



ບົດໂຄງການຈົບຊັ້ນປະລິນຍາຕີ

ລະບົບຈັດການຂໍ້ມູນບົດວິທະຍານິພົນ
ປະລິນຍາໂທຂອງມະຫາວິທະຍາໄລສຸພານຸວົງ

ຂຽນໂດຍ :

ທ້າວ ຢາດາວລີ ຈິ່ງນໍ

ສາຂາວິຊາວິສະວະກໍາຄອມພິວເຕີ

ພາກວິຊາວິສະວະກໍາຄອມພິວເຕີ

ຄະນະວິສະວະກໍາສາດ

ມະຫາວິທະຍາໄລສຸພານຸວົງ

2025

ບົດໂຄງການຈົບຊັ້ນ ລະດັບປະລິນຍາຕີ

ລະບົບຈັດການຂໍ້ມູນບົດວິທະຍານິພົນ
ປະລິນຍາໂທຂອງມະຫາວິທະຍາໄລ ສຸພານຸວົງ

ຂຽນໂດຍ:

ທ້າວ ຢາດາວລີ ຈິ່ງນໍ

ອາຈານຜູ້ນຳພາ:

ປທ ຊ້າງຢ່າງ

ອາຈານຜູ້ຊ່ວຍນຳພາ:

ປທ ບຸນໂຮມ ເມກສະຫວັນ

ບົດໂຄງການຈົບຊັ້ນປະລິນຍາຕີນີ້ເປັນຜົນງານ

ການສຶກສາຕາມຫຼັກສູດປະລິນຍາຕີວິສະວະກຳສາດສາຂາວິສະວະກຳ ຄອມພິວເຕີ

ມະຫາວິທະຍາໄລ ສຸພານຸວົງ

2025

ລິຂະສິດຂອງມະຫາວິທະຍາໄລ ສຸພານຸວົງ

FINAL REPORT
MASTER'S THESIS MANAGEMENT SYSTEM FOR
SUPHANOUVONG UNIVERSITY

Written:

Mr. Yadaolee JONGNOR

Advisor:

Mr. Xang yang

Co-Advisor:

Mr. Bounhome MEKSAVUN

FINAL REPORT SUBMITTED:

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING MAJORING IN COMPUTER ENGINEERING

FACULTY OF ENGINEERING

SOUPHANOUVONG UNIVERSITY

2025

COPYRIGHT © SOUPHANOUVONG UNIVERSITY



ໃບອະນຸມັດວິສະວະກຳສາດ ສາຂາວິສະວະກຳຄອມພິວເຕີ

ຊື່ຫຼັກສູດ: ປະລິນຍາຕີວິສະວະກຳສາດ ສາຂາວິສະວະກຳຄອມພິວເຕີ

ຊື່ຫົວບົດ: ລະບົບຈັດການຂໍ້ມູນບົດວິທະຍານິພົນ ປະລິນຍາໂທຂອງມະຫາວິທະຍາໄລ ສຸພານຸວົງ

ຊື່ນັກສຶກສາ: ທ້າວ ຢາດາວລີ ຈິ່ງນໍ

ຄະນະວິສະວະກຳສາດ ມະຫາວິທະຍາໄລສຸພານຸວົງ

ມະນຸ້ມໃຫ້ໂຄງການຈົບຊັ້ນນີ້ເປັນຜົນງານການສຶກສາຕາມຫຼັກສູດປະລິນຍາຕີ

.....ອາຈານນຳຜາ

(ປທ ຊ້າງຢ່າງ)

.....ອາຈານຜູ້ຊ່ວຍນຳຜາ

(ປທ ບຸນໂຮມ ເມກສະຫວັນ)

..... ປະທານກຳມະການປ້ອງກັນບົດ

(ອຈ.ປທ. ເພັດສະດາຈານ ສາຍປະເສີດ)

.....ກຳມະການປ້ອງກັນບົດ

()

.....ກຳມະການປ້ອງກັນບົດ

()

ຄະນະບໍດີ

ບົດນຳ

ໃນປັດຈຸບັນການນຳໃຊ້ເຕັກໂນໂລຊີນັບມື້ນັບມີການພັດທະນາທີ່ທັນສະໄໝຂຶ້ນ ແລະ ມີຄວາມກ້າວໜ້າຢ່າງບໍ່ຢຸດຢັ້ງ

ນອກຈາກນັ້ນຍັງປະຕິເສດບໍ່ໄດ້ວ່າທຸກບາດກ້າວໃນການດຳລົງຊີວິດຂອງຄົນເຮົາລ້ວນແຕ່ນຳໃຊ້ເຕັກໂນໂລຊີ, ບໍ່ວ່າຈະເປັນວິສະວະກິດ, ທຸລະກິດ ແລະ

ອົງກອນຕ່າງໆລ້ວນແຕ່ຈຳເປັນຕ້ອງນຳໃຊ້ຂໍ້ມູນຂ່າວສານຈາກເຕັກໂນໂລຊີທີ່ທັນສະໄໝເຂົ້າມາຊ່ວຍຍົກລະດັບ ຄຸນນະພາບແລະຄວາມຮູ້ໃນການພັດທະນາເສີມສ້າງປະສິດທິພາບ ໃນການນຳໃຊ້ໂປຣແກຣມ ຫຼື ເວັບໄຊ ເຂົ້າມາຊ່ວຍໃນການເຮັດວຽກຕ່າງໆນັບມື້ນັບມີຄວາມຕ້ອງການຫຼາຍຂຶ້ນ.

ຈຸດປະສົງຫຼັກຂອງການ ການພັດທະນາລະບົບ ຂາຍແບບກໍ່ສ້າງເຄຫະສະຖານ ດ້ວຍຮູບແບບ Online

ເພື່ອອຳນວຍຄວາມສະດວກໃນການຈັດການການຂາຍໃຫ້ມີປະສິດທິພາບ,

ຫຼຸດຜ່ອນຄວາມຊ້ຳຊ້ອນຂອງຂໍ້ມູນ,

ເພື່ອຄວາມເປັນລະບຽບໃນການຈັດການຂໍ້ມູນຕ່າງໆຂອງບົດວິທະຍານິພົນຂອງມະຫາວິທະຍາໄລເພື່ອໃຫ້ການ ຈັດເກັບບົດວິທະຍານິພົນສະດວກຍິ່ງຂຶ້ນ.

ສຸດທ້າຍນີ້ ພວກຂ້າພະເຈົ້າຫວັງຢ່າງຍິ່ງວ່າບົດໂຄງການຈົບຊັ້ນບົດນີ້ຈະມີຄຸນປະໂຫຍດບໍ່ຫຼາຍກໍ ໜ້ອຍຕໍ່ນັກສຶກສາລຸ້ນຕໍ່ໄປ, ຜູ້ສົນໃຈໃນການສ້າງ ແລະ ພັດທະນາລະບົບ,

ພ້ອມດຽວກັນນັ້ນຫວັງວ່າລະບົບຈະຊ່ວຍອຳນວຍຄວາມສະດວກໃນການຈັດການຂໍ້ມູນຕ່າງໆຂອງທີມງານ ແລະ ພະນັກງານທີ່ຮັບຜິດຊອບອີກດ້ວຍ.

ໃນລະບົບທີ່ຂ້າພະເຈົ້າໄດ້ພັດທະນາຂຶ້ນມານີ້ປາສະຈາກບໍ່ໄດ້ຄວາມຂາດຕົກບົກຜ່ອງ

ບາງບັນຫາອາດຈະບໍ່ຈະແຈ້ງ ແລະ

ບໍ່ຊັດເຈນເຊິ່ງພວກຂ້າພະເຈົ້າຍິນດີຮັບເອົາທຸກຄຳຕຳນິຕິຊົມຈາກບັນດາຜູ້ອ່ານ ແລະ

ຜູ້ນຳໃຊ້ເພື່ອຊື້ທິດທາງໃນການປັບປຸງ ແລະ ແກ້ໄຂບັນຫາໃຫ້ດີຂຶ້ນກ່ວາເກົ່າ.

ປະກາກຽດຕິຄຸນ

ຄະນະວິສະວະກຳສາດ ພາກວິຊາວິສະວະກຳຄອມພິວເຕີ, ກໍ່ຕື່ມະຫາວິທະຍາໄລສຸພານຸວົງ.

ຂ້າພະເຈົ້າເປັນນັກສຶກສາລຸ້ນທີ XIV ສຶກສາສາ 2023-2024 ຂໍສະແດງຄວາມຮູ້ບຸນຄຸນ ແລະ

ຂໍຂອບໃຈເປັນຢ່າງສູງມາຍັງບັນດາທ່ານອະທິການບໍດີ, ຄະນະບໍດີ ແລະ ຄູອາຈານຂອງມະຫາວິທະຍາໄລສຸພານຸວົງ,

ຄະນະນຳຂອງຄະນະວິສະວະກຳສາດ, ຄະນະນຳຂອງພາກວິຊາວິສະວະກຳຄອມພິວເຕີ,

ຕະຫຼອດເຖິງການຈັດຕັ້ງທຸກພາກສ່ວນທີ່ກ່ຽວຂ້ອງທີ່ໄດ້ສ້າງໂອກາດ ແລະ

ເອື້ອອຳນວຍຄວາມສະດວກໃຫ້ຂ້າພະເຈົ້າໄດ້ຮັບເຂົ້າສຶກສາທີ່ສະຖາບັນແຫ່ງນີ້

ເພື່ອຝຶກຝົນຫຼັກຫຼອມຕົນເອງໃຫ້ກາຍເປັນຜູ້ມີຄວາມຮູ້,

ຄວາມສາມາດກໍ່ຄືວິຊາສະເພາະທາງດ້ານວິສະວະກຳຄອມພິວເຕີເພື່ອຈະນຳເອົາຄວາມຮູ້ໄປປະກອບສ່ວນເຂົ້າໃນການປະຕິບັດວຽກງານຕົວຈິງໃນອະນາຄົດຕາມຄວາມເໝາະສົມ.

ທີ່ສຳຄັນເປັນພິເສດຂ້າພະເຈົ້າຂໍສະແດງຄວາມຮູ້ບຸນຄຸນ ແລະ ຂອບໃຈມາຍັງ ປທ. ຊ້າງຢ່າງ ແລະ ປທ. ບຸນໂຮມ ເມກສະຫວັນ ທີ່ເປັນອາຈານທີ່ປຶກສາ

ທີ່ປຽບສະເໝືອນແສງທຽນທີ່ເຍືອງທາງໃຫ້ການຂຽນບົດໂຄງການຈົບຊັ້ນໃນຄັ້ງນີ້ ເຊິ່ງໄດ້ໃຫ້ຄຳແນະນຳ,

ຄຳຄິດຄຳເຫັນອັນເປັນບົດຮຽນ ແລະ

ຊ່ວຍເຫຼືອມາໂດຍຕະຫຼອດໄລຍະໃນການດຳເນີນການຂຽນບົດໂຄງການຈົບຊັ້ນຂອງຂ້າພະເຈົ້າຄັ້ງນີ້ໄດ້ຮັບຜົນສຳເລັດເປັນຢ່າງດີ.

ສິ່ງທີ່ຂາດບໍ່ໄດ້ແມ່ນສະແດງຄວາມຮູ້ບຸນຄຸນມາຍັງພໍ່ແມ່ຂອງຂ້າພະເຈົ້າທີ່ໄດ້ໃຫ້ຄວາມອຸ່ມຊຸ່ຊ່ວຍ ເຫຼືອທາງດ້ານທຶນຮອນ ແລະ ໃຫ້ກຳລັງໃຈໃນເວລາມີບັນຫາໃນການສຶກສາຕະຫຼອດມາຈົນຂ້າພະເຈົ້າປະສົບຜົນສຳເລັດຕາມຄວາມມຸ່ງຫວັງ.

ອີກຢ່າງໜຶ່ງຂໍສະແດງຄວາມຂອບໃຈຢ່າງຈິງໃຈມາຍັງໝູ່ເພື່ອນໝົດທຸກຄົນທີ່ພວກເຮົາໄດ້ມີຄວາມສາມັກຄີຊ່ວຍເຫຼືອເຊິ່ງກັນ ແລະ ກັນຕະຫຼອດໄລຍະການຂຽນບົດໂຄງການເຊິ່ງເຮັດໃຫ້ບົດຂອງພວກເຮົາດຳເນີນໄປດ້ວຍດີ.

ສະນັ້ນ ຂ້າພະເຈົ້າຂໍປະຕິຍານຕົນວ່າຈະນຳເອົາຄວາມຮູ້ທີ່ໄດ້ສຶກສາມາຈາກຄູອາຈານໄປຜັນຂະຫຍາຍ ແລະ ພັດທະນາເຂົ້າໃນວຽກງານຕົວຈິງໃນອານາຄົດເພື່ອໃຫ້ເກີດປະໂຫຍດສູງສຸດ ຂໍຂອບໃຈ.

ທ້າວ ຢາດາວລີ ຈິ່ງນໍ

ສາລະບານ

ບົດນຳ	ງ
ປະກາກຽດຕິຄຸນ.....	ຈ
ບົດທີ 1 ບົດນຳ.....	1
I. ປະຫວັດຄວາມເປັນມາ ແລະ ທີ່ຕັ້ງຂອງໂຄງການ	1
1 ປະຫວັດຄວາມເປັນມາຂອງໂຄງການ	1
2 ທີ່ຕັ້ງຂອງໂຄງການ.....	1
3 ຄວາມຕ້ອງການຂອງລະບົບໃຫມ່.....	2
4 ຈຸດປະສົງຂອງບົດໂຄງການ	2
5 ຂອບເຂດຂອງບົດໂຄງການ.....	2
II. ຄາດຄະເນຜົນໄດ້ຮັບ	3
1. ຕໍ່ຫ້ອງຄົ້ນຄວ້າແລະວິໄຈຂອງມະຫາວິທະຍາໄລ ສຸພານຸວົງ.....	3
2. ຕໍ່ຜູ້ຜັດທະນາລະບົບ	3
3. ເຄື່ອງມືທີ່ໃຊ້ໃນການຜັດທະນາ	3
Hardware ທີ່ໃຊ້ຜັດທະນາ	3
Software ທີ່ໃຊ້ຜັດທະນາ.....	3
ບົດທີ 2 ທິດສະດີ ແລະ ຫຼັກການໃນການວິເຄາະອອກແບບລະບົບ	5
1. ຄວາມໝາຍຂອງລະບົບຂໍ້ມູນຂ່າວສານ.....	5
2. ທິດສະດີທີ່ກ່ຽວກັບການວິເຄາະ ແລະ ການອອກແບບລະບົບ	6
3. ນັກວິເຄາະລະບົບ (System Analysis ຫຼື SA)	6

4.ວົງຈອນການພັດທະນາລະບົບ (System Development Life Cycle)	6
5.ນັກວິເຄາະລະບົບ (System Analysis ຫຼື SA)	36
6.ວົງຈອນການພັດທະນາລະບົບ (System Development Life Cycle)	36
7.ການກຳນົດບັນຫາ (Problem Definition).....	36
8.ການວິເຄາະ (System Analysis)	37
9.ອອກແບບ (Design)	38
10. ພັດທະນາ (Development).....	38
11. ທົດສອບ (Testing).....	39
12. 5 ການສ້າງແບບຂໍ້ມູນ (Data Modeling)	39
13. 6 ການສ້າງແບບຈຳລອງລະບົບ (System Modeling).....	39
14. 7 ແຜນຜັງລະບົບ (System Flowchart)	39
15. 8 ແຜນພາບການໄຫຼຂອງກະແສຂໍ້ມູນ (DFD Data Flow Diagram)	40
16. 9 ຂັ້ນຕອນການແຕ້ມແຜນພາບການໄຫຼຂອງກະແສຂໍ້ມູນ	41
17. ສັນຍາລັກທີ່ໃຊ້ໃນແຜນພາບການໄຫຼຂອງກະແສຂໍ້ມູນ	41
18. ບຸນດາລີ (Boundaries).....	42
19. ໂປຣເຊດ (Process)	42
20. ດາຕ້າສະຕໍ (Data Store)	43
21. ດາຕ້າໄຟຣ (Data Flow)	43
22. ຂັ້ນຕອນການແຕ້ມແຜນພາບກະແສຂໍ້ມູນ.....	44
23. ຂັ້ນຕອນໃນການຂຽນແຜນພາບກະແສຂໍ້ມູນ	46
II.ຄວາມຮູ້ທົ່ວໄປກ່ຽວກັບ Next.js.....	54
III. ໂປຣແກຣມ, ເຄື່ອງມື ແລະ ພາສາທີ່ໃຊ້ໃນການພັດທະນາ ລະບົບ.....	57
24. 3 Firebase	62

25. ຖານຂໍ້ມູນ MongoDB	63
ບົດທີ 3	65
ວິເຄາະ ແລະ ອອກແບບລະບົບ	65
I. ສະພາບຂອງລະບົບເກົ່າ.....	65
II. ແຜນວາດລະບົບເກົ່າ.....	66
III. ສະພາບຂອງລະບົບໃໝ່.	66
IV. ສ່ວນປະກອບຂອງລະບົບໃໝ່.....	67
V. ແຜນວາດຂອງລະບົບໃໝ່	68
VIII. ແຜນພາບລວມການໄຫຼຂອງຂໍ້ມູນ (context Diagram)	68
VI. ແຜນພາບກະແສການໄຫຼຂໍ້ມູນຂອງຂໍ້ມູນລະດັບ 1 (DFD Level 1).....	69
VII. ແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນລະດັບ 2 ໂປຣເຊັດ 1.0 (DFD Level 2 Process 1.0)	70
VIII. ແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນລະດັບ 2 ໂປຣເຊັດ 2.0 (DFD Level 2 Process 2.0)	71
IX. ແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນລະດັບ 2 ໂປຣເຊັດ 2.0 (DFD Level 2 Process 2.0)	71
X. ແຜນພາບການໄຫຼຂໍ້ມູນ (ER: Entity Relationship).....	72
XI. ຕາຕະລາງຖານຂໍ້ມູນ (Data Dictionary)	72
1. ຕາຕະລາງຂໍ້ມູນຜູ້ໃຊ້ (tb_user)	72
2. ຕາຕະລາງຂໍ້ມູນປຶ້ມ (tb_book).....	72
3. ຕາຕະລາງຂໍ້ມູນສາຂາ (tb_departments)	73
4. ຕາຕະລາງຂໍ້ມູນພາກວິຊາ (tb_subject_departments).....	73
5. ຕາຕະລາງຂໍ້ມູນຄະນະ (tb_faculties)	73

6. ຕາຕະລາງຂໍ້ມູນການດາວໂຫຼດ (tb_download)	73
XII. ແຜນພາບການເຮັດວຽກຂອງລະບົບ (Algorithm)	74
ຮູບທີ 1. 1 ທີ່ຕັ້ງຂອງໂຄງການ	Error! Bookmark not defined.
ຮູບທີ 1. 2 ໄລຍະການດຳເນີນງານ	5
ຮູບທີ 2. 1 ວົງຈອນການຜັດທະນາລະບົບ	7
ຮູບທີ 2. 2 ຕົວຢ່າງໂປຣເຊສ.....	42
ຮູບທີ 2. 3 ດາຕ້າສະຕໍ	43
ຮູບທີ 2. 4 ສະແດງໂປຣເຊສທີ່ມີການຈັດການຂໍ້ມູນໃນ Data Store ແບບ Input.....	43
ຮູບທີ 2. 5 ສະແດງໂປຣເຊສທີ່ມີການ Output ຫຼື ເພີ່ມຂໍ້ມູນເຂົ້າໄປຍັງດາຕ້າສະຕໍ	43
ຮູບທີ 2. 6 ສະແດງໂປຣເຊສທີ່ມີການຈັດການຂໍ້ມູນໃນດາຕ້າສະຕໍແບບ Input ແລະ Output...	43
ຮູບທີ 2. 7 ເສັ້ນທາງການໄຫຼຂອງກະແສຂໍ້ມູນ	44
ຮູບທີ 2. 8 ຮູບພາບສະແດງແຜນພາບທີ່ຖືກຕ້ອງ	45
ຮູບທີ 2. 9 ຮູບພາບສະແດງແຜນພາບທີ່ບໍ່ຖືກຕ້ອງ	45
ຮູບທີ 2. 10 ຄວາມສຳພັນແບບໜຶ່ງຕໍ່ໜຶ່ງ.....	48
ຮູບທີ 2. 11 ຄວາມສຳພັນແບບໜຶ່ງຕໍ່ກຸ່ມ.....	49
ຮູບທີ 2. 12 ຄວາມສຳພັນແບບກຸ່ມຕໍ່ກຸ່ມ	49
ຮູບທີ 2. 13 ເອັນທິຕີແບບອ່ອນ.....	51
ຮູບທີ 2. 14 ຄວາມສຳພັນແບບປົກກະຕິ.....	52
ຮູບທີ 2. 15 ຄວາມສຳພັນແບບອ່ອນ.....	52
ຮູບທີ 2. 16 ແອັດທິບິວ.....	53

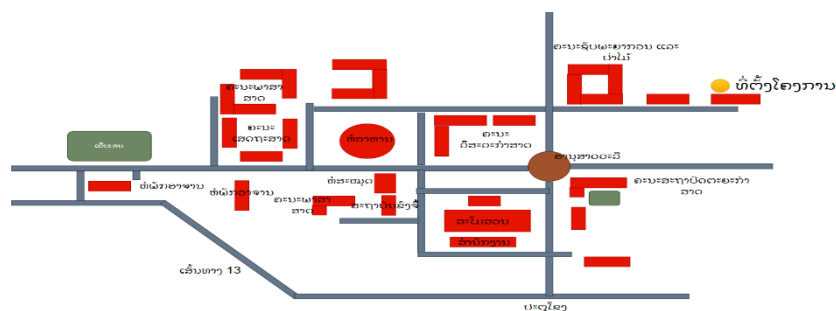
ຮູບທີ 2. 17 ແອັດທິບິວຊະນິດປົກກະຕິ.	53
ຮູບທີ 2. 18 ແອັດທິບິວຊະນິດກະແຈ.	53
ຮູບທີ 2. 19 ໂລໂກ້ຂອງ Next.js	57
ຮູບທີ 2. 20 ຮູບສະແດງໂປຣແກມ Visual Studio Code	58
ຮູບທີ 2. 21 ຮູບສະແດງການຕິດຕັ້ງ Node.js	59
ຮູບທີ 2. 22 ຮູບສະແດງການດາວໂຫຼດ Node.js	59
ຮູບທີ 2. 23 ຮູບສະແດງການກວດສອບ Node.js ແລະ ເວີຊັນ	60
ຮູບທີ 2. 24 ຮູບສະແດງການຕິດຕັ້ງ Extension ລົງໃນ visual studio code	61
ຮູບທີ 2. 25 ຮູບສະແດງການສ້າງໂປຣເຈັກ	61
ຮູບທີ 2. 26 ຮູບສະແດງການລົ້ມໂປຣເຈັກອອກໜ້າບຣາວເສີ	62
ຮູບທີ 2. 27 Firebase(Back-end for Google’s Mobile and Web App Development)	62
ຮູບທີ 2. 28 ຖານຂໍ້ມູນ MongoDB	64
ຮູບທີ 2. 29 ພາສາໂປຣແກຣມ Node.js	65
ຮູບທີ 3. 1 ແຜນວາດຂອງລະບົບເກົ່າ.	66
ຮູບທີ 3. 1 ແຜນວາດຂອງລະບົບເກົ່າ.	Error! Bookmark not defined.
ຮູບທີ 3. 2 ແຜນວາດຂອງລະບົບໃໝ່.	66
ຮູບທີ 3. 3 ແຜນວາດຂອງລະບົບໃໝ່ (Process Hierachy Chart)	68
ຮູບທີ 3. 4 ສະແດງແຜນພາບລວມການໄຫຼຂອງຂໍ້ມູນ (context diagram)	68
ຮູບທີ 3. 5 ສະແດງແຜນພາບກະແສການໄຫຼຂໍ້ມູນລະດັບ 1 (DFD Level 1)	69
ຮູບທີ 3. 6 ສະແດງແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນລະດັບ 2 ໂປຣເຊັດ 1.0 (DFD Level 2 Process 1.0)	70
ຮູບທີ 3. 7 ສະແດງແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນລະດັບ 2 ໂປຣເຊັດ 2.0 (DFD Level 2 Process 2.0)	71
ຮູບທີ 3. 8 ສະແດງແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນລະດັບ 2 ໂປຣເຊັດ 3.0 (DFD Level 2 Process 3.0)	71
ຮູບທີ 3. 9 ຮູບທີ 3.9 ສະແດງແຜນພາບການໄຫຼຂໍ້ມູນ (ER: Entity Relationship)	72

ຮູບທີ່ 3. 10 ແຜນພາບການເຮັດວຽກຂອງລະບົບ (Algorithm)	74
---	----

ບົດນຳ

1 ປະຫວັດຄວາມເປັນມາຂອງໂຄງການ

ຫ້ອງການວິໄຈຂອງຫຼັກສູດປະລິນຍາໂທຢູ່ຕິດກັບ ຄະນະຊັບຜະຍາກອນ ແລະ ປ່າໄມ້



ຮບທິ 1. 1 ທີ່ຕັ້ງຂອງໂຄງການ

ຂອງມະຫາວິທະຍາໄລສຸພານຸວົງທີ່ ບ້ານ ດອນໃໝ່ ນະຄອນຫຼວງພະບາງ ແຂວງຫຼວງພະບາງ.

ຮູບ 1.2 ທີ່ຕັ້ງຂອງຫ້ອງການວິໄຈຫຼັກສູດປະລິນຍາໂທ

ບັນຫາຂອງລະບົບປັດຈຸບັນ

ຍັງບໍ່ທັນມີລະບົບຈັດເກັບປຶ້ມໃຫ້ລະອຽດ ແລະ ເປັນລະບຽບ.

ການເກັບຮັກສາບົດວິທະຍານິພົນແມ່ນໃຊ້ຮູບແບບເປັນ File Zip ຫຼື ເປັນປຶ້ມ ເຊິ່ງເຮັດໃຫ້ມີຄວາມຫຍຸ້ງຍາກໃນການຄົ້ນຫາຂໍ້ມູນ ແລະ ສະຫຼຸບລາຍງາຍຕ່າງໆ.

ຕ້ອງໃຊ້ຜູ້ທີ່ ແລະ ງົບປະມານໃນການເກັບກຳປຶ້ມ.

ຍັງບໍ່ທັນມີລະບົບທີ່ໃຫ້ບໍລິການໃນຈັດເກັບ File ປຶ້ມ.

ເຮັດໃຫ້ການເຂົ້າເຖິງ ບົດວິທະຍານິພົນ ຂອງນັກສຶກສາຫຍຸ້ງຍາກ

3 ຄວາມຕ້ອງການຂອງລະບົບໃຫມ່

ຕ້ອງການລະບົບຖານຂໍ້ມູນ ເພື່ອຮອງຮັບການຈັດເກັບຂໍ້ມູນປຶ້ມ.

ຕ້ອງການລະບົບຖານຂໍ້ມູນທີ່ມີຄວາມປອດໄພ ເພື່ອຈັດເກັບຂໍ້ມູນປຶ້ມໃຫ້ເປັນລະບົບ.

ຕ້ອງການສະແດງຂໍ້ມູນປຶ້ມ ດ້ວຍລະບົບປະຕິບັດການໃນເວັບແອັບ.

ຕ້ອງການໃຫ້ນັກສຶກສາ ແລະ ອາຈານສາມາດນຳໃຊ້ເວັບແອັບໃນການຄົ້ນຫາຂໍ້ມູນປຶ້ມໄດ້ຢ່າງສະດວກ ແລະ ວ່ອງໄວ.

4 ຈຸດປະສົງຂອງບົດໂຄງການ

ເພື່ອສຶກສາ ແລະ ອອກແບບລະບົບຈັດການບົດວິທະຍານິພົນ ຂອງມະຫາວິທະຍາໄລສຸພານຸວົງໃນຮູບແບບ Web

ເພື່ອເປັນບ່ອນຈັດເກັບຂໍ້ມູນບົດໂຄງການຈົບຊັ້ນປະລິນຍາໂທ.

ເພື່ອສ້າງ Web ສາມາດໃຫ້ຂໍ້ມູນເພື່ອການສຶກສາໄດ້ຢ່າງສະດວກ ແລະ ວ່ອງໄວກວ່າເກົ່າ.

ເພື່ອຄວາມສາມາດຄົ້ນຫາຂໍ້ມູນປຶ້ມໄດ້ໂດຍຜ່ານ ຫົວຂໍ້, ຊື່ຜູ້ຂຽນ, ປະເພດຂອງປຶ້ມ ແລະ ປີຕີພິມໄດ້.

ເພື່ອຍົກສູງປະສິດທິພາບທາງດ້ານ ການບໍລິການຂໍ້ມູນປຶ້ມ.

5 ຂອບເຂດຂອງບົດໂຄງການ

ລະບົບສາມາດຈັດການຂໍ້ມູນປຶ້ມ, ຂໍ້ມູນປະເພດປຶ້ມ, ຂໍ້ມູນຄະນະ, ຂໍ້ມູນພາກວິຊາ, ຂໍ້ມູນສາຂາວິຊາ, ຂໍ້ມູນຜູ້ໃຊ້ ແລະ ຂໍ້ມູນຜູ້ດູແລລະບົບໄດ້.

ລະບົບສາມາດສະແດງຂໍ້ມູນປຶ້ມ, ຂໍ້ມູນປະເພດປຶ້ມ, ຂໍ້ມູນພາກວິຊາ, ຂໍ້ມູນສາຂາວິຊາ, ຂໍ້ມູນຜູ້ໃຊ້ ແລະ ຜູ້ຂໍ້ມູນດູແລລະບົບ ຂອງມະຫາໄລໄດ້ຢ່າງລະອຽດ.

ລະບົບສາມາດຄົ້ນຫາຂໍ້ມູນປຶ້ມ, ປະເພດປຶ້ມ, ພາກວິຊາ, ສາຂາວິຊາ, ອາຈານ, ຜູ້ໃຊ້ໄດ້ ແລະ ຂໍ້ມູນຜູ້ດູແລລະບົບ.

ລະບົບສາມາດເກັບໄຟລ໌ປຶ້ມ, ສາມາດເປີດອ່ານ ແລະ ດາວໂຫຼດໄດ້.

II. ຄາດຄະເນຜົນໄດ້ຮັບ

1. ຕໍ່ຫ້ອງຄົ້ນຄວ້າແລະວິໄຈຂອງມະຫາວິທະຍາໄລ ສຸພານຸວົງ

ມີລະບົບຈັດການບົດວິທະຍານິພົນເປັນຂອງຫ້ອງຄົ້ນຄວ້າ ແລະ ວິໄຈສາມາດຈັດເກັບ ແລະ ລາຍງານຂໍ້ມູນໄດ້.

ເຮັດໃຫ້ນັກສຶກສາ ຫຼື ຜູ້ໃຊ້ສາມາດເຂົ້າເຖິງຂໍ້ມູນ ເພື່ອເປັນຊ່ອງທາງໃນການສຶກສາຮ່າຮຽນ, ການຂຽນບົດລາຍງານ, ຂຽນບົດໂຄງການ ແລະ ອື່ນໆໄດ້ຢ່າງສະດວກ.

ປະຢັດຄ່າໃຊ້ຈ່າຍ ແລະ ປະຢັດຜືນທີ່ໃນການຈັດເກັບປຶ້ມໃນຮູບແບບເກົ່າ.

2. ຕໍ່ຜູ້ພັດທະນາລະບົບ

ໄດ້ເຂົ້າໃຈເຖິງວິທີການເຮັດບົດໂຄງການຈົບຊັ້ນ

ໄດ້ສຶກສາເຖິງບັນຫາຂອງລະບົບ ແລະ ເພີ່ມທັກສະໃນການແກ້ໄຂບັນຫາຂອງລະບົບ

ໄດ້ຮຽນຮູ້ການວິເຄາະ ແລະ ອອກແບບ ລະບົບລະບົບຈັດການບົດວິທະຍານິພົນ

ປະລິຍາໂທຂອງມະຫາວິທະຍາໄລສຸພານຸວົງ

ໄດ້ຮຽນຮູ້ ແລະ ເພີ່ມທັກສະໃນການພັດທະນາ Software

3. ເຄື່ອງມືທີ່ໃຊ້ໃນການພັດທະນາ

- Hardware ທີ່ໃຊ້ພັດທະນາ

Notebook Acer.

Processor: Intel® core™ i7-8750H CPU @ 2.20GHz (12 CPUs), ~2.2GHz

Installed RAM: 8.0 GB.

SSD M.2 : 1TB.

- Software ທີ່ໃຊ້ພັດທະນາ

ລະບົບປະຕິບັດການ Windows 10 Home.

Microsoft Office LTSC Professional Plus 2021.

Draw.io

Visual Studio Code.

Framework Next.js

ພາສາໂປຣແກຣມທີ່ໃຊ້: Node.js

ຖານຂໍ້ມູນ (Database): Firebase, MongoDB

III. ໄລຍະການດຳເນີນງານ

ໜ້າວຽກ / ເວລາ	9/2024				10/2024				11/2024				12/2024				01/2025				02/2025				03/2025				04/2025				05/2025				06/2025			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
ບົດໂຄງການ I																																								
ເລືອກຫົວຂໍ້ ແລະ ອາຈານທີ່ປຶກສາ																																								
ຄົ້ນຄວ້າທິດສະດີທີ່ກ່ຽວຂ້ອງ																																								
ຂຽນ ແລະ ປ້ອງກັນຫົວຂໍ້																																								
ຂຽນບົດໂຄງການ I ແລະ ຜ່ານອາຈານທີ່ປຶກສາ																																								
ວິເຄາະ ແລະ ອອກແບບລະບົບ																																								
ປ້ອງກັນບົດໂຄງການ I																																								
ບົດໂຄງການ II																																								
ເລີ່ມຕົ້ນພັດທະນາ ແລະ ທົດສອບລະບົບ																																								
ກວດແກ້ ແລະ ສົ່ງບົດໂຄງການ																																								
ປ້ອງກັນບົດຈົບຊັ້ນ																																								

ຮູບທີ 1. 2 ໄລຍະການດຳເນີນງານ

ບົດທີ 2

ທິດສະດີ ແລະ ຫຼັກການໃນການວິເຄາະອອກແບບລະບົບ

I. ທິດສະດີພື້ນຖານກ່ຽວກັບລະບົບຂໍ້ມູນຂ່າວສານ

1. ຄວາມໝາຍຂອງລະບົບຂໍ້ມູນຂ່າວສານ

ລະບົບຂໍ້ມູນຂ່າວສານ (IS) ເປັນລະບົບການຈັດຕັ້ງທີ່ເກັບກຳ, ເກັບຮັກສາ, ຂະບວນການ ແລະ ແຈກຢາຍຂໍ້ມູນຂ່າວສານເພື່ອສະໜັບສະໜູນການຕັດສິນໃຈ, ການປະສານງານ, ການຄວບຄຸມ, ການວິເຄາະ ແລະ ການເບິ່ງເຫັນພາຍໃນອົງການຈັດຕັ້ງ ຫຼື ສະພາບການ.

ລະບົບຂໍ້ມູນຂ່າວສານປະກອບດ້ວຍປະຊາຊົນ, ຂະບວນການ, ຂໍ້ມູນ, ຊອບແວ, ຮາດແວ ແລະ ເຄືອຂ່າຍ.

ມັນຖືກອອກແບບມາເພື່ອຈັດການແລະປະມວນຜົນຂໍ້ມູນຢ່າງມີປະສິດທິພາບ, ຊ່ວຍໃຫ້ຜູ້ໃຊ້ເຂົ້າເຖິງສາມາດຈັດການ ແລະ ນຳໃຊ້ຂໍ້ມູນຢ່າງມີປະສິດທິພາບ.

ຄວາມສຳຄັນຂອງລະບົບຂໍ້ມູນຂ່າວສານ

ລະບົບຂໍ້ມູນຂ່າວສານມີຄວາມສຳຄັນໃນຫຼາຍດ້ານເຊັ່ນ: ທຸລະກິດ, ການສຶກສາ, ການດູແລສຸຂະພາບ ແລະ ລັດຖະບານ, ຊ່ວຍປັບປຸງການດຳເນີນງານ, ປັບປຸງການສື່ສານ ແລະ ຊຸກຍູ້ການວາງແຜນຍຸດທະສາດ. ປະເພດສຳຄັນຂອງລະບົບຂໍ້ມູນຂ່າວສານປະກອບມີ:

ລະບົບປະມວນຜົນທຸລະກຳ (TPS) - ຈັດການທຸລະກຳປົກກະຕິຢ່າງມີປະສິດທິພາບ.

ລະບົບຂໍ້ມູນຂ່າວສານການຄຸ້ມຄອງ (MIS) - ສະໜອງບົດລາຍງານປົກກະຕິແລະບົດສະຫຼຸບສຳລັບ ການຄຸ້ມຄອງ.

Decision Support Systems (DSS) - ຊ່ວຍເຮັດການຕັດສິນໃຈໂດຍການວິເຄາະຂໍ້ມູນ.

ລະບົບຂໍ້ມູນຂ່າວສານບໍລິຫານ (EIS) - ສະເໜີຄວາມເຂົ້າໃຈລະດັບສູງສຳລັບຜູ້ບໍລິຫານ.

ລະບົບການຄຸ້ມຄອງ (KMS) ຊ່ວຍເຫຼືອໃນການແລກປ່ຽນແລະການຄຸ້ມຄອງຄວາມຮູ້ອົງການຈັດຕັ້ງ.

ໂດຍການນຳໃຊ້ລະບົບເຫຼົ່ານີ້, ອົງການຈັດຕັ້ງສາມາດໄດ້ຮັບຜົນປະໂຫຍດດ້ານການແຂ່ງຂັນ, ປັບປຸງການຜະລິດ ແລະ ເສີມຂະຫຍາຍການຕັດສິນໃຈ.

2. ທິດສະດີທີ່ກ່ຽວກັບການວິເຄາະ ແລະ ການອອກແບບລະບົບ

ລະບົບແມ່ນ ກຸ່ມຂອງອົງປະກອບຕ່າງໆ ທີ່ເຮັດວຽກຮ່ວມກັນເພື່ອຈຸດປະສົງອັນດຽວກັນ ລະບົບອາດຈະປະກອບດ້ວຍ ບຸກຄະລາກອນ ເຄື່ອງມື ເຄື່ອງໃຊ້ ພັດສະດຸ ວິທີການ ເຊິ່ງທັງຫມົດນີ້ຕ້ອງມີລະບົບຈັດການອັນໜຶ່ງ ເພື່ອໃຫ້ບັນລຸຈຸດປະສົງອັນດຽວກັນ ເຊັ່ນ: ລະບົບການຊອກຫາຂໍ້ມູນປຶ້ມ ມີຈຸດປະສົງເພື່ອໃຫ້ຜູ້ທີ່ຊອກຫາຂໍ້ມູນ ໄດ້ຮັບຂໍ້ມູນນັ້ນຢ່າງຖືກຕ້ອງ.

ການວິເຄາະລະບົບ ແລະ ການອອກແບບ (System Analysis and Design) ການວິເຄາະ ແລະ ການອອກແບບລະບົບຄື ວິທີການທີ່ໃຊ້ໃນການສ້າງລະບົບຂໍ້ມູນຂ່າວສານຂຶ້ນມາໃໝ່ໃນທຸລະກິດໃດໜຶ່ງ ຫຼື ລະບົບຍ່ອຍຂອງທຸລະກິດ ນອກຈາກການສ້າງລະບົບຂໍ້ມູນຂ່າວສານໃໝ່ແລ້ວ ການວິເຄາະລະບົບຊ່ວຍໃນການແກ້ໄຂລະບົບຂໍ້ມູນຂ່າວສານໂຕເກົ່າທີ່ມີຢູ່ແລ້ວໃຫ້ດີຂຶ້ນນຳກໍໄດ້ ການວິເຄາະລະບົບແມ່ນ ການຊອກຫາຄວາມຕ້ອງການ (Requirements) ຂອງລະບົບຂໍ້ມູນຂ່າວສານວ່າແມ່ນຫຍັງ ຫຼື ຕ້ອງການເພີ່ມເຕີມຫຍັງເຂົ້າມາໃນລະບົບ ແລະ ການອອກແບບແມ່ນ ການນຳເອົາຄວາມຕ້ອງການຂອງລະບົບມາເປັນແບບແຜນ ຫຼື ເອີ້ນວ່າ ພົມຂຽວ ໃນການການສ້າງລະບົບຂໍ້ມູນຂ່າວສານນັ້ນໃຫ້ໃຊ້ງານໄດ້ຈິງ ຕົວຢ່າງ ລະບົບຂໍ້ມູນຂ່າວສານ ເຊັ່ນ: ລະບົບການໃຫ້ບໍລິການຂໍ້ມູນປຶ້ມ ຄວາມຕ້ອງການຂອງລະບົບຄື ສາມາດສະແດງ ແລະ ຄົ້ນຫາປຶ້ມນັ້ນໄດ້ຢ່າງຖືກຕ້ອງ.

3. ນັກວິເຄາະລະບົບ (System Analysis ຫຼື SA)

ນັກວິເຄາະລະບົບແມ່ນ ບຸກຄົນທີ່ມີໜ້າທີ່ວິເຄາະ ແລະ ອອກແບບລະບົບ ເຊິ່ງປົກກະຕິແລ້ວນັກວິເຄາະລະບົບຄວນຈະຢູ່ໃນທີມລະບົບຂໍ້ມູນຂ່າວສານຂອງອົງກອນ ຫຼື ທຸລະກິດນັ້ນໆ ການທີ່ມີນັກວິເຄາະລະບົບໃນອົງກອນນັ້ນເປັນການໄດ້ປຽບເພາະຈະຮູ້ໄດ້ໂດຍລະອຽດວ່າ ການເຮັດວຽກໃນລະບົບນັ້ນເປັນແບບໃດ ແລະ ຄວາມຕ້ອງການຂອງລະບົບນັ້ນແມ່ນຫຍັງ ໃນລະນີ້ນັກວິເຄາະລະບົບບໍ່ໄດ້ຢູ່ໃນອົງກອນນັ້ນໆ ກໍສາມາດວິເຄາະລະບົບໄດ້ເຊັ່ນກັນໂດຍການສຶກສາສອບຖາມຜູ້ໃຊ້ ແລະ ວິທີການອື່ນໆ ເຊິ່ງຈະກ່າວໃນພາຍຫຼັງ ຜູ້ໃຊ້ໃນທີ່ນີ້ກໍຄືເຈົ້າຂອງ ແລະ ຜູ້ທີ່ກ່ຽວຂ້ອງໃນລະບົບຂໍ້ມູນຂ່າວສານນັ້ນເອງ ຜູ້ໃຊ້ອາດຈະເປັນຄົນດຽວ ຫຼື ຫຼາຍຄົນກໍໄດ້ ເພື່ອເຮັດໃຫ້ນັກວິເຄາະລະບົບເຮັດວຽກໄດ້ຢ່າງວ່ອງໄວ ມີລຳດັບຂັ້ນ ແລະ ເປົ້າໝາຍທີ່ແນ່ນອນ ນັກວິເຄາະລະບົບຄວນຮູ້ວ່າ ລະບົບຂໍ້ມູນຂ່າວສານນັ້ນພັດທະນາຂຶ້ນມາແບບໃດ ມີຂັ້ນຕອນແບບໃດແດ່.

4. ວົງຈອນການພັດທະນາລະບົບ (System Development Life Cycle)

ວົງຈອນການພັດທະນາລະບົບ (System Development Life Cycle) ລະບົບຂໍ້ມູນຂ່າວສານທັງຫລາຍມີວົງຈອນການພັດທະນາທີ່ຄືກັນ ຕັ້ງແຕ່ເກີດຈົນສິ້ນ ວົງຈອນນີ້ຈະເປັນຂັ້ນຕອນທີ່ເປັນລຳດັບຕັ້ງແຕ່ຕົ້ນຈົນຈົບຮຽບຮ້ອຍ ເປັນລະບົບທີ່ໃຊ້ງານໄດ້

ເຊິ່ງນັກວິເຄາະລະບົບຕ້ອງທຳຄວາມເຂົ້າໃຈໃຫ້ດີວ່າ ໃນແຕ່ລະຂັ້ນຕອນຕ້ອງໄດ້ເຮັດຫຍັງແດ່ ແລະ ເຮັດແບບໃດ ຂັ້ນຕອນການພັດທະນາລະບົບມີຢູ່ນຳກັນ 7 ຂັ້ນຕອນຄື:

ກຳນົດບັນຫາ (Problem Recognition)

ວິເຄາະລະບົບ (System Analysis)

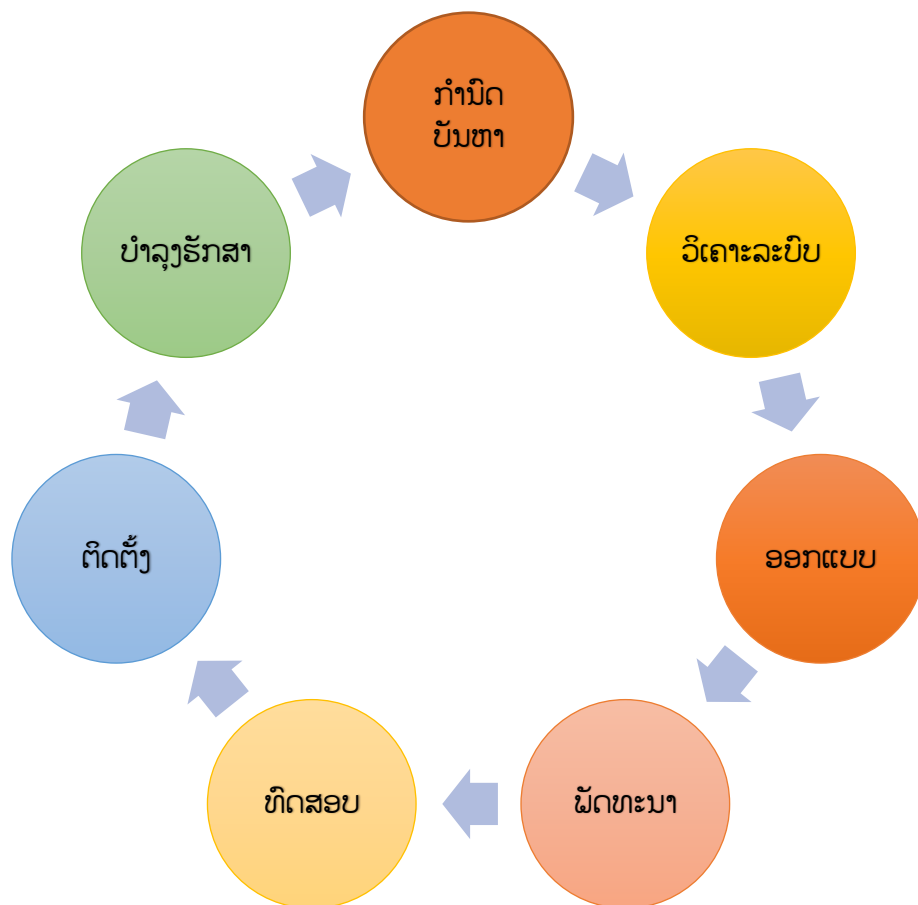
ອອກແບບ (Design)

ພັດທະນາລະບົບ (Development)

ທົບສອດ (Testing)

ຕິດຕັ້ງ (Deployment)

ບຳລຸງຮັກສາ (Maintenance)



ຮູບທີ 2. 1 ວົງຈອນການພັດທະນາລະບົບ

5. ນັກວິເຄາະລະບົບ (System Analysis ຫຼື SA)

ນັກວິເຄາະລະບົບແມ່ນ ບຸກຄົນທີ່ມີໜ້າທີ່ວິເຄາະ ແລະ ອອກແບບລະບົບ ເຊິ່ງປົກກະຕິແລ້ວນັກວິເຄາະລະບົບຄວນຈະຢູ່ໃນທີມລະບົບຂໍ້ມູນຂ່າວສານຂອງອົງກອນ ຫຼື ທຸລະກິດນັ້ນໆ ການທີ່ມີນັກວິເຄາະລະບົບໃນອົງກອນນັ້ນເປັນການໄດ້ປຽບເພາະຈະຮູ້ໄດ້ໂດຍລະອຽດວ່າ ການເຮັດວຽກໃນລະບົບນັ້ນເປັນແບບໃດ ແລະ ຄວາມຕ້ອງການຂອງລະບົບນັ້ນແມ່ນຫຍັງ ໃນລະນັ້ນນັກວິເຄາະລະບົບບໍ່ໄດ້ຢູ່ໃນອົງກອນນັ້ນໆ ກໍສາມາດວິເຄາະລະບົບໄດ້ເຊັ່ນກັນໂດຍການສຶກສາສອບຖາມຜູ້ໃຊ້ ແລະ ວິທີການອື່ນໆ ເຊິ່ງຈະກ່າວໃນພາຍຫຼັງ ຜູ້ໃຊ້ໃນທີ່ນີ້ກໍຄືເຈົ້າຂອງ ແລະ ຜູ້ທີ່ກ່ຽວຂ້ອງໃນລະບົບຂໍ້ມູນຂ່າວສານນັ້ນເອງ ຜູ້ໃຊ້ອາດຈະເປັນຄົນດຽວ ຫຼື ຫຼາຍຄົນກໍໄດ້ ເພື່ອເຮັດໃຫ້ນັກວິເຄາະລະບົບເຮັດວຽກໄດ້ຢ່າງວ່ອງໄວ ມີລຳດັບຂັ້ນ ແລະ ເປົ້າໝາຍທີ່ແນ່ນອນ ນັກວິເຄາະລະບົບຄວນຮູ້ວ່າ ລະບົບຂໍ້ມູນຂ່າວສານນັ້ນພັດທະນາຂຶ້ນມາແບບໃດ ມີຂັ້ນຕອນແບບໃດແດ່.

6. ວົງຈອນການພັດທະນາລະບົບ (System Development Life Cycle)

ວົງຈອນການພັດທະນາລະບົບ (System Development Life Cycle) ລະບົບຂໍ້ມູນຂ່າວສານທັງຫລາຍມີວົງຈອນການພັດທະນາທີ່ຄືກັນ ຕັ້ງແຕ່ເກີດຈົນສິ້ນ ວົງຈອນນີ້ຈະເປັນຂັ້ນຕອນທີ່ເປັນລຳດັບຕັ້ງແຕ່ຕົ້ນຈົນຈົບຮຽບຮ້ອຍ ເປັນລະບົບທີ່ໃຊ້ງານໄດ້ ເຊິ່ງນັກວິເຄາະລະບົບຕ້ອງທຳຄວາມເຂົ້າໃຈໃຫ້ດີວ່າ ໃນແຕ່ລະຂັ້ນຕອນຕ້ອງໄດ້ເຮັດຫຍັງແດ່ ແລະ ເຮັດແບບໃດ ຂັ້ນຕອນການພັດທະນາລະບົບມີຢູ່ນຳກັນ 7 ຂັ້ນຕອນຄື:

ກຳນົດບັນຫາ (Problem Recognition)

ວິເຄາະລະບົບ (System Analysis)

ອອກແບບ (Design)

ພັດທະນາລະບົບ (Development)

ທົບສອດ (Testing)

ຕິດຕັ້ງ (Deployment)

ບຳລຸງຮັກສາ (Maintenance)

7. ການກຳນົດບັນຫາ (Problem Definition)

ໃນການກຳນົດບັນຫາຈະຕ້ອງມີການແຍກແຍະລາຍລະອຽດຂອງບັນ ແລະ ຕ້ອງຊີ້ໃຫ້ວ່າບັນຫາທີ່ເກີດຂຶ້ນແທ້ໆນັ້ນຢູ່ປອນໃດເປັນແນວໃດ

ໂດຍການຫາຂໍ້ມູນຂອງບັນຫາເຊັ່ນຂະໜາດຄວາມສໍາຄັນຕະຫຼອດຈົນໄລຍະເວລາທີ່ຈໍາເປັນຕ້ອງໃຊ້ໃນການແກ້ໄຂບັນຫາໃຫ້ແລ້ວສໍາເລັດໂດຍໃນຂັ້ນທໍາອິດຈະຕ້ອງໃຫ້ຄວາມໝາຍຂອງບັນຫາຢ່າງກວ້າງໆ ແລ້ວຈຶ່ງພະຍາຍາມຫຼຸດຂໍ້ບັງຄັບຂໍ້ຈໍາກັດ ຫຼື ກົດເກນຕ່າງໆລົງ ແລະ ບໍ່ຄວນຈະໃຫ້ຄວາມສໍາຄັນ ຫຼື ຄວາມສົນໃຈກັບວິທີການທີ່ເຮັດຢູ່ໃນຂະນະນັ້ນ (Present Method) ຫຼາຍຈົນເກີນໄປ ເພື່ອໃຫ້ມີອິດສະຫຼະໃນການສ້າງສັນວິທີການແກ້ໄຂບັນຫາບາງເທື່ອອາດຈະແບ່ງບັນຫາທີ່ເກີດຂຶ້ນເປັນບັນຫາຍ່ອຍຫຼາຍໆສ່ວນກໍ່ໄດ້, ວິທີນີ້ອາດເຮັດໃຫ້ບັນຫາຫຼັກທີ່ມີຄວາມຊັບຊ້ອນນັ້ນສາມາດວິເຄາະ ແລະ ແກ້ໄຂໄດ້ໂດຍງ່າຍຂຶ້ນ.

8. ການວິເຄາະ (System Analysis)

ເປັນການສຶກສາເຖິງຂັ້ນຕອນຂອງການດໍາເນີນງານ ແລະ ບັນຫາທີ່ເກີດຂຶ້ນໃນລະບົບປັດຈຸບັນ (Current System) ເພື່ອອອກແບບລະບົບໃໝ່ (New System), ເປົ້າໝາຍໃນການຕ້ອງການປັບປຸງ ແລະ ແກ້ໄຂລະບົບການດໍາເນີນງານເກົ່າໃຫ້ມີທິດທາງທີ່ດີຂຶ້ນ ໂດຍການນໍາລະບົບໃໝ່ມາກໍານົດຄວາມຕ້ອງການ (Requirement specification)

ທີ່ໄດ້ຈາກຂັ້ນຕອນທໍາອິດມາວິເຄາະໃນລາຍລະອຽດເພື່ອທໍາການພັດທະນາເປັນແບບຈໍາລອງ (Logical Design), ເຊິ່ງປະກອບດ້ວຍແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນ (Data Flow Design), ຄໍາອະທິບາຍການປະມວນຜົນຂໍ້ມູນ (Process Description) ແລະ ແບບຈໍາລອງຂໍ້ມູນ (Data Model), ແບບຂອງ (ER Diagram) ເຮັດໃຫ້ຮູ້ເຖິງລາຍລະອຽດຂັ້ນຕອນການດໍາເນີນງານໃນລະບົບປະກອບມີຫຍັງແຕ່ມີຄວາມກ່ຽວຂ້ອງ ຫຼື ຄວາມສໍາພັນກັບສິ່ງໃດແດ່.

ການວິເຄາະລະບົບແມ່ນນໍາເອົາຂໍ້ມູນຕ່າງໆທີ່ໄດ້ຮວບຮວມຈາກຄວາມຕ້ອງການຂອງລະບົບເກົ່າມາວິເຄາະເພື່ອສະຫຼຸບເປັນຂໍ້ກໍານົດທີ່ຊັດເຈນ ແລະ

ນໍາຂໍ້ຈໍາກັດເຫຼົ່ານັ້ນໄປພັດທະນາຄວາມຕ້ອງການຂອງລະບົບໃໝ່ດ້ວຍການພັດທະນາແບບຈໍາລອງຂຶ້ນມາເຊັ່ນ: ແບບຈໍາລອງຂະບວນການ (Data Flow Diagram) ແລະ ແບບຈໍາລອງຂໍ້ມູນ (Data Model) ເປັນຕົ້ນ.

ຊຶ່ງຂັ້ນຕອນການວິເຄາະມີດັ່ງນີ້:

1. ສຶກສາ ແລະ ວິເຄາະລະບົບເກົ່າ.
2. ລວບລວມຄວາມຕ້ອງການ ແລະ ກໍານົດຄວາມຕ້ອງການຂອງລະບົບໃໝ່.
3. ວິເຄາະຄວາມຕ້ອງການເພື່ອສະຫຼຸບເປັນຂໍ້ກໍານົດ.
4. ສ້າງແຜນພາບ DFD ແລະ ແຜນພາບ ER.

9. ອອກແບບ (Design)

ໃນການອອກແບບເປັນການນຳເອົາຜົນລັບທີ່ໄດ້ຈາກການວິເຄາະທາງຕົກກະສາດ,ທີ່ຖືກຕ້ອງຕາມເຫດຜົນ(Logical) ມາພັດທະນາເປັນຮູບແບບທາງກາຍະພາບ ຫຼື ແບບຈຳລອງທາງກາຍະພາບ(Physical Model)ໃຫ້ສອດຄ່ອງກັນໂດຍການອອກແບບຈະນຳຈາກສ່ວນອຸປະກອນ ແລະ ເຕັກໂນໂລຊີຕ່າງໆ ແລະ ໂປຣແກຣມຄອມພິວເຕີທີ່ນຳມາພັດທະນາ, ການອອກແບບຈຳລອງຂໍ້ມູນ (Data Model), ການອອກແບບຈຳພາບໃນການຕິດຕໍ່ກັບຜູ້ໃຊ້ງານ (User interface), ການຈັດເຮັດວັດຈະນານຸກົມຂໍ້ມູນ (Data Dictionary).

10.ພັດທະນາ (Development)

ການພັດທະນາເປັນຂັ້ນຕອນການພັດທະນາໂປຼແກຼມໂດຍການຊຸດຄຳສັ່ງ ຫຼືຂຽນໂປຼແກຼມເພື່ອການສ້າງລະບົບງານໂດຍໂປຼແກຼມທີ່ໃຊ້ໃນການພັດທະນາຈະຕ້ອງພິຈາລະນາເຖິງຄວາມເໝາະສົມກັບເຕັກໂນໂລຢີທີ່ໃຊ້ງານຢູ່.ເຊິ່ງໃນປັດຈຸບັນພາສາລະດັບສູງໄດ້ມີການພັດທະນາໃນຮູບແບບຂອງ 4GL(Fourth-Generation Languages) ເຊິ່ງເປັນການພັດທະນາຂອງພາສາຄອມພິວເຕີໃນແຕ່ລະຍຸກ, 4GL ເປັນການພັດທະນາພາສາຄອມພິວເຕີໃນຍຸກທີ່ 4 ນັ້ນເອງເນື່ອງຈາກພາສາຂັ້ນສູງເປັນພາສາທີ່ຕ້ອງກຳນົດຂັ້ນຕອນການເຮັດວຽກຈຶ່ງເຮັດໃຫ້ບາງເທື່ອຈຳເປັນຕ້ອງໄດ້ຂຽນໂຄດໂປຼແກຼມທີ່ຍາວຫຼາຍແຖວກ່ອນທີ່ຈະໄດ້ຜົນລັບທີ່ຕ້ອງການ,ດັ່ງນັ້ນຈຶ່ງກຳນົດພາສາທີ່ 4 ເຊິ່ງເປັນພາສາທີ່ບໍ່ຕ້ອງກຳນົດຂັ້ນຕອນການເຮັດວຽກເຊິ່ງອຳນວຍຄວາມສະດວກຕໍ່ການພັດທະນາລວມທັງການ CASE(Computer Aided Software Engineering)ຕ່າງໆຢ່າງຫຼວງຫຼາຍໃຫ້ເລືອກໃຊ້ຕາມຄວາມເໝາະສົມ.

ຂັ້ນຕອນການພັດທະນາມີຄື:

1. ພັດທະນາໂປຼແກຼມຈາກທີ່ໄດ້ທຳການວິເຄາະ ແລະ ອອກແບບໄວ້.
2. ເລືອກພາສາທີ່ເໝາະສົມ ແລະ ພັດທະນາຕໍ່ໄດ້ງ່າຍ.
3. ອາດຈຳເປັນຕ້ອງໃຊ້ເຄື່ອງມືໃນການພັດທະນາເພື່ອເພີ່ມຄວາມສະດວກ ແລະ ກວດສອບແກ້ໄຂທີ່ວ່ອງໄວຂຶ້ນ ແລະ ເປັນໄປຕາມມາດຖານດຽວກັນ.
4. ສ້າງເອກະສານໂປຼແກຼມ.

11.ທົດສອບ (Testing)

ການທົດສອບເປັນຂັ້ນຕອນທີ່ສໍາຄັນຫຼາຍຈາກຂຽນໂປຼແກຼມເປັນທີ່ຮຽບຮ້ອຍແລ້ວຈະຕ້ອງມີການທົດສອບຫາຄວາມຜິດພາດ,ເປັນການແກ້ໄຂ ແລະ ປ້ອງກັນການຜິດພາດທີ່ອາດເກີດຂຶ້ນ ແລະ ຍັງເປັນການທົດສອບຄວາມພ້ອມຂອງໂປຼແກຼມເພື່ອຄວາມເຊື່ອຖືກັບໂປຼແກຼມທີ່ໄດ້ພັດທະນາຂຶ້ນ.

ສະຫຼຸບຂັ້ນຕອນການທົດສອບມີດັ່ງນີ້:

1. ທົດສອບການເກັບຂໍ້ມູນ.
2. ທົດສອບໄວຍາກອນພາສາຄອມພິວເຕີ.
3. ທົດສອບຄວາມຖືກຕ້ອງຂອງຜົນໄດ້ຮັບ.
4. ທົດສອບວ່າລະບົບຖືກຕາມຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ ຫຼື ບໍ່.

12. 5 ການສ້າງແບບຂໍ້ມູນ (Data Modeling)

ການສ້າງແບບຂໍ້ມູນ (Data Modeling) ຄື ການອອກແບບຖານຂໍ້ມູນ

ນັກວິເຄາະລະບົບຕ້ອງອອກແບບວ່າຈະເກັບຂໍ້ມູນແບບໃດ ແລະ ການດຶງຂໍ້ມູນມາໃຊ້ຈະໃຊ້ວິທີໃດ ການອອກແບບຖານຂໍ້ມູນໄດ້ ນັກວິເຄາະລະບົບຈະຕ້ອງຮູ້ຢ່າງຊັດເຈນແລ້ວວ່າ ຂໍ້ມູນທີ່ໃຊ້ທັງໝົດມີຫຍັງແຕ່ ຕົວຢ່າງ ຂໍ້ມູນອາດຈະເປັນແບບຕາຕະລາງທໍາມະດາ (Relational Database) ແລະ ການດຶງຂໍ້ມູນມາໃຊ້ ໂດຍມີ (Key) ເປັນຕົວໃຊ້ຄົ້ນຫາເປັນແບບ (Index File) ເປັນຕົ້ນ ເຊິ່ງສ່ວນໃຫຍ່ຂອງຖານຂໍ້ມູນໃນປັດຈຸບັນນີ້ໃຊ້ແບບ ຕາຕະລາງເພາະງ່າຍຕໍ່ການທໍາຄວາມເຂົ້າໃຈ.

13. 6 ການສ້າງແບບຈໍາລອງລະບົບ (System Modeling)

ການສ້າງແບບຈໍາລອງລະບົບ (System Modeling) ຄື

ການນໍາທຸກສິ່ງທຸກຢ່າງທີ່ໄດ້ຈາກວັດຈະນານຸກົມຂໍ້ມູນແຜນພາບສະແດງກະແສຂໍ້ມູນ ຖານຂໍ້ມູນ ມາລວມກັນເປັນລະບົບໃໝ່ ແລະ ທີ່ສໍາຄັນຄື ຄວາມຕ້ອງການໃໝ່ຂອງລະບົບ ຈະຖືກເພີ່ມເຕີມເຂົ້າມາໃນລະບົບໃໝ່ນີ້ ເຊິ່ງລະບົບໃໝ່ນີ້ຈະເປັນລະບົບທີ່ຕ້ອງການ ນອກຈາກນັ້ນຕ້ອງປະມວນວ່າ ຈະຕ້ອງໃຊ້ບຸກຄະລາກອນ ອຸປະກອນ ພັດສະດຸ ຫຍັງແດ່ ແລະ ໃຊ້ຈໍານວນເທົ່າໃດ.

14. 7 ແຜນຜັງລະບົບ (System Flowchart)

ແຜນຜັງລະບົບ ຄື: ຮູບພາບຫຼືສັນຍາລັກທີ່ໃຊ້ແທນລໍາດັບ ແທນຂັ້ນຕອນໃນໂປຼແກຣມຮູບພາບຫຼື

ສັນຍາລັກທີ່ໃຊ້ເປັນເອກະລັກ ເຊິ່ງການໃຊ້ແຜນຜັງລະບົບແມ່ນເພື່ອໃຫ້ເຮົາຮູ້ເຖິງຄວາມກ່ຽວພັນຂອງລະບົບ

ແຕ່ເລີ່ມຕົ້ນ ວ່າມີຂັ້ນຕອນການປະຕິບັດແນວໃດ, ໃຊ້ວິທີການໃດແດ່. ເໝາະສຳຫຼັບຜູ້ບໍລິຫານ, ຜູ້ວິເຄາະ ລະບົບ ແລະ ຜູ້ຂຽນໂປຣແກຣມ ຈະໄດ້ຮູ້ເຖິງຄວາມສຳພັນຂອງພະແນກຕ່າງໆ.

ປະເພດຂອງແຜນຜັງມີ 2 ປະເພດ ຄື:

ແຜນຜັງລະບົບ (System Flow chart).

ແຜນຜັງໂປຣແກຣມ (Program Flow chart).

15. 8 ແຜນພາບການໄຫຼຂອງກະແສຂໍ້ມູນ (DFD Data Flow Diagram)

ແມ່ນແຜນພາບທີ່ໃຊ້ອະທິບາຍການເຄື່ອນໄຫວຂອງຂໍ້ມູນໃນລະບົບເຊິ່ງຈະສະແດງເຖິງຂໍ້ມູນການໄຫຼເຂົ້າ ແລະ ໄຫຼອອກຂອງລະບົບໃນສ່ວນປະມວນຜົນຂອງຂໍ້ມູນ ແລະ ທີ່ເກັບຂໍ້ມູນແມ່ນພາບການໄຫຼຂອງກະແສຂໍ້ມູນເປັນເຕັກນິກທີ່ມີຄວາມສຳຄັນຂອງລະບົບການວິເຄາະເຊິ່ງຈະຊ່ວຍກຳນົດຂອບເຂດທີ່ຊັດເຈນຂອງລະບົບທີ່ນຳສະເໜີ, ຊ່ວຍໃນການສອບຂໍ້ມູນຕ່າງໆ ແລະ ກິດຈະກຳທີ່ລະບົບໃຊ້ໃນການຈັດເກັບຂໍ້ມູນຊ່ວຍໃນການກຳນົດໂປຼແກຼມທີ່ເຮັດໜ້າທີ່ຫຼັກໆຂອງລະບົບ. ແຜນພາບການໄຫຼຂອງກະແສຂໍ້ມູນປະກອບມີ 2 ຊະນິດຄື:

ແຜນພາບການໄຫຼຂອງກະແສຂໍ້ມູນແບບ Logical

ເປັນແບບຈຳລອງທີ່ອະທິບາຍການດຳເນີນງານໃນລະບົບວ່າມີການເຮັດວຽກ ແລະ ຄວາມຕ້ອງການຫຍັງແຕ່ໂດຍຄຳນຶງເຖິງເຕັກໂນໂລຊີທີ່ນຳມາໃຊ້ງານ.

ແຜນພາບການໄຫຼຂອງກະແສຂໍ້ມູນແບບ Physical

ເປັນແບບຈຳລອງທີ່ອະທິບາຍການດຳເນີນງານໃນລະບົບວ່າມີການເຮັດວຽກຫຍັງແດ່, ການດຳເນີນງານເປັນແນວໃດ, ນອກຈາກນີ້ຍັງສະແດງເຖິງປະສິດທິພາບຂອງເຕັກໂນໂລຊີທີ່ເລືອກມາຕິດຕັ້ງໃຊ້ງານເພື່ອຕອບສະໜອງຄວາມຕ້ອງການ ແລະ ສະແດງຂໍ້ຈຳກັດຂອງເຕັກໂນໂລຊີນັ້ນໆ.

ວັດຖຸປະສົງຂອງແຜນພາບການໄຫຼຂອງຂໍ້ມູນ

ເປັນແຜນພາບທີ່ສະຫຼຸບຂໍ້ມູນທັງໝົດທີ່ໄດ້ມາຈາກການວິເຄາະໃນລັກສະນະຂອງຮູບແບບທີ່ເປັນໂຄງສ້າງ.

ເປັນຂໍ້ຕົກລົງຮ່ວມກັນລະຫວ່າງນັກວິເຄາະລະບົບ ແລະ ຜູ້ໃຊ້ລະບົບ.

ເປັນແຜນພາບທີ່ໃຊ້ໃນການພັດທະນາຕໍ່ໃນຂັ້ນຕອນຂອງການອອກແບບ.

ເປັນແຜນພາບທີ່ໃຊ້ສຳຫຼັບການອ້າງອີງ ຫຼື ເພື່ອໃຊ້ໃນການພັດທະນາຕໍ່ໃນອະນາຄົດ.

ຮູ້ທີ່ໄປທີ່ມາຂອງຂໍ້ມູນທີ່ໄຫຼໃນຂະບວນການຕ່າງໆ.

16. 9 ຂັ້ນຕອນການແຕ້ມແຜນພາບການໄຫຼຂອງກະແສຂໍ້ມູນ

ຂັ້ນຕອນການແຕ້ມແຜນພາບການໄຫຼຂອງຂໍ້ມູນປະກອບດ້ວຍ 7 ຂັ້ນຕອນດັ່ງນີ້:

ວິເຄາະໃຫ້ໄດ້ວ່າລະບົບປະກອບດ້ວຍບຸນດາຣີ (Boundary) ໃດແດ່.

ດຳເນີນການອອກແບບໂດຍລວມຂອງລະບົບ (Contact Diagram).

ວິເຄາະຂໍ້ມູນໃນລະບົບວ່າຄວນມີຂໍ້ມູນໃດແດ່.

ວິເຄາະຂະບວນການປະມວນຜົນໃນລະບົບວ່າຄວນມີຂະບວນການປະມວນຜົນຫຼັກໆອັນໃດ ແລະ ປະກອບດ້ວຍຂະບວນການປະມວນຜົນຍ່ອຍອັນໃດໂດຍອາດຈະເຮັດເປັນແຜນພາບ ການວິເຄາະການໄຫຼຂອງຂໍ້ມູນ (Process Hierarchy Chart : PHC).


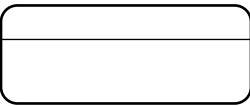

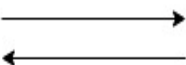
ດຳເນີນການຂຽນແຜນພາບການໄຫຼຂອງຂໍ້ມູນໃນລະບົບຕ່າງໆ.

ຕ້ອງມີການກວດສອບ ແລະ ປັບແກ້ຈົນໄດ້ແຜນພາບການໄຫຼຂອງຂໍ້ມູນທີ່ສົມບູນ.

ອາດໃຊ້ຊຸດເຄື່ອງມືມາຊ່ວຍໃນການແຕ້ມແຜນພາບການໄຫຼຂອງຂໍ້ມູນ.

17. ສັນຍາລັກທີ່ໃຊ້ໃນແຜນພາບການໄຫຼຂອງກະແສຂໍ້ມູນ

ໃນພາບການໄຫຼຂໍ້ມູນສະແດງເຖິງການໄຫຼຂອງຂໍ້ມູນເຂົ້າ ແລະ ຂໍ້ມູນອອກ ຂັ້ນຕອນການເຮັດວຽກງານຕ່າງໆ ຂອງລະບົບເຊິ່ງສັນຍາລັກຂອງແຜນພາບການໄຫຼຂໍ້ມູນ (DFD) ສະແດງດັ່ງລຸ່ມນີ້:

ຊື່	ສັນຍາລັກ	ຄວາມໝາຍ
Boundaries		ສັນຍາລັກບຸກຄົນ, ອົງກອນ ຫຼື ໜ່ວຍງານ
Process		ສັນຍາລັກການປະມວນຜົນ
Data Store		ສັນຍາລັກການເກັບຂໍ້ມູນ
Data Flow		ສັນຍາລັກເສັ້ນທາງການໄຫຼຂອງຂໍ້ມູນ

ຕາຕະລາງທີ 2.1 ສັນຍາລັກທີ່ໃຊ້ໃນການອອກ DFD (Data Flow Diagram)

18. ບຸນດາລີ (Boundaries)

ເປັນສັນຍາລັກທີ່ໃຊ້ແທນໄດ້ທັງຄົນ, ໜ່ວຍງານ ຫຼື ລະບົບງານເຊິ່ງສາມາດຮ້ອງຊື້ໄດ້ຫຼາຍຮູບແບບເຊັ່ນ: Source, Destination, External Agent ເປັນຕົ້ນ.

ໃນການພິຈາລະນາວ່າໃຜເປັນບຸນດາລີໃນລະບົບນັ້ນຈະພິຈາລະນາເຖິງບຸກຄົນທີ່ລະບົບບໍ່ສາມາດຄວບຄຸມໄດ້ເຊັ່ນ: ເປັນສິ່ງທີ່ຢູ່ພາຍນອກຂອບເຂດການປະມວນຜົນເຊັ່ນ: ຜູ້ຈັດການ ຫຼື ທີມງານເປັນຕົ້ນ. Boundaries ມັກຈະຖືກຈັດຢູ່ບໍລິເວນດ້ານນອກຂອງແຜນພາບເພື່ອໃຫ້ມີຄວາມສວຍງາມ ແລະ ງ່າຍຕໍ່ການເບິ່ງ ແລະ ບຸນດາລີສາມາດເຮັດການກະທຳຊ້ຳໄດ້ເຊິ່ງການກະທຳຊ້ຳນັ້ນຈະຕ້ອງໃຊ້ເຄື່ອງໝາຍ (Back Slash) ໄວ້ທີ່ມຸມລຸ່ມດ້ານຊ້າຍດັ່ງຮູບຕົວຢ່າງລຸ່ມນີ້:

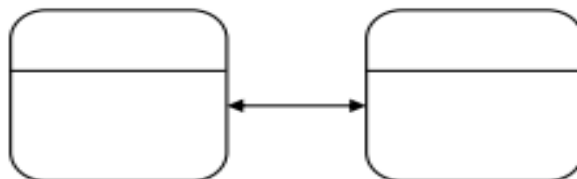


ຮູບທີ 2.2 (Boundaries) ແລະ (Duplicate Boundaries)

19. ໂປຣເຊດ (Process)

ແມ່ນຂະບວນການທີ່ຕ້ອງໄດ້ເຮັດໃນລະບົບ ໂດຍຈະເປັນຄຳກິລິຍາເຊັ່ນ: ກວດສອບ, ຈ່າຍເງິນ, ພິມລາຍງານເປັນຕົ້ນ. ແຕ່ລະໂປຣເຊດຈະຕ້ອງມີໝາຍເລກກຳກັບ ແລະ

ຄຳກິລິຍາອະທິບາຍການເຮັດວຽກ, ໝາຍເລກໂປຣເຊດດັ່ງກ່າວແມ່ນບໍ່ໄດ້ໝາຍເຖິງການເຮັດວຽກຕາມລຳດັບຂອງໂປຣເຊດ ແລະ ທີ່ສຳຄັນໝາຍເລກໂປຣເຊດບໍ່ສາມາດຊ້ຳກັນໄດ້ໃນຂະບວນການໂປຣເຊດຈະມີພຽງຂໍ້ມູນໄຫຼເຂົ້າ ຫຼື ໄຫຼອອກຢ່າງດຽວ.



ຮູບທີ 2. 2 ຕົວຢ່າງໂປຣເຊດ

20. ດາຕ້າສະຕໍ (Data Store)

ເປັນແຫຼ່ງທີ່ໃຊ້ເກັບຂໍ້ມູນເຊັ່ນ: ຂໍ້ມູນນັກສຶກສາ, ຂໍ້ມູນບຸກຄະລາກອນ. ສັນຍາລັກທີ່ມີເລກປະຈຳຂໍ້ມູນລະບຸໄດ້ເຊັ່ນ: Data Store ຕ້ອງມີຊື່ຂໍ້ມູນ ແລະ ມີການກຳນົດເລກກຳນົດຕາມລຳດັບເຊັ່ນ: D1, D2, D3 ເປັນຕົ້ນ. Data Store ສາມາດເຮັດຊຳກັນໄດ້ ແລະ ການເຄື່ອນຍ້າຍຂໍ້ມູນຈະຕ້ອງຜ່ານໂປຣເຊດສະເໝີ.



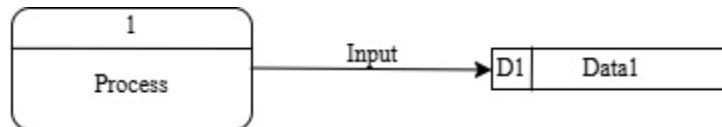
ຮູບທີ 2. 3 ດາຕ້າສະຕໍ

ລູກສອນຂອງດາຕ້າໄຟທີ່ໃຊ້ເຊື່ອມຕໍ່ລະຫວ່າງດາຕ້າສະຕໍກັບໂປຣເຊດສ໌ຈະມີຄວາມໝາຍດັ່ງນີ້:

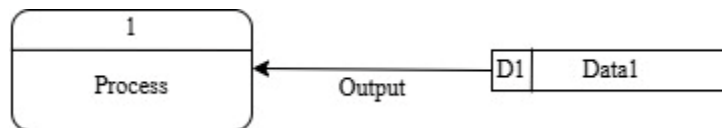
ລູກສອນຈາກ Data Store ຊື່ໄປຫາ Process ໝາຍເຖິງ Input

ລູກສອນຊື່ຈາກ Process ຊື່ໄປຫາດາຕ້າສະຕໍໝາຍເຖິງ Output

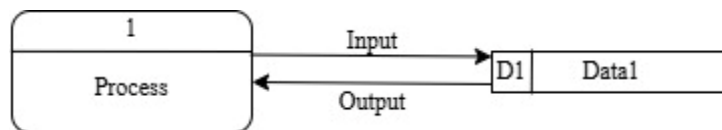
ລູກສອນສອງທາງລະຫວ່າງ Process ແລະ Data Store ໝາຍເຖິງ Input / Output



ຮູບທີ 2. 4 ສະແດງໂປຣເຊດສ໌ທີ່ມີການຈັດການຂໍ້ມູນໃນ Data Store ແບບ Input



ຮູບທີ 2. 5 ສະແດງໂປຣເຊດສ໌ທີ່ມີການ Output ຫຼື ເພີ່ມຂໍ້ມູນເຂົ້າໄປຍັງດາຕ້າສະຕໍ



ຮູບທີ 2. 6 ສະແດງໂປຣເຊດສ໌ທີ່ມີການຈັດການຂໍ້ມູນໃນດາຕ້າສະຕໍແບບ Input ແລະ Output

21. ດາຕ້າໄຟຣ (Data Flow)

ເປັນເສັ້ນທາງການໄຫຼຂອງຂໍ້ມູນ ຫຼື

ສາມາດເວົ້າອີກຢ່າງໜຶ່ງເປັນສັນຍາລັກໃນການບອກທິດທາງຂອງການເຄື່ອນທີ່ຂອງຂໍ້ມູນ.

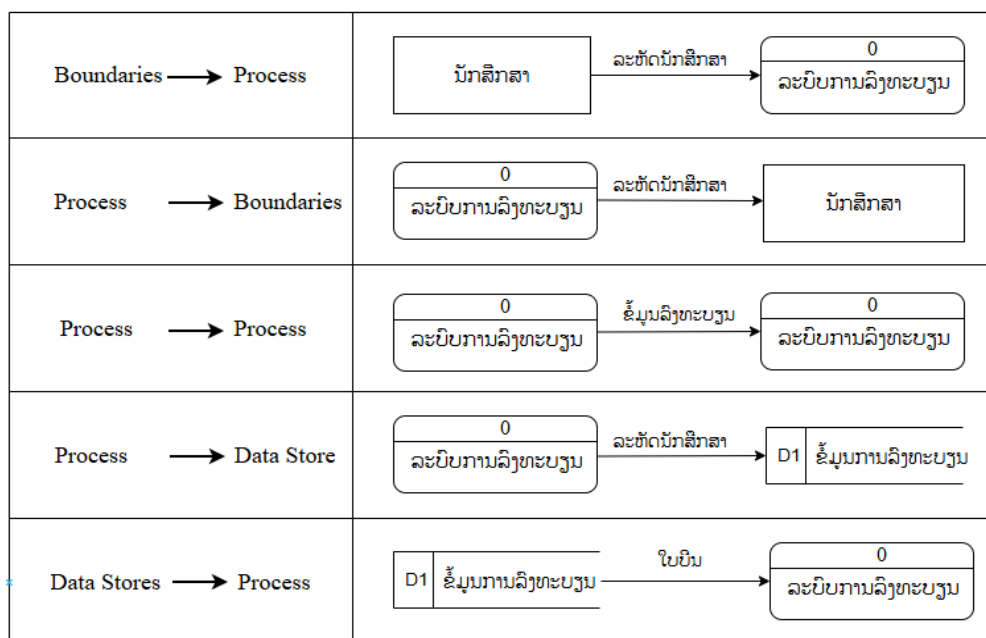


ຮູບທີ 2. 7 ເສັ້ນທາງການໄຫຼຂອງກະແສຂໍ້ມູນ

22. ຂັ້ນຕອນການແຕ້ມແຜນພາບກະແສຂໍ້ມູນ

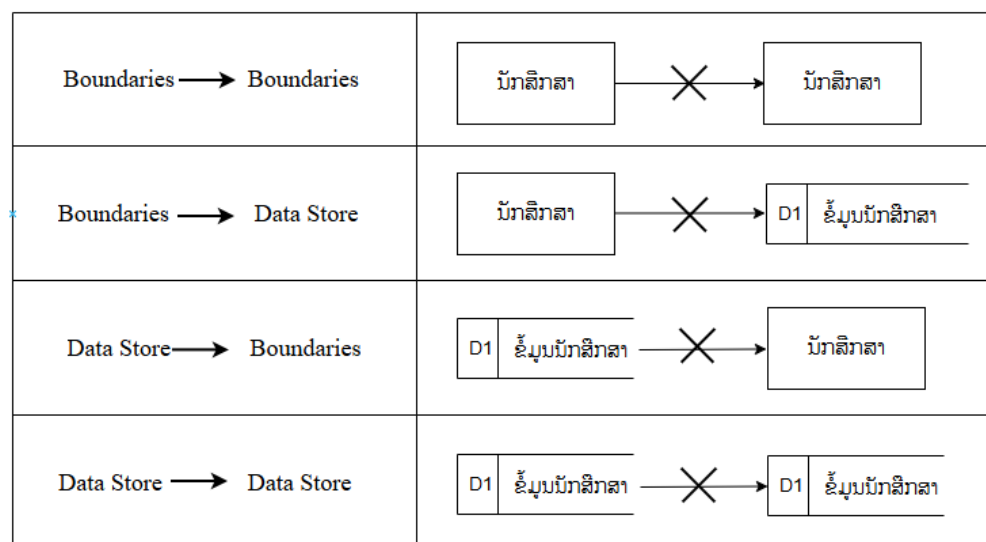
ການໄຫຼຂອງກະແສຂໍ້ມູນປະກອບໄປດ້ວຍສັນຍາລັກຕ່າງໆເຊິ່ງແຕ່ລະສັນຍາລັກລ້ວນແຕ່ມີຄວາມໝາຍໃນຕົວເອງ ດັ່ງນັ້ນ, ການອອກແບບແຜນພາບກະແສຂໍ້ມູນຈຶ່ງຕ້ອງມີກົດເກນຕ່າງໆດັ່ງນີ້:

- 1). ຊື່ຕາຕ້າໄຟ (Data Flow) ຄວນເປັນຊື່ຂໍ້ມູນທີ່ສົ່ງໂດຍບໍ່ຕ້ອງອະທິບາຍວ່າສົ່ງແບບໃດ, ເຮັດ ວຽກແນວໃດ
- 2). ຕາຕ້າໄຟ (Data Flows) ຕ້ອງມີຈຸດເລີ່ມຕົ້ນ ຫຼື ສິ້ນສຸດທີ່ໂປຣເຊດ (Process) ເພາະຕາຕ້າໄຟຮ (Data Flows) ຄືຂໍ້ມູນທີ່ນຳເຂົ້າ ແລະ ຂໍ້ມູນທີ່ສົ່ງອອກຂອງໂປຣເຊດ (Process).
- 3). Data Flows ຈະເດີນທາງລະຫວ່າງ External Agent ກັບ External Agent ບໍ່ໄດ້.
- 4). Data Flows ຈະເດີນທາງລະຫວ່າງ External Agent ໄປ Data Store ບໍ່ໄດ້.
- 5). Data Flows ຈະເດີນທາງລະຫວ່າງ Data Store ກັບ External Agent ບໍ່ໄດ້.
- 6). Data Flows ຈະເດີນທາງລະຫວ່າງ Data Store ກັບ Data Store ບໍ່ໄດ້.
- 7). ການຕັ້ງຊື່ Data Flows ຈະຕ້ອງໃຊ້ຄຳນາມ.
 - ແຜນພາບທີ່ຖືກຕ້ອງ



ຮູບທີ 2. 8 ຮູບພາບສະແດງແຜນພາບທີ່ຖືກຕ້ອງ

- ແຜນພາບທີ່ບໍ່ຖືກຕ້ອງ



ຮູບທີ 2. 9 ຮູບພາບສະແດງແຜນພາບທີ່ບໍ່ຖືກຕ້ອງ

23. ຂັ້ນຕອນໃນການຂຽນແຜນພາບກະແສຂໍ້ມູນ

ໃນການຂຽນແຜນພາບການໄຫຼກະແສຂອງຂໍ້ມູນນອກຈາກຜູ້ຂຽນຈະຮູ້ສັນຍາລັກຕ່າງໆທີ່ໃຊ້ງານລວມທັງຫຼັກການທີ່ໃຊ້ງານຂອງສັນຍາລັກຕ່າງໆແລ້ວການຂຽນແຜນພາບການໄຫຼກະແສຂອງຂໍ້ມູນມີສ່ວນປະກອບອື່ນໆທີ່ກ່ຽວຂ້ອງອີກດັ່ງນີ້:

ວິເຄາະວ່າລະບົບປະກອບດ້ວຍ External Agent ຫຼື ບຸກຄົນໜ່ວຍງານໃດແດ່ທີ່ກ່ຽວຂ້ອງກັບລະບົບ.

ດຳເນີນການອອກແບບໃນລະບົບ Context Diagram.

ວິເຄາະຂໍ້ມູນໃນລະບົບວ່າຄວນມີຂໍ້ມູນໃດແດ່.

ວິເຄາະຂະບວນການ ຫຼື Process ໃນລະບົບວ່າຕົວໃດເປັນຕົວຫຼັກ ແລະ ປະກອບດ້ວຍໂປຣເຊດ(Process) ຍ່ອຍຫຍິບແດ່? ໂດຍອາດຈະສ້າງເປັນ Process Hierarchy Chart ທີ່ສະແດງເຖິງ Process ຫຼັກ ແລະ Process ຍ່ອຍ ໃນລະດັບຕ່າງໆ.

ດຳເນີນການຂຽນແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນໃນລະບົບຕ່າງໆ.

ທຳການກວດສອບ (Balancing) ແລະ ປັບແກ້ (Redraw) ຈົນໄດ້ແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນທີ່ສົມບູນ ແລະ ຖືກຕ້ອງ.

ອາດໃຊ້ Case Tool ເຂົ້າມາຊ່ວຍ.

12.1 ວັດຖຸປະສົງຂອງ DFD

ເປັນແຜນພາບທີ່ສະຫຼຸບລວມທັງໝົດທີ່ໄດ້ຈາກການວິເຄາະໃນລັກສະນະຂອງຮູບແບບໂຄງສ້າງ.

ເປັນຂໍ້ຕົກລົງຮ່ວມກັນລະຫວ່າງການວິເຄາະລະບົບ ແລະ ຜູ້ຊົມໃຊ້.

ເປັນແຜນພາບທີ່ໃຊ້ໃນການພັດທະນາຕໍ່ຫຼັງຂັ້ນຕອນການອອກແບບ.

ເປັນແຜນພາບທີ່ໃຊ້ໃນການອ້າງອີງ ຫຼື ໃຊ້ໃນການພັດທະນາລະບົບຕໍ່ໃນອານາຄົດ.

ເຮັດໃຫ້ຮູ້ເຖິງທີ່ໄປທີ່ມາຂອງຂໍ້ມູນທີ່ໄຫຼໃນຂະບວນການຕ່າງໆ.

12.2 ຜົນປະໂຫຍດຂອງ DFD

ຊ່ວຍໃຫ້ການວິເຄາະລະບົບມີຄວາມສະດວກໂດຍສາມາດເຫັນຂັ້ນຕອນຕ່າງໆຂອງໂປຣແກຣມ.

ເປັນສິ່ງທີ່ຊ່ວຍໃນການວິເຄາະລະບົບໃຫ້ເປັນໄປໄດ້ງ່າຍ ແລະ
ມີຄວາມເຂົ້າໃຈກັນລະຫວ່າງຜູ້ວິເຄາະ, ລະບົບ, ລະຫວ່າງຜູ້ວິເຄາະລະບົບກັບໂປຣແກຣມເມີ ແລະ
ລະຫວ່າງຜູ້ວິເຄາະລະບົບກັບຜູ້ໃຊ້.

ມີຄວາມເປັນອິດສະຫຼະໃນການໃຊ້ງານ, ໂດຍບໍ່ຕ້ອງມີເທັກນິກອື່ນມາຊ່ວຍ,
ເນື່ອງຈາກສາມາດໃຊ້ສັນຍາລັກຕ່າງໆແທນສິ່ງທີ່ວິເຄາະມາ.

ເປັນສິ່ງທີ່ງ່າຍຕໍ່ການສະແດງຄວາມສໍາພັນລະຫວ່າງລະບົບໃຫຍ່ ແລະ ລະບົບຍ່ອຍ
ເຊິ່ງຈະພາໃຫ້ມີຄວາມເຂົ້າໃຈຄວາມສໍາພັນຕ່າງໆໄດ້ດີ.

12.3 ດາຕ້າອີເລເມັນ (Data Element)

ເປັນສ່ວນທີ່ນ້ອຍທີ່ສຸດຂອງຂໍ້ມູນມີຄວາມໝາຍໃນລະບົບເຊັ່ນ:
ການລົງທະບຽນນັກສຶກສາປະກອບດ້ວຍດາຕ້າອີເລເມັນຫຼາຍອັນເຊັ່ນ: ລະຫັດນັກສຶກສາ, ຊື່ນັກສຶກສາ, ລະຫັດຫ້ອງ,
ລະດັບສາຍວິຊາ ແລະ ອື່ນໆດາຕ້າອີເລເມັນແຕ່ລະຕົວຈະມີລັກສະນະສະເພາະເຊັ່ນ:

ຊື່ຂອງຂໍ້ມູນ (Data Name)

ຖືກນໍາມາໃຊ້ເພື່ອແຍກຂໍ້ມູນອອກຈາກຂໍ້ມູນຕົວອື່ນ
ຊື່ຂໍ້ມູນຍ່ອຍແຕ່ລະຕົວຈະມີຊື່ຂອງໃຜລາວເພື່ອໃຊ້ອ້າງອີງອົງຕະຫລອດຂະບວນການພັດທະນາລະບົບ.
ດັ່ງນັ້ນການໃສ່ໃຫ້ມີຄວາມໝາຍເໝາະສົມ, ເຂົ້າໃຈງ່າຍ ແລ
ເປັນມາດຕະຖານທົ່ວໄປໃນການຕັ້ງຊື່ຂໍ້ມູນນັ້ນຈະແມ່ນບໍ່ໃຫ້ມີຕົວອັກສອນຫຼາຍກວ່າ 30
ຕົວ(ເຊິ່ງປະກອບດ້ວຍຕົວອັກສອນພາສາອັງກິດເປັນຕົວຢ່າຍໃຫຍ່ ຫຼື ນ້ອຍເຊັ່ນ: A-Z ຫຼື a-z, ສ່ວນຕົວເລກແມ່ນ 0-9
ບໍ່ໃຫ້ມີຍະຫວ່າງລະຫວ່າງຄໍາ ຫາກຈະໃຊ້ຕ້ອງໃຊ້ຂີດກ້ອງ “_” ຫຼື “-” ແທນ).

ຊະນິດຂໍ້ມູນ (Type)

ບາງຂັ້ນຕອນການປະມວນຜົນຈະເກີດຂຶ້ນໄດ້ກໍຕໍ່ເມື່ອຂໍ້ມູນທີ່ອິນພຸດເຂົ້າໄປມີຄ່າ ຫຼື
ຊະນິດຂໍ້ມູນຕາມທີ່ກໍານົດເທົ່ານັ້ນເຊັ່ນ: ຄ່າຂໍ້ມູນຕົວເລກຈະຮັບໄດ້ພຽງຕົວເລກເພື່ອນໍາໄປປະມວນຜົນ
ຖ້ານໍາຕົວອັກສອນ ໃສ່ເຂົ້າໄປກໍຈະສາມາດປະມວນຜົນໄດ້. ສະນັ້ນເຮົາໃສ່ຂໍ້ມູນທີ່ຖືກຕ້ອງຕາມຊະນິດຂໍ້ມູນທີ່ກໍານົດ
ເພື່ອຫຍໍ້ໃຫ້ການປະມວນຜົນເກີດການຜິດພາດ ຫຼື ບໍ່ສາມາດປະມວນຜົນໄດ້.

ຄວາມຍາວ ຫຼື ຂະໜາດຂອງຂໍ້ມູນ (Data Size)

ໝາຍເຖິງຂະໜາດ ຫຼື ການກໍານົດຄວາມຍາວຂໍ້ມູນເຊັ່ນ: ກໍານົດຄວາມຍາວຂອງຕົວອັກສອນ ຫຼື
ສັນຍາລັກຕ່າງໆວ່າໃຫ້ມີຂະໜາດ ຫຼື ຄວາມຍາວເທົ່າໃດ.

ຄໍາອະທິບາຍຂໍ້ມູນ (Data Description)

ເປັນຄຳອະທິບາຍຂໍ້ມູນທີ່ຢູ່ໃນແຕ່ລະຝັ່ງ (Field) ເພື່ອໃຫ້ຜູ້ພັດທະນາລະບົບ ຫຼື ບຸກຄົນໃນລະບົບສາມາດເຂົ້າໃຈຄວາມໝາຍຂອງຊື່ທີ່ຢູ່ໃນຝັ່ງນັ້ນໆ.

ຕົວຢ່າງ:

Attribute	Type	Size	Description
Std_ID	Ntext	6	ລະຫັດນັກຮຽນ
Std_Name	Ntext	50	ຊື່
Std_Surname	Ntext	50	ນາມສະກຸນ
Birthday	Int	20	ວັນເດືອນປີເກີດ

ຕາຕະລາງທີ 2.2
ຕົວຢ່າງ
ຄຳຕຳອີງ
ລະຫັດ
(Data

Element)

12.4 ຄວາມໝາຍຂອງ DFD Fragments

ຈະເປັນການລວບລວມຂອງ Process ຫຼັກ ແລະ ຂໍ້ມູນຫຼັກໆທີ່ກ່ຽວຂ້ອງໂດຍເລີ່ມຕົ້ນດ້ວຍການແຕ້ມແຜນພາບກະແສຂໍ້ມູນລະດັບທີ 1 ຂອງແຕ່ລະຂະບວນການເອີ້ນວ່າ DFD Fragments ເພື່ອສະແດງເຫດການ. ສະຫຼຸບລວມແລ້ວ DFD Fragments ຄືການນຳສະເໜີແຜນພາບກະແສຂໍ້ມູນໃນລະບົບການດ້ວຍການນຳສະເໜີເຫດການຂອງ Process ນັ້ນໆ.

- ຄວາມສຳພັນລະຫວ່າງເອັນທີຕີ້ (Entity Relationship)
- ຄວາມສຳພັນແບບໜຶ່ງຕໍ່ໜຶ່ງ (One to One)

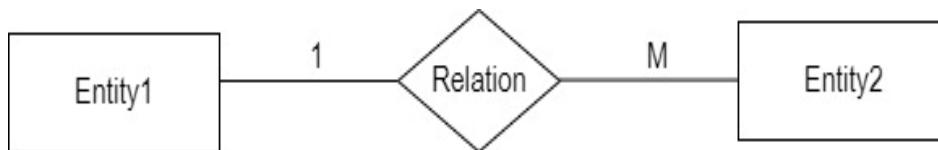
ໝາຍຄວາມວ່າຄືຫຼັກໃນເອັນທີຕີ້ໜຶ່ງມີຄວາມສຳພັນກັບອີກຄືໃນເອັນທີຕີ້ອື່ນໄດ້ຄ່າດຽວເທົ່ານັ້ນ.



ຮູບທີ 2. 10 ຄວາມສຳພັນແບບໜຶ່ງຕໍ່ໜຶ່ງ

- ຄວາມສຳພັນແບບໜຶ່ງຕໍ່ຫຼາຍ (One to Many)

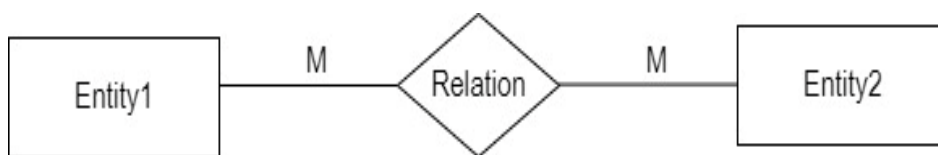
ໝາຍຄວາມວ່າຄືຫຼັກໃນເອັນທີຕີ້ໜຶ່ງມີຄວາມສຳພັນກັບອີກຄືໃນເອັນທີຕີ້ອື່ນໄດ້ຫຼາຍຄ່າ.



ຮູບທີ 2. 11 ຄວາມສຳພັນແບບໜຶ່ງຕໍ່ຫຼາຍ

- ຄວາມສຳພັນແບບຫຼາຍຕໍ່ຫຼາຍ (Many to Many)

ໝາຍຄວາມວ່າຄືຫຼັກໃນເອັນທີທີ່ໜຶ່ງມີຫຼາຍຄ່າມີຄວາມສຳພັນກັບອີກຄືໃນເອັນທີທີ່ອື່ນໄດ້ຫຼາຍຄ່າ.



ຮູບທີ 2. 12 ຄວາມສຳພັນແບບຫຼາຍຕໍ່ຫຼາຍ

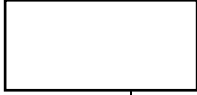
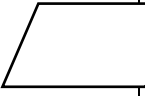
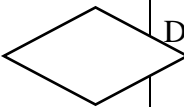
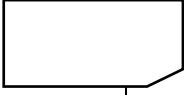

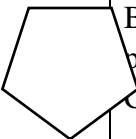
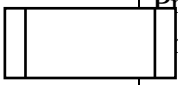

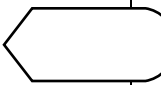
12.6 ແຜນຜັງໂຄງສ້າງ (Flowchart)

ຜັງງານ (Flowchart) ແມ່ນພາບທີ່ມີການໃຊ້ສັນຍາລັກຮູບພາບ ແລະ

ລູກສອນທີ່ສະແດງເຖິງຂັ້ນຕອນການເຮັດວຽກຂອງໂປຣແກຣມ ຫຼື ລະບົບເທື່ອລະຂັ້ນຕອນ

ລວມທັງການໄຫຼຂອງຂໍ້ມູນຕັ້ງແຕ່ທຳອິດຈົນໄດ້ຮັບຜົນຕາມຄວາມຕ້ອງການ.ສັນຍາລັກຂອງການຂຽນຜັງງານຈະປະກອບດ້ວຍການໃຊ້ສັນຍາລັກຕ່າງໆ(ANSI: American National Institute) ດັ່ງນີ້:

ຮູບສັນຍາລັກ	ຊື່ເອີ້ນ	ຄວາມໝາຍຂອງສັນຍາລັກ
	Terminal	ຈຸດເລີ່ມຕົ້ນ, ສິ້ນສຸດການເຮັດວຽກຂອງໂປຣແກຣມ
	Flow Line/ Direction	ລູກສອນສະແດງທິດທາງການໄຫຼຂໍ້ມູນ

 Process	ໃຊ້ສະແດງຄໍາສັ່ງໃນການປະມວນຜົນ ຫຼື ການກຳນົດຄ່າຂໍ້ມູນໃຫ້ກັບຕົວປ່ຽນ
 Input/Output	ການຮັບ ຫລື ສະແດງຂໍ້ມູນໂດຍບໍ່ລະບຸຊະນິດຂອງອຸປະກອນ
 Decision	ການປະມວນຜົນ ຫຼື ການກວດສອບເງື່ອນໄຂເພື່ອເລືອກເຮັດອັນໃດອັນໜຶ່ງ
 Ter	ເອກະສານ, ສະແດງຜົນການສະແດງຜົນທາງແປ້ນພິມ
 In-page connector	ສະແດງຈຸດເຊື່ອມຕໍ່ຂອງແຜນຜັງງານ
 Between-page connector	ການຂ້າມຫນ້າໃໝ່ໃຊ້ໃນກໍລະນີຜັງງານມີຄວາມຍາວເກີນທີ່ຈະສະແດງໃນໜຶ່ງຫນ້າ
 Predefine process	ຂະບວນການນິຍາມໄວ້ການເຮັດວຽກຢ່ອຍ
 Keyboard	ຮັບອ່ານຂໍ້ມູນທີ່ຮັບຈາກແປ້ນພິມ
 Monitor	ສະແດງລາຍລະອຽດຂອງຂໍ້ມູນທາງຈໍພາບ

ຕາຕະລາງທີ 2.3 ສັນຍາລັກແຜນຜັງໂຄງສ້າງ

12.7 ການອອກແບບ ແລະ ຈຳລອງ ອີອາຣ (ER Entity Relationship)

ອີອາຣ (ER) ເປັນການອອກແບບສ້າງຕາຕະລາງຖານຂໍ້ມູນດ້ວຍແບບຈຳລອງອີອາຣເຊິ່ງແບ່ງອອກເປັນ 2

ຂັ້ນຕອນຫຼັກຄື:

ຂັ້ນຕອນທຳອິດເປັນສ້າງຕາຕະລາງແບບຈຳລອງອີອາຣຂັ້ນຕອນນີ້ຈະກ່າວເຖິງຄວາມໝາຍຂອງສັນຍາລັກຕ່າງໆທີ່ໃຊ້ງານແບບຈຳລອງອີອາຣເພື່ອຈະເຮັດໃຫ້ສາມາດສ້າງແບບຈຳລອງອີອາຣຂັ້ນໃຊ້ງານໄດ້ຢ່າງຖືກຕ້ອງ ແລະ ມີປະສິດທິພາບ.

ຂັ້ນຕອນທີ 2

ເປັນການແປງແບບຈຳລອງອີອາຣໃຫ້ເປັນຕາຕະລາງຂໍ້ມູນເພື່ອໃຊ້ງານກັບຕົວຈັດການຖານຂໍ້ມູນເພື່ອໃຊ້ງານກັບຕົວຈັດການຖານຂໍ້ມູນແບບລິເລຊັນ.

ສັນຍາລັກທີ່ໃຊ້ໃນແບບຈຳລອງອີອາຣ

ການອອກແບບຖານຂໍ້ມູນດ້ວຍແບບຈຳລອງອີອາຣນັ້ນ ກ່ອນອື່ນຕ້ອງຮູ້ຈັກຄວາມໝາຍ ແລະ ສັນຍາລັກຕ່າງໆທີ່ໃຊ້ໃນແບບຈຳລອງອີອາຣ ເພື່ອສາມາດສ້າງແບບຈຳລອງອີອາຣໄດ້ຢ່າງຖືກຕ້ອງ.

ເອັນທີຕີ້ (Entity)

ຄືວັດສະດຸສິ່ງຂອງເຊິ່ງເປັນສ່ວນທີ່ໃຊ້ສຳລັບເກັບຂໍ້ມູນແຕ່ລະລາຍການພາຍໃນລະບົບທີ່ເຮົາເຮັດຢູ່ເຊັ່ນ: ບຸກຄົນ (ພະນັກງານ, ນັກສຶກສາ, ລູກຄ້າ), ສະຖານທີ່ (ບ້ານ, ເມືອງ, ແຂວງ, ທີ່ຢູ່), ວັດຖຸ (ລົດ, ເຮືອນ, ສິນຄ້າ) ແລະ ເຫດການ (ປະຫວັດສາດ, ການລົງທະບຽນຊື່, ຊື່, ຂາຍ) ສັນຍາລັກແທນດ້ວຍຮູບ4ຫຼ່ຽມ ເຊິ່ງ ເອັນທີຕີ້ໃຊ້ໃນແບບຈຳລອງອີອາຣມີຢູ່ 2 ຊະນິດຄື:

ເອັນທີຕີ້ປົກກະຕິ

ເປັນສັນຍາລັກສີ່ຫຼ່ຽມໂດຍພາຍໃນຈະມີຕົວໜັງສືບິ່ງບອກຊື່ຂອງເອັນທີຕີ້ນັ້ນ ເຊິ່ງໂດຍທົ່ວໄປການຕັ້ງຊື່ເອັນທີຕີ້ຈະຕ້ອງມີຊື່ຄວາມໝາຍໃຫ້ຊັດເຈນໃນສິ່ງທີ່ເຮົາສົນໃຈ ແລະ ຕ້ອງເປັນຊື່ທີ່ສາມາດໃຊ້ເປັນຕາຕະລາງຖານຂໍ້ມູນໄດ້ ສຳຫຼັບເອັນທີຕີ້ແບບນີ້ຈະໃຊ້ເປັນສັນຍາລັກແທນອົງກອນ, ໜ່ວຍງານ ຫຼື ໂຄງສ້າງງານ.

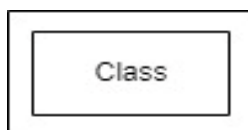
ຕົວຢ່າງ:



ຮູບທີ 2.14 ເອັນທີຕີ້ປົກກະຕິ

ເອັນທີຕີ້ຊະນິດອ່ອນ

ເປັນສັນຍາລັກເປັນຮູບສີ່ຫຼ່ຽມຊ້ອນກັນສອງຮູບໂດຍທີ່ພາຍໃນເປັນຕົວໜັງສືບິ່ງບອກເຖິງຊື່ສອງເອັນທີຕີ້ຊະນິດອ່ອນ. ເອັນທີຕີ້ຊະນິດນີ້ເປັນເອັນທີຕີ້ທີ່ບໍ່ສາມາດເປັນຄືຫຼັກເປັນຂອງຕົວເອງ.



ຮູບທີ 2. 13 ເອັນທີຕີ້ແບບອ່ອນ.

ຄວາມສຳພັນ (Relationship)

ຫມາຍເຖິງເອັນທີຕີ້ (Entity) ທີ່ສະແດງເຖິງຄວາມສຳພັນລະຫວ່າງ 2 ເອັນທີຕີ້ (Entity) ຂຶ້ນໄປ ແລະ ມີຄວາມສຳພັນລະຫວ່າງ Entity ທີ່ມີ Attribute ຮ່ວມກັນ ສັນຍາລັກດ້ວຍຮູບດອກຈັນ.

ຄວາມສຳພັນທີ່ໃຊ້ໃນແບບຈຳລອງອີອາຣມີຢູ່ 2 ຊະນິດຄື:

ຄວາມສຳພັນປົກກະຕິ

ສັນຍາລັກຄວາມສຳພັນຈະສະແດງດ້ວຍຮູບດອກຈັນ ແລະ ມີຕົວອັກສອນຢູ່ພາຍໃນເພື່ອບອກຊື່ຂອງຄວາມສຳພັນນັ້ນ ການຕັ້ງຊື່ຄວາມສຳພັນຄວນມີຄວາມຫມາຍຊັດເຈນວ່າ ແຕ່ລະເອັນທີຕີ້ມີຄວາມສຳພັນກັນ ໝາຍເຖິງການພົວພັນລະຫວ່າງ ເອັນທີຕີ້ ກັບ ເອັນທີຕີ້ ເຊິ່ງມີຄວາມສຳພັນເປັນຕົວກາງ.



ຮູບທີ 2. 14 ຄວາມສຳພັນແບບປົກກະຕິ

ຄວາມສຳພັນຊະນິດອ່ອນ

ສັນຍາລັກຂອງຄວາມສຳພັນຊະນິດອ່ອນເປັນຮູບດອກຈັນຊ້ອນກັນຫຼາຍອັນໂດຍ ຕ້ອງມີຕຸ້ຍບອກວ່າເປັນຄວາມສຳພັນຊະນິດໃດນຳກັນ ໝາຍເຖິງການເຊື່ອມຕໍ່ລະຫວ່າງ ເອັນທີຕີ້ປົກກະຕິ ກັບ ເອັນທີຕີ້ຊະນິດອ່ອນ.

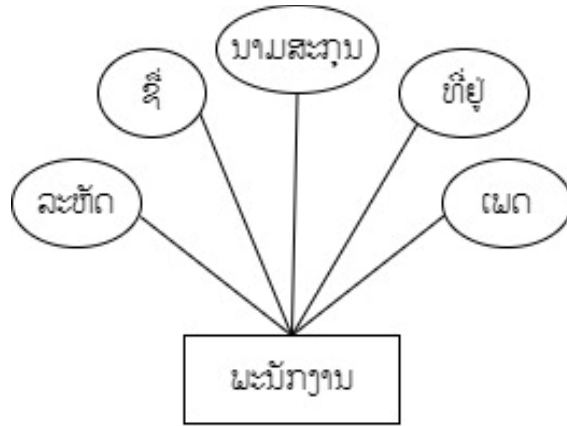


ຮູບທີ 2. 15 ຄວາມສຳພັນແບບອ່ອນ.

ແອັດທິບິວ (Attribute)

ໃຊ້ສະແດງຄຸນສົມບັດຂອງເອັນທີຕີ້ (Entity) ເຊັ່ນ: ຊື່, ນາມສະກຸນ, ທີ່ຢູ່, ເລກທີ, ເພດ, ເງິນເດືອນ ເຊິ່ງສັນຍາລັກດ້ວຍຮູບວົງຮີ ໂດຍມີຊື່ຂອງແອັດທິບິວ (Attribute) ກຳກັບຢູ່ພາຍໃນ ແລະ ມີເສັ້ນເຊື່ອມຕໍ່ກັບເອັນທີຕີ້ ຂອງມັນ.

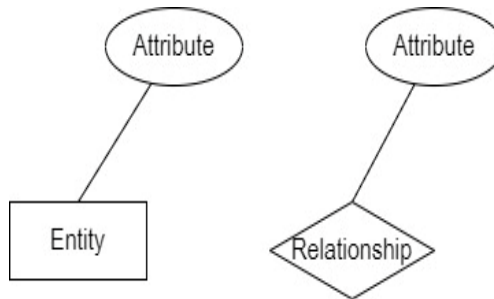
ຕົວຢ່າງ: ເປັນຂໍ້ມູນທີ່ສະແດງລັກສະນະຂອງເອັນທີຕີ້ເຊັ່ນ: ແອັດທິບິວຂອງເອັນທີຕີ້ພະນັກງານຈະມີລະຫັດພະນັກງານ, ຊື່, ທີ່ຢູ່ ເປັນຕົ້ນ.



ຮູບທີ 2. 16 ແອັດທິບົວ.

ແອັດທິບົວຊະນິດປົກກະຕິ.

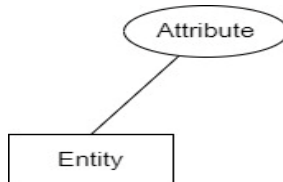
ເປັນສັນຍະລັກຮູບວົງມົນໂດຍພາຍໃນຈະເປັນຕົວໜັງສື ບອກເຖິງຊື່ຂອງແອັດທິບົວນັ້ນ ເຊິ່ງການຕັ້ງຊື່ຕ້ອງມີຄວາມຫມາຍຊັດເຈນຂອງຄຸນສົມບັດຂອງເອັນທີຕີ້.



ຮູບທີ 2. 17 ແອັດທິບົວຊະນິດປົກກະຕິ.

ແອັດທິບົວຊະນິດກະແຈ.

ລັກສະນະເປັນຮູບວົງມົນໂດຍພາຍໃນຈະເປັນຕົວໜັງສືບອກເຖິງຊື່ຂອງແອັດທິບົວ ແລະ ມີຕົວໜັງສືຂີດກ້ອງ,



ຮູບທີ 2. 18 ແອັດທິບົວຊະນິດກະແຈ.

ເປັນແອັດທິບິວທີ່ມີຄວາມຫມາຍພິເສດເປັນຂໍ້ມູນທີ່ບໍ່ຊ້ຳກັນ ແລະ ບໍ່ມີຄ່າສູນ.

II. ຄວາມຮູ້ທົ່ວໄປກ່ຽວກັບ Next.js

ພາສາໂປຣແກຣມຄອມພິວເຕີ (Computer Programming Language)

ພາສາໂປຣແກຣມຄອມພິວເຕີ (Computer Programming Language) ຊຸດຄໍາສັ່ງທີ່ນັກ ຂຽນ ໂປຣແກຣມ ຫຼື ໂປຣແກຣມເມີ (Programmer) ຂຽນໂປຣແກຣມທີ່ຖືກສ້າງໄວ້ຕາມຫຼັກໄວຍະກອນ

ຂອງພາສາໂປຣແກຣມຄອມພິວເຕີເພື່ອໃຫ້ສາມາດຕິດຕໍ່ສື່ສານ, ຄວບຄຸມການຮັບສິ່ງຂໍ້ມູນ ແລະ ສັ່ງໃຫ້ ຄອມພິວເຕີເຮັດວຽກຕາມທີ່ນັກຂຽນໂປຣແກຣມກຳນົດ.

ເຊິ່ງພາສາໂປຣແກຣມຄອມພິວເຕີໃນປະຈຸບັນນັ້ນມີຫຼາຍພາສາໃຫ້ເລືອກໃຊ້ງານຂຶ້ນກັບຄວາມທະນັດ ຫຼື

ຄວາມສາມາດຂອງນັກພັດທະນາໂປຣແກຣມ(Programmer)

ທີ່ຈະເລືອກໃຊ້ພາສາໂປຣແກຣມໃຫ້ເໝາະສົມກັບວຽກງານທີ່ຈະນຳໄປໃຊ້ເຊັ່ນ ພາສາ React.js, ພາສາ Python, ພາສາ PHP, ພາສາ C#, ພາສາ SQL ເປັນຕົ້ນ.

ແຕ່ວ່າໂປຣແກຣມຕ່າງໆເຫຼົ່ານັ້ນຮູບແບບການຂຽນຄໍາສັ່ງເພື່ອໃຫ້ຄອມພິວເຕີເຮັດວຽກຈະເປັນການຮຽບຮຽງໄວຍະກອນ ພາສາອັງກິດ ຕາມຮູບແບບຄໍາສັ່ງຂອງພາສາໂປຣແກຣມຕ່າງໆ ເຊິ່ງໃນການປະມວນຜົນຄໍາສັ່ງໂປຣແກຣມນັ້ນ ຄອມພິວເຕີຈະບໍ່ສາມາດເຂົ້າໃຈໃນຄໍາສັ່ງເຫຼົ່ານີ້ໄດ້ໂດຍກົງ ເນື່ອງຈາກວ່າຄອມພິວເຕີເຂົ້າໃຈ ແລະ ປະມວນຜົນພາສາເຄື່ອງ

(Machine Language) ຫຼື ຮູບແບບຂອງເລກຖານສອງ (Binary digit) ຄື 0 ແລະ 1 ເທົ່າ

ນັ້ນດັ່ງນັ້ນຈຶ່ງມີຂັ້ນຕອນໃນການແປພາສາໂປຣແກຣມຕ່າງໆໃຫ້ເປັນພາສາຂອງເລກຖານສອງກ່ອນ

ໂດຍໃຊ້ໂປຣແກຣມແປພາສາ (Language Translator Program) ເຊັ່ນ: ແອສເຊມເບິລ (Assembler),

ຄອມໄຟລີ (Compiler) ຫຼື ອິນເຕີພີເຕີ (Interpreter) ຄອມພິວເຕີຈຶ່ງຈະເຂົ້າໃຈ ແລະ

ສາມາດປະມວນຜົນຕາມຄໍາສັ່ງຂອງຜູ້ໃຊ້ໂປຣແກຣມຕ່າງໆເຫຼົ່ານັ້ນໄດ້.

Next.js

ໃນໂລກຂອງການພັດທະນາເວັບ,

ແຟຣມເວີກບໍ່ຫລາຍຕົວໄດ້ສົ່ງຜົນກະທົບທີ່ແຂງແກ່ຫມູ່ນັກພັດທະນາຫລາຍເຫມືອນຄື Next.js.

ມັນໄດ້ຖືກແນະນຳໂດຍ Vercel (ທີ່ເຄີຍຮູ້ຈັກໃນຊື່ Zeit) ໃນປີ 2016, Next.js

ໄດ້ເຕີບໂຕເປັນແຟຣມເວີກທີ່ມີຄວາມອາດສະດ້ວນ ແລະ ປະສິດທິພາບສຳລັບການສ້າງແອັບ React ທັນສະໄໝ.

ເສັ້ນທາງຂອງມັນສະທ້ອນເຫັນການພັດທະນາຂອງການພັດທະນາເວັບໃນຕົນເອງ, ຈາກການເພີ່ມຂຶ້ນຂອງ server-rendered applications ສູ່ການນຳໃຊ້ວິທີການ static generation, hybrid rendering, ແລະ

ການປະສົມປະສານຂອງປັນຍາປະດິດ (AI). Next.js ມີຄວາມຕັ້ງໃຈທີ່ຈະມອບຄວາມໄວ, ຄວາມຍືດຫຍຸ່ນ, ແລະ

ປະສົບການທີ່ດີຕໍ່ນັກພັດທະນາ, ເຮັດໃຫ້ Next.js

ເປັນຕົວເລືອກທີ່ຮູ້ຈັກຫລາຍໃນວົງການນັກພັດທະນາຫລາຍສາຂາອຸດສາຫະກຳ.

ຕົ້ນກຳເນີດແລະຄວາມຄິດເຫັນພື້ນຖານ (2016)

Next.js ໄດ້ຖືກປ່ອຍໂຕໃນເດືອນຕຸລາ 2016 ໂດຍ Guillermo Rauch, CEO ຂອງ Vercel.

ແຟຣມເວີກນີ້ໄດ້ເປີດເຜີຍໂດຍໃຊ້ລິຂະສິດ MIT,

ເພື່ອເຊີນຊວນນັກພັດທະນາທົ່ວໂລກໃຫ້ມີສ່ວນຮ່ວມໃນການພັດທະນາຂອງມັນ. ໃນຝື້ນຖານ, Next.js

ໄດ້ພັດທະນາມາເພື່ອແກ້ໄຂບັນຫາທີ່ເພີ່ມຂຶ້ນໃນລະບົບ React ecosystem: ເຖິງແມ່ນ React

ແມ່ນແຜລະຕະຟອມທີ່ມີຄວາມສາມາດສູງສຳຫລັບການສ້າງໜ້າຕາຜູ້ໃຊ້, ແຕ່ມັນຍັງຂາດຄວາມອັດຕະໂນມັດໃນການ

SSR (server-side rendering), ເປັນຄຸນສົມບັດທີ່ຈຳເປັນໃນການພັດທະນາປະສິດທິພາບ ແລະ SEO. Next.js

ໄດ້ນຳ SSR ມາໃຫ້ຜູ້ພັດທະນາ ເຮັດໃຫ້ເວັບໄຊທ໌ຂອງພວກເຂົາໄວ ແລະ ເຮັດໃຫ້ປະສົບການຜູ້ໃຊ້ລຽບຫມາຍຂຶ້ນ.

Next.js 1.0 ໄດ້ນຳ SSR, automatic code splitting ແລະ hot reloading ເຂົ້າມາ,

ເຮັດໃຫ້ນັກພັດທະນາສາມາດສ້າງແອັບພິຄະຊັນທີ່ມີຄວາມໄວສູງແລະມີປະສິດທິພາບສູງ.

ເລີ່ມເຫັນພະລັງງານໃໝ່ໃນລະບົບ JavaScript.

ການຂະຫຍາຍຄຸນສົມບັດຝື້ນຖານ (2017-2018)

ຫລັງຈາກການເປີດຕົວເບື້ອງຕົ້ນ, Next.js ໄດ້ກ່າຍເຕີກຄວາມນິຍົມແລະ Vercel ໄດ້ເພີ່ມຄຸນສົມບັດໃໝ່ເພື່ອໃຫ້

Next.js ມີຄວາມສາມາດສູງຂຶ້ນ. ໃນປີ 2017, Version 2 ໄດ້ນຳສະເໜີ static exporting,

ເຮັດໃຫ້ນັກພັດທະນາສາມາດສ້າງຂໍ້ມູນອອກເປັນ static HTML. ນີ້ເປັນຄວາມສາມາດທີ່ຄວາມຕ້ອງການໃນ

JAMstack architecture—a modern approach emphasizing JavaScript, APIs, and Markup—

ທີ່ກຳລັງເພີ່ມຄວາມນິຍົມ. Static exporting ເຮັດໃຫ້ Next.js ກັບຄວາມນິຍົມໃນເຫລົ່ານັ້ນ,

ສຳລັບນັກພັດທະນາທີ່ຕ້ອງການສ້າງເວັບໄຊທ໌ທີ່ມີຄວາມໄວສູງ, ປອດໄພ, ແລະຂະຫຍາຍໄດ້.

ໃນປີ 2018, Next.js ໄດ້ເພີ່ມຄວາມສາມາດໃຫ້ນັກພັດທະນາສາມາດກຳນົດຄ່າ Webpack ແລະ Babel

ເພື່ອຂຽນໂຄດຂອງຕົນໃຫ້ສອດຄີກັບຄວາມຕ້ອງການຂອງພວກເຂົາ. ການນຳເຂົ້າ dynamic ຍັງເພີ່ມປະສິດທິພາບ,

ເຮັດໃຫ້ Next.js ເປັນທີ່ສະຫນັບສະຫນູນຢ່າງຫນາແຫນນໃນຊຸມຊົນນັກພັດທະນາ.

ການຮັບຄືນຕໍ່ JAMstack ແລະການປະສານ API (2019)

ໃນປີ 2019, Next.js ໄດ້ພັດທະນາເພີ່ມເຕີມເພື່ອຮອງຮັບຫຼັກການຂອງ JAMstack,

ມຸ່ງໄປຫາການສ້າງເວັບທີ່ມີການປະສານ API-driven applications. Version 8 ໄດ້ນຳສະເໜີ Static Site

Generation (SSG), ເຮັດໃຫ້ Next.js ສາມາດສ້າງເວັບທີ່ເປັນລະບົບ static ໄດ້ທີ່ເວລາ build.

Hybrid Rendering ແລະການນຳໃຊ້ທົ່ວໂລກ (2020)

Next.js ໄດ້ສືບຕໍ່ນຳໃຊ້ໃນ 2020 ໂດຍນຳເອົາລັກສະນະ Incremental Static Regeneration (ISR) ມາໃຊ້. ISR ເຮັດໃຫ້ນັກພັດທະນາສາມາດເຮັດເວັບໄຊທ໌ທີ່ມີການປະສົມລະຫວ່າງ static ແລະ dynamic, ເພີ່ມປະສິດທິພາບໃນການປັບປຸງເນື້ອຫາໃນເວັບໄຊທ໌ໂດຍບໍ່ຕ້ອງປະຕິບັດການ build ໃໝ່ທັງໝົດ.

Version 10 ຍັງນຳສະເໜີການປັບປຸງພາບດ້ວຍ <Image /> component ທີ່ສາມາດຈັດການຂະໜາດ ແລະຮູບແບບພາບດ້ວຍຄວາມລຽບຫມາຍ, ສິ່ງຜົນໃຫ້ເວັບໄຊທ໌ເຮັດວຽກໄວຂຶ້ນ.

ຄຸນສົມບັດທັນສະໄໝ ແລະ ເນັ້ນໃສ່ປະສິບການຂອງນັກພັດທະນາ (2021-2022)

ເມື່ອ Next.js ໄດ້ຮັບການນຳໃຊ້ຢ່າງແຜດຫລາຍ, Vercel

ໄດ້ມຸ່ງເນັ້ນໃນການປັບປຸງຄວາມໄວແລະປະສິບການຂອງນັກພັດທະນາ. Version 12

ໄດ້ເປັນຄວາມປ່ຽນແປງທີ່ສຳຄັນ, ໂດຍໄດ້ປ່ຽນຈາກ Babel ເປັນ SWC ທີ່ເປັນ compiler ທີ່ເຮັດດ້ວຍ Rust,

ຊຶ່ງເພີ່ມຄວາມໄວໃນການ build ຢ່າງຫຼາຍ. ການຮອງຮັບ middleware ແລະ edge functions

ໄດ້ກາຍເປັນສ່ວນສຳຄັນຂອງ framework ເຮັດໃຫ້ນັກພັດທະນາສາມາດ deploy ໂຄດໄປຢູ່ network edge ເພື່ອຫຼຸດຄ່າ latency ສຳລັບຜູ້ໃຊ້.

ການປະສານກັບ React 18

ໄດ້ເອື້ອອຳນວຍໃຫ້ມີຄວາມສາມາດໃໝ່ເຊັ່ນການຮອງຮັບຄຸນສົມບັດເຊິ່ງກຳລັງຈະພ້ອມສຳລັບ Concurrent Mode,

ຄື Suspense ແລະ server components, ເຮັດໃຫ້ການໂຫຼດຂໍ້ມູນລົງໄປດ້ວຍການໃຊ້ງານເປັນໄປດ້ວຍຄວາມລຽບງ່າຍ

ແລະປະສິບການທີ່ດີສູງຂຶ້ນ. ຄວາມກ້າວໜ້າເຫລົ່ານີ້ເຮັດໃຫ້ Next.js ກາຍເປັນ framework

ທີ່ຄິດຕໍ່ເອົາໜ້າໄປຫາອະນາຄົດ, ທີ່ສາມາດໃຊ້ງານໄດ້ດີກັບຄຸນສົມບັດໃໝ່ຂອງ React

ແລະມີປະສິດທິພາບທີ່ບໍ່ມີໃຜສຸດຂຶ້ນ.

ຍຸກສະໄໝ AI ແລະ App Router (2023-ປະຈຸບັນ)

ໃນປີຫຼ້າໆມານີ້, Next.js

ໄດ້ຂະຫຍາຍການມຸ່ງໄປຫາການສະໜັບສະໜູນເພື່ອຄວາມສາມາດໃນດ້ານປັນຍາປະດິດແລະການຮຽນຮູ້ຂອງເຄື່ອງ,

ສະທ້ອນຄວາມຕ້ອງການທີ່ເພີ່ມຂຶ້ນສຳລັບປະສິບການເວັບທີ່ຂັບເຄື່ອນດ້ວຍ AI. Next.js

ເວລານີ້ຮອງຮັບການປະສານກັບ framework ຂອງ AI ທີ່ນິຍົມ, ເຊັ່ນ OpenAI ແລະ Hugging Face,

ເຮັດໃຫ້ນັກພັດທະນາສາມາດສ້າງແລະສົ່ງເຄື່ອງສະໜັບສະໜູນປັນຍາປະດິດໄດ້ງ່າຍຂຶ້ນ.

Version 13 ໄດ້ນຳເອົາ App Router, ວິທີໃໝ່ໃນການສ້າງໂຄງສ້າງແອັບທີ່ຫຼຸດຄວາມສັບສົນໃນການບໍລິຫານ routing ແລະປະສົມປະສານຄວາມສຸດພາບກັບ server components.

ການອັບເດດນີ້ເຮັດໃຫ້ນັກພັດທະນາສາມາດສ້າງໂຄງສ້າງໃຫມ່ທີ່ປະສົມປະສານຄວາມດີຂອງ routing ດັ່ງເກົ່າກັບຄວາມສົດໃຫມ່ຂອງ React ທີ່ສາມາດປັບໃຫ້ເໝາະສົມແລະຂະໜາດໄດ້ງ່າຍດາຍຍິ່ງຂຶ້ນ.

ທຸກມື້ນີ້, Next.js ຢູ່ໃນບົດບາດສຳຄັນໃນແວງວົງການພັດທະນາເວັບ, ມີການພັດທະນາຄວາມສາມາດເຂົ້າຮ່ວມກັບຄວາມກ້າວໜ້າໃນອຸດສາຫະກຳ ແລະຄວາມພັດທະນາດ້ານເຕັກໂນໂລຊີ. Vercel, ບໍລິສັດທີ່ພັດທະນາ Next.js, ໄດ້ສ້າງຊຸມຊົນທີ່ເຕີບໂຕຢ່າງດີລອບຝ້າຫລາຍທີ່ຢູ່ລອບ framework ນີ້, ແລະໄດ້ປ່ອຍຄຸນສົມບັດທີ່ມຸ່ງເນັ້ນທີ່ປະສິດທິພາບ, ຄວາມເຂົ້າເຖິງ, ແລະປະສົບການຂອງນັກພັດທະນາ.

ດ້ວຍການຮອງຮັບ SSR, SSG, ISR, ແລະ API routes, Next.js ເຮັດໃຫ້ນັກພັດທະນາສາມາດສ້າງແອັບທີ່ມີປະສິດທິພາບສູງແລະເໝາະສຳລັບ SEO ໃນຫລາຍເຫດການນຳໃຊ້ໄດ້ດີ.

ເມື່ອວົງການພັດທະນາເວັບຍັງຄົງມີການພັດທະນາຢ່າງຕໍ່ເນື່ອງ, ຄວາມຍືດຫຍຸ່ນຂອງ Next.js ຈະເຮັດໃຫ້ມັນຍັງມີຄວາມກ້າວໜ້າແລະປັບຕົວເພື່ອເຊື່ອມຊຸດເຂົ້າກັບຄວາມຕ້ອງການຂອງນັກພັດທະນາໃນການສ້າງແອັບທີ່ໄວ, ປອດໄພ, ແລະມີປະສິດທິພາບສູງໃນອະນາຄົດ.



ຮູບທີ 2. 19 ໂລໂກ້ຂອງ Next.js

III. ໂປຣແກຣມ, ເຄື່ອງມື ແລະ ພາສາທີ່ໃຊ້ໃນການພັດທະນາ ລະບົບ

ໂປຣແກຣມ Visual Studio Code

ຄວາມຮູ້ເບື້ອງຕົ້ນກ່ຽວກັບ Visual Studio Code: ໂປຣແກຣມ Visual Studio Code ອອກມາໃຫ້
ໃຊ້ງານມາຕັ້ງແຕ່ວັນທີ 26 ເມສາ ປີ 2015 ແລະ ພັດທະນາໂດຍໄມໂຄຊອບ ເປັນທັງຕົວແກ້ໄຂ ແລະ ປັບແຕ່ງໂຄດ



ຮູບທີ 2. 20 ຮູບສະແດງໂປຣແກຣມ Visual Studio Code

(Code Optimized Editor) ທີ່ຕັດຄວາມສາມາດມາຈາກ Visual Studio ຮຸ່ນປົກກະຕິ (ພວກ GUI designer) ອອກໄປເຫຼືອແຕ່ຕົວ Editor ຢ່າງດຽວ ສາມາດທຳງານໄດ້ຂ້າມແພດຟອມທັງຢູ່ window, Mac ແລະ Linux ຍັງຮັບຮອງຫຼາຍພາສາ ເຊິ່ງທາງໄມໂຄຊອບໄດ້ເປີດໃຫ້ໃຊ້ຟຣີອີກດ້ວຍ.

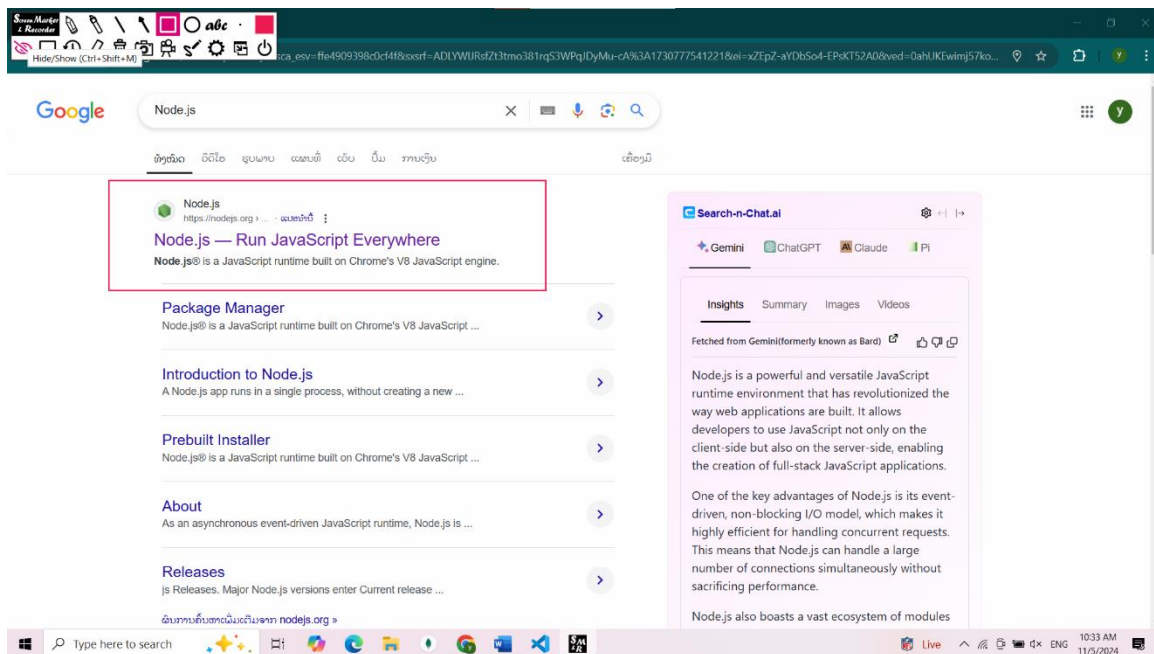
ຄວາມສາມາດຂອງເຈົ້າຕົວ Visual Studio Code: ນັ້ນຈະມີຄວາມສາມາດໃນການເປີດອ່ານ ແກ້ ໄຂ ແລະ ຂຽນໂຄດໄດ້ຄືກັນກັບ Editor ຕົວອື່ນໆ ເຊັ່ນ sublime, Atom, Notepad++ ມີນັ້ນເອງ ລວມ ເຖິງຄວາມສາມາດໃນການຕິດຕັ້ງເຄື່ອງມືເສີມ (Extension) ໂດຍຮັບຮອງໄວ້ຫຼາຍ ແລ້ວມັນຍັງມີຄວາມແຕກຕ່າງຈາກຕົວອື່ນຢູ່ບ່ອນວ່າມັນຖືກພັດທະນາມາໃຫ້ຕອບໂຈດນັກພັດທະນາຫຼາຍທີ່ສຸດ ການອອກແບບໜ້າຕາຂອງມັນຈຶ່ງອອກແບບມາແຕກຕ່າງຈາກຕົວອື່ນ ແລະ ໃຫ້ເປັນຮູບແບບທີ່ເຂົ້າໃຈໃຊ້ງານງ່າຍ ບໍ່ຕ້ອງສຶກສາແມ່ນ ຍັງເພີ່ມເຕີມກໍ່ໃຊ້ງານແບບງ່າຍໆໄດ້ເລີຍ.

ສິ່ງທີ່ເຮັດໃຫ້ ໂດດເດັ່ນກວ່າຕົວອື່ນໆ ຄືການທີ່ອອກແບບໃຫ້ການຄົ້ນຫາສິ່ງຕ່າງໆ ເຮັດອອກມາໃຫ້ໃຊ້ງານໄດ້ ຫຼາຍແລະເບິ່ງງ່າຍນັ້ນເອງ ມີຝັງຊັ້ນໃນການ Commit, Push & Pull ຢູ່ໃນຕົວ ຫຼື ຈະເບິ່ງ Change ຂອງ ໄຟລ໌ທີ່ເກີດຂຶ້ນກໍ່ໄດ້.

Next.js & Install : ກ່ອນອື່ນໝົດເຮົາຕ້ອງ Install Node.js ກ່ອນເຮົາຈຶ່ງສາມາດດຳເນີນການສ້າງໂປຣເຈັກ ແລະ ທຳການຂຽນໂປຣແກຣມໄດ້

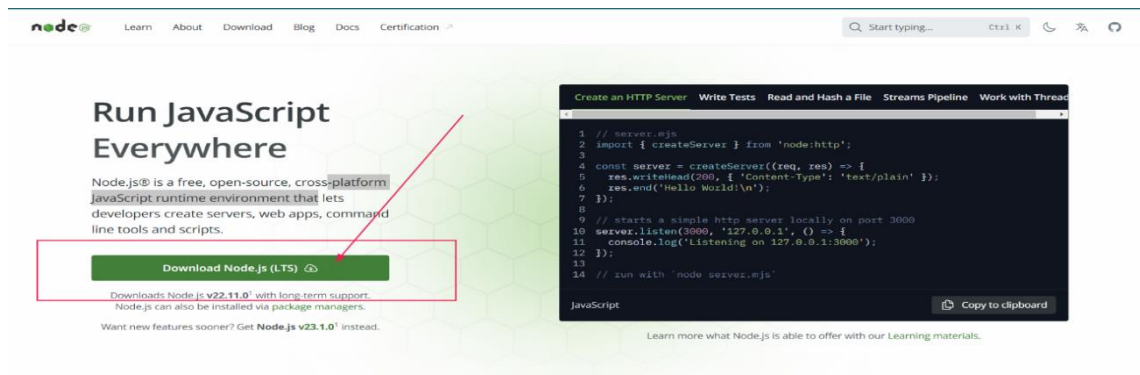
ຂັ້ນຕອນການຕິດຕັ້ງ Node.js ມີດັ່ງນີ້:

ເຂົ້າໄປທີ່ເວັບໄຊ: Node.js



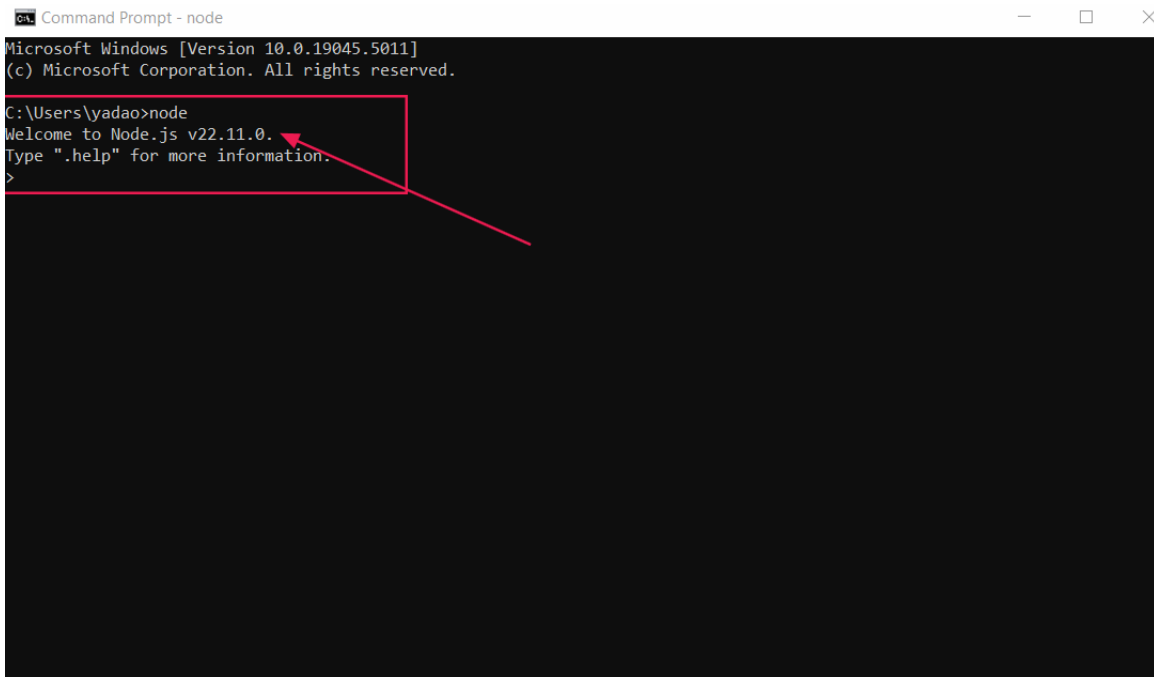
ຮູບທີ 2. 21 ຮູບສະແດງການຕິດຕັ້ງ Node.js

ທຳການດາວໂຫຼດ Node.js



ຮູບທີ 2. 22 ຮູບສະແດງການດາວໂຫຼດ Node.js

ຫຼັງຈາກທຳການດາວໂຫຼດແລ້ວກໍ່ທຳການຕິດຕັ້ງຄືກັບໂປຣແກຣມທຳມະດາ ຈາກນັ້ນກໍ່ເຂົ້າໄປເຊັກທີ່ command ວ່າການຕິດຕັ້ງ Node.js ໄດ້ແລ້ວ ຫຼື ບໍ່ ໂດຍຂຽນຄຳວ່າ node ແລ້ວກົດ Enter

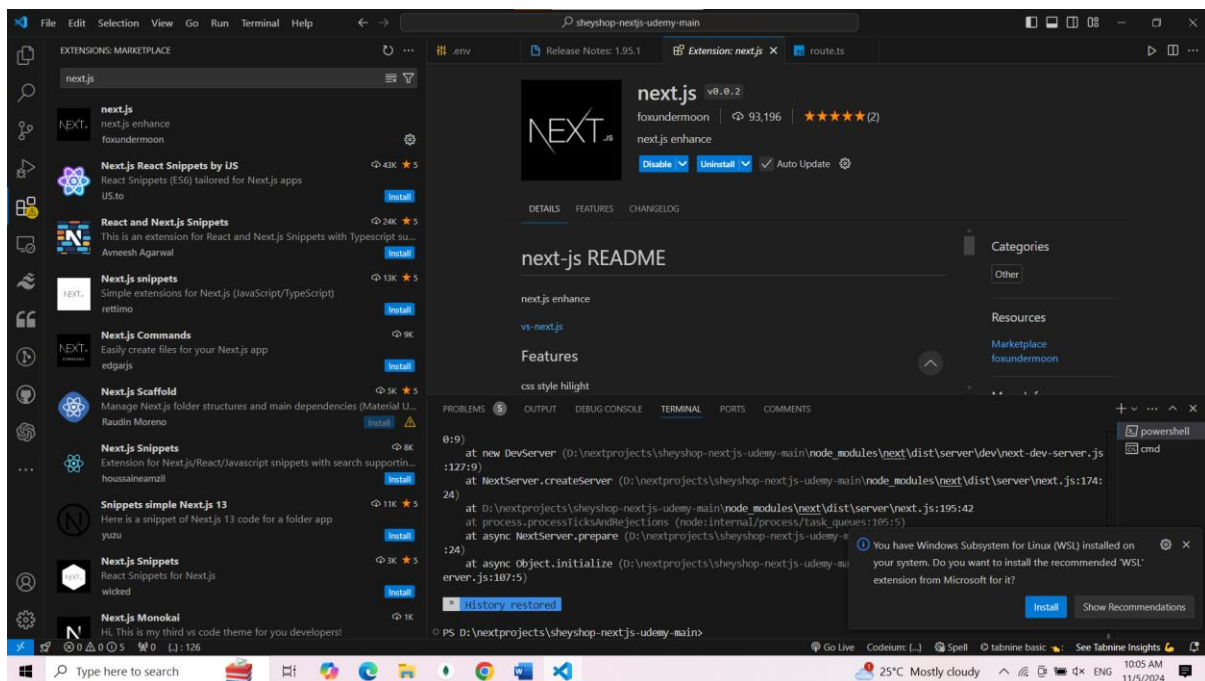


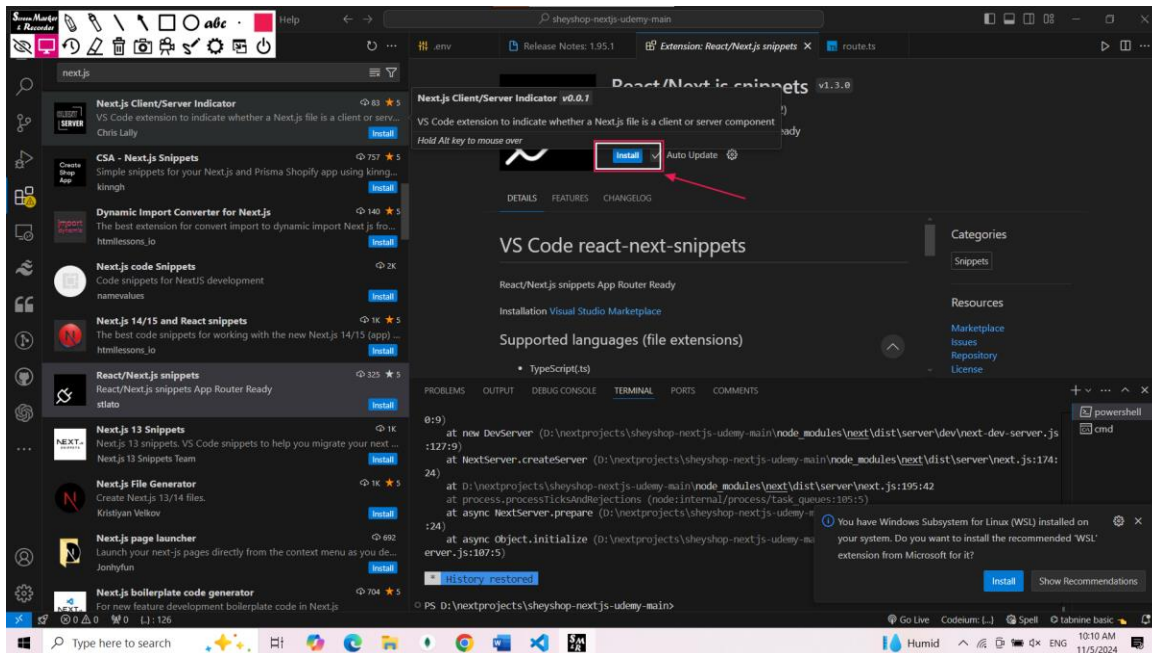
```
Command Prompt - node
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Users\yadao>node
Welcome to Node.js v22.11.0.
Type ".help" for more information.
>
```

ຮູບທີ 2. 23 ຮູບສະແດງການກວດສອບ Node.js ແລະ ເວີຊັນ

ຈາກນັ້ນກໍ່ເປີດ visual studio code ຂຶ້ນມາເພື່ອທຳການຕິດຕັ້ງ extension ເພື່ອຊ່ວຍໃນການຂຽນໂປຣແກຣມງ່າຍຂຶ້ນ ແລະ ມີລະບຽບ.





ຮູບທີ 2. 24 ຮູບສະແດງການຕິດຕັ້ງ Extension ລົງໃນ visual studio code

New Project Next.js ຫຼັງຈາກຕິດຕັ້ງ Extension ສໍາເລັດແລ້ວ ສາມາດສ້າງໂປຣເຈັກທີ່ເຮົາຕ້ອງການໄດ້ເລີຍ ວິທີການສ້າງແມ່ນ ເປີດ command prompt ຂຶ້ນມາແລ້ວ ເລືອກວ່າເຮົາຕ້ອງການໃຫ້ໂປຣເຈັກຂອງເຮົາຢູ່ໃສ່ແລ້ວລົ້ນຄໍາວ່າ : `npx create-next-app next_project_name`. ແລ້ວກົດ Enter.

```

Command Prompt
Would you like to use 'src/' directory? ... No / Yes
Would you like to use App Router? (recommended) ... No / Yes
Would you like to customize the default import alias (@/*)? ... No / Yes
Creating a new Next.js app in D:\demo_mastermam.

Using npm.
Initializing project with template: app-tw

Installing dependencies:
- react
- react-dom
- next

Installing devDependencies:
- autoprefixer
- postcss
- tailwindcss
- eslint
- eslint-config-next

npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated @humanwhodoes/config-array@0.13.0: Use @eslint/config-array instead
npm warn deprecated rimraf@3.0.2: rimraf versions prior to v4 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated @humanwhodoes/object-schema@2.0.3: Use @eslint/object-schema instead
npm warn deprecated eslint@8.57.1: This version is no longer supported. Please see https://eslint.org/version-support for other options.

added 359 packages, and audited 360 packages in 1m
137 packages are looking for funding
  run 'npm fund' for details

1 high severity vulnerability
To address all issues, run:
  npm audit fix --force
Run 'npm audit' for details.
Initialized a git repository.

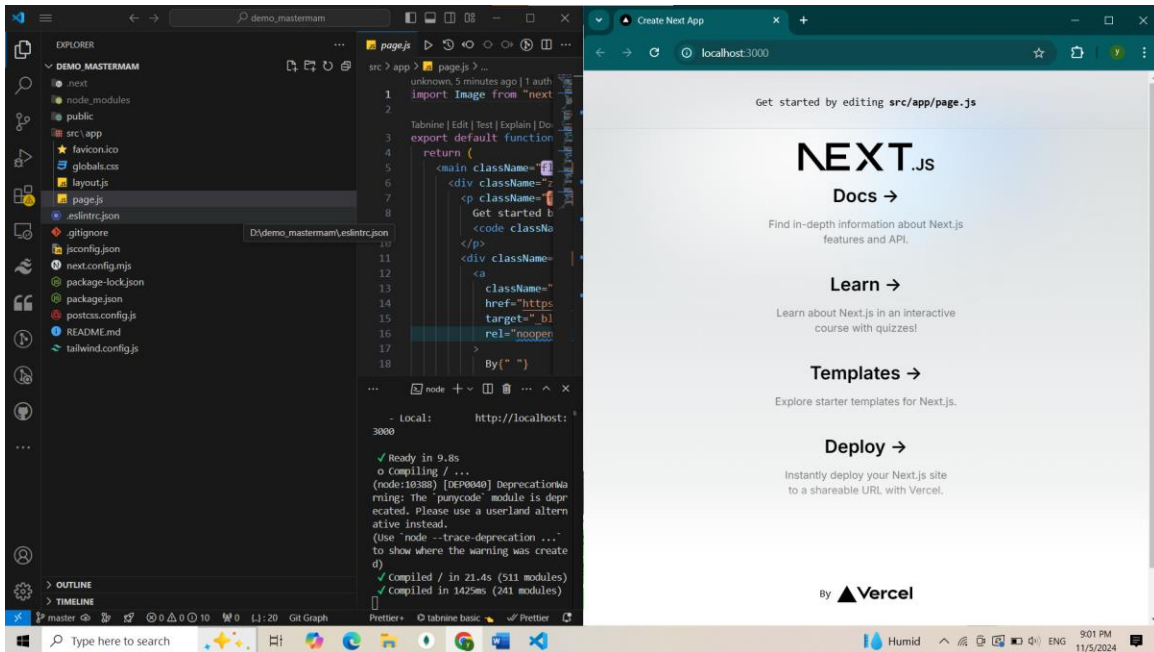
success! Created demo_mastermam at D:\demo_mastermam

A new version of 'create-next-app' is available!
You can update by running: npm i -g create-next-app

D:\>
  
```

ຮູບທີ 2. 25 ຮູບສະແດງການສ້າງໂປຣເຈັກ

ຫຼັງຈາກທຳການສ້າງໂປຣເຈັກແລ້ວກໍ່ທຳການລັນໂປຣເຈັກໂດຍເປີດໂປຣເຈັກໃນ visual studio code ແລ້ວເປີດ terminal ແລ້ວລັນດ້ວຍ npm run dev ແລ້ວຫຼັງຈາກນັ້ນຈະອອກ - Local: <http://localhost:3000> ໂດຍການກົດ Ctrl+Click ມັນຈະທຳການເປີດໜ້າ UI ຜ່ານທາງເວັບບຣາວເສີຂອງເຮົາຫຼັງຈາກນັ້ນຈຶ່ງສາມາດເລີ່ມຂຽນໂປຣແກຣມໄດ້ເລີຍ.



ຮູບທີ 2. 26 ຮູບສະແດງການລັນໂປຣເຈັກອອກໜ້າບຣາວເສີ

24. 3 Firebase

Firebase ເປັນ Backend-as-a-Service (Baas). ເຮັດໃຫ້ນັກພັດທະນາສາມາດໃຊ້ເຄື່ອງມື ແລະການບໍລິການທີ່ຫຼາກຫຼາຍເພື່ອຊ່ວຍໃນການພັດທະນາແອັບທີ່ມີຄຸນນະພາບ, ຂະຫຍາຍຖານຜູ້ໃຊ້ຂອງ, ແລະ ຫາກຳໄລ. ມັນຖືກສ້າງຂຶ້ນໃນໂຄງສ້າງພື້ນຖານຂອງ Google.

Firebase ຖືກຈັດປະເພດເປັນໂຄງການຖານຂໍ້ມູນ NoSQL, ເຊິ່ງເກັບຮັກສາຂໍ້ມູນໃນເອກະສານທີ່ຄ້າຍຄືກັບ JSON.



ຮູບທີ 2. 27 Firebase(Back-end for Google’s Mobile and Web App Development)

25. ຖານຂໍ້ມູນ MongoDB

MongoDB ເປັນຖານຂໍ້ມູນທີ່ໃຊ້ແບບ NoSQL, ຊຶ່ງເຮັດໃຫ້ມີການເກັບຂໍ້ມູນເປັນເອກະສານ (Document) ແທນການເກັບຕາມແຖວຫຼືຄ່າຕາຕະລາງເຊັ່ນຖານຂໍ້ມູນ SQL. MongoDB ເຮັດໃຫ້ການເຮັດວຽກກັບຂໍ້ມູນຍືດຢຸນຂຶ້ນ, ເນື່ອງຈາກສາມາດເກັບຂໍ້ມູນຫຼາຍແບບໄດ້ໃນເອກະສານດຽວ.

ວິທີການເຮັດວຽກຂອງ MongoDB

MongoDB ຈະເກັບຂໍ້ມູນໃນຮູບແບບເອກະສານ JSON (JavaScript Object Notation) ທີ່ເຮັດໃຫ້ງ່າຍຕໍ່ການເຮັດວຽກ ແລະ ການຈັດການຂໍ້ມູນ. ທຸກໆເອກະສານໃນ MongoDB ມີການກຳນົດໃຫ້ມີລະຫັດສະເພາະຂອງມັນ (ID) ແລະມີຄ່າຕາງໆທີ່ຫຼາກຫຼາຍ, ເຊັ່ນຊື່, ອາຍຸ, ທີ່ຢູ່ ຫຼື ລາຍລະອຽດອື່ນໆ.

ປະໂຫຍດທີ່ຈະໄດ້ຮັບເມື່ອໃຊ້ MongoDB

ຄວາມໄວສູງແລະຄວາມຍືດຢຸນ: MongoDB

ສາມາດຈັດການຂໍ້ມູນໄດ້ເໝາະສົມແລະລົງຕົວກັບຄວາມຕ້ອງການຂອງແອັບພິເຄຊັນທີ່ຕ້ອງການການຕອບສິດຂອງຂໍ້ມູນໄວ.

ສະມາດກັບການເກັບຂໍ້ມູນຂະໜາດໃຫຍ່: MongoDB ເຮັດໃຫ້ງ່າຍຕໍ່ການເພີ່ມຂີດຄວາມສາມາດຂອງລະບົບ (scaling up) ແລະ ການເກັບຂໍ້ມູນໂຕໃຫຍ່ໆ.

ເຮັດໃຫ້ການພັດທະນາແບບຄວາມໄວ (Agile Development):

 ເນື່ອງຈາກ MongoDB

ສາມາດປັບແຕ່ງໂຄງສ້າງຂອງຂໍ້ມູນໄດ້ງ່າຍ, ນັກພັດທະນາຈຶ່ງສາມາດເພີ່ມຫຼືປັບແຕ່ງຂໍ້ມູນໄດ້ຄ່ອນຂ້າງດີ.

ວິທີໃຊ້ MongoDB

ຕິດຕັ້ງ MongoDB ໃນເຄື່ອງຂອງທ່ານຫຼືໃຊ້ຮູບແບບບໍລິການຂອງ MongoDB Atlas ທີ່ເປັນຂໍ້ມູນຄລາວ.

ການສ້າງ Collection: ຕ້ອງການກຳນົດໂຄງສ້າງເອກະສານຂອງທ່ານໃນ Collection (ຄືກຸ່ມຂອງເອກະສານ).

ການເກັບຂໍ້ມູນ: ໃສ່ຂໍ້ມູນໃນເອກະສານທີ່ທ່ານສ້າງໃນ Collection ເພື່ອໃຊ້ໃນການວິເຄາະ.

ດັ່ງນັ້ນ MongoDB ຈະເປັນຕົວເລືອກທີ່ເໝາະສົມສໍາລັບການເຮັດວຽກກັບຂໍ້ມູນທີ່ຫຼາກຫຼາຍ ແລະ ບໍ່ມີໂຄງສ້າງຄົງທີ່.



ຮູບທີ 2. 28 ຖານຂໍ້ມູນ MongoDB

Node.js.

ເປັນ runtime ທີ່ເຮັດໃຫ້ນັກພັດທະນາສາມາດເຮັດວຽກ JavaScript ໄດ້ທັງດ້ານເຊີບເວີ (server-side) ແລະ ໄດ້ຮັບຄວາມໄວຂອງໂຄງສ້າງຕົວເຄື່ອນທີ່ເອີ້ນວ່າ V8 Engine ຂອງ Google.

Node.js ຈຶ່ງເປັນຕົວເລືອກທີ່ດີສໍາລັບການພັດທະນາແອັບພິເຄຊັນດ້ານເຊີບເວີແບບ real-time ທີ່ແອັບທີ່ຕ້ອງການຄວາມໄວສູງໃນການຕອບສິດ, ເຊັ່ນ Chat apps ແລະ ບໍລິການສະຖານະອອນໄລນ໌.

ການເຮັດວຽກຂອງ Node.js

Node.js ເປັນເຄື່ອງມືແບບ event-driven, non-blocking I/O, ເຊິ່ງໝາຍຄວາມວ່າມັນສາມາດຮັບຮອງການດໍາເນີນງານຫຼາຍໆກິດຈະກຳໄດ້ພ້ອມກັນໂດຍບໍ່ຕ້ອງລໍຖ້າໃຫ້ສໍາເລັດກ່ອນທີ່ຈະເລີ່ມກິດຈະກຳຕໍ່ໄປ. ເນື່ອງຈາກລັກສະນະທີ່ໄວແລະຍືດຢູ່ນີ້, Node.js ເຮັດໃຫ້ມັນເໝາະສົມສໍາລັບການເຮັດວຽກກັບແອັບພິເຄຊັນທີ່ຕ້ອງການການຕອບສິດແລະການເຊື່ອມຕໍ່ຫຼາຍຢ່າງ.

ຂໍ້ດີຂອງ Node.js

ຄວາມໄວສູງ: Node.js ເຮັດໃຫ້ສາມາດຕອບສິດໄວຂຶ້ນ ຍິ່ງເປັນແອັບເວີບເຊີບເວີຂະໜາດໃຫຍ່ຍິ່ງເໝາະສົມສໍາລັບ Node.js.

ຄວາມເຄື່ອນທີ່ງ່າຍດາຍ: Node.js ສາມາດນຳໃຊ້ຮອບດ້ານ ພັດທະນາທັງ front-end ແລະ back-end ໂດຍໃຊ້ JavaScript ເພື່ອຄວາມສະດວກຂອງທີມ.

ວິທີໃຊ້ Node.js

ຕິດຕັ້ງ Node.js ທີ່ [Node.js official website](https://nodejs.org/).

ເລີ່ມພັດທະນາ: ເຂົ້າທີ່ໄດ້ເວົ້າໃສ່ JavaScript ໃນໄຟລ໌ .js.

ໃຊ້ npm (Node Package Manager): npm ເປັນເຄື່ອງມືທີ່ໃຫ້ທ່ານຈັດການຫ້ອງສຳຫຼັບໂຄງການຂອງທ່ານ ແລະ ຕິດຕັ້ງຫ້ອງສຳຫຼັບເພີ່ມເຕີມໃຫ້ສຳເລັດໄດ້ງ່າຍ.

Node.js ເປັນຄຳຕອບທີ່ດີສຳລັບການພັດທະນາຊ່ອງທາງເຊີບເວີແບບໄວ



ຮູບທີ 2. 29 ພາສາໂປຣແກຣມ Node.js

ບົດທີ 3

ວິເຄາະ ແລະ ອອກແບບລະບົບ

1. ສະພາບຂອງລະບົບເກົ່າ

ຍັງບໍ່ທັນມີລະບົບຈັດເກັບປຶ້ມໃຫ້ລະອຽດ ແລະ ເປັນລະບຽບ.

ການເກັບຮັກສາບົດວິທະຍານິພົນແມ່ນໃຊ້ຮູບແບບເປັນ File Zip ຫຼື ເປັນປຶ້ມ ເຊິ່ງເຮັດໃຫ້ມີຄວາມຫຍຸ້ງຍາກໃນການຄົ້ນຫາຂໍ້ມູນ ແລະ ສະຫຼຸບລາຍງາຍຕ່າງໆ.

ຕ້ອງໃຊ້ຜືນທີ່ ແລະ ງົບປະມານໃນການເກັບກຳປຶ້ມ.

ຍັງບໍ່ທັນມີລະບົບທີ່ໃຫ້ບໍລິການໃນຈັດເກັບ File ປຶ້ມ.

ເຮັດໃຫ້ການເຂົ້າເຖິງ ບົດວິທະຍານິພົນ ຂອງນັກສຶກສາຫຍຸ້ງຍາກ.

II. ແຜນວາດລະບົບເກົ່າ

III. ສະພາບຂອງລະບົບໃໝ່.

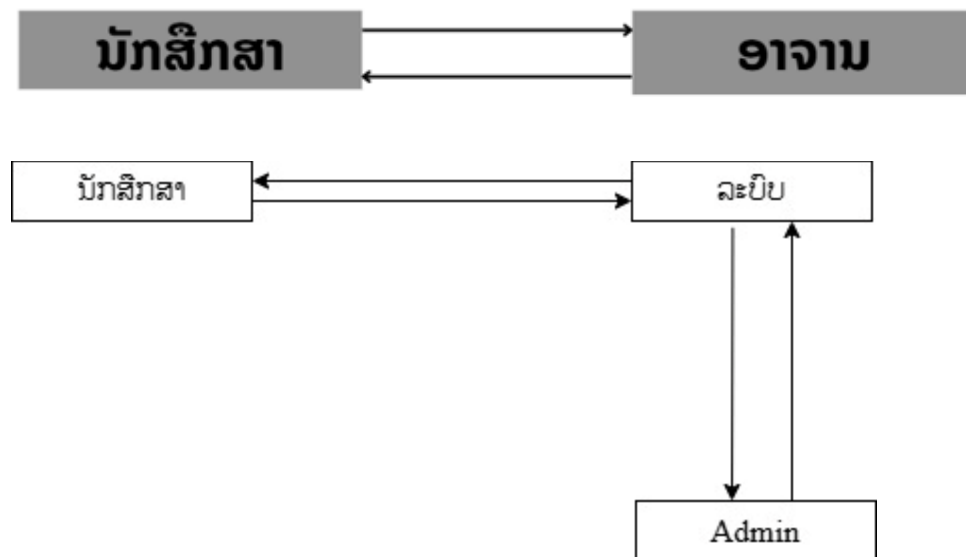
ຕ້ອງການລະບົບຖານຂໍ້ມູນ ເພື່ອຮອງຮັບການຈັດເກັບຂໍ້ມູນປຶ້ມ.

ຕ້ອງການລະບົບຖານຂໍ້ມູນທີ່ມີຄວາມປອດໄພ ເພື່ອຈັດເກັບຂໍ້ມູນປຶ້ມໃຫ້ເປັນລະບົບ.

ຮູບທີ່ 3. 1 ແຜນວາດຂອງລະບົບເກົ່າ.

ຕ້ອງການສະແດງຂໍ້ມູນປຶ້ມ ດ້ວຍລະບົບປະຕິບັດການໃນເວັບແອັບ.

ຕ້ອງການໃຫ້ນັກສຶກສາ ແລະ ອາຈານສາມາດນຳໃຊ້ເວັບແອັບໃນການຄົ້ນຫາຂໍ້ມູນປຶ້ມໄດ້ຢ່າງສະດວກ ແລະ ວ່ອງໄວ.



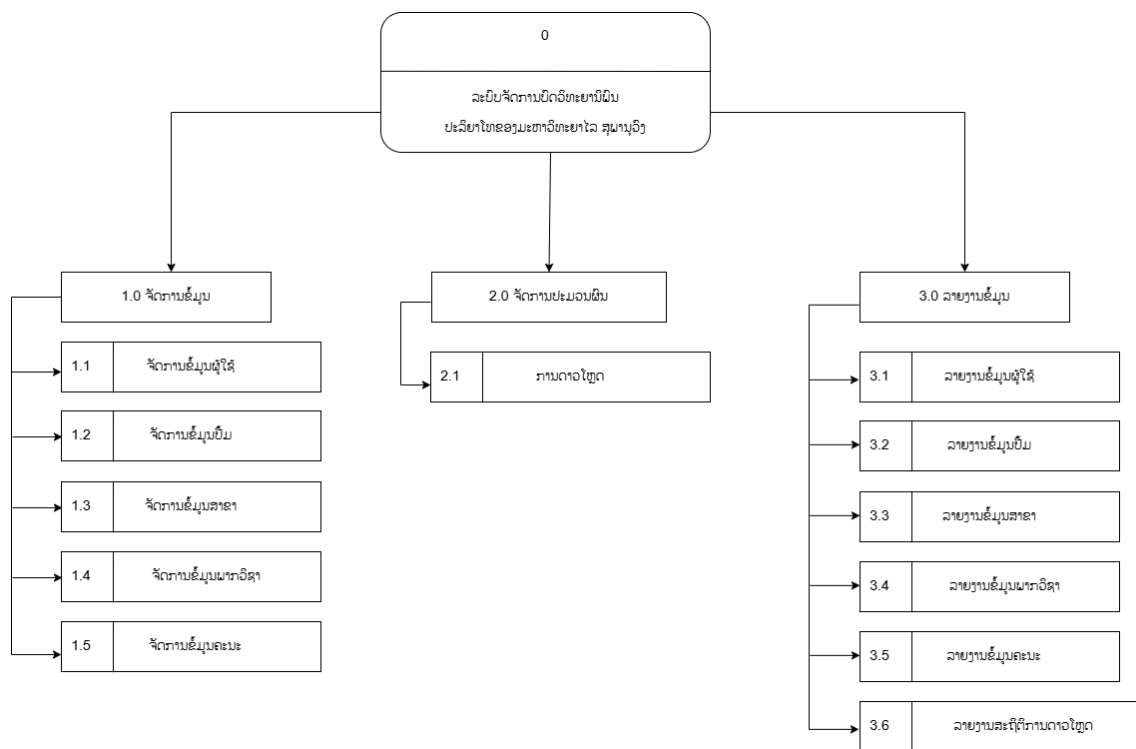
ຮູບທີ່ 3. 2 ແຜນວາດຂອງລະບົບໃໝ່.

IV. ສ່ວນປະກອບຂອງລະບົບໃໝ່.

<p>List of Boundaries</p> <ol style="list-style-type: none"> 1. ອາຈານ/ admin. 2. ນັກສຶກສາ. 	<p>List of Process</p> <p>1.0 ຈັດການຂໍ້ມູນ</p> <ol style="list-style-type: none"> 1.1. ຈັດການຂໍ້ຜູ້ໃຊ້. 1.2. ຈັດການຂໍ້ມູນປຶ້ມ. 1.3. ຈັດການຂໍ້ມູນສາຂາ. 1.4. ຈັດການຂໍ້ມູນພາກວິຊາ. 1.5. ຈັດການຂໍ້ມູນຄະນະ. <p>2.0 ຈັດການປະມວນຜົນ</p> <ol style="list-style-type: none"> 2.1. ຂໍ້ມູນການດາວໂຫຼດ. <p>3.0 ລາຍງານ</p> <ol style="list-style-type: none"> 3.1 ລາຍງານຂໍ້ມູນຜູ້ໃຊ້. 3.2 ລາຍງານຂໍ້ມູນປຶ້ມ. 3.3 ລາຍງານຂໍ້ມູນສາຂາ. 3.4 ລາຍງານຂໍ້ມູນພາກວິຊາ. 3.5 ລາຍງານຂໍ້ມູນຄະນະ. 3.6 ລາຍງານສະຖິຕິການດາວໂຫຼດ
<p>List of Data Store</p> <ol style="list-style-type: none"> 1. ຂໍ້ມູນຜູ້ໃຊ້. 2. ຂໍ້ມູນປຶ້ມ. 3. ຂໍ້ມູນສາຂາ. 4. ຂໍ້ມູນພາກວິຊາ. 5. ຂໍ້ມູນຄະນະ. 6. ຂໍ້ມູນການດາວໂຫຼດ. 	

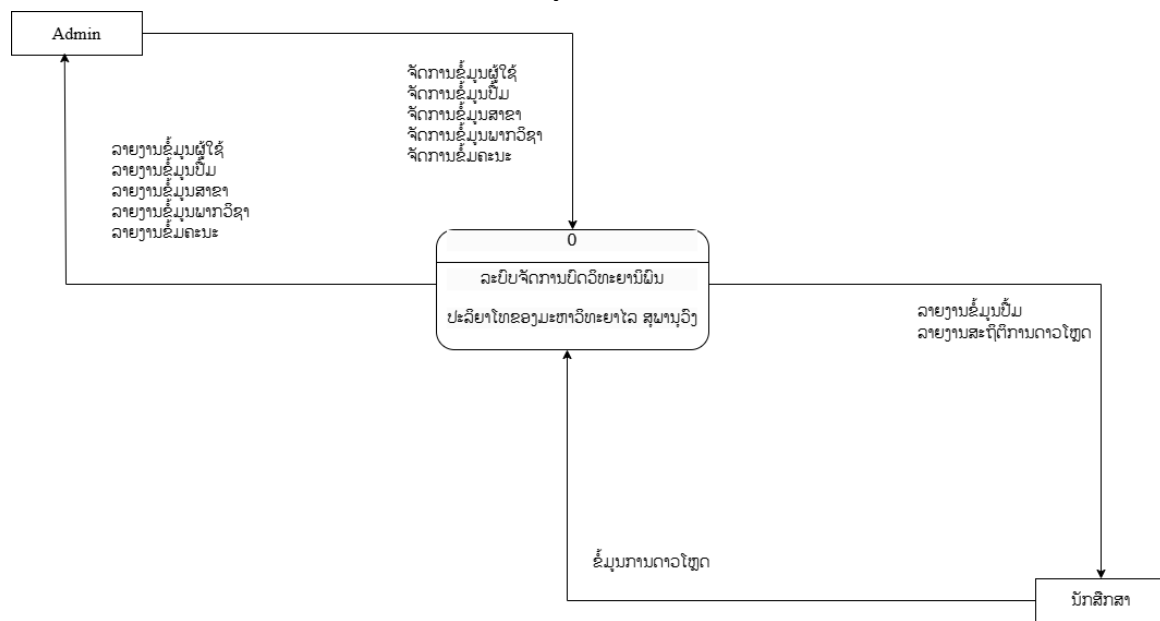
ຕາຕະລາງທີ 3.3 ສ່ວນປະກອບຂອງລະບົບ

V. ແຜນວາດຂອງລະບົບໃໝ່



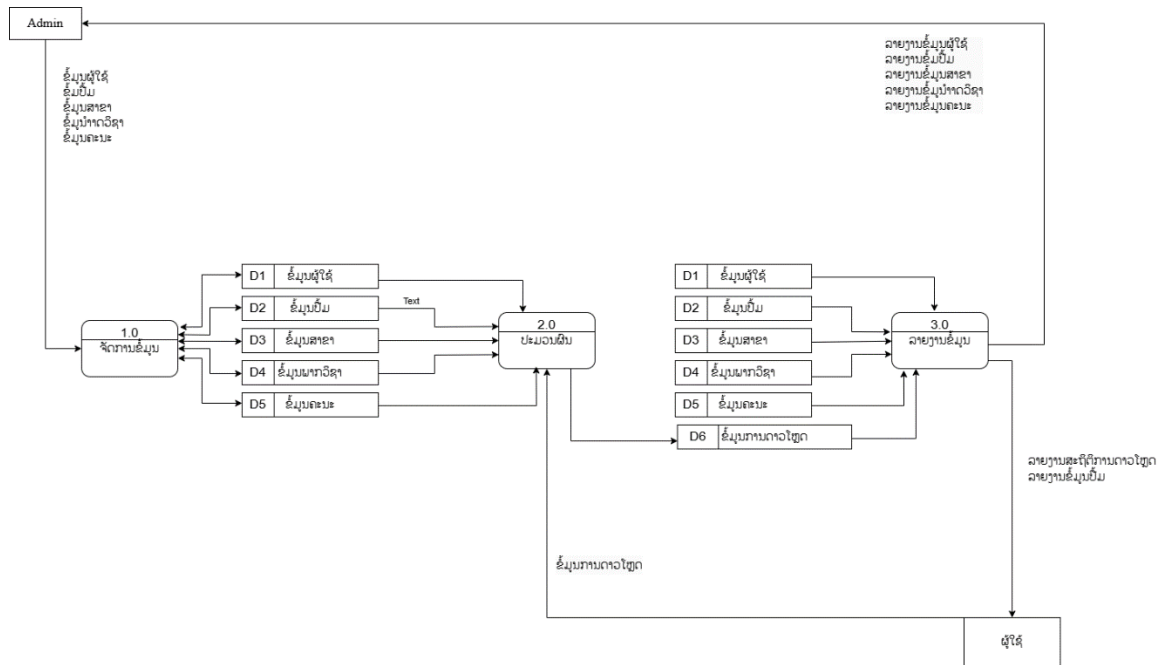
ຮູບທີ່ 3. 3 ແຜນວາດຂອງລະບົບໃໝ່ (Process Hierachy Chart)

VIII. ແຜນພາບລວມການໄຫຼຂອງຂໍ້ມູນ (context Diagram)



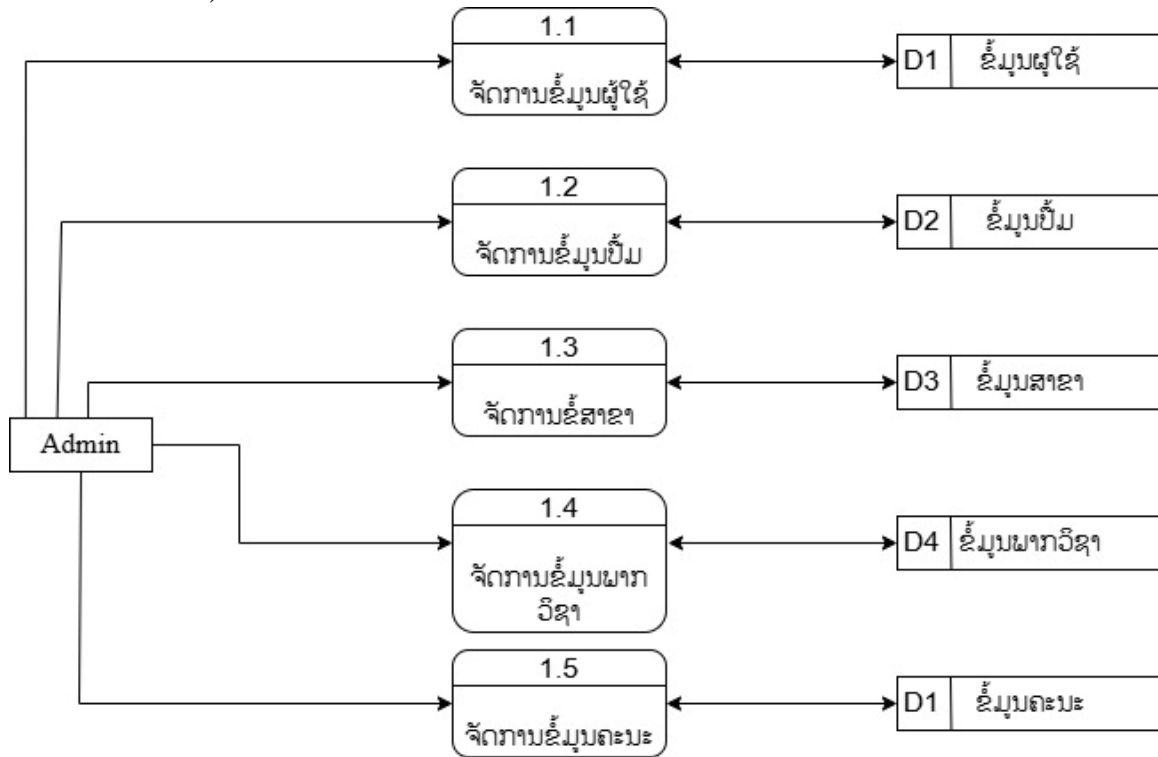
ຮູບທີ່ 3. 4 ສະແດງແຜນພາບລວມການໄຫຼຂອງຂໍ້ມູນ (context diagram)

VI. ແຜນພາບກະແສການໄຫຼຂໍ້ມູນຂອງຂໍ້ມູນລະດັບ 1 (DFD Level 1)



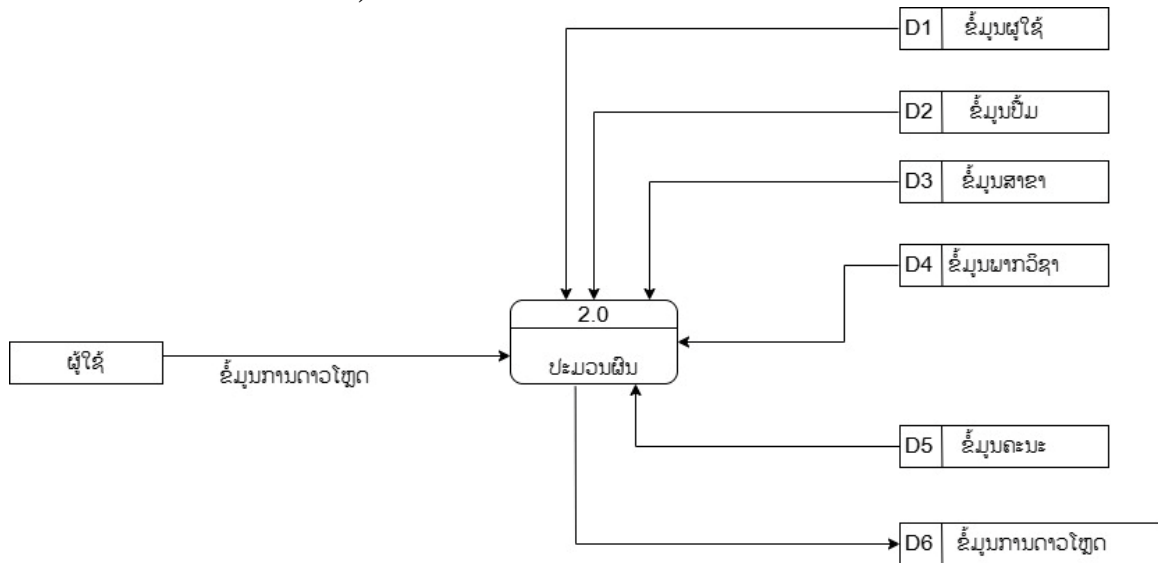
ຮູບທີ 3. 5 ສະແດງແຜນພາບກະແສການໄຫຼຂໍ້ມູນລະດັບ 1 (DFD Level 1)

VII. ແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນລະດັບ 2 ໂປຣເຊັດ 1.0 (DFD Level 2 Process 1.0)



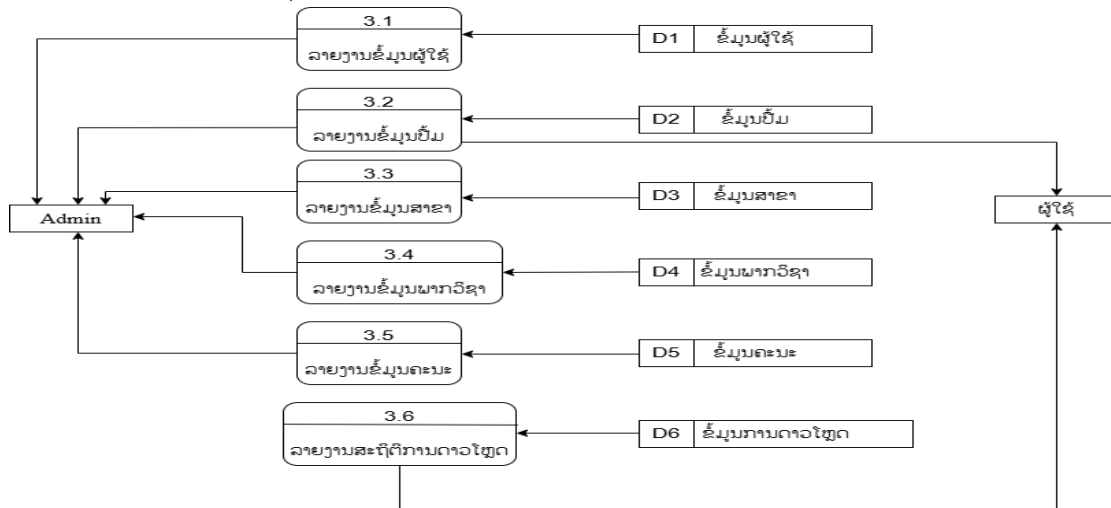
ຮູບທີ່3. 6 ສະແດງແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນລະດັບ 2 ໂປຣເຊັດ 1.0 (DFD Level 2 Process 1.0)

VIII. ແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນລະດັບ 2 ໂປຣເຊັດ 2.0 (DFD Level 2 Process 2.0)



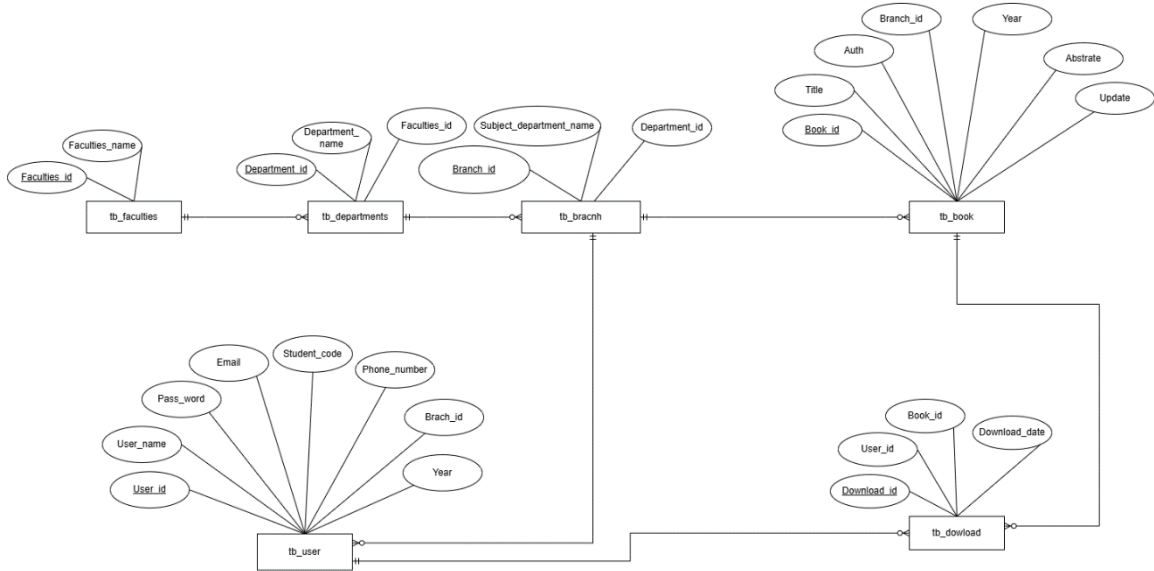
ຮູບທີ່3. 7 ສະແດງແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນລະດັບ 2 ໂປຣເຊັດ 2.0 (DFD Level 2 Process 2.0)

IX. ແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນລະດັບ 2 ໂປຣເຊັດ 2.0 (DFD Level 2 Process 2.0)



ຮູບທີ່3. 8 ສະແດງແຜນພາບກະແສການໄຫຼຂອງຂໍ້ມູນລະດັບ 2 ໂປຣເຊັດ 3.0 (DFD Level 2 Process 3.0)

X. ແຜນພາບການໄຫຼຂໍ້ມູນ (ER: Entity Relationship)



ຮູບທີ່ 3. 9 ຮູບທີ່ 3.9 ສະແດງແຜນພາບການໄຫຼຂໍ້ມູນ (ER: Entity Relationship)

XI. ຕາຕະລາງຖານຂໍ້ມູນ (Data Dictionary)

1. ຕາຕະລາງຂໍ້ມູນຜູ້ໃຊ້ (tb_user)

ລຳດັບ	ຊື່ລາຍການ	ປະເພດຂໍ້ມູນ	ຂະໜາດ	ປະເພດຄື	ຄວາມໝາຍຂອງຝັ່ງ
1	User_id	Int	5	PK	ລະຫັດຜູ້ໃຊ້
2	User_name	Varchar	15		ຊື່ຜູ້ໃຊ້
3	Pass_word	Varchar	15		ລະຫັດເຂົ້າລະບົບ
4	Email	Varchar	50		ອີເມວເຂົ້າລະບົບ
5	Student_code	Varchar	50		ລະຫັດນັກສຶກສາ
6	Phone_number	Int	11		ເບີໂທລະສັບ
7	Branch_id	Varchar	150	FK	ສາຂາວິຊາ
8	Year	Int	1		ປີຮຽນ

ຕາຕະລາງທີ່ 3.1 ຕາຕະລາງຜູ້ໃຊ້

2. ຕາຕະລາງຂໍ້ມູນປຶ້ມ (tb_book)

ລຳດັບ	ຊື່ລາຍການ	ປະເພດຂໍ້ມູນ	ຂະໜາດ	ປະເພດຄື	ຄວາມໝາຍຂອງຝັ່ງ
1	Book_id	Int	5	PK	ລະຫັດປຶ້ມ
2	Title	Varchar	255		ຊື່ປຶ້ມ
3	Author	Varchar	255		ຊື່ຜູ້ຂຽນ
4	Branch_id	Int	150	FK	ສາຂາວິຊາ

5	Year	Year			ສິກຮຽນ
6	Abstract	Text			ບົດຄັດຫຍໍ້
7	Book_file	file	1		ໄຟລ໌ປຶ້ມ
8	Upload_date	Date_time			ວັນທີອັບໂຫຼດ

ຕາຕະລາງທີ 3.2 ຕາຕະລາງປຶ້ມ

3. ຕາຕະລາງຂໍ້ມູນສາຂາ (tb_departments)

ລຳດັບ	ຊື່ລາຍການ	ປະເພດຂໍ້ມູນ	ຂະໜາດ	ປະເພດຄື	ຄວາມໝາຍຂອງຟິວ
1	Department_id	Int	2	PK	ລະຫັດສາຂາ
2	Department_name	Varchar	255		ຊື່ສາຂາ
3	Faculties_id	varchar	3	FK	ລະຫັດຄະນະ

ຕາຕະລາງທີ 3.3 ຕາຕະລາງສາຂາ

4. ຕາຕະລາງຂໍ້ມູນພາກວິຊາ (tb_subject_departments)

ລຳດັບ	ຊື່ລາຍການ	ປະເພດຂໍ້ມູນ	ຂະໜາດ	ປະເພດຄື	ຄວາມໝາຍຂອງຟິວ
1	Subject_department_id	Int	2	PK	ລະຫັດພາກວິຊາ
2	Subject_department_name	Varchar	255		ຊື່ພາກວິຊາ
3	Department_id	int	3	FK	ລະຫັດພາກວິຊາ

ຕາຕະລາງທີ 3.4 ຕາຕະລາງພາກວິຊາ

5. ຕາຕະລາງຂໍ້ມູນຄະນະ (tb_faculties)

ລຳດັບ	ຊື່ລາຍການ	ປະເພດຂໍ້ມູນ	ຂະໜາດ	ປະເພດຄື	ຄວາມໝາຍຂອງຟິວ
1	Faculties_id	Int	2	PK	ລະຫັດຄະນະ
2	Faculties_name	Varchar	255		ຊື່ຄະນະ

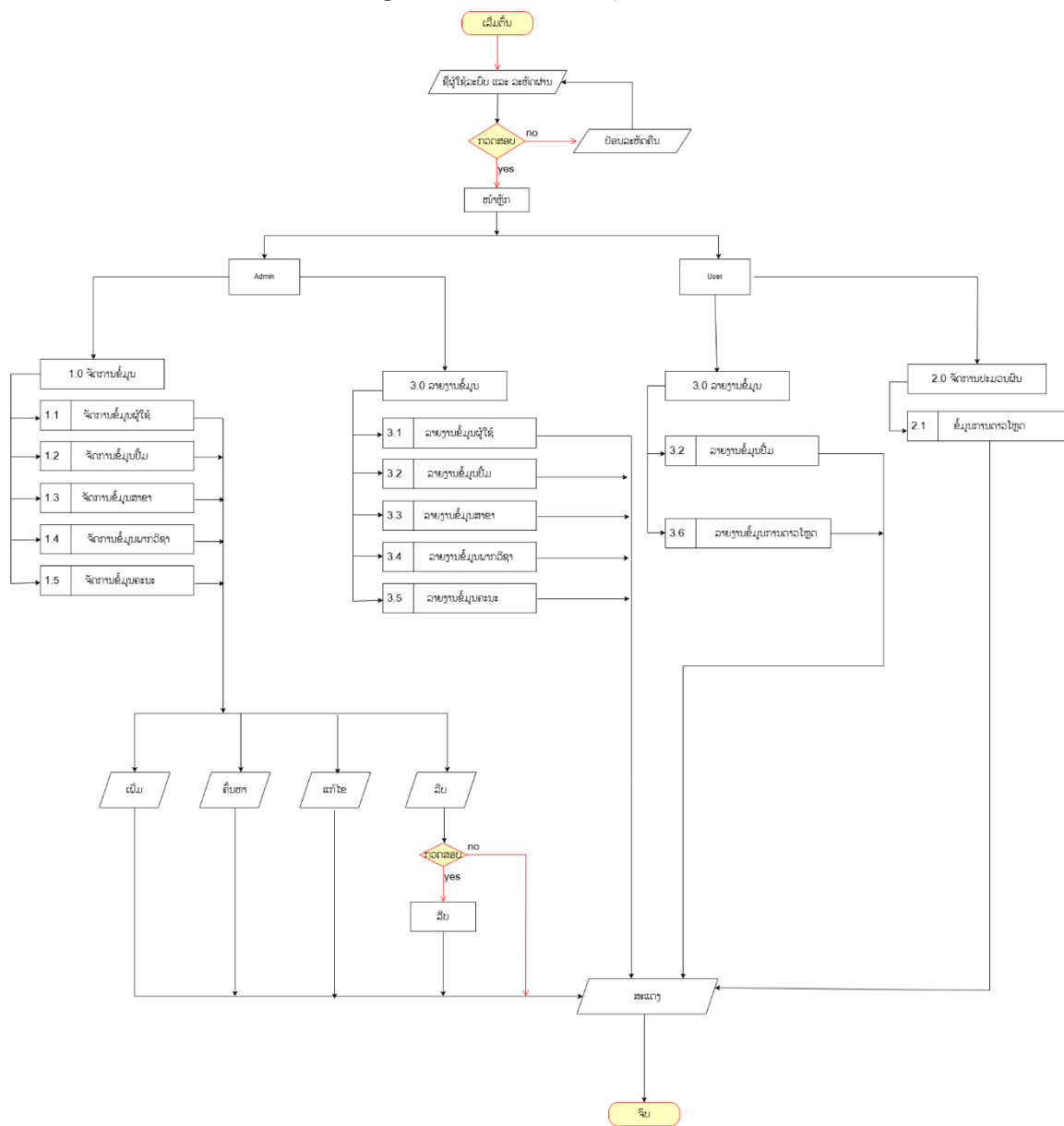
ຕາຕະລາງທີ 3.5 ຕາຕະລາງຄະນະ

6. ຕາຕະລາງຂໍ້ມູນການດາວໂຫຼດ (tb_download)

ລຳດັບ	ຊື່ລາຍການ	ປະເພດຂໍ້ມູນ	ຂະໜາດ	ປະເພດຄື	ຄວາມໝາຍຂອງຟິວ
1	Download_id	Int	2	PK	ລະຫັດການດາວໂຫຼດ
2	User_id	Int	5	FK	ລະຫັດຜູ້ໃຊ້
3	Book_id	Int	5	FK	ລະຫັດປຶ້ມ
4	Download_date	DATETIME			ວັນທີດາວໂຫຼດ

ຕາຕະລາງທີ 3.6 ຕາຕະລາງດາວໂຫຼດ

XII. ແຜນພາບການເຮັດວຽກຂອງລະບົບ (Algorithm)



ຮູບທີ່3. 10 ແຜນພາບການເຮັດວຽກຂອງລະບົບ (Algorithm)