង មុខម្ហូប និង ភេសជ្ជះព្រមទាំងងាយស្រួលក្នុងការរុក រកផង​ ហើយថែមទាំងផ្ទុកចំនួន មុខម្ហូប ភេសជ្ជះ និង​ ទិន្នន័យ បានច្រើននិងអាចបញ្ចូល រឺ ក៏កែប្រែបានភ្លាមៗ ។

-ជាលទ្ធផលចំពោះការបង្កើតប្រព័ន្ធនេះ មានលទ្ធភាពគ្រប់គ្រងទៅលើផ្នែកការលក់ ព្រមទាំងជួយសំរួលដល់ការអាហារដ្ឋាន កត់ត្រាទុកនូវពតមានការចំណូល និង ចំណាយ ដែលជាភាពងាយស្រួលក្នុងការធ្វើរបាយការណ័ ។

-កាត់បន្ថយកំហុស និង ភាពច្រលំដែលមាននៅក្នុងសកម្មភាពការងារ ។

### **៦.២**.អនុសាសន៍

ដោយសព្វថ្ងៃនេះ វិស័យពត៏មាន មានការរីកចំរើនយ៉ាងខ្លាំងនៅលើពិភពលោក ជាពិសេសនៅក្នុងប្រទេសកម្ពុជា ដែលមានការសិក្សាផ្នែកព័ត៌មានវិទ្យាតាមគ្រឹះស្ថានឧត្តមសិក្សាផងដែរ ។

ដូចនេះក្រុមស្រាវជ្រាវមានបំណងចង់ធ្វើអោយប្រព័ន្ធនៃការលក់ អោយកាន់តែមានប្រសិទ្ធភាពថៃមទៀតផង និង ការប្រើប្រាស់ងឲ្យកាន់តែមានភាពទូលំទូលាយ ។

យើងខ្ញុំក៏បានបង្កើត គំនិតដែលាមានការសំណូមពរដូចជា :

-សូមនិសិ្សតជំនាន់ក្រោយជួយអភិវឌ្ឍន៍នូវ Interface ឲ្យមានភាពងាយ ស្រួលប្រើដែលក្រុមយើងខ្ញុំបានបង្កើត ។

-អាហារដ្ឋានគួរតែបង្កើតនូវ Website របស់ខ្លួនដើម្បីបានស្គាល់កាន់តែច្រើន ។

-ត្រូវសិក្សាស្រាវជ្រាវបន្ថែមទៀត ទៅលើប្រព័ន្ធនេះ ដើម្បីការប្រើប្រាស់កាន់តែទូលំទួលាយ និង ទទួលការគាំទ្រ ។

ឯកសារយោង

1. Kennet C.Laudon, Jane P.Laudon(2002), Management Information System: Managing the Firm, 7th  Edition. Prentice-Hall,Inc

2. <https://en.wikipedia.org/wiki/Microsoft_SQL_Server>

3. <https://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Express>

**Form Login**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

using Microsoft.Win32;

using System.Management;

using System.Data.SqlClient;

namespace SecredCaffe

{

public partial class frmLogIn : Form

{

string st;

string ServerName;

string DataName;

double newEncryptKeyDc = 0;

string boardSerial = "";

double newBoardEncryptDc = 0;

public frmLogIn()

{

InitializeComponent();

this.lblInfo.Visible = false;

this.txtUser.SelectAll();

this.txtUser.Focus();

}

public string getMainBoadID()

{

try

{

ManagementObjectSearcher mos = new ManagementObjectSearcher("SELECT SerialNumber FROM Win32\_BaseBoard");

ManagementObjectCollection moc = mos.Get();

foreach (ManagementObject mo in moc)

{

boardSerial = mo["SerialNumber"].ToString();

}

return boardSerial;

}

catch (Exception)

{

return boardSerial;

}

}

private void frmLogIn\_Load(object sender, EventArgs e)

{

//check if exited sub key

RegistryKey subKey = Registry.LocalMachine.OpenSubKey(@"SYSTEM\Select\syncight");

if (subKey != null)//exited

{

//\*Compare encrypted motherboard//

//Create EncryptKey//

string encryptKey = "SSHS7";

Encoding ascii = Encoding.ASCII;

Encoding unicode = Encoding.Unicode;

// Convert the string into a byte array.

byte[] unicodeBytes = unicode.GetBytes(encryptKey);

string newEncryptKey = "";

for (int i = 0; i < unicodeBytes.Length; i++)

{

newEncryptKey += unicodeBytes[i].ToString();

}

newEncryptKeyDc = Convert.ToDouble(newEncryptKey);

//getMother board serial encrypted

string motherBoardID = getMainBoadID();

ascii = Encoding.ASCII;

unicode = Encoding.Unicode;

unicodeBytes = unicode.GetBytes(motherBoardID);

string newBoardID = "";

for (int i = 0; i < unicodeBytes.Length; i++)

{

newBoardID += unicodeBytes[i].ToString();

}

newBoardEncryptDc = Convert.ToDouble(newBoardID);

newBoardEncryptDc += newEncryptKeyDc;

//get value from regidit

if (!(subKey.GetValue("boadSerial").Equals(newBoardEncryptDc.ToString())))//VALUE IN REGISTRY NOT EQUALS ENCRYPTED MOTHERBOARD

{

//open form for register serial

frmRegister frmrg = new frmRegister();

frmrg.ShowDialog();

if (frmrg.DialogResult == DialogResult.Cancel)

{

this.Close();

}

}

subKey.Close();

}

else//no subkey syncight

{

//open form for register serial

frmRegister frmrg = new frmRegister();

frmrg.ShowDialog();

if (frmrg.DialogResult == DialogResult.Cancel)

{

this.Close();

}

}

this.txtUser.SelectAll();

this.txtUser.Focus();

}

//declare variable to count log in not success

int a = 0;

private void btnLogin\_Click(object sender, EventArgs e)

{

//log in code

AccessData mydata = new AccessData();

int count = mydata.logIn("Staff", "staff\_name", txtUser.Text, "pwd", txtPwd.Text);

if (count > 0)

{

this.Hide();

AccessData acc = new AccessData();

ServerName = acc.serverName;

DataName = acc.dbName;

SqlConnection con = new SqlConnection("Data Source=" + ServerName + ";initial Catalog=" + DataName + "; Integrated Security=True");

con.Open();

SqlCommand cmd1 = new SqlCommand("select per\_id from Staff where staff\_name = '" + txtUser.Text + "'", con);

st = cmd1.ExecuteScalar().ToString();

DialogResult dr = new DialogResult();

frmMain frmmain = new frmMain(txtUser.Text, txtPwd.Text);//passing arguments to constructor

frmmain.permis = st;

dr = frmmain.ShowDialog();

if (dr == DialogResult.OK)

{

this.Close();

}

}

else

{

this.Refresh();

a += 1;//increase counter of login in failed

this.lblInfo.Visible = true;

txtUser.SelectAll();

txtUser.Focus();

if (a == 3)

{

this.Close();

}

}

}

private void frmLogIn\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

}

}

**Form Main**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.IO;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace SecredCaffe

{

public partial class frmMain : Form

{

public string permis;

public string userName = "";

public string pwd = "";

string ServerName;

string DataName;

public void setUserPwd(string u, string p)

{

userName = u;

pwd = p;

}

private void Permissions()

{

AccessData acc = new AccessData();

SqlCommand cmd = new SqlCommand("select \* from StaffType where per\_id=" + permis + "", acc.connectDatabase(acc.serverName, acc.dbName));

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

this.btnOrder.Enabled = Convert.ToBoolean(dr["per\_order"].ToString());

this.ctOrder.Enabled = Convert.ToBoolean(dr["per\_orderItems"].ToString());

this.btnCloseAllTbl.Visible = Convert.ToBoolean(dr["per\_closeTbl"].ToString());

this.btnFood.Enabled = Convert.ToBoolean(dr["per\_food"].ToString());

this.btnAddFood. Enabled = Convert.ToBoolean(dr["per\_addfood"].ToString());

this.btnEditFood. Enabled = Convert.ToBoolean(dr["per\_editfood"].ToString());

this.btnDelFood. Enabled = Convert.ToBoolean(dr["per\_delfood"].ToString());

this.btnTable.Enabled = Convert.ToBoolean(dr["per\_table"].ToString());

this.btnAddTbl. Enabled = Convert.ToBoolean(dr["per\_addtable"].ToString());

this.btnEditTbl. Enabled = Convert.ToBoolean(dr["per\_edittable"].ToString());

this.btnDelTbl. Enabled = Convert.ToBoolean(dr["per\_deltable"].ToString());

this.btnStaff.Enabled = Convert.ToBoolean(dr["per\_emp"].ToString());

this.btnAddStaff. Enabled = Convert.ToBoolean(dr["per\_addemp"].ToString());

this.btnEditStaff. Enabled = Convert.ToBoolean(dr["per\_editemp"].ToString());

this.btnDelStaff. Enabled = Convert.ToBoolean(dr["per\_delemp"].ToString());

this.btnSetting.Enabled = Convert.ToBoolean(dr["per\_setsetting"].ToString());

this.btnChangePwd.Enabled = Convert.ToBoolean(dr["per\_cgpwd"].ToString());

this.btnPermiss.Enabled = Convert.ToBoolean(dr["per\_permiss"].ToString());

this.btnAddPermiss. Enabled = Convert.ToBoolean(dr["per\_addpermiss"].ToString());

this.btnEditPermiss. Enabled = Convert.ToBoolean(dr["per\_editpermiss"].ToString());

this.btnDelPermiss. Enabled = Convert.ToBoolean(dr["per\_delpermiss"].ToString());

this.btnReport.Enabled = Convert.ToBoolean(dr["per\_reporder"].ToString());

this.btnBackup.Enabled = Convert.ToBoolean(dr["per\_backup"].ToString());

this.btnRestore.Enabled = Convert.ToBoolean(dr["per\_restore"].ToString());

}

dr.Close();

}

public string getUser()

{

return userName;

}

public string getPwd()

{

return pwd;

}

AccessData mydata = new AccessData();

public frmMain()

{

InitializeComponent();

}

public frmMain(string user,string pwd)//i put the paramaters here to get value from other form

{ //when create instance object from this form by passing arguments

InitializeComponent();

this.setUserPwd(user, pwd);

}

//funtion to control permission

private void DGpropertye()

{

DGFood.Columns[0].Visible = false;

DGFood.Columns[1].Visible = false;

DGFood.Columns[2].HeaderText = "លេខកូដអាហារ/Food Code";

DGFood.Columns[3].HeaderText = "ឈ្មោះអាហារ/Food Name";

DGFood.Columns[4].HeaderText = "ប្រភេទអាហារ/Food Type";

DGFood.Columns[5].HeaderText = "តំលែ/Price";

DGFood.Columns[6].HeaderText = "អធិប្បាយខ្លីៗ/Description";

DGFood.Columns[7].Visible = false;

DGFood.ColumnHeadersHeightSizeMode = DataGridViewColumnHeadersHeightSizeMode.DisableResizing;

}

private void frmMain\_Load(object sender, EventArgs e)

{

Permissions();

this.Cursor = Cursors.WaitCursor;//set wait cursor

this.tabOrder.Visible = true;

this.tabFood.Visible = false;

this.tabTable.Visible = false;

this.tabStaff.Visible = false;

this.tabPermiss.Visible = false;

this.btnOrder.Checked = true;

this.btnFood.Checked = false;

this.btnTable.Checked = false;

this.btnStaff.Checked = false;

this.btnPermiss.Checked = false;

//call method to add image to listview item

AccessData mydata = new AccessData();

string cmdSt = "SELECT \* FROM [Table]";

SqlCommand cmdd = new SqlCommand(cmdSt, mydata.connectDatabase(mydata.serverName,mydata.dbName));

SqlDataReader objReader = cmdd.ExecuteReader();

lsvTable.Items.Clear();

while (objReader.Read())

{

//add item with image to listview

ListViewItem listViewItem = new ListViewItem(objReader["tbl\_name"].ToString(), Convert.ToInt16(objReader["status"].ToString()));

listViewItem.SubItems.Add(objReader["tbl\_id"].ToString());

listViewItem.SubItems.Add(objReader["status"].ToString());

lsvTable.Items.Add(listViewItem);

}

//here i count the record from table Table to fit with listview lsvInv in order to store invNo by index

if (DGFood.Rows.Count == 0)

{

btnEditFood.Enabled = false;

btnDelFood.Enabled = false;

picFood.Load(Application.StartupPath + "\\Photo\\NoRecord.bmp");

}

else

{

btnEditFood.Enabled = true;

btnDelFood.Enabled = true;

}

this.Cursor = Cursors.Default;

}

private void btnOrder\_Click(object sender, EventArgs e)

{

this.tabOrder.Visible = true;

this.tabFood.Visible = false;

this.tabTable.Visible = false;

this.tabStaff.Visible = false;

this.tabPermiss.Visible = false;

this.btnOrder.Checked = true;

this.btnFood.Checked = false;

this.btnTable.Checked = false;

this.btnStaff.Checked = false;

this.btnPermiss.Checked = false;

frmMain\_Load(sender, e);

}

private void loadToCheckTab()

{

this.tabOrder.Visible = false;

this.tabFood.Visible = false;

this.tabTable.Visible = false;

this.tabStaff.Visible = false;

this.tabPermiss.Visible = false;

//--------------------------------//

this.btnOrder.Checked = false;

this.btnFood.Checked = false;

this.btnTable.Checked = false;

this.btnStaff.Checked = false;

this.btnPermiss.Checked = false;

}

private void btnFood\_Click(object sender, EventArgs e)

{

//this.btnDelFood.Enabled = false;

this.tabFood.Visible = true;

this.tabOrder.Visible = false;

this.tabTable.Visible = false;

this.tabStaff.Visible = false;

this.tabPermiss.Visible = false;

this.btnOrder.Checked = false;

this.btnFood.Checked = true;

this.btnTable.Checked = false;

this.btnStaff.Checked = false;

this.btnPermiss.Checked = false;

//set background color

this.DGFood.BackgroundColor = Color.White;

//----------------------------------//

this.cboSerchBy.SelectedIndex = 0;

//fill datagridview for customer

AccessData data = new AccessData();

string cmdStr = "SELECT dbo.Food.food\_id, dbo.Food\_Type.food\_type\_id, dbo.Food.food\_code, dbo.Food.food\_name, dbo.Food\_Type.food\_type\_name, dbo.Food.food\_price, dbo.Food.description, dbo.Food.img FROM dbo.Food LEFT JOIN dbo.Food\_Type ON dbo.Food.food\_type\_id = dbo.Food\_Type.food\_type\_id";

data.fillDataGridView(this.DGFood,cmdStr);

DGpropertye();

if (DGFood.Rows.Count == 0)

{

btnEditFood.Enabled = false;

btnDelFood.Enabled = false;

picFood.Load(Application.StartupPath + "\\Photo\\NoRecord.bmp");

}

else

{

Permissions()

DisplayPicToFormLoad();

}

}

private void btnTable\_Click(object sender, EventArgs e)

{

this.tabTable.Visible = true;

this.tabOrder.Visible = false;

this.tabFood.Visible = false;

this.tabStaff.Visible = false;

this.tabPermiss.Visible = false;

this.btnOrder.Checked = false;

this.btnFood.Checked = false;

this.btnTable.Checked = true;

this.btnStaff.Checked = false;

this.btnPermiss.Checked = false;

//set background color

this.dtgTable.BackgroundColor = Color.White;

//fill datagridview for customer

AccessData data = new AccessData();

string cmdStr = "SELECT tbl\_id, tbl\_name, description FROM [Table]";

data.fillDataGridView(this.dtgTable,cmdStr);

//set property for Datagridview

this.dtgTable.Columns[0].Visible = false;

this.dtgTable.Columns[1].HeaderCell.Value = "ឈ្នោះ/Name";

this.dtgTable.Columns[2].HeaderCell.Value = "ការពិព័ណ៌នា/Description";

this.dtgTable.ColumnHeadersHeightSizeMode = DataGridViewColumnHeadersHeightSizeMode.DisableResizing;

}

private void btnBackup\_Click(object sender, EventArgs e)

{

string getP;

string Fname;

AccessData Acc = new AccessData();

ServerName = Acc.serverName;

DataName = Acc.dbName;

SaveFileDialog fldlg = new SaveFileDialog();

fldlg.Filter = "BackUp File (\*.bak)|\*.bak;";

string filebk;

if (fldlg.ShowDialog() == DialogResult.OK)

{

try

{

this.Cursor = Cursors.WaitCursor;

getP = Path.GetPathRoot("C:\\");

filebk = fldlg.FileName;

Fname = Path.GetPathRoot(filebk);

if (Fname == getP)

{

MessageBox.Show(this, "You can not backup file to drive C \n Please backup to another drive !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

this.Cursor = Cursors.Default;

return;

}

Connection conn = new Connection();

conn.connectDatabase(ServerName, DataName);

string str1 = "USE MySecredCaffe; BACKUP DATABASE MySecredCaffe TO DISK = '" + filebk + "' with init,stats=2,MEDIANAME = 'Z\_SQLServerBackups',NAME = 'Full Backup of MySecredCaffe';";

SqlCommand cmd2 = new SqlCommand(str1, conn.connectDatabase(ServerName, DataName));

cmd2.ExecuteNonQuery();

cmd2.Dispose();

conn.closeDatabase();

frmBR bk = new frmBR();

bk.lblMsg.Text = "The bakup operation has been \n completed successfully.";

bk.Text = "Backup Database";

bk.frmBK = "bk";

bk.ShowDialog();

this.Cursor = Cursors.Default;

}

catch

{

this.Cursor = Cursors.Default;

MessageBox.Show(this, "Sorry the backup is error ! \n Please try to backup again", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void btnRestore\_Click(object sender, EventArgs e)

{

OpenFileDialog fldlg = new OpenFileDialog();

fldlg.Filter = "Restore File (\*.bak)|\*bak;";

string filebk;

if (fldlg.ShowDialog() == DialogResult.OK)

{

try

{

AccessData acc = new AccessData();

ServerName = acc.serverName;

DataName = acc.dbName;

this.Cursor = Cursors.WaitCursor;

filebk = fldlg.FileName;

SqlConnection con = new SqlConnection("Data Source=" + ServerName + ";initial Catalog=" + DataName + "; Integrated Security=True");

con.Open();

string st1 = "USE master; ALTER DATABASE MySecredCaffe SET SINGLE\_USER with ROLLBACk IMMEDIATE";

string st2 = "RESTORE DATABASE MySecredCaffe FROM DISK = '" + filebk + "' WITH REPLACE, STATS = 2;";

SqlCommand cmd1 = new SqlCommand(st1, con);

SqlCommand cmd2 = new SqlCommand(st2, con);

cmd1.ExecuteNonQuery();

cmd2.ExecuteNonQuery();

frmBR bk = new frmBR();

bk.lblMsg.Text = "The Restore operation has been \n completed successfully.\n\n System need to restart.";

bk.Text = "Restore Database";

bk.frmBK = "res";

bk.ShowDialog();

this.Cursor = Cursors.Default;

}

catch

{

this.Cursor = Cursors.Default;

MessageBox.Show(this, "Sorry the restore is error ! \n Please try to restore again", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void btnStaff\_Click\_1(object sender, EventArgs e)

{

this.tabStaff.Visible = true;

this.tabOrder.Visible = false;

this.tabFood.Visible = false;

this.tabTable.Visible = false;

this.tabPermiss.Visible = false;

this.btnOrder.Checked = false;

this.btnFood.Checked = false;

this.btnTable.Checked = false;

this.btnStaff.Checked = true;

this.btnPermiss.Checked = false;

//set background color

this.dtgStaff.BackgroundColor = Color.White;

//fill datagridview for customer

AccessData data = new AccessData();

string cmdStr = "SELECT dbo.Staff.staff\_id, dbo.Staff.staff\_name, dbo.Staff.staff\_sex, dbo.StaffType.per\_id, dbo.StaffType.per\_name, dbo.Staff.staff\_tell, dbo.Staff.staff\_email, dbo.Staff.staff\_address FROM dbo.Staff LEFT JOIN dbo.StaffType ON dbo.Staff.per\_id = dbo.StaffType.per\_id WHERE dbo.Staff.staff\_name<>'admin'";

data.fillDataGridView(this.dtgStaff,cmdStr);

//set property for Datagridview

this.dtgStaff.Columns[0].Visible = false;

this.dtgStaff.Columns[1].HeaderCell.Value = "ឈ្នោះ/Name";

this.dtgStaff.Columns[2].HeaderCell.Value = "ភេទ/Sex";

this.dtgStaff.Columns[3].Visible=false;

this.dtgStaff.Columns[4].HeaderCell.Value="ប្រភេទ/Type";

this.dtgStaff.Columns[5].HeaderCell.Value = "លេខទូរស័ព្ទ/Phone No.";

this.dtgStaff.Columns[6].HeaderCell.Value = "Email";

this.dtgStaff.Columns[7].HeaderCell.Value = "អាស័យដ្ឋាន/Address";

this.dtgStaff.ColumnHeadersHeightSizeMode = DataGridViewColumnHeadersHeightSizeMode.DisableResizing;

if (dtgStaff.Rows.Count == 0)

{

btnEditStaff.Enabled = false;

btnDelStaff.Enabled = false;

}

else

{

btnEditStaff.Enabled = true;

btnDelStaff.Enabled = true;

}

}

private void btnDelFood\_Click(object sender, EventArgs e)

{

try

{

string Fid;

frmDeleteFood DelFood = new frmDeleteFood();

Fid = DGFood.SelectedRows[0].Cells[0].Value.ToString();

DelFood.Fid = Fid;

DelFood.delFood = "delFood";

DelFood.ShowDialog();

if (DelFood.DialogResult == DialogResult.OK)

{

btnFood\_Click(sender, e);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void btnAbout\_Click(object sender, EventArgs e)

{

frmAbout frm = new frmAbout();

frm.ShowDialog();

}

private void txtFoodName\_TextChanged(object sender, EventArgs e)

{

//call function filter

//AccessData datafilter = new AccessData();

//datafilter.filterDatagridView("Food", this.DGFood,"food\_name", this.txtSerchFood.Text);

string sql;

AccessData Acc = new AccessData();

ServerName = Acc.serverName;

DataName = Acc.dbName;

Connection conn = new Connection();

conn.connectDatabase(ServerName, DataName);

SqlCommand cmd = new SqlCommand();

SqlDataAdapter DA = new SqlDataAdapter();

DataSet DS = new DataSet();

if (cboSerchBy.SelectedIndex==0)

{

sql = "SELECT dbo.Food.food\_id, dbo.Food\_Type.food\_type\_id, dbo.Food.food\_code, dbo.Food.food\_name, dbo.Food\_Type.food\_type\_name, dbo.Food.food\_price, dbo.Food.description, dbo.Food.img FROM dbo.Food LEFT JOIN dbo.Food\_Type ON dbo.Food.food\_type\_id = dbo.Food\_Type.food\_type\_id WHERE Food.food\_code LIKE N'" + this.txtSerchFood.Text + "%'";

cmd.CommandText = sql;

cmd.Connection = conn.connectDatabase(ServerName, DataName);

DA.SelectCommand = cmd;

DS.Clear();

DA.Fill(DS, "tblFood");

DGFood.DataSource = DS.Tables["tblFood"];

DA.Dispose();

conn.closeDatabase();

}

else

{

sql = "SELECT dbo.Food.food\_id, dbo.Food\_Type.food\_type\_id, dbo.Food.food\_code, dbo.Food.food\_name, dbo.Food\_Type.food\_type\_name, dbo.Food.food\_price, dbo.Food.description, dbo.Food.img FROM dbo.Food LEFT JOIN dbo.Food\_Type ON dbo.Food.food\_type\_id = dbo.Food\_Type.food\_type\_id WHERE Food.food\_name LIKE N'" + txtSerchFood.Text + "%'";

cmd.CommandText = sql;

cmd.Connection = conn.connectDatabase(ServerName, DataName);

DA.SelectCommand = cmd;

DS.Clear();

DA.Fill(DS, "tblFood");

DGFood.DataSource = DS.Tables["tblFood"];

DA.Dispose();

conn.closeDatabase();

}

if (DS.Tables["tblFood"].Rows.Count == 0)

{

picFood.Load(Application.StartupPath + "\\Photo\\NoRecord.bmp");

btnEditFood.Enabled = false;

btnDelFood.Enabled = false;

}

if (DGFood.Rows.Count > 0)

{

btnEditFood.Enabled = true;

btnDelFood.Enabled = true;

if (picFood.Image != null)

picFood.Image.Dispose();

AccessData Ac = new AccessData();

ServerName = Acc.serverName;

DataName = Ac.dbName;

Connection con = new Connection();

conn.connectDatabase(ServerName, DataName);

SqlCommand cmds = new SqlCommand("ReadImage", conn.connectDatabase(ServerName, DataName));

cmds.CommandType = CommandType.StoredProcedure;

cmds.Parameters.Add("@fid", SqlDbType.Int).Value = DGFood.Rows[0].Cells[0].Value.ToString();

SqlDataAdapter adp = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

if (conn.connectDatabase(ServerName, DataName).State == ConnectionState.Closed)

conn.connectDatabase(ServerName, DataName).Open();

adp.Fill(dt);

if (dt.Rows.Count > 0)

{

MemoryStream ms = new MemoryStream((byte[])dt.Rows[0]["img"]);

picFood.Image = Image.FromStream(ms);

picFood.SizeMode = PictureBoxSizeMode.StretchImage;

picFood.Refresh();

}

}

}

private void txtTblName\_TextChanged(object sender, EventArgs e)

{

//call function filter

AccessData datafilter = new AccessData();

string cmdStr = "SELECT \* FROM [Table] WHERE tbl\_name LIKE N'"+ this.txtTblName.Text +"%'";

datafilter.filterDatagridView(this.dtgTable,cmdStr);

if (dtgTable.Rows.Count > 0)

{

btnEditTbl.Enabled = true;

btnDelTbl.Enabled = true;

}

else

{

btnEditTbl.Enabled = false;

btnDelTbl.Enabled = false;

}

}

private void btnDelTbl\_Click(object sender, EventArgs e)

{

AccessData acc=new AccessData();

string oTID = this.dtgTable.SelectedRows[0].Cells[0].Value.ToString();

SqlCommand cmdChkTvl = new SqlCommand("select status from [Table] where tbl\_id=" + oTID + "", acc.connectDatabase(acc.serverName, acc.dbName));

string chkTbl = cmdChkTvl.ExecuteScalar().ToString();

if (!chkTbl.Equals("0"))

{

MessageBox.Show("អ្នកមិនអាចលុបតុនេះបានទេព្រោះតុនេះកំពុងជាប់រវល់\nYou can't delete this table because it is busy !", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

return;

}

try

{

AccessData data = new AccessData();

frmDeleteS frmdeleteS = new frmDeleteS();

DialogResult dialogResult = new DialogResult();

string Tname = data.getDataStrID("[Table]", "tbl\_name", "tbl\_id", oTID);

frmdeleteS.frm = "T";

frmdeleteS.SID = oTID;

frmdeleteS.Text = "Delete Table";

dialogResult = frmdeleteS.ShowDialog();

if (dialogResult == DialogResult.OK)

{

//fill datagridview for Table

string cmdStr = "SELECT tbl\_id, tbl\_name, description FROM [Table]";

data.fillDataGridView(this.dtgTable, cmdStr);

}

}

catch

{

MessageBox.Show("Please select record to delete!");

}

}

private void txtStaffName\_TextChanged(object sender, EventArgs e)

{

//call function filter

AccessData datafilter = new AccessData();

string cmdStr = "SELECT dbo.Staff.staff\_id, dbo.Staff.staff\_name, dbo.Staff.staff\_sex, dbo.StaffType.per\_id, dbo.StaffType.per\_name, dbo.Staff.staff\_tell, dbo.Staff.staff\_email, dbo.Staff.staff\_address FROM dbo.Staff LEFT JOIN dbo.StaffType ON dbo.Staff.per\_id = dbo.StaffType.per\_id WHERE dbo.Staff.staff\_name<>'admin' AND Staff.staff\_name LIKE N'" + this.txtStaffName.Text + "%'";

datafilter.filterDatagridView(this.dtgStaff,cmdStr);

if (dtgStaff.Rows.Count > 0)

{

btnEditStaff.Enabled = true;

btnDelStaff.Enabled = true;

}

else

{

Permissions()

}

}

private void btnDelStaff\_Click(object sender, EventArgs e)

{

try

{

frmDeleteS frmdeleteS = new frmDeleteS();

DialogResult DLR = new System.Windows.Forms.DialogResult();

string SID = dtgStaff.SelectedRows[0].Cells[0].Value.ToString();

frmdeleteS.SID = SID;

string Sname = dtgStaff.SelectedRows[0].Cells[1].Value.ToString();

frmdeleteS.frm = "S";

DLR = frmdeleteS.ShowDialog();

if (DLR == DialogResult.OK)

{

AccessData data = new AccessData();

string cmdStr = "SELECT dbo.Staff.staff\_id, dbo.Staff.staff\_name, dbo.Staff.staff\_sex, dbo.StaffType.per\_id, dbo.StaffType.per\_name, dbo.Staff.staff\_tell, dbo.Staff.staff\_email, dbo.Staff.staff\_address FROM dbo.Staff LEFT JOIN dbo.StaffType ON dbo.Staff.per\_id = dbo.StaffType.per\_id WHERE dbo.Staff.staff\_name<>'admin'";

data.fillDataGridView(this.dtgStaff, cmdStr);

if (dtgStaff.Rows.Count == 0)

{

btnEditStaff.Enabled = false;

btnDelStaff.Enabled = false;

}

else

{

Permissions();

}

}

}

catch

{

MessageBox.Show("No Record To Delete!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void dtgStaff\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

this.btnEditStaff\_Click(sender, e);

}

private void btnChangePwd\_Click(object sender, EventArgs e)

{

frmChangePwd frmchangePwd = new frmChangePwd(this.getUser(),this.getPwd());

frmchangePwd.ShowDialog();

}

private void btnSetting\_Click(object sender, EventArgs e)

{

frmSetting frmsetting = new frmSetting();

frmsetting.user = userName;

frmsetting.ShowDialog();

}

private void btnAddStaff\_Click(object sender, EventArgs e)

{

AccessData obj = new AccessData();

DialogResult DLR = new DialogResult();

frmAddstaff frmaddStaff = new frmAddstaff();

frmaddStaff.Text = "Add Staff";

frmaddStaff.cbosex.Items.Add("Male");

frmaddStaff.cbosex.Items.Add("Female");

frmaddStaff.cbosex.SelectedIndex = 0;

obj.fillComboBox("StaffType", "per\_name", "per\_id", frmaddStaff.cbotype);

frmaddStaff.btnOKedit.Visible = false;

DLR = frmaddStaff.ShowDialog();

if (DLR == DialogResult.OK)

{

this.btnStaff\_Click\_1(sender, e);

}

}

private void btnEditStaff\_Click(object sender, EventArgs e)

{

try

{

frmAddstaff frmaddS = new frmAddstaff();

AccessData data = new AccessData();

string SID = dtgStaff.SelectedRows[0].Cells[0].Value.ToString();

frmaddS.StaffID = SID;

frmaddS.staffName = dtgStaff.SelectedRows[0].Cells[1].Value.ToString();

string sex = null;

sex = dtgStaff.SelectedRows[0].Cells[2].Value.ToString();

frmaddS.cbosex.Items.Clear();

frmaddS.cbosex.Items.Add("Male");

frmaddS.cbosex.Items.Add("Female");

frmaddS.btnok.Visible = false;

if (sex.Trim() == "Male")

{

frmaddS.cbosex.SelectedIndex = 0;

}

if (sex.Trim() == "Female")

{

frmaddS.cbosex.SelectedIndex = 1;

}

DialogResult DLR = new System.Windows.Forms.DialogResult();

data.fillComboBox("StaffType", "per\_name", "per\_id", frmaddS.cbotype);

frmaddS.AcceptButton = frmaddS.btnOKedit;

frmaddS.Text = "Edit Staff";

frmaddS.txtname.Text = dtgStaff.SelectedRows[0].Cells[1].Value.ToString();

frmaddS.cbotype.SelectedValue = data.getDataStrID("Staff", "per\_id", "staff\_id", SID).ToString().Trim();

frmaddS.txttel.Text = dtgStaff.SelectedRows[0].Cells[5].Value.ToString();

frmaddS.txtemail.Text = dtgStaff.SelectedRows[0].Cells[6].Value.ToString();

frmaddS.txtaddress.Text = dtgStaff.SelectedRows[0].Cells[7].Value.ToString();

DLR = frmaddS.ShowDialog();

if (DLR == DialogResult.OK)

{

AccessData acc = new AccessData();

string cmdStr = "SELECT dbo.Staff.staff\_id, dbo.Staff.staff\_name, dbo.Staff.staff\_sex, dbo.StaffType.per\_id, dbo.StaffType.per\_name, dbo.Staff.staff\_tell, dbo.Staff.staff\_email, dbo.Staff.staff\_address FROM dbo.Staff LEFT JOIN dbo.StaffType ON dbo.Staff.per\_id = dbo.StaffType.per\_id WHERE dbo.Staff.staff\_name<>'admin'";

data.fillDataGridView(this.dtgStaff, cmdStr);

if (dtgStaff.Rows.Count == 0)

{

btnEditStaff.Enabled = false;

btnDelStaff.Enabled = false;

}

else

{

Permissions() }

}

}

catch

{

MessageBox.Show("No data to Edit!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void btnAddTbl\_Click(object sender, EventArgs e)

{

frmAddTable frmaddT = new frmAddTable();

DialogResult DLR = new System.Windows.Forms.DialogResult();

frmaddT.btnOKeditT.Visible = false;

frmaddT.Text = "Add Table";

DLR = frmaddT.ShowDialog();

if (DLR == DialogResult.OK)

{

//fill datagridview for Table

AccessData data = new AccessData();

string cmdStr = "SELECT tbl\_id, tbl\_name, description FROM [Table]";

data.fillDataGridView(this.dtgTable, cmdStr);

}

}

private void btnEditTbl\_Click(object sender, EventArgs e)

{

frmAddTable frmeditT = new frmAddTable();

DialogResult DLR = new System.Windows.Forms.DialogResult();

string oTID = this.dtgTable.SelectedRows[0].Cells[0].Value.ToString();

frmeditT.TID = oTID;

frmeditT.Text = "Edit Table";

frmeditT.AcceptButton = frmeditT.btnOKeditT;

frmeditT.txtTname.Text = this.dtgTable.SelectedRows[0].Cells[1].Value.ToString().Trim();

frmeditT.txtTdescription.Text = this.dtgTable.SelectedRows[0].Cells[2].Value.ToString().Trim();

DLR = frmeditT.ShowDialog();

if (DLR == DialogResult.OK)

{

//fill datagridview for Table

AccessData data = new AccessData();

string cmdStr = "SELECT tbl\_id, tbl\_name, description FROM [Table]";

data.fillDataGridView(this.dtgTable, cmdStr);

}

}

public void DisplayPicToFormLoad()

{

if (DGFood.Rows.Count > 0)

{

if (picFood.Image != null)

picFood.Image.Dispose();

AccessData Acc = new AccessData();

ServerName = Acc.serverName;

DataName = Acc.dbName;

Connection con = new Connection();

con.connectDatabase(ServerName, DataName);

SqlCommand cmded = new SqlCommand("ReadImage", con.connectDatabase(ServerName, DataName));

cmded.CommandType = CommandType.StoredProcedure;

cmded.Parameters.Add("@fid", SqlDbType.Int).Value = DGFood.Rows[0].Cells[0].Value;

SqlDataAdapter adp = new SqlDataAdapter(cmded);

DataTable dt = new DataTable();

if (con.connectDatabase(ServerName, DataName).State == ConnectionState.Closed)

con.connectDatabase(ServerName, DataName).Open();

adp.Fill(dt);

if (dt.Rows.Count > 0)

{

MemoryStream ms = new MemoryStream((byte[])dt.Rows[0]["img"]);

picFood.Image = Image.FromStream(ms);

picFood.SizeMode = PictureBoxSizeMode.StretchImage;

picFood.Refresh();

}

}

}

private void dtgTable\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

this.btnEditTbl\_Click(sender, e);

}

private void btnAddFood\_Click(object sender, EventArgs e)

{

frmFood fFood = new frmFood();

fFood.FrmEnertry = "frmAdd";

fFood.PF = "no";

fFood.ShowDialog();

if (fFood.DialogResult == DialogResult.OK)

{

btnFood\_Click(sender, e);

}

}

private void btnEditFood\_Click(object sender, EventArgs e)

{

frmFood fFood = new frmFood();

fFood.FrmEnertry = "frmEdit";

string f;

string f\_type;

f = DGFood.SelectedRows[0].Cells[0].Value.ToString();

fFood.food = f;

f\_type = DGFood.SelectedRows[0].Cells[1].Value.ToString();

fFood.F\_type = f\_type;

fFood.txtFcode.Text = DGFood.SelectedRows[0].Cells[2].Value.ToString();

fFood.txtFname.Text = DGFood.SelectedRows[0].Cells[3].Value.ToString();

fFood.txtFprice.Text = DGFood.SelectedRows[0].Cells[5].Value.ToString();

fFood.txtFdes.Text = DGFood.SelectedRows[0].Cells[6].Value.ToString();

fFood.P = "no";

fFood.Pcode = DGFood.SelectedRows[0].Cells[2].Value.ToString();

fFood.ShowDialog();

if (fFood.DialogResult == DialogResult.OK)

{

btnFood\_Click(sender, e);

}

}

private void DGFood\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (DGFood.SelectedRows.Count > 0)

{

if (picFood.Image != null)

picFood.Image.Dispose();

AccessData Acc = new AccessData();

ServerName = Acc.serverName;

DataName = Acc.dbName;

Connection conn = new Connection();

conn.connectDatabase(ServerName, DataName);

SqlCommand cmd = new SqlCommand("ReadImage", conn.connectDatabase(ServerName, DataName));

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@fid", SqlDbType.Int).Value = DGFood.SelectedRows[0].Cells[0].Value.ToString();

SqlDataAdapter adp = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

if (conn.connectDatabase(ServerName, DataName).State == ConnectionState.Closed)

conn.connectDatabase(ServerName, DataName).Open();

adp.Fill(dt);

if (dt.Rows.Count > 0)

{

MemoryStream ms = new MemoryStream((byte[])dt.Rows[0]["img"]);

picFood.Image = Image.FromStream(ms);

picFood.SizeMode = PictureBoxSizeMode.StretchImage;

picFood.Refresh();

}

}

}

private void DGFood\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

if (DGFood.SelectedRows.Count > 0)

{

btnEditFood\_Click(sender, e);

}

}

private void btnReport\_Click(object sender, EventArgs e)

{

frmReports objReport = new frmReports();

objReport.ShowDialog();

}

private void btnExit\_Click(object sender, EventArgs e)

{

frmConfirmClose frmClose = new frmConfirmClose();

if (frmClose.ShowDialog() == DialogResult.OK)

{

this.DialogResult = DialogResult.OK;

}

}

private void btnLogOff\_Click(object sender, EventArgs e)

{

frmLogIn frmlog = new frmLogIn();

frmlog.Show();

this.Dispose();

}

private void cboSerchBy\_SelectedIndexChanged(object sender, EventArgs e)

{

this.txtSerchFood.Focus();

this.txtSerchFood.SelectAll();

}

private void btnHelpOnline\_Click(object sender, EventArgs e)

{

try

{

System.Diagnostics.Process.Start("https://www.facebook.com/CheamBunhie");

}

catch { }

}

private void buttonX1\_Click(object sender, EventArgs e)

{

this.WindowState = FormWindowState.Minimized;

}

private void btnPermiss\_Click(object sender, EventArgs e)

{

this.tabStaff.Visible = false;

this.tabOrder.Visible = false;

this.tabFood.Visible = false;

this.tabTable.Visible = false;

this.tabPermiss.Visible = true;

this.btnOrder.Checked = false;

this.btnFood.Checked = false;

this.btnTable.Checked = false;

this.btnStaff.Checked = false;

this.btnPermiss.Checked = true;

//set background color

this.dgPermiss.BackgroundColor = Color.White;

//fill datagridview for customer

AccessData data = new AccessData();

string cmdStr = "SELECT per\_id,per\_name,per\_des from StaffType where per\_name<>'Administrator'";

data.fillDataGridView(this.dgPermiss, cmdStr);

//set property for Datagridview

this.dgPermiss.Columns[0].Visible = false;

this.dgPermiss.Columns[1].HeaderCell.Value = "ឈ្នោះ/Name";

this.dgPermiss.Columns[2].HeaderCell.Value = "ពិពណ៍នា/Description";

this.dgPermiss.ColumnHeadersHeightSizeMode = DataGridViewColumnHeadersHeightSizeMode.DisableResizing;

if (dgPermiss.Rows.Count == 0)

{

btnEditPermiss.Enabled = false;

btnDelPermiss.Enabled = false;

}

else

{

Permissions()

}

}

private void btnAddPermiss\_Click(object sender, EventArgs e)

{

frmAddPermissions objPermiss = new frmAddPermissions();

objPermiss.frmEntry = "frmAdd";

objPermiss.ShowDialog();

if (objPermiss.DialogResult == DialogResult.OK)

{

btnPermiss\_Click(sender, e);

}

}

private void btnEditPermiss\_Click(object sender, EventArgs e)

{

frmAddPermissions objPermiss = new frmAddPermissions();

objPermiss.frmEntry = "frmEdit";

objPermiss.pername = this.dgPermiss.SelectedRows[0].Cells[1].Value.ToString();

objPermiss.txtPerName.Text = this.dgPermiss.SelectedRows[0].Cells[1].Value.ToString();

objPermiss.txtDes.Text = this.dgPermiss.SelectedRows[0].Cells[2].Value.ToString();

objPermiss.ValID = dgPermiss.SelectedRows[0].Cells[0].Value.ToString();

objPermiss.ShowDialog();

if (objPermiss.DialogResult == DialogResult.OK)

{

btnPermiss\_Click(sender, e);

}

}

private void btnDelPermiss\_Click(object sender, EventArgs e)

{

frmDeletePermiss objDelPermiss = new frmDeletePermiss();

objDelPermiss.per\_id = this.dgPermiss.SelectedRows[0].Cells[0].Value.ToString();

objDelPermiss.ShowDialog();

if (objDelPermiss.DialogResult == DialogResult.OK)

{

btnPermiss\_Click(sender, e);

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

DateTime dateTime = DateTime.Now;

this.panelHour.Text = dateTime.ToString();

}

private void dgPermiss\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

btnEditPermiss\_Click(sender, e);

}

private void txtPermiss\_TextChanged(object sender, EventArgs e)

{

AccessData data = new AccessData();

string cmdStr = "SELECT per\_id,per\_name,per\_des from StaffType where per\_name<>'Administrator' AND per\_name LIKE N'" + this.txtPermiss.Text + "%'";

data.fillDataGridView(this.dgPermiss, cmdStr);

if (dgPermiss.Rows.Count > 0)

{

btnEditPermiss.Enabled = true;

btnDelPermiss.Enabled = true;

}

else

{

Permissions()

}

}

private void dgPermiss\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (dgPermiss.SelectedRows[0].Cells[1].Value.ToString().Equals("Administrator"))

{

btnDelPermiss.Enabled = false;

}

else

{

btnDelPermiss.Enabled = true;

}

}

private void ctOrder\_Opening(object sender, CancelEventArgs e)

{

AccessData accData = new AccessData();

try

{

if (lsvTable.SelectedItems.Count == 0)

{

ctOrder.Items.Clear();

}

if (this.lsvTable.SelectedItems[0].ImageIndex == 0)

{

ctOrder.Items.Clear();

string tbl\_id = lsvTable.FocusedItem.SubItems[1].Text.ToString();

SqlCommand cmdRecipt = new SqlCommand("select recipt\_code from Receipt where tbl\_id='" + tbl\_id + "' AND status='1' ORDER BY cont\_receipt DESC", accData.connectDatabase(accData.serverName, accData.dbName));

SqlDataReader drRecipt = cmdRecipt.ExecuteReader();

while (drRecipt.Read())

{

ctOrder.Items.Add(drRecipt["recipt\_code"].ToString());

}

drRecipt.Close();

ctOrder.Items.Add("Add New", Properties.Resources.add\_16);

ctOrder.ItemClicked += new ToolStripItemClickedEventHandler(contexMenuuu\_ItemClicked);

}

else if (this.lsvTable.SelectedItems[0].ImageIndex == 1)

{

ctOrder.Items.Clear();

string tbl\_id = lsvTable.FocusedItem.SubItems[1].Text.ToString();

SqlCommand cmdRecipt = new SqlCommand("select recipt\_code from Receipt where tbl\_id='" + tbl\_id + "' AND status='1' ORDER BY cont\_receipt DESC", accData.connectDatabase(accData.serverName, accData.dbName));

SqlDataReader drRecipt = cmdRecipt.ExecuteReader();

while (drRecipt.Read())

{

ctOrder.Items.Add(drRecipt["recipt\_code"].ToString());

}

drRecipt.Close();

ctOrder.ItemClicked += new ToolStripItemClickedEventHandler(contexMenuuu\_ItemClicked);

}

}

catch

{

}

}

private string add1ToStringNoChecking(string str)

{

return string.Format("{0}{1:d7}","LLR", Convert.ToInt16(str.Substring(str.Length - 7, 7)) + 1);

}

void contexMenuuu\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

AccessData seneyData = new AccessData();

ToolStripItem item = e.ClickedItem;

string selectOrder=item.ToString();

if (this.lsvTable.SelectedItems.Count > 0)

{

if (selectOrder.Equals("Add New"))

{

string receipCode;

string receiptLast;

string tbl\_id = lsvTable.FocusedItem.SubItems[1].Text.ToString();

string tbl\_name = lsvTable.FocusedItem.SubItems[0].Text.ToString();

//update table to close when addNew

SqlCommand cmdUpdateTable = new SqlCommand("update Receipt set cont\_receipt='1' WHERE tbl\_id='" + tbl\_id + "'", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

cmdUpdateTable.ExecuteNonQuery();

//get recipt code

SqlCommand cmdCReciptCode = new SqlCommand("select count(recipt\_code) as cBcode from Receipt", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

Int16 strCountReciptCode =Convert.ToInt16(cmdCReciptCode.ExecuteScalar().ToString());

if (strCountReciptCode == 0)

{

receipCode = "LLR0000001";

}

else

{

SqlCommand cmdLastRepCode = new SqlCommand("SELECT top 1 recipt\_code FROM Receipt ORDER BY receipt\_id DESC", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

receiptLast = cmdLastRepCode.ExecuteScalar().ToString();

receipCode = add1ToStringNoChecking(receiptLast);

}

//to open order form

var selectedItems = this.lsvTable.SelectedItems;

string[] field = { "recipt\_code", "tbl\_id", "status", "cont\_receipt" };

string[] values = { receipCode, tbl\_id, "1", "0" };

seneyData.insertData("Receipt", field, values);

//update statuse Table

string[] fieldStatus = { "status" };

string[] valuesStatus = { "1" };

seneyData.updateData("[Table]", fieldStatus, valuesStatus, "tbl\_id", tbl\_id);

if (selectedItems.Count > 0)

{

SqlCommand cmdReciptID = new SqlCommand("select MAX(receipt\_id) from Receipt", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

frmOrder objOrder = new frmOrder();

objOrder.user = userName;

objOrder.tbl\_id = tbl\_id;

objOrder.lblTableName.Text = tbl\_name;

objOrder.lblInvNo.Text = receipCode;

objOrder.reciptID = cmdReciptID.ExecuteScalar().ToString();

objOrder.frmEntry = "frmAdd";

objOrder.ShowDialog();

if (objOrder.DialogResult == DialogResult.OK)

{

frmMain\_Load(sender, e);

}

}

}

else

{

string tbl\_id = this.lsvTable.FocusedItem.SubItems[1].Text.ToString();

SqlCommand cmdReciptID = new SqlCommand("select receipt\_id from Receipt where recipt\_code='" + selectOrder + "'", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

frmOrder objOrder = new frmOrder();

objOrder.tbl\_id = tbl\_id;

objOrder.table\_id = tbl\_id;

objOrder.frmEntry = "frmEdit";

objOrder.reciptID = cmdReciptID.ExecuteScalar().ToString();

objOrder.ShowDialog();

if (objOrder.DialogResult == DialogResult.OK)

{

frmMain\_Load(sender, e);

}

}

}

}

private void btnCloseAllTbl\_Click(object sender, EventArgs e)

{

frmConfirmDelTable frmconfirm = new frmConfirmDelTable();

frmconfirm.labelX1.Text = "Do you want to close all tables ?";

frmconfirm.ShowDialog();

if (frmconfirm.DialogResult == DialogResult.OK)

{

AccessData acc = new AccessData();

SqlCommand cmdTable = new SqlCommand("update [Table] set status=0", acc.connectDatabase(acc.serverName, acc.dbName));

cmdTable.ExecuteNonQuery();

SqlCommand cmdOrder = new SqlCommand("update [Receipt] set status=0", acc.connectDatabase(acc.serverName, acc.dbName));

cmdOrder.ExecuteNonQuery();

frmMain\_Load(sender, e);

}

}

}

}

**Form Order**

namespace SecredCaffe

{

public partial class frmOrder : Form

{

SqlCommand cmdObj;

SqlDataReader readerObj;

AccessData seneyData = new AccessData();

public string tbl\_id = "";

public string table\_id = "";

public string frmEntry = "";

public string user;

public string order\_id = "";

public string inv = "no";

public string reciptID="";

public frmOrder()

{

InitializeComponent();

}

private void fillDataToFormLoad()

{

//fill object

this.lblCashier.Text = user;

//get max id of setting

cmdObj = new SqlCommand("select max(id) from Setting", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

string settingID = cmdObj.ExecuteScalar().ToString();

this.txtDescount.Text = seneyData.getData("Setting", "discount", "id", settingID);

this.dtInputOrder.Text = DateTime.Now.ToString();

}

private void lsvFood\_MouseDoubleClick(object sender, MouseEventArgs e)

{

/\*HERE USE WE CALL STORE PROCEDURE\*/

double validateDes = Convert.ToDouble(this.txtDescount.Text.ToString());

if (validateDes < 0 || validateDes > 100)

{

MessageBox.Show("Descount Must be In Rank 0 to 100.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

cmdObj = new SqlCommand("select max(id) from Setting", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

string settingID = cmdObj.ExecuteScalar().ToString();

this.txtDescount.Text = seneyData.getData("Setting", "discount", "id", settingID);

return;

}

if (txtQty.Text.Equals(""))

{

txtQty.Text = "1";

}

cmdObj = new SqlCommand("MAKE\_ORDER", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

cmdObj.CommandType = CommandType.StoredProcedure;

//ADD PARAMATERS

cmdObj.Parameters.Add(new SqlParameter("@RECEIPT\_ID", Convert.ToInt16(reciptID)));

cmdObj.Parameters.Add(new SqlParameter("@RECEIPT\_DATE", this.dtInputOrder.Text.ToString()));

cmdObj.Parameters.Add(new SqlParameter("@STAFF\_ID", Convert.ToInt32(seneyData.getDataStrID("Staff", "staff\_id", "staff\_name", this.lblCashier.Text))));

cmdObj.Parameters.Add(new SqlParameter("@TBL\_ID", Convert.ToInt32(tbl\_id)));

cmdObj.Parameters.Add(new SqlParameter("@DISCOUNT", float.Parse(this.txtDescount.Text.ToString(), CultureInfo.InvariantCulture.NumberFormat)));

cmdObj.Parameters.Add(new SqlParameter("@FOOD\_ID", Convert.ToInt32(lsvFood.FocusedItem.SubItems[1].Text.ToString())));

cmdObj.Parameters.Add(new SqlParameter("@QTY", Convert.ToInt32(this.txtQty.Text.ToString())));

cmdObj.Parameters.Add(new SqlParameter("@PRICE", seneyData.getData("Food", "food\_price", "food\_id", lsvFood.FocusedItem.SubItems[1].Text)));

cmdObj.ExecuteNonQuery();

cmdObj.Dispose();

//FILL DATAGRIDVIEW

fillDtgInvoice();

//call method

calPayment();

//set price for receive

double total = Convert.ToDouble(this.lblGrandtotalD.Text.ToString().Replace("$", ""));

this.btnExit.Text = "រក្សាចាកចេញ/SaveClose";

}

private void fillDtgInvoice()

{

string cmdStr = "select \* from viewReceiptDetail where receipt\_id=" + reciptID + "";

seneyData.fillDataGridView(this.dgtInvoice, cmdStr);

//SET PROPERTIES COLUMN HEADER

this.dgtInvoice.Columns[0].Visible = false;

this.dgtInvoice.Columns[1].Visible = false;

this.dgtInvoice.Columns[2].Visible = false;

this.dgtInvoice.Columns[3].HeaderCell.Value = "អាហារ";

this.dgtInvoice.Columns[4].HeaderCell.Value = "តម្លៃ";

this.dgtInvoice.Columns[5].HeaderCell.Value = "បរិមាណ";

this.dgtInvoice.Columns[6].HeaderCell.Value = "សរុប";

this.dgtInvoice.ColumnHeadersHeightSizeMode = DataGridViewColumnHeadersHeightSizeMode.DisableResizing;

}

private void calPayment()

{

//CALCULATE PAYMENT INFO

double sum = 0;

cmdObj = new SqlCommand("select max(id) from Setting", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

string id = cmdObj.ExecuteScalar().ToString();

double rate = Convert.ToDouble(seneyData.getData("Setting", "exchangerate", "id", id));

if (this.dgtInvoice.Rows.Count > 0)

{

for (int i = 0; i < this.dgtInvoice.Rows.Count; i++)

{

sum += Convert.ToDouble(this.dgtInvoice[6, i].Value.ToString());

}

}

this.lblTotal.Text = sum.ToString();

this.lblSubTotal.Text = Convert.ToDouble(sum / rate).ToString();

double dis = Convert.ToDouble(this.txtDescount.Text);

this.lblGrandtotalD.Text = (sum \* (1 - dis / 100) / rate).ToString();

this.lblGtotalRoum.Text = RoundVal(sum \* (1 - dis / 100)).ToString();

}

//round method

private static double RoundVal(double values)

{

int res = (int)Math.Round(values / 100);

return (int)res \* 100;

}

private void food\_TypeDataGridView\_CellClick(object sender, DataGridViewCellEventArgs e)

{

// clear old items

this.lsvFood.Items.Clear();

//clear images

this.imageList1.Images.Clear();

string type\_id = "";

if (this.food\_TypeDataGridView.SelectedRows.Count > 0)

{

type\_id = this.food\_TypeDataGridView.SelectedCells[0].Value.ToString();

lblFoodType.Text = "ប្រភេទ : " + this.food\_TypeDataGridView.SelectedCells[1].Value.ToString();

}

// TODO: This line of code loads data into the 'mySecredCaffeDataSet.Food\_Type' table. You can move, or remove it, as needed.

// this.food\_TypeTableAdapter.Fill(this.mySecredCaffeDataSet.Food\_Type);

/\*

\*TO GET IMAGE FROM DATABASE

\*AND FILL INTO IMAGE LIST

\*/

cmdObj = new SqlCommand("select (CONVERT(nvarchar(50),food\_name) + ': ' + CONVERT(nvarchar(50), food\_price)+ 'R') as food\_info, food\_type\_id, food\_id, img from Food where food\_type\_id=" + type\_id + "", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

readerObj = cmdObj.ExecuteReader();

int i = 0;

//ListViewItem listViewItem = new ListViewItem();

while (readerObj.Read())

{

// load the bytes from the database that represent your image

var imageBytes = (byte[])readerObj["img"];

// put those bytes into a memory stream and "img" the memory stream

MemoryStream memStm = new MemoryStream(imageBytes);

memStm.Seek(0, SeekOrigin.Begin);

// create an "Image" from that memory stream

Image image = Image.FromStream(memStm);

// add image to list

imageList1.Images.Add(image);

//add image to listview

//listView1.Items.Add(readerObj["food\_name"].ToString()).ImageIndex =i;

ListViewItem listViewItem = new ListViewItem(readerObj["food\_info"].ToString(), i);

listViewItem.SubItems.Add(readerObj["food\_id"].ToString());

lsvFood.Items.Add(listViewItem);

i++;

}

}

private void dgtInvoice\_CellClick(object sender, DataGridViewCellEventArgs e)

{

this.txtQty.Text = this.dgtInvoice.SelectedRows[0].Cells[5].Value.ToString();

}

private void btnQtyEdit\_Click(object sender, EventArgs e)

{

if (this.dgtInvoice.RowCount > 0)

{

if (this.dgtInvoice.SelectedRows.Count > 0)

{

//UPDATE QTY

string order\_detail\_id = this.dgtInvoice.SelectedRows[0].Cells[0].Value.ToString();

string[] field = { "quantity" };

string[] value = { this.txtQty.Text };

seneyData.updateData("ReceiptDetail", field, value, "receipt\_detail\_id", order\_detail\_id);

//FILL DATAGRIDVIEW

fillDtgInvoice();

//call method

calPayment();

double total = Convert.ToDouble(this.lblGrandtotalD.Text.ToString().Replace("$", ""));

}

}

}

private void btnDel\_Click(object sender, EventArgs e)

{

if (this.dgtInvoice.RowCount > 0)

{

if (this.dgtInvoice.SelectedRows.Count > 0)

{

frmConfirmDelTable frmconfirm = new frmConfirmDelTable();

frmconfirm.labelX1.Text = "Do you want to delete this food ?";

frmconfirm.ShowDialog();

if (frmconfirm.DialogResult == DialogResult.OK)

{

//delete

string order\_detail\_id = this.dgtInvoice.SelectedRows[0].Cells[0].Value.ToString();

seneyData.deleteRecord("ReceiptDetail", "receipt\_detail\_id", order\_detail\_id);

//FILL DATAGRIDVIEW

fillDtgInvoice();

//call method

calPayment();

double total = Convert.ToDouble(this.lblGrandtotalD.Text.ToString().Replace("$", ""));

}

}

}

}

private void txtSerch\_TextChanged(object sender, EventArgs e)

{

string sql = "";

string ftype = "";

if (this.txtSerch.Text.Equals(""))

{

ftype = this.food\_TypeDataGridView.Rows[0].Cells[0].Value.ToString();

sql = "select (CONVERT(nvarchar(50),food\_name) + ': $' + CONVERT(nvarchar(50), food\_price)) as food\_info, food\_type\_id, food\_id, img from Food where food\_type\_id LIKE N'" + ftype + "%'";

// TODO: This line of code loads data into the 'mySecredCaffeDataSet.Food\_Type' table. You can move, or remove it, as needed.

this.food\_TypeTableAdapter.Fill(this.mySecredCaffeDataSet.Food\_Type);

}

else

{

sql = "select (CONVERT(nvarchar(50),food\_name) + ': $' + CONVERT(nvarchar(50), food\_price)) as food\_info, food\_type\_id, food\_id, img from Food where food\_name LIKE N'" + txtSerch.Text + "%'";

}

this.lsvFood.Items.Clear();

//clear images

this.imageList1.Images.Clear();

cmdObj = new SqlCommand(sql, seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

readerObj = cmdObj.ExecuteReader();

int i = 0;

//ListViewItem listViewItem = new ListViewItem();

while (readerObj.Read())

{

// load the bytes from the database that represent your image

var imageBytes = (byte[])readerObj["img"];

// put those bytes into a memory stream and "img" the memory stream

MemoryStream memStm = new MemoryStream(imageBytes);

memStm.Seek(0, SeekOrigin.Begin);

// create an "Image" from that memory stream

Image image = Image.FromStream(memStm);

// add image to imagelist

imageList1.Images.Add(image);

//add image to listview

//listView1.Items.Add(readerObj["food\_name"].ToString()).ImageIndex =i;

ListViewItem listViewItem = new ListViewItem(readerObj["food\_info"].ToString(), i);

listViewItem.SubItems.Add(readerObj["food\_id"].ToString());

lsvFood.Items.Add(listViewItem);

i++;

}

string fid = seneyData.getDataStrIDByLike("Food", "food\_type\_id", "food\_name", txtSerch.Text.ToString());

if (fid.Equals(""))

{

this.lblFoodType.Text = "អត់មានឈ្មោះអារហារនេះទេ !";

}

else

{

this.lblFoodType.Text = "ប្រភេទ : " + seneyData.getDataStrID("Food\_Type", "food\_type\_name", "food\_type\_id", fid);

}

}

private void btnPrintInv\_Click(object sender, EventArgs e)

{

if(this.dgtInvoice.Rows.Count==0)

{

MessageBox.Show(this, "អ្នកមិនអាចចេញវិក័យប័ត្របានទេពីព្រោះតុនេះអត់ទាន់បានកម៉្មង់អាហារ\nYou cannot invoice because this table is not order food", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

return;

}

try

{

if (frmEntry.Equals("frmEdit") || frmEntry.Equals("frmAdd"))

{

frmConfirmDelTable frmconfirm = new frmConfirmDelTable();

frmconfirm.labelX1.Text = "Do you want to recipt invoice ?";

frmconfirm.ShowDialog();

if (frmconfirm.DialogResult == DialogResult.OK)

{

SqlCommand cmdCheckReciptCont = new SqlCommand("update Receipt set discount='" + float.Parse(this.txtDescount.Text.ToString(), CultureInfo.InvariantCulture.NumberFormat) + "', cont\_receipt='1' where receipt\_id='" + reciptID + "'", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

cmdCheckReciptCont.ExecuteNonQuery();

//update table to open

SqlCommand cmdUpdateTbl = new SqlCommand("update [Table] set status='0' where tbl\_id='" + tbl\_id + "'", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

cmdUpdateTbl.ExecuteNonQuery();

seneyData.PrintReportWithParameter(reciptID);

this.btnPrintInv.Visible = false;

this.btnContinue.Visible = true;

this.lsvFood.Enabled = false;

this.btnDel.Enabled = false;

this.btnQtyEdit.Enabled = false;

inv = "yes";

btnExit\_Click\_1(sender, e);

}

}

}

catch

{

MessageBox.Show(this, "អ្នកមិនអាចចេញវិក័យប័ត្របានទេពីព្រោះតុនេះអត់ទាន់បានកម៉្មង់អាហារ\nYou cannot invoice because this table is not order food", "Warnning", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

}

private void btnExit\_Click\_1(object sender, EventArgs e)

{

if (inv.Equals("no"))

{

AccessData acc = new AccessData();

SqlCommand cmd = new SqlCommand("update Receipt set receipt\_date='" + this.dtInputOrder.Text.ToString() + "', staff\_id=" + Convert.ToInt32(seneyData.getDataStrID("Staff", "staff\_id", "staff\_name", this.lblCashier.Text)) + ", discount='"+float.Parse(this.txtDescount.Text.ToString(), CultureInfo.InvariantCulture.NumberFormat)+"' where receipt\_id='" + reciptID + "'", acc.connectDatabase(acc.serverName, acc.dbName));

cmd.ExecuteNonQuery();

this.DialogResult = DialogResult.OK;

}

this.DialogResult = DialogResult.OK;

}

private void btnChangeTbl\_Click(object sender, EventArgs e)

{

frmChangeTable objMergeTable = new frmChangeTable();

objMergeTable.table\_id = tbl\_id; ;

objMergeTable.order\_id = reciptID;

objMergeTable.frmEntryCgTable = "OrderCgTable";

objMergeTable.ShowDialog();

if (objMergeTable.DialogResult == DialogResult.OK)

{

frmChangeTable objCgTable = new frmChangeTable();

this.lblTableName.Text = objMergeTable.cboMergeTable.Text;

this.tbl\_id = objMergeTable.cboMergeTable.SelectedValue.ToString();

}

}

private void conTextOrder\_Opening(object sender, CancelEventArgs e)

{

if (this.lsvFood.SelectedItems.Count == 0)

{

this.conTextAddFood.Enabled = false;

this.addFoodFreeToolStripMenuItem.Enabled = false;

}

else

{

this.conTextAddFood.Enabled = true;

this.addFoodFreeToolStripMenuItem.Enabled = true;

}

}

private void conTextAddFood\_Click(object sender, EventArgs e)

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*HERE USE WE CALL STORE PROCEDURE\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

double validateDes = Convert.ToDouble(this.txtDescount.Text.ToString());

if (validateDes < 0 || validateDes > 100)

{

MessageBox.Show("Descount Must be In Rank 0 to 100.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

cmdObj = new SqlCommand("select max(id) from Setting", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

string settingID = cmdObj.ExecuteScalar().ToString();

this.txtDescount.Text = seneyData.getData("Setting", "discount", "id", settingID);

return;

}

if (txtQty.Text.Equals(""))

{

txtQty.Text = "1";

}

cmdObj = new SqlCommand("MAKE\_ORDER", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

cmdObj.CommandType = CommandType.StoredProcedure;

//ADD PARAMATERS

cmdObj.Parameters.Add(new SqlParameter("@RECEIPT\_ID", Convert.ToInt32(reciptID)));

cmdObj.Parameters.Add(new SqlParameter("@RECEIPT\_DATE", this.dtInputOrder.Text.ToString()));

cmdObj.Parameters.Add(new SqlParameter("@STAFF\_ID", Convert.ToInt32(seneyData.getDataStrID("Staff", "staff\_id", "staff\_name", this.lblCashier.Text))));

cmdObj.Parameters.Add(new SqlParameter("@TBL\_ID", Convert.ToInt32(tbl\_id)));

cmdObj.Parameters.Add(new SqlParameter("@DISCOUNT", float.Parse(this.txtDescount.Text.ToString(), CultureInfo.InvariantCulture.NumberFormat)));

cmdObj.Parameters.Add(new SqlParameter("@FOOD\_ID", Convert.ToInt32(lsvFood.FocusedItem.SubItems[1].Text.ToString())));

cmdObj.Parameters.Add(new SqlParameter("@QTY", Convert.ToInt32(this.txtQty.Text.ToString())));

cmdObj.Parameters.Add(new SqlParameter("@PRICE", seneyData.getData("Food", "food\_price", "food\_id", lsvFood.FocusedItem.SubItems[1].Text)));

cmdObj.ExecuteNonQuery();

cmdObj.Dispose();

//FILL DATAGRIDVIEW

fillDtgInvoice();

//call method

calPayment();

//set price for receive

double total = Convert.ToDouble(this.lblGrandtotalD.Text.ToString().Replace("$", ""));

this.btnExit.Text = "រក្សាចាកចេញ/SaveClose";

}

private void btnContinue\_Click(object sender, EventArgs e)

{

if (this.btnContinue.Tag.Equals("1"))

{

SqlCommand cmdCheckReciptCont = new SqlCommand("update Receipt set cont\_receipt='0' where receipt\_id='" + reciptID + "'", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

cmdCheckReciptCont.ExecuteNonQuery();

// update table to close

SqlCommand cmdUpdateTbl = new SqlCommand("update [Table] set status='1' where tbl\_id='" + tbl\_id + "'", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

cmdUpdateTbl.ExecuteNonQuery();

this.btnPrintInv.Visible = true;

this.lsvFood.Enabled = true;

this.btnDel.Enabled = true;

this.btnQtyEdit.Enabled = true;

this.txtDescount.Enabled = true;

this.btnContinue.Text = "បិទតុ/Close table";

this.btnContinue.Tag = "2";

}

else if (this.btnContinue.Tag.Equals("2"))

{

SqlCommand cmdCheckReciptCont = new SqlCommand("update Receipt set cont\_receipt='1' where receipt\_id='" + reciptID + "'", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

cmdCheckReciptCont.ExecuteNonQuery();

// update table to close

SqlCommand cmdUpdateTbl = new SqlCommand("update [Table] set status='0' where tbl\_id='" + tbl\_id + "'", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

cmdUpdateTbl.ExecuteNonQuery();

this.btnPrintInv.Visible = false;

this.lsvFood.Enabled = false;

this.btnDel.Enabled = false;

this.btnQtyEdit.Enabled = false;

this.btnContinue.Text = "បន្តទៀត/Continue";

this.btnContinue.Tag = "1";

}

}

private void addFoodFreeToolStripMenuItem\_Click(object sender, EventArgs e)

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*HERE USE WE CALL STORE PROCEDURE\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if (txtQty.Text.Equals(""))

{

txtQty.Text = "1";

}

cmdObj = new SqlCommand("makeOrderFree", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

cmdObj.CommandType = CommandType.StoredProcedure;

//ADD PARAMATERS

cmdObj.Parameters.Add(new SqlParameter("@RECEIPT\_ID", Convert.ToInt32(reciptID)));

cmdObj.Parameters.Add(new SqlParameter("@FOOD\_ID", Convert.ToInt32(lsvFood.FocusedItem.SubItems[1].Text.ToString())));

cmdObj.Parameters.Add(new SqlParameter("@QTY", Convert.ToInt32(this.txtQty.Text.ToString())));

cmdObj.ExecuteNonQuery();

cmdObj.Dispose();

//FILL DATAGRIDVIEW

fillDtgInvoice();

//call method

calPayment();

//set price for receive

double total = Convert.ToDouble(this.lblGrandtotalD.Text.ToString().Replace("$", ""));

this.btnExit.Text = "រក្សាចាកចេញ/SaveClose";

}

private void frmOrder\_Load(object sender, EventArgs e)

{

try

{

// TODO: This line of code loads data into the 'mySecredCaffeDataSet.Food\_Type' table. You can move, or remove it, as needed.

this.food\_TypeTableAdapter.Fill(this.mySecredCaffeDataSet.Food\_Type);

// fill data to form load

fillDataToFormLoad();

if (frmEntry.Equals("frmEdit"))

{

//fill discount to formload with edit

SqlCommand cmdDisc = new SqlCommand("select discount from Receipt where receipt\_id ='" + reciptID + "'", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

this.txtDescount.Text = cmdDisc.ExecuteScalar().ToString();

//checkReciptCont();

SqlCommand cmdCheckReciptCont = new SqlCommand("select cont\_receipt from Receipt where receipt\_id='" + reciptID + "'", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

string strCheckReciptCont = cmdCheckReciptCont.ExecuteScalar().ToString();

if (strCheckReciptCont.Equals("0"))

{

this.btnContinue.Visible = true;

this.btnContinue.Tag = "2";

this.btnContinue.Text = "បិទតុ/Close Table";

this.btnPrintInv.Visible = true;

this.lsvFood.Enabled = true;

this.btnDel.Enabled = true;

this.btnQtyEdit.Enabled = true;

this.txtDescount.Enabled = true;

}

else

{

this.btnContinue.Visible = true;

this.btnContinue.Tag = "1";

this.btnContinue.Text = "បន្តទៀត/Continue";

this.btnPrintInv.Visible = false;

this.lsvFood.Enabled = false;

this.btnDel.Enabled = false;

this.btnQtyEdit.Enabled = false;

this.txtDescount.Enabled = false;

}

this.lblInvNo.Text = seneyData.getData("Receipt", "recipt\_code", "receipt\_id",reciptID);

//retrive data

this.dtInputOrder.Text = seneyData.getData("Receipt", "receipt\_date", "receipt\_id", reciptID);

this.lblCashier.Text = seneyData.getData("Staff", "staff\_name", "staff\_id", seneyData.getData("Receipt", "staff\_id", "receipt\_id", reciptID));

this.lblTableName.Text = seneyData.getData("[Table]", "tbl\_name", "tbl\_id", seneyData.getData("Receipt", "tbl\_id", "receipt\_id", reciptID));

//fill data to dg for edit

string cmdStr = "select \* from viewReceiptDetail where receipt\_id=" + reciptID + "";

seneyData.fillDataGridView(this.dgtInvoice, cmdStr);

//SET PROPERTIES COLUMN HEADER

this.dgtInvoice.Columns[0].Visible = false;

this.dgtInvoice.Columns[1].Visible = false;

this.dgtInvoice.Columns[2].Visible = false;

this.dgtInvoice.Columns[3].HeaderCell.Value = "អាហារ";

this.dgtInvoice.Columns[4].HeaderCell.Value = "តម្លៃ";

this.dgtInvoice.Columns[5].HeaderCell.Value = "បរិមាណ";

this.dgtInvoice.Columns[6].HeaderCell.Value = "សរុប";

this.dgtInvoice.ColumnHeadersHeightSizeMode = DataGridViewColumnHeadersHeightSizeMode.DisableResizing;

//call method

calPayment();

double total = Convert.ToDouble(this.lblGrandtotalD.Text.ToString().Replace("$", ""));

//set button to visable

this.btnChangeTbl.Visible = true;

}

// TODO: This line of code loads data into the 'mySecredCaffeDataSet.Food\_Type' table. You can move, or remove it, as needed.

this.food\_TypeTableAdapter.Fill(this.mySecredCaffeDataSet.Food\_Type);

/\*

\*TO GET IMAGE FROM DATABASE

\*AND FILL INTO IMAGE LIST

\*/

//if (txtReceipt.Text.Equals(""))

this.dgtInvoice.BackgroundColor = Color.White;

string f\_type\_id = "";

if (food\_TypeDataGridView.Rows.Count > 0)

{

f\_type\_id = this.food\_TypeDataGridView.Rows[0].Cells[0].Value.ToString();

this.lblFoodType.Text = "ប្រភេទ : " + this.food\_TypeDataGridView.Rows[0].Cells[1].Value.ToString();

}

cmdObj = new SqlCommand("select (CONVERT(nvarchar(50),food\_name) + ': ' + CONVERT(nvarchar(50), food\_price) + ' R') as food\_info, food\_type\_id, food\_id, img from Food where food\_type\_id = '" + f\_type\_id + "'", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

readerObj = cmdObj.ExecuteReader();

int i = 0;

//ListViewItem listViewItem = new ListViewItem();

while (readerObj.Read())

{

// load the bytes from the database that represent your image

var imageBytes = (byte[])readerObj["img"];

// put those bytes into a memory stream and "img" the memory stream

MemoryStream memStm = new MemoryStream(imageBytes);

memStm.Seek(0, SeekOrigin.Begin);

// create an "Image" from that memory stream

Image image = Image.FromStream(memStm);

// add image to imagelist

imageList1.Images.Add(image);

//add image to listview

//listView1.Items.Add(readerObj["food\_name"].ToString()).ImageIndex =i;

ListViewItem listViewItem = new ListViewItem(readerObj["food\_info"].ToString(), i);

listViewItem.SubItems.Add(readerObj["food\_id"].ToString());

lsvFood.Items.Add(listViewItem);

i++;

}

this.dgtInvoice.Columns[0].Width = 1000;

}

catch (Exception ex)

{

//MessageBox.Show(ex.Message);

}

}

private void btnReset\_Click(object sender, EventArgs e)

{

frmConfirmDelTable frmconfirm = new frmConfirmDelTable();

frmconfirm.labelX1.Text = "Do you want to reset foods ?";

frmconfirm.ShowDialog();

if (frmconfirm.DialogResult == DialogResult.OK)

{

if (dgtInvoice.Rows.Count > 0)

{

SqlCommand cmdResetFoods = new SqlCommand("delete ReceiptDetail WHERE receipt\_id='" + reciptID + "'", seneyData.connectDatabase(seneyData.serverName, seneyData.dbName));

cmdResetFoods.ExecuteNonQuery();

//FILL DATAGRIDVIEW

fillDtgInvoice();

//call method

calPayment();

}

}

}

private void txtQty\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(Char.IsDigit(e.KeyChar) || (e.KeyChar == (char)Keys.Back)))

{

e.Handled = true;

System.Media.SystemSounds.Exclamation.Play();

}

}

private void txtDescount\_KeyPress(object sender, KeyPressEventArgs e)

{

if (this.txtDescount.Text.Contains("."))

{

if (!(Char.IsDigit(e.KeyChar) || (e.KeyChar == (char)Keys.Back)))

{

e.Handled = true;

System.Media.SystemSounds.Exclamation.Play();

}

}

else

{

if (!(Char.IsDigit(e.KeyChar) || (e.KeyChar == (char)Keys.Back || (e.KeyChar == (char)46))))

{

e.Handled = true;

System.Media.SystemSounds.Exclamation.Play();

}

}

}

private void txtDescount\_TextChanged(object sender, EventArgs e)

{

if (this.txtDescount.Text.Equals(""))

{

this.txtDescount.Text = "0";

}

calPayment();

}

}

}

**Form Add Food**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO;

using System.Data.SqlClient;

namespace SecredCaffe

{

public partial class frmFood : Form

{

public frmFood()

{

InitializeComponent();

}

public string FrmEnertry;

public string food;

public string P;

public string PF;

public string Pcode;

public string F\_type;

public string ServerName;

public string DataName;

public string imagename;

private void frmFood\_Load(object sender, EventArgs e)

{

lblFid.Text = food;

AccessData Acc = new AccessData();

ServerName = Acc.serverName;

DataName = Acc.dbName;

Acc.BindDataToCbo(cboFood, "Food\_Type", "food\_type\_name", "food\_type\_id");

if (FrmEnertry.Equals("frmAdd"))

{

this.Text = "Add Food";

btnEdit.Visible = false;

btnOK.Visible = true;

this.AcceptButton = btnOK;

imagename = Application.StartupPath+"\\Photo\\FoodEmpty.png";

picFood.Load(Application.StartupPath + "\\Photo\\FoodEmpty.png");

}

else if (FrmEnertry.Equals("frmEdit"))

{

this.Text ="Edit Food";

btnOK.Visible = false;

btnEdit.Visible = true;

this.AcceptButton = btnEdit;

cboFood.SelectedValue = F\_type;

if (picFood.Image != null)

picFood.Image.Dispose();

Connection conn = new Connection();

conn.connectDatabase(ServerName, DataName);

SqlCommand cmd = new SqlCommand("ReadImage", conn.connectDatabase(ServerName, DataName));

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@fid", SqlDbType.Int).Value = food;

SqlDataAdapter adp = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

try

{

if (conn.connectDatabase(ServerName, DataName).State == ConnectionState.Closed)

conn.connectDatabase(ServerName, DataName).Open();

adp.Fill(dt);

if (dt.Rows.Count > 0)

{

MemoryStream ms = new MemoryStream((byte[])dt.Rows[0]["img"]);

picFood.Image = Image.FromStream(ms);

picFood.SizeMode = PictureBoxSizeMode.StretchImage;

picFood.Refresh();

}

}

catch

{

}

finally

{

if (conn.connectDatabase(ServerName, DataName).State == ConnectionState.Open)

conn.connectDatabase(ServerName, DataName).Close();

}

}

}

private void txtFprice\_KeyPress(object sender, KeyPressEventArgs e)

{

if (this.txtFprice.Text.Contains("."))

{

if (!(Char.IsDigit(e.KeyChar) || (e.KeyChar == (char)Keys.Back)))

{

e.Handled = true;

System.Media.SystemSounds.Exclamation.Play();

}

}

else

{

if (!(Char.IsDigit(e.KeyChar) || (e.KeyChar == (char)Keys.Back || (e.KeyChar == (char)46))))

{

e.Handled = true;

System.Media.SystemSounds.Exclamation.Play();

}

}

}

private void buttonX1\_Click\_1(object sender, EventArgs e)

{

}

private void txtFprice\_Enter(object sender, EventArgs e)

{

foreach (InputLanguage Lng in InputLanguage.InstalledInputLanguages)

{

if (Lng.Culture.DisplayName.ToString().ToUpper().StartsWith("ENG"))

{

InputLanguage.CurrentInputLanguage = Lng;

}

}

if (txtFprice.Focused)

{

foreach (InputLanguage Lng in InputLanguage.InstalledInputLanguages)

{

if (Lng.Culture.DisplayName.ToString().ToUpper().StartsWith("ENG"))

{

InputLanguage.CurrentInputLanguage = Lng;

}

}

}

}

private void btnBrows\_Click(object sender, EventArgs e)

{

try

{

FileDialog fldlg = new OpenFileDialog();

fldlg.Filter = "Image File (\*.jpg;\*.bmp;\*.gif)|\*.jpg;\*.bmp;\*.gif";

if (fldlg.ShowDialog() == DialogResult.OK)

{

imagename = fldlg.FileName;

Bitmap newimg = new Bitmap(imagename);

picFood.SizeMode = PictureBoxSizeMode.StretchImage;

picFood.Image = (Image)newimg;

P = "yes";

PF = "yes";

}

fldlg = null;

}

catch (System.ArgumentException ae)

{

imagename = " ";

MessageBox.Show(ae.Message.ToString());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

private void btnRemove\_Click(object sender, EventArgs e)

{

imagename = Application.StartupPath + "\\Photo\\FoodEmpty.png";

picFood.Load(Application.StartupPath + "\\Photo\\FoodEmpty.png");

PF = "yes";

P = "yes";

}

private void btnOK\_Click(object sender, EventArgs e)

{

string Fcode = txtFcode.Text.Trim();

string Fname = txtFname.Text.Trim();

string Fdes = txtFdes.Text.Trim();

string Fid = lblFid.Text.Trim();

string tString = txtFprice.Text.Trim();

string ValCodeAdd;

string q;

AccessData Acc = new AccessData();

ServerName = Acc.serverName;

DataName = Acc.dbName;

Connection cn = new Connection();

cn.connectDatabase(ServerName, DataName);

q = "select count(\*) as Cfood from Food where food\_code='" + Fcode + "'";

SqlCommand cmded = new SqlCommand(q, cn.connectDatabase(ServerName, DataName));

ValCodeAdd = cmded.ExecuteScalar().ToString();

//int price = 0;

if (txtFcode.Text.Equals(""))

{

MessageBox.Show(this, "Please Enter Code !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtFcode.Focus();

return;

}

else if (!ValCodeAdd.Equals("0"))

{

MessageBox.Show(this, "Code is existing !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtFcode.Focus();

txtFcode.Clear();

return;

}

else if (cboFood.Items.Count == 0)

{

MessageBox.Show(this, "No Food Type To Select !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

buttonX1\_Click\_1(sender, e);

frmFood\_Type ftype = new frmFood\_Type();

if (ftype.DialogResult == DialogResult.OK)

{

frmFood\_Load(sender, e);

}

return;

}

else if (txtFname.Text.Equals(""))

{

MessageBox.Show(this, "Please Enter Name !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtFname.Focus();

return;

}

else if (txtFprice.Text.Equals(""))

{

MessageBox.Show(this, "Please Enter Price !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtFprice.Focus();

return;

}

Fcode = Fcode.Replace("'", "''");

Fname = Fname.Replace("'", "''");

Fdes = Fdes.Replace("'", "''");

try

{

FileStream fs;

fs = new FileStream(@imagename, FileMode.Open, FileAccess.Read);

//a byte array to read the image

byte[] picbyte = new byte[fs.Length];

fs.Read(picbyte, 0, System.Convert.ToInt32(fs.Length));

fs.Close();

string Qry;

Connection con = new Connection();

con.connectDatabase(ServerName, DataName);

Qry = "insert into Food(food\_type\_id,food\_code,food\_name,food\_price,description,img) values(" + cboFood.SelectedValue + ", N'" + Fcode + "', N'" + Fname + "', " + txtFprice.Text + ", N'" + Fdes + "', " + " @pic)";

SqlParameter picparameter = new SqlParameter();

picparameter.SqlDbType = SqlDbType.Image;

picparameter.ParameterName = "pic";

picparameter.Value = picbyte;

SqlCommand Cmd = new SqlCommand(Qry, con.connectDatabase(ServerName, DataName));

Cmd.Parameters.Add(picparameter);

Cmd.ExecuteNonQuery();

this.DialogResult = DialogResult.OK;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void btnEdit\_Click(object sender, EventArgs e)

{

string Fcode = txtFcode.Text;

string Fname = txtFname.Text;

string Fdes = txtFdes.Text;

string Fid = lblFid.Text;

string tString = txtFprice.Text;

string ValCodeAdd;

//int price = 0;

string q;

AccessData Acc = new AccessData();

ServerName = Acc.serverName;

DataName = Acc.dbName;

Connection cnn = new Connection();

cnn.connectDatabase(ServerName, DataName);

q = "select count(\*) as Cfood from Food where food\_code='" + Fcode + "'";

SqlCommand cmde = new SqlCommand(q, cnn.connectDatabase(ServerName, DataName));

ValCodeAdd = cmde.ExecuteScalar().ToString();

for (int i = 0; i < tString.Length; i++)

if (txtFcode.Text.Equals(""))

{

MessageBox.Show(this, "Please Enter Code !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtFcode.Focus();

return;

}

else if (txtFcode.Text != Pcode)

{

if (!ValCodeAdd.Equals("0"))

{

MessageBox.Show(this, "Code is existing !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtFcode.Focus();

txtFcode.Clear();

return;

}

}

else if (txtFname.Text.Equals(""))

{

MessageBox.Show(this, "Please Enter Name !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtFname.Focus();

return;

}

else if (txtFprice.Text.Equals(""))

{

MessageBox.Show(this, "Please Enter Price !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtFprice.Focus();

return;

}

Fcode = Fcode.Replace("'", "''");

Fname = Fname.Replace("'", "''");

Fdes = Fdes.Replace("'", "''");

if (cboFood.Items.Count == 0)

{

try

{

if (P.Equals("yes"))

{

FileStream fs;

fs = new FileStream(@imagename, FileMode.Open, FileAccess.Read);

//a byte array to read the image

byte[] picbyte = new byte[fs.Length];

fs.Read(picbyte, 0, System.Convert.ToInt32(fs.Length));

fs.Close();

string query;

Connection conn = new Connection();

conn.connectDatabase(ServerName, DataName);

query = "update Food set food\_code=N'" + Fcode + "', food\_name=N'" + Fname + "', food\_price=" + txtFprice.Text + ", description=N'" + Fdes + "', img=" + " @pic where food\_id=" + lblFid.Text + "";

SqlParameter picparameter = new SqlParameter();

picparameter.SqlDbType = SqlDbType.Image;

picparameter.ParameterName = "pic";

picparameter.Value = picbyte;

SqlCommand cmd = new SqlCommand(query, conn.connectDatabase(ServerName, DataName));

cmd.Parameters.Add(picparameter);

cmd.ExecuteNonQuery();

this.DialogResult = DialogResult.OK;

this.Close();

}

else

{

string query;

Connection conn = new Connection();

conn.connectDatabase(ServerName, DataName);

query = "update Food set food\_code=N'" + Fcode + "', food\_name=N'" + Fname + "', food\_price='" + txtFprice.Text + "', description=N'" + Fdes + "' where food\_id=" + lblFid.Text + "";

SqlCommand cmd = new SqlCommand(query, conn.connectDatabase(ServerName, DataName));

cmd.ExecuteNonQuery();

this.DialogResult = DialogResult.OK;

this.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

else

{

try

{

if (P.Equals("yes"))

{

FileStream fs;

fs = new FileStream(@imagename, FileMode.Open, FileAccess.Read);

//a byte array to read the image

byte[] picbyte = new byte[fs.Length];

fs.Read(picbyte, 0, System.Convert.ToInt32(fs.Length));

fs.Close();

string query;

Connection conn = new Connection();

conn.connectDatabase(ServerName, DataName);

query = "update Food set food\_type\_id=" + cboFood.SelectedValue + ", food\_code=N'" + Fcode + "', food\_name=N'" + Fname + "', food\_price=" + txtFprice.Text + ", description=N'" + Fdes + "', img=" + " @pic where food\_id=" + lblFid.Text + "";

SqlParameter picparameter = new SqlParameter();

picparameter.SqlDbType = SqlDbType.Image;

picparameter.ParameterName = "pic";

picparameter.Value = picbyte;

SqlCommand cmd = new SqlCommand(query, conn.connectDatabase(ServerName, DataName));

cmd.Parameters.Add(picparameter);

cmd.ExecuteNonQuery();

this.DialogResult = DialogResult.OK;

this.Close();

}

else

{

string query;

Connection conn = new Connection();

conn.connectDatabase(ServerName, DataName);

query = "update Food set food\_type\_id=" + cboFood.SelectedValue + ",food\_code=N'" + Fcode + "', food\_name=N'" + Fname + "', food\_price='" + txtFprice.Text + "', description=N'" + Fdes + "' where food\_id=" + lblFid.Text + "";

SqlCommand cmd = new SqlCommand(query, conn.connectDatabase(ServerName, DataName));

cmd.ExecuteNonQuery();

this.DialogResult = DialogResult.OK;

this.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

private void buttonX1\_Click(object sender, EventArgs e)

{

frmFood\_Type ftype = new frmFood\_Type();

ftype.ShowDialog();

if (ftype.DialogResult == DialogResult.Yes)

{

AccessData acc = new AccessData();

acc.BindDataToCbo(cboFood, "Food\_Type", "food\_type\_name", "food\_type\_id");

}

}

private void btnCancel\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

**Form Food Type**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace SecredCaffe

{

public partial class frmFood\_Type : Form

{

public frmFood\_Type()

{

InitializeComponent();

}

private void DGpropertye()

{

DGFoodType.Columns[0].Visible = false;

DGFoodType.BackgroundColor = Color.White;

DGFoodType.Columns[1].HeaderText = "ប្រភេទអាហារ/Food Type";

DGFoodType.Columns[2].HeaderText = "អធិប្បាយខ្លីៗ/Description";

}

private void frmFood\_Type\_Load(object sender, EventArgs e)

{

string SqlStr = "select \* from Food\_Type";

AccessData BindToDG = new AccessData();

BindToDG.fillDataGridView(DGFoodType,SqlStr);

DGpropertye();

if (DGFoodType.Rows.Count == 0)

{

btnEdit.Enabled = false;

btnDelete.Enabled = false;

}

else

{

Permissions()

}

}

private void frmFood\_Type\_FormClosing(object sender, FormClosingEventArgs e)

{

this.DialogResult = DialogResult.Yes;

}

private void DGFoodType\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

frmAddFoodType fFood\_type = new frmAddFoodType();

fFood\_type.FrmEnertry = "frmEdit";

fFood\_type.FtypeID = DGFoodType.SelectedRows[0].Cells[0].Value.ToString();

fFood\_type.txtFtype.Text = DGFoodType.SelectedRows[0].Cells[1].Value.ToString();

fFood\_type.txtFDes.Text = DGFoodType.SelectedRows[0].Cells[2].Value.ToString();

fFood\_type.ShowDialog();

if (fFood\_type.DialogResult == DialogResult.OK)

{

frmFood\_Type\_Load(sender, e);

}

}

private void button1\_Click(object sender, EventArgs e)

{

this.DialogResult = DialogResult.Cancel;

}

private void btnDelete\_Click(object sender, EventArgs e)

{

string Vdg;

Vdg = DGFoodType.SelectedRows[0].Cells[0].Value.ToString();

if (Convert.ToInt64(Vdg) == 30)

{

MessageBox.Show("This value can't delete !", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

return;

}

try

{

string Fid;

frmDeleteFood DelFood = new frmDeleteFood();

Fid = DGFoodType.SelectedRows[0].Cells[0].Value.ToString();

DelFood.FtypeID = Fid;

DelFood.delFood = "delFoodType";

DelFood.ShowDialog();

if (DelFood.DialogResult == DialogResult.OK)

{

frmFood\_Type\_Load(sender, e);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void btnEdit\_Click(object sender, EventArgs e)

{

frmAddFoodType fFood\_type = new frmAddFoodType();

fFood\_type.FrmEnertry = "frmEdit";

fFood\_type.FtypeID = DGFoodType.SelectedRows[0].Cells[0].Value.ToString();

fFood\_type.txtFtype.Text = DGFoodType.SelectedRows[0].Cells[1].Value.ToString();

fFood\_type.txtFDes.Text = DGFoodType.SelectedRows[0].Cells[2].Value.ToString();

fFood\_type.ShowDialog();

if (fFood\_type.DialogResult == DialogResult.OK)

{

frmFood\_Type\_Load(sender, e);

}

}

private void btnAdd\_Click(object sender, EventArgs e)

{

frmAddFoodType fFood\_type = new frmAddFoodType();

fFood\_type.FrmEnertry = "frmAdd";

fFood\_type.ShowDialog();

if (fFood\_type.DialogResult == DialogResult.OK)

{

frmFood\_Type\_Load(sender, e);

}

}

}

}

**Form Add Table**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.OleDb;

namespace SecredCaffe

{

public partial class frmAddTable : Form

{

public string TID;

OleDbCommand CMD = new OleDbCommand();

OleDbConnection conn = new OleDbConnection();

public frmAddTable()

{

InitializeComponent();

}

private void frmAddTable\_FormClosing(object sender, FormClosingEventArgs e)

{

this.DialogResult = DialogResult.OK;

}

private void frmAddTable\_Load(object sender, EventArgs e)

{

}

private void btnOKeditT\_Click(object sender, EventArgs e)

{

AccessData data = new AccessData();

if (txtTname.Text == "")

{

MessageBox.Show("Please input Table Name!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtTname.Focus();

return;

}

string name, des;

name = txtTname.Text.Trim();

name = name.Replace("'", "''");

Boolean count = data.checkExistData("[Table]", "tbl\_name", name, "tbl\_id", Convert.ToInt32(TID));

if (count == false)

{

des = txtTdescription.Text.Trim();

des = des.Replace("'", "''");

string[] fieldNames = { "tbl\_name", "description" };

string[] values = { name, des };

data.updateData("[Table]", fieldNames, values, "tbl\_id", TID);

this.DialogResult = DialogResult.OK;

}

else

{

MessageBox.Show(name + " is Exist!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void btnOK\_Click(object sender, EventArgs e)

{

AccessData data = new AccessData();

if (txtTname.Text == "")

{

MessageBox.Show("Please input Table Name!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtTname.Focus();

return;

}

string name, des;

name = txtTname.Text.Trim();

name = name.Replace("'", "''");

Boolean count = data.checkAddExist("[Table]", "tbl\_name", name);

if (count == true)

{

MessageBox.Show(txtTname.Text.Trim() + " has already added!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else

{

des = txtTdescription.Text.Trim();

des = des.Replace("'", "''");

string[] fieldNames = { "tbl\_name", "status", "description" };

string[] values = { name, "0", des };

data.insertData("[Table]", fieldNames, values);

this.DialogResult = DialogResult.OK;

}

}

private void button3\_Click(object sender, EventArgs e)

{

this.DialogResult = DialogResult.OK;

}

}

}

**Form Change Table**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace SecredCaffe

{

public partial class frmChangeTable : Form

{

public frmChangeTable()

{

InitializeComponent();

}

public string order\_id = "";

public string frmEntryCgTable = "";

public string table\_id = "";

AccessData acc = new AccessData();

private void frmMergeTable\_Load(object sender, EventArgs e)

{

SqlDataAdapter DA = new SqlDataAdapter();

DataSet DS = new DataSet();

DA.SelectCommand = new SqlCommand("select \* from [Table] where status=0 AND tbl\_id<>'" + table\_id + "'", acc.connectDatabase(acc.serverName, acc.dbName));

DA.Fill(DS, "tbltable");

cboMergeTable.DataSource = DS.Tables["tblTable"];

cboMergeTable.DisplayMember = "tbl\_Name";

cboMergeTable.ValueMember = "tbl\_id";

cboMergeTable.SelectedIndex = 0;

//fill table name

lblTableName.Text = acc.getDataStrID("[Table]", "tbl\_name", "tbl\_id", table\_id);

}

private void btnOk\_Click(object sender, EventArgs e)

{

AccessData acc = new AccessData();

if (frmEntryCgTable.Equals("OrderCgTable"))

{

string[] fieldReceipt = { "tbl\_id", "status" };

string[] valuesReceipt = { this.cboMergeTable.SelectedValue.ToString() ,"1"};

acc.updateData("Receipt", fieldReceipt, valuesReceipt, "receipt\_id", order\_id);

//update Table New

string[] fieldTableNew = { "status" };

string[] valuesTableNew = { "1" };

acc.updateData("[Table]", fieldTableNew, valuesTableNew, "tbl\_id", this.cboMergeTable.SelectedValue.ToString());

//update table in recipt to close

string[] fieldCloseTble = {"status"};

string[] valuesCloseTble = {"0"};

acc.updateData("[Table]", fieldCloseTble, valuesCloseTble, "tbl\_id", table\_id);

this.DialogResult = DialogResult.OK;

}

}

}

}

**Form Delete Table**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace SecredCaffe

{

public partial class frmConfirmDelTable : Form

{

public frmConfirmDelTable()

{

InitializeComponent();

}

}

}

**Form Close Table**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace SecredCaffe

{

public partial class frmCloseTable : Form

{

public frmCloseTable()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

this.DialogResult = DialogResult.OK;

}

private void button2\_Click(object sender, EventArgs e)

{

this.DialogResult = DialogResult.Cancel;

}

}

}

**Form Report**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace SecredCaffe

{

public partial class frmReports : Form

{

public frmReports()

{

InitializeComponent();

}

string sql = "";

AccessData acc = new AccessData();

private void rd1\_CheckedChanged(object sender, EventArgs e)

{

if (rd1.Checked == true)

{

dt2.Visible = false;

}

else

{

dt2.Visible = true;

}

}

private void rd2\_CheckedChanged(object sender, EventArgs e)

{

if (rd2.Checked == true)

{

dt2.Visible = true;

}

else

{

dt2.Visible = false;

}

}

private void frmReports\_Load(object sender, EventArgs e)

{

dt1.Value = DateTime.Now;

dt2.Value = DateTime.Now;

}

private void btnShowReport\_Click(object sender, EventArgs e)

{

if (rd1.Checked == true)

{

Connection conn = new Connection();

AccessData Acc = new AccessData();

ADODB.Recordset rs = new ADODB.Recordset();

DataSet ds = new DataSet();

SqlDataAdapter da;

string ServerName;

string DataName;

SqlCommand cmd = new SqlCommand();

ServerName = Acc.serverName;

DataName = Acc.dbName;

conn.connectDatabase(ServerName, DataName);

cmd.Connection = conn.connectDatabase(ServerName, DataName);

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "proSaleDetailBetweenDate";

cmd.Parameters.AddWithValue("@d1", this.dt1.Text.ToString());

cmd.Parameters.AddWithValue("@d2", this.dt1.Text.ToString());

da = new SqlDataAdapter(cmd);

da.Fill(ds, "001");

rs.ActiveConnection = null;

rs.LockType = ADODB.LockTypeEnum.adLockOptimistic;

rs.Fields.Append("food\_code", ADODB.DataTypeEnum.adVarWChar, 50);

rs.Fields.Append("food\_name", ADODB.DataTypeEnum.adVarWChar, 50);

rs.Fields.Append("food\_type\_id", ADODB.DataTypeEnum.adInteger);

rs.Fields.Append("food\_type\_name", ADODB.DataTypeEnum.adVarWChar, 50);

rs.Fields.Append("price", ADODB.DataTypeEnum.adDecimal);

rs.Fields.Append("Sqty", ADODB.DataTypeEnum.adInteger);

rs.Open();

if (ds.Tables[0].Rows.Count > 0)

{

for (int i = 0; i < ds.Tables[0].Rows.Count; i++)

{

rs.AddNew();

rs.Fields["food\_code"].Value = ds.Tables[0].Rows[i][0].ToString();

rs.Fields["food\_name"].Value = ds.Tables[0].Rows[i][1].ToString();

rs.Fields["food\_type\_id"].Value = ds.Tables[0].Rows[i][2];

rs.Fields["food\_type\_name"].Value = ds.Tables[0].Rows[i][3].ToString();

rs.Fields["price"].Value = ds.Tables[0].Rows[i][4];

rs.Fields["Sqty"].Value = ds.Tables[0].Rows[i][5];

}

}

try

{

CRAXDDRT.Application crpApp = new CRAXDDRT.Application();

CRAXDDRT.Report crpRpt = new CRAXDDRT.Report();

// ADODB.Recordset recordSet = new ADODB.Recordset();

// ADODB.Connection adodbConn = new ADODB.Connection();

crpRpt = crpApp.OpenReport(Application.StartupPath + @"\Report\rptSaleDetail.rpt");

crpRpt.Database.SetDataSource(rs);

crpRpt.DiscardSavedData();

//pass parameter

crpRpt.ParameterFields[1].AddCurrentValue(this.dt1.Text);//add parameter1

crpRpt.ParameterFields[2].AddCurrentValue(this.dt1.Text);//add parameter2

rpView.ReportSource = crpRpt;

rpView.ViewReport();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cmd.Dispose();

da.Dispose();

ds.Clear();

}

else

{

Connection conn = new Connection();

AccessData Acc = new AccessData();

ADODB.Recordset rs = new ADODB.Recordset();

DataSet ds = new DataSet();

SqlDataAdapter da;

string ServerName;

string DataName;

SqlCommand cmd = new SqlCommand();

ServerName = Acc.serverName;

DataName = Acc.dbName;

conn.connectDatabase(ServerName, DataName);

cmd.Connection = conn.connectDatabase(ServerName, DataName);

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "proSaleDetailBetweenDate";

cmd.Parameters.AddWithValue("@d1", this.dt1.Text.ToString());

cmd.Parameters.AddWithValue("@d2", this.dt2.Text.ToString());

da = new SqlDataAdapter(cmd);

da.Fill(ds, "001");

rs.ActiveConnection = null;

rs.LockType = ADODB.LockTypeEnum.adLockOptimistic;

rs.Fields.Append("food\_code", ADODB.DataTypeEnum.adVarWChar, 50);

rs.Fields.Append("food\_name", ADODB.DataTypeEnum.adVarWChar, 50);

rs.Fields.Append("food\_type\_id", ADODB.DataTypeEnum.adInteger);

rs.Fields.Append("food\_type\_name", ADODB.DataTypeEnum.adVarWChar, 50);

rs.Fields.Append("price", ADODB.DataTypeEnum.adDecimal);

rs.Fields.Append("Sqty", ADODB.DataTypeEnum.adInteger);

rs.Open();

if (ds.Tables[0].Rows.Count > 0)

{

for (int i = 0; i < ds.Tables[0].Rows.Count; i++)

{

rs.AddNew();

rs.Fields["food\_code"].Value = ds.Tables[0].Rows[i][0].ToString();

rs.Fields["food\_name"].Value = ds.Tables[0].Rows[i][1].ToString();

rs.Fields["food\_type\_id"].Value = ds.Tables[0].Rows[i][2];

rs.Fields["food\_type\_name"].Value = ds.Tables[0].Rows[i][3].ToString();

rs.Fields["price"].Value = ds.Tables[0].Rows[i][4];

rs.Fields["Sqty"].Value = ds.Tables[0].Rows[i][5];

}

}

try

{

CRAXDDRT.Application crpApp = new CRAXDDRT.Application();

CRAXDDRT.Report crpRpt = new CRAXDDRT.Report();

// ADODB.Recordset recordSet = new ADODB.Recordset();

// ADODB.Connection adodbConn = new ADODB.Connection();

crpRpt = crpApp.OpenReport(Application.StartupPath + @"\Report\rptSaleDetail.rpt");

crpRpt.Database.SetDataSource(rs);

crpRpt.DiscardSavedData();

//pass parameter

crpRpt.ParameterFields[1].AddCurrentValue(this.dt1.Text);//add parameter1

crpRpt.ParameterFields[2].AddCurrentValue(this.dt2.Text);//add parameter2

rpView.ReportSource = crpRpt;

rpView.ViewReport();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

cmd.Dispose();

da.Dispose();

ds.Clear();

}

}

}

}

**Form Setting**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Globalization;

using CrystalDecisions.CrystalReports.Engine;

namespace SecredCaffe

{

public partial class frmSetting : Form

{

public frmSetting()

{

InitializeComponent();

}

public string user = "";

private void frmSetting\_Load(object sender, EventArgs e)

{

//display printer device name

foreach (string printer in System.Drawing.Printing.PrinterSettings.InstalledPrinters)

{

cboPrinter.Items.Add(printer);

}

this.cboPrinter.SelectedIndex = 0;

//fill textboxes

AccessData mydata = new AccessData();

string id = mydata.selectMaxID("id", "Setting");

Int32 staffid = Convert.ToInt32(mydata.getDataStrID("Staff", "staff\_id", "staff\_name", user));

this.txtDiscount.Text = mydata.getData("Setting", "discount", "id", id);

this.txtExchange.Text = mydata.getData("Setting", "exchangerate", "id", id);

this.txtCmpKh.Text = mydata.getData("Setting", "cmp\_name\_kh", "id", id);

this.txtCpmEng.Text = mydata.getData("Setting", "cmp\_name\_eng", "id", id);

this.txtAddress.Text = mydata.getData("Setting", "address", "id", id);

this.txtPh1.Text = mydata.getData("Setting", "phone1", "id", id);

this.txtPh2.Text = mydata.getData("Setting", "phone2", "id", id);

string pr\_name = mydata.getData("Setting", "printer\_name", "id", id);

this.cboPrinter.Text = pr\_name;

mydata.fillComboBox("Staff", "staff\_name", "staff\_id", this.cboDefaultUser);

this.cboDefaultUser.SelectedValue=staffid;

}

private void txtDiscount\_Enter(object sender, EventArgs e)

{

foreach (InputLanguage Lng in InputLanguage.InstalledInputLanguages)

{

if (Lng.Culture.DisplayName.ToString().ToUpper().StartsWith("ENG"))

{

InputLanguage.CurrentInputLanguage = Lng;

}

}

if (txtDiscount.Focused)

{

foreach (InputLanguage Lng in InputLanguage.InstalledInputLanguages)

{

if (Lng.Culture.DisplayName.ToString().ToUpper().StartsWith("ENG"))

{

InputLanguage.CurrentInputLanguage = Lng;

}

}

}

}

private void txtExchange\_Enter(object sender, EventArgs e)

{

foreach (InputLanguage Lng in InputLanguage.InstalledInputLanguages)

{

if (Lng.Culture.DisplayName.ToString().ToUpper().StartsWith("ENG"))

{

InputLanguage.CurrentInputLanguage = Lng;

}

}

if (txtExchange.Focused)

{

foreach (InputLanguage Lng in InputLanguage.InstalledInputLanguages)

{

if (Lng.Culture.DisplayName.ToString().ToUpper().StartsWith("ENG"))

{

InputLanguage.CurrentInputLanguage = Lng;

}

}

}

}

private void txtDiscount\_KeyPress(object sender, KeyPressEventArgs e)

{

if (this.txtDiscount.Text.Contains("."))

{

if (!(Char.IsDigit(e.KeyChar) || (e.KeyChar == (char)Keys.Back)))

{

e.Handled = true;

System.Media.SystemSounds.Exclamation.Play();

}

}

else

{

if (!(Char.IsDigit(e.KeyChar) || (e.KeyChar == (char)Keys.Back || (e.KeyChar == (char)46))))

{

e.Handled = true;

System.Media.SystemSounds.Exclamation.Play();

}

}

}

private void txtExchange\_KeyPress(object sender, KeyPressEventArgs e)

{

if (this.txtExchange.Text.Contains("."))

{

if (!(Char.IsDigit(e.KeyChar) || (e.KeyChar == (char)Keys.Back)))

{

e.Handled = true;

System.Media.SystemSounds.Exclamation.Play();

}

}

else

{

if (!(Char.IsDigit(e.KeyChar) || (e.KeyChar == (char)Keys.Back || (e.KeyChar == (char)46))))

{

e.Handled = true;

System.Media.SystemSounds.Exclamation.Play();

}

}

}

private void btnSave\_Click(object sender, EventArgs e)

{

//update setting

AccessData mydata = new AccessData();

string id = mydata.selectMaxID("id", "Setting");

//check validate discount condition

double validateDes = Convert.ToDouble(this.txtDiscount.Text.ToString());

if (validateDes < 0 || validateDes > 100)

{

MessageBox.Show("Descount Must be In Rank 0 to 100.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string[] fields = { "exchangerate", "staff\_id", "cmp\_name\_eng", "cmp\_name\_kh", "address", "phone1", "phone2", "printer\_name" };

string[] values = { this.txtExchange.Text.ToString(), this.cboDefaultUser.SelectedValue.ToString(), this.txtCpmEng.Text.Trim(), this.txtCmpKh.Text.Trim(), this.txtAddress.Text.Trim(), this.txtPh1.Text.Trim(), this.txtPh2.Text.Trim(), this.cboPrinter.Text };

mydata.updateData("Setting", fields, values, "id", id);

SqlCommand cmdUpdateSetting = new SqlCommand("UPDATE [Setting] SET discount='" + float.Parse(this.txtDiscount.Text.ToString(), CultureInfo.InvariantCulture.NumberFormat) + "'", mydata.connectDatabase(mydata.serverName, mydata.dbName));

cmdUpdateSetting.ExecuteNonQuery();

this.Close();

}

}

}

**Form Add Staff**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace SecredCaffe

{

public partial class frmAddstaff : Form

{

public string StaffID;

public string staffName;

public string ServerName;

public string DataName;

AccessData data = new AccessData();

public frmAddstaff()

{

InitializeComponent();

}

private void frmAddstaff\_Load(object sender, EventArgs e)

{

}

private void frmAddstaff\_FormClosing(object sender, FormClosingEventArgs e)

{

this.DialogResult = DialogResult.OK;

}

private void btnBrowPermiss\_Click(object sender, EventArgs e)

{

AccessData acc = new AccessData();

frmAddPermissions objPermiss = new frmAddPermissions();

objPermiss.frmEntry = "frmAdd";

objPermiss.ShowDialog();

if (objPermiss.DialogResult == DialogResult.OK)

{

acc.fillComboBox("StaffType", "per\_name", "per\_id", this.cbotype);

}

}

private void button1\_Click(object sender, EventArgs e)

{

string name, sex, tel, email, address, pwd, type,validateName,q;

name = txtname.Text.Trim();

name = name.Replace("'", "''");

tel = txttel.Text.Trim();

email = txtemail.Text.Trim();

address = txtaddress.Text.Trim();

address = address.Replace("'", "''");

pwd = txtpass.Text.Trim();

sex = cbosex.Text;

AccessData Acc = new AccessData();

ServerName = Acc.serverName;

DataName = Acc.dbName;

Connection cn = new Connection();

cn.connectDatabase(ServerName, DataName);

q = "select count(\*) as CstName from Staff where staff\_name='" + name + "'";

SqlCommand cmded = new SqlCommand(q, cn.connectDatabase(ServerName, DataName));

validateName = cmded.ExecuteScalar().ToString();

if (txtname.Text == "")

{

MessageBox.Show("Please input staff name!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtname.Focus();

return;

}

else if (name != staffName)

{

if (!validateName.Equals("0"))

{

MessageBox.Show(this, "Name is existing !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtname.Focus();

txtname.Clear();

return;

}

}

else if (cbotype.Text == "")

{

MessageBox.Show("Please choose Staff Type!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

cbotype.Focus();

return;

}

type = cbotype.SelectedValue.ToString();

string[] fields = { "staff\_name", "staff\_sex", "per\_id", "staff\_tell", "staff\_email", "staff\_address", "pwd" };

string[] values = { name, sex, type, tel, email, address, pwd };

if ((txtpass.Text) != "" || (txtrepass.Text) != "")

{

if ((txtpass.Text) != (txtrepass.Text))

{

MessageBox.Show("Password not match!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtpass.Text = "";

txtrepass.Text = "";

txtpass.Focus();

return;

}

}

data.updateData("[Staff]", fields, values, "staff\_id", StaffID);

this.DialogResult = DialogResult.OK;

}

private void btnCancel\_Click(object sender, EventArgs e)

{

this.DialogResult = DialogResult.OK;

}

private void btnok\_Click(object sender, EventArgs e)

{

string name, sex, tel, email, address, pwd, type,validateName,q;

name = txtname.Text.Trim();

name = name.Replace("'", "''");

tel = txttel.Text.Trim();

email = txtemail.Text.Trim();

address = txtaddress.Text.Trim();

address = address.Replace("'", "''");

pwd = txtpass.Text.Trim();

sex = cbosex.Text;

AccessData Acc = new AccessData();

ServerName = Acc.serverName;

DataName = Acc.dbName;

Connection cn = new Connection();

cn.connectDatabase(ServerName, DataName);

q = "select count(\*) as CstName from Staff where staff\_name='" + name + "'";

SqlCommand cmded = new SqlCommand(q, cn.connectDatabase(ServerName, DataName));

validateName = cmded.ExecuteScalar().ToString();

if (txtname.Text == "")

{

MessageBox.Show("Please input staff name!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtname.Focus();

return;

}

if (!validateName.Equals("0"))

{

MessageBox.Show(this, "Name is existing !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtname.Focus();

txtname.Clear();

return;

}

if (cbotype.Text == "")

{

MessageBox.Show("Please choose Staff Type!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

cbotype.Focus();

return;

}

type = cbotype.SelectedValue.ToString();

string[] fields = { "staff\_name", "staff\_sex", "per\_id", "staff\_tell", "staff\_email", "staff\_address", "pwd" };

string[] values = { name, sex, type, tel, email, address, pwd };

if ((txtpass.Text) != "" || (txtrepass.Text) != "")

{

if ((txtpass.Text) != (txtrepass.Text))

{

MessageBox.Show("Password not match!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtpass.Text = "";

txtrepass.Text = "";

txtpass.Focus();

return;

}

}

data.insertData("[Staff]", fields, values);

this.DialogResult = DialogResult.OK;

}

}

}

**Form Add Permission**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace SecredCaffe

{

public partial class frmAddPermissions : Form

{

public frmAddPermissions()

{

InitializeComponent();

}

public string frmEntry = "";

public string ValID = "";

public string pername = "";

string per\_operation = "";

string per\_order = "";

string per\_orderItems = "";

string per\_closeTble= "";

string per\_food = "";

string per\_addFood = "";

string per\_editFood = "";

string per\_delFood = "";

string per\_table = "";

string per\_addTable = "";

string per\_editTable = "";

string per\_delTable = "";

string per\_setting = "";

string per\_emp = "";

string per\_addEmp = "";

string per\_editEmp = "";

string per\_delEmp = "";

string per\_setSetting = "";

string per\_cgPwd = "";

string per\_permiss = "";

string per\_addPermiss = "";

string per\_editPermiss = "";

string per\_delPermiss = "";

string per\_report = "";

string per\_repOrder = "";

string per\_data = "";

string per\_backup = "";

string per\_restore = "";

private void btnCancel\_Click(object sender, EventArgs e)

{

this.Close();

}

private void frmAddPermissions\_Load(object sender, EventArgs e)

{

AccessData acc = new AccessData();

string server = acc.serverName;

string db = acc.dbName;

acc.connectDatabase(server, db);

if (frmEntry.Equals("frmAdd"))

{

this.AcceptButton = btnOk;

this.btnOk.Visible = true;

this.btnEdit.Visible = false;

this.Text = "Add Permission";

}

else if (frmEntry.Equals("frmEdit"))

{

this.Text = "Edit Permission";

this.AcceptButton = btnEdit;

this.btnOk.Visible = false;

this.btnEdit.Visible = true;

SqlCommand cmd = new SqlCommand("select \* from StaffType where per\_id=" + ValID + "", acc.connectDatabase(server, db));

SqlDataReader DR = cmd.ExecuteReader();

while (DR.Read())

{

tvPermissions.Nodes["NodeOperation"].Checked = Convert.ToBoolean(DR["per\_op"].ToString());

tvPermissions.Nodes["NodeSetting"].Checked = Convert.ToBoolean(DR["per\_setting"].ToString());

tvPermissions.Nodes["NodeReport"].Checked = Convert.ToBoolean(DR["per\_report"].ToString());

tvPermissions.Nodes["NodeData"].Checked = Convert.ToBoolean(DR["per\_data"].ToString());

try

{

foreach (TreeNode NodeOperation in tvPermissions.Nodes.Find("NodeOperation", true))

{

NodeOperation.Nodes["NodeOrder"].Checked = Convert.ToBoolean(DR["per\_order"].ToString());

NodeOperation.Nodes["NodeOrderItems"].Checked = Convert.ToBoolean(DR["per\_orderItems"].ToString());

NodeOperation.Nodes["NodeCloseTbl"].Checked = Convert.ToBoolean(DR["per\_closeTbl"].ToString());

NodeOperation.Nodes["NodeFood"].Checked = Convert.ToBoolean(DR["per\_food"].ToString());

NodeOperation.Nodes["NodeAddFood"].Checked = Convert.ToBoolean(DR["per\_addfood"].ToString());

NodeOperation.Nodes["NodeEditFood"].Checked = Convert.ToBoolean(DR["per\_editfood"].ToString());

NodeOperation.Nodes["NodeDelFood"].Checked = Convert.ToBoolean(DR["per\_delfood"].ToString());

NodeOperation.Nodes["NodeTable"].Checked = Convert.ToBoolean(DR["per\_table"].ToString());

NodeOperation.Nodes["NodeAddTable"].Checked = Convert.ToBoolean(DR["per\_addtable"].ToString());

NodeOperation.Nodes["NodeEditTable"].Checked = Convert.ToBoolean(DR["per\_edittable"].ToString());

NodeOperation.Nodes["NodeDelTable"].Checked = Convert.ToBoolean(DR["per\_deltable"].ToString());

}

}

catch

{

}

try

{

foreach (TreeNode NodeSetting in tvPermissions.Nodes.Find("NodeSetting", true))

{

NodeSetting.Nodes["NodeEmp"].Checked = Convert.ToBoolean(DR["per\_emp"].ToString());

NodeSetting.Nodes["NodeAddEmp"].Checked = Convert.ToBoolean(DR["per\_addemp"].ToString());

NodeSetting.Nodes["NodeEditEmp"].Checked = Convert.ToBoolean(DR["per\_editemp"].ToString());

NodeSetting.Nodes["NodeDelEmp"].Checked = Convert.ToBoolean(DR["per\_delemp"].ToString());

NodeSetting.Nodes["NodeSetSetting"].Checked = Convert.ToBoolean(DR["per\_setsetting"].ToString());

NodeSetting.Nodes["NodeCgPwd"].Checked = Convert.ToBoolean(DR["per\_cgpwd"].ToString());

NodeSetting.Nodes["NodePermiss"].Checked = Convert.ToBoolean(DR["per\_permiss"].ToString());

NodeSetting.Nodes["NodeAddPermiss"].Checked = Convert.ToBoolean(DR["per\_addpermiss"].ToString());

NodeSetting.Nodes["NodeEditPermiss"].Checked = Convert.ToBoolean(DR["per\_editpermiss"].ToString());

NodeSetting.Nodes["NodeDelPermiss"].Checked = Convert.ToBoolean(DR["per\_delpermiss"].ToString());

}

}

catch

{

}

try

{

foreach (TreeNode NodeRep in tvPermissions.Nodes.Find("NodeReport", true))

{

NodeRep.Nodes["NodeRepOrder"].Checked = Convert.ToBoolean(DR["per\_reporder"].ToString());

}

}

catch

{

}

try

{

foreach (TreeNode NodeData in tvPermissions.Nodes.Find("NodeData", true))

{

NodeData.Nodes["NodeBackup"].Checked = Convert.ToBoolean(DR["per\_backup"].ToString());

NodeData.Nodes["NodeRestore"].Checked = Convert.ToBoolean(DR["per\_restore"].ToString());

}

}

catch

{

}

}

}

}

private void CheckAllChildNodes(TreeNode treeNode, bool nodeChecked)

{

foreach (TreeNode node in treeNode.Nodes)

{

node.Checked = nodeChecked;

if (node.Nodes.Count > 0)

{

// If the current node has child nodes, call the CheckAllChildsNodes method recursively.

this.CheckAllChildNodes(node, nodeChecked);

}

}

}

private void SelectParents(TreeNode node, Boolean isChecked)

{

var parent = node.Parent;

if (parent == null)

return;

if (!isChecked && HasCheckedNode(parent))

return;

parent.Checked = isChecked;

SelectParents(parent, isChecked);

}

private bool HasCheckedNode(TreeNode node)

{

return node.Nodes.Cast<TreeNode>().Any(n => n.Checked);

}

private void tvPermissions\_AfterCheck(object sender, TreeViewEventArgs e)

{

// bool updatingTreeView=false;

if (e.Action != TreeViewAction.Unknown)

{

if (e.Node.Nodes.Count > 0)

{

this.CheckAllChildNodes(e.Node, e.Node.Checked);

}

}

this.SelectParents(e.Node, e.Node.Checked);

}

private void btnOk\_Click\_1(object sender, EventArgs e)

{

AccessData acc = new AccessData();

string ServerName;

string DataName;

string q = "";

ServerName = acc.serverName;

DataName = acc.dbName;

string ValCodeAdd;

Connection cnn = new Connection();

cnn.connectDatabase(ServerName, DataName);

q = "select count(\*) as CPermiss from StaffType where per\_name='" + txtPerName.Text + "'";

SqlCommand cmde = new SqlCommand(q, cnn.connectDatabase(ServerName, DataName));

ValCodeAdd = cmde.ExecuteScalar().ToString();

if (this.txtPerName.Text.Equals(""))

{

MessageBox.Show(this, "Enter Name !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

this.txtPerName.Focus();

return;

}

else if (!ValCodeAdd.Equals("0"))

{

MessageBox.Show(this, "Name is existing !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtPerName.Focus();

txtPerName.Clear();

return;

}

try

{

try

{

if (tvPermissions.Nodes["NodeOperation"].Checked == true) per\_operation = "true";

else per\_operation = "false";

if (tvPermissions.Nodes["NodeSetting"].Checked == true) per\_setting = "true";

else per\_setting = "false";

if (tvPermissions.Nodes["NodeReport"].Checked == true) per\_report = "true";

else per\_report = "false";

if (tvPermissions.Nodes["NodeData"].Checked == true) per\_data = "true";

else per\_data = "false";

foreach (TreeNode Node in tvPermissions.Nodes.Find("NodeOperation", true))

{

if (Node.Nodes["NodeOrder"].Checked == true) per\_order = "true";

else per\_order = "false";

if (Node.Nodes["NodeOrderItems"].Checked == true) per\_orderItems = "true";

else per\_orderItems = "false";

if (Node.Nodes["NodeCloseTbl"].Checked == true) per\_closeTble = "true";

else per\_closeTble = "false";

if (Node.Nodes["NodeFood"].Checked == true) per\_food = "true";

else per\_food = "false";

if (Node.Nodes["NodeAddFood"].Checked == true) per\_addFood = "true";

else per\_addFood = "false";

if (Node.Nodes["NodeEditFood"].Checked == true) per\_editFood = "true";

else per\_editFood = "false";

if (Node.Nodes["NodeDelFood"].Checked == true) per\_delFood = "true";

else per\_delFood = "false";

if (Node.Nodes["NodeTable"].Checked == true) per\_table = "true";

else per\_table = "false";

if (Node.Nodes["NodeAddTable"].Checked == true) per\_addTable = "true";

else per\_addTable = "false";

if (Node.Nodes["NodeEditTable"].Checked == true) per\_editTable = "true";

else per\_editTable = "false";

if (Node.Nodes["NodeDelCus"].Checked == true) per\_delTable = "true";

else per\_delTable = "false";

}

}

catch

{

}

try

{

foreach (TreeNode NodeSetting in tvPermissions.Nodes.Find("NodeSetting", true))

{

if (NodeSetting.Nodes["NodeEmp"].Checked == true) per\_emp = "true";

else per\_emp = "false";

if (NodeSetting.Nodes["NodeAddEmp"].Checked == true) per\_addEmp = "true";

else per\_addEmp = "false";

if (NodeSetting.Nodes["NodeEditEmp"].Checked == true) per\_editEmp = "true";

else per\_editEmp = "false";

if (NodeSetting.Nodes["NodeDelEmp"].Checked == true) per\_delEmp = "true";

else per\_delEmp = "false";

if (NodeSetting.Nodes["NodeSetSetting"].Checked == true) per\_setSetting = "true";

else per\_setSetting = "false";

if (NodeSetting.Nodes["NodeCgPwd"].Checked == true) per\_cgPwd = "true";

else per\_cgPwd = "false";

if (NodeSetting.Nodes["NodePermiss"].Checked == true) per\_permiss = "true";

else per\_permiss = "false";

if (NodeSetting.Nodes["NodeAddPermiss"].Checked == true) per\_addPermiss = "true";

else per\_addPermiss = "false";

if (NodeSetting.Nodes["NodeEditPermiss"].Checked == true) per\_editPermiss = "true";

else per\_editPermiss = "false";

if (NodeSetting.Nodes["NodeDelPermiss"].Checked == true) per\_delPermiss = "true";

else per\_delPermiss = "false";

}

}

catch

{

}

try

{

foreach (TreeNode NodeRep in tvPermissions.Nodes.Find("NodeReport", true))

{

if (NodeRep.Nodes["NodeRepOrder"].Checked == true) per\_repOrder = "true";

else per\_repOrder = "false";

}

}

catch

{

}

try

{

foreach (TreeNode NodeData in tvPermissions.Nodes.Find("NodeData", true))

{

if (NodeData.Nodes["NodeBackup"].Checked == true) per\_backup = "true";

else per\_backup = "false";

if (NodeData.Nodes["NodeRestore"].Checked == true) per\_restore = "true";

else per\_restore = "false";

}

}

catch

{

}

string[] field = { "per\_name", "per\_des", "per\_op", "per\_orderItems", "per\_closeTbl","per\_order", "per\_food", "per\_addfood", "per\_editfood", "per\_delfood", "per\_table", "per\_addtable", "per\_edittable", "per\_deltable", "per\_setting", "per\_emp", "per\_addemp", "per\_editemp", "per\_delemp", "per\_setsetting", "per\_cgpwd", "per\_permiss", "per\_addpermiss", "per\_editpermiss", "per\_delpermiss", "per\_report", "per\_reporder", "per\_data", "per\_backup", "per\_restore" };

string[] values = { this.txtPerName.Text.Trim(), this.txtDes.Text.Trim(), per\_operation, per\_order, per\_orderItems,per\_closeTble, per\_food, per\_addFood, per\_editFood, per\_delFood, per\_table, per\_addTable, per\_editTable, per\_delTable, per\_setting, per\_emp, per\_addEmp, per\_editEmp, per\_delEmp, per\_setSetting, per\_cgPwd, per\_permiss, per\_addPermiss, per\_editPermiss, per\_delPermiss, per\_report, per\_repOrder, per\_data, per\_backup, per\_restore };

acc.insertData("StaffType", field, values);

this.DialogResult = DialogResult.OK;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void btnEdit\_Click(object sender, EventArgs e)

{

AccessData acc = new AccessData();

string ServerName;

string DataName;

string q = "";

ServerName = acc.serverName;

DataName = acc.dbName;

string ValCodeAdd;

Connection cnn = new Connection();

cnn.connectDatabase(ServerName, DataName);

q = "select count(\*) as CPermiss from StaffType where per\_name='" + txtPerName.Text + "'";

SqlCommand cmde = new SqlCommand(q, cnn.connectDatabase(ServerName, DataName));

ValCodeAdd = cmde.ExecuteScalar().ToString();

if (this.txtPerName.Text.Equals(""))

{

MessageBox.Show(this, "Enter Name !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

this.txtPerName.Focus();

return;

}

else if (txtPerName.Text != pername)

{

if (!ValCodeAdd.Equals("0"))

{

MessageBox.Show(this, "Name is existing !", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtPerName.Focus();

txtPerName.Clear();

return;

}

}

try

{

if (tvPermissions.Nodes["NodeOperation"].Checked == true) per\_operation = "true";

else per\_operation = "false";

if (tvPermissions.Nodes["NodeSetting"].Checked == true) per\_setting = "true";

else per\_setting = "false";

if (tvPermissions.Nodes["NodeReport"].Checked == true) per\_report = "true";

else per\_report = "false";

if (tvPermissions.Nodes["NodeData"].Checked == true) per\_data = "true";

else per\_data = "false";

foreach (TreeNode Node in tvPermissions.Nodes.Find("NodeOperation", true))

{

if (Node.Nodes["NodeOrder"].Checked == true) per\_order = "true";

else per\_order = "false";

if (Node.Nodes["NodeOrderItems"].Checked == true) per\_orderItems = "true";

else per\_orderItems = "false";

if (Node.Nodes["NodeCloseTbl"].Checked == true) per\_closeTble = "true";

else per\_closeTble = "false";

if (Node.Nodes["NodeFood"].Checked == true) per\_food = "true";

else per\_food = "false";

if (Node.Nodes["NodeAddFood"].Checked == true) per\_addFood = "true";

else per\_addFood = "false";

if (Node.Nodes["NodeEditFood"].Checked == true) per\_editFood = "true";

else per\_editFood = "false";

if (Node.Nodes["NodeDelFood"].Checked == true) per\_delFood = "true";

else per\_delFood = "false";

if (Node.Nodes["NodeTable"].Checked == true) per\_table = "true";

else per\_table = "false";

if (Node.Nodes["NodeAddTable"].Checked == true) per\_addTable = "true";

else per\_addTable = "false";

if (Node.Nodes["NodeEditTable"].Checked == true) per\_editTable = "true";

else per\_editTable = "false";

if (Node.Nodes["NodeDelCus"].Checked == true) per\_delTable = "true";

else per\_delTable = "false";

}

}

catch

{

}

try

{

foreach (TreeNode NodeSetting in tvPermissions.Nodes.Find("NodeSetting", true))

{

if (NodeSetting.Nodes["NodeEmp"].Checked == true) per\_emp = "true";

else per\_emp = "false";

if (NodeSetting.Nodes["NodeAddEmp"].Checked == true) per\_addEmp = "true";

else per\_addEmp = "false";

if (NodeSetting.Nodes["NodeEditEmp"].Checked == true) per\_editEmp = "true";

else per\_editEmp = "false";

if (NodeSetting.Nodes["NodeDelEmp"].Checked == true) per\_delEmp = "true";

else per\_delEmp = "false";

if (NodeSetting.Nodes["NodeSetSetting"].Checked == true) per\_setSetting = "true";

else per\_setSetting = "false";

if (NodeSetting.Nodes["NodeCgPwd"].Checked == true) per\_cgPwd = "true";

else per\_cgPwd = "false";

if (NodeSetting.Nodes["NodePermiss"].Checked == true) per\_permiss = "true";

else per\_permiss = "false";

if (NodeSetting.Nodes["NodeAddPermiss"].Checked == true) per\_addPermiss = "true";

else per\_addPermiss = "false";

if (NodeSetting.Nodes["NodeEditPermiss"].Checked == true) per\_editPermiss = "true";

else per\_editPermiss = "false";

if (NodeSetting.Nodes["NodeDelPermiss"].Checked == true) per\_delPermiss = "true";

else per\_delPermiss = "false";

}

}

catch

{

}

try

{

foreach (TreeNode NodeRep in tvPermissions.Nodes.Find("NodeReport", true))

{

if (NodeRep.Nodes["NodeRepOrder"].Checked == true) per\_repOrder = "true";

else per\_repOrder = "false";

}

}

catch

{

}

try

{

foreach (TreeNode NodeData in tvPermissions.Nodes.Find("NodeData", true))

{

if (NodeData.Nodes["NodeBackup"].Checked == true) per\_backup = "true";

else per\_backup = "false";

if (NodeData.Nodes["NodeRestore"].Checked == true) per\_restore = "true";

else per\_restore = "false";

}

}

catch

{

}

string[] field = { "per\_name", "per\_des", "per\_op", "per\_order", "per\_orderItems","per\_closeTbl", "per\_food", "per\_addfood", "per\_editfood", "per\_delfood", "per\_table", "per\_addtable", "per\_edittable", "per\_deltable", "per\_setting", "per\_emp", "per\_addemp", "per\_editemp", "per\_delemp", "per\_setsetting", "per\_cgpwd", "per\_permiss", "per\_addpermiss", "per\_editpermiss", "per\_delpermiss", "per\_report", "per\_reporder", "per\_data", "per\_backup", "per\_restore" };

string[] values = { this.txtPerName.Text.Trim(), this.txtDes.Text.Trim(), per\_operation, per\_order, per\_orderItems,per\_closeTble, per\_food, per\_addFood, per\_editFood, per\_delFood, per\_table, per\_addTable, per\_editTable, per\_delTable, per\_setting, per\_emp, per\_addEmp, per\_editEmp, per\_delEmp, per\_setSetting, per\_cgPwd, per\_permiss, per\_addPermiss, per\_editPermiss, per\_delPermiss, per\_report, per\_repOrder, per\_data, per\_backup, per\_restore };

acc.updateData("StaffType", field, values, "per\_id", ValID);

this.DialogResult = DialogResult.OK;

}

private void button3\_Click(object sender, EventArgs e)

{

this.DialogResult = DialogResult.Cancel;

}

}

}

**Form Delete Permission**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace SecredCaffe

{

public partial class frmDeletePermiss : Form

{

public frmDeletePermiss()

{

InitializeComponent();

}

public string per\_id = "";

private void btnOk\_Click(object sender, EventArgs e)

{

AccessData acc = new AccessData();

acc.deleteRecord("StaffType", "per\_id", per\_id);

this.DialogResult = DialogResult.OK;

}

private void btnCancel\_Click(object sender, EventArgs e)

{

this.DialogResult = DialogResult.Cancel;

}

}

}

**Form Change Password**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace SecredCaffe

{

public partial class frmChangePwd : Form

{

public string userName = "";

public string pwd = "";

public void setUserPwd(string u, string p)

{

userName = u;

pwd = p;

}

public string getUser()

{

return userName;

}

public string getPwd()

{

return pwd;

}

public frmChangePwd()

{

InitializeComponent();

}

public frmChangePwd(string userName,string pwd)

{

InitializeComponent();

this.setUserPwd(userName, pwd);

}

private void btnOk\_Click\_1(object sender, EventArgs e)

{

if (!(this.txtOldPwd.Text.Equals(this.getPwd())))

{

frmChangePwdInfo frmInfo = new frmChangePwdInfo("ពាក្យសម្ងាត់ចាស់មិនត្រឹមត្រូវ\nOld password doesn't match!");

frmInfo.ShowDialog();

}

else if (!(txtNewPwd.Text.Equals(txtConfirmPwd.Text)))

{

frmChangePwdInfo frmInfo = new frmChangePwdInfo("ពាក្យសម្ងាត់ថ្មីមិនផ្គូរផ្គង\nNew password doesn't match!");

frmInfo.ShowDialog();

}

else

{

AccessData dataPwd = new AccessData();

string id = dataPwd.getDataTwoStrID("Staff", "staff\_id", "staff\_name", this.getUser(), "pwd", this.getPwd());

string[] field = { "pwd" };

string[] val = { txtNewPwd.Text };

dataPwd.updateData("Staff", field, val, "staff\_id", id);

frmChangePwdInfo frmInfo = new frmChangePwdInfo("Password updated!");

frmInfo.ShowDialog();

this.Close();

}

}

private void btnCancel\_Click\_1(object sender, EventArgs e)

{

this.DialogResult = DialogResult.Cancel;

}

}

}

**Form Comfirm Backup**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace SecredCaffe

{

public partial class frmBR : Form

{

public frmBR()

{

InitializeComponent();

}

public string fRes;

public string frmBK;

private void btnClose\_Click(object sender, EventArgs e)

{

if (frmBK.Equals("res"))

{

Application.Exit();

}

else if (frmBK.Equals("bk"))

{

this.DialogResult = DialogResult.OK;

}

}

private void frmBR\_Load(object sender, EventArgs e)

{

}

private void frmBR\_FormClosing(object sender, FormClosingEventArgs e)

{

if (frmBK.Equals("res"))

{

Application.Exit();

}

else if (frmBK.Equals("bk"))

{

this.DialogResult = DialogResult.OK;

}

}

}

}

.