



ບົດໂຄງການຈົບຊັ້ນປະລິນຍາຕີວິທະຍາສາດ

ສາຂາ: ວິທະຍາສາດຄອມພິວເຕີ

**ລະບົບຈອງປີ້ລົດເມສາຍໃຕ້ອອນລາຍ
Southern Bus Ticket Online Booking System**

ທ້າວ ມະໂນພອນ ມະໂນກຸນ

ທ້າວ ພອນຄຳ ແກ້ວມະນີ

ຄະນະວິທະຍາສາດທຳມະຊາດ

ສົກສິກສາ 2020-2021

ບົດໂຄງການຈົບຊັ້ນລະດັບປະລິນຍາຕີວິທະຍາສາດ
ສາຂາ ວິທະຍາສາດຄອມພິວເຕີ

ລະບົບຈອງປີລິດເມສາຍໃຕ້ອອນລາຍ
Southern Bus Ticket Online Booking System

ນັກສຶກສາ:

ທ້າວ ມະໂນພອນ ມະໂນກຸນ
ທ້າວ ພອນຄຳ ແກ້ວມະນີ

ອາຈານຜູ້ນຳພາ:

ອຈ.ປທ ມູນພິນ ພອນປັນຍາ

ຊ່ວຍນຳພາໂດຍ:

ຊອ.ປທ ອິລະດີ ຄຳມະນີວົງ

ຄະນະວິທະຍາສາດທຳມະຊາດ
ພາກວິຊາ ວິທະຍາສາດຄອມພິວເຕີ
ສົກສຶກສາ 2020-2021

**Final Project of Bachelor Degree of Natural Sciences in
Computer science**

ລະບົບຈອງປີ້ລົດເມສາຍໃຕ້ອອນລາຍ

Southern Bus Ticket Online Booking System

Student:

Mr Manophone MANOKOUN

Mr Phonekham KEOMANY

Advisor:

Mounphine PHONEPANYA

Co-Adviser:

Orlady KHAMMANIVONG

Faculty of Natural Sciences

Department of Computer Sciences

Academic Year 2020-2021

ຄຳນຳ

ໃນປັດຈຸບັນ ເທັກໂນໂລຊີນັບມື້ນັບທັນສະໄໝຢຶ່ງຂຶ້ນ ແລະ ມີຄວາມກ້າວໜ້າຢ່າງບໍ່ຢຸດຢັ້ງ ເຊິ່ງມັນມີບົດບາດສຳຄັນໃນການດຳລົງຊີວິດຂອງພວກເຮົາ, ໃນນີ້ການນຳໃຊ້ເທັກໂນໂລຊີເຂົ້າມາຊ່ວຍໃນການຈັດການຂໍ້ມູນໂດຍສະເພາະແມ່ນການຈັດການກ່ຽວກັບຂໍ້ມູນຖືວ່າແມ່ນວຽກງານໜຶ່ງທີ່ສຳຄັນຕໍ່ກັບບັນດາວິສາຫະກິດ, ທຸລະກິດ ແລະ ອົງກອນຕ່າງໆ ເພື່ອເຮັດໃຫ້ການຈັດການຂໍ້ມູນຕ່າງໆມີຄວາມວ່ອງໄວທັນຕໍ່ສະພາບການ ແລະ ສາມາດເກັບຮັກສາຂໍ້ມູນຕ່າງໆທີ່ມີຄວາມສຳຄັນຕໍ່ກັບອົງກອນໃຫ້ມີຄວາມປອດໄພຂຶ້ນ.

ຈຸດປະສົງຫຼັກຂອງການສ້າງລະບົບຂາຍປີ້ລົດເມອອນລາຍຂອງສະຖານຂົນສົ່ງໂດຍສານສາຍໃຕ້ຂຶ້ນມາແມ່ນເພື່ອຈະນຳເອົາເທັກໂນໂລຊີທີ່ທັນສະໄໝເຂົ້າມາຊ່ວຍໃນການຂາຍປີ້ ແລະ ຈັດການຂໍ້ມູນໃຫ້ມີຄວາມສະດວກ, ເພີ່ມປະສິດທິພາບໃນການເຮັດວຽກໃຫ້ມີຄວາມຖືກຕ້ອງ ແລະ ຊັດເຈນທີ່ສຸດ.

ເຖິງຢ່າງໃດກໍຕາມ ພວກຂ້າພະເຈົ້າຫ້ວງຢ່າງຍິ່ງວ່າບົດໂຄງການຈົບຊັ້ນຈະມີຄຸນປະໂຫຍດບໍ່ຫຼາຍກໍໜ້ອຍຕໍ່ນັກສຶກສາລຸ້ນຕໍ່ໄປສຳລັບຜູ້ທີ່ສົນໃຈສ້າງ ຫຼື ພັດທະນາລະບົບທີ່ກ່ຽວຂ້ອງ. ພ້ອມດຽວກັນກໍຫ້ວງວ່າລະບົບດັ່ງກ່າວນີ້ຈະຊ່ວຍອຳນວຍຄວາມສະດວກໃນການບໍລິການ ແລະ ຈັດການຂໍ້ມູນຕ່າງໆຂອງທາງສະຖານນີຂົນສົ່ງໂດຍສານສາຍໃຕ້. ໃນລະບົບທີ່ພວກຂ້າພະເຈົ້າໄດ້ສ້າງຂຶ້ນມານີ້ບໍ່ອາດຫຼີກລ້ຽງຂໍ້ຜິດພາດໄດ້, ສະນັ້ນ ພວກຂ້າພະເຈົ້າຈຶ່ງຂໍອະໄພມາຍັງບັນດາທ່ານ ແລະ ຍອມຮັບຄຳຕຳນິຕິຊົມຈາກທ່ານ ແລະ ນຳມາແກ້ໄຂ ເຮັດໃຫ້ປຶ້ມຫົວນີ້ສົມບູນຂຶ້ນ.

ບົດຄັດຫຍໍ້

ບົດໂຄງການຈົບຊັ້ນນີ້ເປັນການສຶກສາພັດທະນາລະບົບການຈອງປີລົດເມແບບອອນລາຍ ເຊັ່ນ: ການຈອງປີລົດແບບອອນລາຍ, ການຈັດການຂໍ້ມູນ, ລາຍງານຂໍ້ມູນຕ່າງໆ, ລວມທັງການຂາຍປີ ແມ່ນຍັງໃຊ້ແບບຈົດ ແລະ ຜູ້ໂດຍສານທີ່ຕ້ອງການຈອງປີລວງໜ້າຕ້ອງໄດ້ໂທຫາພະນັກງານຂາຍ ປີເພື່ອຈອງ ເຊິ່ງເຮັດໃຫ້ການບໍລິການມີການຊັກຊ້າ ແລະ ຂໍ້ມູນຍັງມີການຕົກເຫຍ່ຍເສຍຫາຍ ດັ່ງນັ້ນ ພວກຂ້າພະເຈົ້າຈຶ່ງເຫັນຄວາມສໍາຄັນຂອງບັນຫາຈຶ່ງມີແນວຄວາມຄິດທີ່ຈະສ້າງລະບົບຈອງປີລົດເມ ແບບອອນລາຍຂອງສະຖານນີຂົນສົ່ງໂດຍສານສາຍໃຕ້ນັ້ນຂຶ້ນມາເພື່ອຊ່ວຍຫຼຸດຜ່ອນຄວາມ ຫຍຸ້ງຍາກໃນການຈອງປີລົດ, ຈັດເກັບຂໍ້ມູນ, ຫຼຸດຜ່ອນຄວາມຊັກຊ້າໃນການຈັດການຂໍ້ມູນ, ເຮັດໃຫ້ ຂໍ້ມູນມີຄວາມເປັນລະບຽບຮຽບຮ້ອຍ ແລະ ເພື່ອໃຫ້ມີຄວາມສະດວກວ່າໄວຕໍ່ການຄົ້ນຫາຂໍ້ມູນ.

ດັ່ງນັ້ນ ຈຶ່ງຮຽກຮ້ອງໃຫ້ນຳເອົາລະບົບການຈັດການຖານຂໍ້ມູນ ທີ່ມີປະສິດທິພາບສູງເຂົ້າມາ ຊ່ວຍໃນການບໍລິຫານເກັບກຳເອົາບັນດາຂໍ້ມູນຕ່າງໆທີ່ເພີ່ມຂຶ້ນນັ້ນໃຫ້ມີຄວາມເປັນລະບຽບຮຽບ ຮ້ອຍ, ຄົບຖ້ວນ ແລະ ສົມບູນກ່ອນເກົ່າ.

ເຊິ່ງນອກຈາກໄດ້ນຳເອົາວິຊາການວິເຄາະ ແລະ ອອກແບບລະບົບມານຳໃຊ້ແລ້ວກໍໄດ້ນຳ ໃຊ້ໂປແກມ Microsoft Visio 2016 ໃຊ້ແຕ້ມແຜນວາດການໄຫຼຂໍ້ມູນ (DFD, ER, Flowchart) , Adobe XD ໃຊ້ອອກແບບ UX/UI , Studio 3T For MongoDB ແລະ Moon Modeler ໃຊ້ ຈັດການຖານຂໍ້ມູນ ແລະ ອອກແບບ Database Model , Visual Studio Code ໃຊ້ຂຽນໂຄດດ້ວຍ ພາສາ JavaScript (ReactJS, NodeJS, GraphQL).

Abstract

This is the study project to develop Southern Station Bus Booking Online System was developed to make the sale system be modern, serve customers more efficiently and quickly, create reports, when customer want to book they have to call employee to make booking, so it make booking slow and some information are lost the development of the Southern Station Bus Booking Online System consists of 2 main parts which are software and hardware. The software part we use MongoDB Compass for storing data for the program. For the database, we use MongoDB, Visual Studio code for writing code We use Microsoft Visio 2016 for drawing data flow diagrams, flowchart and entity relationship diagrams (ER-Diagram), We use we write code in JavaScript, use Nodejs for Backend and Reactjs for frontend and use GraphQL for Creating API. The hardware part we use 2 laptops for developing the application. The results from Southern Station Bus Booking Online System are a system that will reduce redundancy and complexity of the data, to facilitate the customers. Making the users of the system be more convenient and have accurate information.

ຄຳສະແດງຄວາມຂອບໃຈ

ພວກຂ້າພະເຈົ້າເປັນນັກສຶກສາ ປີທີ 2 (ຕໍ່ເນື່ອງ) ພາກວິຊາວິທະຍາສາດຄອມພິວເຕີ, ຄະນະວິທະຍາສາດທຳມະຊາດ, ມະຫາວິທະຍາໄລແຫ່ງຊາດ ສຶກສາສາ 2020-2021 ພວກຂ້າພະເຈົ້າມີຄວາມພາກພູມໃຈ ແລະ ເປັນກຽດຢ່າງສູງທີ່ໄດ້ໂອກາດສຶກສາໃນລະດັບປະລິນຍາຕີ. ດັ່ງນັ້ນ ພວກຂ້າພະເຈົ້າຈຶ່ງຂໍສະແດງຄວາມຮູ້ບຸນຄຸນ ແລະ ຂອບໃຈຢ່າງສູງຕໍ່ຄະນະນຳຂອງຄະນະວິທະຍາສາດທຳມະຊາດ, ຄະນະນຳຂອງພາກວິຊາວິທະຍາສາດຄອມພິວເຕີ ຕະຫຼອດເຖິງການຈັດຕັ້ງທຸກພາກສ່ວນທີ່ກ່ຽວຂ້ອງທີ່ໄດ້ໃຫ້ຄວາມຊ່ວຍເຫຼືອ ແລະ ໄດ້ໃຫ້ຄວາມຮ່ວມມືໃນບົດໂຄງການຈົບຊັ້ນປະລິນຍາຕີສະບັບນີ້.

ບົດໂຄງການຈົບຊັ້ນນີ້ ໄດ້ສຳເລັດລຸ່ວງລົງໄດ້ເພາະວ່າໄດ້ຮັບຄຳປຶກສາ ແລະ ຄວາມຊ່ວຍເຫຼືອຈາກບຸກຄົນຫຼາຍທ່ານ. ພິເສດພວກຂ້າພະເຈົ້າ ຂໍຖືໂອກາດນີ້ສະແດງຄວາມຂອບໃຈ ແລະ ຮູ້ບຸນຄຸນຢ່າງສຸດຊຶ້ງມາຍັງອາຈານທີ່ປຶກສາ ອ.ຈ.ປທ ມູນພິນ ພອນປັນຍາ ແລະ ອາຈານຊ່ວຍນຳພາ ຊອ.ປທ ອິລະດີ ຄຳມະນີວົງ ທີ່ໃຫ້ຄຳປຶກສາ ແລະ ຊີ້ນຳ-ນຳພາ ພວກຂ້າພະເຈົ້າໃນການຂຽນບົດໂຄງການຈົບຊັ້ນຄັ້ງນີ້ຈົນໄດ້ຮັບຜົນສຳເລັດລຸ່ວງໄປດ້ວຍດີ, ຂໍສະແດງຄວາມຂອບໃຈ ແລະ ຮູ້ບຸນຄຸນມາຍັງບັນດາອາຈານທຸກທ່ານຂອງພາກວິຊາວິທະຍາສາດຄອມພິວເຕີທີ່ໄດ້ທຸ່ມເທສະຕິປັນຍາສິດສອນທັງຄວາມຮູ້ວິຊາການ ແລະ ຄວາມຮູ້ດ້ານອື່ນໆອີກໃຫ້ແກ່ພວກຂ້າພະເຈົ້າຕະຫຼອດໄລຍະ 2 ປີ.

ຂໍສະແດງຄວາມຂອບໃຈມາຍັງບັນດາໝູ່ເພື່ອນໝົດທຸກຄົນທີ່ໄດ້ໃຫ້ຄວາມຊ່ວຍເຫຼືອ ແລະ ໃຫ້ກຳລັງແກ່ພວກຂ້າພະເຈົ້າມາຕະຫຼອດໄລຍະ 2 ປີ.

ຂໍຂອບໃຈ ທາງສະຖານນີຂົນສົ່ງໂດຍສານສາຍໃຕ້ ທີ່ໃຫ້ຄວາມຮ່ວມມືຊ່ວຍເຫຼືອໃນການເກັບກຳຂໍ້ມູນ, ສະໜອງຂໍ້ມູນ ແລະ ອຳນວຍຄວາມສະດວກໃນການລົງເອົາຂໍ້ມູນ ພ້ອມທັງລາຍລະອຽດຕ່າງໆທີ່ເປັນປະໂຫຍດແກ່ການຂຽນບົດຈົບຊັ້ນຂອງພວກຂ້າພະເຈົ້າ.

ສຸດທ້າຍນີ້ ພວກຂ້າພະເຈົ້າຂໍສະແດງຄວາມຮູ້ບຸນຄຸນຢ່າງໃຫຍ່ຫຼວງມາຍັງຄອບຄົວ ແລະ ການຈັດຕັ້ງຂອງພວກຂ້າພະເຈົ້າ ທີ່ໄດ້ໃຫ້ການສົ່ງເສີມຊ່ວຍເຫຼືອໃຫ້ພວກຂ້າພະເຈົ້າໄດ້ມີໂອກາດສຶກສາຮ່ຳຮຽນຕາມຄວາມໄຝ່ຝັນ, ມອບທັງຄວາມຮັກ, ຄວາມອົບອຸ່ນ, ກຳລັງໃຈ ແລະ ເວລາອັນເຕັມທີ່ນັບບໍ່ຖ້ວນ ໃຫ້ພວກຂ້າພະເຈົ້າໄດ້ສຶກສາຮ່ຳຮຽນຈົນສຳເລັດ.

ສາລະບານ

ໜ້າທີ

ບົດທີ 1	1
ບົດສະເໜີ.....	1
1.1 ຄວາມສໍາຄັນຂອງບັນຫາ.....	1
1.2 ຈຸດປະສົງຂອງການສຶກສາ.....	1
1.3 ຂອບເຂດການຄົ້ນຄວ້າ.....	2
1.4 ປະໂຫຍດທີ່ຄາດວ່າຈະໄດ້ຮັບ.....	2
ບົດທີ 2	3
ທົບທວນເອກະສານ ແລະ ບົດຄົ້ນຄວ້າທີ່ກ່ຽວຂ້ອງ.....	3
2.1 ທົບທວນທິດສະດີທີ່ກ່ຽວຂ້ອງ.....	3
2.1.1 ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບຖານຂໍ້ມູນ.....	3
1) ຄວາມໝາຍຂອງຖານຂໍ້ມູນ.....	3
2) ຄຸນລັກສະນະຂອງຖານຂໍ້ມູນ.....	3
3) ຄວາມສໍາຄັນຂອງລະບົບຖານຂໍ້ມູນ	4
4) ສະຖາປັດຕະຍະກຳລະບົບຖານຂໍ້ມູນ	4
2.1.2 ທິດສະດີທີ່ກ່ຽວຂ້ອງກັບ DFD (Data Flow Diagram).....	4
1) ຈຸດປະສົງຂອງ DFD	4
2) ສັນຍະລັກທີ່ໃຊ້ໃນແຜນວາດການໄຫຼຂໍ້ມູນ.....	5
3) ກົດຂອງ Process.....	5
4) ຫຼັກການຂຽນແຜນວາດການໄຫຼຂໍ້ມູນ.....	6
2.1.3 ຄວາມໝາຍ ແລະ ສັນຍະລັກຂອງ Flowchart.....	7
2.1.3 ແຜນວາດຄວາມສໍາພັນຂອງ Entity (ER Diagram).....	8

ສາລະບານ

ໜ້າທີ

2.1.4 ສັນຍາລັກທີ່ໃຊ້ໃນແບບຈຳລອງ ER	9
2.1.5 ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບພາສາ NOSQL	10
2.1.6 ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບພາສາ JavaScript.....	15
1) ປະຫວັດຄວາມເປັນມາຂອງພາສາ JavaScript	16
2) ຄຸນສົມບັດຂອງພາສາ JavaScript	17
3) JavaScript Engine ແມ່ນຫຍັງ?	18
2.1.7 ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບ Application Programming Interface (API)	18
1) Libraries and Frameworks	19
2) Operating Systems	19
3) Remote APIs	19
4) Web APIs	19
2.1.8 ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບ Nodejs	20
2.1.9 ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບ Reactjs (JavaScript Library).....	22
1) ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບ GraphQL (GraphQL API).....	23
2) ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບ Expressjs (Nodejs Library)	24
3) ຄວາມຮູ້ກ່ຽວກັບໂປຣແກຣມ Microsoft Office 2016.....	25
ກ. ປະໂຫຍດຂອງໂປຣແກຣມ Microsoft Word Office 2016	25
ຂ. ຄວາມສາມາດຂອງ Microsoft Word	26
4) ຄວາມຮູ້ກ່ຽວກັບໂປຣແກຣມ Microsoft Visio 2016	26
5) ຄວາມຮູ້ກ່ຽວກັບໂປຣແກຣມ Visual Studio Code	27

ສາລະບານ (ຕໍ່)

ໜ້າທີ

2.2 ບົດຄົ້ນຄວ້າທີ່ກ່ຽວຂ້ອງ.....	28
2.2.1 Southern Bus Station Booking online System.....	28
ບົດທີ 3	30
ວິທີດໍາເນີນການຄົ້ນຄວ້າ	30
3.1 ວິທີສຶກສາ ແລະ ຄົ້ນຄວ້າ.....	30
3.1.1 ໄລຍະທີ 1 ການວາງແຜນໂຄງການ (Project planning phase).....	31
3.1.2 ໄລຍະທີ 2 ການວິເຄາະ (Analysis Phase).....	31
3.1.3 ໄລຍະທີ 3 ການອອກແບບ (Design Phase).....	33
3.1.4 ໄລຍະທີ 4 ການນໍາໄປໃຊ້ (Implementation Phase)	34
3.1.5 ໄລຍະທີ 5 ການບໍາລຸງຮັກສາ (Maintenance Phase)	34
3.2 ເຄື່ອງມືທີ່ໃຊ້ໃນການພັດທະນາ (Development Tools).....	35
3.3 ການວິເຄາະຂໍ້ມູນ.....	36
3.3.1 ແຜນວາດລວມຂອງລະບົບ	36
3.1.2 ຕາຕະລາງສະແດງລາຍລະອຽດຕ່າງໆທີ່ກ່ຽວກັບລະບົບ	37
3.3.3 ແຜນວາດເນື້ອຫາ (Context Diagram)	38
3.1.4 ແຜນວາດລຳດັບຊັ້ນໜ້າທີ່ (Functional Hierarchy Diagram)	39
3.1.5 ແຜນວາດການໄຫຼຂໍ້ມູນ (Data Flow Diagram: DFD).....	40
1) ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງແຕ່ລະ Process.....	40
2) ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 1.....	41
3) ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 2.....	41
4) ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 3.....	42

ສາລະບານ (ຕໍ່)

ໜ້າທີ

5) ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 4.....	42
6) ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 1.....	43
7) ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 3.....	44
8) ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 4.....	45
9) ແຜນວາດຄວາມສໍາພັນຂອງຂໍ້ມູນ (ER Diagram).....	46
3.4 ການອອກແບບລະບົບ.....	47
3.4.1 ການອອກແບບຮ່າງສະແດງຜົນ (Output Design)	47
1) ລາຍງານຂໍ້ມູນລົດ.....	47
2) ລາຍງານຂໍ້ມູນສາຍທາງ.....	48
3) ລາຍງານຂໍ້ມູນພະນັກງານ	49
4) ລາຍງານຂໍ້ມູນການຈອງປີ	50
3.4.2 ການອອກແບບຮ່າງປ້ອນຂໍ້ມູນ (Input Design).....	51
1) ຟອມເຂົ້າສູ່ລະບົບແອດມິນ	51
2) ຟອມການຈັດການຂໍ້ມູນບໍລິສັດ	52
3) ຟອມການເພີ່ມຂໍ້ມູນບໍລິສັດ	53
4) ຟອມການຈັດການຂໍ້ມູນສາຍທາງ.....	54
5) ຟອມການເພີ່ມຂໍ້ມູນສາຍທາງ.....	55
6) ຟອມການຈັດການຖັງວລົດ.....	56
7) ຟອມການເພີ່ມຂໍ້ມູນຖັງວລົດ.....	57
8) ຟອມການຈັດການປະເພດລົດ	58

ສາລະບານ (ຕໍ່)

ໜ້າທີ

9) ຟອມການເພີ່ມປະເພດລົດ	59
3.5 ການອອກແບບຖານຂໍ້ມູນ	60
3.5.1 ຕາຕະລາງພະນັກງານ (Employee)	60
3.5.2 ຕາຕະລາງຂໍ້ມູນປີ້ (Ticket)	60
3.5.3 ຕາຕະລາງຂໍ້ມູນບໍລິສັດ (Company)	61
3.5.4 ຕາຕະລາງຂໍ້ມູນປະເພດລົດ (BusType)	61
3.5.5 ຕາຕະລາງຂໍ້ມູນລົດ (Bus)	62
3.5.6 ຕາຕະລາງຂໍ້ມູນປ່ອນນັ່ງ (Seat)	62
3.5.7 ຕາຕະລາງຂໍ້ມູນສະມາຊິກ (Member)	63
3.5.8 ຕາຕະລາງຂໍ້ມູນການຈອງ (Booking)	63
3.5.9 ຕາຕະລາງຂໍ້ມູນລາຍລະອຽດການຈອງ (BookingItem)	64
3.5.10 ຕາຕະລາງຂໍ້ມູນສາຍທາງ (Route)	64
3.5.11 ຕາຕະລາງຂໍ້ມູນເວລາລົດອອກ (DepartureTime)	65
3.6 ແຜນວາດຂັ້ນຕອນການເຮັດວຽກ (Flowchart)	66
3.6.1 ແຜນວາດ Flowchart ການສະໝັກສະມາຊິກ	66
3.6.2 ແຜນວາດ Flowchart ການເຂົ້າສູ່ລະບົບ	67
3.6.3 ແຜນວາດ Flowchart ການສັ່ງຊື້	68
3.6.4 ແຜນວາດ Flowchart ການຊໍາລະເງິນ	69
3.6.5 ແຜນວາດ Flowchart ການອອກປີ້	70
3.6.6 ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນພະນັກງານ	71

ສາລະບານ (ຕໍ່)

ໜ້າທີ

3.6.7 ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນລູກຄ້າ	72
3.6.8 ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນລົດ	73
3.6.9 ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນປະເພດລົດ	74
3.5.10 ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນສາຍທາງ	75
ບົດທີ 4	76
ຜົນຂອງການວິເຄາະຂໍ້ມູນ ແລະ ການອະທິບາຍຜົນ	76
4.1 ໜ້າຕ່າງການ (Login Form).....	76
4.2 ຟອມຫຼັກ (Main Form).....	77
4.2.1 ຟອມຄົ້ນຫາຖ້ຽວລົດ	77
4.2.2 ຟອມເລືອກບ່ອນນັ່ງ	78
4.2.3 ຟອມປ້ອນລາຍລະອຽດຜູ້ໂດຍສານ	79
4.2.4 ຟອມຊຳລະເງິນ	80
ບົດທີ 5	81
ສະຫຼຸບ ແລະ ຂໍ້ສະເໜີ.....	81
5.1 ສະຫຼຸບຜົນການຄົ້ນຄວ້າ.....	81
5.2 ຈຸດດີ.....	81
5.4 ແນວທາງໃນການພັດທະນາ ແລະ ຂະຫຍາຍຕໍ່ຂອງສະຖານນີ.....	82
ເອກະສານອ້າງອີງ.....	83
ເອກະສານຊ້ອນທ້າຍ	87
ປະຫວັດຜູ້ຂຽນ.....	89

ສາລະບານຕາຕະລາງ

ໜ້າທີ

ຕາຕະລາງທີ 1: ສະແດງສັນຍາລັກ (DATA FLOW DIAGRAM)	5
ຕາຕະລາງທີ 2 : ຮູບການປຸງປະໂຫຍດແຜນວາດການໄຫຼຂໍ້ມູນທີ່ບໍ່ຖືກຕ້ອງ ແລະ ຖືກຕ້ອງ	7
ຕາຕະລາງທີ 3 : ສະແດງສັນຍາລັກ FLOWCHART	8
ຕາຕະລາງທີ 4 : ຕາຕະລາງປຸງປະໂຫຍດລະຫວ່າງ SYNCHRONOUSE ກັບ ASYNCHRONOUS	21
ຕາຕະລາງທີ 5 : ສະແດງລາຍລະອຽດຕ່າງໆທີ່ກ່ຽວກັບລະບົບ	37
ຕາຕະລາງທີ 6 : ຕາຕະລາງພະນັກງານ (EMPLOYEE)	60
ຕາຕະລາງທີ 7 : ຕາຕະລາງຂໍ້ມູນປີ້ (TICKET)	61
ຕາຕະລາງທີ 8 : ຕາຕະລາງຂໍ້ມູນບໍລິສັດ (COMPANY)	61
ຕາຕະລາງທີ 9 : ຕາຕະລາງຂໍ້ມູນປະເພດລົດ (BUS TYPE)	62
ຕາຕະລາງທີ 10 : ຕາຕະລາງຂໍ້ມູນລົດ (BUS)	62
ຕາຕະລາງທີ 11 : ຕາຕະລາງຂໍ້ມູນບ່ອນນັ່ງ (SEAT)	63
ຕາຕະລາງທີ 12 : ຕາຕະລາງຂໍ້ມູນສະມາຊິກ (MEMBER)	63
ຕາຕະລາງທີ 13 : ຕາຕະລາງຂໍ້ມູນການຈອງ (BOOKING)	64
ຕາຕະລາງທີ 14: ຕາຕະລາງຂໍ້ມູນລາຍລະອຽດການຈອງ (BOOKING ITEM)	64
ຕາຕະລາງທີ 15: ຕາຕະລາງຂໍ້ມູນສາຍທາງ (ROUTE)	65
ຕາຕະລາງທີ 16: ຕາຕະລາງຂໍ້ມູນເວລາລົດອອກ (DEPARTURE TIME)	65

ສາລະບານແຜນວາດ

ໜ້າທີ

ແຜນວາດທີ 1 : ແຜນວາດລວມຂອງລະບົບ	30
ແຜນວາດທີ 2 : ແຜນວາດລວມຂອງລະບົບ	36
ແຜນວາດທີ 3 : ແຜນວາດເນື້ອຫາ (CONTEXT DIAGRAM)	38
ແຜນວາດທີ 4 : ແຜນວາດລຳດັບຊັ້ນໜ້າທີ (FUNCTIONAL HIERARCHY DIAGRAM)	39
ແຜນວາດທີ 5 : ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງແຕ່ລະ PROCESS	40
ແຜນວາດທີ 6 : ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ PROCESS 1	41
ແຜນວາດທີ 7 : ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ PROCESS 2.....	41
ແຜນວາດທີ 8 : ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ PROCESS 3.....	42
ແຜນວາດທີ 9 : ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ PROCESS 4	42
ແຜນວາດທີ 10 : ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ PROCESS 1	43
ແຜນວາດທີ 11 : ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ PROCESS 3.....	44
ແຜນວາດທີ 12 : ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ PROCESS 4.....	45
ແຜນວາດທີ 13 : ແຜນວາດຄວາມສຳພັນຂອງຂໍ້ມູນ (ER DIAGRAM).....	46
ແຜນວາດທີ 14 : ແຜນວາດ FLOWCHART ການສະໜັກສະນາຂີກ	66
ແຜນວາດທີ 15 : ແຜນວາດ FLOWCHART ການເຂົ້າສູ່ລະບົບ	67
ແຜນວາດທີ 16 : ແຜນວາດ FLOWCHART ການສົ່ງຂໍ້.....	68

ສາລະບານຮູບພາບ

ໜ້າທີ

ຮູບທີ 1 : ຮູບ STRONG ENTITY	9
ຮູບທີ 2 : ຮູບ WEAK ENTITY	9
ຮູບທີ 3 : ຮູບ ATTRIBUTE	10
ຮູບທີ 4 : ຮູບ RELATION	10
ຮູບທີ 5 : ຮູບຕົວຢ່າງ GraphQL	24
ຮູບທີ 6 : ຮູບໂປຣແກຣມ VS CODE	28
ຮູບທີ 7 : ຮູບອອກແບບສະແດງລາຍງານຂໍ້ມູນລົດ	47
ຮູບທີ 8 : ຮູບອອກແບບລາຍງານຂໍ້ມູນສາຍທາງ	48
ຮູບທີ 9 : ຮູບອອກແບບສະແດງລາຍງານຂໍ້ມູນພະນັກງານ	49
ຮູບທີ 10 : ຮູບອອກແບບສະແດງລາຍງານການຈອງປີ	50
ຮູບທີ 11 : ຮູບແບບຟອມເຂົ້າສູ່ລະບົບແອດມິນ	51
ຮູບທີ 12 : ຮູບແບບຟອມການຈັດການຂໍ້ມູນບໍລິສັດ	52
ຮູບທີ 13 : ຮູບແບບຟອມການເພີ່ມຂໍ້ມູນບໍລິສັດ	53
ຮູບທີ 14 : ຮູບແບບຟອມການຈັດການຂໍ້ມູນສາຍທາງ	54
ຮູບທີ 15 : ຮູບແບບຟອມການເພີ່ມຂໍ້ມູນສາຍທາງ	55
ຮູບທີ 16 : ຮູບແບບຟອມການຈັດການຖັງວລົດ	56
ຮູບທີ 17 : ຮູບແບບຟອມການເພີ່ມຂໍ້ມູນຖັງວລົດ	57
ຮູບທີ 18 : ຮູບແບບຟອມການຈັດການປະເພດລົດ	58
ຮູບທີ 19 : ຮູບແບບຟອມການເພີ່ມປະເພດລົດ	59
ຮູບທີ 20 : ຮູບໜ້າຕ່າງການ (LOGIN FORM)	76
ຮູບທີ 21 : ຮູບແບບຟອມຄົ້ນຫາຖັງວລົດ	77
ຮູບທີ 22 : ຮູບແບບຟອມຈັດການຂໍ້ມູນບ່ອນນັ່ງ	78
ຮູບທີ 23 : ຮູບແບບຟອມບ້ອນລາຍລະອຽດຜູ້ໂດຍສານ	79
ຮູບທີ 24 : ຮູບແບບຟອມຊໍາລະເງິນ	80

ຄຳອະທິບາຍສັບ

DBMS	=	Database Management System
SQL	=	Structure Query Language
PHP	=	Personal Home Page
HTML	=	Hypertext Markup Language
CSS	=	Cascading Style Sheet
DFD	=	Data Flow Diagram
ERD	=	Entity Relationship Diagram
DDL	=	Data Definition Language
DML	=	Data Manipulation Language
QR Code	=	Quick Response Code
SDLC	=	Software Development Life Cycle

ບົດທີ 1

ບົດສະເໜີ

1.1 ຄວາມສໍາຄັນຂອງບັນຫາ

ອີງໃສ່ເພື່ອຄວາມສະດວກສະບາຍຂອງປະຊາຊົນລາວເຮົາ ແລະ ຄົນຕ່າງປະເທດ, ນັກທ່ອງທ່ຽວ, ພະນັກງານ, ພໍ່ຄ້າຊາວຂາຍຕະຫຼອດຮອດພໍ່ແມ່ປະຊາຊົນທີ່ເດີນທາງເຂົ້າມາທ່ອງທ່ຽວ ຫຼື ມາເຮັດວຽກເຮັດງານທຶນະຄອນຫຼວງວຽງຈັນຂອງພວກເຮົາເພື່ອໃຫ້ການໄປມາສະຖານທີ່ຕ່າງໆ ພວກເຮົາຈຶ່ງໄດ້ສ້າງຕັ້ງສະຖານນີຂົນສົ່ງໂດຍສານນີ້ຂຶ້ນມາ.

ສະຖານນີຂົນສົ່ງໂດຍສານສາຍໃຕ້ເປັນບ່ອນບໍລິການຮັບ-ສົ່ງຜູ້ໂດຍສານ, ສິນຄ້າວັດຖຸສິ່ງຂອງ ແລະ ສັດຈາກຈຸດໜຶ່ງໄປຫາອີກຈຸດໜຶ່ງຊຶ່ງສະຖານນີຂົນສົ່ງໂດຍສານທາງໄກສາຍໃຕ້ນີ້ແມ່ນໄດ້ສ້າງຕັ້ງຂຶ້ນໃນວັນທີ 1 ກັນຍາ 2016 ເຊິ່ງມີລາຍລະອຽດດັ່ງລຸ່ມນີ້: ທີ່ຕັ້ງ ແລະ ພາລະບົດບາດຂອງສະຖານນີຂົນສົ່ງທາງໄກສາຍໃຕ້ແມ່ນສະຖານນີໜຶ່ງຊຶ່ງຕັ້ງຢູ່ ບ້ານ ສະພັງມຶກ, ເມືອງ ໄຊທານີ, ນະຄອນຫຼວງວຽງຈັນ, ຖະໜົນເລກທີ 450 ປີ ໃກ້ກັບສີ່ແຍກໄຟແດງດົງໂດກ. ສະຖານນີຂົນສົ່ງທາງໄກສາຍໃຕ້ປະກອບມີຫຼາຍໜ່ວຍງານຄື: ອຳນວຍການໃຫ່ຍມີ 1 ທ່ານ, ເລຂານຸການມີ 1 ທ່ານ, ໜ່ວຍງານແຜນການມີ 1 ທ່ານ, ໜ່ວຍງານຮັບ-ຈ່າຍເງິນມີ 1 ທ່ານ, ໜ່ວຍງານບໍລິການຂາຍປີ້ມີ 7 ທ່ານ, ໜ່ວຍງານຮັກສາຄວາມປອດໄພມີ 6 ທ່ານ, ໜ່ວຍງານບໍລິການເຮືອນພັກມີ 8 ທ່ານ ແລະ ບັນດາບໍລິສັດ ທີ່ເຂົ້າມາດຳເນີນທຸລະກິດໃນສະຖານນີຂົນສົ່ງໂດຍສານປະກອບມີ: ບໍລິສັດ ຂົນສົ່ງໂດຍສານຈົດປະສົງຍອດນິຍົມ, ບໍລິສັດ ແສງສົມບູນ ຂົນສົ່ງໂດຍສານ, ບໍລິສັດ ແສນສະບາຍ ຂົນສົ່ງໂດຍສານ, ບໍລິສັດ ຈັນທະຈອນ ຂົນສົ່ງໂດຍສານ, ບໍລິສັດ ຈຳປາສັກ ຂົນສົ່ງໂດຍສານ, ບໍລິສັດ ແສງຈະເລີນ ລົດຕຽງນອນ, ບໍລິສັດ ກຽງໄກ VIP, ບໍລິສັດ ສີທອນ ພວງປະເສີດ ລົດຕຽງນອນ. ນອກຈາກນີ້ສະຖານນີຍັງມີສະຖານທີ່ພັກ, ຮ້ານຄ້າ, ຮ້ານຂາຍຍ່ອຍ, ຮ້ານອາຫານ ແລະ ສິ່ງອຳນວຍຄວາມສະດວກຕ່າງໆໄວ້ເພື່ອບໍລິການຜູ້ໂດຍສານທີ່ມາລໍຖ້າລົດໄປຈຸດໝາຍປາຍທາງ.

1.2 ຈຸດປະສົງຂອງການສຶກສາ

ວັດຖຸປະສົງຫຼັກຂອງບົດຈົບຊັ້ນໃນຄັ້ງນີ້ແມ່ນເພື່ອອີງໃສ່ໃນການພັດທະນາ ແລະ ເພີ່ມປະສິດທິພາບ ໃນການບໍລິການຂອງສະຖານນີຂົນສົ່ງໂດຍສານສາຍໃຕ້, ພວກຂ້າພະເຈົ້າຈຶ່ງໄດ້ມີ

ແນວຄິດທີ່ຈະສ້າງ ແລະ ພັດທະນາ ລະບົບຈອງປຶ້ລິດເມສາຍໃຕ້ອອນລາຍຂຶ້ນມາເພື່ອແກ້ໄຂບັນຫາໃຫ້ກັບທາງສະຖານນີຕາມວຽກຂອງແຕ່ລະພະແນກເຖິງບັນຫາໃນການເຮັດວຽກ. ດັ່ງນັ້ນພວກຂ້າພະເຈົ້າຈຶ່ງໄດ້ກຳນົດເຫດຜົນຂອງການຄົ້ນຄວ້າຕາມຈຸດປະສົງດັ່ງນີ້:

- ເພື່ອສຶກສາບັນຫາທີ່ເກີດຂຶ້ນໃນປະຈຸບັນ ແລະ ຄວາມຕ້ອງການຂອງລະບົບ.
- ເພື່ອສ້າງລະບົບຂາຍປຶ້ລິດອອນລາຍຂອງສະຖານນີຂົນສົ່ງໂດຍສານສາຍໃຕ້.
- ເພື່ອສ້າງຮູບແບບການຈັດການຂໍ້ມູນການໃຫ້ບໍລິການ.
- ເພື່ອເຜີຍແຜ່ຂໍ້ມູນການຂາຍປຶ້ລິດເມຂອງສະຖານນີຂົນສົ່ງໂດຍສານສາຍໃຕ້.
- ເພື່ອການລາຍງານໃຫ້ສະດວກ ແລະ ຖືກຕ້ອງ.

1.3 ຂອບເຂດການຄົ້ນຄວ້າ

ລະບົບຈອງປຶ້ລິດເມສາຍໃຕ້ແບບອອນລາຍຂອງສະຖານນີຂົນສົ່ງສາຍໃຕ້ເປັນລະບົບແບບ Client-Server Web Application ເຊິ່ງປະກອບດ້ວຍໜ້າວຽກຫຼັກດັ່ງນີ້:

- ຈັດການຂໍ້ມູນພື້ນຖານ (ຂໍ້ມູນພະນັກງານ, ຂໍ້ມູນລົດ, ຂໍ້ມູນປະເພດລົດ, ຂໍ້ມູນສາຍທາງ).
- ສະໝັກສະມາຊິກ.
- ບໍລິການ (ຈອງປີ້, ອອກປີ້).
- ລາຍງານ (ລາຍງານການຈອງ, ລາຍງານຂໍ້ມູນພະນັກງານ, ລາຍງານຂໍ້ມູນສາຍທາງ, ລາຍງານຂໍ້ມູນລົດ, ລາຍງານຂໍ້ມູນຊໍາລະເງິນ)

1.4 ປະໂຫຍດທີ່ຄາດວ່າຈະໄດ້ຮັບ

ໃນການຂຽນບົດຈົບຊັ້ນໃນຄັ້ງນີ້ ຫຼັງຈາກສຳເລັດໂຄງການນີ້ແລ້ວ Web Application ແມ່ນຕ້ອງໃຫ້ໄດ້ຮັບຜົນຕົວຈິງ ແລະ ສາມາດນຳເຂົ້າມາໃຊ້ໃນວຽກງານຕົວຈິງໄດ້ຢ່າງແນ່ນອນ.

- ໄດ້ລະບົບຈອງປຶ້ລິດເມແບບອອນລາຍຂອງສະຖານນີຂົນສົ່ງໂດຍສານສາຍໃຕ້.
- ໄດ້ລະບົບທີ່ຈະຊ່ວຍແກ້ໄຂບັນຫາການຈອງໄດ້ສະດວກ ແລະ ວ່ອງໄວຂຶ້ນ.
- ໄດ້ລະບົບຊ່ວຍເພີ່ມຊ່ອງທາງໃນການຂາຍປີ້ໃຫ້ກັບຜູ້ປະກອບການ.
- ມີລະບົບເຜີຍແຜ່.
- ໄດ້ລະບົບທີ່ສາມາດສ້າງລາຍງານໄດ້ຢ່າງສະດວກ ແລະ ຖືກຕ້ອງ.

ບົດທີ 2

ທົບທວນເອກະສານ ແລະ ບົດຄົ້ນຄວ້າທີ່ກ່ຽວຂ້ອງ

2.1 ທົບທວນທິດສະດີທີ່ກ່ຽວຂ້ອງ

2.1.1 ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບຖານຂໍ້ມູນ

1) ຄວາມໝາຍຂອງຖານຂໍ້ມູນ

Databases ຫຼື ຖານຂໍ້ມູນຄືກຸ່ມຂອງຂໍ້ມູນທີ່ຖືກເກັບລວບລວມໄວ້ໂດຍມີຄວາມສໍາພັນເຊິ່ງກັນ ແລະ ກັນໂດຍບໍ່ໄດ້ບັງຄັບວ່າຂໍ້ມູນທັງໝົດນີ້ຈະຕ້ອງເກັບໄວ້ໃນແຟມຂໍ້ມູນດຽວກັນ ຫຼື ແຍກເກັບຫຼາຍໆແຟມຂໍ້ມູນ.

ລະບົບຖານຂໍ້ມູນ ຄືລະບົບທີ່ລວບລວມຂໍ້ມູນຕ່າງໆທີ່ກ່ຽວຂ້ອງກັນເຂົ້າໄວ້ດ້ວຍກັນຢ່າງມີລະບົບ, ມີຄວາມສໍາພັນລະຫວ່າງຂໍ້ມູນຕ່າງໆທີ່ຊັດເຈນ. ໃນລະບົບຖານຂໍ້ມູນຈະປະກອບດ້ວຍແຟມຂໍ້ມູນຫຼາຍແຟມທີ່ມີຂໍ້ມູນກ່ຽວຂ້ອງກັນ, ສໍາພັນກັນເຂົ້າໄວ້ດ້ວຍກັນຢ່າງເປັນລະບົບ ແລະ ເປີດໂອກາດໃຫ້ຜູ້ໃຊ້ສາມາດໃຊ້ງານ ແລະ ຮັກສາປ້ອງກັນຂໍ້ມູນເຫຼົ່ານີ້ໄດ້ຢ່າງມີປະສິດທິພາບໂດຍມີຊອບແວທີ່ປຸງປະເພື່ອນຊື່ກາງລະຫວ່າງຜູ້ໃຊ້ ແລະ ໂປຣແກຣມຕ່າງໆທີ່ກ່ຽວຂ້ອງກັບການໃຊ້ຖານຂໍ້ມູນເອີ້ນວ່າລະບົບຈັດການຖານຂໍ້ມູນ ຫຼື DBMS (Databases Management System) ມີໜ້າທີ່ຊ່ວຍໃຫ້ຜູ້ໃຊ້ເຂົ້າເຖິງຂໍ້ມູນໄດ້ງ່າຍສະດວກ ແລະ ມີປະສິດທິພາບການເຂົ້າເຖິງຂໍ້ມູນຂອງຜູ້ໃຊ້ອາດເປັນການສ້າງຖານຂໍ້ມູນ, ການແກ້ໄຂຖານຂໍ້ມູນ ຫຼື ການຕັ້ງຄໍາຖາມເພື່ອໃຫ້ໄດ້ຂໍ້ມູນມາໂດຍຜູ້ໃຊ້ບໍ່ຈໍາເປັນຕ້ອງຮັບຮູ້ກ່ຽວກັບລາຍລະອາດພາຍໃນໂຄງສ້າງຂອງຖານຂໍ້ມູນ.

2) ຄຸນລັກສະນະຂອງຖານຂໍ້ມູນ

- ຫຼຸດຄວາມຊໍ້າຊ້ອນຂອງຖານຂໍ້ມູນໃຫ້ເໝືອນນ້ອຍທີ່ສຸດ (Minimum Redundancy).
- ມີຄວາມຖືກຕ້ອງສູງສຸດ (Maximum Integrity).
- ມີຄວາມເປັນອິດສະຫຼະຂອງຂໍ້ມູນ (Data Independence).
- ມີລະບົບຄວາມປອດໄພຂອງຂໍ້ມູນສູງ (High Degree of Data Security).
- ການຄວບຄຸມຖານຂໍ້ມູນຈະຢູ່ສ່ວນກາງ (Logically Centralized Control).

3) ຄວາມສໍາຄັນຂອງລະບົບຖານຂໍ້ມູນ

- ສາມາດຫຼຸດຄວາມຊໍ້າຊ້ອນຂອງຂໍ້ມູນໄດ້.
- ຫຼີກລ້ຽງຄວາມຂັດແຍ້ງຂອງຂໍ້ມູນໄດ້.
- ສາມາດໃຊ້ຂໍ້ມູນຮ່ວມກັນໄດ້.
- ສາມາດກຳນົດຄວາມເປັນມາດຕະຖານດຽວກັນຂອງຂໍ້ມູນໄດ້.
- ສາມາດກຳນົດລະບົບຄວາມປອດໄພຂອງຂໍ້ມູນໄດ້.

4) ສະຖາປັດຕະຍະກຳລະບົບຖານຂໍ້ມູນ

ສະຖາປັດຕະຍະກຳລະບົບຖານຂໍ້ມູນ ມີ 3 ລະດັບຄື:

- ລະດັບພາຍໃນ (Internal Level)
- ລະດັບລັກສະນະແນວຄິດ (Conceptual Level)
- ລະດັບພາຍນອກ (External Level)

ກ. ລະດັບພາຍໃນ (Internal Level)

ເປັນການນຳເອົາຂໍ້ມູນທີ່ໄດ້ຈາກລະດັບແນວຄິດມາວິເຄາະ ແລະ ອອກແບບໂດຍແປງໃຫ້ຢູ່ໃນຮູບແບບຂອງການຈັດການຖານຂໍ້ມູນ (DBMS) ທີ່ເລືອກໃຊ້ໂດຍເສີມແນວຄິດການເຮັດ Normalization Car Demoralization.

ຂ. ລະດັບລັກສະນະແນວຄິດ (Conceptual Level)

ເປັນການນຳເອົາຂໍ້ມູນທີ່ໄດ້ຈາກການວິເຄາະຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ຂໍ້ມູນໃນລະດັບພາຍນອກມາອອກແບບຖານຂໍ້ມູນເພື່ອໄດ້ໂຄງຮ່າງຂອງຖານຂໍ້ມູນໃນລະດັບແນວຄິດທີ່ປະກອບດ້ວຍໂຄງສ້າງຂອງຖານຂໍ້ມູນສ່ວນໜຶ່ງເປັນຄວາມສໍາພັນກັນ.

ຄ. ລະດັບພາຍນອກ (External Level)

ເປັນການນຳເອົາຂໍ້ມູນທີ່ໄດ້ຈາກລະດັບພາຍໃນມາກຳນົດໂຄງສ້າງຂໍ້ມູນ ແລະ ການຈັດເກັບວິທີການເຂົ້າເຖິງການຈັດການດ້ານລະບົບຄວາມປອດໄພຂອງຂໍ້ມູນເພື່ອຖານຂໍ້ມູນເຮັດວຽກໄດ້ຢ່າງມີປະສິດທິພາບ.


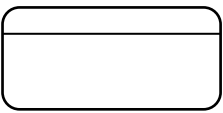
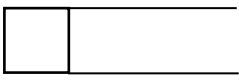
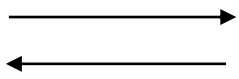

2.1.2 ທິດສະດີທີ່ກ່ຽວຂ້ອງກັບ DFD (Data Flow Diagram)

1) ຈຸດປະສົງຂອງ DFD

- ເປັນແຜນພາບທີ່ສະຫຼຸບລວມຂໍ້ມູນທັງໝົດໄດ້ຈາກການວິເຄາະໃນລັກສະນະຂອງຮູບແບບທີ່ເປັນໂຄງສ້າງ.
- ເປັນຂໍ້ຕົກລົງຮ່ວມກັນລະຫວ່າງນັກວິເຄາະລະບົບ ແລະ ຜູ້ຊົມໃຊ້.

- ເປັນແຜນພາບທີ່ໃຊ້ໃນການພັດທະນາຕໍ່ໃນຂັ້ນຕອນຂອງການອອກແບບ.
- ຮູ້ທີ່ໄປທີ່ມາຂອງຂໍ້ມູນທີ່ໄຫຼໃນຂະບວນການຕ່າງໆ.

2) ສັນຍະລັກທີ່ໃຊ້ໃນແຜນວາດການໄຫຼຂໍ້ມູນ

ຊື່	ສັນຍາລັກ	ຄວາມໝາຍ
Boundary Or External Entity		ຂອບເຂດໝາຍເຖິງພາກສ່ວນທີ່ກ່ຽວຂ້ອງກັບລະບົບ ເຊິ່ງລະບົບບໍ່ສາມາດຄວບຄຸມໄດ້
Process		ປະມວນຜົນ ຫຼື ຫນ້າວຽກທີ່ເຮັດໃນໂຄງການນັ້ນໆ
Data Store		ບ່ອນຈັດເກັບຂໍ້ມູນ
Data Flow		ການໄຫຼຂອງຂໍ້ມູນ
Real-Time Link		ການເຊື່ອມໂຍງແບບໄກທີ່ມີການຕອບກັບແບບທັນທີທັນໃດ

ຕາຕະລາງທີ 1: ສະແດງສັນຍາລັກ (Data Flow Diagram)

3) ກົດຂອງ Process

- ຕ້ອງບໍ່ມີຂໍ້ມູນເຂົ້າພຽງຢ່າງດຽວ.
- ຕ້ອງບໍ່ມີຂໍ້ມູນອອກພຽງຢ່າງດຽວ.
- ຂໍ້ມູນທີ່ສົ່ງເຂົ້າຕ້ອງພຽງພໍໃນການສ້າງຖານຂໍ້ມູນທີ່ສົ່ງອອກ.
- ການຕັ້ງຊື່ Process ຕ້ອງໃຊ້ຄຳກິລິຍາ (Verb) ເຊັ່ນ: ບັນທຶກຂໍ້ມູນ, ໃບບິນກວດສອບຂໍ້ມູນລູກຄ້າ, ຈຳນວນເງິນເດືອນ.

4) ຫຼັກການຂຽນແຜນວາດການໄຫຼຂໍ້ມູນ

ກ. Process:

- ເມື່ອມີຂໍ້ມູນເຂົ້າໄປທີ່ Process ກໍ່ຕ້ອງມີຂໍ້ມູນ ຫຼື ຜົນຮັບອອກມາຈາກ Process ເຊັ່ນກັນຈະເປັນໄປບໍ່ໄດ້ທີ່ມີສະເພາະຂໍ້ມູນເຂົ້າຢ່າງດຽວ.

ຂ. Data store:

- ຂໍ້ມູນຈະໄຫຼຈາກ Data Store ໜຶ່ງໄປຫາ Data Store ໜຶ່ງໂດຍກົງບໍ່ໄດ້ຈະຕ້ອງຜ່ານ Process ເທົ່ານັ້ນ.
- ຂໍ້ມູນທີ່ສົ່ງຜ່ານ External Entity ບໍ່ສາມາດໄຫຼເຂົ້າໄປ Data Store ໂດຍກົງໄດ້ຈະຕ້ອງໃຊ້ Process ເປັນຕົວກາງໃນການເຊື່ອມໂຍງເພື່ອຈັດເກັບຂໍ້ມູນໃນ Data Store.
- ຂໍ້ມູນທີ່ໄຫຼຜ່ານຈາກ Data Store ບໍ່ສາມາດເຊື່ອມໂຍງເຂົ້າກັບ External Entity ໄດ້ໂດຍກົງຈະຕ້ອງຜ່ານ Process ເທົ່ານັ້ນ.

ຄ. External Entity:

- External Entity ບໍ່ສາມາດເຊື່ອມໂຍງເຂົ້າຫາກັນໄດ້ຈະຕ້ອງໃຊ້ Process ເປັນຕົວກາງເພື່ອສົ່ງຜ່ານ ແລະ ຊື່ຂອງ External Entity ຈະໃຊ້ຄຳນາມເທົ່ານັ້ນ.

ງ. Data Flow:

- ການໄຫຼຂໍ້ມູນທີ່ມີຫົວຊີ້ໄປທີ່ Process ໝາຍເຖິງ Process ມີການອ່ານ ຫຼື ການດຶງຂໍ້ມູນຈາກ Data Store ມາໃຊ້ວຽກ.
- ການໄຫຼຂໍ້ມູນຈາກ Process ທີ່ມີຫົວລູກສອນຊີ້ໄປຍັງ Data Store ໝາຍເຖິງການ Update ຫຼື ການເພີ່ມຂໍ້ມູນລົງໄປທີ່ Data Store.
- ການໄຫຼຂໍ້ມູນທີ່ມີຫົວລູກສອນທັງສອງດ້ານທີ່ເຊື່ອມໂຍງລະຫວ່າງ Process ກັບ Data Store ໝາຍເຖິງມີການດຶງຂໍ້ມູນຈາກ Data Store ມາປັບປຸງ ແລະ ມີການ Update ຂໍ້ມູນລົງໄປໃນ Data Store.
- ການໄຫຼຂໍ້ມູນບໍ່ສາມາດຍ້ອນກັບໄປຍັງ Process ເດີມໄດ້ຢ່າງໜ່ວຍຕ້ອງເຊື່ອມໂຍງຜ່ານ Process ໜຶ່ງເພື່ອສົ່ງຜ່ານຍ້ອນກັບມາຍັງ Process ເດີມ ແລະ ຊື່ທີ່ລະບຸໃນການໄຫຼຂໍ້ມູນຈະໃຊ້ຄຳນາມ.

ອະນຸຍາດ	ບໍ່ອະນຸຍາດ

ຕາຕະລາງທີ 2 : ຮູບການປຸງທຽບແຜນວາດການໄຫຼຂໍ້ມູນທີ່ບໍ່ຖືກຕ້ອງ ແລະ ຖືກຕ້ອງ




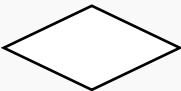



2.1.3 ຄວາມໝາຍ ແລະ ສັນຍະລັກຂອງ Flowchart

ສັນຍະລັກ Flowchart ຄືຮູບພາບທີ່ໃຊ້ແທນຄວາມໝາຍການເຮັດວຽກງານໃນລັກສະນະຕ່າງໆ ພາຍໃນແຜນຜັງ (Flowchart) ປະກອບໄປດ້ວຍການເລີ່ມຕົ້ນ (Start), ການຈົບ (End), ການກະທຳ (Process), ການນຳເຂົ້າຂໍ້ມູນ (Input), ການສະແດງຜົນຂໍ້ມູນ (Output), ການຕັດສິນໃຈ (Decision), ຄຳອະທິບາຍ (Annotation), ຈຸດເຊື່ອມຕໍ່ (Connector), ທິດທາງການເຮັດວຽກງານ (Direction Flow).

ສັນຍະລັກເຫຼົ່ານີ້ເມື່ອຖືກນຳມາເຊື່ອມຕໍ່ກັນຈະກາຍເປັນ "ແຜນຜັງ (Flowchart)" ທີ່ສະແດງລຳດັບຂັ້ນຕອນການເຮັດວຽກງານເພື່ອ

- ເປັນເຄື່ອງມືໃນການຈັດລຳດັບຄວາມຄິດ.
- ເຫັນລຳດັບຂັ້ນຕອນການເຮັດວຽກງານທີ່ຊັດເຈນ.

➤ ສັນຍະລັກ Flowchart

ຮູບພາບສັນຍະລັກ	ຄວາມໝາຍຂອງສັນຍະລັກ
	ການເລີ່ມຕົ້ນ ຫຼື ຈົບ Flowchart (Start ຫຼື End)
	ການກະທຳ (Process) ຖືກໃຊ້ເພື່ອສະແດງທີ່ການກະທຳໃນ Flowchart
	ຮັບຂໍ້ມູນ
	ການຕັດສິນໃຈ (Decision)
	ສະແດງຜົນທາງຈຳພາບ
	ຈຸດເຊື່ອມຕໍ່ (Connector)
	ທິດທາງການເຮັດວຽກງານ (Direction Flow)

ຕາຕະລາງທີ 3 : ສະແດງສັນຍະລັກ Flowchart

2.1.3 ແຜນວາດຄວາມສຳພັນຂອງ Entity (ER Diagram)

ການອອກແບບການສ້າງຕາຕະລາງຖານຂໍ້ມູນແບບຈຳລອງ ER ແບ່ງອອກເປັນ 2 ຂັ້ນຕອນຫຼັກຄື: ຂັ້ນຕອນທຳອິດເປັນການສ້າງແບບຈຳລອງ ER ຂັ້ນຕອນນີ້ຈະກ່າວເຖິງຄວາມໝາຍຂອງສັນຍະລັກຕ່າງໆທີ່ໃຊ້ໃນແບບຈຳລອງ ER, ຂັ້ນຕອນທີ 2 ແມ່ນການແປງແບບຈຳລອງ ER ໃຫ້ເປັນຕາຕະລາງຂໍ້ມູນເພື່ອໃຊ້ເປັນຕົວຈັດການຂໍ້ມູນ. ເມື່ອເຂົ້າໃຈໃນທັງສອງແບບນີ້ແລ້ວສາມາດອອກແບບຖານຂໍ້ມູນ ແລະ ສ້າງຖານຂໍ້ມູນໃນແບບຈຳລອງ ER ຂຶ້ນມາໃຊ້ງານດ້ວຍຕົວເອງ.

2.1.4 ສັນຍາລັກທີ່ໃຊ້ໃນແບບຈຳລອງ ER

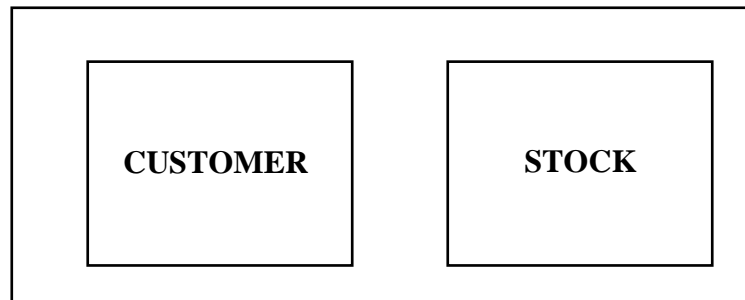
ການອອກແບບຈຳລອງຖານຂໍ້ມູນ ER ນັ້ນກ່ອນອື່ນຕ້ອງຮູ້ຈັກຄວາມໝາຍສັນຍາລັກຕ່າງໆ ທີ່ໃຊ້ໃນການອອກແບບຈຳລອງ ER ໄດ້ຢ່າງຖືກຕ້ອງ. ດັ່ງນັ້ນ ໃນຫົວຂໍ້ນີ້ຈະເວົ້າເຖິງຄວາມໝາຍ ແລະ ການໃຊ້ງານສັນຍາລັກຕ່າງໆຂອງ ER.

1) ເອັນຕິຕີ (Entity)

ເອັນຕິຕີ ຄືວັດຖຸທີ່ເຮົາສົນໃຈເຊິ່ງອາດເປັນໄດ້ທັງບຸກຄົນ, ສະຖານທີ່, ວັດຖຸ, ເຫດການ ຫຼື ແນວຄິດທີ່ກໍ່ໃຫ້ເກີດກຸ່ມຂອງຂໍ້ມູນທີ່ຕ້ອງການເອັນຕິຕີແບ່ງອອກເປັນ 2 ປະເພດຄື:

2) Strong Entity:

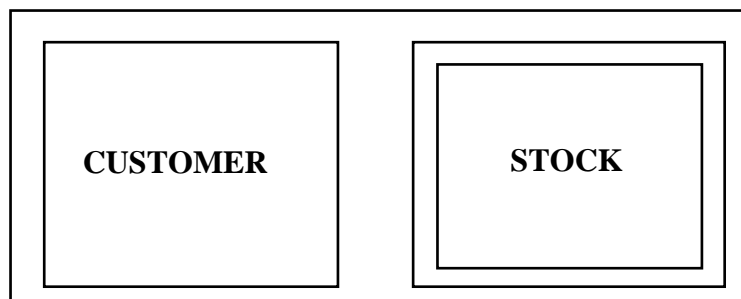
ເປັນເອັນຕິຕີທີ່ເກີດຂຶ້ນດ້ວຍຕົນເອງເປັນອິດສະຫຼະບໍ່ຂຶ້ນກັບເອັນຕິຕີໃດສັນຍາລັກທີ່ໃຊ້ຄື ຮູບສີ່ຫຼ່ຽມ ແລະ ສາມາດເອີ້ນ Strong Entity ໄດ້ອີກຊື່ໜຶ່ງວ່າ Regular Entity.



ຮູບທີ 1 : ຮູບ Strong Entity

3) Weak Entity:

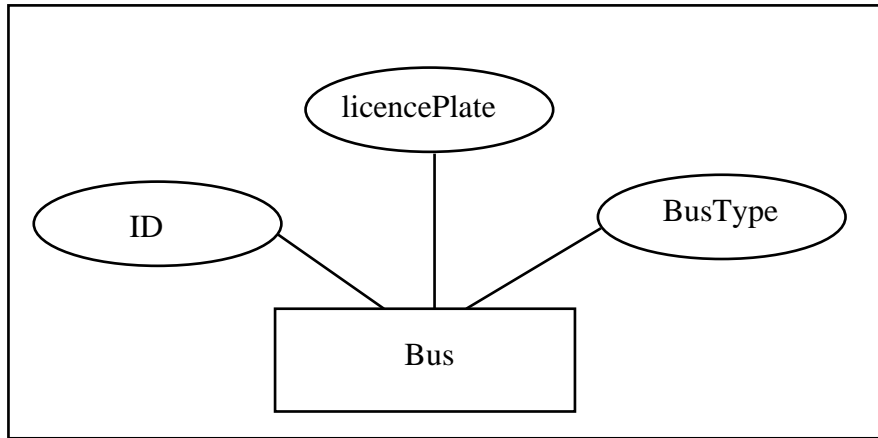
ເອັນຕິຕີຊະນິດນີ້ຈະຂຶ້ນກັບເອັນຕິຕີຊະນິດອື່ນໆບໍ່ສາມາດເກີດຂຶ້ນໄດ້ຕາມລຳພັງ ແລະ ຈະຖືກລົບເມື່ອເອັນຕິຕີຫຼັກຖືກລົບອອກ ສັນຍາລັກທີ່ໃຊ້ຄືຮູບສີ່ຫຼ່ຽມຊ້ອນກັນ.



ຮູບທີ 2 : ຮູບ Weak Entity

4) ແອັດທິຣິບິວ (Attribute)

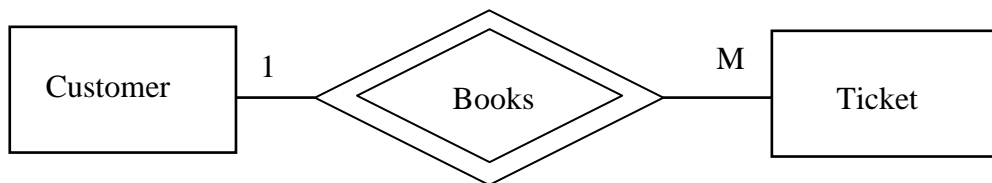
ແອັດທິຣິບິວຄືຄຸນສົມບັດຂອງສົມບັດຂອງເອັນຕິຕີສັນຍາລັກຂອງແອັດທິຣິບິວຈະເປັນຮູບວົງຮີແອັດທິຣິບິວໃດທີ່ຖືກໃຊ້ເປັນຄືຫຼັກຈະຖືກຂີດເສັ້ນກ້ອງກຳກັບໄວ້.



ຮູບທີ 3 : ຮູບ Attribute

5) ຄວາມສຳພັນ (Relation)

ຄວາມສຳພັນໃນທີ່ນີ້ ໝາຍເຖິງຄວາມສຳພັນລະຫວ່າງເອັນຕິຕີ ໂດຍແຕ່ລະຄວາມສຳພັນຄວນມີຊື່ລະບຸໄວ້ເພື່ອໃຊ້ອະທິບາຍເຊິ່ງປົກກະຕິຈະໃຊ້ສັນຍາລັກຮູບດອກຈັນທີ່ພາຍໃນລະບຸຄຳກິລິຍາໄວ້ເພື່ອອະທິບາຍຄວາມສຳພັນ.



ຮູບທີ 4 : ຮູບ Relationship

2.1.5 ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບພາສາ NOSQL

ຫຼາຍຄົນຄົງພໍຈະເຄີຍໄດ້ຍິນກັນມາແຕ່ກ່ຽວກັບເທັກໂນໂລຢີ ການຈັດການຂໍ້ມູນແບບໃໝ່ນີ້ ຊຶ່ງກໍຄື NoSQL ເມື່ອເວົ້າເຖິງ NoSQL ຈະໄດ້ຍິນຊື່ເວັບໄຊທີ່ໃຫຍ່ໆເຊັ່ນ: Facebook, Twitter, FourSquare, Digg ແລະ ອື່ນໆ. ເຮັດໃຫ້ເຮົາຮັບຮູ້ວ່າ NoSQL ເປັນລະບົບຖານຂໍ້ມູນສຳລັບງານທີ່ຕ້ອງຮອງຮັບຂໍ້ມູນຂະໜາດໃຫຍ່ໆ ຮອງຮັບການຂະຫຍາຍລະບົບໄດ້ງ່າຍເປັນຕົ້ນ.

ຊຶ່ງກໍເປັນເຊັ່ນນັ້ນແທ້ແຕ່ວຽກທີ່ນ້ອຍໆຈະເຮັດຢ່າງໃດໃຊ້ງານໄດ້ບໍ່ຄຸ້ມຄ່າທີ່ຈະນຳ NoSQL ມາໃຊ້ງານຂະໜາດນ້ອຍ ຫຼື ບໍ່ ຫຼືໃຊ້ Relational Database ກໍພຽງພໍແລ້ວ ຄຳຕອບຄືຂຶ້ນຢູ່ກັບ ລັກສະນະໃນການນຳມາໃຊ້ງານ ກ່ອນທີ່ຈະຕອບຄຳຖາມວ່າ NoSQL ເປັນຄຳຕອບຂອງລະບົບ ຈັດເກັບຂໍ້ມູນຂອງຫຼື ບໍ່ ລອງພິຈາລະນາຫົວຂໍ້ຕ່າງໆດັ່ງຕໍ່ໄປນີ້:

➤ ຜູ້ໃຊ້ນັບມື້ນັບຫຼາຍ (BigUsers)

ຈະເຫັນໄດ້ວ່າໃນຊ່ວງເວລາທີ່ຜ່ານມານີ້ ແລະ ໃນປັດຈຸບັນຜູ້ທີ່ໃຊ້ງານ Internet ມີແນວ ໂມ້ມຫຼາຍຂຶ້ນເລື້ອຍໆ ບໍ່ວ່າຈະໃຊ້ງານຜ່ານ Desktop PC ຫຼື Smartphone ຊຶ່ງເທັກໂນໂລຊີຂອງ ອຸປະກອນ (Devices) ມີຫຼາຍຂຶ້ນ ແລະ ໃຊ້ງານໄດ້ງ່າຍຂຶ້ນ.

ການພັດທະນາລະບົບສາມາດຮອງຮັບປະລິມານການເຂົ້າໃຊ້ງານແຕ່ລະອຸປະກອນ (Devices) ເປັນສິ່ງໜຶ່ງທີ່ຕ້ອງນຳມາພິຈາລະນາ ແລະ ບໍ່ພຽງແຕ່ຕ້ອງຮອງຮັບການເຂົ້າໃຊ້ງານ ຂອງຜູ້ໃຊ້ໄດ້ເທົ່ານັ້ນ ເຮົາຕ້ອງຮອງຮັບວິທີການປ້ອນຂໍ້ມູນແບບໃໝ່ຄືແຕ່ກ່ອນຜູ້ຈັດການເນື້ອຫາ ຕ່າງໆ ຄື Web Master, Web Editor, ຜູ້ເບິ່ງແຍງລະບົບເປັນຕົ້ນ. ແຕ່ປະຈຸບັນຜູ້ທີ່ປ້ອນຂໍ້ມູນຄືຜູ້ ໃຊ້ບໍລິການ (Users) ໂດຍກົງຜ່ານອຸປະກອນ (Devices) ຕ່າງໆທີ່ມີຫຼາກຫຼາຍ ແລະ ການປ້ອນ ຂໍ້ມູນກໍງ່າຍກວ່າແຕ່ກ່ອນອີກດ້ວຍ.

ຍັງມີປັດໄຈອື່ນໆເຊັ່ນ ເທດສະການສຳຄັນໆທີ່ຄົນຈະເຂົ້າມາໃຊ້ງານຫຼາຍເປັນພິເສດ ຫຼື ຜູ້ ໃຊ້ງານທີ່ບໍ່ແມ່ນແຕ່ປະເທດເຮົາເທົ່ານັ້ນເພາະໂລກອິນເຕີເນັດເຖິງກັນ ອາດຈະຕ້ອງເບິ່ງວ່າລະບົບ ເຮົາມີຜູ້ເຂົ້າໃຊ້ງານຈາກຕ່າງປະເທດ ຫຼື ທົ່ວໂລກ ຫຼື ບໍ່ອີກດ້ວຍ.

ດັ່ງນັ້ນ ພວກເຮົາຕ້ອງກັບມາທົບທວນວິທີການຈັດການຖານຂໍ້ມູນແລ້ວວ່າບໍ່ແມ່ນແຕ່ເຮັດ ໃຫ້ຮອງຮັບກັບການເຂົ້າມາໃຊ້ງານຂອງຜູ້ໃຊ້ບໍລິການເທົ່ານັ້ນແຕ່ຕ້ອງຮອງຮັບການຈັດເກັບຂໍ້ມູນທີ່ ຫຼາຍຂຶ້ນເລື້ອຍໆໄດ້ອີກ.

1. ປະເພດຂໍ້ມູນຕ່າງໆ ແລະ ຂໍ້ມູນທີ່ຕ້ອງການຈັດເກັບຫຼາຍຂຶ້ນເລື້ອຍໆ (BigData)

ຈາກຕົວແປຂອງຜູ້ໃຊ້ງານມີຫຼາຍຂຶ້ນເລື້ອຍໆ ອຸປະກອນໃນການເຂົ້າໃຊ້ງານກໍຫຼາກຫຼາຍ ປະເພດຂອງຂໍ້ມູນທີ່ໄດ້ຈາກແຕ່ລະອຸປະກອນກໍຫຼາກຫຼາຍປະເພດ ເຊັ່ນ ຂໍ້ຄວາມ, ຮູບພາບ, ສຽງ, ວິດີໂອ, ຕຳແໜ່ງສະຖານທີ່ (GeoLocation) ແລະ ອື່ນໆ. ການປ້ອນຂໍ້ມູນເຫຼົ່ານີ້ກໍງ່າຍແສນໆ ເພາະເທັກໂນໂລຊີຂອງຮາດແວຣ໌ ແລະ ຊອບແວຣ໌ມີການພັດທະນາຫຼາຍຂຶ້ນເລື້ອຍໆ ໃຊ້ງານງ່າຍ ຂຶ້ນສະດວກຂຶ້ນວ່າໄວຂຶ້ນເລື້ອຍໆ.

ດັ່ງນັ້ນ ການຈັດເກັບຂໍ້ມູນທີ່ລົງໄຫຼເຂົ້າມາຈາກອຸປະກອນຕ່າງໆເຫຼົ່ານີ້ ເຮົາອາດຈະຕ້ອງ ນຳມາວິເຄາະພຶດຕິກຳຂອງຜູ້ໃຊ້ບໍລິການ ການສົ່ງເສີມການຕະຫຼາດ ເຮັດຂໍ້ມູນການຕັດສິນໃຈຂອງ ຜູ້ບໍລິຫານ, ຂໍ້ມູນລູກຄ້າສຳພັນ ແລະ ອື່ນໆອີກຫຼວງຫຼາຍການນຳລະບົບຖານຂໍ້ມູນແບບເດີມ (Relational Database) ອາດຈະບໍ່ເໝາະກັບລັກສະນະງານບາງຢ່າງອີກຕໍ່ໄປ.

2. ເຕັກໂນໂລຢີຮາດແວໄດ້ປ່ຽນແປງລາຄາຖືກຫຼຸດລົງແຕ່ວ່າປະສິດທິພາບດີຂຶ້ນ (Cloud Computing).

ເຮົາອາດຈະເລີ່ມໄດ້ຍິນຄຳວ່າ Cloud Technology ຊຶ່ງກໍມີຫຼາຍປະເພດ ແຕ່ໃນນີ້ເຮົາຈະເວົ້າເຖິງໃນແງ່ຂອງການນຳມາໃຊ້ງານໂດຍປະຈຸບັນຖ້າໃຜເຄີຍໄດ້ລອງໃຊ້ງານ EC2 ຂອງ Amazon ມາແດ່ແລ້ວຈະຮູ້ຈັກເປັນຢ່າງດີວ່າການຈະມີເຄື່ອງ Server ແຮງໆຈັກເຄື່ອງເປັນເລື່ອງທີ່ງ່າຍຫຼາຍ ຫຼື ການຈະມີ Server 10 ເຄື່ອງ ຫຼື 20 ເຄື່ອງ ນຳມາຕໍ່ເປັນ Database Cluster ນັ້ນງ່າຍຫຼາຍພຽງແຕ່ຄລິກສ້າງ Instance ບໍ່ຈັກເທື່ອກໍໄດ້ Server ມາໃຊ້ງານແລ້ວ ແລະ ລາຄາກໍຖືກຫຼາຍ ຖ້າເຮົາບໍ່ໃຊ້ງານແລ້ວກໍຍົກເລີກການໃຊ້ງານ ແລະ ຄືນກັບໄປໄດ້ທັນທີ ເມື່ອທຽບກັບສະໄໝກ່ອນທີ່ເຮົາຕ້ອງການມີ Server ຈັກ 10 ເຄື່ອງ ຈະຕ້ອງລົງທຶນຊື້ເຄື່ອງມາຫຼາຍ ຖ້າເຮົາໃຊ້ງານແລ້ວຕ້ອງແບກຮັບພາລະເຄື່ອງ Server ເຫຼົ່ານີ້ໄວ້ຊຶ່ງເປັນຕົ້ນທຶນທີ່ແພງຫຼາຍ.

ຈາກທີ່ກ່າວມາຂ້າງເທິງເຮົາບໍ່ໄດ້ເນັ້ນຂໍ້ດີຂອງ EC2 ແຕ່ຢ່າງໃດແຕ່ກຳລັງຈະເນັ້ນວ່າພາບລວມຂອງການໃຊ້ງານ Server ເລີ່ມປ່ຽນໄປຄືໃຊ້ງານໄດ້ງ່າຍຂຶ້ນ, ລາຄາຖືກລົງ ແຕ່ປະສິດທິພາບດີຂຶ້ນຊຶ່ງເປັນສິ່ງທີ່ສຳຄັນໃນການນຳມາພິຈາລະນາຂອງເທັກໂນໂລຢີດ້ານຖານຂໍ້ມູນຄືຖ້າຕ້ອງການຈັດເກັບຖານຂໍ້ມູນໃຫຍ່ໆ ຫຼື ຮອງຮັບຜູ້ໃຊ້ງານໄດ້ເປັນຈຳນວນຫຼາຍ, ການຂະຫຍາຍລະບົບຖານຂໍ້ມູນເປັນເລື່ອງທີ່ງ່າຍຂຶ້ນ ຊຶ່ງເຮັດໄດ້ໂດຍການເອົາ Server ມາຕັ້ງກັນອອກໄປ ຫຼື ເອີ້ນວ່າການຂະຫຍາຍອອກແນວນອນ (Scale Out) ບໍ່ແມ່ນການຂະຫຍາຍລະບົບຄືແຕ່ກ່ອນ ຄືຂະຫຍາຍອອກແນວຕັ້ງ (Scale Up) ແລະ ຕ້ອງໃຊ້ເຄື່ອງ Server ທີ່ມີປະສິດທິພາບສູງ ຊຶ່ງຈະມີຕົ້ນທຶນທີ່ແພງກວ່າການຂະຫຍາຍແບບແນວນອນຫລາຍ.

ດັ່ງນັ້ນ ການຂະຫຍາຍລະບົບທີ່ຢູ່ເທິງພື້ນຖານຂອງ NoSQL ຄືຮອງຮັບການຂະຫຍາຍລະບົບແບບແນວນອນ (Scale Out) ຊຶ່ງຈະກະຈາຍຂໍ້ມູນໄປເກັບທີ່ເຄື່ອງ Server ຫຼາຍເຄື່ອງ ແລະ ໃຊ້ເຄື່ອງ Server ທົ່ວໄປທີ່ເອີ້ນວ່າ (Commodity Server) ບໍ່ຈຳເປັນຕ້ອງໃຊ້ Server ທີ່ເອີ້ນວ່າ Enterprise Server ທີ່ມີລາຄາແພງຕາມ Spec ທີ່ສູງຂຶ້ນເລື້ອຍໆ ແລະ ການບໍລິຫານຈັດການກໍຍາກຂຶ້ນອີກດ້ວຍ.

3. ບັນຫາຂອງ Relational Database

ຖ້າໃຜທີ່ໃຊ້ງານ Relation Database ທີ່ຕ້ອງການຮອງຮັບການຈັດເກັບຂໍ້ມູນຂະໜາດໃຫຍ່ ຕ້ອງຫຼີກລ້ຽງບໍ່ໄດ້ເລື່ອງການເຮັດ Sharding ແລະ Distributed Cache ເພາະເປັນຕົວຫຼັກທີ່ຕ້ອງເຮັດເພື່ອຂະຫຍາຍລະບົບຖານຂໍ້ມູນຂອງ Relational Database ໃຫ້ສາມາດຮອງຮັບຂໍ້ມູນທີ່ຫຼາຍຂຶ້ນ ແລະ ຮອງຮັບຈຳນວນການເຂົ້າມາໃຊ້ງານລະບົບໄດ້ຫຼາຍຂຶ້ນ.

4. Manual Sharding

ການແບ່ງຕາຕະລາງຖານຂໍ້ມູນ (Table) ອອກເປັນສ່ວນແລ້ວກໍກະຈາຍໄປຈັດເກັບໃນຫຼາຍໆ Server ເພື່ອໃຫ້ແຕ່ລະຕາຕະລາງ (Table) ຂອງຖານຂໍ້ມູນບໍ່ຈັດເກັບຂໍ້ມູນທີ່ຫຼາຍເກີນໄປ ເພາະຖ້າຂໍ້ມູນໃນແຕ່ລະຖານຂໍ້ມູນຫຼາຍເກີນໄປຈະເຮັດໃຫ້ລະບົບຖານຂໍ້ມູນຊ້າແຕ່ບັນຫາກໍຈະຕາມ

ມາອີກຄື ເມື່ອຕ້ອງກະຈາຍຂໍ້ມູນອອກໄປໃນແຕ່ລະ Server ການຈະເກັບຂໍ້ມູນ ເຊັ່ນ: ເພີ່ມ, ແກ້ໄຂ, ລຶບ, ດຶງຂໍ້ມູນມາສະແດງຕ່າງໆ ຈະຕ້ອງເຮັດຜ່ານ Application ຫຼື ຕ້ອງມີ Server ບາງໂຕທີ່ຖ້າດຶງຂໍ້ມູນແຕ່ລະ Server ມາທັງໝົດເປັນກ້ອນດຽວ ນັ້ນໝາຍຄວາມວ່າເຮົາຕ້ອງເຮັດດ້ວຍໂຕເຮົາເອງ ບໍ່ແມ່ນລະບົບຖານຂໍ້ມູນຈັດການໃຫ້ (Manual Sharding).

5) Distributed Cache

ເມື່ອເຮົາຕ້ອງການໃຫ້ລະບົບຮອງຮັບການເຂົ້າມາໃຊ້ງານຫຼາຍໄດ້ນັ້ນ ຖ້າຈະຕ້ອງເຂົ້າມາອ່ານຂໍ້ມູນຜ່ານ Database ໂດຍກົງມັນອາດຈະຮອງຮັບບໍ່ໄຫວ ຫຼື ເຮັດໄດ້ຊ້າ ດັ່ງນັ້ນ ຈະຕ້ອງມີການເຮັດ Cache Layer ຂຶ້ນມາ ຄືແທນທີ່ຈະເຂົ້າໄປອ່ານຈາກຖານຂໍ້ມູນໂດຍກົງກໍໃຫ້ອ່ານຜ່ານ Cache ກ່ອນ ດັ່ງນັ້ນ ການອ່ານຂໍ້ມູນຈາກ Cache ເປັນການອ່ານຈາກ Memory ໂດຍກົງເຮັດໃຫ້ຮອງຮັບປະລິມານການເຂົ້າມາໃຊ້ງານໄດ້ຫຼາຍຂຶ້ນ.

ແຕ່ບັນຫາຄືການເຮັດ Cache Layer ນີ້ຮອງຮັບສະເພາະການອ່ານຂໍ້ມູນເທົ່ານັ້ນ, ບໍ່ຮອງຮັບການຂຽນຂໍ້ມູນໄດ້ ຖ້າຫາກຕ້ອງການຮອງຮັບການຂຽນຂໍ້ມູນປະລິມານຫຼາຍ ແລະ ອ່ານຂໍ້ມູນປະລິມານຫຼາຍຈຶ່ງເປັນສິ່ງທີ່ Relational Database ບໍ່ສາມາດຮອງຮັບງານໃນລັກສະນະອ່ານ, ຂຽນຂໍ້ມູນປະລິມານຫຼາຍໄດ້ດີ ແລະ ສິ່ງສໍາຄັນການເຮັດ Cache Layer ຈະຕ້ອງມີການດູແລຮັກສາ ແລະ ໃຊ້ Server ແຍກອອກໄປຕ່າງຫາກ.

ຈາກຈຸດນີ້ການເຮັດ Sharding ແລະ Caching ເປັນສິ່ງທີ່ຖືກພັດທະນາຂຶ້ນໃນ NoSQL ເທັກໂນໂລຢີ ໂດຍຮອງຮັບ Auto-Sharding ແລະ Integrated Caching ໃນຕົວເອງ ດັ່ງນັ້ນ ເຮົາຈຶ່ງໄດ້ເຫັນ NoSQL ຖືກນໍາໄປໃຊ້ງານກັບລະບົບໃຫຍ່ໆເຊັ່ນ: Facebook, Twitter, FourSquare, Digg ແລະ ອື່ນໆເພາະວ່າ NoSQL ອອກແບບມາເພື່ອຮອງຮັບຄວາມຕ້ອງການງານໃຫຍ່ໆໄດ້ດີ ໂດຍສະເພາະຢູ່ແລ້ວ ແຕ່ເຖິງຢ່າງໃດກໍຕາມຍັງມີຄຸນສົມບັດອື່ນໆທີ່ໜ້າສົນໃຈໃນ NoSQL ເທັກໂນໂລຢີ.

ກ. ຄຸນສົມບັດຂອງ NoSQL Database

➤ Dynamic Schemas

ການຈັດເກັບຂໍ້ມູນຕ່າງໆໃນຖານຂໍ້ມູນແບບ Relational Database ເຮົາຈະຕ້ອງມີການສ້າງ Schema ຫຼື ຮູບແບບຂອງໂຄງສ້າງຕາຕະລາງວ່າຈະຈັດເກັບຂໍ້ມູນຫຍັງເມື່ອຕ້ອງການຈັດເກັບຂໍ້ມູນເພີ່ມເຕີມຕ້ອງປ່ຽນ Schema ພາຍຫຼັງ (Alter-Table) ກ່ອນຈະຈັດເກັບຂໍ້ມູນຮູບແບບໃໝ່ໄດ້.

ແຕ່ໃນປະຈຸບັນການຈັດເກັບຂໍ້ມູນມີການປ່ຽນແປງຕະຫຼອດເວລາເພາະຄວາມຕ້ອງການຈັດເກັບຂໍ້ມູນຕ່າງໆມີຫຼາກຫຼາຍຂຶ້ນເລື້ອຍໆ ການກໍານົດໂຄງສ້າງຂອງຕາຕະລາງຖານຂໍ້ມູນ ຫຼື ການຕ້ອງປ່ຽນໂຄງສ້າງຖານຂໍ້ມູນເລື້ອຍໆໂດຍທີ່ຂໍ້ມູນຍັງມີຢູ່ແລ້ວເປັນເລື່ອງທີ່ຍາກຫຼາຍ ຫຼື ເຮັດບໍ່ໄດ້ເລີຍ, ວິທີການຄືອາດຈະຕ້ອງແຍກອອກເປັນຕາຕະລາງໃໝ່ຊຶ່ງເປັນວິທີແກ້ບັນຫາຊ່ວຍເຫຼືອເທົ່ານັ້ນ.

ລະບົບຖານຂໍ້ມູນແບບ NoSQL ເຮົາບໍ່ຈຳເປັນຕ້ອງມີ Schema ທີ່ຕາຍຕົວ ຫຼື ບໍ່ຕ້ອງມີ Schema ກ່ອນທີ່ຈະຈັດເກັບຂໍ້ມູນ ຂໍ້ມູນແຕ່ລະແຖວສາມາດຈັດເກັບໄດ້ຕາມຕ້ອງການຈະເພີ່ມ ຫຼື ຫຼຸດກໍບໍ່ມີບັນຫາກັບລະບົບເຮັດໃຫ້ເຮົາສາມາດຈັດເກັບຂໍ້ມູນໄດ້ຕາມທີ່ຕ້ອງການປ່ຽນແປງໄດ້ ຕະຫຼອດເວລາສະດວກ ແລະ ວ່ອງໄວ.

- Auto - Sharding

ເມື່ອຂໍ້ມູນມີຂະໜາດໃຫຍ່ ຫຼື ເຮົາຕ້ອງການເພີ່ມປະສິດທິພາບການອ່ານ ແລະ ຂຽນຂໍ້ມູນ ປະລິມານຫຼາຍ, ການເຮັດ Sharding ໃນລະບົບ NoSQL Database ຈະກໍ່ການກະຈາຍຂໍ້ມູນໄປ ຫາ Server ອັດຕະໂນມັດ (Auto-Sharding) ຜູ້ພັດທະນາ (Developer) ບໍ່ຕ້ອງຂຽນໂປຣແກຣມໃນການກະຈາຍຂໍ້ມູນເອງຄືກັບ Relational Database.

ການກະຈາຍຂໍ້ມູນອອກໄປຫຼາຍໆ Server ນີ້ຍັງເຮັດໃຫ້ມີຂໍ້ດີປະຢັດຕົ້ນທຶນໃນການຂະຫຍາຍລະບົບ ເພາະເປັນການຂະຫຍາຍແບບແນວນອນ (Scale Out) ຊຶ່ງສາມາດນຳ Server ປົກກະຕິທີ່ໄປມາໃຊ້ງານໄດ້ ບໍ່ຈຳເປັນຕ້ອງເປັນ Enterprise Server

- Replication

ການສຳເນົາຂໍ້ມູນຈາກເຄື່ອງໜຶ່ງໄປອີກເຄື່ອງໜຶ່ງ (Replication) ເມື່ອ Server ໜຶ່ງເສຍຫາຍອີກເຄື່ອງໜຶ່ງຈະຂຶ້ນມາເຮັດວຽກແທນທັນທີໂດຍຂໍ້ມູນຂອງແຕ່ລະເຄື່ອງຈະມີຂໍ້ມູນຄືກັນ ດັ່ງນັ້ນ Replication ເປັນໜຶ່ງຄຸນສົມບັດທີ່ຕອບສະໜອງຕໍ່ການໃຊ້ງານທີ່ຕ້ອງການຄວາມຕໍ່ເນື່ອງໄດ້ ຕະຫຼອດເວລາ (High Availability).

- Integrated Caching

ການຈັດເກັບຂໍ້ມູນທີ່ໃຊ້ງານເລື້ອຍໆເຂົ້າໄວ້ໃນ Memory (RAM) ຊຶ່ງເປັນຄຸນສົມບັດເດັ່ນຂອງ NoSQL ທີ່ທັງຫມົດ Caching ໄວ້ໃນຕົວເອງຢູ່ແລ້ວເຮົາບໍ່ຈຳເປັນຕ້ອງເຮັດ Cache Layer ຄືກັບ Relational Database ອີກຕໍ່ໄປທີ່ຕ້ອງເຮັດ Cache Layer ແຍກຕ່າງຫາກ ແລະ ເບິ່ງແຍງຮັກສາແຍກອອກໄປຕ່າງຫາກອີກ.

ຂ. ປະເພດຂອງຖານຂໍ້ມູນ NoSQL

NoSQL ຖືກແບ່ງປະເພດຕາມລັກສະນະການຈັດເກັບຂໍ້ມູນທີ່ແຕກຕ່າງກັນ. ດັ່ງນັ້ນ ການຈະເລືອກ NoSQL Database ໂຕໃດໂຕໜຶ່ງຈະຕ້ອງເບິ່ງອີກວ່າການຈັດເກັບຂໍ້ມູນຂອງຖານຂໍ້ມູນເປັນແບບໃດເຊັ່ນ:

- Document Databases ເຊັ່ນ MongoDB, CouchDB, Elasticsearch
- Graph Stores ເຊັ່ນ Neo4J, Infinite Graph, InfoGrid
- Key-value stores ເຊັ່ນ DynamoDB, Redis, MemcacheDB
- Wide-Column Stores ເຊັ່ນ Cassandra, Amazon SimpleDB, Hadoop / HBase

ຄ. Open Source License

ໂດຍສ່ວນໃຫຍ່ແລ້ວ NoSQL ຈະເປັນລິຂະສິດແບບ Open source ຊຶ່ງບໍ່ຕ້ອງເສຍຄ່າໃຊ້ຈ່າຍໃນການນຳມາໃຊ້ງານ ດັ່ງນັ້ນ ເຮົາສາມາດນຳ NoSQL Database ແຕ່ລະຕົວມາຕິດຕັ້ງໃຊ້ງານໄດ້ໂດຍບໍ່ເສຍຄ່າໃຊ້ຈ່າຍໃດໆ (ຟຣີ).

ງ. ນຳ NoSQL ມາໃຊ້ງານຂະໜາດນ້ອຍໄດ້ ຫຼື ບໍ່?

ຈາກທີ່ກ່າວມາແລ້ວ ຄົງພໍຈະຕອບຄຳຖາມນີ້ໄດ້ວ່າການນຳ NoSQL Database ເມື່ອນຳມາໃຊ້ໃນງານຂະໜາດໃຫຍ່ນັ້ນເພາະສົມຢ່າງແນ່ນອນ, ແຕ່ຖ້າເປັນລະບົບທີ່ວາງໄປຄວນຈະນຳ NoSQL ມາໃຊ້ງານ ຫຼື ບໍ່.

ຄຳຕອບຄື ຂຶ້ນຢູ່ກັບລັກສະນະວຽກວ່າເຮົາຈະໃຊ້ຄຸນສົມບັດຫຍັງຂອງ NoSQL ຖ້າເຮົາຕ້ອງການຈັດເກັບຂໍ້ມູນທີ່ບໍ່ຕ້ອງຍຶດຕິດກັບໂຄງສ້າງ (Dynamic Schema) ແລະ ຕ້ອງການເຂົ້າໃຊ້ງານລະບົບທີ່ວ່ອງໄວ (Integrated Caching) ຂໍ້ມູນອາດຈະຍັງບໍ່ຫຼາຍເທົ່າໃດອາດຈະໃຊ້ NoSQL ໄດ້ຢ່າງແນ່ນອນ.

ແຕ່ຖ້າຫາກບໍ່ຕ້ອງການໃຊ້ງານ (Dynamic Schema) ບໍ່ຕ້ອງການເຂົ້າໃຊ້ງານທີ່ວ່ອງໄວ (Integrated Caching) ເພາະໃຊ້ Relational Database ກໍເຮັດໄດ້ດີຢູ່ແລ້ວ Database Server ກັບ Web Server ກໍຢູ່ທີ່ Server ດຽວກັນ, ຂໍ້ມູນບໍ່ຫຼາຍນັ້ນບໍ່ຕ້ອງການຈັດເກັບຂໍ້ມູນທີ່ເພີ່ມຂະຫຍາຍຂຶ້ນທຸກມື້ຜູ້ເຂົ້າໃຊ້ງານກໍບໍ່ໄດ້ຫຼາຍເບິ່ງແລ້ວວ່າລະບົບບໍ່ມີແນວໂນ້ມຈະຕ້ອງຂະຫຍາຍລະບົບໃນອະນາຄົດອັນໃກ້ຈະສາມາດໃຊ້ງານ Relational Database ໄດ້ດີຢູ່ແລ້ວຢ່າງບໍ່ມີບັນຫາ.

2.1.6 ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບພາສາ JavaScript

ພາສາ JavaScript ຫຼື ຫຍໍ້ JS ເປັນພາສາຂຽນໂປຣແກຣມທີ່ຖືກພັດທະນາ ແລະ ປະຕິບັດຕາມຂໍ້ກຳນົດມາດຕະຖານຂອງ ECMAScript, ພາສາ JavaScript ນັ້ນເປັນພາສາລະດັບສູງຄອມພາຍໃນຂະນະທີ່ໂປຣແກຣມຮັ້ນ (JIT) ແລະ ເປັນພາສາຂຽນໂປຣແກຣມແບບຫຼາຍຂະບວນເຊັ່ນ: ການຂຽນໂປຣແກຣມແບບຂັ້ນຕອນ, ການຂຽນໂປຣແກຣມແບບວັດຖຸ, ພາສາ JavaScript ມີໄວຍະກອນທີ່ຄືກັບພາສາ C ໃຊ້ວົງເລັບເພື່ອກຳນົດບ່ອກຂອງຄຳສັ່ງ ນອກຈາກນີ້ JavaScript ຍັງເປັນພາສາທີ່ມີປະເພດຂໍ້ມູນແບບໄດນາມິກ (Dynamic) ເປັນພາສາແບບ Prototype-Based ແລະ First-Class Function.

ພາສາ JavaScript ນັ້ນຖືວ່າເປັນເທັກໂນໂລຊີຫຼັກຂອງການພັດທະນາເວັບໄຊ (World Wide Web) ມັນເຮັດໃຫ້ໜ້າເວັບສາມາດຕອບໂຕ້ກັບຜູ້ໃຊ້ໄດ້ໂດຍທີ່ບໍ່ຈຳເປັນຕ້ອງລືເຟດໜ້າໃໝ່ (Dynamic website) ເວັບໄຊຈຳນວນຫຼາຍໃຊ້ພາສາ JavaScript ສຳລັບຄວບຄຸມການເຮັດວຽກງານທີ່ດ້ານ Client-side ນັ້ນເຮັດໃຫ້ເວັບບຣາວເຊີຕ່າງໆ ມີ JavaScript Engine ທີ່ໃຊ້ສຳລັບປະມວນຜົນສະຄິບຂອງພາສາ JavaScript ທີ່ຮັ້ນເທິງເວັບບຣາວເຊີເນື່ອງຈາກພາສາ JavaScript

ເປັນພາສາຂຽນໂປຣແກຣມແບບຫຼາຍຮູບແບບເຮັດໃຫ້ມັນຮອງຮັບການຂຽນໂປຣແກຣມທັງແບບ Event-Driven, Functional ແລະ ແບບລຳດັບຂັ້ນຕອນມັນມີ Library (APIs) ສຳລັບເຮັດວຽກກັບຂໍ້ຄວາມ, ວັນທີ, Regular Expression ແລະ ໂຄງສ້າງຂໍ້ມູນພື້ນຖານຢ່າງ Array ແລະ Map ຫຼື ທັງ Document Object Model (DOM) ຊຶ່ງເປັນ API ທີ່ໂດຍທົ່ວໄປແລ້ວສາມາດໄດ້ເທິງເວັບບຣາວເຊີ.

ຢ່າງໃດກໍຕາມຕົວຂອງພາສາ JavaScript ເອງບໍ່ໄດ້ມີຟັງຊັນສຳລັບອິນພຸດ/ເອົາພຸດ(I/O) ທີ່ມາກັບພາສາເຊັ່ນ: ຟັງຊັນກ່ຽວກັບ Network ວຽກກ່ຽວກັບໄຟລ ຫຼື Library ກ່ຽວກັບກຣາຟິກ ໂດຍທົ່ວໄປແລ້ວສິ່ງເຫຼົ່ານີ້ຈະຖືກໃຫ້ມາໂດຍ Host Environment (ສະພາບແວດລ້ອມທີ່ໃຊ້ຮັບພາສາ JavaScript) ເຊັ່ນ ເວັບເວັບບຣາວເຊີ ຫຼື Node.js ຊຶ່ງຈະແຕກຕ່າງກັນອອກໄປ ຕົວຢ່າງ ເຊັ່ນ: ການຮັບຄ່າໃນເວັບເວັບບຣາວເຊີຈະຜ່ານຟັງຊັນ Prompt ຊຶ່ງເປັນສ່ວນໜຶ່ງຂອງ Browser Object Model (BOM) ຫຼື ຮັບຄ່າຈາກ HTML ຟອມຊຶ່ງເປັນສ່ວນໜຶ່ງຂອງ Document Object Model (DOM) ໃນຂະນະທີ່ເທິງ Node.js ເຮົາສາມາດຮັບຄ່າໄດ້ຈາກ Input/Output Stream ຂອງ Command line ເຖິງແມ່ນວ່າມັນຈະມີຄວາມຄ້າຍຄືກັນລະຫວ່າງພາສາ Java ແລະ JavaScript ເຊັ່ນ: ຊື່ຂອງພາສາໄວຍະກອນ ຫຼື Library ມາດຕະຖານຕ່າງໆ ຢ່າງໃດກໍຕາມທັງສອງພາສາແຕກຕ່າງກັນຢ່າງສິ້ນເຊີງໃນແງ່ຂອງການອອກແບບພາສາ Java ເປັນພາສາທີ່ມີປະເພດຂໍ້ມູນແບບຄົງທີ່ (Static-Typing) ໃນຂະນະທີ່ພາສາ JavaScript ມີປະເພດຂໍ້ມູນແບບໄດນາມິກ (Dynamic-Typing) ພາສາ Java ຖືກຄອມພາຍເປັນ Byte-Code ກ່ອນທີ່ຈະລັນໃນຂະນະທີ່ພາສາ JavaScript ຈະຄອມພາຍໃນຕອນທີ່ໂປຣແກຣມຮັບພາສາ Java ເປັນພາສາແບບ Class-Based ໃນຂະນະທີ່ພາສາ JavaScript ເປັນພາສາແບບ Prototypebased.

1) ປະຫວັດຄວາມເປັນມາຂອງພາສາ JavaScript

ພາສາ JavaScript ຖືກອອກແບບແລະສ້າງໂດຍ Brendan Eich ສຳລັບເປັນພາສາສະຄິບທີ່ເຮັດວຽກເທິງເວັບບຣາວເຊີ Navigator ທີ່ເປັນຜະລິດຕະພັນຂອງບໍລິສັດ Netscape ເພື່ອເຮັດໃຫ້ໜ້າເວັບທີ່ໃນຕອນທຳອິດນັ້ນເປັນແບບ Static ສາມາດຕອບໂຕ້ກັບຜູ້ໃຊ້ໄດ້ໂດຍທີ່ບໍ່ຈຳເປັນຕ້ອງຮູ້ເຟດໜ້າໃໝ່ (Dynamic) ເຊັ່ນ: ການສົ່ງຂໍ້ມູນເບື້ອງໜຶ່ງໄປຍັງ Server ແລະ ລໍຖ້າຮັບຜົນຕອບກັບມາດ້ວຍ AJAX; ພາສາ JavaScript ໄດ້ຖືກເປີດຕົວ ແລະ ເປັນສ່ວນໜຶ່ງຂອງເວັບບຣາວເຊີ Navigator ໃນເດືອນກັນຍາ 1995 ໂດຍໃຊ້ຊື່ວ່າ LiveScript ແລະ ໄດ້ປ່ຽນເປັນ JavaScript ໃນອີກສາມເດືອນຕໍ່ມາ.

ໃນເດືອນພະຈິກ 1996 Netscape ໄດ້ສົ່ງພາສາ JavaScript ໄປຍັງ ECMA International ເພື່ອເປັນຈຸດເລີ່ມຕົ້ນສຳລັບກຳນົດມາດຕະຖານໃຫ້ທຸກເວັບບຣາວເຊີປະຕິບັດຕາມ

ມາດຕະຖານດັ່ງກ່າວເພື່ອໃຫ້ການພັດທະນາ JavaScript engine ເປັນໄປໃນທິດທາງດຽວກັນນັ້ນ ໃຫ້ເກີດການເປີດຕົວຢ່າງເປັນທາງການສໍາລັບຂໍ້ກຳນົດມາດຕະຖານ ECMAScript ໃນເດືອນມິຖຸນາ 1997 ໃນຊ່ວງເວລາຫຼັງຈາກນີ້ ບໍລິສັດຕ່າງໆທີ່ພັດທະນາເວັບບຣາວເຊີຕ່າງກໍຍັງພັດທະນາ JavaScript engine ບໍ່ເປັນໄປໃນທິດທາງດຽວກັນເທົ່າໃດ ນັ້ນເຮັດໃຫ້ນັກພັດທະນາເວັບຕ້ອງຂຽນໂຄດຫຼາຍເວັ້ນເພື່ອໃຫ້ເຮັດວຽກໄດ້ໃນທຸກເວັບບຣາວເຊີຈົນກະທັ້ງໃນເດືອນກໍລະກົດ 2008 ໄດ້ມີການຈັດການປະຊຸມຂຶ້ນທີ່ Oslo ຈາກອົງກອນ ແລະ ຝ່າຍຕ່າງໆທີ່ພັດທະນາ JavaScript engine ເກີດຂໍ້ຕົກລົງຂຶ້ນໃນຕົ້ນປີ 2009 ເພື່ອລວບລວມງານທີ່ກ່ຽວຂ້ອງທັງໝົດຂອງພາສາ JavaScript ແລະ ຊຸກຍູ້ພາສາໃຫ້ຢ່າງໄປຂ້າງໜ້ານັ້ນເຮັດໃຫ້ເກີດຂໍ້ກຳນົດມາດຕະຖານ ECMAScript ເວີຊັນທີ 5 (ES5) ອອກມາໃນເດືອນທັນວາ 2009 ແລະ ກ່ອນໜ້ານີ້ໃນປີ 2008 Google ໄດ້ເປີດຕົວເວັບ.

ບຣາວເຊີ Chrome ທີ່ມາພ້ອມກັບ V8 JavaScript Engine ທີ່ມີແນວຄິດໃນການພັດທະນາແບບຄອມພາຍໃນຕອນທີ່ໂປຣແກຣມຮັ້ນ (Just-In-Time Compilation: JIT) ຊຶ່ງມັນເຮັດວຽກໄດ້ໄວກວ່າເຮັດໃຫ້ຜູ້ພັດທະນາເວັບບຣາວເຊີອື່ນໆຕ້ອງປັບປຸງ JavaScript engine ຂອງພວກເຂົາໃຫ້ເຮັດວຽກໃນຮູບແບບ JIT ຫຼັງຈາກທີ່ພັດທະນາຕໍ່ເນື່ອງມາອີກຫຼາຍປີ. ໃນປີ 2015 ໄດ້ມີການເພີ່ມຄຸນສົມບັດໃໝ່ໆທີ່ຫຼາກຫຼາຍເຂົ້າມາ ຊຶ່ງຖືວ່າເປັນການປ່ຽນແປງເທື່ອສໍາຄັນ ແລະ ເຮັດໃຫ້ເກີດຂໍ້ກຳນົດມາດຕະຖານ ECMAScript 2015 ຫຼື ເວີຊັນທີ 6 (ES6) ຈົນເຖິງໃນປີ 2015 ຕອນນີ້ເບິ່ງຄືວ່າພາສາ JavaScript ຈະພັດທະນາມາຈົນເຖິງທີ່ສຸດແລ້ວເຮັດໃຫ້ລະຫວ່າງປີ

2016 - 2019 ເວີຊັນໃໝ່ຂອງ ECMAScript ທີ່ຖືກເຜີຍແຜ່ອອກມາໃນແຕ່ລະປີມີການປ່ຽນແປງ ແລະ ເພີ່ມຄຸນສົມບັດພຽງນ້ອຍໆເທົ່ານັ້ນ.

2) ຄຸນສົມບັດຂອງພາສາ JavaScript

ECMAScript 2015 (ES6) ເປັນພາສາ JavaScript ທີ່ຖືວ່າພັດທະນາມາຈົນເຖິງຈຸດສູງສຸດແລ້ວກໍວ່າໄດ້ ມັນຖືກເຜີຍແຜ່ໃນເດືອນມິຖຸນາ 2015 ຊຶ່ງໃນເວີຊັນນີ້ ໄດ້ເພີ່ມໄວຍະກອນໃໝ່ຂອງພາສາຫຼວງຫຼາຍເຊັ່ນ: ການສ້າງຄາດດ້ວຍຄໍາສັ່ງ class ການສ້າງໂມດູນ ແລະ ໃຊ້ງານມັນດ້ວຍຄໍາສັ່ງ import ແລະ export ແລະ ຄໍາສັ່ງສໍາລັບປະກາດຕົວປ່ຽນ let ແລະ ປະກາດຄ່າຄົງທີ່ const ຊຶ່ງເຮັດໃຫ້ຕົວປ່ຽນສາມາດມີຂອບເຂດໃນບ່ອນທີ່ມັນຖືກສ້າງຂຶ້ນໄດ້ ແລະ ສິ່ງອື່ນໆທີ່ຖືກເພີ່ມເຂົ້າມາເປັນຈຳນວນຫລາຍເຊັ່ນ: Map, Set, WeakMap, Promise, Reflection, Proxies, Template String ແລະ ອື່ນໆ.

ໃນເດືອນມິຖຸນາ 2016 ໄດ້ມີການເປີດໂຕເວີຊັນ 7 ຫຼື ECMAScript 2016 (ES7) ໄດ້ມີການເພີ່ມຕົວດໍາເນີນການຍົກກຳລັງ (ທີ່ກ່ອນໜ້ານີ້ເຮົາຈະໃຊ້ຜ່ານຟັງຊັນ Math.pow) ຄໍາສັ່ງ

Await Async ສໍາລັບການຂຽນໂປຣແກຣມທີ່ເຮັດວຽກບໍ່ພ້ອມກັນ ແລະ ຟັງຊັນ Includes ຂອງ ອາເລ ແລະ ໃນປະຈຸບັນພາສາ JavaScript ຖືກພັດທະນາມາຈົນເຖິງ ECMAScript 2020 (ES11) ຊຶ່ງມີການປ່ຽນແປງທີ່ເພີ່ມຂຶ້ນບໍ່ເທົ່າໃດຫຼັງຈາກ ES7.

3) JavaScript Engine ແມ່ນຫຍັງ?

JavaScript Engine ແມ່ນໂປຣແກຣມຄອມພິວເຕີທີ່ໃຊ້ສໍາລັບປະມວນຜົນໂຄດຂອງພາສາ JavaScript ຊຶ່ງ JavaScript Engine ໃນຊ່ວງເລີ່ມຕົ້ນເປັນພຽງຕົວປ່ຽນພາສາ (Interpreter) ເທົ່ານັ້ນ. ແຕ່ໃນປະຈຸບັນໄດ້ມີການພັດທະນາມາໃຫ້ຢູ່ໃນຮູບແບບຂອງຄອມພາຍເລີທີ່ມີການຄອມພາຍໃນຕອນທີ່ໂປຣແກຣມລັນ (Just-In-Time Compilation: JIT) ເພື່ອເພີ່ມປະສິດທິພາບການເຮັດວຽກງານຂອງໂປຣແກຣມໂດຍທົ່ວໄປແລ້ວ JavaScript Engine ຈະຖືກພັດທະນາໂດຍຜູ້ພັດທະນາເວັບບຣາວເຊີທີ່ປະຕິບັດຕາມຂໍ້ກຳນົດມາດຕະຖານຂອງ ECMAScript.

2.1.7 ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບ Application Programming Interface (API)

ຫຼາຍຄົນອາດຈະສົງໄສວ່າ API ຄືຫຍັງຄວາມຈິງແລ້ວຫຍໍ້ມາຈາກ Application Program Interface (API) ຊຶ່ງຄືຄໍາສັ່ງ (Code) ທີ່ອະນຸຍາດໃຫ້ Software Program ສາມາດສື່ສານລະຫວ່າງກັນໄດ້ຖ້າຈະເວົ້າໃນພາສາຄົນຂຽນ program ແລ້ວ API ເປັນຊ່ອງທາງສໍາລັບຂໍໃຊ້ບໍລິການຄໍາສັ່ງຈາກ Operation System (OS) ຫຼື Application ອື່ນໆຊຶ່ງມັນໃຊ້ງານໂດຍຕິດຕັ້ງ Function ແລະ ເອີ້ນໃຊ້ງານຕາມ Document ທີ່ຂຽນໄວ້.

➤ ສ່ວນປະກອບຂອງ APIs

APIs ສ້າງຂຶ້ນຈາກສ່ວນສໍາຄັນ 2 ຢ່າງຄື:

- ຂໍ້ກຳນົດທີ່ຈະອະທິບາຍການແລກປ່ຽນຂໍ້ມູນລະຫວ່າງ Program ຊຶ່ງເຮັດອອກມາໃນລັກສະນະ Document ເພື່ອບອກວ່າ Request/Response ຕ້ອງເປັນຢ່າງໃດ.
- Software ທີ່ຂຽນຂຶ້ນຕາມຂໍ້ກຳນົດ ແລະ ກໍານົດເສີຍແຜ່ອອກໄປໃຫ້ໃຊ້ງານ.

ໂດຍປົກກະຕິແລ້ວ Application ທີ່ມີ APIs ຈະຕ້ອງຖືກຂຽນເປັນພາສາ Programming ແລະ ພັດທະນາເພີ່ມໄດ້ງ່າຍຈຶ່ງຈຳເປັນຕ້ອງມີການກວດສອບໂຄງສ້າງ API. ສະນັ້ນ API ທີ່ດີຜູ້ທີ່ອອກແບບຕ້ອງໃຫ້ຄວາມສໍາຄັນໃນການ Test ເພື່ອກວດສອບ Logic ທີ່ສາມາດເກີດຂຶ້ນໄດ້ຈາກການໃຊ້ງານ.

➤ ການໃຊ້ງານ APIs

ປັດຈຸບັນນີ້ API ຖືກໃຊ້ງານໃນ Application ເພື່ອສື່ສານກັບ User ໂດຍບໍ່ຈຳເປັນຕ້ອງມີຄວາມຮູ້ບໍລິສັດໃຫຍ່ໆຫຼາຍບໍລິສັດມີການເປີດ API ໃຫ້ພາຍນອກເຂົ້າມາໃຊ້ງານເຊັ່ນ facebook,

Google, Twitter ຜູ້ພັດທະນາລະບົບທີ່ສົນໃຈສາມາດນຳເອົາ API ເຫຼົ່ານີ້ໄປໄປຕໍ່ຍອດຊຶ່ງທາງບໍລິສັດກໍສາມາດຂະຫຍາຍຖານລູກຄ້າອອກໄປໄດ້ອີກຮູບແບບການນຳເອົາ API ໄປໃຊ້ງານມີດັ່ງນີ້:

1) Libraries and Frameworks

API ມັກຈະເອົາໄປໃຊ້ເປັນ Software Library ຊຶ່ງຂຽນຂຶ້ນຕາມ Document ໃນຮູບແບບພາສາຂອງ Program ທີ່ຕ່າງກັນອອກໄປ ແລະ ຕາມຄວາມເໝາະສົມກັບວຽກເພື່ອເອົາໄປເຮັດເປັນ Framework ໃຫ້ກັບລະບົບໃຊ້ໃນການສື່ສານຫາກັນ.

2) Operating Systems

API ສາມາດໃຊ້ງານໃນການສື່ສານລະຫວ່າງ Application ແລະ Operating System ເຊັ່ນ: POSIX ຫຼື ມາດຕະຖານການສື່ສານຂອງ OS ເອງກໍມີ API ເປັນ Command line ເພື່ອຄວບຄຸມການເຮັດວຽກຂອງ OS.

3) Remote APIs

Remote APIs ເຮັດໃຫ້ Developer ສາມາດເຂົ້າຄວບຄຸມຊັບພະຍາກອນຜ່ານທາງ Protocol ເພື່ອໃຫ້ມີມາດຕາຖານການສື່ສານດຽວກັນເຖິງແມ່ນວ່າຈະເປັນຄົນລະ Technology ເຊັ່ນ: Database API ສາມາດອະນຸຍາດໃຫ້ Developer ເຂົ້າມາດຶງຂໍ້ມູນໃນ Database ຫຼາກຫຼາຍຊະນິດໄດ້ຜ່ານ Function ດຽວກັນ ສະນັ້ນ Remote API ຈຶ່ງຖືກໃຊ້ເລື້ອຍໆໃນ Maintenance ດ້ວຍການເຮັດວຽກທີ່ຜັງ Client ໃຫ້ໄປດຶງຂໍ້ມູນຈາກ Server ກັບລົງມາເຮັດວຽກ.

4) Web APIs

ໃນປະຈຸບັນແມ່ນນິຍົມໃຊ້ກັນຫຼາຍ ເພາະຢູ່ໃນກຸ່ມຂອງ HTTP ແລະ ຂະຫຍາຍອອກໄປສູ່ຮູບແບບ XML ແລະ JSON ຊຶ່ງໂດຍລວມແລ້ວກໍຄືຢູ່ເທິງ Web Service ເຊັ່ນ:

- SOAP (Simple Object Access Protocol) ໃຊ້ XML Format ສົ່ງຂໍ້ມູນ.
- REST (Representational State Transfer) ສາມາດໃຊ້ XML ຫຼື JSON Format ສົ່ງຂໍ້ມູນ.
- ຕົວຢ່າງ API ທີ່ນິຍົມໃນປະຈຸບັນ
- Google Maps API: ເປີດໃຫ້ໃຊ້ງານເພື່ອນຳເອົາແຜນທີ່ຂອງ Google ມາລົງໃນ webpage ໂດຍອາໄສ JavaScript ຫຼື Flash.
- YouTube APIs: Google ຍອມໃຫ້ developer ສາມາດນຳເອົາ Clip Video ເທິງ YouTube ໄປລົງໃນ Website ຫຼື Application ໄດ້.
- Flickr API: ເພື່ອໃຫ້ Developer ສາມາດເຂົ້າເຖິງ ຄັງຮູບພາບໃນ Community
- Twitter APIs: ມີ REST API ໃຫ້ຄົ້ນຫາແລ້ວກວດສອບຂໍ້ມູນ Trends ໄດ້.

- Amazon Product Advertising API: ເປີດ API ໃຫ້ໃຊ້ຄົ້ນຫາສິນຄ້າ ແລະ ການໂຄສະນາຜ່ານທາງ Wrebsite.

2.1.8 ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບ Nodejs

Node.js ເປັນ Open-Source ແລະ Cross-Platform JavaScript Runtime Environment ທີ່ກຳລັງໄດ້ຮັບຄວາມນິຍົມສູງ ໂດຍທົ່ວໄປເຮົາຈະໃຊ້ JavaScript ໃນຝັ່ງ client ແຕ່ Node.js ເຮັດໃຫ້ເຮົາໃຊ້ JavaScript ໃນຝັ່ງ Server ໄດ້ດ້ວຍຊື່ງ Node.js ສາມາດ Run ໄດ້ເທິງ Platform ທີ່ຫຼາກຫຼາຍທັງ Windows, Linux, Unix, Mac OS X ແລະ ອື່ນໆ.



ຮູບທີ 5 : ຮູບ Nodejs

ຈຸດເດັ່ນທີ່ສຸດຂອງ Node.js ຄືມັນເຮັດວຽກແບບ Asynchronous ຜູ້ອ່ານອາດຈະສົງໃສວ່າແລ້ວມັນດີແນວໃດ? ລອງມາເບິ່ງຕົວຢ່າງການຈັດການໄຟລເມື່ອມີການຮ້ອງຂໍຈາກ Client ມາທີ່ Server ຂອງ PHP ກັບ Node.js ທຽບກັນເບິ່ງເພື່ອໃຫ້ເຂົ້າໃຈຫຼາຍຂຶ້ນ.

- ກໍລະນີ PHP ເມື່ອມີການຮ້ອງຂໍເຂົ້າມາມັນຈະເຮັດດັ່ງນີ້:
 1. ສົ່ງ Task ໄປທີ່ລະບົບໄຟຂອງຄອມພິວເຕີ.
 2. ລໍຖ້າຈົນກະທັ້ງລະບົບໄຟລເປີດ ແລະ ອ່ານໄຟລສຳເລັດ.
 3. ສົ່ງເນື້ອຫາຂອງໄຟລກັບມາໃຫ້ Client.
 4. ພ້ອມສຳລັບຮັບຄຳຮ້ອງຂໍຖັດໄປ.
- ກໍລະນີ Node.js ເມື່ອມີການຮ້ອງຂໍເຂົ້າມາມັນຈະເຮັດດັ່ງນີ້:
 1. ສົ່ງ Task ໄປທີ່ລະບົບໄຟຂອງຄອມພິວເຕີ.

2. ພ້ອມສຳລັບຮັບຄຳຮ້ອງຂໍຖັດໄປ.

3. ພໍລະບົບໄຟລເປີດ ແລະ ອ່ານໄຟລສຳເລັດ Server ຈະສົ່ງເນື້ອຫາຂອງໄຟລກັບມາໃຫ້ Client.

ຈາກຕົວຢ່າງຂ້າງເທິງຈະເຫັນວ່າ Node.js ຈະຕັດຂັ້ນຕອນການລໍຖ້າຖິ້ມແລ້ວໄປເຮັດຄຳຮ້ອງຖັດໄປເລີຍທີ່ ເປັນແບບນີ້ເພາະ Node.js ຈະ Run ແບບ Single-Threaded ແລະ ໃນ Library ມາດຕະຖານກໍຈະມີເຊັດຂອງ Asynchronous I/O Primitives ທີ່ຊ່ວຍປ້ອງກັນໂຄດ JavaScript ຈາກການ Blocking ເຮັດໃຫ້ລະບົບຄ່ອງແຄ້ວ ແລະ ມີປະສິດທິພາບຫຼາຍຂຶ້ນ.

Synchronous vs Asynchronous	
<p>Synchronous ຄືການ Run ໂຄດຕາມລຳດັບທີ່ເຮົາຂຽນໄວ້ ເຊັ່ນ</p> <pre>alert(1); alert(2); alert(3);</pre> <p>ຜົນລັບທີ່ໄດ້ຄືໜ້າຈໍຈະສະແດງຜົນ 1 ຈາກນັ້ນຈຶ່ງສະແດງຜົນ 2 ແລ້ວສຸດທ້າຍຈຶ່ງສະແດງຜົນ 3 ຕາມລຳດັບ</p>	<p>Asynchronous ຄືການຮັບໂຄດທີ່ບໍ່ຈຳເປັນຕ້ອງເປັນໄປຕາມລຳດັບທີ່ເຮົາຂຽນໄວ້ ເຊັ່ນ:</p> <pre>alert(1); setTimeout(() => alert(2), 0); alert(3);</pre> <p>ໃນຕົວຢ່າງນີ້ໂຄດ alert(2) ໃຊ້ເວລາດຳເນີນການດົນກວ່າ ເຮັດໃຫ້ຜົນລັບທີ່ໄດ້ຄືໜ້າຈໍຈະສະແດງຜົນ 1 ຈາກນັ້ນຈຶ່ງສະແດງຜົນ 3 ແລ້ວສຸດທ້າຍຈຶ່ງສະແດງຜົນ 2</p>
Blocking vs Non-blocking	
<p>Blocking ໝາຍເຖິງການທີ່ເຮົາບໍ່ສາມາດດຳເນີນການຕໍ່ໄປໄດ້ຈົນກວ່າຕົວດຳເນີນການທີ່ກຳລັງລັນຢູ່ຈະສຳເລັດເສຍກ່ອນ ເຊັ່ນ</p> <pre>alert(1); var value = localStorage.getItem('foo'); alert(2);</pre> <p>ຄຳສັ່ງ localStorage ຈະເປັນຕົວ blocking ເຮັດໃຫ້ບໍ່ສາມາດເຮັດຄຳສັ່ງ alert(2) ໄດ້ຈົນກວ່າມັນຈະດຳເນີນການສຳເລັດ ດັ່ງນັ້ນຜົນລັບທີ່ໄດ້ຄືໜ້າຈໍຈະສະແດງຜົນ 1 ຈາກນັ້ນຈຶ່ງລໍຖ້າຈົນກວ່າຄຳສັ່ງ localStorage ຈະສຳເລັດຈຶ່ງສະແດງຜົນ 2</p>	<p>Non-blocking ໝາຍເຖິງການໂຕທີ່ດຳເນີນການສາມາດເຮັດຄຳສັ່ງຖັດໄປໄດ້ເລີຍໂດຍບໍ່ຕ້ອງລໍຖ້າໃຫ້ຄຳສັ່ງເດີມເຮັດສຳເລັດກ່ອນເຊັ່ນ:</p> <pre>alert(1); fetch('example.com').then(() => alert(2)); alert(3);</pre> <p>ໃນຕົວຢ່າງນີ້ ຄຳສັ່ງ fetch ເປັນ non-blocking operation ດັ່ງນັ້ນຜົນລັບທີ່ໄດ້ຄືໜ້າຈໍຈະສະແດງຜົນ 1 ຈາກນັ້ນຈຶ່ງສະແດງຜົນ 3 ແລ້ວພໍຄຳສັ່ງ fetch ສຳເລັດກໍສະແດງຜົນ 2</p>

ຕາຕະລາງທີ 4 : ຕາຕະລາງປຽບທຽບລະຫວ່າງ Synchronous ກັບ Asynchronous

➤ ປະຫວັດ Node.js

ແຕ່ເດີມພາສາ JavaScript ຖືກພັດທະນາມາເພື່ອໃຊ້ສຳລັບ Browser ທີ່ຊື່ Netscape Navigator ໃນປີ 1995 ໃນຕອນນັ້ນ Netscape ຕັ້ງໃຈຈະຂາຍ Web Server ທີ່ມີ Environment ຊື່ Netscape LiveWire ຊຶ່ງສາມາດສ້າງ Dynamic Page ໂດຍໃຊ້ JavaScript ທາງຝັ່ງ server ອີກດ້ວຍ ແຕ່ໜ້າເສຍດາຍທີ່ Netscape LiveWire ບໍ່ປະສົບຄວາມສຳເລັດ ແລະ ການໃຊ້ JavaScript ທາງຝັ່ງ Server ກໍບໍ່ໄດ້ຮັບຄວາມນິຍົມເລີຍຈົນກະທັ້ງ Node.js ຖືກກຳເນີດຂຶ້ນມາ.

ສິ່ງທີ່ເຮັດໃຫ້ Node.js ເປັນທີ່ນິຍົມຂຶ້ນມາຄືການທີ່ມັນມາໃນຊ່ວງເວລາທີ່ເໝາະສົມ ເມື່ອທຽບກັບ JavaScript ທີ່ເກີດມາຕັ້ງແຕ່ປີ 1995 ແລ້ວ Node.js ຫາກໍເກີດມາເມື່ອປີ 2009 ເທົ່ານັ້ນ ຕ້ອງຂອບໃຈ "Web 2.0" applications ເຊັ່ນ Flickr, Gmail ແລະ ອື່ນໆທີ່ສະແດງໃຫ້ໂລກຮູ້ວ່າເວັບສະໄໝໃໝ່ຄວນໜ້າຕາເປັນຢ່າງໃດ.

ບໍ່ດົນກ່ອນທີ່ Node.js ຈະເກີດ ນັກພັດທະນາເບື້ອງໜຶ່ງ Browser ຊື່ດັງທັງຫຼາຍແຂ່ງຂັນກັນເຮັດວຽກຢ່າງໜັກເພື່ອຈະໃຊ້ JavaScript ໃຫ້ໄດ້ດີທີ່ສຸດ ແລະ ຫາທາງເຮັດໃຫ້ JavaScript ສາມາດຮັບໄດ້ໄວຫຼາຍຂຶ້ນເພື່ອໃຫ້ຜູ້ໃຊ້ງານໄດ້ຮັບປະສິດທິພາບທີ່ດີທີ່ສຸດຊຶ່ງຜົນຈາກການແຂ່ງຂັນນີ້ເຮັດໃຫ້ເກີດການພັດທະນາ Chrome V8 (Open-Source JavaScript Engine ຂອງ The Chromium Project) ຂຶ້ນມາ ແລະ Node.js ກໍໃຊ້ Engine.

ແຕ່ການທີ່ Node.js ເປັນທີ່ນິຍົມຂຶ້ນມາບໍ່ແມ່ນແຕ່ວ່າມັນມາຖືກທີ່ຖືກເວລາເທົ່ານັ້ນ ແຕ່ມັນໄດ້ສະແດງໃຫ້ເຫັນແລ້ວວ່າການອອກແບບ ແລະ ແນວຄິດຂອງມັນຊ່ວຍນັກພັດທະນາທັງຫຼາຍໃຫ້ສາມາດໃຊ້ JavaScript ທາງຝັ່ງ Server ໄດ້ງ່າຍຂຶ້ນຫຼາຍອີກດ້ວຍ.

2.1.9 ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບ Reactjs (JavaScript Library)

React ເປັນເທັກໂນໂລຢີໜຶ່ງທີ່ມາແຮງຫຼາຍໂດຍສ້າງມາຈາກພື້ນຖານແນວຄວາມຄິດແບບ MVC (Model View Controller) ຊຶ່ງໝາຍຄວາມວ່າ React ມີໜ້າທີ່ຈັດການກັບ Model ຫຼື View ແຕ່ສ່ວນໃຫຍ່ຈະເປັນ View ກ່ອນໜ້ານັ້ນເວລາຈະຂຽນໜ້າເວັບເຮົາກໍຈະເຮັດຜ່ານ HTML ມີການໃຊ້ CSS ໃນການປັບປຸງໜ້າຕາຂອງ UI ແຕ່ໃນ React ຖ້າຈະສ້າງໜ້າເວັບຂຶ້ນມາໄດ້ນັ້ນ ເຮົາຈະໃຊ້ເປັນ Component ປຸງໄດ້ວ່າ Component ເປັນ Block ສ່ວນຍ່ອຍຂອງເວັບເຮົາທີ່ຈະສ້າງອອກມາ.

➤ ສະຫຼຸບງ່າຍໆດັ່ງນີ້:

- React ເປັນ Javascript Library ສ້າງ ແລະ ພັດທະນາຂຶ້ນໂດຍ Facebook ຂຽນໄດ້ແຕ່ UI ເທົ່ານັ້ນ ແລະ ເປີດໃຫ້ໃຊ້ຟຣີ.
- React ໃຊ້ໂຄດ HTML , CSS ແລະ Java Script.

- React ມີ 3 ຄອນເຊບທີ່ເຮົາຕ້ອງຮຽນຮູ້ຄື Component State ແລະ Props.

➤ ຂໍ້ດີ ແລະ ຂໍ້ເສຍຂອງ React

ກ. ຂໍ້ດີ

- Component ເຂົ້າໃຈງ່າຍເຮົາສາມາດຮຽນຮູ້ໄດ້ດ້ວຍຕົວເອງ.

- Tool ຫຼາຍພຽງ React ຢ່າງດຽວກໍ່ສາມາດຂຽນເວັບໄດ້ທັງເວັບແລ້ວ ໂດຍບໍ່ຕ້ອງ
ຫາ Tool ເພີ່ມເຕີມ ແລະ ຍັງມີ Tool ພັດທະນາອອກມາຢູ່ເລື້ອຍໆສາມາດເຮັດ
App ໄດ້ React ມີເຄື່ອງມືໜຶ່ງທີ່ຊື່ວ່າ React Native ເປັນການຂຽນ JavaScript
ແລ້ວແປງເປັນ App ແບບ Native ໄດ້ທັງເທິງ Android ແລະ Ios.

ຂ. ຂໍ້ເສຍ

- ຕ້ອງມີພື້ນຖານໃນ Java Script ໃນລະດັບໜຶ່ງ ຄົນທີ່ສົນໃຈຮຽນກໍ່ຈະລຳບາກໜ້ອຍ
ໜຶ່ງຖ້າບໍ່ໄດ້ມີພື້ນຖານ Java Script ອາດຈະຕ້ອງໃຊ້ເວລາໜ້ອຍໜຶ່ງ.

- Documentation ອ່ານຍາກ React ມີ Documentation ທີ່ຍັງບໍ່ຄ່ອຍດີໃນອະນາຄົດ
ອາດຈະມີການພັດທະນາອີກ.

1) ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບ GraphQL (GraphQL API)

GraphQL ຄືພາສາສຳລັບການເຂົ້າເຖິງຂໍ້ມູນ (Query Language) ເພື່ອການໃຊ້ງານ API
ຂອງລະບົບ ແລະ ຈະປະມວນຜົນຄຳສັ່ງທີ່ຝັງ Server ຫຼື ທີ່ເອີ້ນວ່າ Server-Side Runtime ໂດຍ
ໃຊ້ໂຄງສ້າງຂໍ້ມູນທີ່ເຮົາກຳນົດໄວ້. ແຕ່ GraphQL ບໍ່ໄດ້ພັດທະນາຂຶ້ນມາເພື່ອແທນທີ່ພາສາສຳລັບ
ການເຂົ້າເຖິງຂໍ້ມູນເຊັ່ນ SQL ຫຼື ເຮັດໜ້າທີ່ເປັນລະບົບຈັດເກັບຖານຂໍ້ມູນ (Storage Engine) ແຕ່
ຢ່າງໃດ.

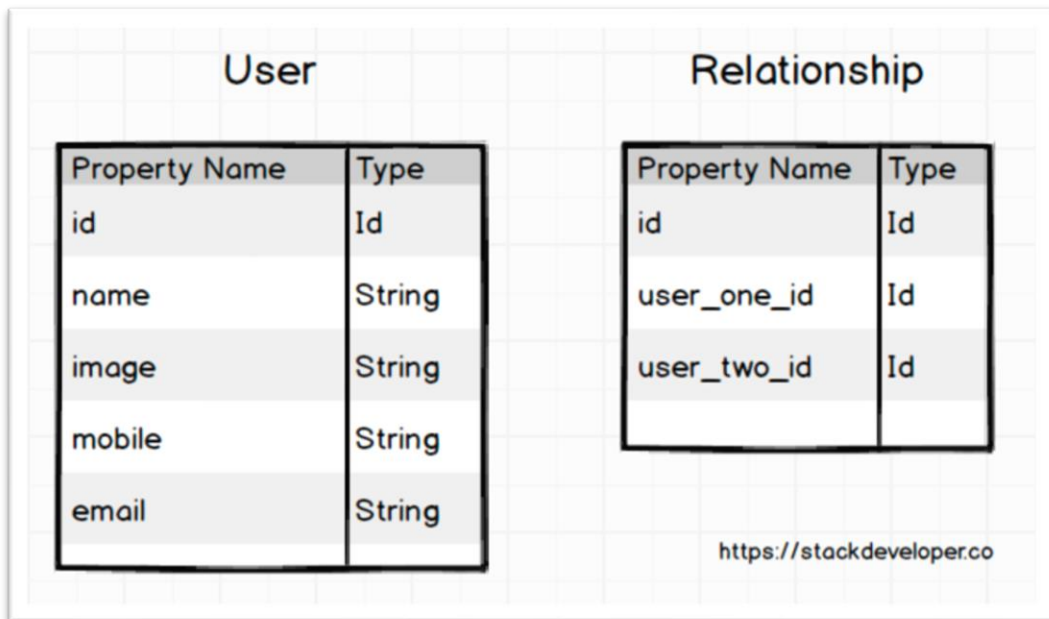
ໃນປະຈຸບັນວິທີທີ່ເຂົ້າເຖິງຂໍ້ມູນຜ່ານເວັບເຊີວິດທີ່ໄດ້ຮັບຄວາມນິຍົມກໍ່ຈະເປັນ REST API
ຜ່ານທາງ HTTP Methods ຕົວຢ່າງເຊັ່ນ: ຕ້ອງການດຶງຂໍ້ມູນຜູ້ໃຊ້ງານທັງໝົດກໍ່ສາມາດຮຽກຜ່ານ
API ໄດ້ດັ່ງນີ້:

GET /users

ຫຼື ຫາກຕ້ອງການດຶງເພື່ອນທັງໝົດຂອງຜູ້ໃຊ້ງານ ID 25

GET /users/25/friends

ແຕ່ຫາກຕ້ອງການດຶງເບີໂທລະສັບຂອງເພື່ອນທີ່ເປັນເພື່ອນກັບຜູ້ໃຊ້ງານ ID 25 ຈະຕ້ອງດຶງ
ແລະ ຂຽນໂປຣແກຣມແບບໃດ? ໂດຍມີການຈັດເກັບຂໍ້ມູນດັ່ງນີ້:



ຮູບທີ 4 : ຮູບຕົວຢ່າງ GraphQL

ຈາກຕົວຢ່າງນີ້ຈະເຫັນໄດ້ວ່າເຮົາຈຳເປັນຕ້ອງມີ API 2 end point ນຳກັນຄື GET /users/:id/friends ເພື່ອດຶງຂໍ້ມູນເພື່ອນຂອງຜູ້ໃຊ້ ID 25 ຈາກນັ້ນ ກໍ່ວິນລູບດຶງຂໍ້ມູນຜູ້ໃຊ້ໂດຍໃຊ້ GET /users/:id ເພື່ອດຶງຂໍ້ມູນເບີໂທສັບອີກຄັ້ງ ແລະ ວິທີນີ້ກໍ່ຈະໄດ້ຂໍ້ມູນອື່ນໆ ທີ່ບໍ່ຈຳເປັນຂອງຜູ້ໃຊ້ມາອີກບໍ່ວ່າຈະເປັນຊື່, ຮູບພາບ ແລະ ອີເມວໃນຄວາມເປັນຈິງແລ້ວອົງກອນໃຫຍ່ໆທີ່ມີຫຼາກຫຼາຍພະແນກຕ່າງກໍ່ຮ້ອງຂໍຂໍ້ມູນທີ່ບໍ່ຄືກັນເຊັ່ນ : ບາງພະແນກຕ້ອງການສະເພາະຂໍ້ມູນພະນັກງານ, ບາງພະແນກຕ້ອງການຂໍ້ມູນພະນັກງານພ້ອມຂໍ້ມູນເງິນເດືອນ ຫຼື ບາງພະແນກຕ້ອງການຂໍ້ມູນພະນັກງານກັບຈຳນວນວັນທີຂາດ ຫຼື ລາເທົ່ານັ້ນເປັນຕົ້ນ. ການໃຊ້ງານ REST API ຈຶ່ງສ້າງຄວາມລຳບາກຕໍ່ການພັດທະນາ ແລະ ດູແລຮັກສາເຊີວິດ.

ດ້ວຍເຫດນີ້ GraphQL ຈຶ່ງໄດ້ເຂົ້າມາເພື່ອແກ້ບັນຫາໃນຈຸດນີ້ໂດຍສະເພາະເພື່ອໃຫ້ເຮົາສາມາດດຶງຂໍ້ມູນໄດ້ກົງກັບຄວາມຕ້ອງການຫຼຸດຄວາມຊັບຊ້ອນໃນການຂຽນໂຄດເພື່ອດຶງຂໍ້ມູນ, ສາມາດຈັດການ ແລະ ດູແລຮັກສາໂຄ້ດໃນຝັ່ງ Server-Side ໄດ້ງ່າຍຫຼາຍຢິ່ງຂຶ້ນ ແລະ ທີ່ສຳຄັນ GraphQL ບໍ່ໄດ້ຜູກຕິດກັບ Database ແລະ ທີ່ຈັດເກັບຂໍ້ມູນໃດໆທັງນັ້ນ.

2) ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບ Expressjs (Nodejs Library)

Express.js ເປັນ Web Application Framework ຊື່ດັງທີ່ໄດ້ຮັບຄວາມນິຍົມຫຼາຍສຳລັບເຮັດວຽກເທິງ Platform ຂອງ Node.js ຊຶ່ງເປັນ Server ໂຕໜຶ່ງໂດຍທັງ Express.js ແລະ

Node.js ຕ່າງກໍໃຊ້ພາສາ JavaScript ໃນການພັດທະນາຖ້າເປັນ Web Application Framework ໃນສະໄໝກ່ອນຄົນທີ່ພັດທະນາຈະຕ້ອງມີຄວາມຮູ້ຫຼາຍກວ່າ 1 ພາສາ, ພາສາທີ່ເຮັດວຽກທາງຝັ່ງ Server ຢ່າງ PHP ຫຼື ASP ແລະ ພາສາທີ່ເຮັດວຽກທາງຝັ່ງ Client ຢ່າງ JavaScript ເພື່ອຫຼຸດຄວາມຫຍຸ້ງຍາກທັງໝົດເຖິງເວລາໃນການຕ້ອງຮຽນຮູ້ຫຼາຍໆພາສາເຮັດໃຫ້ເກີດ Node.js ກັບ Express.js ພຽງແຕ່ມີຄວາມຮູ້ JavaScript ກໍສາມາດຂຽນໄດ້ທັງ Server ແລະ Client ນອກຈາກນີ້ ຖ້າໃຜເຄີຍຂຽນ JavaScript ຈະຮູ້ວ່າມັນມີການຕອບສະໜອງທີ່ວ່ອງໄວແນ່ນອນວ່າ Express.js ກໍຍົກເອົາມາເປັນຂໍ້ເດັ່ນໃນເລື່ອງຄວາມໄວ ໃນເລື່ອງການຮຽນຮູ້ການຂຽນ Express.js ຈະໃຊ້ຮູບແບບທີ່ງ່າຍໃນການຮຽນຮູ້ຫຼາຍທີ່ສຸດສໍາລັບການພັດທະນາ Express.js ໃນເວັບໄຊຈະເວົ້າເຖິງການໃຊ້ Routing (ການກຳນົດເສັ້ນທາງຂອງລະບົບ) ແລະ Middleware (ການຮັບສິ່ງຂໍ້ມູນຂອງລະບົບ) ສາມາດຂຽນໄດ້ໃນຮູບແບບ MVC ສ່ວນການເຊື່ອມຕໍ່ກັບຖານຂໍ້ມູນສາມາດໃຊ້ MongoDB ຫຼື ຈະໃຊ້ MySQL ກໍໄດ້ສໍາລັບນາມສະກຸນຂອງໄຟລຕີ .js ຂະນະນີ້ໄດ້ພັດທະນາມາເຖິງເວີຊັນທີ່ 4.x ແລ້ວ.

3) ຄວາມຮູ້ກ່ຽວກັບໂປຣແກຣມ Microsoft Office 2016

Microsoft Word ແມ່ນເປັນໂປຣແກຣມປະມວນຜົນເຊິ່ງອອກແບບມາເພື່ອຊ່ວຍໃຫ້ສ້າງເອກະສານທີ່ມີຄຸນນະພາບໃນລະດັບມືອາຊີບ ແລະ ເຄື່ອງມືການຈັດຮູບແບບເອກະສານທີ່ດີທີ່ສຸດຂອງ Word ຈະເຮັດໃຫ້ສາມາດຈັດລະບຽບ ແລະ ຂຽນເອກະສານຂອງທ່ານໄດ້ຢ່າງມີປະສິດທິພາບຫຼາຍຂຶ້ນ Word ຍັງມີເຄື່ອງມືການແກ້ໄຂ ແລະ ກວດສອບຄໍາຜິດທີ່ມີປະສິດທິພາບສາມາດເຮັດວຽກຮ່ວມກັບຜູ້ອື່ນໄດ້ຢ່າງງ່າຍດາຍ.

ກ. ປະໂຫຍດຂອງໂປຣແກຣມ Microsoft Word Office 2016

- ມີລະບົບປະຕິບັດຕ່າງໆທີ່ຊ່ວຍໃນການເຮັດວຽກໃຫ້ສະດວກສະບາຍຂຶ້ນເຊັ່ນ: ການກວດຄໍາສະກົດ, ການກວດສອບໄວຍະກອນ, ການໃສ່ຂໍ້ຄວາມອັດຕະໂນມັດເປັນຕົ້ນ.
- ສາມາດໃຊ້ສ້າງຈົດໝາຍໄດ້ຢ່າງວ່ອງໄວໂດຍສາມາດກຳນົດໃຫ້ຜູ້ວິເສດ (Wizard) ໃນ Word ສ້າງແບບຟອມຂອງຈົດໝາຍໄດ້ຫຼາຍຮູບແບບຕາມທີ່ຕ້ອງການ.
- ສາມາດໃຊ້ Word ສ້າງຕາຕະລາງທີ່ສະຫຼັບຊັບຊ້ອນໄດ້.
- ບັບປຸງເອກະສານໄດ້ງ່າຍ ແລະ ວ່ອງໄວ ສາມາດຕົກແຕ່ງເອກະສານ ຫຼື ເພື່ອຄວາມສະດວກ.
- ຈະໃຫ້ Word ສາມາດແຊກຮູບພາບ, ກຣາບພິກ ຫຼື ຜັງອົງກອນລົງໃນເອກະສານໄດ້.

- Word ບັບແຕ່ງໃຫ້ກໍໄດ້ ໂດຍສາມາດເປັນຜູ້ກຳນົດຮູບແບບຂອງເອກະສານເອງ.
- ຄວາມສາມາດໃນການເຊື່ອມຕໍ່ໂປຣແກຣມອື່ນໆ ໃນຊຸດໂປຣແກຣມ Microsoft Office ສາມາດໂອນຍ້າຍຂໍ້ມູນຕ່າງໆລະຫວ່າງໂປຣແກຣມໄດ້.
- ສ້າງເອກະສານໃຫ້ໃຊ້ວຽກໃນອິນເຕີເນັດໄດ້ຢ່າງງ່າຍດາຍ.

ຂ. ຄວາມສາມາດຂອງ Microsoft Word

- ສາມາດພິມ ແລະ ແກ້ໄຂເອກະສານ.
- ສາມາດລຶບ, ຍ້າຍ ແລະ ຮ່າງຂໍ້ຄວາມ.
- ສາມາດພິມຕົວອັກສອນປະເພດຕ່າງໆ.
- ສາມາດຂະຫຍາຍຕົວອັກສອນ.
- ສາມາດຈັດຕົວອັກສອນໃຫ້ເປັນຕົວໜາ,ຕົວຫງ່ຽງ ແລະ ຂີດກ້ອງໄດ້.
- ສາມາດໃສ່ເຄື່ອງໝາຍ ແລະ ຕົວເລກລຳດັບຕາມຫົວຂໍ້.
- ສາມາດແບ່ງເສົາໄດ້.
- ສາມາດກວດການສະກົດ ແລະ ແກ້ໄຂໃຫ້ຖືກຕ້ອງ.
- ສາມາດຄົ້ນຫາ ແລະ ປຸງແປງຂໍ້ຄວາມທີ່ພິມຜິດ.
- ສາມາດຈັດຂໍ້ຄວາມເອກະສານໃຫ້ພິມໄປທາງຊ້າຍ, ທາງຂວາ ແລະ ລະຫວ່າງກາງ.
- ສາມາດໃສ່ຮູບພາບໃນເອກະສານ.
- ສາມາດຕົກແຕ່ງໂຕອັກສອນ,ພິມຕາຕະລາງ.

4) ຄວາມຮູ້ກ່ຽວກັບໂປຣແກຣມ Microsoft Visio 2016

ໂປຣແກຣມ Microsoft Visio ເປັນເຄື່ອງມືເສີມການເຮັດວຽກຂອງ Microsoft Office ໃນການສ້າງແຜນວາດ (Diagram) ປະເພດຕ່າງໆ,ເຊິ່ງເຮັດໄດ້ງ່າຍດາຍ,ສະດວກ ແລະ ວ່ອງໄວ ເປັນທີ່ນິຍົມໃຊ້,ມີປະໂຫຍດຫຼາຍສຳລັບການຈັດເອກະສານ,ລວມໄປເຖິງແຜນວາດການອອກແບບຂັ້ນຕອນເຮັດການວິເຄາະ ແລະ ອອກແບບລະບົບຕ່າງໆ.

➤ ສຳລັບໂປຣແກຣມ Microsoft Visio ມີໃຫ້ເລືອກຢູ່ 2 ປະເພດຄື:

ກ. Microsoft Visio Standard ເໝາະກັບວຽກດ້ານທຸລະກິດເຊັ່ນ: ຜູ້ບໍລິຫານໂຄງການ, ນັກການຕະຫຼາດ, ພະນັກງານຝ່າຍຊັບພະຍາກອນມະນຸດ ແລະ ທີມງານມີໜ້າທີ່ເບິ່ງແຍງການດຳເນີນງານເພື່ອຊ່ວຍໃນການເບິ່ງແຜນວາດ ແລະ ຂ່າວສານ.

ຂ. Microsoft Visio Professional ເໝາະສຳລັບມືອາຊີບທາງດ້ານເຕັກນິກ, ພະນັກງານໄອທີ, ນັກພັດທະນາ ແລະ ວິສະວະກອນທີ່ຊ່ວຍໃນການອອກແບບຂໍ້ມູນລະບົບເພື່ອໃຊ້ໃນການເຮັດຕົ້ນແບບ Microsoft Visio Professional ເປັນໂປຣ

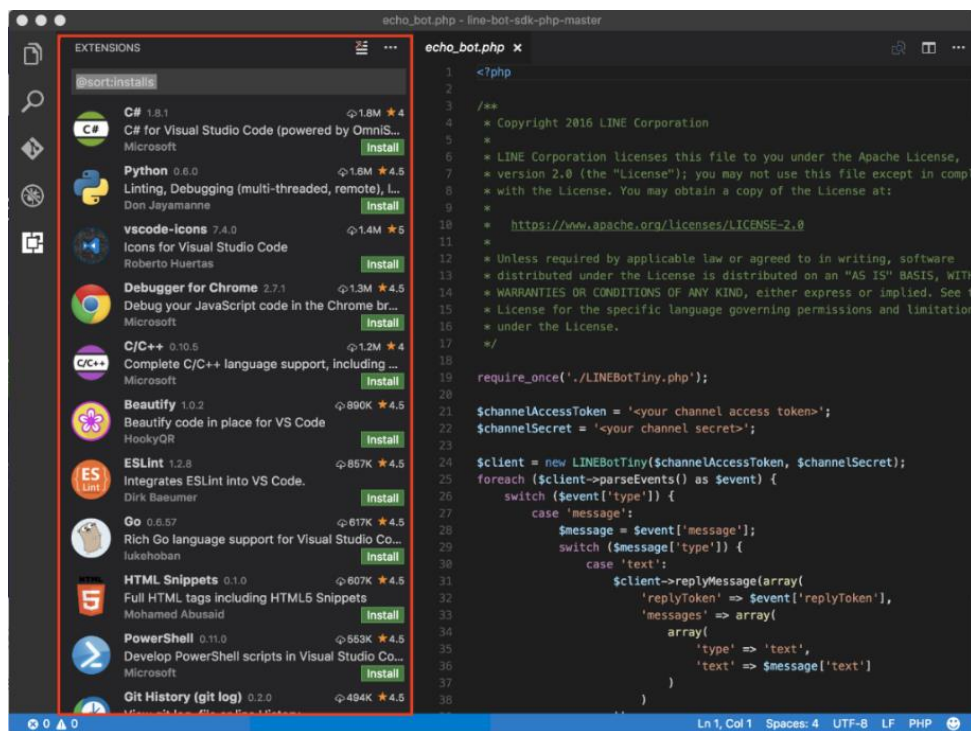
ແກຣມທີ່ຖືກສ້າງຂຶ້ນມາເພື່ອຊ່ວຍໃນການສ້າງ Flow chart ຫຼື Diagram ຂອງວຽກ
ໃນສາຂາຕ່າງໆໃຫ້ເຮັດວຽກໄດ້ງ່າຍຂຶ້ນ. ລັກສະນະຢ່າງໜຶ່ງໃນການສ້າງ Flow
Chart ໃນ Microsoft Visio Professional ກໍຄືມີຮູບແບບ Diagram ພື້ນຖານ
ຕ່າງໆຈັດກຸ່ມໄວ້ໃຫ້ ເຊິ່ງງ່າຍໃນການອອກແບບ ແລະ ໃຊ້ວຽກ.

ຈຸດດີ Microsoft Visio ຄື: ຊອກຫາຂໍ້ຜິດພາດໄດ້ຢ່າງສະດວກ ແລະ ເຂົ້າໃຈງ່າຍໃນ
ການສະແດງທິດທາງການໄຫຼຂໍ້ມູນ.

5) ຄວາມຮູ້ກ່ຽວກັບໂປຣແກຣມ Visual Studio Code

Visual Studio Code ຫຼື ທີ່ຫຼາຍຄົນນິຍົມເອີ້ນຫຍໍ້ວ່າ “Vs Code” ບອກກ່ອນວ່າ Editor
ໂຕນີ້ມັນອອກມາຕັ້ງແຕ່ 29 ເມສາ ປີ 2015 ແລ້ວພັດທະນາຂຶ້ນໂດຍບໍລິສັດຍັກໃຫຍ່ໄມໂຄຣຊອບ
(Microsoft) ເປັນທັງໂຕແກ້ໄຂ ແລະ ປັບແຕ່ງໂຄດ (Code Optimized Editor) ທີ່ຕັດຄວາມ
ສາມາດມາຈາກ Visual Studio ລຸ້ນປົກກະຕິ (ພວກ GUI designer) ອອກໄປເຫຼືອແຕ່ໂຕ Editor
ຢ່າງດຽວສາມາດເຮັດວຽກໄດ້ຂ້າມແພລດຟອມທັງໝົດວິນ Windows, Mac ແລະ Linux ຊັບ
ພອດພາສາຫຼາຍຮ້ອຍພາສາອີກ ຊຶ່ງທາງໄມໂຄຊອບເອງນັ້ນໄດ້ເປີດໃຫ້ໃຊ້ຟີອີກດ້ວຍ.

ຄວາມສາມາດຂອງ “Vs Code” ນັ້ນຈະມີຄວາມສາມາດໃນການເປີດໄດ້ຄືກັບ Editor ໂຕ
ອື່ນໆເຊັ່ນ: Sublime, Atom, Notepad++ ທັງໝົດເຖິງຄວາມສາມາດໃນການຕິດຕັ້ງເຄື່ອງມືເສີມ
(Extension) ໂດຍຮັບຮອງໄດ້ວ່າມີຊັບພອດຢ່າງແນ່ນອນ ເພາະວ່າມັນຖືກພັດທະນາມາໃຫ້ຕອບ
ໂຈດນັກພັດທະນາຫຼາຍທີ່ສຸດແມ່ນການດຶງຊ້າຍໜ້າຕາໃຫ້ເປັນຮູບແບບທີ່ເຂົ້າໃຈ ແລະ ໃຊ້ງານໄດ້
ງ່າຍບໍ່ຕ້ອງສຶກສາຫຍັງເພີ່ມເຕີມກໍໃຊ້ງານໄດ້ເລີຍ, ສິ່ງທີ່ເຮັດໃຫ້ມັນໂດດເດັ່ນກວ່າໂຕອື່ນໆຄືການທີ່
ອອກແບບໃຫ້ການຄົ້ນຫາສິ່ງຕ່າງໆເຮັດອອກມາໃຫ້ໃຊ້ງານໄດ້ງ່າຍ ແລະ ເບິ່ງງ່າຍກວ່າໂຕອື່ນ
ການທີ່ສ້າງໃຫ້ສາມາດເຊື່ອມຕໍ່ກັບ Git ໄດ້ຢ່າງວ່ອງໄວ ແລະ ງ່າຍດາຍມີຟັງຊັນໃນການ
Commit, Push & Pull ຢູ່ໃນໂຕ ຫຼື ຈະເບິ່ງ Change ຂອງຟາຍທີ່ເກີດຂຶ້ນກໍໄດ້ແບບງ່າຍດາຍບໍ່
ຕ້ອງສຶກສາຫຍັງເພີ່ມແຕ່ກໍໃຊ້ງານໄດ້ເລີຍ.



ຮູບທີ 5 : ຮູບໂປຣແກຣມ VS Code

2.2 ບົດຄົ້ນຄວ້າທີ່ກ່ຽວຂ້ອງ

2.2.1 Southern Bus Station Booking online System

ປຶ້ມໂຄງການຈົບຊັ້ນ ລະດັບປະລິນຍາຕີ ສາຂາວິທະຍາສາດຄອມພິວເຕີ ຫົວຂໍ້ເວບໄຊ Southern Bus Station Booking online System ສົກສຶກສາ 2019 - 2020 ເຊິ່ງມີລາຍລະອຽດດັ່ງຕໍ່ໄປນີ້:

ເຄື່ອງມືທີ່ໃຊ້ໃນການພັດທະນາລະບົບຂາຍປິລິດເມອອນລາຍຂອງສະຖານນີຂົນສົ່ງ ໂດຍສານສາຍໃຕ້ປະກອບດ້ວຍ 2 ພາກສ່ວນໃຫຍ່ຄື: ພາກສ່ວນ Software ແລະ ພາກສ່ວນ Hardware, ເຊິ່ງພາກສ່ວນ Software ນຳໃຊ້ Subline Text 3 ສຳລັບອອກແບບຟອມພ້ອມທັງ ຂຽນໂປຣແກຣມໃນການຄວບຄຸມ ແລະ ເຊື່ອມຕໍ່ຖານຂໍ້ມູນ, ສຳລັບສ້າງຖານຂໍ້ມູນແມ່ນນຳໃຊ້ phpMyAdmin ເພື່ອຈັດເກັບຂໍ້ມູນຂອງລະບົບ ແລະ ນຳໃຊ້ Microsoft Visio 2016 ໃນການ ແຕ້ມແຜນວາດການໄຫຼຂໍ້ມູນ ແລະ ແຕ້ມແຜນວາດຄວາມສຳພັນ (ER-Diagram). ພາກສ່ວນ Hardware ແມ່ນນຳໃຊ້ຄອມພິວເຕີ 1 ໜ່ວຍ.

ຜົນທີ່ຈະໄດ້ຮັບຈາກລະບົບຂາຍປິລິດເມອອນລາຍຂອງສະຖານນີຂົນສົ່ງໂດຍສານສາຍໃຕ້ ຄື:ລະບົບຈະຊ່ວຍຫຼຸດຜ່ອນຄວາມຊ້ຳຊ້ອນ ແລະ ຫຍຸ້ງຍາກຂອງຂໍ້ມູນເພື່ອອຳນວຍຄວາມສະດວກ

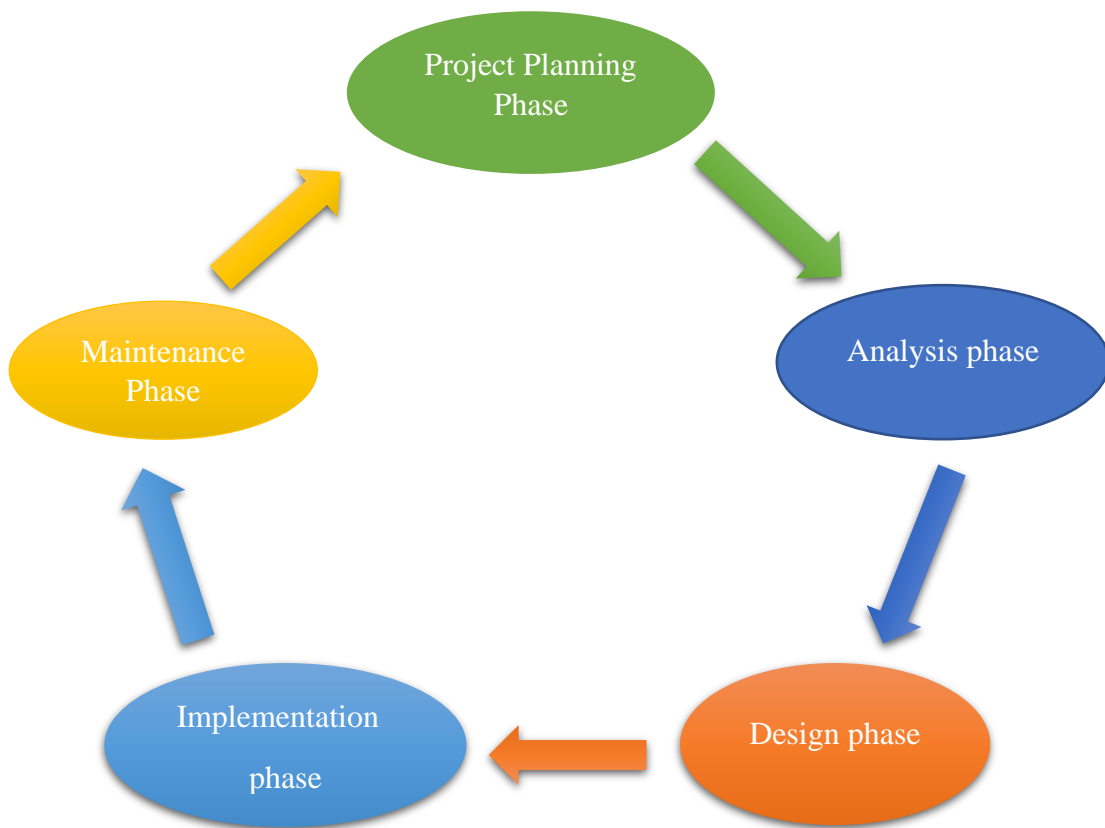
ໃຫ້ແກ່ລູກຄ້າທີ່ມາໃຊ້ບໍລິການ, ສະດວກວ່ອງໄວແກ່ຜູ້ໃຊ້ງານ ແລະ ສ້າງຄວາມເປັນລະບຽບໃຫ້ແກ່ຂໍ້ມູນ. ນອກຈາກນີ້ຍັງບໍລິການລູກຄ້າ, ຄົ້ນຫາ ແລະ ລາຍງານໄດ້ຢ່າງສະດວກວ່ອງໄວພ້ອມຂໍ້ມູນທີ່ຖືກຕ້ອງຊັດເຈນ.

ບົດທີ 3

ວິທີດໍາເນີນການຄົ້ນຄວ້າ

3.1 ວິທີສຶກສາ ແລະ ຄົ້ນຄວ້າ

ເປັນຂະບວນການທີ່ສະແດງເຖິງການດໍາເນີນຂັ້ນຕອນການເຮັດວຽກຂອງລະບົບຕັ້ງແຕ່ຕົ້ນຈົນ
ຈົບ, ບົດໂຄງການຈົບຊັ້ນນີ້ພວກຂ້າພະເຈົ້າໄດ້ນຳໃຊ້ທິດສະດີການວິເຄາະ ແລະ ອອກແບບລະບົບ
ແບບໂຄງທີ່ປະກອບມີ 5 ໄລຍະຄື:



ແຜນວາດທີ 1 : ແຜນວາດລວມຂອງລະບົບ

3.1.1 ໄລຍະທີ 1 ການວາງແຜນໂຄງການ (Project planning phase)

ການວາງແຜນໂຄງການຈັດເປັນຂະບວນການພື້ນຖານໃນຄວາມເຂົ້າໃຈຢ່າງເລິກເຊິ່ງວ່າເປັນ ທັງຕ້ອງສ້າງລະບົບໃໝ່ທີ່ມີງານຕ້ອງພິຈາລະນາວ່າຈະຕ້ອງດຳເນີນງານຕໍ່ໄປແນວໃດກ່ຽວກັບ ຂະບວນການສ້າງລະບົບໃໝ່, ກ່ອນອື່ນໝົດຕ້ອງມີຈຸດກຳເນີດຂອງລະບົບວຽກ ເຊິ່ງໂດຍປົກກະຕິແລ້ວ ຈຸດກຳເນີດຂອງລະບົບວຽກມັກເກີດຂຶ້ນຈາກຜູ້ໃຊ້ລະບົບເປັນຜູ້ທີ່ຕິດແທດກັບລະບົບໂດຍກົງຈຶ່ງເຮັດໃຫ້ ມີຄວາມໃກ້ສືດກັບລະບົບວຽກທີ່ດຳເນີນຢູ່ຫຼາຍທີ່ສຸດເມື່ອຜູ້ໃຊ້ລະບົບມີຄວາມຕ້ອງການປັບປຸງລະບົບ ວຽກ ດັ່ງນັ້ນ ຈຶ່ງຖືເປັນຈຸດເລີ່ມຕົ້ນໃນບົດບາດຂອງນັກວິເຄາະລະບົບວ່າຈະຕ້ອງສຶກສາເຖິງຂອບເຂດ ຂອງບັນຫາທີ່ຜູ້ໃຊ້ລະບົບກຳລັງປະສົບບັນຫາຢູ່ ແລະ ຈະດຳເນີນການແກ້ໄຂແນວໃດເພື່ອສຶກສາເຖິງ ຄວາມເປັນໄປໄດ້ວ່າລະບົບໃໝ່ທີ່ຈະພັດທະນາຂຶ້ນມານັ້ນມີຄວາມເປັນໄປໄດ້ ແລະ ຄຸ້ມຄ່າກັບການຈະ ລົງທຶນ ຫຼື ບໍ່.

ແນວໃດກໍ່ຕາມໄລຍະຂອງການວາງແຜນໂຄງການປົກກະຕິມັກຈະມີໄລຍະເວລາທີ່ສັ້ນ ແຕ່ກໍ່ຖື ວ່າເປັນຂັ້ນຕອນທີ່ສຳຄັນທີ່ຈະໃຫ້ເກີດຜົນສຳເລັດໄດ້. ດັ່ງນັ້ນ ໃນໄລຍະຂອງການວາງແຜນໂຄງການ ຈຶ່ງໄດ້ອາໄສນັກວິເຄາະລະບົບທີ່ມີຄວາມຮູ້ ແລະ ປະສົບການສູງເນື່ອງຈາກວ່າທາງນັກວິເຄາະລະບົບ ບໍ່ເຂົ້າໃຈເຖິງບັນຫາອັນແທ້ຈິງທີ່ເກີດຂຶ້ນກໍ່ຈະບໍ່ສາມາດພັດທະນາລະບົບຂຶ້ນມາເພື່ອແກ້ໄຂບັນຫາໃຫ້ ຖືກຈຸດໄດ້ ແລະ ມັກຈະມີໂຄງການພັດທະນາລະບົບຫຼາຍໂຄງການທີ່ຫຼັງຈາກໄດ້ດຳເນີນການ ພັດທະນາ ແລະ ນຳມາໃຊ້ງານແລ້ວປະກົດວ່າບໍ່ສາມາດຕອບສະໜອງຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ງານ ເຊິ່ງຖືວ່າເປັນເລື່ອງທີ່ກໍ່ໃຫ້ເກີດຄວາມສູນເສຍທັງທາງດ້ານການລົງທຶນ ແລະ ໄລຍະເວລາ.

ສະຫຼຸບລວມແລ້ວໄລຍະຂອງການວາງແຜນໂຄງການປະກອບມີກິດຈະກຳຕ່າງໆດັ່ງນີ້:

- ກຳນົດບັນຫາ Problem Definition.
- ສຶກສາຄວາມເປັນໄປໄດ້ຂອງໂຄງການ Feasibility Study.
- ສ້າງຕາຕະລາງກຳນົດເວລາໂຄງການ Project Scheduling.
- ຈັດຕັ້ງທີມງານໂຄງການ Staff The Project.

3.1.2 ໄລຍະທີ 2 ການວິເຄາະ (Analysis Phase)

ໄລຍະການວິເຄາະຈະຕ້ອງມີຄຳຕອບກ່ຽວກັບຄຳຖາມວ່າໃຜເປັນຜູ້ທີ່ໃຊ້ລະບົບ ແລະ ມີຫຍັງ ແດ່ທີ່ຈະຕ້ອງເຮັດໃນໄລຍະນີ້ນັກວິເຄາະລະບົບຈະຕ້ອງດຳເນີນການໃນຂັ້ນຕອນຂອງການວິເຄາະ ລະບົບງານປັດຈຸບັນ (Current System) ເພື່ອນຳມາພັດທະນາແນວຄວາມຄິດສຳລັບລະບົບໃໝ່ (New System).

ຈຸດປະສົງຫຼັກຂອງການວິເຄາະຄືຈະຕ້ອງສຶກສາ ແລະ ສ້າງຄວາມເຂົ້າໃຈໃນຄວາມຕ້ອງການ ຕ່າງໆທີ່ໄດ້ລວບລວມມາ. ດັ່ງນັ້ນ ການລວບລວມຄວາມຕ້ອງການ (Requirements Gathering) ຈຶ່ງ ຈັດເປັນວຽກພື້ນຖານຂອງການວິເຄາະລະບົບໂດຍຂໍ້ມູນຄວາມຕ້ອງການເຫຼົ່ານີ້ນັກວິເຄາະລະບົບຈະນຳ ມາວິເຄາະເພື່ອທີ່ຈະປະເມີນວ່າຄວນມີຫຍັງແດ່ທີ່ລະບົບໃໝ່ຕ້ອງດຳເນີນການ ແລະ ດ້ວຍເຫດນີ້ເອງ ການກຳນົດລາຍລະອຽດກ່ຽວກັບຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ (User Requirements) ຈະເພີ່ມຄວາມ ສຳຄັນຫຼາຍຂຶ້ນເປັນລຳດັບສຳລັບລະບົບທີ່ມີຄວາມສັບຊ້ອນສູງ ແລະ ນັກວິເຄາະຕ້ອງເອົາໃນໃສ່ກັບ ການລວບລວມຄວາມຕ້ອງການຈາກຜູ້ໃຊ້ບໍ່ຄວນກຳນົດຄວາມຕ້ອງການຂຶ້ນເອງໂດຍໃຊ້ຄວາມຄິດ ສ່ວນຕົວຂອງຕົນເອງເປັນຫຼັກ ຫຼື ປະເມີນຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ລະບົບບໍ່ເຂົ້າກັບຈຸດປະສົງ ແລະ ຫາກມີການພັດທະນາລະບົບຕໍ່ໄປຈົນແລ້ວລະບົບວຽກທີ່ໄດ້ກໍຈະບໍ່ກົງກັບຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ ລະບົບຢ່າງແທ້ຈິງເຮັດໃຫ້ຕ້ອງມີຄວາມປັບປຸງ ຫຼື ປ່ຽນແປງພາຍຫຼັງ.

ນັກວິເຄາະລະບົບ ສາມາດລວບລວມຄວາມຕ້ອງການຕ່າງໆໄດ້ຈາກການສັງເກດ, ການເຮັດ ວຽກຂອງຜູ້ໃຊ້, ການໃຊ້ເຕັກນິກ, ການສຳພາດ ຫຼື ການໃຊ້ແບບສອບຖາມການອ່ານເອກະສານກ່ຽວ ກັບການປະຕິບັດງານຂອງລະບົບວຽກປັດຈຸບັນລະບຽບກົດເກນຂອງບໍລິສັດ ແລະ ການມອບໝາຍຕຳ ແໜ່ງໜ້າທີ່ຮັບຜິດຊອບເຊິ່ງໃນຊ່ວງຂອງການເກັບກຳຂໍ້ມູນຄວາມຕ້ອງການກໍຈະພົບກັບຜູ້ໃຊ້ໃນລະດັບ ຕ່າງໆທີ່ເຮັດໃຫ້ຮູ້ເຖິງບັນຫາ ແລະ ແນວທາງການແກ້ໄຂບັນຫາທີ່ແນະນຳໂດຍຜູ້ໃຊ້. ດັ່ງນັ້ນ ການ ເກັບກຳຄວາມຕ້ອງການນັ້ນຈຶ່ງເປັນກົດຈະກຳທີ່ສຳຄັນເພື່ອຄົ້ນຫາຄວາມຈິງ ແລະ ຕ້ອງສ້າງຄວາມ ຕ້ອງການເຊິ່ງກັນ ແລະ ກັນ ເພື່ອສະຫຼຸບອອກມາເປັນຂໍ້ກຳນົດທີ່ມີຄວາມຊັດເຈນ ໂດຍຂໍ້ກຳນົດເຫຼົ່ານີ້ ເມື່ອຜູ້ໃຊ້ທີ່ກ່ຽວຂ້ອງໄດ້ອ່ານແລ້ວຈະຕ້ອງສຶກສາຄວາມໝາຍໄດ້ກົງກັນ.

ຫຼັງຈາກໄດ້ນຳຄວາມຕ້ອງການຕ່າງໆມາສະຫຼຸບເປັນຂໍ້ກຳນົດທີ່ຊັດເຈນແລ້ວ ຂັ້ນຕອນຕໍ່ໄປກໍ ຄືນຳຂໍ້ກຳນົດເຫຼົ່ານັ້ນໄປພັດທະນາອອກມາເປັນຄວາມຕ້ອງການຂອງລະບົບໃໝ່ ໂດຍເຕັກນິກທີ່ໃຊ້ກໍຄື ການພັດທະນາແບບຈຳລອງຂະບວນການ (Process Model) ເຊິ່ງເປັນແຜ່ນພາບທີ່ໃຊ້ອະທິບາຍເຖິງ ຂະບວນການທີ່ຕ້ອງດຳເນີນໃນລະບົບວ່າມີຫຍັງແດ່ ແລະ ຕໍ່ໄປກໍດຳເນີນການພັດທະນາແບບຈຳລອງ ຂໍ້ມູນ(Data Model) ເພື່ອອະທິບາຍເຖິງຂໍ້ມູນທີ່ຈັດເກັບໄວ້ສຳລັບສະໜັບສະໜູນການເຮັກວຽກຕ່າງໆ.

- ສະຫຼຸບໄລຍະຂອງການວິເຄາະລະບົບປະກອບມີກົດຈະກຳຕ່າງໆດັ່ງນີ້:
 - ວິເຄາະລະບົບງານປັດຈຸບັນ.
 - ເກັບກຳຄວາມຕ້ອງການໃນດ້ານຕ່າງໆ ແລະ ນຳມາວິເຄາະເພື່ອສະຫຼຸບເປັນຂໍ້ກຳນົດ ຊັດເຈນ.

- ນຳຂໍ້ກຳນົດມາພັດທະນາອອກມາເປັນຄວາມຕ້ອງການຂອງລະບົບໃໝ່.
- ສ້າງແບບຈຳລອງຂະບວນການຂອງລະບົບໃໝ່ໂດຍການແຕ້ມແຜນພາບກະແສຂໍ້ມູນ (DataFlow Diagram: DFD).
- ສ້າງແບບຈຳລອງຂໍ້ມູນໂດຍການແຕ້ມ Entity Relationship Diagram: ERD.

3.1.3 ໄລຍະທີ 3 ການອອກແບບ (Design Phase)

ໄລຍະການອອກແບບເປັນການພິຈາລະນາວ່າລະບົບລະດຳເນີນການໄປໄດ້ແນວໃດ ເຊິ່ງກ່ຽວຂ້ອງກັບຍຸດທະວິທີການຂອງການອອກແບບທີ່ວ່າດ້ວຍການຕັດສິນໃຈວ່າຈະພັດທະນາລະບົບໃໝ່ດ້ວຍແນວທາງໃດເຊັ່ນ ພັດທະນາຂຶ້ນເອງ, ຊື້ໂປຣແກຣມສຳເລັດຮູບ ຫຼື ວ່າຈ້າງບໍລິສັດພັດທະນາລະບົບໃຫ້ເປັນຕົ້ນ. ນອກຈາກນີ້ໄລຍະການອອກແບບຈະກ່ຽວຂ້ອງກັບການອອກແບບທາງດ້ານສະຖາປັດຕະຍະກຳລະບົບທີ່ກ່ຽວຂ້ອງກັບອຸປະກອນຮາດແວ, ຊອບແວ ແລະ ເຄືອຂ່າຍ.

ການອອກແບບລາຍງານ (Out Design) ການອອກແບບໜ້າຈໍເພື່ອປະຕິສຳພັນກັບຜູ້ໃຊ້ງານ (User Interface), ການອອກແບບຜັງງານລະບົບ (System Flowchart), ເຊິ່ງລວມເຖິງລາຍລະອຽດຂອງໂປຣແກຣມ (Specific Program), ຖານຂໍ້ມູນ (Database) ແລະ ຂໍ້ມູນທີ່ກ່ຽວຂ້ອງແນວໃດກໍຕາມເຖິງວ່າກິດຈະກຳບາງສ່ວນຂອງໄລຍະອອກແບບນີ້ສ່ວນໃຫຍ່ຈະຖືກດຳເນີນການໄປບາງສ່ວນແລ້ວໃນໄລຍະຂອງການວິເຄາະແຕ່ໄລຍະການອອກແບບນີ້ຈະເນັ້ນເຖິງການດຳເນີນການແກ້ໄຂບັນຫາແນວໃດຫຼາຍກວ່າໂດຍການນຳຜົນຂອງແບບຈຳລອງທາງ Logical Model. ທີ່ໄດ້ຈາກການວິເຄາະມາພັດທະນາມາເປັນແບບຈຳລອງທາງ Physical Model.

- ການວິເຄາະຈະເນັ້ນແກ້ໄຂບັນຫາຫຍັງແດ່.
- ການອອກແບບຈະເນັ້ນການແກ້ໄຂບັນຫາແນວໃດ.
- ສະຫຼຸບໄລຍະການອອກແບບ.
- ພິຈາລະນາແນວທາງໃນການພັດທະນາລະບົບ.
- ອອກແບບສະຖາປັດຕະຍະກຳລະບົບ (Architecture Design).
- ອອກແບບຖານຂໍ້ມູນ (Database Design).
- ອອກແບບການສະແດງຜົນ (Output Design).
- ອອກແບບການປ້ອນຂໍ້ມູນ (Input Design).
- ອອກແບບສ່ວນຕິດຕໍ່ກັບຜູ້ໃຊ້ (User Interface).
- ສ້າງຕົ້ນແບບ (Prototype).

- ອອກແບບໂປຣແກມ (Structure Chart).

3.1.4 ໄລຍະທີ 4 ການນຳໄປໃຊ້ (Implementation Phase)

ໃນໄລຍະການນຳໄປໃຊ້ຈະເຮັດໃຫ້ລະບົບເກີດຜົນຂຶ້ນມາໂດຍການສ້າງລະບົບທົດສອບລະບົບ ແລະ ການຕິດຕັ້ງລະບົບໂດຍຈຸດປະສົງຫຼັກຂອງກິດຈະກຳໃນໄລຍະນີ້ ບໍ່ແມ່ນພຽງຄວາມໜ້າເຊື່ອຖືຂອງລະບົບ ຫຼື ລະບົບສາມາດເຮັດວຽກໄດ້ດີເທົ່ານັ້ນ, ແຕ່ຕ້ອງໝັ້ນໃຈວ່າຜູ້ໃຊ້ລະບົບຕ້ອງໄດ້ຮັບການເຝິກອົບຮົມເພື່ອໃຊ້ງານລະບົບ ແລະ ຄວາມຄາດຫວັງໃນອົງກອນທີ່ຕ້ອງການຜົນຕອບແທນໃນດ້ານດີກັບການໃຊ້ລະບົບໃໝ່ລຳດັບກິດຈະກຳຕ່າງໆ. ທຸກກິດຈະກຳຕ້ອງເຂົ້າມາດຳເນີນການຮ່ວມກັນໃນໄລຍະນີ້ເພື່ອໃຫ້ລະບົບການປະຕິບັດງານໄດ້ຮັບຄວາມປະສົບຜົນສຳເລັດໄດ້ໂດຍດີ.

➤ ສະຫຼຸບໄລຍະການນຳໄປໃຊ້ຈະປະກອບມີກິດຈະກຳຕ່າງໆດັ່ງຕໍ່ໄປນີ້:

- ສ້າງລະບົບຂຶ້ນມາດ້ວຍການຂຽນໂປຣແກຣມ.
- ກວດສອບຄວາມຖືກຕ້ອງທາງດ້ານ Verification ແລະ Validation ແລະ ດຳເນີນການທົດສອບລະບົບ.
- ແປງຂໍ້ມູນ (Convert Data).
- ຕິດຕັ້ງລະບົບ (System Installation) ແລະ ສ້າງເອກກະສານຄູ່ມື.
- ເຝິກອົບຮົມຜູ້ໃຊ້ ແລະ ປະເມີນຜົນລະບົບໃໝ່.

ສຳລັບການສ້າງລະບົບ ຫຼື ການຂຽນໂປຣແກຣມນັ້ນ ສາມາດໃຊ້ວິທີການຂຽນໂປຣແກຣມດ້ວຍພາສາຄອມພິວເຕີເຊັ່ນ: ການໃຊ້ພາສາ Visual Basic, C#, PHP, Java... ນອກຈາກນີ້ ຍັງມີເຕັກນິກອື່ນໆເຊັ່ນ: ເຄື່ອງມືໃນການພັດທະນາ Application ເຊິ່ງເປັນຊອບແວທີ່ເປັນແຫຼ່ງລວມຂອງເຄື່ອງມືຕ່າງໆທີ່ໃຊ້ເພື່ອພັດທະນາ Application ເຮັດໃຫ້ຜູ້ຂຽນໂປຣແກຣມບໍ່ເຮັດວຽກໜັກຄືເມື່ອກ່ອນມີແຕ່ຮຽນຮູ້ ແລະ ປະຍຸກໃຊ້ເຄື່ອງມືເຫຼົ່ານັ້ນກໍ່ສາມາດພັດທະນາລະບົບງ່າຍຂຶ້ນ.

3.1.5 ໄລຍະທີ 5 ການບຳລຸງຮັກສາ (Maintenance Phase)

ໂດຍປົກກະຕິແລ້ວໄລຍະການບຳລຸງຮັກສາຈະບໍ່ນຳເຂົ້າໄປລວມໃນສ່ວນຂອງ SDLC ຈົນລະບົບມີການຕິດຕັ້ງເພື່ອໃຊ້ງານແລ້ວເທົ່ານັ້ນ. ໄລຍະນີ້ຈະໃຊ້ເວລາຍາວນານທີ່ສຸດເມື່ອທຽບກັບໄລຍະອື່ນໆທີ່ຜ່ານມາເນື່ອງຈາກລະບົບຈະຕ້ອງໄດ້ຮັບການບຳລຸງຮັກສາຕະຫຼອດໄລຍະເວລາທີ່ມີການໃຊ້ລະບົບສິ່ງທີ່ຄາດຫວັງຂອງໜ່ວຍງານກໍ່ຄືຕ້ອງການໃຫ້ລະບົບໃຊ້ງານຍາວນານຫຼາຍປີລະບົບສາມາດຮອງຮັບເຕັກໂນໂລຊີໃໝ່ໆໃນອະນາຄົດໄດ້. ດັ່ງນັ້ນ ໃນຊ່ວງໄລຍະເວລາດັ່ງກ່າວຈຶ່ງສາມາດເພີ່ມເຕີມ

ຄວາມສາມາດຂອງລະບົບໃຫ້ມີປະສິດທິພາບສູງຂຶ້ນພ້ອມທັງການແກ້ໄຂບັບປຸງໂປຣແກຣມໃນກໍລະນີທີ່ເຫັນຂໍ້ຜິດພາດ.

- ການບໍາລຸງຮັກສາລະບົບ (System Maintenance).
- ການເພີ່ມເຕີມຄວາມສາມາດໃໝ່ໆເຂົ້າໃນລະບົບ (Enhance System).
- ສະໜັບສະໜູນງານຂອງຜູ້ໃຊ້ (Support the User).

ຈາກໄລຍະຕ່າງໆຕາມຂັ້ນຕອນການພັດທະນາລະບົບຕາມແບບແຜນຂອງ SDLC ຈະເຫັນວ່າມີການໃຊ້ຄຳວ່າໄລຍະ ແລະ ກິດຈະກຳເຊິ່ງສາມາດອະທິບາຍລາຍລະອຽດເພື່ອໃຫ້ເກີດຄວາມເຂົ້າໃຈກົງກັນດັ່ງນີ້:

- ໄລຍະ (Phase) ຄືກຸ່ມກິດຈະກຳທີ່ກ່ຽວຂ້ອງກັນ.
- ກິດຈະກຳ (Activity) ຄືກຸ່ມຂອງງານທີ່ກ່ຽວຂ້ອງກັນ.
- ໜ້າວຽກ (Task) ຄືວຽກທີ່ດຳເນີນການເຊິ່ງຖືເປັນວຽກທີ່ນ້ອຍທີ່ສຸດ.

3.2 ເຄື່ອງມືທີ່ໃຊ້ໃນການພັດທະນາ (Development Tools)

1) Hardware:

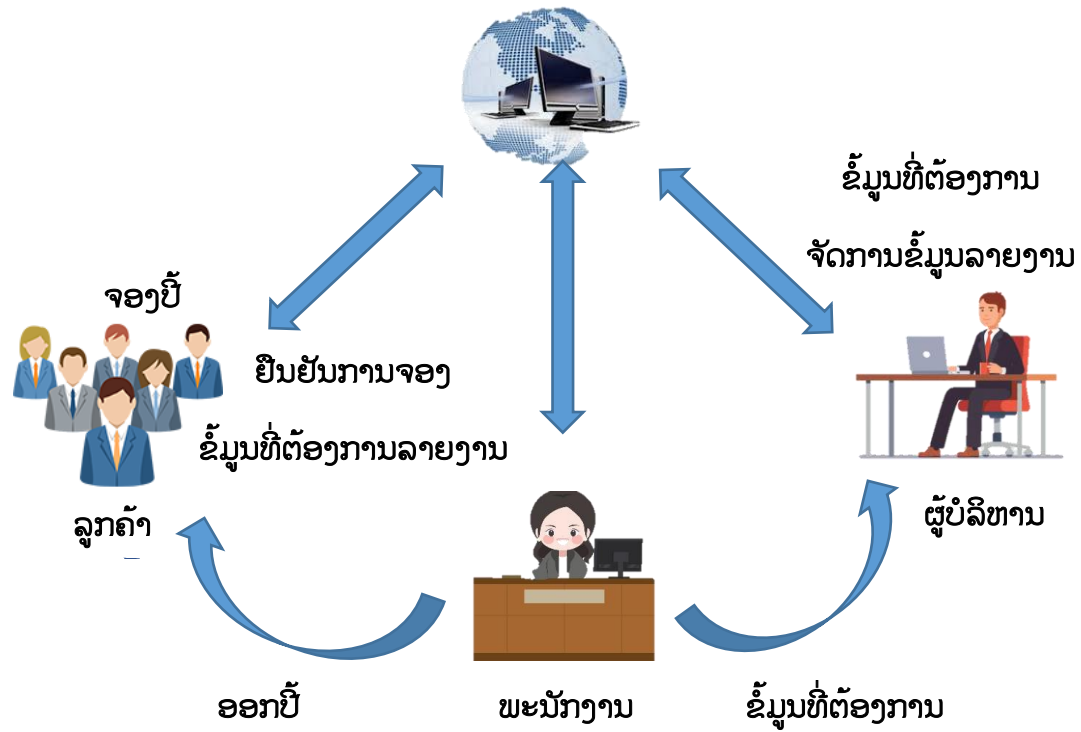
- ເຄື່ອງຄອມພິວເຕີ 1 ໜ່ວຍລຸ້ນ Lenovo Intel(R) Core(TM) I5-3230M
- ເຄື່ອງຄອມພິວເຕີທີ່ໃຊ້ສຳລັບຂຽນໂປຣແກຣມມີ Spec ດັ່ງນີ້:
- CPU @2.60GHz,
- RAM 8GB DDR3L 1600MHz.
- SSD 240 GB, HDD 500 GB.
- Pocket WiFi Modem ເພື່ອໃຊ້ເຊື່ອມຕໍ່ Internet.

2) Software:

- ລະບົບປະຕິບັດການ Windows 10 Professional 64 Bit.
- Microsoft Visio 2016 ໃຊ້ແຕ້ມແຜນວາດການໄຫຼຂໍ້ມູນ (DFD, ER, Flowchart).
- Adobe XD ໃຊ້ອອກແບບ UX/UI.
- Studio 3T For MongoDB ແລະ Moon Modeler ໃຊ້ຈັດການຖານຂໍ້ມູນ ແລະ ອອກແບບ Database Model.
- Visual Studio Code ໃຊ້ຂຽນໂຄດດ້ວຍພາສາ JavaScript (ReactJS, NodeJS, GraphQL).
- MS Office 2013 Professional ໃຊ້ເພື່ອສ້າງ.

3.3 ການວິເຄາະຂໍ້ມູນ

3.3.1 ແຜນວາດລວມຂອງລະບົບ



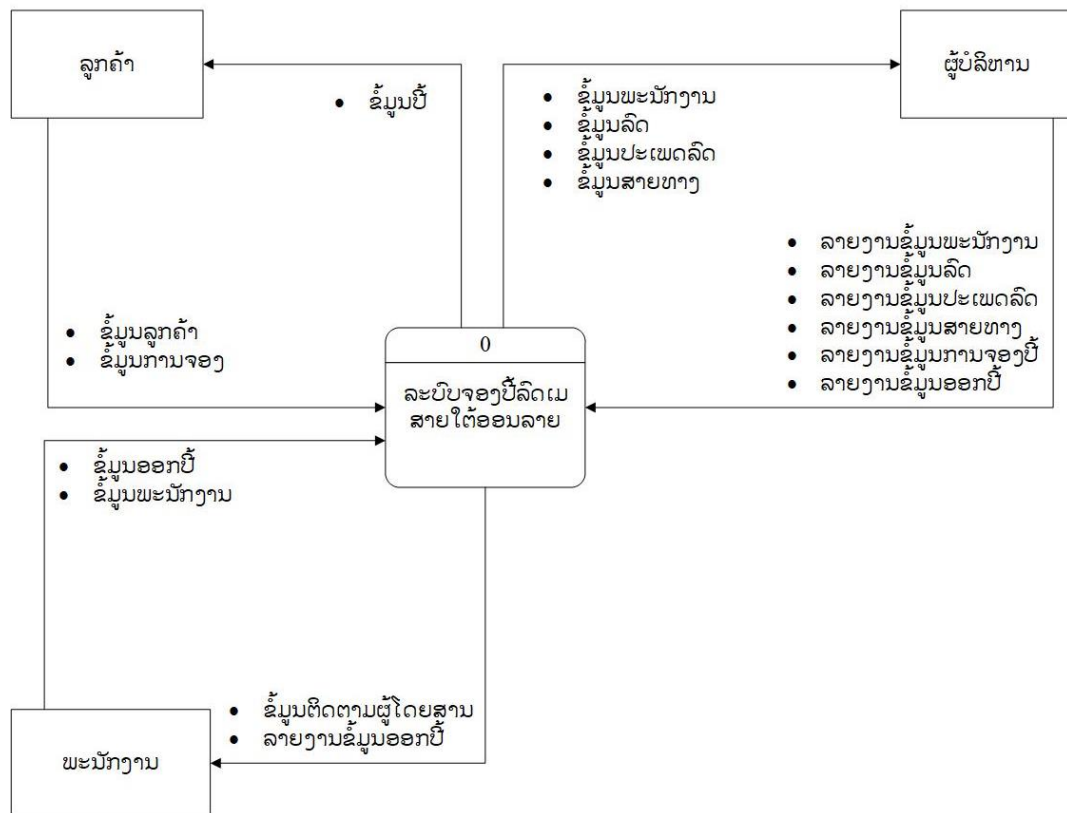
ແຜນວາດທີ 2 : ແຜນວາດລວມຂອງລະບົບ

3.1.2 ຕາຕະລາງສະແດງລາຍລະອຽດຕ່າງໆທີ່ກ່ຽວກັບລະບົບ

External Entity	Process	Data Store
ລູກຄ້າ ພະນັກງານ ຜູ້ບໍລິຫານ	1. ຈັດການຂໍ້ມູນພື້ນຖານ 1.1. ຈັດການຂໍ້ມູນພະນັກງານ 1.2. ຈັດການຂໍ້ມູນບໍລິສັດ 1.3. ຈັດການຂໍ້ມູນປະເພດລົດ 1.4. ຈັດການຂໍ້ມູນລົດ 1.5. ຈັດການຂໍ້ມູນສາຍທາງ 2. ລົງທະບຽນ 2.1. ສະມັກສະມາຊິກ 3. ບໍລິການ 3.1. ຈອງປີ້ 3.2. ອອກປີ້ 4. ລາຍງານ 4.1. ຂໍ້ມູນການຈອງ 4.2. ຂໍ້ມູນພະນັກງານ 4.3. ຂໍ້ມູນສາຍທາງ 4.4. ຂໍ້ມູນລົດ 4.5. ຂໍ້ມູນອອກປີ້	D1 ຂໍ້ມູນພະນັກງານ D2 ຂໍ້ມູນລູກຄ້າ D3 ຂໍ້ມູນບໍລິສັດ D4 ຂໍ້ມູນລົດ D5 ຂໍ້ມູນປະເພດລົດ D6 ຂໍ້ມູນບ່ອນນັ່ງ D7 ຂໍ້ມູນສາຍທາງ D8 ຂໍ້ມູນເວລາລົດອອກ D9 ຂໍ້ມູນການຈອງ D10 ຂໍ້ມູນລາຍລະອຽດການຈອງ D11 ຂໍ້ມູນອອກປີ້ D12 ຂໍ້ມູນຕິດຕາມຜູ້ໂດຍສານ

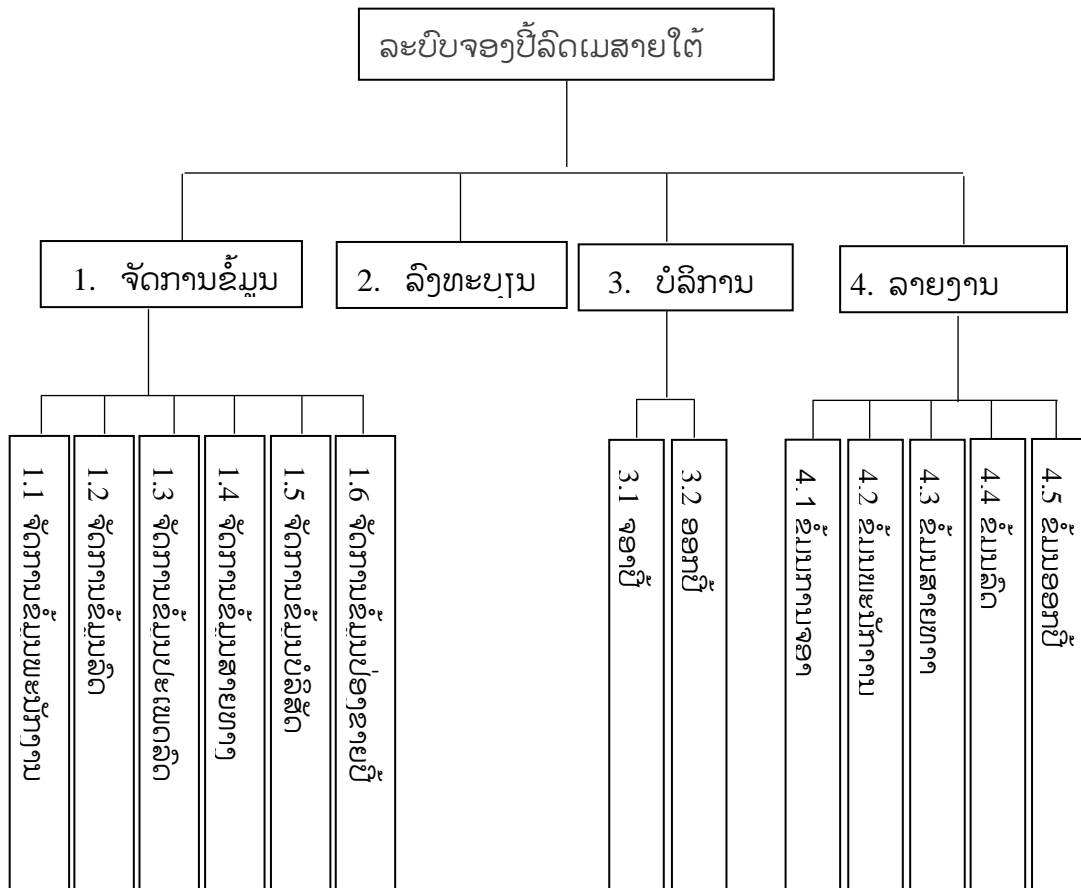
ຕາຕະລາງທີ 5 : ສະແດງລາຍລະອຽດຕ່າງໆທີ່ກ່ຽວກັບລະບົບ

3.3.3 ແຜນວາດເນື້ອຫາ (Context Diagram)



ແຜນວາດທີ 3 : ແຜນວາດເນື້ອຫາ (Context Diagram)

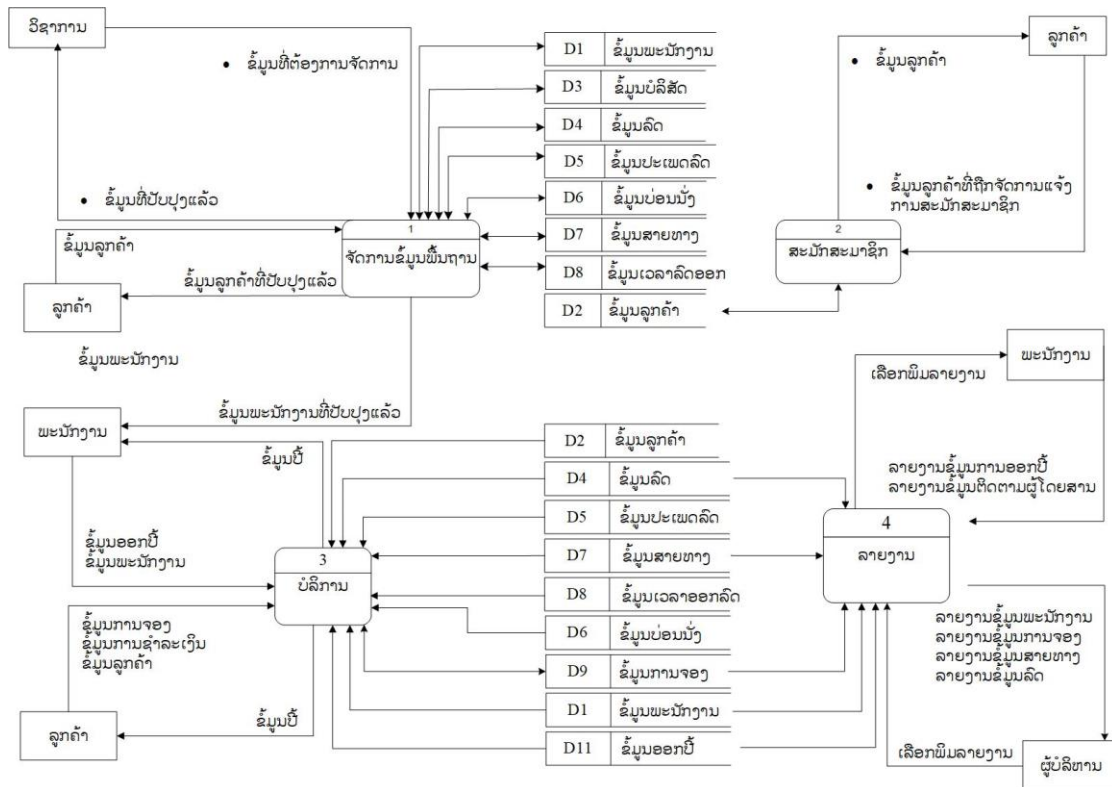
3.1.4 ແຜນວາດລຳດັບຊັ້ນໜ້າທີ່ (Functional Hierarchy Diagram)



ແຜນວາດທີ 4 : ແຜນວາດລຳດັບຊັ້ນໜ້າທີ່ (Functional Hierarchy Diagram)

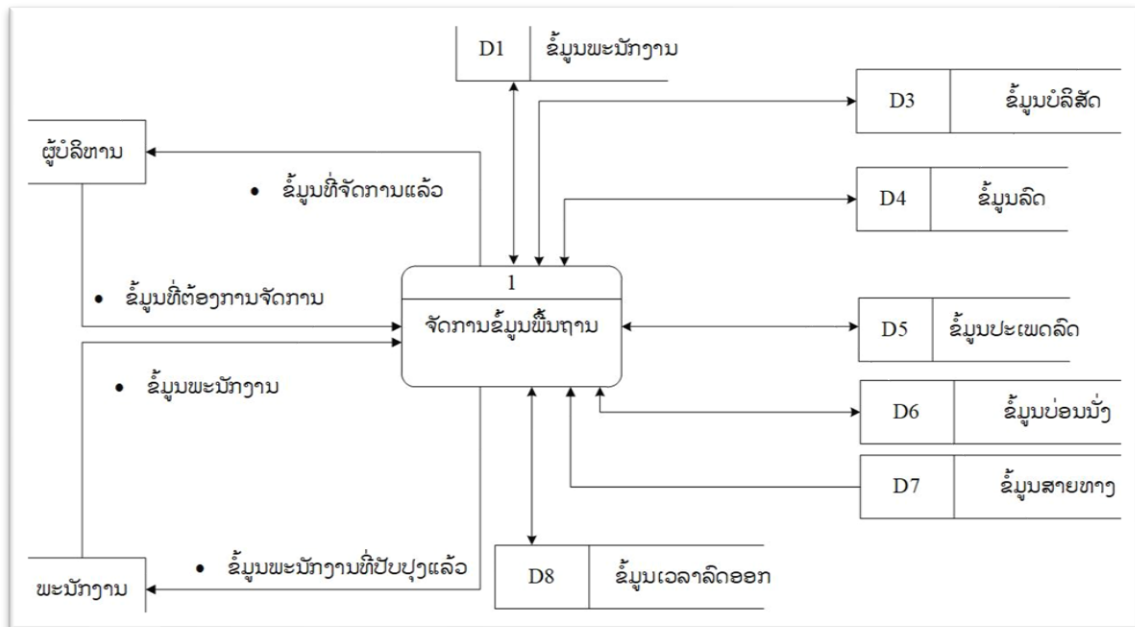
3.1.5 ແຜນວາດການໄຫຼຂໍ້ມູນ (Data Flow Diagram: DFD)

1) ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງແຕ່ລະ Process



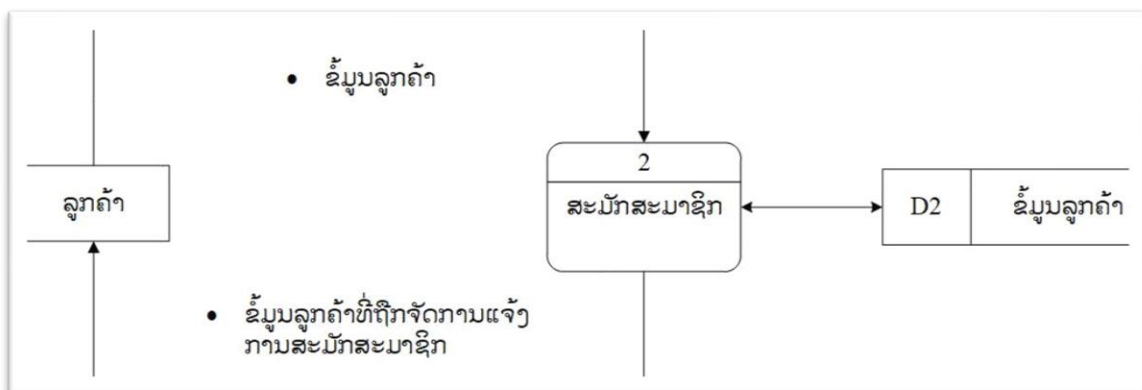
ແຜນວາດທີ 5 : ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງແຕ່ລະ Process

2) ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 1



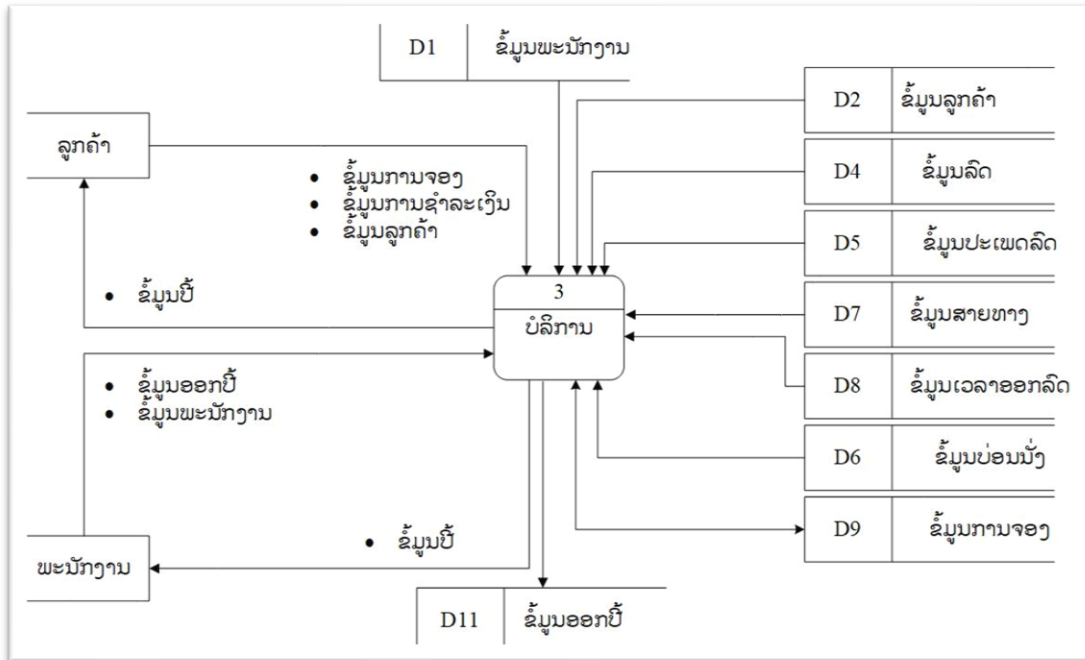
ແຜນວາດທີ 6 : ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 1

3) ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 2



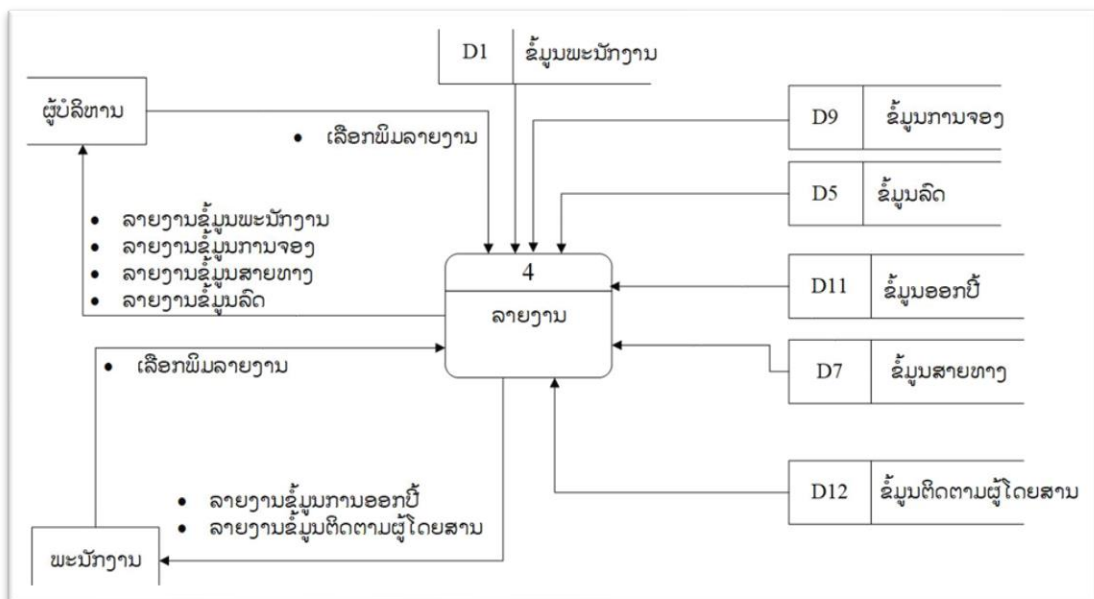
ແຜນວາດທີ 7 : ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 2

4) ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 3



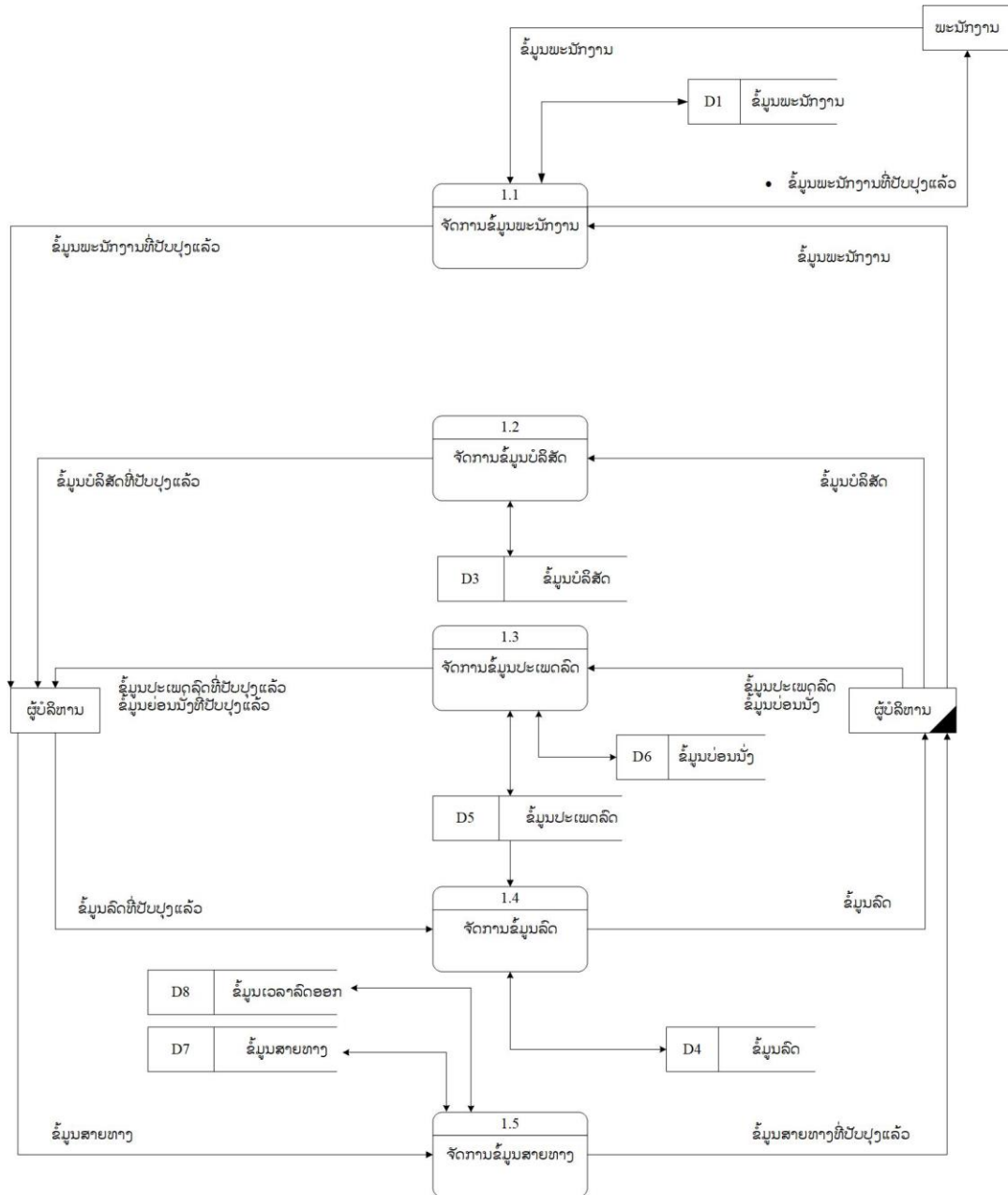
ແຜນວາດທີ 8 : ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 3

5) ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 4



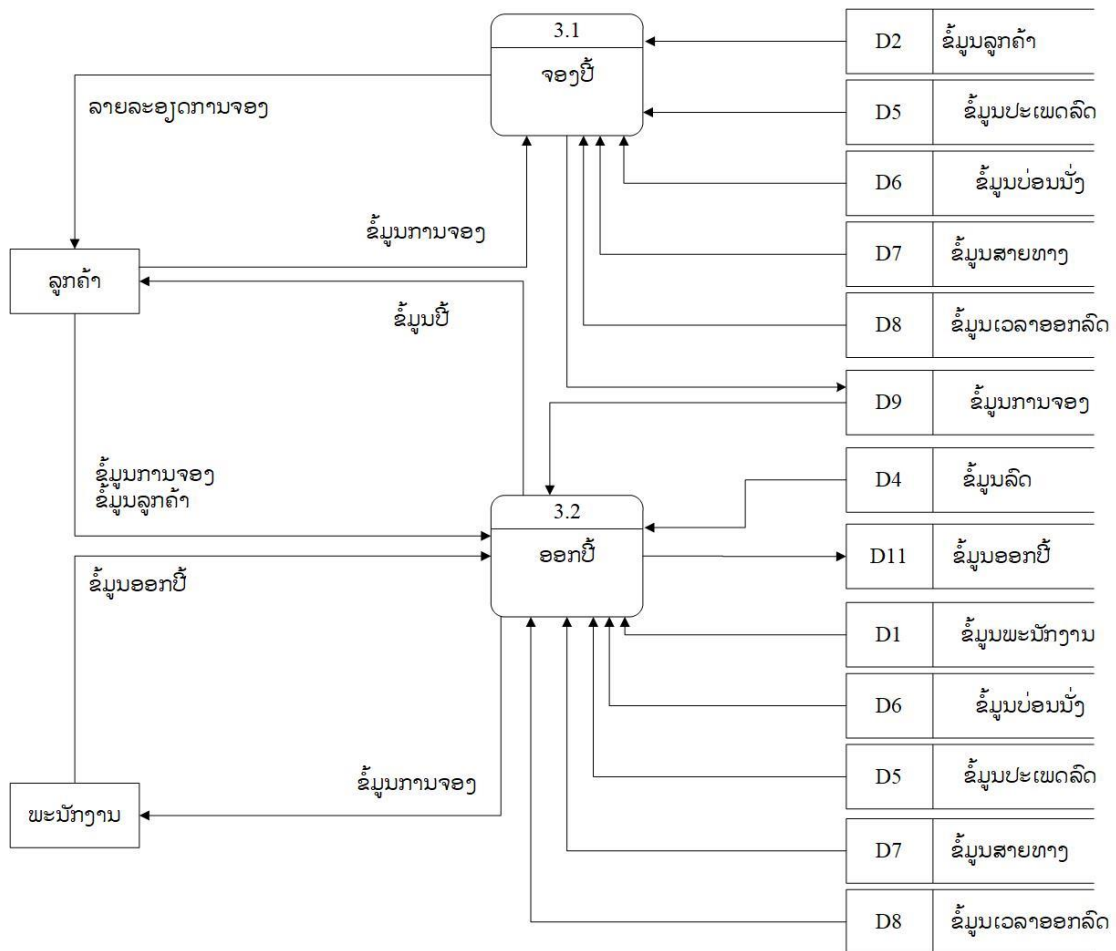
ແຜນວາດທີ 9 : ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 4

6) ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 1



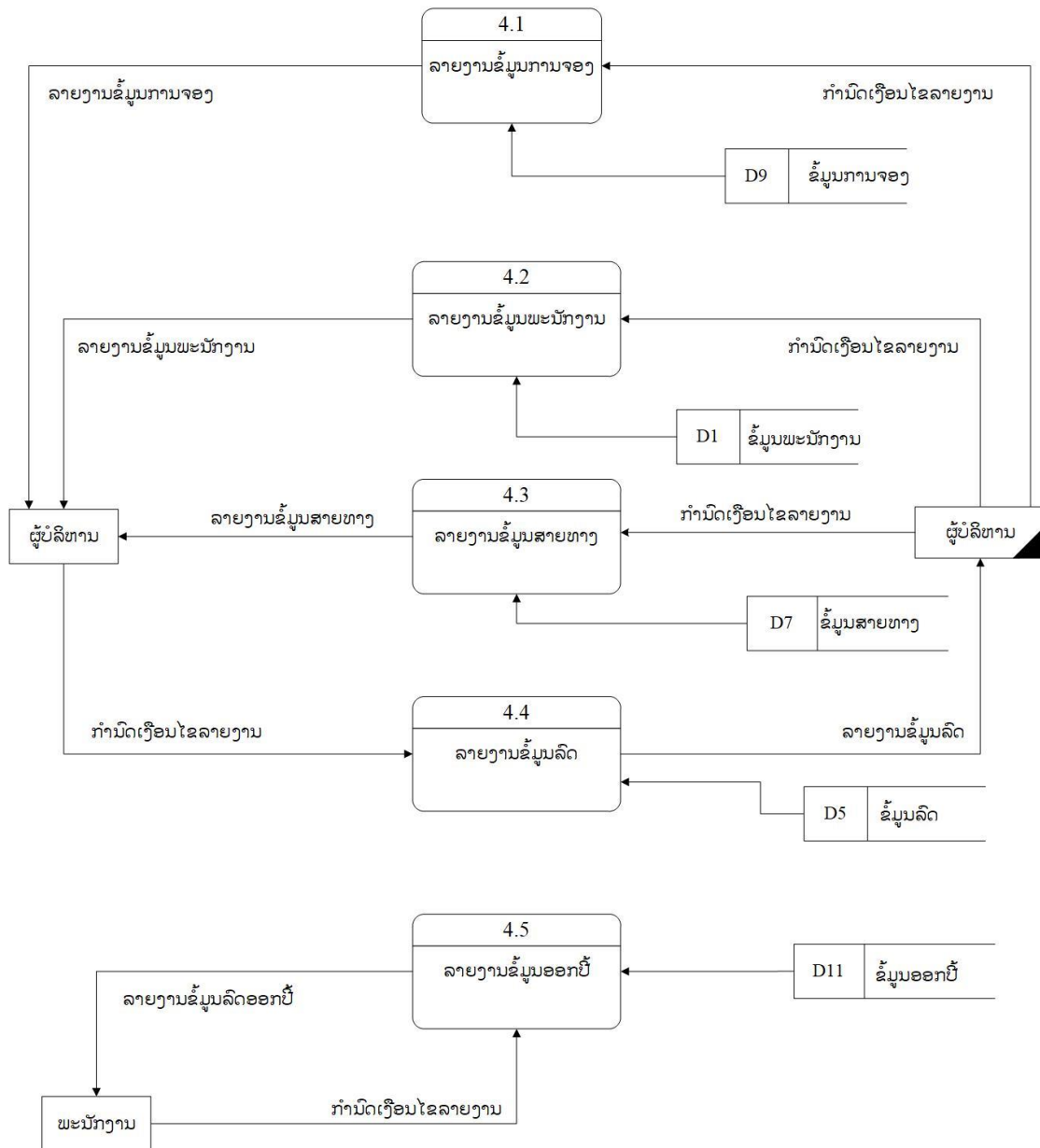
ແຜນວາດທີ 10 : ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 1

7) ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 3



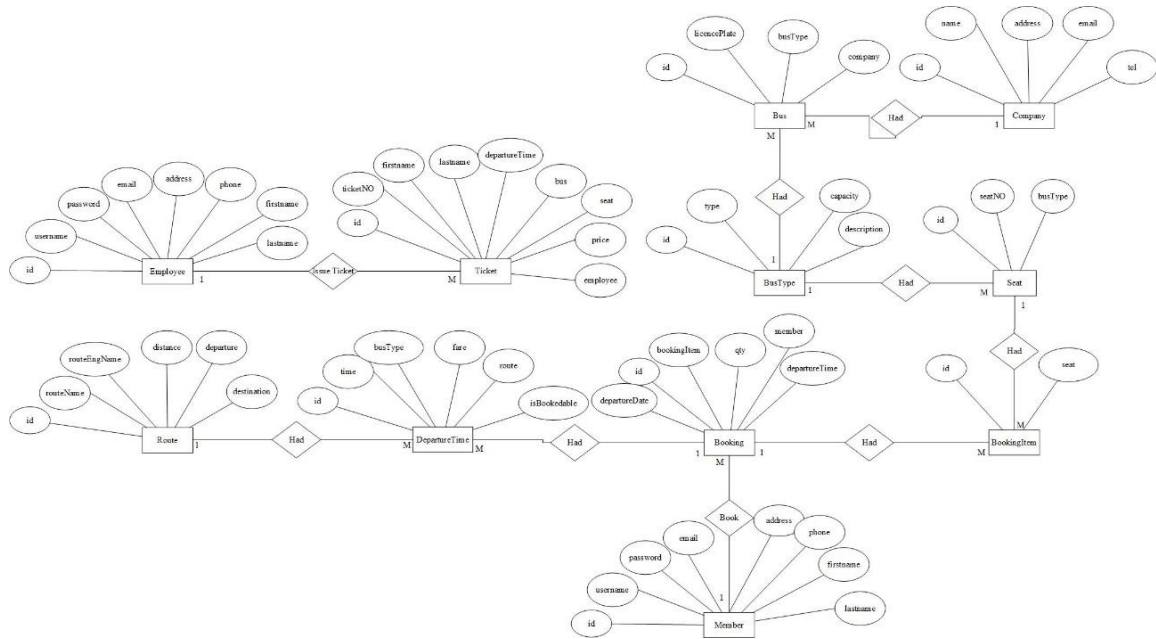
ແຜນວາດທີ 11 : ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 3

8) ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 4



ແຜນວາດທີ 12 : ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 4

9) ແຜນວາດຄວາມສຳພັນຂອງຂໍ້ມູນ (ER Diagram)



ແຜນວາດທີ 13 : ແຜນວາດຄວາມສຳພັນຂອງຂໍ້ມູນ (ER Diagram)

3.4 ການອອກແບບລະບົບ

3.4.1 ການອອກແບບຮ່າງສະແດງຜົນ (Output Design)

1) ລາຍງານຂໍ້ມູນລົດ

ສາທາລະນະລັດ ປະຊາທິປະໄຕ ປະຊາຊົນລາວ

ສັນຕິພາບ ເອກະລາດ ປະຊາທິປະໄຕ ເອກະພາບ ວັດທະນາຖາວອນ



ລາຍງານຂໍ້ມູນລົດ

#	ທະບຽນລົດ	ປະເພດລົດ	ລົດບໍລິສັດ
1	ຈຂ 3232	ທຳມະດາ	ບໍລິສັດກຽງໄກ
2	ຈກ 3342	VIP	ບໍລິສັດຈິດປະສົງ
3	ຈດ 254	VIP	ບໍລິສັດກຽງໄກ

ວັນທີ 08/08/2021

ລາຍເຊັນຜູ້ລາຍງານ.....

ຮູບທີ 6 : ຮູບອອກແບບສະແດງລາຍງານຂໍ້ມູນລົດ

2) ລາຍງານຂໍ້ມູນສາຍທາງ

ສາທາລະນະລັດ ປະຊາທິປະໄຕ ປະຊາຊົນລາວ

ສັນຕິພາບ ເອກະລາດ ປະຊາທິປະໄຕ ເອກະພາບ ວັດທະນາຖາວອນ



ລາຍງານຂໍ້ມູນສາຍທາງ

#	ຊື່ສາຍທາງ	ຊື່ພາສາອັງກິດ	ໄລຍະທາງ
1	ວຽງຈັນ - ຫຼັກ 20	VTE - LAK 20	110
2	ວຽງຈັນ - ປາກເຊ	VTE-PAKSE	120
3	ວຽງຈັນ - ຈຳປາສັກ	VTE-CS	254

ວັນທີ 08/08/2021

ລາຍເຊັນຜູ້ລາຍງານ.....

ຮູບທີ 7 : ຮູບອອກແບບລາຍງານຂໍ້ມູນສາຍທາງ

3) ລາຍງານຂໍ້ມູນພະນັກງານ

ສາທາລະນະລັດ ປະຊາທິປະໄຕ ປະຊາຊົນລາວ

ສັນຕິພາບ ເອກະລາດ ປະຊາທິປະໄຕ ເອກະພາບ ວັດທະນາຖາວອນ



ລາຍງານຂໍ້ມູນພະນັກງານ

#	ຊື່ນາມສະກຸນ	ທີ່ຢູ່	ອີເມວ	ເບີໂທ	USERNAME
1	phonekham keomany	sangkhou	phone.madridista@gmail.com	99491232	phone
2	phonekham keomany	sangkhou	phone.madridista@gmail.com	99491232	phone2
3	phonekham keomany	sangkhou	phone.madridista@gmail.com	99491232	phonekham
4	rt4t wrw3r	w3rw3r3	phone.madridista@gmail.com	2028022677	phones

ວັນທີ 08/08/2021


ລາຍເຊັນລາຍງານ.....

ຮູບທີ 8 : ຮູບອອກແບບສະແດງລາຍງານຂໍ້ມູນພະນັກງານ

4) ລາຍງານຂໍ້ມູນການຈອງປີ້

ສາທາລະນະລັດ ປະຊາທິປະໄຕ ປະຊາຊົນລາວ

ສັນຕິພາບ ເອກະລາດ ປະຊາທິປະໄຕ ເອກະພາບ ວັດທະນາຖາວອນ



ລາຍງານຂໍ້ມູນການຈອງປີ້

#	ຈອງວັນທີ	ຊື່ແລະນາມສະກຸນ	ເບີໂທ	ອີເມວ	ວັນທີ່ເດີນທາງ	ສາຍທາງ	ຖ້ຽວລືດ	ສະຖານະ
1	07/08/2021	phone keomany	99491232	phone@gmail.com	27/08/2021	ວຽງຈັນ - ຫຼີກ 20	8:30	<input type="button" value="ຈ່າຍແລ້ວ"/>
2	07/08/2021	phone keomany	99491232	phone@gmail.com	26/08/2021	ວຽງຈັນ - ປາກເຊ	8:00	<input type="button" value="ລໍຖ້າກວດສອບ"/>
3	09/08/2021	phone keomany	99491232	phone@gmail.com	18/08/2021	ວຽງຈັນ - ປາກເຊ	8:00	<input type="button" value="ຈ່າຍແລ້ວ"/>
4	09/08/2021	phone keomany	99491232	phone@gmail.com	25/08/2021	ວຽງຈັນ - ປາກເຊ	8:00	<input type="button" value="ລໍຖ້າກວດສອບ"/>
5	09/08/2021	phone keomany	99491232	phone@gmail.com	25/08/2021	ວຽງຈັນ - ປາກເຊ	8:00	<input type="button" value="ລໍຖ້າກວດສອບ"/>

ວັນທີ 09/08/2021

ລາຍເຊັນຜູ້ລາຍງານ.....

ຮູບທີ 9 : ຮູບອອກແບບສະແດງລາຍງານການຈອງປີ້

3.4.2 ການອອກແບບຮ່າງປ້ອນຂໍ້ມູນ (Input Design)

1) ຟອມເຂົ້າສູ່ລະບົບແອດມິນ

ຢືນດີຕ້ອນຮັບເຂົ້າສູ່ລະບົບຈັດການຂໍ້ມູນ! 🖐️

1

Username

demo@gogo.com

Password

.....

2

ເຂົ້າສູ່ລະບົບ

The image shows a login form for an admin system. It features a title in Lao, a username field with the text 'demo@gogo.com', a password field with masked characters and a toggle icon, and a blue login button. Two callout boxes are present: box 1 points to the title area, and box 2 points to the login button.

ຮູບທີ 10 : ຮູບແບບຟອມເຂົ້າສູ່ລະບົບແອດມິນ

1. ພາກສ່ວນປ້ອນຊື່ຜູ້ໃຊ້ ແລະ ລະຫັດຜ່ານ
2. ເຂົ້າສູ່ລະບົບ

2) ຟອມການຈັດການຂໍ້ມູນບໍລິສັດ

The screenshot shows a web interface for managing companies. At the top right, there is a green button with a plus sign and the text '+ ເພີ່ມ' (Add), labeled with callout 1. Below this is a table with the title 'ຈັດການຂໍ້ມູນບໍລິສັດ' (Manage Company Information). The table has columns for '#', 'ບໍລິສັດ' (Company), 'ທີ່ຢູ່' (Address), 'ອີເມວ' (Email), 'ເບີໂທ' (Phone), and 'ACTIONS'. There are three rows of data. The 'ACTIONS' column contains edit (pencil icon) and delete (trash icon) buttons for each row. Callout 2 points to the edit button of the first row, and callout 3 points to the delete button of the third row.

#	ບໍລິສັດ	ທີ່ຢູ່	ອີເມວ	ເບີໂທ	ACTIONS
1	ບໍລິສັດກຽງໄກ	ນະຄອນຫຼວງ	company@gmail.com	030 9873845	
2	ບໍລິສັດຈິດປະສົງ	ມ ໄຊທາ	company@gmail.com	030 9873845	
3	ແກ້ໄຂທິດ	ຄຳຮຸກເກີກ	phonekham.dev@gmail.com	030 78073677	

ຮູບທີ 11 : ຮູບແບບຟອມການຈັດການຂໍ້ມູນບໍລິສັດ

1. ປຸ່ມເພີ່ມຂໍ້ມູນ
2. ປຸ່ມແກ້ໄຂ
3. ປຸ່ມລຶບ

3) ຟອມການເພີ່ມຂໍ້ມູນບໍລິສັດ

The image shows a web form titled "ເພີ່ມຂໍ້ມູນບໍລິສັດ" (Add Company Information). The form contains four input fields: "ບໍລິສັດ" (Company), "ທີ່ຢູ່" (Address), "ອີເມວ" (Email), and "ເບີໂທ" (Phone). Below the form are two buttons: "ບັນທຶກ" (Save) and "ຍົກເລີກ" (Cancel). Three numbered callouts are present: 1 points to the form area, 2 points to the "ບັນທຶກ" button, and 3 points to the "ຍົກເລີກ" button.

ຮູບທີ 12 : ຮູບແບບຟອມການເພີ່ມຂໍ້ມູນບໍລິສັດ

1. ພາກສ່ວນປ້ອນຂໍ້ມູນ.
2. ບັນທຶກ.
3. ຍົກເລີກ.

4) ຟອມການຈັດການຂໍ້ມູນສາຍທາງ

ຈັດການຂໍ້ມູນສາຍທາງ

1

2

ຄົ້ນຫາ

ສາຍທາງ	ສາຍທາງພາສາອັງກິດ	ໄລຍະທາງ KM	ຕົ້ນທາງ	ປາຍທາງ	ຈັດການສະຖານະ	ຈັດການ
ວຽງຈັນ - ຫຼັກ 20	VTE - LAK 20	110	ວຽງຈັນ	ຫຼັກ 20	ຈັດການສະຖານະ	✓ ✕
ວຽງຈັນ - ປາກເຊ	VTE-PAKSE	120	ວຽງຈັນ	ປາກເຊ	ຈັດການສະຖານະ	✓ ✕

Rows per page: 10 1-2 of

3

4

ຮູບທີ 13 : ຮູບແບບຟອມການຈັດການຂໍ້ມູນສາຍທາງ

1. ພາກສ່ວນປ້ອນຂໍ້ມູນ
2. ບັນທຶກ
3. ຍົກເລີກ
4. ປຸ່ມເພີ່ມຂໍ້ມູນ
5. ຊ່ອງຄົ້ນຫາ
6. ການແກ້ໄຂ
7. ການລຶບ

5) ຟອມການເພີ່ມຂໍ້ມູນສາຍທາງ

The image shows a web form titled "ເພີ່ມຂໍ້ມູນສາຍທາງໃໝ່" (Add New Route). The form contains four input fields: "ປາຍທາງ" (Destination), "ໄລຍະທາງ" (Distance), "ສາຍທາງພາສາອັງກິດ" (English Route Name), and "ສາຍທາງ" (Route Name). Below the "ສາຍທາງ" field is a dropdown menu labeled "ວຽງຈັນ -". At the bottom of the form are two buttons: "ບັນທຶກ" (Save) and "ຍົກເລີກ" (Cancel). Numbered callouts are present: '1' points to the top of the form, '2' points to the "ບັນທຶກ" button, and '3' points to the "ຍົກເລີກ" button.

ຮູບທີ 14 : ຮູບແບບຟອມການເພີ່ມຂໍ້ມູນສາຍທາງ

1. ພາກສ່ວນການປ້ອນຂໍ້ມູນ
2. ບັນທຶກ
3. ຍົກເລີກ

6) ຟອມການຈັດການຖ້ວງລົດ

The screenshot shows a web application for vehicle management. At the top, it says 'ຈັດການຂໍ້ມູນທຳຮວງລົດປະຈຳສາຍ' (Manage vehicle registration information by lane). Below this, it specifies 'ວຽງຈັນ - ຫຼັກ 20' (Vientiane - Main 20) and 'VTE - LAK 20'. A green button with a plus sign and the text '+ ເພີ່ມຖ້ວງໃໝ່' (+ Add new vehicle) is on the left. A table with 6 columns is shown: '#', 'ເວລາອອກ' (Exit time), 'ຄ່າໂດຍສານ' (Fare), 'ປະເພດລົດ' (Vehicle type), 'ສາມາດດອງຝ່າຍເວັບ' (Can use website), and 'ຈັດການ' (Manage). The first row of data shows a vehicle with ID 1, exit time 8:30, fare 100000, vehicle type 'ທຳມະດາ' (General), and a red 'X' in the 'Can use website' column. The 'Manage' column for this row contains two icons: a green checkmark and a red trash can. Three callout boxes are present: Callout 1 points to the '+ ເພີ່ມຖ້ວງໃໝ່' button; Callout 2 points to the 'ຈັດການ' (Manage) column header; Callout 3 points to the two icons in the 'Manage' column of the first row.

#	ເວລາອອກ	ຄ່າໂດຍສານ	ປະເພດລົດ	ສາມາດດອງຝ່າຍເວັບ	ຈັດການ
1	8:30	100000	ທຳມະດາ	✗	✓ ✖

ຮູບທີ 15 : ຮູບແບບຟອມການຈັດການຖ້ວງລົດ

1. ການເພີ່ມຖ້ວງໃໝ່
2. ການແກ້ໄຂ
3. ການລຶບ

7) ຟອມການເພີ່ມຂໍ້ມູນຖ້ວນລົດ

ເພີ່ມຂໍ້ມູນທໍຽວໂດຍສານ

ເວລາອອກ

ຄ່າໂດຍສານ

ປະເພດລົດ

ຈອງຜ່ານເວັບ

ເລືອກປະເພດລົດ

ບັນທຶກ

ຮູບທີ 16 : ຮູບແບບຟອມການເພີ່ມຂໍ້ມູນຖ້ວນລົດ

1. ພາກສ່ວນປ້ອນຂໍ້ມູນຖ້ວນໂດຍສານ
2. ເລືອກປະເພດລົດໂດຍສານ
3. ເລືອກຊ່ອງທາງການສັ່ງຈອງ
4. ບັນທຶກການເພີ່ມຂໍ້ມູນຖ້ວນລົດ

8) ຟອມການຈັດການປະເພດລົດ

The screenshot shows a web interface for managing vehicle categories. At the top, there is a title 'ຈັດການຂໍ້ມູນປະເພດລົດ' (Manage Vehicle Category Information). Below the title is a table with the following columns: '#', 'ປະເພດ' (Category), 'ລາຍບະຫຼຽດ' (Description), 'ປະເພດປ່ອນນັ່ງ' (Vehicle Type), 'ຈຳນວນປ່ອນນັ່ງ' (Quantity), 'ຈັດການປ່ອນນັ່ງ' (Manage Vehicle Type), and 'ຈັດການ' (Manage). The table contains four rows of data. Callout 1 points to a '+ ເພີ່ມ' (Add) button. Callout 2 points to the 'ຈັດການປ່ອນນັ່ງ' column header and the corresponding buttons in the first row. Callout 3 points to the 'ຈັດການ' column header and the corresponding buttons in the first row. Callout 4 points to the delete icon in the first row.

#	ປະເພດ	ລາຍບະຫຼຽດ	ປະເພດປ່ອນນັ່ງ	ຈຳນວນປ່ອນນັ່ງ	ຈັດການປ່ອນນັ່ງ	ຈັດການ
1	ທຳມະດາ	ລົດທຳມະດາ 45 ປ່ອນນັ່ງ	ຕຽງນັ່ງຊັ້ນຕຽງ	45	ຈັດການ	✎ 🗑
2	VIP	ລົດ VIP 45 ປ່ອນນອນ	ຕຽງນອນສອງຊັ້ນ	45	ຈັດການ	✎ 🗑
3	ລົດດ່ວນ2	ລົດຕຽງນັ່ງ ທຳມະດາ 45 ປ່ອນນັ່ງ	ຕຽງນອນສອງຊັ້ນ	45	ຈັດການ	✎ 🗑
4	VIP 2	ລົດຕຽງນອນ 45 ປ່ອນນັ່ງ	ຕຽງນອນສອງຊັ້ນ	45	ຈັດການ	✎ 🗑

ຮູບທີ 17 : ຮູບແບບຟອມການຈັດການປະເພດລົດ

1. ການເພີ່ມ
2. ການຈັດການປະເພດປ່ອນນັ່ງ
3. ການແກ້ໄຂ
4. ການລຶບ

9) ຟອມການເພີ່ມປະເພດລົດ

The screenshot shows a web form titled "ຟອມເພີ່ມປະເພດລົດ" (Vehicle Type Addition Form). It contains several input fields and buttons. Numbered callouts are as follows:

- 1:** Points to the top right corner of the form area.
- 2:** Points to the "ປະເພດປ່ອນນັ່ງ" (Vehicle Type) section, which includes radio buttons for "ຕັ້ງຊັ້ນດຽວ" (Single Class) and "ຕຽງສອງຊັ້ນ" (Two Classes).
- 3:** Points to the "ບັນທຶກ" (Save) button.
- 4:** Points to the "ຍົກເລີກ" (Delete) button.

Other visible fields include "ປະເພດ" (Type), "ລາຍລະອຽດ" (Details), and "ຈຳນວນປ່ອນນັ່ງ" (Number of Vehicles).

ຮູບທີ 18 : ຮູບແບບຟອມການເພີ່ມປະເພດລົດ

1. ພາກສ່ວນປ້ອນຂໍ້ມູນການເພີ່ມປະເພດລົດ
2. ເລືອກປະເພດປ່ອນນັ່ງ
3. ບັນທຶກ
4. ຍົກເລີກ

3.5 ການອອກແບບຖານຂໍ້ມູນ

3.5.1 ຕາຕະລາງພະນັກງານ (Employee)

Table: Employee					
Field Name	Data Type	Allow Null	Key	Description	Reference
id	ObjectId	No	PK	ລະຫັດພະນັກງານ	
firstname	String	No		ຊື່ພະນັກງານ	
lastname	String	No		ນາມສະກຸນ	
username	String	No		ຊື່ຜູ້ໃຊ້ພະນັກງານ	
password	String	No		ລະຫັດຜ່ານ	
email	String	No		ອີເມວ	
address	String	No		ທີ່ຢູ່	
phone	String	No		ເບີໂທ	
role	String	No		ໜ້າທີ່	
status	String	No		ສະຖານະ	

ຕາຕະລາງທີ 6 :ຕາຕະລາງພະນັກງານ (Employee)

3.5.2 ຕາຕະລາງຂໍ້ມູນປີ້ (Ticket)

Table: Ticket					
Field Name	Data Type	Allow Null	Key	Description	Reference
id	ObjectId	No	PK	ລະຫັດປີ້	
ticketNo	Number	No		ເລກທີປີ້	

firstname	String	No		ຊື່	
lastname	String	No		ນາມສະກຸນ	
departureTime	ObjectId	No	FK	ເວລາລົດອອກ	DepartureTime
bus	ObjectId	No		ລະຫັດລົດ	
Seat	ObjectId	No		ລະຫັດບ່ອນນັ່ງ	Seat
price	Number	No		ລາຄາປີ້	
employee	ObjectId	No	FK	ລະຫັດຜູ້ອອກປີ້	Employee

ຕາຕະລາງທີ 7 :ຕາຕະລາງຂໍ້ມູນປີ້ (Ticket)

3.5.3 ຕາຕະລາງຂໍ້ມູນບໍລິສັດ (Company)

Table: Company					
Field Name	Data Type	Allow Null	Key	Description	Reference
id	ObjectId	No	PK	ລະຫັດລູກຄ້າ	
name	String	No		ຊື່	
address	String	No		ທີ່ຢູ່	
tel	String	No		ເບີໂທ	
email	String	No		ອີເມວ	

ຕາຕະລາງທີ 8 : ຕາຕະລາງຂໍ້ມູນບໍລິສັດ (Company)

3.5.4 ຕາຕະລາງຂໍ້ມູນປະເພດລົດ (BusType)

Table: Bus Type					
Field Name	Data Type	Allow Null	Key	Description	Reference

id	ObjectId	No	PK	ລະຫັດລູກຄ້າ	
type	String	No		ປະເພດ	
description	String	No		ລາຍລະອຽດ	
capacity	Number	No		ຈຳນວນບ່ອນນັ່ງ	
floorType	Number	No		ປະເພດບ່ອນນັ່ງ	

ຕາຕະລາງທີ 9 : ຕາຕະລາງຂໍ້ມູນປະເພດລົດ (BusType)

3.5.5 ຕາຕະລາງຂໍ້ມູນລົດ (Bus)

Table: Bus					
Field Name	Data Type	Allow Null	Key	Description	Reference
id	ObjectId	No	PK	ລະຫັດລົດ	
licencePlate	String	No		ທະບຽນລົດ	
busType	ObjectId	No	FK	ລະຫັດປະເພດລົດ	BusType
company	ObjectId	No	FK	ລະຫັດບໍລິສັດ	Company

ຕາຕະລາງທີ 10 : ຕາຕະລາງຂໍ້ມູນລົດ (Bus)

3.5.6 ຕາຕະລາງຂໍ້ມູນບ່ອນນັ່ງ (Seat)

Table: Seat					
Field Name	Data Type	Allow Null	Key	Description	Reference
id	ObjectId	No	PK	ລະຫັດບ່ອນນັ່ງ	
seatNo	String	No		ເບີບ່ອນນັ່ງ	
busType	ObjectId	No	FK	ລະຫັດປະເພດລົດ	BusType

floor	Number	No		ຊັ້ນບ່ອນນັ່ງ	
-------	--------	----	--	--------------	--

ຕາຕະລາງທີ 11 :ຕາຕະລາງຂໍ້ມູນບ່ອນນັ່ງ (Seat)

3.5.7 ຕາຕະລາງຂໍ້ມູນສະມາຊິກ (Member)

Table: Member					
Field Name	Data Type	Allow Null	Key	Description	Reference
id	ObjectId	No	PK	ລະຫັດສະມາຊິກ	
firstname	String	No		ຊື່	
lastname	String	No		ນາມສະກຸນ	
phone	String	No		ເບີໂທ	
username	String	No		ຊື່ຜູ້ໃຊ້	
password	String	No		ລະຫັດຜ່ານ	
email	String	No		ອີເມວ	

ຕາຕະລາງທີ 12 : ຕາຕະລາງຂໍ້ມູນສະມາຊິກ (Member)

3.5.8 ຕາຕະລາງຂໍ້ມູນການຈອງ (Booking)

Table: Booking					
Field Name	Data Type	Allow Null	Key	Description	Reference
id	ObjectId	No	PK	ລະຫັດການຈອງ	
bookingItem	ObjectId	No		ລາຍລະອຽດການຈອງ	
qty	Number	No		ຈຳນວນປີ້	

member	ObjectId	No	FK	ລະຫັດສະມາຊິກ	Member
fullname	String	No		ຊື່ເຕັມ	
tel	String	No		ເບີໂທ	
email	String	No		ອີເມວ	
totalAmount	Number	No		ເປັນເງິນ	

ຕາຕະລາງທີ 13 : ຕາຕະລາງຂໍ້ມູນການຈອງ (Booking)

3.5.9 ຕາຕະລາງຂໍ້ມູນລາຍລະອຽດການຈອງ (BookingItem)

Table: BookingItem					
Field Name	Data Type	Allow Null	Key	Description	Reference
id	ObjectId	No	PK	ລະຫັດລາຍລະອຽດການຈອງ	
seat []	ObjectId	No	FK	ລະຫັດບ່ອນນັ່ງ	Seat
departureTime	ObjectId	No	FK	ເວລາລົດອອກ	DepartureTime
departureDate	Date	No		ວັນທີເດີນທາງ	

ຕາຕະລາງທີ 14 : ຕາຕະລາງຂໍ້ມູນລາຍລະອຽດການຈອງ (BookingItem)

3.5.10 ຕາຕະລາງຂໍ້ມູນສາຍທາງ (Route)

Table: Route					
Field Name	Data Type	Allow Null	Key	Description	Reference
id	ObjectId	No	PK	ລະຫັດສາຍທາງ	
routeName	String	No		ຊື່ສາຍທາງ	

routeEngName	String	No		ຊື່ສາຍທາງອັງກິດ	
distance	Number	No		ໄລຍະທາງ	
departure	String	No		ຕົ້ນທາງ	
destination	String	No		ປາຍທາງ	

ຕາຕະລາງທີ 15 : ຕາຕະລາງຂໍ້ມູນສາຍທາງ (Route)

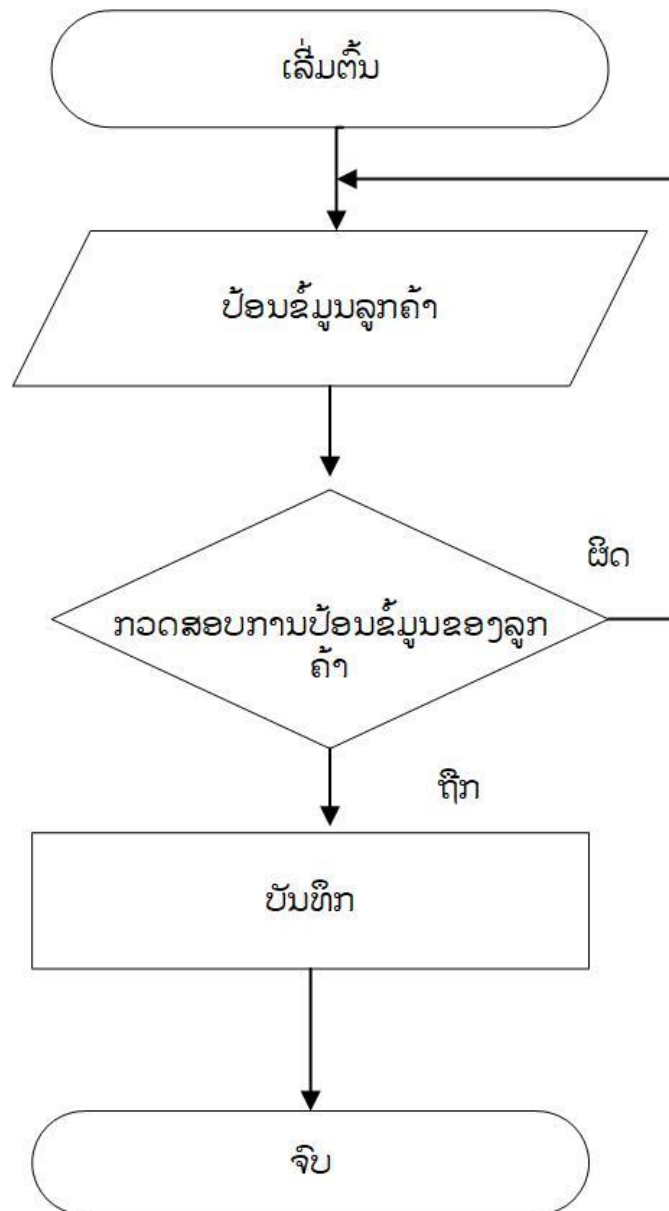
3.5.11 ຕາຕະລາງຂໍ້ມູນເວລາລົດອອກ (DepartureTime)

Table: DepartureTime					
Field Name	Data Type	Allow Null	Key	Description	Reference
id	ObjectId	No	PK	ລະຫັດເວລາລົດອອກ	
time	String	No		ເວລາ	
busType	ObjectId	No	FK	ລະຫັດປະເພດລົດ	BusType
fare	Number	No		ຄ່າໂດຍສານ	
route	ObjectId	No	FK	ລະຫັດສາຍທາງ	Route
isBookable	Boolean	No		ສາມາດຈອງຜ່ານເວັບ	

ຕາຕະລາງທີ 16 : ຕາຕະລາງຂໍ້ມູນເວລາລົດອອກ (DepartureTime)

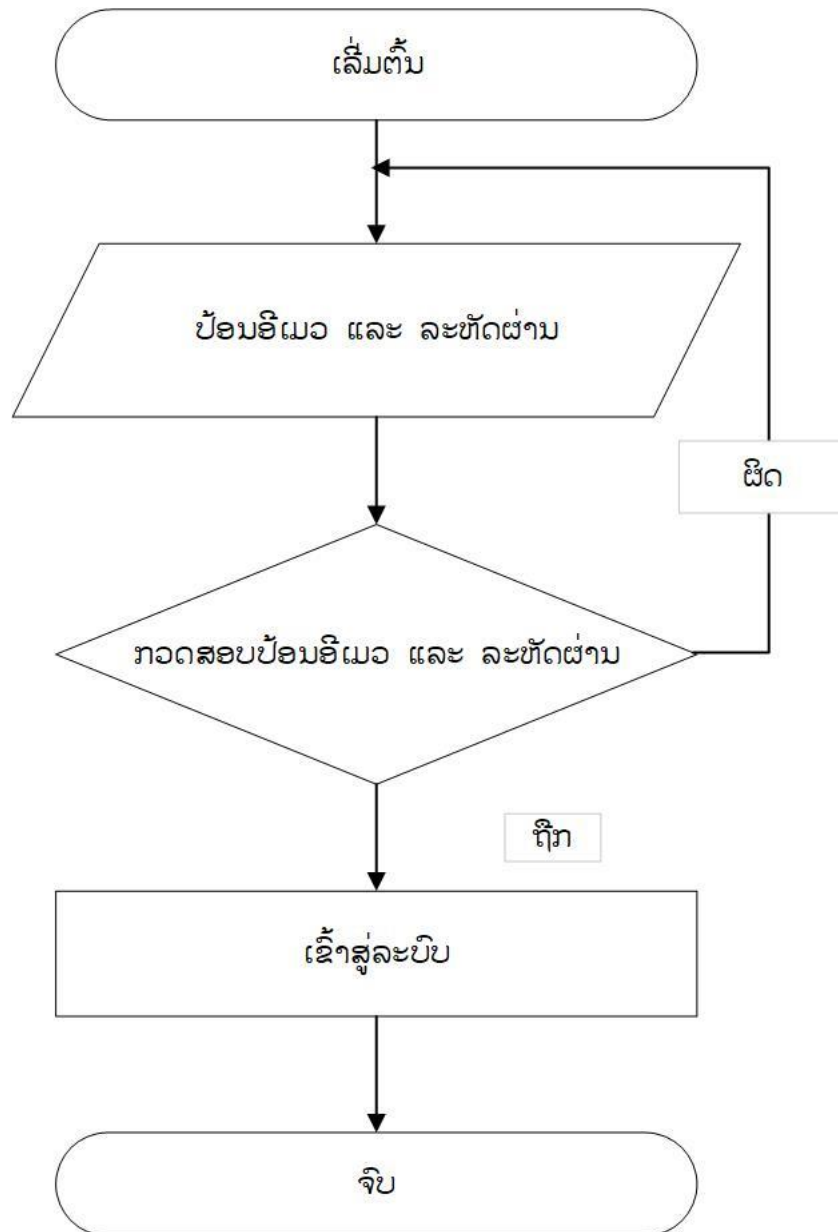
3.6 ແຜນວາດຂັ້ນຕອນການເຮັດວຽກ (Flowchart)

3.6.1 ແຜນວາດ Flowchart ການສະໝັກສະມາຊິກ



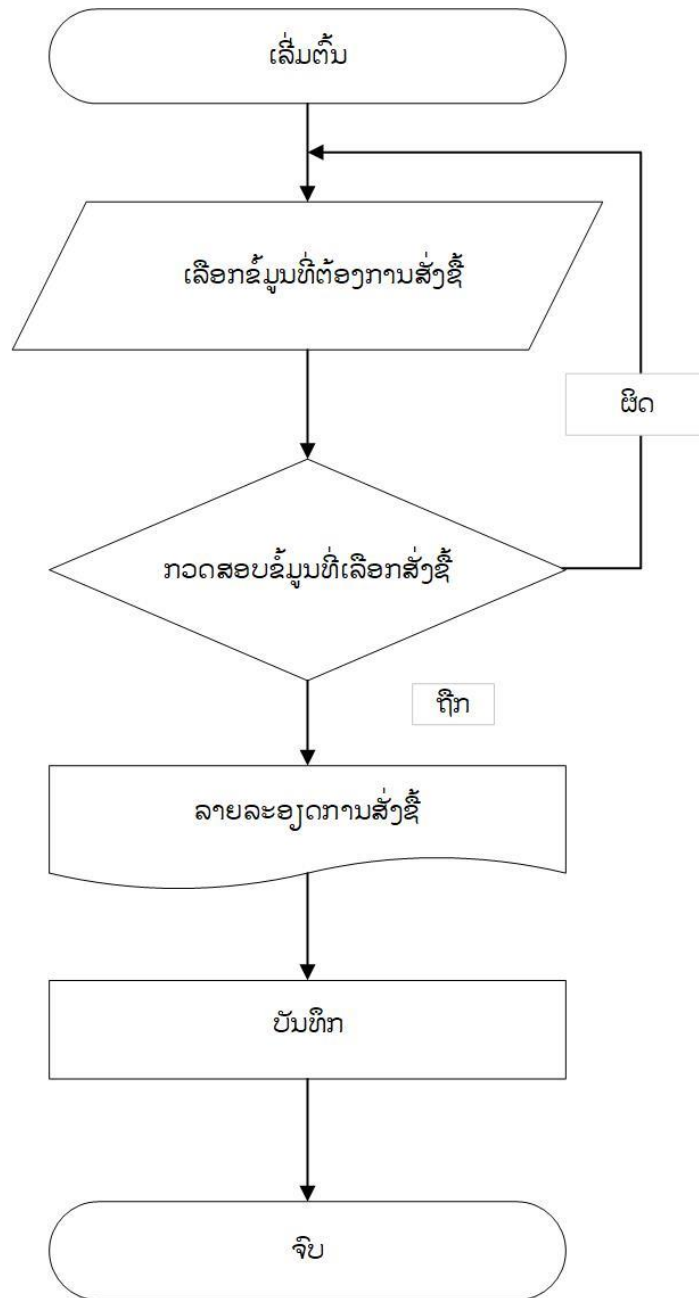
ແຜນວາດທີ 14 : ແຜນວາດ Flowchart ການສະໝັກສະມາຊິກ

3.6.2 ແຜນວາດ Flowchart ການເຂົ້າສູ່ລະບົບ



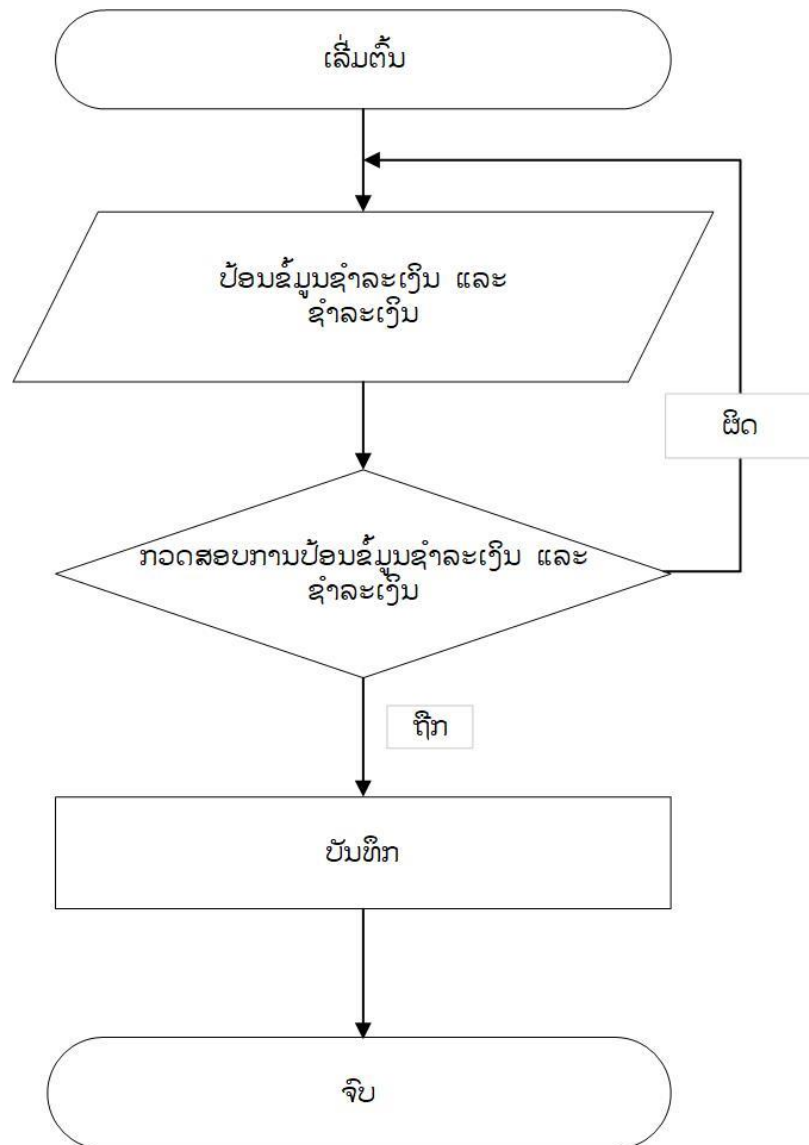
ແຜນວາດທີ 15 : ແຜນວາດ Flowchart ການເຂົ້າສູ່ລະບົບ

3.6.3 ແຜນວາດ Flowchart ການສັ່ງຊື້



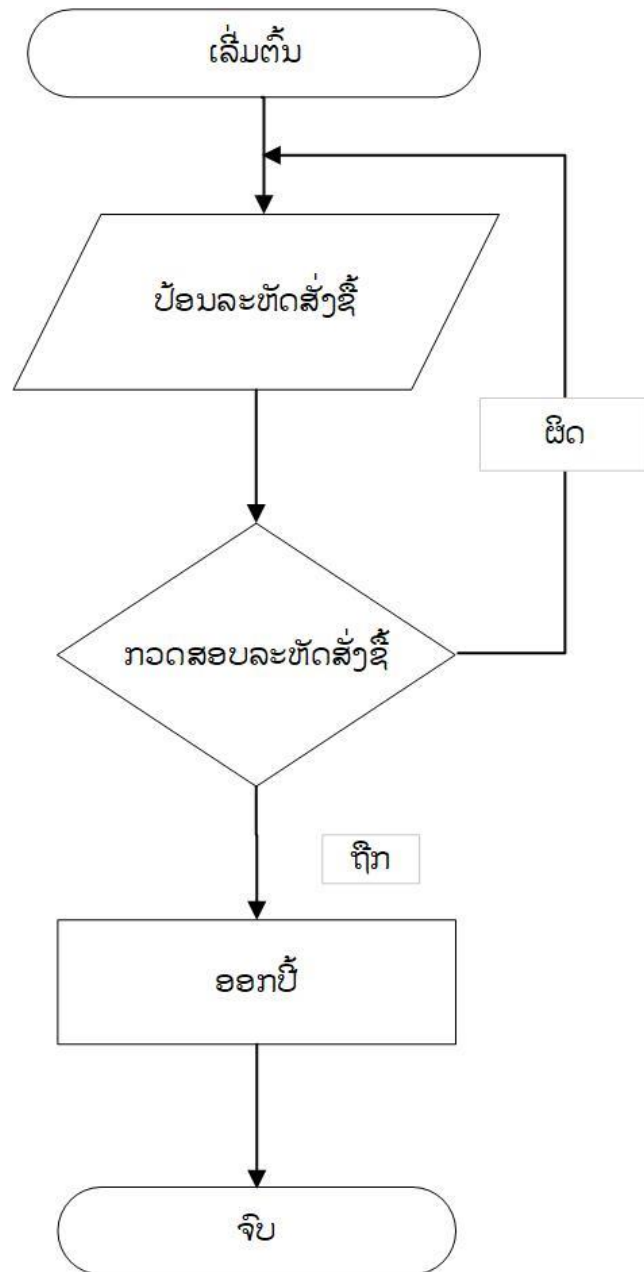
ແຜນວາດທີ 16 : ແຜນວາດ Flowchart ການສັ່ງຊື້

3.6.4 ແຜນວາດ Flowchart ການຊຳລະເງິນ



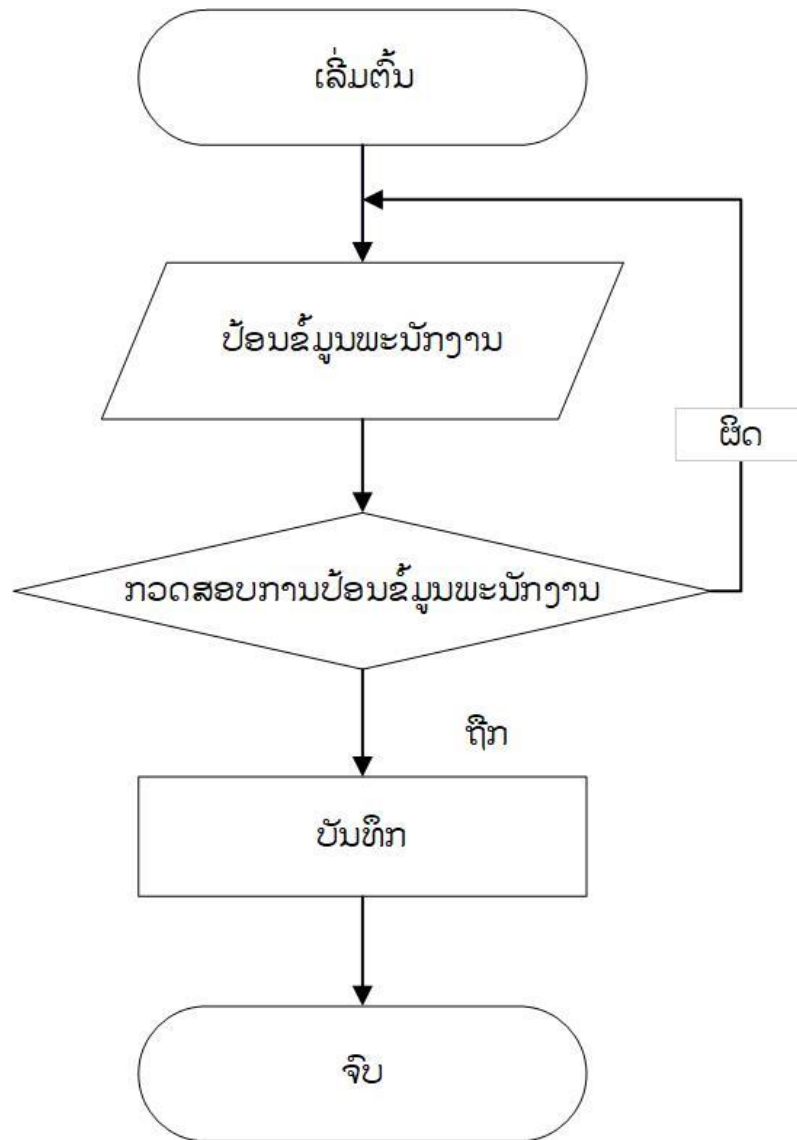
ແຜນວາດທີ 17 : ແຜນວາດ Flowchart ການຊຳລະເງິນ

3.6.5 ແຜນວາດ Flowchart ການອອກປີ້



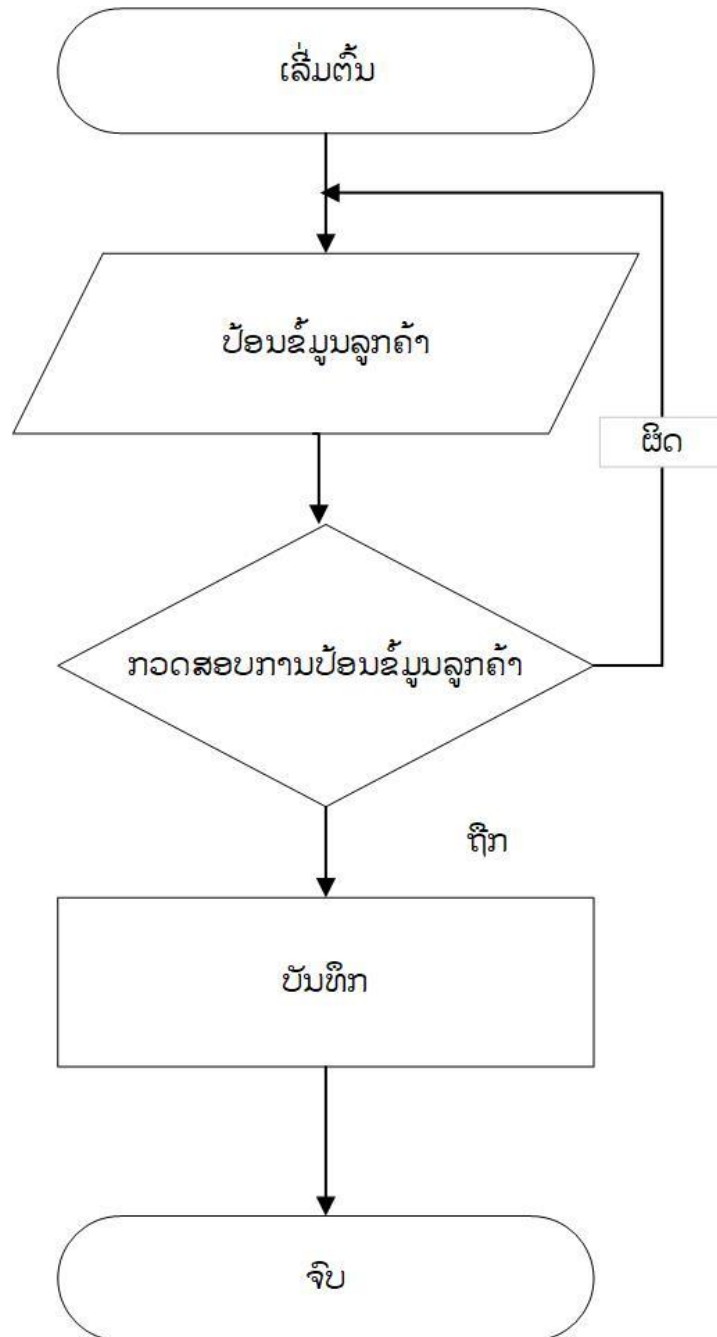
ແຜນວາດທີ 18 : ແຜນວາດ Flowchart ການອອກປີ້

3.6.6 ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນພະນັກງານ



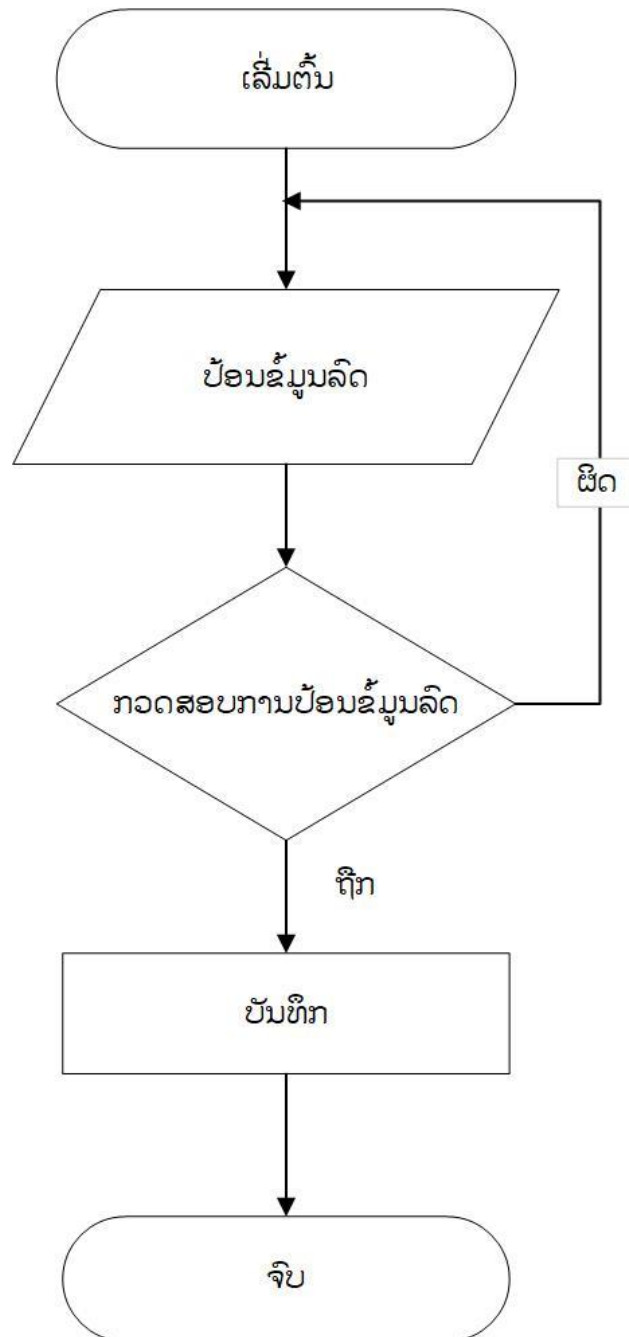
ແຜນວາດທີ 19 : ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນພະນັກງານ

3.6.7 ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນລູກຄ້າ



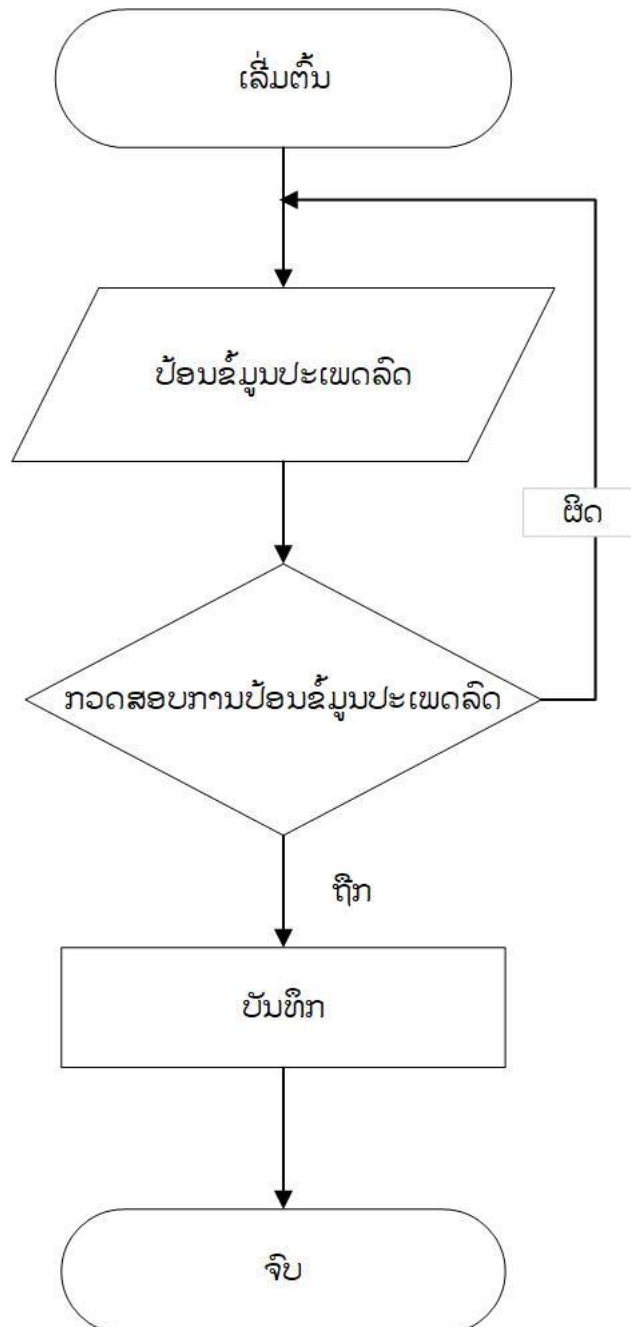
ແຜນວາດທີ 20 : ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນລູກຄ້າ

3.6.8 ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນລົດ



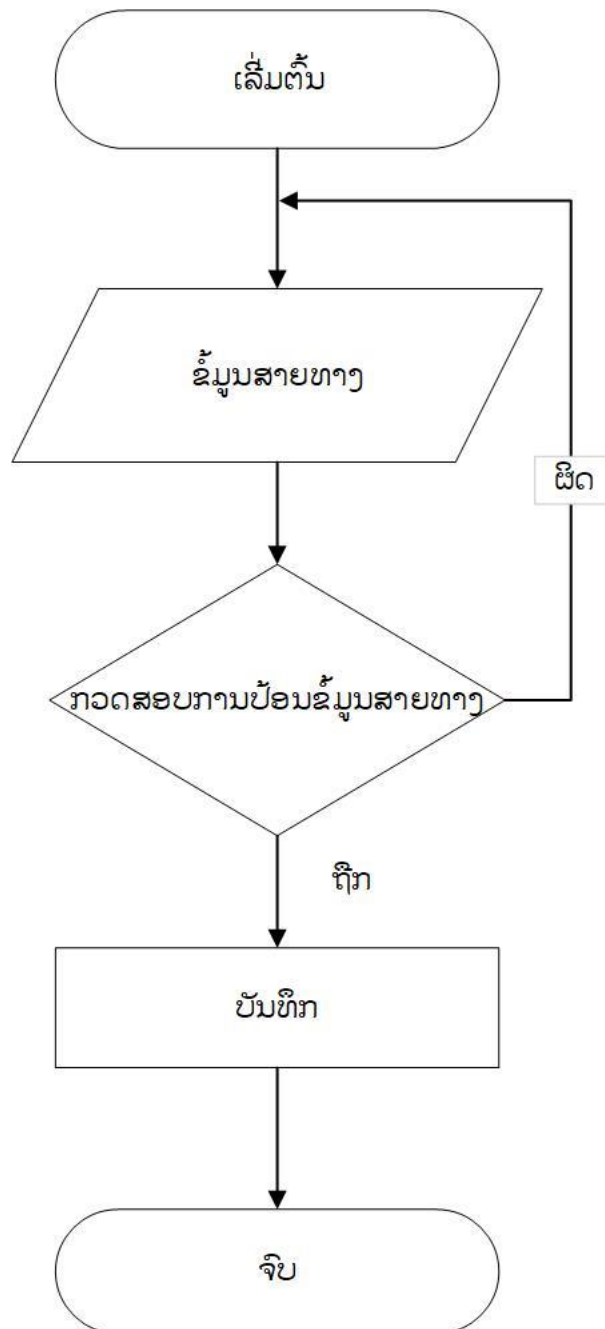
ແຜນວາດທີ 21 : ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນລົດ

3.6.9 ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນປະເພດລົດ



ແຜນວາດທີ 22 : ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນປະເພດລົດ

3.5.10 ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນສາຍທາງ



ແຜນວາດທີ 23 : ແຜນວາດ Flowchart ຈັດການຂໍ້ມູນສາຍທາງ

ບົດທີ 4

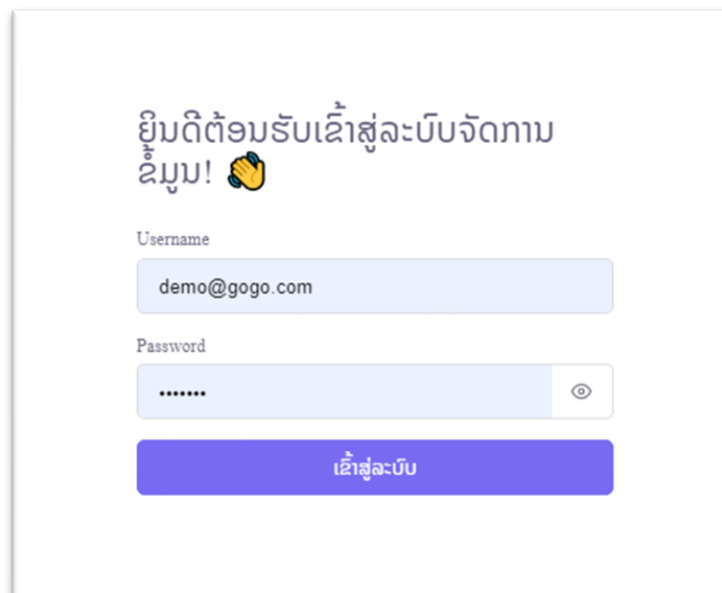
ຜົນຂອງການວິເຄາະຂໍ້ມູນ ແລະ ການອະທິບາຍຜົນ

4.1 ໜ້າຕ່າງການ (Login Form)

ເມື່ອເປີດໂປຣແກຣມຂຶ້ນມາກໍຈະເຫັນວ່າເຂົ້າສູ່ລະບົບເປັນໜ້າທຳອິດຈາກນັ້ນຈຶ່ງທຳການເຂົ້າສູ່ລະບົບດ້ວຍຊື່ ແລະ ລະຫັດເຂົ້າໃຊ້ເພື່ອເຂົ້າໄປດ້ານໃນຂອງໂປຣແກຣມດັ່ງຮູບລຸ່ມນີ້:

- ວິທີການເຂົ້າສູ່ລະບົບມີຄື :

- 1) ໃສ່ຊື່ຜູ້ໃຊ້ລະບົບ.
- 2) ໃສ່ລະຫັດຜ່ານ.
- 3) ກົດປຸ່ມເພື່ອເຂົ້າລະບົບ.
- 4) ຖ້າຊື່ກັບລະຫັດຖືກຈະເປີດນຳໃຊ້ໂປຣແກຣມໄດ້ປົກກະຕິແລ້ວ.
- 5) ຖ້າຊື່ກັບລະຫັດຜິດຈະມີຂໍ້ຄວາມຂຶ້ນມາວ່າ: ກະລຸນາກວດສອບຂໍ້ມູນຂອງທ່ານໃຫ້ຖືກຕ້ອງ ແລ້ວກົດປຸ່ມ OK.



The image shows a login form with the following elements:

- Header text: ຍິນດີຕ້ອນຮັບເຂົ້າສູ່ລະບົບຈັດການຂໍ້ມູນ! 🖐️
- Username field: Labeled "Username" with the text "demo@gogo.com" entered.
- Password field: Labeled "Password" with masked characters "*****" and a toggle icon.
- Login button: A blue button with the text "ເຂົ້າສູ່ລະບົບ".

ຮູບທີ 19 : ຮູບໜ້າຕ່າງການ (Login Form)

4.2 ຟອມຫຼັກ (Main Form)

ຟອມຫຼັກແມ່ນຟອມທີ່ເຮົາສາມາດເຂົ້າຫາທຸກໆຟອມໄດ້ເຊັ່ນ: ຟອມຈັດການຂໍ້ມູນ, ຟອມປ້ອນຂໍ້ມູນ, ຟອມການບໍລິການ, ແລະ ຟອມລາຍງານ.

4.2.1 ຟອມຄົ້ນຫາຖ້ວລົດ

The screenshot shows a web application interface for vehicle search. It features a top navigation bar (1) with links in Lao: ໜ້າຫຼັກ (Home), ກ່ຽວກັບພວກເຮົາ (About Us), ຕິດຕໍ່ (Contact), and a settings icon. The main content area (2) is titled 'ຄົ້ນຫາຖ້ວລົດ' (Vehicle Search) and includes a search form with a dropdown menu for 'ປາຍທາງ' (Destination) and a date input field for 'ວັນທີ' (Date) in mm/dd/yyyy format. A blue button labeled 'ຄົ້ນຫາລົດ' (Search Vehicle) is positioned below the form. To the right of the form is a large image (3) showing a bus stop with a shelter and a bus. The interface is designed with a yellow and white color scheme.

ຮູບທີ 20 : ຮູບແບບຟອມຄົ້ນຫາຖ້ວລົດ

1. ພາກສ່ວນຂໍ້ມູນຂອງຖ້ວລົດ.
2. ພາກສ່ວນປ້ອນຂໍ້ມູນຖ້ວລົດ.
3. ຄົ້ນຫາຖ້ວລົດ.

4.2.2 ຟອມເລືອກບ່ອນນັ່ງ

The screenshot shows a web-based seating arrangement system. At the top, there are fields for 'ເວລາລົດອອກ' (7:00), 'ປະເພດລົດ' (VIP), 'ຄ່າໂດຍສານ' (150000), and a 'ເບິ່ງລາຍລະອຽດ' button. Below this is a legend (1) with three icons: a grey car icon for 'ຍັງຫວ່າງ' (Still empty), a red car icon for 'ຖືກຈອງແລ້ວ' (Already booked), and a green car icon for 'ກຳລັງເລືອກ' (Being selected). To the right is a callout (2) pointing to the top right area. The main area shows two car layouts: 'ຊັບ 1' (Car 1) and 'ຊັບ 2' (Car 2). Car 1 has 7 seats, with the bottom right seat highlighted in green. Car 2 has 6 seats, with the bottom left seat highlighted in green. To the right of the car layouts is a table titled 'ລາຍລະອຽດບ່ອນນັ່ງ' (Seating Details):

ເບິ່ງ	ຊັບ
B5	2
A4	1

Below the table is a callout (3) pointing to a 'ດຳເນີນການຕໍ່' (Proceed) button. At the bottom, there are three identical rows for other departure times: 15:00, 14:00, and 13:00, each with a 'ລາຍລະອຽດ' (Details) button.

ຮູບທີ 21 : ຮູບແບບຟອມຈັດການຂໍ້ມູນບ່ອນນັ່ງ

1. ພາກສ່ວນຄຳອະທິບາຍຂໍ້ມູນບ່ອນນັ່ງ.
2. ລາຍລະອຽດບ່ອນນັ່ງທີ່ກຳລັງເລືອກ.
3. ດຳເນີນການຕໍ່.

4.2.3 ຟອມປ້ອນລາຍລະອຽດຜູ້ໂດຍສານ

1

ໜ້າຫຼັກ ກ່ຽວກັບພວກເຮົາ ຕິດຕໍ່

ລາຍລະອຽດຜູ້ໂດຍສານ

ຊື່ເດີມ: phone keo

ເບີໂທ: 2028022677

ອີເມວ: phone.madridista@gmail.com

ລາຍລະອຽດການຂົນສົ່ງ

ສາຍທາງ: ວຽງຈັນ - ປາກເຊ

ເວລາອອກເດີນທາງ: 7:00

ປະເພດລົດ: VIP

ເບີປ່ອນນັ່ງ: B5,A4,

ລາຄາຕີ້: 150,000 ກີບ

ເບີປ່ອນນັ່ງ: B5,A4,

ລາຄາຕີ້: 150,000 ກີບ

2

ຮູບທີ 22 : ຮູບແບບຟອມປ້ອນລາຍລະອຽດຜູ້ໂດຍສານ

1. ພາກສ່ວນປ້ອນຂໍ້ມູນຜູ້ໂດຍສານ.
2. ລາຍລະອຽດການຂົນສົ່ງ.

4.2.4 ຟອມຊຳລະເງິນ

ລາຍລະອຽດການຊຳລະເງິນ

ຊຳລະດ້ວຍບໍລິການທະນາຄານ

ຈ່າຍຕອນນີ້ ຈ່າຍພາຍຫຼັງ

ຊື່ບັນຊີ: Phonekham Keomany
ເລກບັນຊີ: 6582 3658 25487 23658

ອັບໂລດສະລິບຢືນຢັນການຊຳລະເງິນ

Choose File No file chosen

ເລືອກຮູບ

ຍືນຢັນການຈອງ

ຮູບທີ 23 : ຮູບແບບຟອມຊຳລະເງິນ

1. ເລືອກວ່າຈະຊຳລະຕອນນີ້ ຫຼື ຈະຊຳລະພາຍຫຼັງ
2. ພາກສ່ວນອັບໂຫຼດສະລິບຢືນຢັນການຢຳລະ.
3. ຢືນຢັນການຈອງ.

ບົດທີ 5

ສະຫຼຸບ ແລະ ຂໍ້ສະເໜີ

5.1 ສະຫຼຸບຜົນການຄົ້ນຄວ້າ

ລະບົບຂາຍປີ້ລົດເມອອນລາຍຂອງສະຖານນີຂົນສົ່ງໂດຍສານສາຍໃຕ້ ແມ່ນຖືກພັດທະນາຂຶ້ນ ເພື່ອຊ່ວຍໃຫ້ສະຖານນີໄດ້ມີລະບົບທີ່ທັນສະໄໝ, ສະດວກສະບາຍ ແລະ ເຮັດໃຫ້ການຈັດການຂໍ້ມູນ, ການບໍລິການພາຍໃນສະຖານນີຢ່າງວ່ອງໄວ, ມີຄວາມເປັນລະບຽບ, ສະດວກໃນການລາຍງານ, ຂໍ້ມູນທີ່ມີຄວາມຖືກຕ້ອງ ແລະ ຊັດເຈນ. ຂອບເຂດຂອງການດຳເນີນງານໂປຣແກຣມຂອງພວກຂ້າພະເຈົ້າ ມີຈັດການຂໍ້ມູນ, ສະໝັກສະມາຊິກ, ການບໍລິການ ແລະ ລາຍງານ. ເຊິ່ງມັນເປັນການທົດແທນການເຮັດວຽກໃນລະບົບເກົ່າໂດຍລະບົບໃໝ່ເພື່ອຫຼຸດຜ່ອນບັນຫາການເສຍຫາຍຂອງຂໍ້ມູນ. ພວກຂ້າພະເຈົ້າສາມາດພັດທະນາໂປຣແກຣມເພື່ອຊ່ວຍໃຫ້ການເຮັດວຽກສະດວກສະບາຍຍິ່ງຂຶ້ນ ແລະ ໄດ້ອອກແບບໜ້າຟອມການປ້ອນຂໍ້ມູນຕ່າງໆ.

- ສ້າງຟອມຈັດການໄດ້ 5 ຟອມ.
- ຟອມການຄົ້ນຫາໄດ້ 5 ຟອມ.
- ພິມລາຍງານທັງໝົດໄດ້ 6 ລາຍງານ.

5.2 ຈຸດດີ

- ຮູ້ວິເຄາະຫາບັນຫາ ແລະ ສາເຫດຂອງລະບົບເກົ່າ.
- ຈະໄດ້ລະບົບຈອງປີ້ລົດເມສາຍໃຕ້ແບບອອນລາຍໃໝ່.
- ສາມາດນຳໃຊ້ເວບໄຊທີ່ສ້າງຂຶ້ນມາເຂົ້າຊ່ວຍໃນການຈອງປີ້ລົດເມແບບອອນລາຍ.
- ເວບໄຊທີ່ສ້າງຂຶ້ນມາສາມາດຈອງອອນລາຍໄດ້.
- ການເຮັດບົດລາຍງານສະດວກສະບາຍ ແລະ ວ່ອງໄວຂຶ້ນ.

5.3 ຈຸດອ່ອນ

ໂປຣແກຣມນີ້ແມ່ນຍັງບໍ່ສົມບູນ ແລະ ຄົບຖ້ວນຕາມຄວາມຕ້ອງການຂອງສະຖານນີເທື່ອ ເນື່ອງຈາກວ່າໃນການສຶກສາແມ່ນຍັງບໍ່ທັນມີປະສົບການໃນການຂຽນເວບໄຊມາກ່ອນ ແລະ ຍັງບໍ່ຄວບ ຄຸມເຖິງຄວາມຕ້ອງການຕົວຈິງຂອງຜູ້ໃຊ້ລະບົບ.

5.4 ແນວທາງໃນການພັດທະນາ ແລະ ຂະຫຍາຍຕໍ່ຂອງສະຖານນີ

ເນື່ອງຈາກວ່າເວບນີ້ເປັນເວບໄຊທີ່ສ້າງຂຶ້ນມາແລ້ວ ດັ່ງນັ້ນ ເພື່ອເປັນການກວດສອບຫາຂໍ້ພິດ ພາດ ແລະ ເພື່ອຫາຈຸດດີຈຸດອ່ອນມາທຳການປັບປຸງ, ແກ້ໄຂຈຶ່ງຄວນນຳເອົາເວບໄຊນີ້ໄປປັບປຸງຈຸດບົກ ຜ່ອງຂອງເວບໄຊໃນບາງສ່ວນເພື່ອເຮັດໃຫ້ເວບໄຊມີປະສິດທິພາບ ແລະ ເຮັດວຽກໄດ້ດີຂຶ້ນເພື່ອຈະນຳ ເອົາໄປປະກອບສ່ວນເຂົ້າໃນການເຮັດວຽກງານຕົວຈິງ.

ດັ່ງນັ້ນ ເວບໄຊນີ້ເປັນເວບໄຊໜຶ່ງທີ່ມີຄຸນສົມບັດໃນການໃຊ້ຖານຂໍ້ມູນໄດ້.

ເອກະສານອ້າງອີງ

[1] ທ້າວ ສາຍຄຳ ຊົງວິຊົງ, ທ້າວຊິມົວ ຢິງຈີ້ (2019-2020) ລະບົບຂາຍປີ້ອອນໄລຂອງສະຖານີຂົນສົ່ງໂດຍສານສາຍໃຕ້.

[2] ສົມມິດ ທຸມມະລີ ແລະ ອາມອນ ຈັນທະພາວົງ, (2012). ການວິເຄາະ ແລະ ອອກແບບລະບົບ ພິມຄັ້ງທີ 1 ນະຄອນຫຼວງວຽງຈັນ

[3] ຄວາມຮູ້ທົ່ວໄປກ່ຽວກັບ Nodejs ແຫຼ່ງທີມາ: https://expert-programming-tutor.com/tutorial/article/A06_NodeJS01_HelloWorld.php

[4] ຄວາມຮູ້ທົ່ວໄປກ່ຽວກັບ Reactjs ແຫຼ່ງທີມາ: <https://www.mindphp.com/บทเรียนออนไลน์/tutorial-reactjs/4883-react.html>

[5] ຄວາມຮູ້ທົ່ວໄປກ່ຽວກັບ GraphQL ແຫຼ່ງທີມາ: [https://medium.com/@athivvat/ทำความเข้าใจกับ-graphql-980ac00d6c17#:~:text=GraphQL%20คือ%20ภาษาสำหรับการเข้าถึงข้อมูล%20\(Query%20Language\)%20เพื่อ,ข้อมูลที่เรากำหนดไว้](https://medium.com/@athivvat/ทำความเข้าใจกับ-graphql-980ac00d6c17#:~:text=GraphQL%20คือ%20ภาษาสำหรับการเข้าถึงข้อมูล%20(Query%20Language)%20เพื่อ,ข้อมูลที่เรากำหนดไว้)

ເອກະສານຊ້ອນທ້າຍ

1. ໄລຍະປະຕິບັດງານ

ID	ບັນຍາຍ	ເລີ່ມຕົ້ນ	ສິ້ນສຸດ	ໄລຍະເວລາ	ຕ.ລ. 2020			ພ.ຈ. 2020				ອ.ຈ. 2020				ມ.ກ. 2021				ກ.ສ. 2021				ມ.ພ. 2021				ມ.ສ. 2021				ພ.ພ. 2021				ມິ.ຖ. 2021			
					11/10	18/10	25/10	1/11	8/11	15/11	22/11	29/11	6/12	13/12	20/12	27/12	3/1	10/1	17/1	24/1	31/1	7/2	14/2	21/2	28/2	7/3	14/3	21/3	28/3	4/4	11/4	18/4	25/4	2/5	9/5	16/5	23/5	30/5	6/6
1	I. ວາງແຜນ	12/10/2020	19/10/2020	6d																																			
2	1. ຈັດໝູ່ມ	12/10/2020	12/10/2020	1d																																			
3	2. ກຳນົດຫົວຂໍ້	13/10/2020	14/10/2020	2d																																			
4	3. ແຜນດຳເນີນງານ	15/10/2020	19/10/2020	3d																																			
5	II. ປຶດຕະ	20/10/2020	30/11/2020	30d																																			
6	1. ສຶກສາລະບົບເດີມ	20/10/2020	27/10/2020	6d																																			
7	2. ລວມລວມຂໍ້ມູນ	28/10/2020	30/10/2020	3d																																			
8	3. ປຶດຕະກຳຄວາມກ້ອງກາງຂອງສູ່ໃຊ້	2/11/2020	6/11/2020	5d																																			
9	4. ປຶດຕະກຳຄວາມກ້ອງກາງຂອງລະບົບ	9/11/2020	13/11/2020	5d																																			
10	5. ແຕ້ມແບບຈຳລອງ	16/11/2020	30/11/2020	11d																																			
11	III. ບູລະນາ	1/12/2020	29/1/2021	44d																																			
12	1. ອອກແບບ UX	1/12/2020	14/12/2020	10d																																			
13	2. ອອກແບບ ລະບົບສູ່ຢ່າງ	15/12/2020	30/12/2020	12d																																			
14	3. ອອກແບບຍົກເປັນຈອງ	1/1/2021	14/1/2021	10d																																			
15	4. ລວມລວມຖານຂໍ້ມູນ	15/1/2021	29/1/2021	11d																																			
16	IV. ສັດຕະນາ	1/2/2021	31/5/2021	86d																																			
17	1. ສ້າງຖານຂໍ້ມູນ	1/2/2021	2/2/2021	2d																																			
18	2. ພັດທະນາ API	3/2/2021	24/5/2021	79d																																			
19	3. ເຊື່ອມຕໍ່ API ກັບ Frontend	25/5/2021	31/5/2021	5d																																			
20	V. ທົດລອງ	1/6/2021	8/6/2021	6d																																			
21	1. ກຳນົດວິທີທົດລອງ	1/6/2021	2/6/2021	2d																																			
22	2. ທົດລອງຂໍ້ມູນຕົວຢ່າງ	3/6/2021	8/6/2021	4d																																			
23	VI. ສ້າງເອກະສານ	1/6/2021	1/7/2021	23d																																			
24	1. ສ້າງປຶ້ມບົດຈົບສິ້ນ	1/6/2021	1/7/2021	23d																																			

ປະຫວັດຜູ້ຊຽນ

ປະຫວັດຫຍໍ້ຜູ້ຊຽນບົດ



ຊື່ ແລະ ນາມສະກຸນ: ທ້າວ ມະໂນພອນ ມະໂນກຸນ
ວັນ,ເດືອນ, ປີເກີດ: 16 ເດືອນ ທັນວາ ປີ 1995
ບ້ານເກີດ: ທົ່ງຂັນຄຳ ເມືອງ ຈັນທະບູລີ ແຂວງ ນະຄອນຫຼວງວຽງຈັນ
ບ້ານຢູ່ປັດຈຸບັນ: ທົ່ງຂັນຄຳ ເມືອງ ຈັນທະບູລີ ແຂວງ ນະຄອນຫຼວງ
ວຽງຈັນ

ການສຶກສາ: ປີ 2016 ຈົບຊັ້ນສູງ ທີ່ ສະຖາບັນພັດທະນາສີມືແຮງງານລາວ - ເກົາຫຼີ

ປີ 2013 ຈົບມັດທະຍົມຕອນປາຍ: ທີ່ ມ.ສ ເຈົ້າອານຸວົງ

ປີ 2010 ຈົບມັດທະຍົມຕອນຕົ້ນ: ທີ່ ມ.ສ ເຈົ້າອານຸວົງ

ປີ 2007 ຈົບປະຖົມສົມບູນ: ທີ່ ໂຮງຮຽນປະຖົມສົມບູນອານຸ

ເບີໂທ: 020 5400 0003 , 020 2814 9278

ອີເມວ: dou_2020@hotmail.com

ລາຍເຊັນເຈົ້າຂອງປະຫວັດ

ຊື່ແຈ້ງ.....

ປະຫວັດຫຍໍ້ຜູ້ຊຽນບົດ



ຊື່ ແລະ ນາມສະກຸນ: ທ້າວ ພອນຄຳ ແກ້ວມະນີ
ວັນ,ເດືອນ,ປີເກີດ: 07 ເດືອນ ມີນາ ປີ 1995
ບ້ານເກີດ:ບ້ານ ທ້ວ, ເມືອງ: ວຽງພູຄາ, ແຂວງ: ຫຼວງນ້ຳທາ
ບ້ານຢູ່ປັດຈຸບັນ:ບ້ານ ທ້ວ, ເມືອງ: ວຽງພູຄາ, ແຂວງ: ຫຼວງນ້ຳ

ການສຶກສາ: ປີ 2016 ຈົບຊັ້ນສູງ: ທີ່ ສະຖາບັນພັດທະນາສີມືແຮງງານລາວ - ເກົາຫຼີ

ປີ 2013 ຈົບມັດທະຍົມສົມບູນ: ທີ່ ເມືອງວຽງພູຄາ

ປີ 2007 ຈົບປະຖົມສົມບູນ: ທີ່ ໂຮງຮຽນປະຖົມສົມບູນບ້ານດົງວຽງ

ເບີໂທ: 020 2802 2677

ອີເມວ: phonekham.dev@gmail.com

ລາຍເຊັນເຈົ້າຂອງປະຫວັດ

ຊື່ແຈ້ງ.....

