



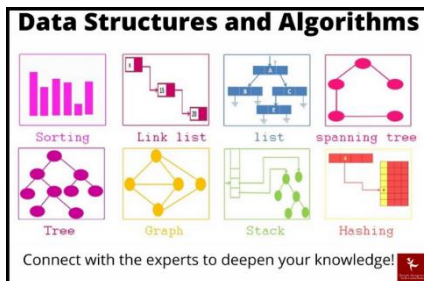
សាកលវិទ្យាល័យ នំរតុន
NORTON UNIVERSITY

ព្រះរាជាណាចក្រកម្ពុជា
ជាតិ សាសនា ព្រះមហាក្សត្រ

Norton University
College of Sciences
Department of computer

Year: II, Group: A 1
Session: Afternoon
Semester II

Data Structure and Algorithms in Python
Topic : School Management System



Submitted by :

Name :

1. Boun Hongly
2. Vorn Chiva
3. Seng Kimsan
4. Chea Sopheaktra

ID:

B20230881
B20232652
B20233491
B20215189

Submitted to
Mrs. Chhay Nuppkun

Academic Year
2023 -2024

អារម្ភកថា

កម្រងកិច្ចការស្រាវជ្រាវមួយនេះ ត្រូវបានចងក្រងឡើងក្នុងគោលបំណងដើម្បីពង្រឹងចំណេះដឹងរបស់យើងខ្ញុំទាំងអស់គ្នា និងដើម្បីជាជំនួយដល់ សិស្សានុសិស្ស យុវជន យុវនារី រួមទាំងបងប្អូនគ្រប់គ្នាដែលពុំ ទាន់ចំណេះដឹងទាក់ទងទៅនឹង Data Structure and Algorithms និងការបង្កើតកម្មវិធី ឬការគ្រប់គ្រងប្រព័ន្ធ នៅក្នុងក្រុមហ៊ុនឬ ហាង។ Data Structure and Algorithms គឺជាការសរសេរកម្មវិធី ការដោះស្រាយបញ្ហា ទៅតាមដំណាក់កាលៗ ជា Flowchart និងម្យ៉ាងទៀត Algorithms គឺជាខ្សែរ Statement ដ៏ច្បាស់ សាសន៍ដែលក្រោយពីការកំណត់ជាក់លាក់រហូតយើងទទួលបានលទ្ធផលដែលយើងចង់បាន។ កម្រងកិច្ចការស្រាវជ្រាវនេះ គឺចាប់ផ្តើមពីកម្រិតដំបូងដែលមានការណែនាំឲ្យស្គាល់ថា តើអ្វី Data Structure and Algorithms និងមានប្រាប់ពីវិធីគ្រប់គ្រងទៅលើប្រព័ន្ធបណ្តាញមួយដែលមានភាពងាយ ស្រួលដល់អ្នកគ្រប់គ្រង រួមទាំងពន្យល់មេរៀនយ៉ាងក្លាយ គ្រប់ចំណុចទាំងអស់របស់មេរៀនផងដែរ។ ក្នុងនោះផងដែរ ក្រុមនិសិ្សយើងខ្ញុំបានខិតខំប្រឹងប្រែងសិក្សាស្រាវជ្រាវគ្រប់ឯកសារដោយស្រង់ យកចំណុចសំខាន់ៗមកចងក្រងដើម្បីជាជំនួយស្តារពីដល់សិស្សានុសិស្ស និសិ្សជំនាន់ក្រោយឬមិត្តអ្នកអាន ដែលចង់ចេះនិងចង់យល់ដឹងច្បាស់លាស់អំពី Data Structure and Algorithms ។ ក្នុងនាមយើងខ្ញុំជា ក្រុមនិសិ្សឆ្នាំទី២ ផ្នែកវិទ្យាសាស្ត្រកុំព្យូទ័រ នៃសកលវិទ្យាល័យនំតុន ជំនាន់ទី២៤ ក្នុងឆ្នាំសិក្សា២០២០-២០២១ យើងខ្ញុំទាំងអស់គ្នា សូមខន្តីអភ័យទោស រាល់កំហុសខុសឆ្គងដែលកើត ឡើងដោយអចេតនាក្នុងការប្រើពាក្យពេចន៍អក្ខរាវិរុទ្ធិ និងមានបញ្ហាបច្ចេកទេសណាមួយក្នុងការរៀបចំឯកសារ នេះឡើងក្រុមនិសិ្សយើងខ្ញុំរីករាយទទួលនូវការរិះគន់និងកែលម្អពីសំណាក់ប្រិយមិត្តអ្នកអាន អ្នកស្តាប់គ្រប់រូប ឬគ្រប់មជ្ឈដ្ឋានទាំងអស់សូមមេត្តាអភ័យទោសដល់ក្រុមនិសិ្សយើងខ្ញុំដោយក្តីអនុគ្រោះ។ ខ្ញុំបាទ-នាងខ្ញុំទាំងអស់គ្នា សូមថ្លែងអំណរអរគុណដ៏ជ្រាលជ្រៅចំពោះលោកគ្រូដែលបានខិតខំ ប្រឹងប្រែងបង្រៀនដល់ខ្ញុំបាទ-នាងខ្ញុំនិងនិសិ្សតាច្រើនទៀតដោយមិនគិតពីភាពឆ្លើយហត់ ក្នុងគោលបំណងចង់ឲ្យនិសិ្សគ្រប់គ្នាទទួលបាននូវចំណេះដឹងដើម្បីយកទៅអនុវត្ត ឬជាផ្លូវក្នុងការប្រកបមុខរបរជាក់ស្តែង។ ជាចុងក្រោយ យើងខ្ញុំទាំងអស់គ្នាសូមជូនពរ ជ័យសឹមង្គល វិបុលសុខ មហាប្រសើរជូនចំពោះ លោកគ្រូអ្នកគ្រូជាទីគោរពដ៏ខ្ពង់ខ្ពស់។ សូមឲ្យទទួលបាននូវសុខភាពល្អ បរិបូរណ៍ កម្លាំងពល់មាំមួន អាយុយឺនយូរ ធ្វើដំណើរទៅទីជិតទីឆ្ងាយ ប្រកបដោយសុខសុវត្ថិភាពជានិច្ច ជួបតែពុទ្ធពរទាំងបួនប្រការ គឺ អាយុ វណ្ណៈ សុខៈ ពលៈ កុំបីឃ្លៀងឃ្លៀតឡើយ។

Contents

I. Introduction	3
II. Problem of Library Management System	5
2.1 Defining of Problem'	
2.2 Solution	
III. Design and Implementation	6
3.1 Drawing flowchart	
3.2 Constructing algorithm (Pseudo-code)	
3.3 Data Structure Design	
A. Structure design in database.	
3.4 Input and Output Design(coding)	
B. Input Design..	
3.5 Output Design	
3.6 Testing	
IV. Discussions	16
V. Conclusions	17
VI. References	17

I. Introduction

សព្វថ្ងៃនេះបច្ចេកវិទ្យាគឺពិតជាមានការរីកចម្រើនជឿនលឿនខ្លាំង។ ក្នុងប្រព័ន្ធគ្រប់គ្រងម៉ាស៊ីនបម្រើសម្រាប់គ្រប់គ្រងប្រព័ន្ធសាលារៀន ត្រូវបានបង្កើតដើម្បីគ្រប់គ្រងសាលារៀន របស់អ្នក។ ប្រព័ន្ធគ្រប់គ្រងសាលារៀនត្រូវបានបង្កើតឡើងដើម្បីជួយបុគ្គលិកធ្វើការងារតិច ជួយសិស្ស និងគ្រូសន្សំពេលវេលា កាត់បន្ថយការចំណាយ និងកែលម្អសុវត្ថិភាពទិន្នន័យ។ វាអាចជួយអ្នកគ្រប់គ្រងសេវាកម្មក្នុងការគ្រប់គ្រងប្រព័ន្ធសេវាកម្មដែលអាចប្រើបានដូចជាការបញ្ជូនព័ត៌មានរបស់សិស្សទៅកាន់អ្នកគ្រប់គ្រងសេវាកម្មព័ត៌មានរបស់សិស្សនិងការគ្រប់គ្រងប្រព័ន្ធសេវាកម្ម។ ការប្រើប្រាស់នៃក្នុងប្រព័ន្ធគ្រប់គ្រងនេះមានប្រភេទច្រើនដែលអាចគ្រប់គ្រងព័ត៌មានសិស្ស។

ប្រព័ន្ធគ្រប់គ្រងសាលារៀនគឺជាប្រព័ន្ធដែលត្រូវបានប្រើដើម្បីគ្រប់គ្រងព័ត៌មានអំពីអ្នកជំនួយនិងសិស្សដែលបានចុះឈ្មោះចូលរៀននៅក្នុងសាលារៀននីមួយៗ។ វាមានគុណភាពដូចជា៖ គ្រប់គ្រងបញ្ជីអ្នកជំនួយនិងសិស្សសាលាដែលបានចុះឈ្មោះចូលរៀនមិនថាសិស្សថ្មីឬចាស់ឡើយវាគឺត្រូវបានគ្រប់គ្រងនិងរក្សាទុកព័ត៌មានរបស់សិស្សដែលបានចុះឈ្មោះចូលរៀន, គ្រប់គ្រងការបញ្ជាទិញនិងការបង់ប្រាក់, គ្រប់គ្រងការពិនិត្យមើលសិស្សដែលចូលរៀន គ្រប់គ្រងការបង្កើតរូបភាពសិស្ស និងគ្រប់គ្រងការបញ្ចូលព័ត៌មានសិស្ស ទៅក្នុងប្រព័ន្ធគ្រប់គ្រងសាលារៀន ។ ក្នុងគោលបំណងនេះគឺជាការប្រើប្រាស់វិធីសាស្ត្រក្នុងការបង្កើតកិច្ចសន្យាដែលអាចគ្រប់គ្រងព័ត៌មានអំពីសិស្ស ដែលបានចុះឈ្មោះចូលរៀន ។ វាត្រូវបានបង្កើតជាមួយនិងវិធីសាស្ត្រក្នុងការប្រើប្រាស់ប្រព័ន្ធទិន្នន័យដើម្បីគ្រប់គ្រងព័ត៌មានអំពីសិស្ស ។ វាអាចប្រើបានដោយការបញ្ជូនព័ត៌មានអំពីសិស្ស ដែលបានចូលមកក្នុងសាលារៀន និងការបង្ហាញព័ត៌មានទាំងអស់នៅក្នុងប្រព័ន្ធគ្រប់គ្រងសាលារៀន ។ វាត្រូវបានបង្កើតឡើងដើម្បីស្វ័យប្រវត្តិការ និងសម្រួលដល់ដំណើរការរដ្ឋបាល, រួមទាំងការចុះឈ្មោះសិស្ស, ការតាមដានវត្តមាន និងការគ្រប់គ្រងនៃពិន្ទុសិស្ស ហើយនិងធ្វើការ ការគ្រប់គ្រងទិន្នន័យ: ប្រព័ន្ធ SMS អនុញ្ញាតឱ្យមានការផ្ទុកទិន្នន័យ, ទាញយកនិងគ្រប់គ្រងទិន្នន័យសិស្សនិងបុគ្គលិកយ៉ាងមានប្រសិទ្ធភាព, ជួយក្នុងការរក្សាទុកនិងថែរក្សាការណ៍។ នាពេលបច្ចុប្បន្ននេះ យើងឃើញថាពិភពលោកទាំងមូលមានការអភិវឌ្ឍន៍និងភាពរីកចម្រើនកាន់តែជឿនលឿនទៅមុខជាបន្តបន្ទាប់យ៉ាងឆាប់រហ័ស ប្រទេសខ្លះក៏មានភាពជឿនលឿនមុន រី ឯប្រទេសខ្លះទៀតក៏កំពុងតែអភិវឌ្ឍន៍។ ប្រទេសដែលមានភាពជឿនលឿនមុន គេមិនបង្អង់យូរ ឬឈប់ ក្នុងការអភិវឌ្ឍន៍ប្រទេសរបស់គេឡើយ យើងក្រឡេកមកមើល ប្រទេសកំពុងអភិវឌ្ឍន៍វិញក៏មិនចាញ់ និងប្រទេសដែលជឿនលឿនប៉ុន្មានទេ រាជរដ្ឋាភិបាលគ្រប់ប្រទេសទាំងមូលបានខិតខំប្រឹងប្រែងធ្វើយ៉ាងណាឱ្យប្រទេសរបស់ខ្លួនរីកចម្រើនលូតលាស់ឆ្ពោះជាមុខជានិច្ច ។

ជាពិសេសជាងនេះទៅទៀតខាងបច្ចេកវិទ្យា យើងបានដឹងរួចមកហើយថាគ្រប់បណ្តា ប្រទេសទាំងអស់សុទ្ធសឹងតែប្រើប្រាស់ទៅលើប្រព័ន្ធ Digital System, Computer System, Technology ក្នុងការធ្វើផុតកិច្ច រួមទាំងការសិក្សាស្រាវជ្រាវទៅលើការសាងសង់ ឬការបង្កើតកម្មវិធី ដោយមានការសរសេរកូដ (code) ចងក្រងជាប្រព័ន្ធ Program ក្នុងការបង្កើត ។ ការរីកចម្រើនខាងប ច្ចេកវិទ្យា បានធ្វើឱ្យនិស្សិតគ្រប់រូបមានចំណេះដឹងកាន់តែខ្លាំងពូកែដែលអាចធ្វើវិនិយោគ ឬជំនួញ កាន់តែប្រសើរឡើង ហើយធ្វើឱ្យប្រទេសកាន់តែរីកចម្រើនជាមុនកាន់តែឆាប់រហ័ស ។

ក្នុងនាមខ្ញុំបាទ-នាងខ្ញុំ ដែលជាក្រុមនិស្សិតរៀននៅសាកលវិទ្យាល័យ នំតុន ឆ្នាំទី២ នៃ ផ្នែកវិទ្យាសាស្ត្រកុំព្យូទ័រ។ ទន្ទឹមនឹងនេះផងដែរ វិទ្យាសាស្ត្រកុំព្យូទ័រ មានមុខវិជ្ជាជាច្រើនក្នុងការសិក្សា ដោយឡែកចំពោះមុខវិជ្ជាដែលកំពុងតែរៀនមាន ចំណងជើងថា Data Structure and Algorithms ។ នៅក្នុងមេរៀន Data Structure បានផ្តល់នូវចំណេះដឹងជាច្រើន ឬ វិធីសាស្ត្រជាច្រើនក្នុងការបង្កើត Program ។ តើ Program ជាអ្វី? តើការបង្កើត Program មានប៉ុន្មានដំណាក់កាល? អ្វី ខ្លះ?

Program: គឺជាសំនុំមួយនៃដំណោះស្រាយបញ្ហាក្នុងប្រព័ន្ធកុំព្យូទ័រ ដែលក្រោយពីអនុវត្តន៍ ទៅតាមដំណើរការ Process របស់ម៉ាស៊ីនយើងនឹងទទួលបានលទ្ធផលដែលយើងចង់បាន។ គេប្រើ ប្រាស់ការសរសេរកម្មវិធី (បង្កើតកម្មវិធី) ក្នុងគោលបំណងផ្សេងៗគ្នា ដូចជាការបង្កើតកម្មវិធីទូរស័ព្ទ ការ ផលិតភាពយន្ត ឬក៏បង្កើតមនុស្សយន្ត។ ម្យ៉ាងវិញទៀតក្នុង ការសរសេរកម្មវិធី(បង្កើតកម្មវិធី) នីមួយៗ មានច្រើនភាសាចូលរួមចងក្រងជាប្រព័ន្ធទើបក្លាយជា Program មួយ ហើយ មានភាសាដែលចូលរួម មានដូចជា C/C++ Program Language, Java Script, Java, Python ជាដើម ។ យើង ឃើញថា មាន Programming Languages ជាច្រើនក្នុងការបង្កើតកម្មវិធី ព្រោះកម្មវិធីនីមួយៗមិនអាចបង្កើត ដោយ ប្រើតែមួយភាសាក្នុងការសរសេរកូដចងក្រងជាប្រព័ន្ធដើម្បីបង្កើតកម្មវិធីបានឡើយ។

លើសពីនេះ ដើម្បីឲ្យការអនុវត្តផ្សាភ្ជាប់ទៅនឹងទ្រឹស្តីក្រុមនិស្សិតយើងបានខ្ញុំធ្វើការពាំ នាំឲ្យស្គាល់អំពីប្រធានបទនៃ ការអនុវត្តជាក់ស្តែងទៅលើកិច្ចការស្រាវជ្រាវជាមួយ Data Structure និង Algorithms នេះ ដែលមានប្រធានបទចំណង ជើងថា Library Management System ។ ដើម្បី ឲ្យយល់កាន់តែច្បាស់អំពីប្រព័ន្ធគ្រប់គ្រងបណ្ណាល័យយើងត្រូវចោទជា សំណួរសួរថា តើអ្វីទៅជា Data Structure? ចុះចំណែកឯ Algorithms ជាអ្វី? តើ Data Structure និង Algorithms មានតួនាទីនិង ទំនាក់ទំនងយ៉ាងដូចម្តេច? តើការសិក្សាមេរៀន Data Structure and Algorithms មានគោល បំណង អ្វីខ្លះ? តើការបង្កើត Library Management System មានសារសំខាន់អ្វីខ្លះដល់ការងារ?

Data Structures: គឺជាការរៀបចំ Data របស់អ្នកសរសេរ Program ទៅតាម Code របស់ម៉ាស៊ីននិង រាល់ Data ដែលយើងឲ្យរួចទាញយកមកប្រើប្រាស់វិញក្នុងគោលបំណងអ្វីមួយ។ ចំណែកឯ Algorithms: គឺជាជំហាននៃការ ដោះស្រាយបញ្ហាទៅតាមដំណាក់កាលៗ ជា Flowchart និងម្យ៉ាងទៀត Algorithms គឺជាខ្សែរ Statement ដ៏ច្បាស់សា ស់ដែលក្រោយពីការកំណត់ជាក់លាក់ រហូតយើងទទួលបានលទ្ធផលដែលយើងចង់បាន។ ហើយ Algorithms មានតួនាទី ធ្វើការ Process របស់បញ្ហា រីឯ Data Structuresមានតួនាទីធ្វើការ Process ដោយជាមួយ Machine ដើម្បីនាំដល់ លទ្ធផលរបស់ចំណោទបញ្ហា ។ ការសិក្សាមុខវិជ្ជា Data Structure and Algorithms មានគោល បំណងសម្រាប់បណ្តុះប ណ្តាលអ្នកបច្ចេកទេស ព័ត៌មានវិទ្យា ពីព្រោះមុខវិជ្ជានេះអាចជួយឲ្យនិស្សិត ស្វែងយល់អំពី Data Structure and Algorithms ដែលមានទំនាក់ទំនងព្រមទាំងជួយបង្កើននូវ ចំណេះដឹងអំពី បច្ចេកទេសសរសេរកម្មវិធី អំពីវិធីសាស្ត្រដោះ ស្រាយរាល់បញ្ហា និងជួយឲ្យនិស្សិតមាន សមត្ថភាពកាន់តែស៊ីជម្រៅថែមទៀតផងទៅលើមុខជំនាញដូចជា Database, Analysis System, Programming Languages ។ ចំពោះការបង្កើត Library Management System មាន គោល បំណងទៅលើការគ្រប់គ្រងប្រព័ន្ធនៅក្នុងបណ្ណាល័យដើម្បីផ្តល់ភាពងាយស្រួលដល់បុគ្គលិក នៅពេល និស្សិតចូលទៅ អានសៀវភៅលើកដំបូង ក្រុមការងារយើងខ្ញុំមានការផ្តល់ជូនជាក្រដាសដើម្បីចុះឈ្មោះ ជាសមាជិក ហើយនៅក្នុងក្រដាស

មានការបំពេញព័ត៌មានផ្ទាល់ខ្លួនឲ្យមានភាពងាយស្រួលនៅពេល និស្សិតចូលទៅម្តងទៀតដោយមិនចាំបាច់បំពេញព័ត៌មានម្តងទៀត ដោយគ្រាន់តែប្រាប់លេខសម្គាល់ ខ្លួន(Student ID) ជាការស្រេច។ បន្ទាប់ពីនេះក្រុមការងារយើងខ្ញុំបានរៀបចំសៀវភៅទៅតាមលំដាប់ នៃប្រភេទសៀវភៅនីមួយៗ ងាយស្រួលដល់និស្សិតក្នុងការស្វែងរក មិនតែប៉ុណ្ណោះ នៅក្នុងបណ្ណាល័យ មានកន្លែងសម្រាប់ដាក់សម្ភារៈ (កាបូប របស់របរផ្សេងៗ) ហើយតម្រូវឲ្យយកសម្ភារៈមានតម្លៃទៅតាម ខ្លួនបានដោយមិនពិបាកភ័យខ្លាចបាត់បង់នៅពេលចូលអានសៀវភៅ ឬធ្វើកិច្ចការស្រាវជ្រាវផ្សេងៗ។

លើសពីនេះទៅទៀត ក្រុមការងារយើងខ្ញុំបានដាក់កុំព្យូទ័រ និងបានបំពាក់នូវប្រព័ន្ធអ៊ីនធឺណិត ដើម្បី មានភាពងាយស្រួលដល់និស្សិតដែលគ្មានកុំព្យូទ័របានធ្វើកិច្ចការស្រាវជ្រាវនូវអ្វីដែលពួកគេចង់ធ្វើ ហើយក៏ជា ការកាត់បន្ថយថវិការបស់និស្សិតដែលចំណាយទៅលើប្រព័ន្ធអ៊ីនធឺណិតខាងក្រៅថែមទៀតផង។ ម្យ៉ាងវិញ ទៀត និស្សិតក៏អាចធ្វើការខ្លីសៀវភៅពីបណ្ណាល័យយកទៅអាននៅផ្ទះបាន ហើយខាងយើងខ្ញុំបានកំណត់ ពេលវេលាច្បាស់លាស់នៅពេលខ្លី និងពេលសងសៀវភៅប្រសិនបើនិស្សិតមានការសង និងការខូចខាត ឬបាត់បង់ ខាងយើងខ្ញុំតម្រូវឲ្យសងជាសៀវភៅ ឬពិន័យជាសាច់ប្រាក់(លុយ) នេះជាគោលការណ៍ខាងយើងខ្ញុំ តម្រូវឲ្យគោរព ប្រសិនបើមិនធ្វើដូចនេះទេធ្វើឲ្យបណ្ណាល័យគ្មានគុណភាព ។

II. Problem of Library Management System

សព្វថ្ងៃនេះ ការធ្វើការដោយដៃមនុស្សគឺពិតជាមានការយឺតយ៉ាវណាស់ ជួនកាលអាចធ្វើអោយមានការយល់ច្រឡំរឺក៏ខុសផងដែរ។ ជាពិសេសសាលារៀនគឺតែងតែមានសិស្សរឺនិស្សិតជាច្រើនដែលបានចុះឈ្មោះចូលដើម្បីសិក្សា វាគឺសុទ្ធសឹងតែការងាររបស់បុគ្គលិកនៃសាលានីមួយៗ ទាំងនេះហើយគឺធ្វើអោយមនុស្សយើងមានការពិបាកក្នុងការគ្រប់គ្រង និងធ្វើការយឺតយ៉ាវ។ ហេតុដូច្នេះហើយ ទើបធ្វើឲ្យក្រុមនិស្សិតខ្ញុំជ្រើសរើសយកប្រធានបទ School management system មកសិក្សាស្រាវជ្រាវអំពីបញ្ហា និងដំណោះស្រាយ។ដើម្បីចម្លងជាងនេះទៅទៀត ក្រុមនិស្សិតខ្ញុំនឹងពន្យល់ចំណុចសំខាន់ៗដូចខាងក្រោម៖

2.1. Defining of Problem

នៅក្នុងនេះផងដែរ ក្រុមនិស្សិតយើងខ្ញុំបានសិក្សាស្រាវជ្រាវជាច្រើនមកបានជួបបញ្ហាមួយចំនួនដូច

ខាងក្រោម៖

- ការស្រង់យកព័ត៌មានផ្ទាល់ខ្លួនរបស់និស្សិតមានភាពមិនច្បាស់លាស់
- មានភាពលំបាកក្នុងការកត់ត្រាឈ្មោះរបស់និស្សិតនៅពេលនិស្សិតចុះឈ្មោះចូលរៀន
- មានភាពលំបាកពេលនិស្សិតធ្វើការរក្សាព័ត៌មានរបស់ខ្លួន ថែមទាំងចំណាយពេលវេលា
- គ្រប់គ្រងបញ្ជីឈ្មោះរបស់និស្សិតមិនបានគ្រប់ជ្រុងជ្រោយ
- ពិបាកគ្រប់គ្រងទិន្នន័យរបស់និស្សិតព្រោះសរសេរដោយដៃ

2.2. Solution

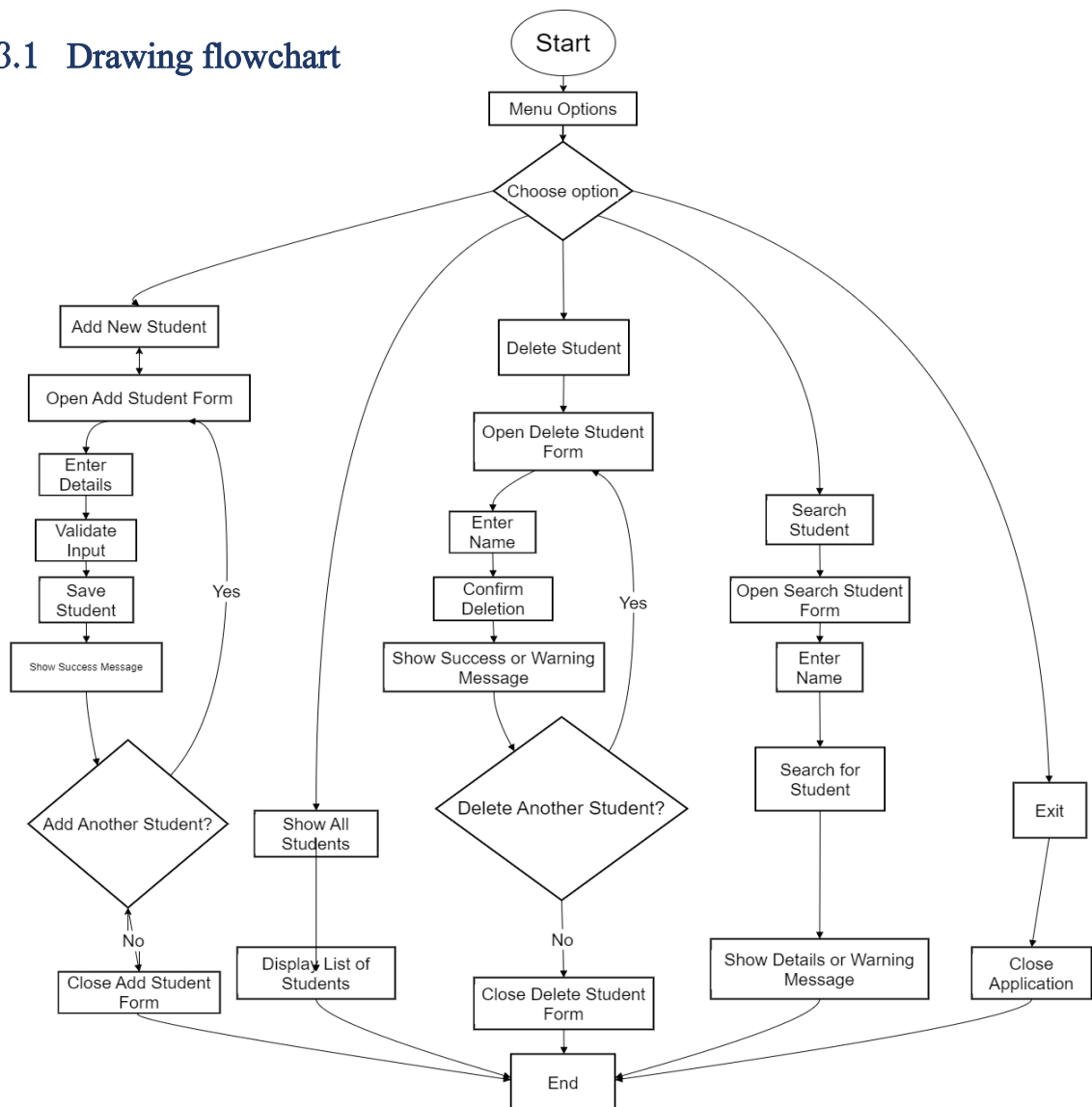
ក្រោយពេលបានសិក្សាស្រាវជ្រាវអំពីបញ្ហាចម្បង ក្រុមនិស្សិតយើងខ្ញុំបានខិតខំប្រឹងប្រែងស្វែងរក នៅដំណោះស្រាយ បញ្ហាមកបង្ហាញដូចខាងក្រោម៖

- ធ្វើការបង្កើតប្រព័ន្ធគ្រប់គ្រងមួយនៅក្នុងកុំព្យូទ័រដើម្បីមានភាពងាយស្រួលដល់អ្នកគ្រប់គ្រង
- ចំណេញពេលវេលាក្នុងការចុះឈ្មោះសិស្សនិង ស្វែងរកព័ត៌មានរបស់និមួយៗ
- ធ្វើការដោយមានភាពច្បាស់លាស់គ្មានភាពយល់ច្រឡំរឺមិនរញ្ជ្រោច

III. Design and Implementation

យោងតាមការសិក្សាអំពីបញ្ហា និងដំណោះស្រាយខាងលើរួចមក ធ្វើអោយក្រុមនិស្សិតយើងខ្ញុំមានការយល់ដឹង បន្ថែមទាក់ទងទៅនឹងការដោះស្រាយបញ្ហាជាច្រើនវិធី និងមានការអនុវត្តផ្ទាល់នូវចំណុចសំខាន់ៗដូចខាងក្រោម៖

3.1 Drawing flowchart



បកស្រាយ៖

- Start: នេះជាកន្លែងដែលកម្មវិធីចាប់ផ្តើម។
- Display Main Menu: កម្មវិធីបង្ហាញម៉ឺនុយដល់អ្នកប្រើប្រាស់ដែលមានជម្រើសផ្សេងៗ។
- Input: User choice: អ្នកប្រើប្រាស់ជ្រើសរើសជម្រើសមួយ (ដូចជាការបន្ថែមសិស្ស បង្ហាញសិស្សទាំងអស់។ល។)

អាស្រ័យលើជម្រើសរបស់អ្នកប្រើ៖

+ Add student: កម្មវិធីជំរុញឱ្យអ្នកប្រើប្រាស់បញ្ចូលព័ត៌មានលម្អិតសិស្ស (ដូចជាឈ្មោះ អាយុ ។ល។) +

Show all Students ៖ កម្មវិធីបង្ហាញបញ្ជីសិស្សទាំងអស់។

+ Delete student: អ្នកប្រើប្រាស់ផ្តល់ឈ្មោះសិស្សដែលត្រូវលុប។

+ Search Student: អ្នកប្រើប្រាស់បញ្ចូលឈ្មោះសិស្ស និងទទួលបានព័ត៌មានលម្អិតអំពីសិស្សនោះ។

+ Exit: គឺជាការបញ្ចប់កម្មវិធីឬចាកចេញពីកម្មវិធី

3.2. Constructing algorithm (Pseudo-code)

Pseudocode វាគឺជាពាក្យដែលត្រូវបានដោះស្រាយនៅក្នុងផ្នែកដែលទាក់ទងទៅនឹងកូដដោះស្រាយនិងការសរសេរកម្មវិធីកុំព្យូទ័រ។ វាគឺជាវិធីសាស្ត្រមួយដែលអាចអោយអ្នកសរសេរកម្មវិធីបង្កើតការអភិវឌ្ឍនៃកូដដោះស្រាយបានយ៉ាងងាយស្រួល។ ដូចដែលឈ្មោះបានបញ្ជាក់វាគឺជា តំណាងក្នុងដែលអាចងាយ យល់សូម្បីតែអ្នកដែលមានសញ្ញាណខ្លះនៃការសរសេរកម្មវិធីនៅកម្រិតបឋម។

```

START

SHOW main menu

WHILE application is running
    userChoice = getUserChoice()

    IF userChoice is "Add" THEN
        INPUT student details
        VALIDATE and ADD student
        ASK to add more or exit

    ELSE IF userChoice is "Show" THEN
        DISPLAY all students

    ELSE IF userChoice is "Delete" THEN
        INPUT name to delete
        DELETE student if found
        ASK to delete more or exit

    ELSE IF userChoice is "Search" THEN
        INPUT name to search
        SHOW student details if found

    ELSE IF userChoice is "Exit" THEN
        CLOSE application

ENDWHILE

END
    
```


1.Add Student:

- User inputs student details.
- If valid, student is added and user is asked if they want to add another.

2.Show Students:

- Displays a list of all students or a message if no students are available.

3.Delete Student:

- User inputs the name of the student to delete.
- If found, student is deleted and user is asked if they want to delete another.

4.Search Student:

- User inputs the name of the student to search.
- If found, student details are shown; otherwise, a not found message is displayed.

4.Exit:

- Close the application.

ក្រោយពីបានបង្ហាញអំពី Pseudo-code ខាងលើរួច យើងអាចបំប្លែងទៅជា code ដើម្បី Run នៅលើកុំព្យូទ័របានមានដូចខាងក្រោម៖

ការសរសេរកូដគ្រប់គ្រងប្រព័ន្ធសាលារៀននេះគឺសម្រេចប្រើប្រាស់ក្រាម Python ដែលជាភាសាសរសេរកម្មវិធីកម្រិតខ្ពស់ដែលគេស្គាល់សម្រាប់ភាពសាមញ្ញ និងអាចអានបាន ដែលធ្វើឱ្យវាស័ក្តិសមសម្រាប់ការបង្កើតកម្មវិធីដូចជាប្រព័ន្ធគ្រប់គ្រងសាលារៀន។ ដើម្បីបង្កើតប្រព័ន្ធគ្រប់គ្រងនេះបានគឺយើងធ្វើការបង្កើត Class និង Function ចំនួនចំនួន។

Class គឺជាប្លង់មេ ឬគំរូសម្រាប់បង្កើតវត្ថុ។ វាកំណត់លក្ខណៈសម្បត្តិ (គុណលក្ខណៈ) និងអាកប្បកិរិយា (វិធីសាស្ត្រ) ដែលវត្ថុនៃ Class នោះនឹងមាន។ Class នេះរួមបញ្ចូលគុណលក្ខណៈដូចជា ឈ្មោះ អាយុ ភេទ អាសយដ្ឋាន ទូរស័ព្ទ និងសាលារៀនសម្រាប់សិស្សម្នាក់ៗ។

```
import tkinter as tk
from tkinter import messagebox

class Student:
    """Class representing a student with basic attributes."""

    def __init__(self, name, age, gender, address, phone, school):
        self.name = name
```

```
        self.age = age
        self.gender = gender
        self.phone = phone
        self.address = address
        self.school = school

class StudentManagementApp:
    """Class representing the student management application."""

    def __init__(self, root):
        self.root = root
        self.root.title("Student Management System")
        self.students = []

        # Initialize the user interface
        self.setup_gui()

    def setup_gui(self):
        """Setup the main menu and buttons."""
        # Create a frame to hold the menu buttons
        frame = tk.Frame(self.root, padx=20, pady=20)
        frame.pack()

        # Add buttons for each operation
        tk.Button(frame, text="Add New Student", command=self.add_student,
width=30, height=2).pack(pady=5)
        tk.Button(frame, text="Show All Students", command=self.show_students,
width=30, height=2).pack(pady=5)
        tk.Button(frame, text="Delete Student", command=self.delete_student,
width=30, height=2).pack(pady=5)
        tk.Button(frame, text="Search Student", command=self.search_student,
width=30, height=2).pack(pady=5)
        tk.Button(frame, text="Exit", command=self.root.quit, width=30, height=2,
bg="red", fg="white").pack(pady=5)

    def add_student(self):
        """Open a window to add a new student."""
        def save_student():
            name = entry_name.get()
            age = entry_age.get()
            gender = entry_gender.get()
            phone = entry_phone.get()
            address = entry_address.get()
            school = entry_school.get()

            if not name or not age or not gender or not phone or not address or
not school:
```

```
        messagebox.showwarning("Input Error", "All fields are required.")
        return

    try:
        age = int(age)
    except ValueError:
        messagebox.showwarning("Input Error", "Age must be a number.")
        return

    student = Student(name, age, gender, address, phone, school)
    self.students.append(student)
    messagebox.showinfo("Success", f"{name} has been added.")

    # Ask the user if they want to continue adding students
    if messagebox.askyesno("Continue?", "Do you want to add another
student?"):

        # Clear the entries for new input
        entry_name.delete(0, tk.END)
        entry_age.delete(0, tk.END)
        entry_gender.delete(0, tk.END)
        entry_phone.delete(0, tk.END)
        entry_address.delete(0, tk.END)
        entry_school.delete(0, tk.END)
    else:
        add_window.destroy()

    # Create a new window for adding a student
    add_window = tk.Toplevel(self.root)
    add_window.title("Add New Student")

    # Layout for the form
    labels = ["Name", "Age", "Gender", "Phone", "Address", "School"]
    entries = {}

    for label in labels:
        tk.Label(add_window, text=f"{label}:").pack(anchor="w")
        entry = tk.Entry(add_window)
        entry.pack(fill="x", padx=10, pady=5)
        entries[label.lower()] = entry

    entry_name = entries['name']
    entry_age = entries['age']
    entry_gender = entries['gender']
    entry_phone = entries['phone']
    entry_address = entries['address']
    entry_school = entries['school']
```

```

        tk.Button(add_window, text="Save", command=save_student,
width=20).pack(pady=10)

    def delete_student(self):
        """Open a window to delete a student."""
        def confirm_delete():
            name = entry_name.get()
            for i, student in enumerate(self.students):
                if student.name.lower() == name.lower():
                    del self.students[i]
                    messagebox.showinfo("Success", f"Deleted student {name}")

                    # Ask the user if they want to continue deleting students
                    if messagebox.askyesno("Continue?", "Do you want to delete
another student?"):
                        # Clear the entry for new input
                        entry_name.delete(0, tk.END)
                    else:
                        delete_window.destroy()
                    return
            messagebox.showwarning("Not Found", f"{name} not found")

        delete_window = tk.Toplevel(self.root)
        delete_window.title("Delete Student")

        tk.Label(delete_window, text="Enter the name of the student to
delete:").pack(anchor="w")
        entry_name = tk.Entry(delete_window)
        entry_name.pack(fill="x", padx=10, pady=5)
        tk.Button(delete_window, text="Delete", command=confirm_delete,
width=20).pack(pady=10)

    def show_students(self):
        """Display all students in a new window."""
        show_window = tk.Toplevel(self.root)
        show_window.title("List of Students")

        if not self.students:
            tk.Label(show_window, text="No students available.").pack(pady=10)
        else:
            for i, student in enumerate(self.students):
                tk.Label(show_window, text=f"Student {i+1}:
{student.name}").pack(anchor="w")

    def search_student(self):
        """Open a window to search for a student."""
        def perform_search():
            name = entry_name.get()

```

```

        for student in self.students:
            if student.name.lower() == name.lower():
                messagebox.showinfo("Student Found",
                                    f"Name: {student.name}\n"
                                    f"Age: {student.age}\n"
                                    f"Gender: {student.gender}\n"
                                    f"Phone: {student.phone}\n"
                                    f"Address: {student.address}\n"
                                    f"School: {student.school}")
                search_window.destroy()
                return
            messagebox.showwarning("Not Found", f"{name} not found")

        search_window = tk.Toplevel(self.root)
        search_window.title("Search Student")

        tk.Label(search_window, text="Enter the name of the student to
search:").pack(anchor="w")
        entry_name = tk.Entry(search_window)
        entry_name.pack(fill="x", padx=10, pady=5)
        tk.Button(search_window, text="Search", command=perform_search,
width=20).pack(pady=10)

if __name__ == "__main__":
    root = tk.Tk()
    app = StudentManagementApp(root)
    root.mainloop()

```

3.3. Data Structure Design

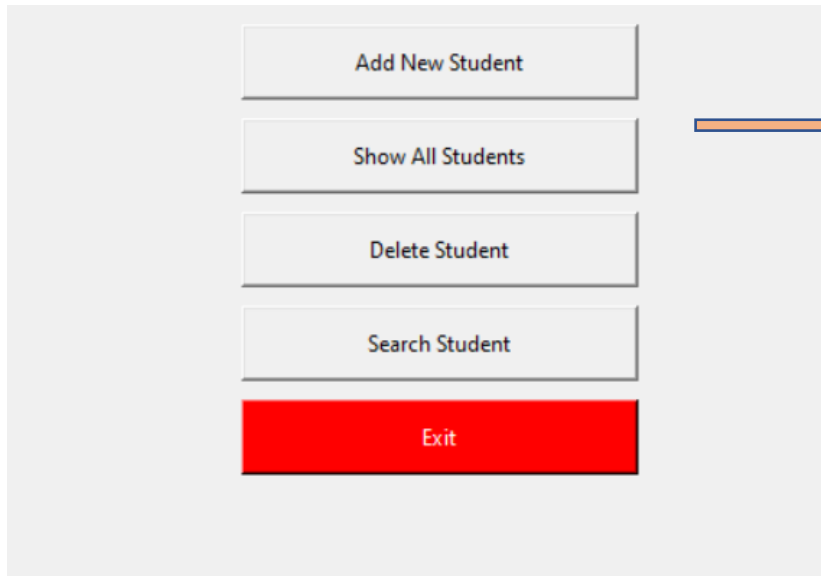
A. Structure design in database

No database

students	
id 	INTEGER
name	VARCHAR
age	INTEGER
gender	VARCHAR
phone	VARCHAR
address	TEXT
school	VARCHAR

3.4 Input and Output Design(coding)

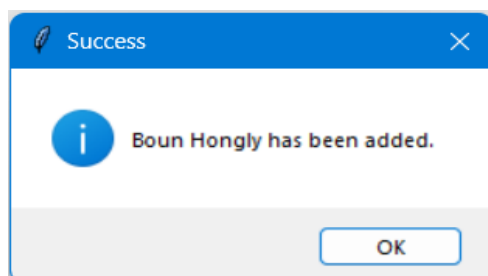
A. Input Design



នៅពេលដែលអ្នកដំណើរការកូដវានឹងចេញម៉ឺនុយសម្រាប់អោយជ្រើសរើស។ If you chose button add student

Input information student

បន្ទាប់អ្នកប្រើប្រាស់បានវាយព័ត៌មានលម្អិតរបស់និស្សិតរួចរាល់ហើយ
ឈ្មោះនិងព័ត៌មានរបស់និស្សិតនេះត្រូវបញ្ចូលក្នុងបញ្ជីនៃសាលារៀន។



—

□

×

Name:

Boun Hongly

Age:

19

Gender:

Male

Phone:

043435343

Address:

Phnom Phenh

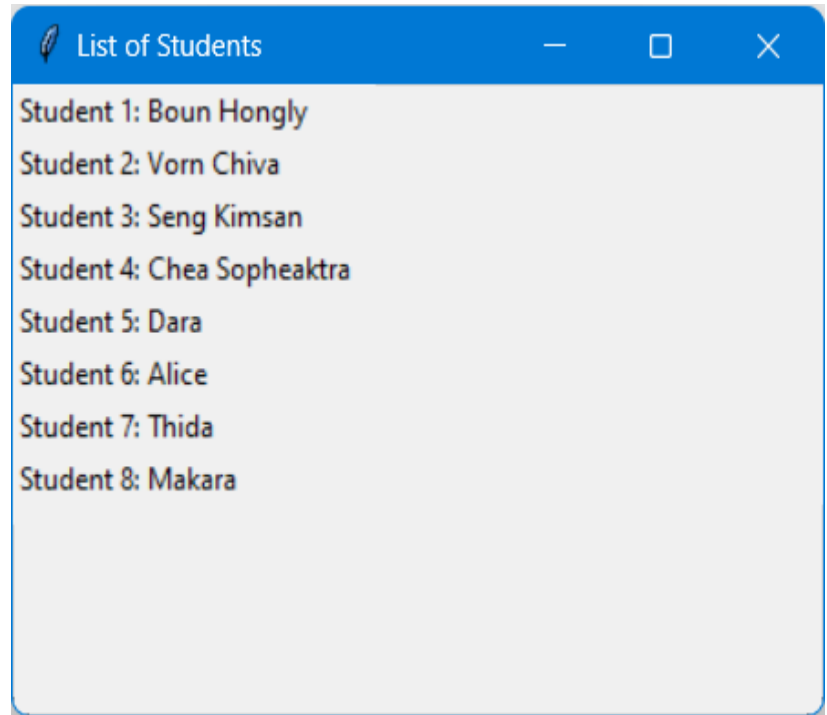
School:

Khmer high school

Save

3.5 Output Design

ក្រោយពេលវាយបញ្ចូលព័ត៌មានសិស្សរួចរាល់មក យើងឃើញថាមានបញ្ជីរាយឈ្មោះរបស់និស្សិតដែលបានចុះឈ្មោះចូលរៀននៅសាលា។ ហើយ User អាចតាមដានព័ត៌មានរបស់និស្សិតតាមរយៈប្រព័ន្ធកុំព្យូទ័រ នេះនឹងផ្តល់របស់និស្សិតដែលបានចុះឈ្មោះចូលរៀន៖



3.6 Testing

ក្រោយពីបានសិក្សាអំពីបញ្ហា ដំណោះស្រាយនិងចំណុចសំខាន់ៗជាច្រើនទៀតដូចជា៖ Flowchart សរសេរ Code Pseudo-code ដែលជាចំណុចសំខាន់ក្នុងមក យើងមានការធ្វើ Testing ដែលមានការអនុវត្ត ផ្ទាល់។

ខាងក្រោមនេះគឺជាប្រភេទនៃការប្រើប្រាស់ប្រព័ន្ធគ្រប់គ្រងនៃកូដនេះបានយ៉ាងរលូន៖

១. បន្ថែមសិស្សថ្មី៖

- នៅពេលអ្នកដំណើរការកូដ អ្នកនឹងឃើញម៉ឺនុយដែលមានជម្រើស។
- ប្រសិនបើ user ជ្រើសរើស Add Student។
- បញ្ចូលឈ្មោះសិស្ស អាយុ ភេទ លេខទូរស័ព្ទ អាសយដ្ឋាន និងឈ្មោះសាលាចាស់ ដូចដែលបានសួរ។
- កម្មវិធីនឹងបន្ថែមសិស្សនេះទៅក្នុងបញ្ជីសិស្ស។

២. បង្ហាញសិស្សទាំងអស់៖

- ជ្រើសរើសជម្រើសទី ២ ពីម៉ឺនុយ ដើម្បីមើលបញ្ជីឈ្មោះសិស្សទាំងអស់។

៣. ការលុបសិស្ស៖

- ជ្រើសរើសជម្រើសទី ៣ ពីម៉ឺនុយដើម្បីលុបសិស្ស។
- បញ្ចូលឈ្មោះសិស្សដែលអ្នកចង់លុបនៅពេលត្រូវបានសួរ។
- កម្មវិធីនឹងដកសិស្សនោះចេញពីបញ្ជី ប្រសិនបើពួកគេមាន។

៤. ស្វែងរកសិស្ស៖

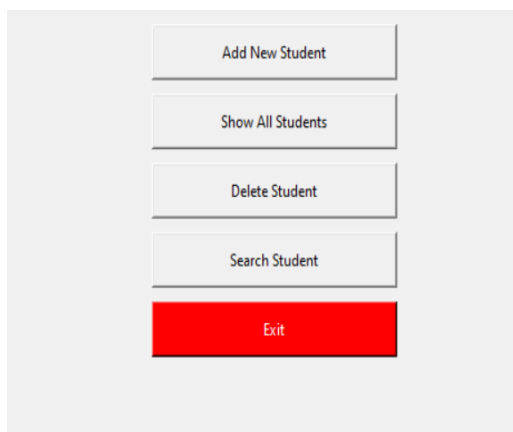
- ជ្រើសរើសជម្រើសទី ៤ ពីម៉ឺនុយដើម្បីស្វែងរកសិស្ស។
- បញ្ចូលឈ្មោះសិស្សដែលអ្នកចង់ស្វែងរក។
- ប្រសិនបើសិស្សមាននៅក្នុងបញ្ជី ព័ត៌មានលម្អិតរបស់ពួកគេ (ឈ្មោះ អាយុ ភេទ លេខទូរស័ព្ទ អាស័យដ្ឋាន

និងសាលារៀន) នឹងត្រូវបានបង្ហាញ។

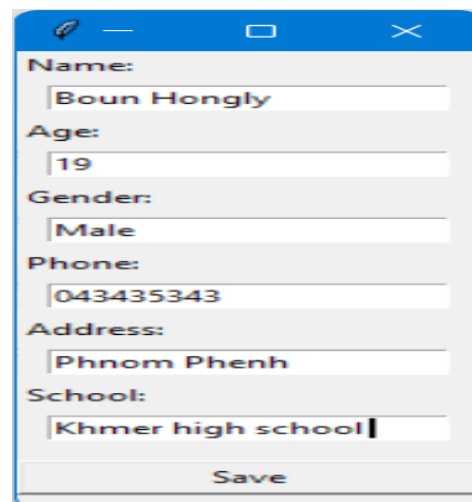
៥. ការចាកចេញពីកម្មវិធី៖

- ជ្រើសរើសជម្រើស ៥ ពីម៉ឺនុយដើម្បីចេញពីកម្មវិធី។

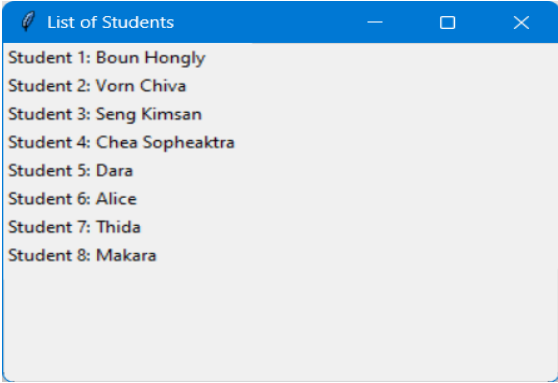
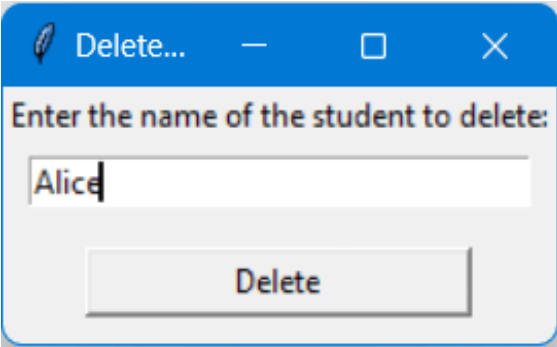
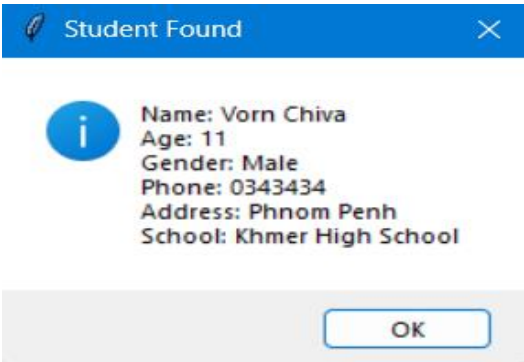
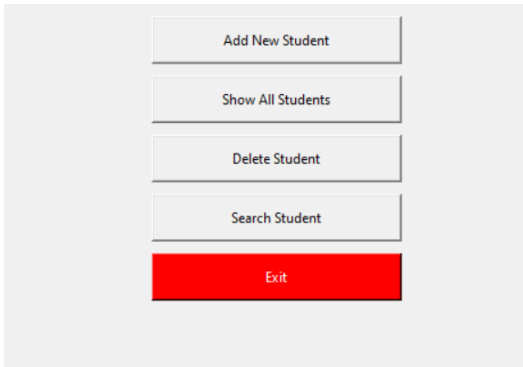
លទ្ធផលដែលទទួលបាន ក្រោយពេលបញ្ចប់ការសរសេរកូដ និងធ្វើការសាកល្បង ទទួលបានកម្មវិធីមួយត្រឹមត្រូវដូចមានបង្ហាញនៅក្នុងខាងក្រោម៖



(១) នេះជាលទ្ធផលនៃការ Run កូដ



(១) ជ្រើសរើស Add Student សម្រាប់បំពេញព័ត៌មានសិស្ស ចូលទៅក្នុងប្រព័ន្ធ

 <p>(២) ជ្រើសរើស Show Students ដើម្បីមើលបញ្ជីឈ្មោះសិស្សដែលបានបំពេញ ព័ត៌មានរួចរាល់</p>	 <p>(៣) ជ្រើសរើសប្រអប់ Delete Student ដើម្បីលុបឈ្មោះសិស្សដែលយើងលុបដោយគ្រាន់តែធ្វើការវាយបញ្ចូលឈ្មោះ</p>
 <p>(៤) ជ្រើសរើស Search Student សម្រាប់ស្វែងរកសិស្សដើម្បីពិនិត្យមើលព័ត៌មានតាមរយៈការវាយបញ្ចូលឈ្មោះ</p>	 <p>(៥) ជ្រើសរើស Exit សម្រាប់ពេលដែលយើងចង់ចាកចេញពីកម្មវិធីឬក៏បញ្ចប់</p>

ក្នុងប្រព័ន្ធគ្រប់គ្រងនេះតាមរយៈការដាក់ឱ្យដំណើរការទទួលបានលទ្ធផលមួយដូចការគ្រោងទុក។ ប្រព័ន្ធគ្រប់គ្រងនេះវាដំណើរការដំបូងគឺចាំបាច់ត្រូវអោយអ្នកប្រើប្រាស់ជ្រើសរើសលេខទៅតាមអ្វីដែលគេចង់ធ្វើ។

IV. Discussions

ក្នុងអត្ថបទសិក្សាស្រាវជ្រាវនេះ ធ្វើការពន្យល់ និងណែនាំពីវិធីសាស្ត្រនៅក្នុងការសរសេរកូដ គ្រប់គ្រងសាលារៀនតាមរយៈ Python។ កម្មវិធីគ្រប់គ្រងសាលារៀនមួយនេះ ត្រូវបានបង្កើតឡើងដោយតម្រូវឱ្យសាលាមួយដើម្បីគ្រប់គ្រងសិស្សនៅទូទាំងសាលាទាំងមូលសម្រាប់ការសិក្សា និងការសរសេរកូដនេះ ត្រូវធ្វើការបង្ហាញពីមុខងារនៃកូដមួយបន្ទាត់ៗផងដែរ។ កូដដែលសរសេរនៅក្នុងប្រព័ន្ធគ្រប់គ្រងសាលារៀននេះមានហ្វាំងហ្សិនចំនួន៣និងមានមុខងារផ្សេងៗគ្នា។

នៅក្នុងក្នុងគ្រប់គ្រងសាលារៀននេះ គឺអ្នកប្រើប្រាស់ខាងសាលាអាចបញ្ចូលលេខ ១ ដើម្បីទទួលបានចុះឈ្មោះសិស្សថ្មី ចូលរៀនដោយគ្រាន់តែវាយ ឈ្មោះ អាយុ ភេទ អាសយដ្ឋាន ទូរសព្ទ សាលារបស់សិស្សថ្មីដែលទើបនឹងចុះឈ្មោះចូលរៀន។ វាយលេខ ២ គឺដើម្បីមើលឈ្មោះសិស្សទាំងអស់នៅក្នុងសាលា វាយលេខ ៣ គឺដើម្បីលុបឈ្មោះសិស្សដោយគ្រាន់តែវាយ ឈ្មោះសិស្សដែលចង់លុបនោះ វាយលេខ ៤ គឺដើម្បីស្វែងរកឈ្មោះសិស្សណាម្នាក់ ចំណែកវាយលេខ ៥ គឺវាដាក់ចេញពី កម្មវិធី។

V. Conclusions

លទ្ធផលនៅក្នុងអត្ថបទកិច្ចការស្រាវជ្រាវ និងការអភិវឌ្ឍកម្មវិធីគ្រប់គ្រងប្រព័ន្ធសាលារៀននេះ បានបង្ហាញពីវិធីសាស្ត្រនៅក្នុងការសិក្សាស្រាវជ្រាវលក្ខណៈវិទ្យាសាស្ត្រនៅក្នុងការអភិវឌ្ឍកម្មវិធីកុំព្យូទ័រ ដើម្បីតម្រង់ទិសដល់និស្សិត និងអ្នកសិក្សាជំនាញបច្ចេកវិទ្យាកុំព្យូទ័រឱ្យមានសមត្ថភាពនៅក្នុងការអភិវឌ្ឍសមត្ថភាព ឬជំនាញផ្នែកសរសេរកូដក្នុង Python និងយល់ដឹងពីរបៀបនៅក្នុងការសរសេរអត្ថបទសិក្សាបែបវិជ្ជាជីវៈ។ ក្នុងគ្រប់គ្រងប្រព័ន្ធសាលារៀននេះ គ្រាន់តែជាផ្នែកមួយតូចនៅក្នុងការចាប់ផ្តើមអាជីពជាអ្នកជំនាញសរសេរកម្មវិធីកុំព្យូទ័រប៉ុណ្ណោះ។ ចំពោះកូដកម្មវិធីគ្រប់គ្រងសាលារៀនដែលបានបង្កើតឡើងនេះ សម្រាប់ការគ្រប់គ្រងសិស្ស ។

ផ្អែកទៅលើចក្ខុវិស័យនាពេលបច្ចុប្បន្ន យើងឃើញថាការប្រកបអាជីវកម្មកាន់តែកើនឡើងពីមួយ ថ្ងៃទៅមួយថ្ងៃ រីឯបច្ចេកវិទ្យាមានការវិវឌ្ឍនជាបន្តបន្ទាប់។ ចំណែកឯអាជីវកម្មប្រសិនបើគ្មានប្រព័ន្ធគ្រប់គ្រង ទិន្នន័យបានល្អទេ ធ្វើឱ្យអាជីវកម្មចុះខ្សោយជាដាច់ខាត ដោយសារតែយល់ឃើញនៅបញ្ហាមួយនេះទើបធ្វើឱ្យក្រុម និស្សិតយើងខ្ញុំបង្កើត System នេះឡើង ក្រុមនិស្សិតខ្ញុំបាន-នាងខ្ញុំសង្ឃឹមយ៉ាងមុតមាំថា ប្រជាជនទាំងអស់និង ទទួលស្គាល់ពីProject ក្រុមនិស្សិតយើងខ្ញុំជាក់ជាពុំខាន។

សម្រាប់ថ្ងៃអនាគត និស្សិត និងអ្នកសិក្សាស្រាវជ្រាវផ្សេងទៀតអាចសរសេរកូដកម្មវិធីគ្រប់គ្រងប្រព័ន្ធសាលារៀនតាមរយៈការដាក់ឱ្យប្រើប្រាស់នៅតាមគ្រឹះស្ថានដទៃទៀតផងដែរ ដើម្បីបង្កើនសមត្ថភាពក្នុងការផលិតកម្មវិធីកុំព្យូទ័ររបស់និស្សិតសិក្សាជំនាញផលិតកម្មវិធី។

VI. References

- [1] ដកស្រង់ចេញពីវេបសាយ: <https://m.freshnewsasia.com/index.php/en/freshnewsplus/190992-2021-03-26-12-00-09.html>
- [2] ដកស្រង់ចេញពីវេបសាយ :<https://sites.google.com/view/salariendigital>
- [3] ដកស្រង់ចេញពីវេបសាយ: <https://www.studocu.com/row/document/university-of-engineering-and-technology-lahore/computer-science/school-management-system-project-abstract-pdf/41702385?origin=home-recent-2>

Noted: ការសរសេររបស់ខ្ញុំភាគច្រើនគឺជាការផ្សំឡើងដោយគំនិតខ្លួនអែងនិងការអានចេញពីវេបសាយមួយចំនួន។

សូមអរគុណ!

