**ບົດທີ 1**

# **ບົດສະເໜີ**

## ຄວາມສຳຄັນຂອງບັນຫາ.

# ໃນຍຸກແຫ່ງຄວາມກ້າວໜ້າທາງດ້ານເທັກໂນຊີຂໍ້ມູນຂ່າວສານທີ່ນັບມື້ມີການພັດທະນາຢ່າງບໍ່ຢຸດຢັ້ງທຸກບາດກ້າວໃນການດຳລົງຊີວິດລ້ວນແລ້ວແຕ່ນຳໃຊ້ເທັກໂນໂລຊີ, ບໍ່ວ່າຈະເປັນວິສາຫະກິດ, ທຸລະກິດ ແລະ ອົງກອນຕ່າງໆ.ໄດ້ນຳໃຊ້ເທັກໂນໂລຊີເຂົ້າມາຊ່ວຍໃນການເສີມສ້າງປະສິດ ທິພາບຂອງການເຮັດວຽກເພື່ອເຮັດໃຫ້ການເຮັດວຽກໄວຂື້ນ.ການນຳໃຊ້ໂປຮແກຮມເຂົ້າມາຊ່ວຍໃນການເຮັດວຽກຕ່າງໆ ແມ່ນນັບຫຼາຍຂື້ນ ໃນນັ້ນລະບົບຈັດການການຂາຍ ແມ່ນໂປຣແກຣມໜື່ງທີ່ຊ່ວຍໃຫ້ທຸລະກິດ ດຳເນີນໄປຢ່າງມີປະສິດທິພາບ ແລະ ວ່ອງໄວ. ຮ້ານຂາຍຢາ ດຣ.ດວງຈັນ ກໍເປັນອີກຮ້ານໜື່ງທີ່ຕ້ອງການນຳໃຊ້ເທັກໂນຊີທີ່ທັນສະໄໝເຂົ້າມາຊ່ວຍແກ້ໄຂບັນ ຫາຕ່າງໆໃນການເຮັດວຽກ.ດັ່ງນັ້ນ,ທາງຮ້ານ ດຣ ດວງຈັນ ຈື່ງຢາກພັດທະນາລະບົບດັ່ງກ່າວນີ້ມາແກ້ໄຂບັນຫາທີ່ພົບໃນປັດຈຸບັນ. ການຄົ້່ນຫາຂໍ້ມູນມີຄວາມຫຍຸ້ງຍາກ, ຂໍ້ມູນມີການຕົກເຮ່ຍເສຍຫາຍເຮັດໃຫ້ເກີດຄວາມຫຍຸ້ງຍາກຊັກຊ້າຕາມມາໃນເວລາທີ່ຕ້ອງການ, ບໍ່ສາມາດກວດສອບຈຳນວນຢາໄດ້, ການສະ ຫຼຸບບັນຊີແມ່ນມີຄວາມຊັກຊ້າ ບໍ່ມີປະສິດທີພາບ ແລະ ມີຂໍ້ຜິດພາດໃນການຂາຍ.

ຮ້ານ ຂາຍຢາ ດຮ.ດວງຈັນ ຕັ້ງຢູ່ບ້ານໂສກປ່າຫຼວງ,ເມືອງສີສັດຕະນາກ, ນະຄອນຫຼວງວຽງຈັນ. ເຈົ້າຂອງຮ້ານຊື່ວ່າ: ດຣ ດວງຈັນ ກອງຜາລີ.ເປີດມາໄດ້ 20 ປີ ແລ້ວຂຶ້ນທະບຽນວິສາຫະກິດສ່ວນບຸກຄົນລົງວັນທີ 3/2/2000,ເລກທະບຽນ1328/ຈທວ. ເປີດບໍລິການ ຈັນ-ວັນເສົາ ວັນ ຕັ້ງແຕ່ເວລາ 8:00 ຕອນເຊົ້າ ເຖິງ 22:00 ຕອນແລງ ຮ້ານຂາຍຢາ ດຣ ດວງຈັນ ແມ່ນຮ້ານຂາຍຢາທີປະກອບດ້ວຍຢາຫຼາຍຊະນິດບໍ່ວ່າຈະເປັນ. ຢາແກ້ປວດ, ຢາດີຫວັດ, ວີຕາມິນ, ຢາຕ້ານເຊື້ອ ແລະອື່ນໆ….

ຮ້ານຂາຍຢາ ດຣ ດວງຈັນ ມີຄົນເຮັດວຽກທັງໝົດ 4 ຄົນຄື: ເຈົ້າຂອງຮ້ານ, ລູກເຈົ້າຂອງຮ້ານ ແລະ ພະນັກງານ 2 ຄົນ ຍິງ 2 ຄົນ ພະນັກງານຍິງຄົນທໍາອິດຈະເຮັດໜ້າທີຈັດການຢາ ແລະ ພະນັກງານຍິງອີກຄົນ ໜຶ່ງແມ່ນປະຈໍາຢູ່ຮ້ານເພື່ອຂາຍ ແລະ ຮັບລາຍການສັ່ງຊື້ຈາກລູກຄ້າ.ເຈົ້າຂອງຮ້ານເຮັດໜ້າທີເກັບກຳເງິນ ທີໄດ້ຮັບຈາກການຂາຍແຕ່ລະມື້.ກໍລະນີເຈົ້າຂອງຮ້ານບໍ່ຢູ່ຈະແມ່ນໜ້າທີຂອງລູກເພິ່ນ.

ເນື່ອງຈາກວ່າການບັນທຶກຂໍ້ມູນຕ່າງໆ ບໍ່ວ່າຈະແມ່ນຂໍ້ມູນການຂາຍຢາ ແລະ ລາຍລະອຽດການຈັດການຢາ ຍັງໃຊ້ວິທີການຈົດກາຍໃສ່ເຈ້ຍ ແລະ ບາງຄັ້ງເຮັດໃຫ້ເກີດມີຄວາມຜິດພາດເຊັ່ນ: ເນື່ອງຈາກ ການຄິດໄລ່ຕ້ອງຄິດໄລ່ດ້ວຍມືເຮັດໃຫ້ຂາດຄວາມຊັດເຈນ ແລະ ຊັກຊ້າ, ການກວດສອບຢາ ແລະ ຂໍ້ມູນຂອງຢາບໍ່ມີຄວາມລະອຽດ ການກວດກາບັນຊີການຂາຍ, ສະຫຼຸບລາຍຮັບປະ ຈຳວັນ ທາງຮ້ານໄດ້ໃຊ້ການຈົດກ່າຍໃສ່ປື້ມເຮັດໃຫ້ຂໍ້ມູນຕົກເຮ່ຍເສຍຫາຍ ແລະ ຊັກຊ້າ

ຈາກບັນຫາຕ່າງໆທີກ່າວມາຂ້າງເທິງພວກຂ້າພະເຈົ້າຈຶ່ງໄດ້ມີຄວາມຄິດຢາກສ້າງລະບົບຈັດ ການການຂາຍຢາຂຶ້ນມາເພື່ອຊ່ວຍຫຼຸດຜ່ອນໃນຄວາມຫຍຸ້ງຍາກໃນການຈັດການ ແລະ ເພີ່ມຄວາມສະດວກສະບາຍໃຫ້ແກ່ຜູ້ໃຊ້ໃນການຈັດການລະບົບ ຫຼຸດຜ່ອນຄວາມຜິດພາດ ໃນການຄິດໄລ່ລາຍໄດ້ຈາກການຂາຍ, ກໍ່ໃຫ້ເກີດຄວາມສະດວກສະບາຍ ແລະ ວ່ອງໄວໃນການຄົ້ນຫາ ແລະ ແກ້ໄຂຂໍ້ມູນການຂາຍຢາ.

## **ຈຸດປະສົງຂອງການສຶກສາ**

ເພື່ອຫຼຸດຜ່ອນຄວາມຜິດພາດໃນການເຮັດວຽກຂອງຮ້ານຂາຍຢາ ດຣ ດວງຈັນ ບໍ່ວ່າຈະເປັນການເກັບກໍາຂໍ້ມູນການຂາຍ ແລະ ຄິດໄລ່ລາຍໄດ້ຈາກການຂາຍແຕ່ລະມື້, ບັນທຶກລາຍລະອຽດການສັງຊື້ ແລະ ນຳເຂົ້າຂອງຢາ. ຈຸດປະສົງຫຼັກຂອງການຄົ້ນຄ້ວາມີດັ່ງນີ້:

* ເພື່ອສຶກສາບັນຫາທີ່ເກີດຂຶ້ນຂອງຮ້ານຂາຍຢາ ດຣ ດວງຈັນ.
* ເພື່ອແກ້ໄຂບັນຫາໃນການຈັດການຂໍ້ມູນການຂາຍຢາ.ຮ້ານຂາຍຢາ ດຣ ດວງຈັນ ໃຫ້ມີຄວາມ

ຫຼຸດຜ່ອນຄວາມຜິດພາດ ແລະ ປ້ອງກັນການຕົກເຮ່ຍເສຍຫາຍຂອງຂໍ້ມູນ.

* ເພື່ອພັດທະນາລະບົບຈັດການຮ້ານຂາຍຢາຂອງຮ້ານ ດຣ ດວງຈັນ

## **ຂອບເຂດການຄົ້ນຄວ້າ**

ລະບົບຈັດການຮ້ານຂາຍຢາຂອງຮ້ານ ດຣ ດວງຈັນຮູບແບບຂອງລະບົບແມ່ນ ເປັນລະບົບແບບ Web-based Applications ເຊິ່ງປະກອບດ້ວຍໜ້າວຽກຫຼັກດັ່ງນີ້:

* ຈັດການຂໍ້ມູນພື້ນຖານ : (ຈັດການຂໍ້ມູນຢາ, ຈັດການຂໍ້ມູນປະເພດຢາ, ຈັດການຂໍ້ມູນຫົວໜ່ວຍ, ຈັດການຂໍ້ມູນຜູ້ສະໜອງ, ຈັດການຂໍ້ມູນຜູ້ໃຊ້,)
* ກວດສອບ : (ກວດສອບຢາໃກ້ໝົດອາຍຸ, ກວດສອບຢາໃກ້ຈະໝົດ)
* ບໍລິການ(ສັງຊື້ຢາ, ນຳເຂົ້າຢາ, ຂາຍຢາ)
* ລາຍງານ : (ລາຍງານຂໍ້ມູນຢາທັງໝົດ, ລາຍງານຂໍ້ມູນສະຕັອກຢາ, ລາຍງານການນຳເຂົ້າຢາ, ລາຍງານການສັງຊື້, ລາຍງານຂໍ້ມນຜູ້ສະໜອງ, ລາຍງານລາຍຮັບ-ລາຍຈ່າຍ)

## **ປະໂຫຍດທີ່ຄາດວ່າຈະໄດ້ຮັບ**

* ຈະໄດ້ລະບົບຈັດການຮ້ານຂາຍຢາຂອງຮ້ານ ດຣ ດວງຈັນ.
* ຈະສະດວກໃນການເກັບກຳຂໍ້ມູນການຂາຍຢາ ແລະ ການສັ່ງຊື້ຢາເຂົ້າໃນຮ້ານ.
* ຈະຊ່ວຍໃຫ້ການຂາຍຢາມີຄວາມສະດວກ ແລະ ວ່ອງໄວຂື້ນກວ່າເກົ່າ.
* ຈະໄດ້ລະບົບລາຍງານທີ່ມີຄວາມຖືກຕ້ອງ ແລະ ຊັດເຈນ.

# **ບົດທີ 2**

# **ທົບທວນເອກະສານ ແລະ ບົດຄົ້ນຄວ້າທີ່ກ່ຽວຂ້ອງ**

## ທົບທວນເອກະສານທີ່ກ່ຽວຂ້ອງ

### ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບຖານຂໍ້ມູນ (Database)

Database ຫຼື ຖານຂໍ້ມູນຄືກຸ່ມຂອງຂໍ້ມູນທີ່ຖືກເກັບລວບລວມໄວ້ໂດຍມີຄວາມສໍາພັນເຊິ່ງກັນ ແລະ ກັນໂດຍບໍ່ໄດ້ບັງຄັບວ່າຂໍ້ມູນທັງຫມົດນີ້ຈະຕ້ອງເກັບໄວ້ໃນແຟ້ມຂໍ້ມູນດຽວກັນ ຫຼື ແຍກເກັບຫຼາຍໆແຟ້ມຂໍ້ມູນ.

ລະບົບຖານຂໍ້ມູນຄືລະບົບທີ່ລວບລວມຂໍ້ມູນຕ່າງໆທີ່ກ່ຽວຂ້ອງກັນເຂົ້າໄວ້ດ້ວຍກັນຢ່າງມີລະບົບ, ມີຄວາມສໍາພັນລະຫວ່າງຂໍ້ມູນຕ່າງໆທີ່ຊັດເຈັນໃນລະບົບຖານຂໍ້ມູນຈະປະກອບດ້ວຍແຟ້ມຂໍ້ມູນຫຼາຍແພ້ມທີ່ມີຂໍ້ມູນກ່ຽວຂ້ອງກັນ,ສໍາພັນກັນເຂົ້າໄວ້ດ້ວຍກັນຢ່າງເປັນລະບົບ ແລະ ເປີດໂອກາດໃຫ້ ຜູ້ໃຊ້ສາມາດໃຊ້ງານ ແລະ ຮັກສາປ້ອງກັນຂໍ້ມູນເຫຼົ່ານີ້ໄດ້ຢ່າງມີປະສິດທິພາບໂດຍມີຊອບແວຣ໌ທີ່ປຽບສະເຫມືອນຊື່ກາງລະຫວ່າງຜູ້ໃຊ້ ແລະໂປຣແກຣມຕ່າງໆທີ່ກ່ຽວຂ້ອງກັບການໃຊ້ຖານຂໍ້ມູນເອີ້ນວ່າລະບົບຈັດການຖານຂໍ້ມູນ ຫຼື DBMS(Database Management system)ມີໜ້າທີ່ໃຫ້ຜູ້ໃຊ້ເຂົ້າເຖິ່ງຂໍ້ມູນໄດ້ງ່າຍສະດວກ ແລະ ມີປະສິດທິພາບ ການເຂົ້າເຖິງຂໍ້ມູນຂອງຜູ້ໃຊ້ອາດເປັນການສ້າງຖານຂໍ້ມູນ,ການແກ້ໄຂຂໍ້ມູນ ຫຼື່ ການຕັ້ງຄໍາຖາມເພື່ອໃຫ້ໄດ້ຂໍ້ມູນມາໂດຍບໍ່ຈໍເປັນຕ້ອງກ່ຽວກັບລາຍລະອຽດພາຍໃນໂຄງສ້າງຖານຂໍ້ມູນ.

* **ຈຸດດີໃນການສ້າງຖານຂໍ້ມູນ:**
* ສາມາດເກັບຂໍ້ມູນເປັນຈໍານວນຫຼວງຫຼາຍເຮັດໃຫ້ປະຍັດເນື້ອທີ່ໃນການຈັດເກັບເອກະສານທີ່ບັນທຶກໃສ່ເຈ້ຍ.
* ສາມາດປັບປຸງແກ້ໄຂຂໍ້ມູນໃນຖານຂໍ້ມູນໄດ້ຖືກຕ້ອງ ມີຄວາມທັນສະໄໝໄດ້ຕະຫຼອດເວລາ ແລະ ວ່ອງໄວ ນອກຈາກນີ້ຫັຍງສາມາດຄົ້ນຫາຂໍ້ມູນທີ່ຕ້ອງການໄດ້ຢ່າງວ່ອງໄວ,ຖືກຕ້ອງ ແລະ ປະຍັດເວລາ.
* ຫຼຸດຜ່ອນການຊໍ້າຊ້ອນໃນການບັນທຶກຂໍ້ມູນ ແລະ ເຮັດໃຫ້ມີຄວາມເປັນລະບຽບຮຽບຮ້ອຍ.
* ຫຼຸດຜ່ອນໃນການເກັບຮັກສາຂໍ້ມູນທີ່ມີຫຼາຍ,ງ່າຍຕໍ່ການຄົ້ນຫາຂໍ້ມູນດັ່ງກ່າວໄດ້ຢ່າງວ່ອງໄວ ແລະ ບໍ່ເກີດບັນຫາ.
  + 1. **ແຜນວາດຄວາມສຳພັນ Entity (ER Diagram)**

ການອອກແບບຈໍາລອງຖານຂໍ້ມູນ ER ນັ້ນກ່ອນອື່ນຕ້ອງຮູ້ຈັກຄວາມໝາຍສັນຍາລັກຕ່າງໆທີ່ໃຊ້ໃນການອອກແບບຈໍາລອງ ER ໄດ້ຢ່າງຖືກຕ້ອງດັ່ງນັ້ນໃນຫົວຂໍ້ນີ້ຈະເວົ້າເຖິງຄວາມໝາຍ ແລະ ການໃຊ້ງານສັນຍາລັກຕ່າງໆຂອງ ER.

* 1. ເອັນຕີຕີ້ (Entity)

ເອັນຕີຕີ້ຄືວັດຖຸທີ່ເຮົາສົນໃຈເຊິ່ງອາດເປັນໄດ້ທັງບຸກຄົນ, ສະຖານທີ່, ວັດຖຸ, ເຫດການ ຫຼື ແນວຄິດທີ່ກໍ່ໃຫ້ເກີດກຸ່ມຂອງຂໍ້ມູນທີ່ຕ້ອງການເອັນຕີຕີ້ແບ່ງອອກເປັນ 2 ປະເພດຄື:

* 1. Strong Entity:

ເປັນເອັນຕີຕີ້ທີ່ເກີດຂຶ້ນດ້ວຍຕົນເອງເປັນອິດສະຫຼະບໍ່ຂຶ້ນກັບເອັນຕີຕີໃດສັນຍາລັກທີ່ໃຊ້ຄືຮູບສີ່ຫຼ່ຽມ ແລະ ສາມາດເອີ້ນ Strong Entity ໄດ້ອີກຊື່ໜື່ງວ່າ Regular Entity.

CUSTOMER

STOCK

ຮູບທີ 1: ຮູບ Strong Entity

* 1. Weak Entity:

ເອັນຕີຕີ້ຊະນິດນີ້ຈະຂຶ້ນກັບເອັນຕີຕີ້ຊະນິດອື່ນໆບໍ່ສາມາດເກີດຂຶ້ນໄດ້ຕາມລໍາພັງ ແລະ ຈະຖືກລົບເມື່ອເອັນຕີຕີ້ຫຼັກຖືກລົບອອກ ສັນຍາລັກທີ່ໃຊ້ຄືຮູບສີ່ຫຼ່ຽມຊ້ອນກັນ.

CUSTOMER

STOCK

ຮູບທີ 2 ຮູບ Weak Entity

* 1. **ແອັດທຣິບີວ (Attribute)**

ແອັດທີຣບີວຄືຄຸນສົມບັດຂອງສົມບັດຂອງເອັນຕີຕີ້ສັນຍາລັກຂອງແອັດທີຣບີວຈະເປັນຮູບວົງມົນແອັດທຣິບີວໃດທີ່ຖືກໃຊ້ເປັນຄີຫຼັກຈະຖືກຂີດເສັ້ນກ້ອງກໍາກັບໄວ້.

emName

Position

emNo

EMPLOYEE

ຮູບທີ 4 ຮູບ Attribute

* 1. **ຄວາມສໍາພັນ (Relation)**

ຄວາມສໍາພັນໃນທີ່ນີ້ໝາຍເຖິງຄວາມສໍາພັນລະຫວ່າງເອັນຕີຕີ້ໂດຍແຕ່ລະຄວາມສໍາພັນຄວນມີຊື່ລະບຸໄວ້ເພື່ອໃຊ້ອະທິບາຍເຊິ່ງປົກກະຕິຈະໃຊ້ສັນຍາລັກຮູບດອກຈັນທີ່ພາຍໃນລະບຸຄໍາກິລິຍາໄວ້ເພື່ອອະທິບາຍຄວາມສໍາພັນ.

SUBJECT

teaches

LUCTURE

M

1

ຮູບທີ 5 ຮູບ Relation

* + 1. **ຄວາມຮູ້ກ່ຽວກັບການວິເຄາະ ແລະ ອອກແບບລະບົບ**

### ນັກວິເຄາະລະບົບ

ນັກວິເຄາະລະບົບຄືເປັນຄົນກາງໃນການຕິດຕໍ່ລະຫວ່າງຂໍ້ມູນກັບຜູ້ທີ່ກ່ຽວຂ້ອງເຊິ່ງໄດ້ແກ່ເຈົ້າຂອງລະບົບ(System Owner)ຜູ້ໃຊ້ລະບົບ(System User)ແລະ ຜູ້ສ້າງລະບົບ(SystemBuilders)ເພື່ອພັດທະນາລະບົບຂອງອົງກອນຂຶ້ນມາເຊິ່ງໜ້າທີ່ຫຼັກຂອງນັກວິເຄາະລະບົບແບ່ງອອກເປັນ 2 ສວນຄື: ວິເຄາະລະບົບ ແລະ ອອກແບບລະບົບ,ເປັນການສຶກສາ,ວິເຄາະ ແລະ ຈໍາແນກບັນຫາທີເກີດຂຶ້ນໃນລະບົບພ້ອມທັງສະເໜີແນວທາງ ແກ້ໄຂຕາມຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ງານ ແລະ ຄວາມເໝາະສົມຕໍ່ສະຖານະການເງິນຂອງອົງກອນເປັນວິທີອອກແບບ ແລະ ກໍານົດຄຸນສົມບັດທາງດ້ານເຕັກນິກໂດຍນໍາເອົາລະບົບຄອມພິວເຕີມາປະຍຸກໃຊ້ເພື່ອແກ້ໄຂບັນຫາທີ່ໄດ້ເຮັດການວິເຄາະມາແລ້ວ.

ສຶກສາເຖິງບັນຫາຂອງລະບົບພ້ອມທັງໄຈ້ແຍກຢ່າງມີຫຼັກການ ແລະ ນໍາໄປສູ່ການດັດແປງແກ້ໄຂລະບົບໃຫ້ມີປະສິດທິພາບດີຂຶ້ນ ແລະ ການກໍານົດຄວາມຕ້ອງການຂອງຜູ້ໃຊ້.

ສຶກສາວ່າວຽກທີ່ຈະນໍາເອົາຄອມພິວເຕີມາໃຊ້ງານໃນລະບົບນັ້ນມີຂັ້ນຕອນສະຫຼັບຊັບຊ້ອນແນວໃດ, ໃຊ້ຂໍ້ມູນອັນໃດ, ວຽກງານເຫຼົ່ານັ້ນມີບັນຫາຫຍັງແດ່ຄິດຫາແນວທາງນໍາເອົາຄອມພິວເຕີໄປໃຊ້ໃຫ້ເກີດປະໂຫຍດອັນໃດ.

### ໜ້າທີ່ຂອງນັກວິເຄາະລະບົບ

ນັກວິເຄາະລະບົບເປັນຜູ້ທີ່ສຶກສາເຖິງບັນຫາ ແລະຄວາມຕ້ອງການຂອງນັກທຸລະກິດໂດຍນໍາເອົາປັດໄຈ 3 ປະການຄື: ຄົນ (People),ວີທີການ (Method)ແລະ ເທັກໂນໂລຢີ (Technology)ໃຊ້ໃນການປັບປຸງ ຫຼື ແກ້ບັນຫາໃຫ້ກັບນັກທຸລະກິດ.

ເມື່ອໄດ້ມີການນໍາເອົາພັດທະນາການທາງດ້ານເທັກໂນໂລຢີຂອງຄອມພີວເຕີມາໃຊ້ ນັກວິເຄາະລະບົບຈະຕ້ອງຮັບຜິດຊອບເຖິງການກໍານົດລັກສະນະຂອງຂໍ້ມູນ(Data)ທີ່ຈະເກັບເຂົ້າສູ່ລະບົບຄອມພີວເຕີການໝູນວຽນ, ການປ່ຽນແປງຂອງຂໍ້ມູນ ແລະ ໄລຍະເວລາເພື່ອໃຫ້ເກີດປະໂຫຍດສູງສຸດແກ່ຜູ້ໃຊ້ ຫຼື ທຸລະກິດ.

ນັກວິເຄາະລະບົບ (System Analysis)ເປັນການສຶກສາ,ວິເຄາະ ແລະ ຈໍາແນກບັນຫາທີ່ເກີດຂຶ້ນໃນລະບົບພ້ອມທັງສະເໜີແນວທາງແກ້ໄຂຕາມຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ງານ ແລະ ຄວາມເໝາະສົມຕໍ່ສະຖານະການເງິນຂອງອົງກອນນັກວິເຄາະທີ່ດີມີໜ້າທີ່ດັ່ງນີ້:

* ລວບລວມຂໍ້ມູນ
* ຈັດການເອກະສານ
* ຈັດການພົດຈະນານຸກົມຂໍ້ມູນ
* ອອກແບບລະບົບ
* ທົດສອບໂປຣແກຣມ
* ຕິດຕັ້ງ ແລະ ປ່ຽນແປງລະບົບ
* ຈັດການເອກະສານຄູ່ມື
* ຈັດການແບບສອບຖາມ
* ບໍາລຸງຮັກສາ ແລະ ປະເມີນຜົນການເຮັດວຽກຂອງລະບົບ
* ເປັນຜູ້ໃຫ້ຄໍາປຶກສາ
* ເປັນຜູ້ປະສານງານ
* ເປັນຜູ້ແກ້ໄຂບັນຫາ
* ເປັນຕົວແທນການປ່ຽນແປງລະບົບ
* ເປັນຜູ້ກະກຽມຂໍ້ມູນໃຫ້ກັບ

**Management**



**User**

**Programmer**

**System Analysis**



ຖານຂໍ້ມູນຕ່າງໆລວມທັງຂໍ້ມູນດີບທີ່ຈະຕ້ອງປ້ອນເຂົ້າລະບົບວຽກຂອງນັກວິເຄາະລະບົບບໍ່ໄດ້ຢູ່ໃນລັກ ສະນະທີ່ແນ່ນອນແບບໂປຣແກຣມເມີບໍ່ມີຄໍາຕອບແນ່ນອນຈາກລະບົບທີ່ວາງໄວ້ບໍ່ວ່າຈະຖືກ ຫຼື ຜິດກໍ່ຕາມວຽກຂອງນັກວິເຄາະເກີດຈາກການຜະສົມຜະສານຈາກປັດໄຈຕ່າງໆທີ່ກ່ຽວຂ້ອງກັບລະບົບວຽກຄື: ຜູ້ໃຊ້, ວິທີການ, ເທັກໂນໂລຢີ ແລະ ອຸປະກອນຈົນໄດ້ຜົນລັບທີ່ເໝາະສົມອອກມາເປັນລະບົບວຽກຂອງນັກວິເຄາະລະບົບຈຶ່ງຕ້ອງກ່ຽວຂ້ອງກັບຄົນຫຼາຍລະດັບຕັ້ງແຕ່ລູກຄ້າ ຫຼື ຜູ້ໃຊ້ ແລະ ນັກທຸລະ ກິດ.

### ສັນຍາລັກໃນການແຕ້ມ Data Flow Diagram ແລະ ກົດຂອງ Process

|  |  |  |
| --- | --- | --- |
| **ຊື່** | **ສັນຍາລັກ** | **ຄວາມຫມາຍ** |
| Boundary Or External Entity |  | ຂອບເຂດຫມາຍເຖິງພາກ ສ່ວນທີ່ກ່ຽວຂ້ອງເກັບລະບົບເຊິ່ງລະບົບບໍ່ສາມາດຄວບຄຸມໄດ້ |
| Process |  | ປະມວນຜົນຫຼືຫນ້າວຽກທີ່ເຮັດໃນໂຄງການນັ້ນໆ |
| Data Store |  | ບ່ອນຈັດເກັບຂໍ້ມູນ |
| Data Flow |  | ການໄຫຼຂອງຂໍ້ມູນ |
| Real-Time Link |  | ການເຊື່ອມໂຍງແບບໄກທີ່ມີການຕອບກັບແບບທັນທີທັນໃດ |

ຕາຕະລາງທີ 2 ສະແດງສັນຍາລັກ **Data Flow Diagram**

1. **ຈຸດປະສົງຂອງ DFD**

* ເປັນແຜນພາບທີ່ສະຫຼຸບລວມຂໍ້ມູນທັງໝົດໄດ້ຈາກການວິເຄາະໃນລັກສະນະຂອງຮູບແບບທີ່ ເປັນໂຄງສ້າງ.
* ເປັນຂໍ້ຕົກລົງຮ່ວມກັນລະຫວ່າງນັກວິເຄາະລະບົບ ແລະ ຜູ້ຊົມໃຊ້.
* ເປັນແຜນພາບທີ່ໃຊ້ໃນການພັດທະນາຕໍ່ໃນຂັ້ນຕອນຂອງການອອກແບບ.
* ຮູ້ທີ່ໄປທີ່ມາຂອງຂໍ້ມູນທີ່ໄຫຼໃນຂະບວນການຕ່າງໆ.

1. **ກົດຂອງ Process**

* ຕ້ອງບໍ່ມີຂໍ້ມູນເຂົ້າພຽງຢ່າງດຽວ
* ຕ້ອງບໍ່ມີຂໍ້ມູນອອກພຽງຢ່າງດຽວ
* ຂໍ້ມູນທີ່ສົ່ງເຂົ້າຕ້ອງພຽງພໍໃນການສ້າງຖານຂໍ້ມູນທີ່ສົ່ງອອກ
* ການຕັ້ງຊື່ Process ຕ້ອງໃຊ້ຄໍາກິລິຍາ(Verb) ເຊັ່ນ: ບັນທຶກຂໍ້ມູນ,ໃບບິນ,ກວດສອບຂໍ້ມູນລູກຄ້າ,ຈໍານວນເງິນເດືອນ.

### ປຽບທຽບແຜນວາດການໄຫຼຂໍ້ມູນທີ່ບໍ່ຖືກຕ້ອງ ແລະ ຖືກຕ້ອງ

1. **Process:** 
   * ເມື່ອມີຂໍ້ມູນເຂົ້າໄປທີ່ Processກໍ່ຕ້ອງມີຂໍ້ມູນ ຫຼື ຜົນຮັບອອກມາຈາກ Processເຊັ່ນກັນຈະເປັນໄປບໍ່ໄດ້ທີ່ມີສະເພາະຂໍ້ມູນເຂົາຢ່າງດຽວ.
2. **Data store:**

* ຂໍ້ມູນຈະໄຫຼຈາກData storeໜຶ່ງໄປຫາ Data storeໜຶ່ງໂດຍກົງບໍ່ໄດ້ຈະຕ້ອງຜ່ານProcessເທົ່ານັ້ນ
* ຂໍ້ມູນທີ່ສົ່ງຜ່ານ External entityບໍ່ສາມາດໄຫຼເຂົ້າໄປ Data storeໂດຍກົງໄດ້ຈະຕ້ອງໃຊ້ Processເປັນຕົວກາງໃນການເຊື່ອມໂຍງເພື່ອຈັດເກັບຂໍ້ມູນໃນ Data store
* ຂໍ້ມູນທີ່ໄຫຼຜ່ານຈາກ Data storeບໍ່ສາມາດເຊື່ອໂຍງເຂົ້າກັບ External entityໄດ້ໂດຍກົງຈະຕ້ອງຜ່ານ Processເທົ່ານັ້ນ.

1. **External entity:**

* External entityບໍ່ສາມາດເຊື່ອມໂຍງເຂົ້າຫາກັນໄດ້ຈະຕ້ອງໃຊ້ Processເປັນຕົວກາງເພື່ອ  
  ສົ່ງຜ່ານ ແລະ ຊື່ຂອງ External entityຈະໃຊ້ຄໍານາມເທົ່ານັ້ນ.

1. **Data flow:**

* ການໄຫຼຂໍ້ມູນທີ່ມີຫົວຊີ້ໄປທີ່ Processໝາຍເຖິງ Processມີການອ່ານ ຫຼື ການດືງຂໍ້ມູນຈາກ Data storeມາໃຊ້ວຽກ
* ການໄຫຼຂໍ້ມູນຈາກ Process ທີ່ມີຫົວລູກສອນຊີ້ໄປຍັງData storeໝາຍເຖິງການ Updateຫຼື ການເພີ່ມຂໍ້ມູນລົງໄປທີ່ Data store

|  |  |
| --- | --- |
| ອະນຸຍາດ | ບໍ່ອະນຸຍາດ |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

ຕາຕະລາງທີ 10 ຮູບການປຽບທຽບແຜນວາດການໄຫຼຂໍ້ມູນທີ່ບໍ່ຖືກຕ້ອງ ແລະ ຖືກຕ້ອງ

* ການໄຫຼຂໍ້ມູນທີ່ມີຫົວລູກສອນທັງສອງດ້ານທີ່ເຊື່ອມໂຍງລະຫວ່າງ Processກັບ Data storeໝາຍເຖິງມີການດືງຂໍ້ມູນຈາກ Data storeມາປັບປຸງ ແລະ ມີການUpdateຂໍ້ມູນລົງໄປໃນ Data store
* ການໄຫຼຂໍ້ມູນບໍ່ສາມາດຢ້ອນກັບໄປຍັງ Processເດີມໄດ້ ຢ່າງໝ່ອຍຕ້ອງເຊື່ອມໂຍງຜ່ານProcessໜື່ງເພື່ອສົ່ງຜ່ານຢ້ອນກັບມາຍັງ Processເດີມ ແລະ ຊື່ທີ່ລະບຸໃນການໄຫຼຂໍ້ມູນຈະໃຊ້ຄໍານາມ

### ຄວາມຮູ້ພື້ນຖານກ່ຽວກັບ ພາສາ NOSQL

NoSQL ຫຼື Not Only SQL ໝາຍ​ເຖິງ​ການທີ່ [database](https://saixiii.com/what-is-database/)ໃຊ້​ວິທີ​ການສ້າງ​ຖານ​ຂໍ້​ມູນ​ ແລະ​ການ​ດຶງ​ຂໍ້​ມູນ ທີ່​ແຕ່ງ​ຕາກ​ຈາກ​ຮູບ​ແບບ​ປົກກະຕິ ([SQL](https://saixiii.com/sql-statement/)) ທີ່​ອອກ​ມາ​ໃນ​ຮູບ​ແບບ​ຕາ​ຕະລາງ​ເຊັ່ນ​ດຽວ​ກັບ [relational databases](https://saixiii.com/what-is-database/)  ຊຶ່ງ​ຕັ້ງ​ແຕ່​ຊ່ວງ 1960 ທີ່ database ໄດ້​​ກຳ​ເນີດ​ຂຶ້ນ​ແຕ່ NoSQL ກໍ​ຍັງ​ບໍ່​ເປັນ​ທີ່​ນິຍົມ ຈົນ​ກະ​ທັງ​ຊ່ວງ​ປີ 2000 NoSQL ເລີ່ມ​ຖືກ​ເພີ່ມ​ເຂົ້າ​ມາ​ໃນ Web2.0 ເຮັດໃຫ້​ຫຼາຍ​ບໍລິສັດ​ຊັ້ນ​ນຳ​ເຊັ່ນ Facebook, Google ແລະ Amazon.com ໄດ້​ເລີ່ມ​ນຳ​ມາ​ໃຊ້ ສ່ວນ​ຫລາຍ​ຖືກ​ນຳ​ມາ​ໃຊ້​ງານ​ກ່ຽວກັບ big data ແລະ real-time web applications ນັ້ນ​ຄື​ທີ່​ມາ​ຂອງ NoSQL ໂດຍ​ປົກກະຕິ​ແລ້ວ NoSQL ງ່າຍ​ຕໍ່​ການ​ຂະ ຫຍາຍ​ລະບົບ ທີ່​ເປັນ​ຮູບ​ແບບ cluster (ແນ່​ນອນ) ຊຶ່ງ​ນີ້​ເປັນ​ຈຸດ​ອ່ອນ​ຂອງ relational database ຫຼັກ​ການ​ຂອງ NoSQL ຄື​ໃຊ້​ໂຄງ​ສ້າງ​ຂອງ​ຂໍ້​ມູນ​ໄດ້​ຫຼາຍ​ຮູບ​ແບບ​ key-value, wide column, graph ຫລື document ຊຶ່ງ​ຈະ​ແຕກ​ຕ່າງ​ຈາກ​ຮູບ​ແບບ​ຂອງ relational database ຊຶ່ງ​ບາງເທື່ອ​ເຮັດໃຫ້ NoSQL ມີ​ຄວາມ​ວ່ອງໄວ​ຫລາຍກວ່າ

* **ປະ​ເພດ​ຂອງ NoSQL**

ມີ​ການ​ຈຳ​ແນກ​ໄດ້​ຫຼາຍ​ຮູບ​ແບບ​ສຳລັບ NoSQL database ໂດຍ​ແຕ່​ລະ​ຮູບ​ແບບ​ໄດ້​ມີ​ການ​ຈັດ​ກຸ່ມ ​ແລະ​ກຸ່ມ​ຍ່ອຍ​ໄວ້ ບາງເທື່ອ​ອາດຈະ​ມີ​ບາງ​ຊະນິດ​ກາບກ່ຽວ​ກັນ​ດ້ວຍເຊັ່ນ:

**ກ. Key-value store**

ເຮັດວຽກ​ໃນ​ຮູບ​ແບບ array (Map ຫລື Dict) ຊຶ່ງ​ເປັນ​ໂຄງ​ສ້າງ​ພື້ນ​ຖານ​ທີ່​ສ້າງ​ຄວາມ​ສຳພັນ​ເປັນ​ຄູ່ ຊຶ່ງ Key ຈະ​ຕ້ອງ​ບໍ່​ຊ້ຳ​ກັນ ຮູບ​ແບບ​ນີ້​ຄື​ແບບ​ທີ່​ງ່າຍ​ທີ່ສຸດ ແລະ ມີ​ຄວາມ​ສາມາດ​ໃນ​ການ​ປະ​ມວມ​ຜົນ​ທີ່​ສູງ​ຫລາຍ

**ຂ. Document store**

ຮູບ​ແບບ​ຂອງ document store ຄື​ການ ເຂົ້າ​ລະ​ຫັດ​ຂໍ້​ມູນ​ດ້ວຍ​ຮູບ​ແບບ format ມາດຕາ​ຖານ​ເຊັ່ນ [XML](https://saixiii.com/what-is-xml/),YAML,[JSON](https://saixiii.com/what-is-json/) ຫຼື binary ແລະ​ເກັບ​ໄວ້​ໃນ database ດ້ວຍ unique key ທີ່​ເຊື່ອມ​ກັບ​**ຂໍ້​**ມູນ ຄ້າຍ​ຮູບ​ແບບ​ຂອງ key-value ໂດຍ database ຈະ​ມີ API ຫລື ພາສາ​ທີ່​ໃຊ້ query ຂໍ້​ມູນສາມາດ​ຕິດ​ຕັ້ງ​ໄດ້​ຫຼາຍ​ຮູບ​ແບບ ຕາມ​ກຸ່ມ documents

* Collections
* Tags
* Non-visible metadata
* Directory hierarchies

ຖ້າ​ທຽບ​ກັບ relational database ຈະ​ເຫັນ​ວ່າ ແຕ່​ລະ record ສາມາດ​ມີ​ຈຳນວນ field ບໍ່​ເທົ່າ​ກັນ​ໄດ້ ຊຶ່ງ​ບໍ່​ຄືກັບ​ຮູບ​ແບບ​ຕາ​ຕະລາງ

**ຄ. Graph**

ຖືກ​ອອກ​ແບບ​ສຳລັບ​ຂໍ້​ມູນ​ທີ່​ມີ​ຕ້ອງ​ສະແດງ​ເປັນ graph ທີ່​ປະກອບ​ໄປ​ດ້ວຍ​ຂໍ້​ມູນ​ຕັວ​ເລກ​ນັບ​ບໍ່​ຖ້ວນ ເຊັ່ນ Social Relations, link ຂອງ​ການ​ຂົ່ນ​ສົ່ງ, ຖະໜົນ​ແຜນ​ທີ່ ຫຼື ລະບົບ network

### ຄວາມຮູ້ກ່ຽວກັບ HTML

HTML ແມ່ນພາສາທີ່ໃຊ້ໃນການຂຽນ Web page ທີ່ສາມາດສະແດງຜົນເຖິງ Website ຫຼື່ເຮົາເອີ້ນວ່າ: web page ຖືກພັດທະນາ ແລະ ກໍານົດມາດຕະຖານທີ່ແນ່ນອນໂດຍອົງກອນ World Wide Web Consortium (W3C) ແລະ ຈາກການພັດທະນາທາງດ້ານ Software ເຮັດໃຫ້ HTML ເປັນອີກພາສາໜຶ່ງທີ່ຂຽນໂປຣແກມ HTML Application ເປັນພາສາປະເພດຂອງ Markup ສຳລັບການສ້າງ web page ໂດຍນໍາໃຊ້ໂປຣແກມຂຽນໂຄດ ແມ່ນໂປຣແກມ Dreamweaver, Notepad, Editplus, ສ່ວນການເອີ້ນໃຊ້ງານ ຫຼື່ ທົດສອບການເຮັດວຽກຂອງເວັບຈະໃຊ້ໂປຣແກມ Web Browser ເຊັ່ນ: IE Microsoft, internet Explorer(IE),Google, Chrome, Mozilla Firefox, Safari, Opera, Netscape Navigator ເປັນຕົ້ນ.

ຈຸດເດັ່ນຂອງ HTML ປະກອບມີ Semantic Markup (ໂຄດເປັນລະບຽເຮັດໃຫ້ Search Engine ເຫັນຂໍ້ມູນໄດ້ງ່າຍ),From Enhancement(ເພີ່ມປະສິດທິພາບຂອງຟອມ),ສຽງ ແລະ ວິດີໂອ.

### ຄວາມຮູ້ກ່ຽວກັບ CSS

CSS (Cascading Style Sheet) ເປັນພາສາທີ່ໃຊ້ຈັດການຮູບແບບເອກະສານ HTML ໃຫ້ມີຄວາມສວຍງາມ ມີຄຸນສົມບັດເຊັ່ນ: ການກໍານົດຂອບ,ສີ,ເສັ້ນ,ພື້ນຫຼັງ,ໄລຍະຫາງ ແລະ ຮູບແບບຕົວໜັງສືຕ່າງໆເຊັ່ນ: ການສະແດງຜົນເຖິງໜ້າຈໍ,ການສະແດງຜົນເຖິງມືຖື ແລະ ຊ່ວຍຫຼຸດການໃຊ້ພາສາ HTML ໃນການຕົບແຕ່ງ web page ເຮັດໃຫ້ Code ເຫຼື່ອສ່ວນເນື້ອໃນທີ່ເຂົ້າໃຈງ່າຍ,ການແກ້ໄຂງ່າຍຂຶ້ນ ແລະ ຂະໜາດຟາຍນ້ອຍລົງ ດາວໂຫຼດໄດ້ງ່າຍຂຶ້ນ ແລະ ທີ່ສຳຄັນຫັຍງສາມາດຄວບຄຸມການສະແດງຜົນໃຫ້ຄືກັນໄດ້ຫຼາຍໆ web Bowser.

### ຄວາມຮູ້ກ່ຽວກັບພາສາ JavaScript

ພາສາ JavaScript ຫລື​ຫຍໍ້ JS ເປັນ​ພາສາ​ຂຽນ​ໂປຣ​ແກຣມ​ທີ່​ຖືກ​ພັດທະນາ​ແລະ​ປະຕິບັດ​ຕາມ​ຂໍ້​ກຳນົດ​ມາດຕະຖານ​ຂອງ ECMAScript; ພາສາ JavaScript ນັ້ນ​ເປັນ​ພາສາ​ລະ​ດັບ​ສູງ ຄອມ​ພາຍ​ໃນ​ຂະນະ​ທີ່​ໂປຣ​ແກຣມ​ຣັນ(JIT) ແລະ​ ເປັນ​ພາສາ​ຂຽນ​ໂປຣ​ແກຣມ​ແບບ​ຫຼາຍ​ຂະ​ບວນ​ເຊັ່ນ: ການ​ຂຽນ​ໂປຣ​ແກຣມ​ແບບ​ຂັ້ນ​ຕອນ, ການ​ຂຽນ​ໂປຣ​ແກຣມ​ແບບ​ວັດ​ຖຸ, ພາສາ JavaScript ມີ​ໄວ​ຍະກອນທີ່​ຄື​ກັບ​ພາສາ C ໃຊ້​ວົງ​ເລັບ​ເພື່ອ​ກຳນົດ​ບ່ອກ​ຂອງ​ຄຳ​ສັ່ງ ນອກ​ຈາກ​ນີ້ JavaScript ຢັງ​ເປັນ​ພາສາ​ທີ່​ມີ​ປະ​ເພດ​ຂໍ້​ມູນ​ແບບ​ໄດ​ນາ​ມິກ (Dynamic)​ ເປັນ​ພາສາ​ແບບ Prototype-based ແລະ First-class function.

ພາສາ JavaScript ນັ້ນ​ຖື​ວ່າ​ເປັນ​ເທັກ​ໂນ​ໂລ​ຊີ​ຫຼັກ​ຂອງ​ການ​ພັດທະນາ​ເວັບ​ໄຊ​ (World Wide Web) ມັນ​ເຮັດໃຫ້​ໜ້າ​ເວັບ​ສາມາດ​ຕອບ​ໂຕ້​ກັບ​ຜູ້​ໃຊ້​ໄດ້​ໂດຍ​ທີ່​ບໍ່​ຈຳ​ເປັນ​ຕ້ອງ​ລີເຟດໜ້າ​ໃໝ່ (Dynamic website) ເວັບ​ໄຊ​ຈຳນວນ​ຫຼາຍ​ໃຊ້​ພາສາ JavaScript ສຳລັບ​ຄວບ​ຄຸມ​ການ​ເຮັດ​ວຽກງານ​ທີ່​ດ້ານ Client-side ນັ້ນ​ເຮັດໃຫ້​ເວັບ​ເບ​ຣາ​ວ​ເຊີຕ່າງ​ໆ ມີ JavaScript engine ທີ່​ໃຊ້​ສຳລັບ​ປະ​ມວນ​ຜົນສະຄິບ​ຂອງ​ພາສາ JavaScript ທີ່​ຣັນ​ເທິງ​ເວັບ​ບາວ​ເຊີເນື່ອງ​ຈາກ​ພາສາ JavaScript ເປັນ​ພາສາ​ຂຽນ​ໂປຣ​ແກຣມ​ແບບ​ຫຼາຍ​ຮູບແບບ ເຮັດໃຫ້​ມັນ​ຮອງ​ຮັບ​ການ​ຂຽນ​ໂປຣ​ແກຣມ​ທັງ​ແບບ Event-driven, Functional ແລະ ​ແບບ​ລຳ​ດັບ​ຂັ້ນ​ຕອນ ມັນ​ມີ​ Library (APIs) ສຳລັບ​ເຮັດວຽກ​ກັບ​ຂໍ້ຄວາມ, ວັນທີ, Regular expression ແລະ ​ໂຄງ​ສ້າງ​ຂໍ້​ມູນ​ພື້ນ​ຖານ​ຢ່າງ Array ແລະ Map ຫຼື ​ແມ່ນ​ທັງ Document Object Model (DOM) ຊຶ່ງ​ເປັນ API ທີ່​ໂດຍ​ທັ່ວ​ໄປ​ແລ້ວ​ສາມາດ​ໄດ້​ເທິງ​ເວັບບາວເຊີ.

ຢ່າງໃດ​ກໍ​ຕາມ ຕົວ​ຂອງ​ພາສາ JavaScript ເອງ​ບໍ່​ໄດ້​ມີ​ຟັງ​ຊັນ​ສຳລັບອິນພຸດ/ເອົາພຸດ(I/O) ທີ່​ມາ​ກັບ​ພາສາເຊັ່ນ ຟັງ​​ຊັນ​ກ່ຽວກັບ Network ວຽກກ່ຽວ​ກັບ​ໄຟ​ລ ຫຼື Library​ ກ່ຽວກັບ​ກຣາ​ຟິກ ໂດຍ​ທົ່ວ​ໄປ​ແລ້ວ​ສິ່ງເຫຼົ່ານີ້​ຈະ​ຖືກ​ໃຫ້​ມາ​ໂດຍ Host environment (ສະພາບ​ແວດ​ລ້ອມ​ທີ່​ໃຊ້​ຣັນ​ພາສາ JavaScript) ເຊັ່ນ ເວັບ​​ເວັບບາວເຊີ ຫຼື Node.js ຊຶ່ງ​ຈະ​ແຕກ​ຕ່າງ​ກັນ​ອອກ​ໄປ ​ຕົວ​ຢ່າງ​ເຊັ່ນ ການ​ຮັບ​ຄ່າ​ໃນ​ເວັບ​​ເວັບບາວເຊີ​ຈະ​ຜ່ານ​ຟັງ​ຊັນ prompt ຊຶ່ງ​ເປັນ​ສ່ວນຫນຶ່ງ​ຂອງ Browser Object Model (BOM) ຫລື​ຮັບ​ຄ່າ​ຈາກ HTML ຟອມຊຶ່ງ​ເປັນ​ສ່ວນຫນຶ່ງ​ຂອງ Document Object Model (DOM) ໃນຂະນະ​ທີ່​ເທິງ Node.js ເຮົາ​ສາມາດ​ຮັບ​ຄ່າ​ໄດ້​ຈາກ Input/Output Stream ຂອງ Command line ເຖິງ​ແມ່ນ​ວ່າ​ມັນ​ຈະ​ມີ​ຄວາມ​ຄ້າຍ​ຄືກັນ​ລະຫວ່າງ​ພາສາ Java ແລະ JavaScript ເຊັ່ນ: ຊື່​ຂອງ​ພາສາ ໄວ​ຍະກອນຫລື​ Library​ ​ມາດຕະຖານ​ຕ່າງ​ໆ ຢ່າງໃດ​ກໍ​ຕາມ ທັງ​ສອງ​ພາສາ​ແຕກ​ຕ່າງ​ກັນ​ຢ່າງ​ສິ້ນ​ເຊີງ​ໃນ​ແງ່​ຂອງ​ການ​ອອກ​ແບບ ພາສາ Java ເປັນ​ພາສາ​ທີ່​ມີ​ປະ​ເພດ​ຂໍ້​ມູນ​ແບບ​ຄົງ​ທີ່ (Static-typing) ໃນ​ຂະນະ​ທີ່​ພາສາ JavaScript ມີ​ປະ​ເພດ​ຂໍ້​ມູນ​ແບບ​ໄດ​ນາ​ມິກ​ (Dynamic-typing) ພາສາ Java ຖືກ​ຄອມ​ພາຍເປັນ Byte-code ກ່ອນທີ່ຈະ​ລັນ ໃນ​ຂະນະ​ທີ່​ພາສາ JavaScript ຈະ​ຄອມ​ພາຍໃນ​ຕອນ​ທີ່​ໂປຣ​ແກຣມ​ລັນພາສາ Java ເປັນ​ພາສາ​ແບບ Class-based ໃນ​ຂະນະ​ທີ່​ພາສາ JavaScript ເປັນ​ພາສາ​ແບບ Prototypebased

****

ຮູບທີ 10 ພາສາ JavaScript

**ກ. ປະຫວັດ​ຄວາມ​ເປັນ​ມາ​ຂອງ​ພາສາ JavaScript**

ພາສາ JavaScript ຖືກ​ອອກ​ແບບ​ແລະ​ສ້າງ​ໂດຍ Brendan Eich ສຳລັບ​ເປັນ​ພາສາ​ສະຄິບ​ທີ່​ເຮັດວຽກ​ເທິງ​ເວັບ​ເບ​ຣາ​ວ​ເຊີ Navigator ທີ່​ເປັນ​ຜະລິດຕະພັນ​ຂອງ​ບໍລິສັດ Netscape ເພື່ອ​ເຮັດໃຫ້​ໜ້າ​ເວັບ​ທີ່​ໃນ​ຕອນ​ທຳອິດ​ນັ້ນ​ເປັນ​ແບບ Static ສາມາດ​ຕອບ​ໂຕ້​ກັບ​ຜູ້​ໃຊ້​ໄດ້​ໂດຍ​ທີ່​ບໍ່​ຈຳ​ເປັນ​ຕ້ອງ​ຣີ​ເຟ​ດ​ໜ້າ​ໃໝ່ (Dynamic) ເຊັ່ນ ການ​ສົ່ງ​ຂໍ້​ມູນ​ເບື້ອງ​ຫຼັງ​ໄປ​ຢັງ Server ແລະ​ລໍຖ້າ​ຮັບ​ຜົນ​ຕອບ​ກັບ​ມາ​ດ້ວຍ AJAX; ພາສາ JavaScript ໄດ້​ຖືກ​ເປີດ​ຕົວ​ ແລະ ​ເປັນ​ສ່ວນຫນຶ່ງ​ຂອງ​ເວັບ​ບ​າ​ວ​ເຊີ Navigator ໃນ​ເດືອນ​ກັນຍາ 1995 ໂດຍ​ໃຊ້​ຊື່​ວ່າ LiveScript ແລະ ​ໄດ້​ປ່ຽນ​ເປັນ JavaScript ໃນ​ອີກ​ສາມ​ເດືອນ​ຕໍ່​ມາ.

ໃນ​ເດືອນ​ພ​ະຈິກ 1996 Netscape ໄດ້​ສົ່ງ​ພາສາ JavaScript ໄປ​ຢັງ ECMA International ເພື່ອ​ເປັນ​ຈຸດ​ເລີ່ມ​ຕົ້ນ​ສຳລັບ​ກຳນົດ​ມາດຕະຖານ​ໃຫ້​ທຸກ​ເວັບ​ບາວເຊີປະຕິບັດຕາມ​ມາດຕະຖານ​ດັ່ງ​ກ່າວ ເພື່ອ​ໃຫ້​ການ​ພັດທະນາ JavaScript engine ເປັນ​ໄປ​ໃນ​ທິດ​ທາງ​ດຽວ​ກັນ ນັ້ນ​ໃຫ້​ເກີດ​ການ​ເປີດ​ຕົວ​ຢ່າງ​ເປັນ​ທາງ​ການ​ສຳລັບ​ຂໍ້​ກຳນົດ​ມາດຕະຖານ ECMAScript ໃນ​ເດືອນ​ມິ​ຖຸ​ນາ 1997 ໃນ​ຊ່ວງ​ເວລາ​ຫຼັງ​ຈາກ​ນີ້ ບໍລິສັດ​ຕ່າງ​ໆ ທີ່​ພັດທະນາ​ເວັບ​ບາ​ວ​ເຊີຕ່າງ​ກໍ​ຍັງ​ພັດທະນາ JavaScript engine ບໍ່​ເປັນ​ໄປ​ໃນ​ທິດ​ທາງ​ດຽວ​ກັນ​ເທົ່າໃດ ນັ້ນ​ເຮັດໃຫ້​ນັກ​ພັດທະນາ​ເວັບ​ຕ້ອງ​ຂຽນ​ໂຄດຫຼາຍ​ເວີຊັນ​ເພື່ອ​ໃຫ້​ເຮັດວຽກ​ໄດ້​ໃນ​ທຸກ​ເວັບ​ບ​າ​ວ​ເຊີຈົນກະທັ້ງ​ໃນ​ເ​ດືອນກໍລະກົດ 2008 ໄດ້​ມີ​ການ​ຈັດການ​ປະ​ຊຸມ​ຂຶ້ນ​ທີ່ Oslo ຈາກ​ອົງ​ກ​ອນ​ແລະ​ຝ່າຍ​ຕ່າງ​ໆ ທີ່​ພັດທະນາ JavaScript engine ເຮັດໃຫ້​ເກີດ​ຂໍ້​ຕົກລົງ​ຂຶ້ນ​ໃນ​ຕົ້ນ​ປີ 2009 ເພື່ອ​ລວບລວມ​ງານ​ທີ່​ກ່ຽວ​ຂ້ອງ​ທັງ​ໝົດ​ຂອງ​ພາສາ JavaScript ແລະ ​ຊຸກຍູ້​ພາສາ​ໃຫ້​ຍ່າງ​ໄປ​ຂ້າງ​ໜ້າ ນັ້ນ​ເຮັດໃຫ້​ເກີດ​ຂໍ້​ກຳນົດ​ມາດຕະຖານ ECMAScript ເວີຊັນ​ທີ່ 5 (ES5) ອອກ​ມາ​ໃນ​ເດືອນ​ທັນວາ 2009 ແລະ ​ກ່ອນ​ໜ້າ​ນີ້​ໃນ​ປີ 2008 Google ໄດ້​ເປີດ​ຕົວ​ເວັບ​ບ​າ​ວ​ເຊີ Chrome ທີ່​ມາ​ພ້ອມ​ກັບ V8 JavaScript engine ທີ່​ມີ​ແນວ​ຄິດ​ໃນ​ການ​ພັດທະນາ​ແບບ​ຄອມ​ພາຍ​ໃນ​ຕອນ​ທີ່​ໂປຣ​ແກຣມ​ລັນ (Just-in-time compilation: JIT) ຊຶ່ງ​ມັນ​ເຮັດວຽກ​ໄດ້​ໄວ​ກວ່າ​ຫຼາຍ ເຮັດໃຫ້​ຜູ້​ພັດທະນາ​ເວບບ​າ​ວ​ເຊີ​ອື່ນ​ໆ ຕ້ອງ​ປັບ​ປຸງ JavaScript engine ຂອງ​ພວກ​ເຂົາ​ໃຫ້​ເຮັດວຽກ​ໃນ​ຮູບ​ແບບ JIT ຫຼັງ​ຈາກ​ທີ່​ພັດທະນາ​ຕໍ່​ເນື່ອງ​ມາ​ອີກ​ຫຼາຍ​ປີ ໃນ​ປີ 2015 ໄດ້​ມີ​ການ​ເພີ່ມ​ຄຸນສົມບັດ​ໃໝ່​ໆ ທີ່​ຫຼາກ​ຫຼາຍ​ເຂົ້າ​ມາ ຊຶ່ງ​ຖື​ວ່າ​ເປັນ​ການ​ປ່ຽນ​ແປງ​ເທື່ອ​ສຳຄັນ ແລະ​ເຮັດໃຫ້​ເກີດ​ຂໍ້​ກຳນົດ​ມາດຕະຖານ ECMAScript 2015 ຫຼື ​ເວີຊັນທີ່ 6 (ES6) ຈົນກະທັ້ງ​ໃນ​ປີ 2015 ຕອນ​ນີ້​ເບິ່ງ​ຄື​ວ່າ​ພາສາ JavaScript ຈະ​ພັດທະນາ​ມາ​ຈົນ​ເຖິງ​ທີ່ສຸດ​ແລ້ວ ເຮັດໃຫ້​ລະຫວ່າງ​ປີ 2016 - 2019 ເວີຊັນ​ໃໝ່​ຂອງ ECMAScript ທີ່​ຖືກ​ເຜີຍແຜ່​ອອກ​ມາ​ໃນ​ແຕ່​ລະ​ປີ​ມີ​ການ​ປ່ຽນ​ແປງ​ແລະ​ເພີ່ມ​ຄຸນສົມບັດ​ພຽງ​ນ້ອຍໆ​ເທົ່າ​ນັ້ນ.

**ຂ. ຄຸນສົມບັດ​ຂອງ​ພາສາ JavaScript**

ECMAScript 2015 (ES6) ເປັນ​ພາສາ JavaScript ທີ່​ຖື​ວ່າ​ພັດທະນາ​ມາ​ຈົນ​ເຖິງ​ຈຸດ​ສູງ​ສຸດ​ແລ້ວ​ກໍ​ວ່າ​ໄດ້ ມັນ​ຖືກ​ເຜີຍແຜ່​ໃນ​ເດືອນ​ມິ​ຖຸ​ນາ 2015 ຊຶ່ງ​ໃນ​ເວີຊັນ​ນີ້ ໄດ້​ເພີ່ມ​ໄວ​ຍະກອນໃໝ່​ຂອງ​ພາສາ​ຫຼວງ​ຫລາຍເຊັ່ນ ການສ້າງ​ຄາດ​ດ້ວຍ​ຄຳ​ສັ່ງ class ການສ້າງ​ໂມ​ດູນ​ແລະ​ໃຊ້​ງານ​ມັນ​ດ້ວຍ​ຄຳ​ສັ່ງ import ແລະ export ແລະ​ ຄຳ​ສັ່ງ​ສຳລັບ​ປະ​ກາດ​ຕົວ​ປ່ຽນ let ແລະ ​ປະ​ກາດ​ຄ່າ​ຄົງທີ const ຊຶ່ງ​ເຮັດໃຫ້​ຕົວ​ປ່ຽນສາມາດ​ມີ​ຂອບ​ເຂດ​ໃນ​ບ​ລັອກ​ທີ່​ມັນ​ຖືກ​ສ້າງ​ຂຶ້ນ​ໄດ້ ແລະ​ສິ່ງ​ອື່ນ​ໆ ທີ່​ຖືກ​ເພີ່ມ​ເຂົ້າ​ມາ​ເປັນ​ຈຳນວນ​ຫລາຍ ເຊັ່ນ Map, Set, WeakMap, Promise, Reflection, Proxies, Template string ແລະ​ອື່ນ​ໆ.

ໃນ​ເດືອນ​ມິ​ຖຸ​ນາ 2016 ໄດ້​ມີ​ການ​ເປີດ​ຕັວ​ເວີຊັນ 7 ຫລື ECMAScript 2016 (ES7) ໄດ້​ມີ​ການ​ເພີ່ມ​ຕົວ​ດຳ​ເນີນ​ການ​ຍົກ​ກຳ​ລັງ \*\* (ທີ່​ກ່ອນ​ໜ້າ​ນີ້​ເຮົາ​ຈະ​ໃຊ້​ຜ່ານ​ຟັງຊັນ Math.pow) ຄຳ​ສັ່ງawait async ສຳລັບ​ການ​ຂຽນ​ໂປຣ​ແກຣມ​ທີ່​ເຮັດວຽກ​ບໍ່​ພ້ອມ​ກັນ ແລະ ​ຟັງຊັນ includes ຂອງ​ອາ​ເລແລະ ​ໃນ​ປະຈຸບັນ ພາສາ JavaScript ຖືກ​ພັດທະນາ​ມາ​ຈົນ​ເຖິງ ECMAScript 2020 (ES11) ຊຶ່ງ​ມີ​ການ​ປ່ຽນ​ແປງ​ທີ່​ເພີ່ມ​ຂຶ້ນ​ບໍ່​ເທົ່າໃດ​ຫຼັງ​ຈາກ ES7

**ຄ. JavaScript engine ແມ່ນຫຍັງ?**

JavaScript engine ຄື​ໂປຣ​ແກຣມ​ຄອມພິວເຕີ​ທີ່ໃຊ້​ສຳລັບ​ປະ​ມວນ​ຜົນ​ໂຄດຂອງ​ພາສາ JavaScript ຊຶ່ງ JavaScript engine ໃນ​ຊ່ວງ​ເລີ່ມ​ຕົ້ນ​ເປັນພຽງ​ແຄ່​ຕົວປ່ຽນພາສາ (Interpreter) ເທົ່າ​ນັ້ນ ແຕ່​ໃນ​ປະຈຸບັນ​ໄດ້​ມີ​ການ​ພັດທະນາ​ມາ​ໃຫ້​ຢູ່ໃນ​ຮູບ​ແບບ​ຂອງ​ຄອມ​ພາຍເລິ​ທີ່​ມີ​ການ​ຄອມ​ພາຍໃນ​ຕອນ​ທີ່​ໂປຣ​ແກຣມ​ລັນ (Just-in-time compilation: JIT) ເພື່ອ​ເພີ່ມ​ປະ​ສິດ​ທິ​ພາບ​ການ​ເຮັດ​ວຽກງານ​ຂອງ​ໂປຣ​ແກຣມ ໂດຍ​ທົ່ວ​ໄປ​ແລ້ວ JavaScript engine ຈະ​ຖືກ​ພັດທະນາ​ໂດຍ​ຜູ້​ພັດທະນາ​ເວັບບາວເຊີ້ທີ່​ປະຕິບັດ​ຕາມ​ຂໍ້​ກຳນົດ​ມາດຕະຖານ​ຂອງ ECMAScript

### ຄວາມຮູ້ກ່ຽວກັບຖານຂໍ້ມູນ MongoDB

### ຄວາມຮູ້ກ່ຽວກັບ Nodejs

Node.js ເປັນ open-source ແລະ cross-platform JavaScript runtime environment ທີ່​ກຳ​ລັງ​ໄດ້ຮັບ​ຄວາມ​ນິຍົມ​ສູງ ໂດຍ​ທົ່ວ​ໄປ​ເຮົາ​ຈະ​ໃຊ້ JavaScript ໃນ​ຝັ່ງ client ແຕ່ Node.js ເຮັດໃຫ້​ເຮົາ​ໃຊ້ JavaScript ໃນ​ຝັ່ງ Server ໄດ້​ດ້ວຍ ຊຶ່ງ Node.js ສາມມາດ run​ ໄດ້​ເທິງ platform ທີ່​ຫຼາກ​ຫຼາຍ​ທັງ Windows, Linux, Unix, Mac OS X ແລະ ອື່ນໆ.



**ຮູບທີ 12 Nodejs**

ຈຸດ​ເດັ່ນ​ທີ່ສຸດ​ຂອງ Node.js ຄື​ມັນ​ເຮັດວຽກ​ແບບ asynchronous ຜູ້​ອ່ານ​ອາດຈະ​ສົງ​ໃສ​ວ່າ​ແລ້ວ​ມັນ​ດີ​ແນວໃດ? ລອງ​ມາ​ເບິ່ງ​ຕົວ​ຢ່າງ​ການ​ຈັດການ​ໄຟ​ລ​ເມື່ອ​ມີ​ການ​ຮ້ອງ​ຂໍ​ຈາກ client ມາ​ທີ່ server ຂອງ PHP ກັບ Node.js ທຽບ​ກັນ​ເບິ່ງ​ເພື່ອ​ໃຫ້​ເຂົ້າໃຈ​ຫລາຍ​ຂຶ້ນ​

* **ກໍລະນີ PHP ເມື່ອ​ມີ​ການ​ຮ້ອງ​ຂໍ​ເຂົ້າ​ມາ​ມັນ​ຈະ​ເຮັດ​ດັ່ງ​ນີ້:**

1. ສົ່ງ task ໄປ​ທີ່​ລະບົບ​ໄຟ​ຂອງ​ຄອມພິວເຕີ
2. ລໍຖ້າຈົນກະທັ້ງ​ລະບົບ​ໄຟ​ລ​ເປີດ​ແລະ​ອ່ານ​ໄຟ​ລ​ສຳເລັດ
3. ສົ່ງ​ເນື້ອ​ຫາ​ຂອງ​ໄຟ​​ກັບ​ມາ​ໃຫ້ client
4. ພ້ອມ​ສຳລັບ​ຮັບ​ຄຳ​ຮ້ອງ​ຂໍ​ຖັດ​ໄປ

* **ກໍລະນີ Node.js ເມື່ອ​ມີ​ການ​ຮ້ອງ​ຂໍ​ເຂົ້າ​ມາ​ມັນ​ຈະ​ເຮັດ​ດັ່ງ​ນີ້:**

1. ສົ່ງ task ໄປ​ທີ່​ລະບົບ​ໄຟ​ຂອງ​ຄອມພິວເຕີ
2. ພ້ອມ​ສຳລັບ​ຮັບ​ຄຳ​ຮ້ອງ​ຂໍ​ຖັດ​ໄປ
3. ພໍ​ລະບົບ​ໄຟ​ລ​​ເປີດ​ແລະ​ອ່ານ​ໄຟ​ລ​ສຳເລັດ server ຈະ​ສົ່ງ​ເນື້ອ​ຫາ​ຂອງ​ໄຟ​ກັບ​ມາ​ໃຫ້ client

ຈາກ​ຕັວ​ຢ່າງ​ດ້ານ​ເທິງ​ຈະ​ເຫັນ​ວ່າ Node.js ຈະ​ຕັດ​ຂັ້ນ​ຕອນ​ການ​ລໍຖ້າຖີ້ມ​ແລ້ວ​ໄປ​ເຮັດ​ຄຳ​ຮ້ອງ​ຖັດ​ໄປ​ເລຍ ທີ່​ເປັນ​ແບບນີ້​ເພາະ Node.js ຈະ​ run ​ແບບ single-threaded ແລະ​ໃນ​ Libaryມາດຕະຖານ​ກໍ​ຈະ​ມີ​ເຊັດ​ຂອງ asynchronous I/O primitives ທີ່​ຊ່ວຍ​ປ້ອງ​ກັນ​ໂຄດ JavaScript ຈາກ​ການ blocking ເຮັດໃຫ້​ລະບົບ​ຄ່ອງແຄ້ວ ແລະ ​ມີ​ປະ​ສິດ​ທິ​ພາບ​ຫຼາຍ​ຂຶ້ນ

|  |  |
| --- | --- |
| Synchronous vs Asynchronous | |
| Synchronous ຄື​ການ​ run​ໂຄດ​ຕາມ​ລຳ​ດັບ​ທີ່​ເຮົາ​ຂຽນ​ໄວ້ ເຊັ່ນ  alert(1);  alert(2);  alert(3); ຜົນ​ລັບ​​ທີ່​ໄດ້​ຄື​ໜ້າ​ຈໍ​ຈະ​ສະແດງ​ຜົນ 1 ຈາກ​ນັ້ນ​ຈຶ່ງ​ສະແດງ​ຜົນ 2 ແລ້ວ​ສຸດ​ທ້າຍ​ຈຶ່ງ​ສະແດງ​ຜົນ 3 ຕາມ​ລຳ​ດັບ | Asynchronous ຄື​ການ​ຣັນໂຄ​ດ​ທີ່​ບໍ່​ຈຳ​ເປັນ​ຕ້ອງ​ເປັນ​ໄປ​ຕາມ​ລຳ​ດັບ​ທີ່​ເຮົາ​ຂຽນ​ໄວ້ ເຊັ່ນ:  alert(1);  setTimeout(() => alert(2), 0);  alert(3); ໃນ​ຕົວ​ຢ່າງ​ນີ້​ໂຄດ alert(2) ໃຊ້​ເວລາ​ດຳ​ເນີນ​ການ​ດົນ​ກວ່າ ເຮັດໃຫ້​ຜົນ​ລັບ​​ທີ່​ໄດ້​ຄື​ໜ້າ​ຈໍ​ຈະ​ສະແດງ​ຜົນ 1 ຈາກ​ນັ້ນ​ຈຶ່ງ​ສະແດງ​ຜົນ 3 ແລ້ວ​ສຸດ​ທ້າຍ​ຈຶ່ງ​ສະແດງ​ຜົນ 2 |
| Blocking vs Non-blocking | |
| Blocking ໝາຍ​ເຖິງ​ການ​ທີ່​ເຮົາ​ບໍ່​ສາມາດ​ດຳ​ເນີນ​ການ​ຕໍ່ໄປ​ໄດ້​ຈົນ​ກວ່າ​ຕົວ​ດຳ​ເນີນ​ການ​ທີ່​ກຳ​ລັງ​ລັນ​ຢູ່​ຈະ​ສຳເລັດ​ເສຍ​ກ່ອນ ເຊັ່ນ  alert(1);  var value = localStorage.getItem('foo');  alert(2); ຄຳ​ສັ່ງ localStorage ຈະ​ເປັນ​ຕົວ blocking ເຮັດໃຫ້​ບໍ່​ສາມາດ​ເຮັດ​ຄຳ​ສັ່ງ alert(2) ໄດ້​ຈົນ​ກວ່າ​ມັນ​ຈະ​ດຳ​ເນີນ​ການ​ສຳເລັດ ດັ່ງ​ນັ້ນ​ຜົນ​ລັບ​​ທີ່​ໄດ້​ຄື​ໜ້າ​ຈໍ​ຈະ​ສະແດງ​ຜົນ 1 ຈາກ​ນັ້ນ​ຈຶ່ງ​ລໍຖ້າຈົນກວ່າ​ຄຳ​ສັ່ງ localStorage ຈະ​ສຳເລັດ​ຈຶ່ງ​ສະແດງ​ຜົນ 2 | Non-blocking ໝາຍ​ເຖິງ​ການ​ໂຕທີ່​ດຳ​ເນີນ​ການ​ສາມາດ​ເຮັດ​ຄຳ​ສັ່ງ​ຖັດ​ໄປ​ໄດ້​ເລີຍ​ໂດຍ​ບໍ່​ຕ້ອງ​ລໍຖ້າ​ໃຫ້​ຄຳ​ສັ່ງ​ເດີມ​ເຮັດ​ສຳເລັດ​ກ່ອນເຊັ່ນ:  alert(1);  fetch('example.com').then(() => alert(2));  alert(3); ໃນ​ຕົວ​ຢ່າງ​ນີ້ ຄຳ​ສັ່ງ fetch ເປັນ non-blocking operation ດັ່ງ​ນັ້ນ​ຜົນ​ລັບ​ທີ່​ໄດ້​ຄື​ໜ້າ​ຈໍ​ຈະ​ສະແດງຜົນ 1 ຈາກ​ນັ້ນ​ຈຶ່ງ​ສະແດງ​ຜົນ 3 ແລ້ວ​ພໍ​ຄຳ​ສັ່ງ fetch ສຳເລັດ​ກໍ​ສະແດງ​ຜົນ 2 |

* **​ປະຫວັດ Node.js**

ແຕ່​ເດີມ​ພາສາ JavaScript ຖືກ​ພັດທະນາ​ມາ​ເພື່ອ​ໃຊ້​ສຳລັບ Browser ທີ່​ຊື່ Netscape Navigator ໃນ​ປີ 1995 ​ໃນ​ຕອນ​ນັ້ນ Netscape ຕັ້ງ​ໃຈ​ຈະ​ຂາຍ Web Server ທີ່​ມີ environment ຊື່ Netscape LiveWire ຊຶ່ງ​ສາມາດ​ສ້າງ dynamic page ໂດຍ​ໃຊ້ JavaScript ທາງ​ຝັ່ງ server ດ້ວຍ ແຕ່​ໜ້າ​ເສຍ​ດາຍ​ທີ່ Netscape LiveWire ບໍ່​ປະ​ສົບ​ຄວາມ​ສຳ​ເລັດ ແລະ​ ການ​ໃຊ້ JavaScript ທາງ​ຝັ່ງ server ກໍ​ບໍ່​ໄດ້ຮັບ​ຄວາມ​ນິຍົມ​ເລີຍ​ຈົນກະທັ້ງ Node.js ຖືກ​ກຳ​ເນີດ​ຂຶ້ນ​ມາ.

ສິ່ງ​ທີ່​ເຮັດໃຫ້ Node.js ເປັນ​ທີ່​ນິຍົມ​ຂຶ້ນ​ມາ​ຄື​ການ​ທີ່​ມັນ​ມາ​ໃນ​ຊ່ວງ​ເວລາ​ທີ່​ເໝາະ​ສົມ ເມື່ອ​ທຽບ​ກັບ JavaScript ທີ່​ເກີດ​ມາ​ຕັ້ງ​ແຕ່​ປີ 1995 ແລ້ວ Node.js ຫາກໍ່​ເກີດ​ມາ​ເມື່ອ​ປີ 2009 ເທົ່າ​ນັ້ນ ຕ້ອງ​ຂອບໃຈ "Web 2.0" applications (ເຊັ່ນ Flickr, Gmail ແລະ ອື່ນ ໆ) ທີ່​ສະແດງ​ໃຫ້​ໂລກ​ຮູ້​ວ່າ​ເວັບ​ສະໄໝ​ໃໝ່​ຄວນ​ໜ້າ​ຕາ​ເປັນ​ຢ່າງໃດ

ບໍ່ດົນ​ກ່ອນ​ທີ່ Node.js ຈະ​ເກີດ ນັກ​ພັດທະນາ​ເບື້ອງ​ຫຼັງ browser ຊື່​ດັ່ງ​ທັງ​ຫຼາຍ​ແຂ່ງ​ຂັນ​ກັນ​ເຮັດວຽກ​ຢ່າງ​ໜັກ​ເພື່ອຈະ​ໃຊ້ JavaScript ໃຫ້​ໄດ້​ດີ​ທີ່ສຸດ​ແລະ​ຫາ​ທາງ​ເຮັດໃຫ້ JavaScript ສາມາດ​ລັນ​ໄດ້​ໄວ​ຫລາຍ​ຂຶ້ນ ເພື່ອ​ໃຫ້​ຜູ້​ໃຊ້​ງານ​ໄດ້ຮັບ​ປະ​ສິດ​ທິ​ພາບ​ທີ່​ດີ​ທີ່ສຸດ ຊຶ່ງ​ຜົນ​ຈາກ​ການ​ແຂ່ງ​ຂັນ​ນີ້​ເຮັດໃຫ້​ເກີດ​ການ​ພັດທະນາ Chrome V8 (open-source JavaScript engine ຂອງ The Chromium Project) ຂຶ້ນ​ມາ ແລະ Node.js ກໍ​ໃຊ້ engine ນີ້​​ເອງ

* ແຕ່​ການ​ທີ່ Node.js ເປັນ​ທີ່​ນິຍົມ​ຂຶ້ນ​ມາ​ບໍ່​ແມ່ນ​ແຕ່ວ່າ​ມັນ​ມາ​ຖືກ​ທີ່​ຖືກ​ເວລາ​ເທົ່າ​ນັ້ນ​ ແຕ່​ເພາະ​ມັນ​ໄດ້​ສະແດງ​ໃຫ້​ເຫັນ​ແລ້ວ​ວ່າ ການ​ອອກ​ແບບ​ແລະ​ແນວ​ຄິດ​ຂອງ​ມັນ​ຊ່ວຍ​ນັກ​ພັດທະນາ​ທັງ​ຫຼາຍ​ໃຫ້​ສາມາດ​ໃຊ້ JavaScript ທາງ​ຝັ່ງ server ໄດ້​ງ່າຍ​ຂຶ້ນ​ຫລາຍ​ອີກດ້ວຍ.
  + - 1. **ExpressJS**

Express.js ເປັນ Web Application Framework ຊື່ດັ່ງທີ່ໄດ້ຮັບຄວາມນິຍົມຫຼາຍ ສຳລັບເຮັດວຽກເທິງ platform ຂອງ Node.js ຊຶ່ງເປັນ Server ຕົວໜຶ່ງໂດຍທັງ Express.js ແລະ Node.js ຕ່າງກໍໃຊ້ພາສາ JavaScript ໃນການພັດທະນາ ຖ້າເປັນ Web Application Framework ໃນສະໄໝກ່ອນ ຄົົນທີ່ພັດທະນາຈະຕ້ອງມີຄວາມຮູ້ຫຼາຍກວ່າ 1 ພາສາ, ພາສາທີ່ເຮັດວຽກທາງຝັ່ງ Server ຢ່າງ PHP ຫລື ASP ແລະ ພາສາທີ່ເຮັດວຽກທາງຝັ່ງ Client ຢ່າງ JavaScript ເພື່ອລຸດຄວາມຫຍຸ້ງຍາກທັງໝົດເຖິງເວລາໃນການຕ້ອງຮຽນຮູ້ຫຼາຍໆ ພາສາເຮັດໃຫ້ເກີດ Node.js ກັບ Express.js ພຽງແຕ່ມີຄວາມຮູ້ JavaScript ກໍສາມາດຂຽນໄດ້ທັງ Server ແລະ Client ນອກຈາກນີ້ຖ້າໃຜເຄີຍຂຽນ JavaScript ຈະຮູ້ວ່າມັນມີການຕອບສະໜອງທີ່ວ່ອງໄວ ແນ່ນອນວ່າ Express.js ກໍຢົກເອົາມາເປັນຂໍ້ເດັ່ນໃນເລື່ອງຄວາມໄວ ໃນເລື່ອງການການຮຽນຮູ້ການຂຽນ Express.js ຈະໃຊ້ຮູບແບບທີ່ງ່າຍໃນການຮຽນຮູ້ຫລາຍທີ່ສຸດສຳລັບການ ພັດທະນາ Express.js ໃນເວັບໄຊຈະເວົ້າເຖິງການໃຊ້ Routing (ການກຳນົດເສັ້ນທາງຂອງລະບົບ) ແລະ Middleware (ການຮັບສົ່ງຂໍ້ມູນຂອງລະບົບ) ສາມາດຂຽນໄດ້ໃນຮູບແບບ MVC ສ່ວນການເຊື່ອມຕໍ່ກັບຖານຂໍ້ມູນສາມາດໃຊ້ MongoDB ຫລືຈະໃຊ້ MySql ກໍໄດ້ສຳລັບນາມສະກຸນຂອງໄຟລຄື .js ຂະນະນີ້ໄດ້ພັດທະນາມາເຖິງເວີຊັ່ນທີ່ 4.x ແລ້ວ.

### ຄວາມຮູ້ກ່ຽວກັບ Microsoft office 2016

Microsoft Word ແມ່ນໂປຣແກມປະມວນຜົນເຊິ່ງອອກແບບເພື່ອຊ່ວຍສ້າງໃຫ້ເອກະສານທີ່ມີຄຸນນະພາບໃນລະດັບມືອາຊີບ ເຄື່ອງມືການຈັດຮູບແບບເອກະສານທີ່ດີທີສຸດຂອງ Word ຈະສາມາດເຮັດໃຫ້ຈັດລະບຽບ ແລະ ຂຽນເອກະສານຂອງທ່ານໄດ້ຢ່າງມີປະສິດທິພາບຫຼາຍຂຶ້ນ Word ຫັຍງມີເຄື່ອງມືການແກ້ໄຂ ແລະ ກວດສອບຄໍາຜິດທີ່ມີປະສິດທິພາບ ສາມາດເຮັດວຽກຮ່ວມກັບຜູ້ອື່ນໄດ້ງ່າຍ.

1. **ປະໂຫຍດຂອງໂປຣແກມ Microsoft word office 2016**

* ມີລະບົບປະຕິບັດຕ່າງໆທີ່ຊ່ວຍໃນການເຮັດວຽກໃຫ້ສະດວກສະບາຍຂຶ້ນເຊັ່ນ: ການກວດຄຳສະກົດ,ການກວດສອບໄວຍະກອນ,ການໃສ່ຂໍ້ຄວາມອັດຕະໂນມັດເປັນຕົ້ນ.
* ສາມາດໃຊ້ສ້າງຈົດໝາຍໄດ້ຢ່າງວ່ອງໄວໂດຍສາມາດກຳນົດໃຫ້ຜູ້ວິເສດ(Wizard) ໃນ Word ສ້າງແບບຟອມຂອງຈົດໝາຍໄດ້ຫຼາຍຮູບແບບຕາມທີ່ຕ້ອງການ.
* ສາມາດໃຊ້ word ສ້າງຕາຕະລາງທີ່ສະຫຼັບຊັບຊ້ອນໄດ້.
* ປັບປຸງເອກສານໄດ້ງ່າຍ ແລະ ວ່ອງໄວ ສາມາດຕົບແຕ່ງເອກະສານ ຫຼື່ເພື່ອຄວາມສະດວກຈະໃຫ້ word ປັບແຕ່ງໃຫ້ກໍ່ໄດ້ ໂດຍສາມາດເປັນຜູ້ກໍານົດຮູບແບບຂອງເອກະສານເອງ.
* ສາມາດແຊກຮູບພາບ ກຣາບຟິກ ຫຼື່ ຜັງອົງກອນລົງໃນເອກະສານໄດ້.
* ຄວາມສາມາດໃນການເຊື່ອມຕໍ່ເອກະສານອື່ນໆໃນຊຸດໂປຣແກມ Microsoft office ສາມາດໂອນຍ້າຍຂໍ້ມູນຕ່າງໆລະຫ່ວາງໂປຣແກມໄດ້.
* ສ້າງເອກະສານໃຫ້ໃຊ້ວຽກໃນອິນເຕີເນັດໄດ້ຢ່າງງ່າຍດາຍ.

1. **ຄວາມສາມາດຂອງ Microsoft word.**

* ສາມາພິມ ແລະ ແກ້ໄຂເອກະສານໄດ້.
* ສາມາດລຶບ,ຍ້າຍ ແລະ ຮ່າງຂໍ້ຄວາມ.
* ສາມາດພິມຕົວອັກສອນປະເພດຕ່າງໆ.
* ສາມາດຂະຍາຍໂຕອັກສອນ.
* ສາມາດຈັດໂຕອັກສອນໃຫ້ເປັນໂຕໜາ,ໂຕງ່ຽງ ແລະ ຂີດກ້ອງໄດ້.
* ສາມາດໃສ່ເຄື່ອງໝາຍ ແລະ ໂຕເລກລໍາດັບຕາມຫົວຂໍ້.
* ສາມາດກວດການສະກົດ ແລະ ແກ້ໄຂໃຫ້ຖືກຕ້ອງ.
* ສາມາດຄົ້ນຫາ ແລະ ປ່ຽນແປງຂໍ້ຄວາມທີ່ພິມຜິດ.
* ສາມາດຈັດຂໍ້ຄວາມເອກະສານໃຫ້ພິມໄປທາງຊ້າຍ,ທາງຂວາ ແລະ ລະຫ່ວາງກາງ.
* ສາມາດໃສ່ຮູບພາບໃນເອກະສານ.
* ສາມາດຕົບແຕ່ງໂຕອັກສອນ,ພິມຕາຕະລາງ.

### ຄວາມຮູ້ກ່ຽວກັບ Microsoft Visio 2016

ໂປຣແກມ Microsoft Visio ເປັນເຄື່ອງມືເສີມການເຮັດວຽກຂອງ Microsoft office ໃນການສ້າງແຜນວາດ (Diagram) ປະເພດຕ່າງໆ ເຊິ່ງເຮັດໄດ້ງ່າຍດາຍ ສະດວກ ແລະ ວ່ອງໄວ ເປັນທີ່ນິຍົມໃຊ້ ມີປະໂຫຍດຫຼາຍໃນການຈັດເອກະສານ ລວມໄປເຖິງແຜນວາດການອອກແບບ ຂັ້ນຕອນການເຮັດວິເຄາະ ແລະ ອອກແບບລະບົບຕ່າງໆ

ສໍາລັບໂປຣແກມ Microsoft Visio ມີໃຫ້ເລື່ອກຢູ່ 2 ປະເພດຄື:

1. Microsoft Visio standard ເໝາະກັບວຽກດ້ານທຸລະກິດເຊັ່ນ ຜູ້ບໍລິຫານໂຄງການ ນັກການຕະຫຼາດ ພະນັກງານຝ່າຍຊັບພະຍາກອນມະນຸດ ແລະ ທິມງານມີໜ້າທີ່ເບິ່ງແຍງການດຳເນີນງານເພື່ອຊ່ວຍການເບິ່ງແຜນວາດ ແລະ ຂ່າວສານ
2. Microsoft Visio Professional ເໝາະສຳລັບມືອາຊີບທາງດ້ານເຕັກນິກ ພະນັກງານໄອທີ ນັກພັດທະນາ ແລະ ວິສະວະກອນທີ່ຊ່ວຍໃນການອອກແບບຂໍ້ມູນລະບົບເພື່ອຊ່ວຍໃນການເຮັດຕົ້ນແບບ Microsoft Visio Professional ເປັນໂປຣແກມທີ່ຖືກສ້າງຂຶ້ນມາເພື່ອຊ່ວຍໃນການສ້າງ Flow Chart ຫຼື່ Diagram ຂອງວຽກງານໃນສາຂາຕ່າງໆໃຫ້ເຮັດວຽກໄດ້ງ່າຍຂຶ້ນ ລັກສະນະຢ່າງໜຶ່ງໃນການສ້າງ Flow Chart ໃນ Microsoft Visio Professional ກໍ່ຄືມີຮູບແບບ Diagram ພື້ນຖານຕ່າງໆຈັດກຽມໄວ້ໃຫ້ ເຊິ່ງງ່າຍໃນການອອກແບບ ແລະ ໃຊ້ວຽກ

ຈຸດດີຂອງ Microsoft Office Visio ຄື ຊອກຫາຂໍ້ຜິດພາດໄດ້ຢ່າງສະດວກ ແລະ ເຂົ້າໃຈງ່າຍໃນການສະແດງທິດທາງການໄຫຼຂໍ້ມູນ

# **ບົດທີ 3**

# **ວິທີດຳເນີນການຄົ້ນຄວ້າ**

## ວິທີການສຶກສາ ແລະ ຄົ້ນຄວ້າ

ເປັນ​ຂະ​ບວນ​ການ​ທີ່​ສະ​ແດງ​ເຖິງ​ການ​ດຳ​ເນີນ​ຂັ້ນ​ຕອນ​ການ​ເຮັດ​ວຽກ​ຂອງ​ລະ​ບົບ​ຕັ້ງ​ແຕ່​ຕົ້ນ​ຈົນ​ຈົບ, ບົດ​ໂຄງ​ການ​ຈົບ​ຊັ້ນ​ນີ້​ພ​ວກ​ຂ້າ​ພະ​ເຈົ້າໄດ້​ນຳ​ໃຊ້​ທິດ​ສະ​ດີ​ການ​ວິ​ເຄາະ ແລະ ອອກ​ແບບ​ລະ​ບົບ​ແບບ​ໂຄງ​ທີ່​ປະ​ກອບ​ມີ 5 ໄລ​ຍະ​ຄື:

* ໄລຍະທີ່ 1 ການວາງແຜນໂຄງການ (Project Planning phase)
* ໄລຍະທີ່ 2 ການວິເຄາະ (Analysis phase)
* ໄລຍະທີ່ 3 ການອອກແບບ (Design phase)
* ໄລຍະທີ່ 4 ການນຳໄປໄຊ້ (Implementation phase)
* ໄລຍະທີ່ 5 ການບໍາລຸງຮັກສາ (Maintenance phase)

### ໄລຍະທີ່ 1 ການວາງແຜນໂຄງການ (Project planning phase)

ການວາງແຜນໂຄງການຈັດເປັນຂະບວນການພື້ນຖານໃນຄວາມເຂົ້າໃຈຢ່າງເລິກເຊິ່ງວ່າເປັນຫຍັງຕ້ອງສ້າງລະບົບໃໝ່ ທິມງານຕ້ອງພິຈາລະນາວ່າ ຈະຕ້ອງດຳເນີນງານຕໍ່ໄປແນວໃດກ່ຽວກັບຂະບວນການສ້າງລະບົບໃໝ່ ກ່ອນອື່ນໝົດຕ້ອງມີຈຸດກຳເນີດຂອງລະບົບງານເຊິ່ງໂດຍປົກກະຕິແລ້ວຈຸດກຳເນີດຂອງລະບົບງານມັກເກີດຂຶ້ນຈາກຜູ້ໃຊ້ລະບົບເປັນຜູ້ທືີຕິດແທດກັບລະບົບໂດຍກົງເຮັດໃຫ້ມີຄວາມໃກ້ສິດກັບລະບົບງານທີ່ດຳເນີນຢູ່ຫຼາຍທີ່ສຸດເມື່ອຜູ້ໃຊ້ລະບົບມີຄວາມຕ້ອງການປັບປຸງລະບົບ ງານດັ່ງນັ້ນຈຶ່ງຖືເປັນຈຸດເລີ່ມຕົ້ນໃນບົດບາດຂອງນັກວິເຄາະລະບົບວ່າຈະຕ້ອງສຶກສາເຖິ່ງຂອບເຂດ ຂອງບັນຫາທີ່ຜູ້ໃຊ້ລະບົບກຳລັງປະສົບບັນຫາຢູ່ ແລະ ຈະດຳເນີນການແກ້ໄຂແນວໃດ ສຶກສາເຖິງຄວາມເປັນໄປໄດ້ວ່າລະບົບໃໝ່ທີ່ຈະພັດທະນາຂຶ້ນມານັ້ນມີຄວາມເປັນໄປໄດ້ ແລະ ຄຸ້ມຄ່າກັບການຈະລົງທຶນຫຼື່ບໍ່.

ແນວໃດກໍ່ຕາມໄລຍະຂອງການວາງແຜນໂຄງການປົກກະຕິມັກຈະມີໄລຍະເວລາທີ່ສັ້ນ ແຕ່ກໍ່ຖືວ່າເປັນຂັ້ນຕອນທີ່ສຳຄັນທີ່ຈະໃຫ້ເກີດຜົນສຳເລັດໄດ້ດັ່ງນັ້ນໃນໄລຍະຂອງການວາງແຜນໂຄງການຈຶ່ງໄດ້ອາໄສນັກວິເຄາະລະບົບທີ່ມີຄວາມຮູ້ ແລະ ປະສົບການສູງເນື່ອງຈາກວ່າຫາກນັກວິເຄາະລະບົບບໍ່ເຂົ້າໃຈເຖິ່ງບັນຫາອັນແທ້ຈິງທີ່ເກີດຂຶ້ນ ກໍ່ຈະບໍ່ສາມາດພັດທະນາລະບົບຂຶ້ນມາເພື່ອແກ້ໄຂບັນຫາໃຫ້ຖືກຈຸດໄດ້ ແລະ ມັກຈະມີໂຄງການພັດທະນາລະບົບຫຼາຍໂຄງການທີ່ຫຼັງຈາກໄດ້ດຳເນີນການພັດທະນາ ແລະ ນໍາມາໃຊ້ງານແລ້ວປະກົດວ່າບໍ່ສາມາດຕອບສະໜອງຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ງານເຊິ່ງຖືວ່າເປັນເລື່ອງທີ່ກໍ່ໃຫ້ເກີດຄວາມສູນເສຍທັງທາງດ້ານການລົງທຶນ ແລະ ໄລຍະເວລາ.

ສະຫຼຸບລວມແລ້ວໄລຍະຂອງການວາງແຜນໂຄງການປະກອບມີກິດຈະກໍາຕ່າງໆຕໍ່ໄປນີ້:

- ກໍານົດບັນຫາ Problem Definition

- ສຶກສາຄວາມເປັນໄປໄດ້ຂອງໂຄງການ Feasibility study

- ສ້າງຕາຕະລາງກໍານົດເວລາໂຄງການ Project scheduling

- ຈັດຕັ້ງທິມງານໂຄງການ Staff the project

### ໄລຍະທີ່ 2 ການວິເຄາະ (Analysis Phase)

ໄລຍະການວິເຄາະຈະຕ້ອງມີຄຳຕອບກ່ຽວກັບຄຳຖາມວ່າໃຜເປັນຜູ້ທີ່ໃຊ້ລະບົບ ແລະ ມີຫຍັງແດ່ທີ່ຈະຕ້ອງເຮັດໃນໄລຍະນີ້ນັກວິເຄາະລະບົບຈະຕ້ອງດຳເນີນການໃນຂັ້ນຕອນຂອງການວິເຄາະລະບົບງານປັດຈຸບັນ (Current system) ເພື່ອນໍາມາພັດທະນາແນວຄວາມຄິດສໍາລັບລະບົບໃໝ່ (New System).

ຈຸດປະສົງຫຼັກຂອງການວິເຄາະຄືຈະຕ້ອງສຶກສາ ແລະ ສ້າງຄວາມເຂົ້າໃຈໃນຄວາມຕ້ອງການຕ່າງໆທີ່ໄດ້ລວບລວມມາດັ່ງນັ້ນການລວບລວມຄວາມຕ້ອງການ (Requirements Gathering) ຈຶ່ງຈັດເປັນງານສ່ວນພື້ນຖານຂອງການວິເຄາະລະບົບໂດຍຂໍ້ມູນຄວາມຕ້ອງການເຫຼົ່ານີ້ນັກວິເຄາະລະບົບຈະນຳມາວິເຄາະເພື່ອທີ່ຈະປະເມີນວ່າຄວນມີຫຍັງແນ່ທີ່ລະບົບໃໝ່ຕ້ອງດຳເນີນການ ແລະ ດ້ວຍເຫດນີ້ເອງການກຳນົດລາຍລະອຽດກ່ຽວກັບຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ (User Requirements) ຈະທະວີຄວາມສຳຄັນຫຼາຍຂຶ້ນເປັນລຳດັບສຳລັບລະບົບທີ່ມີຄວາມສັບຊ້ອນສູງ ແລະ ນັກວິເຄາະຕ້ອງເອົາໃນໃສ່ກັບການລວບລວມຄວາມຕ້ອງການຈາກຜູ້ໃຊ້ບໍ່ຄວນກຳນົດຄວາມຕ້ອງການຂຶ້ນເອງໂດຍໃຊ້ຄວາມຄິດສ່ວນຕົວຂອງຕົນເອງເປັນຫຼັກຫຼື່ປະເມີນຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ລະບົບບໍ່ເຂົ້າກັບຈຸດປະສົງ ແລະ ຫາກມີການພັດທະນາລະບົບຕໍ່ໄປຈົນແລ້ວລະບົບງານທີ່ໄດ້ກໍ່ຈະບໍ່ກົງກັບຄາມຕ້ອງການຂອງຜູ້ໃຊ້ລະບົບຢ່າງແທ້ຈິງເຮັດໃຫ້ຕ້ອງມີຄວາມປັບປຸງຫຼືປ່ຽນແປງພາຍຫຼັງ.

ນັກວິເຄາະລະບົບສາມາດລວບລວມຄວາມຕ້ອງການຕ່າງໆໄດ້ຈາກການສັງເກດ, ການເຮັດວຽກຂອງຜູ້ໃຊ້, ການໃຊ້ເຕັກນິກ, ການສຳພາດ ຫຼື່ ການໃຊ້ແບບສອບຖາມການອ່ານເອກະສານກ່ຽວກັບການປະຕິບັດງານຂອງລະບົບງານປັດຈຸບັນລະບຽບກົດເກນຂອງບໍລີສັດ ແລະ ການມອບໝາຍຕໍາ ແໜ່ງໜ້າທີ່ຮັບຜິດຊອບເຊິ່ງໃນຊ່ວງຂອງການເກັບກຳຂໍ້ມູນຄວາມຕ້ອງການກໍ່ຈະພົບກັບຜູ້ໃຊ້ໃນລະດັບຕ່າງໆທີ່ເຮັດໃຫ້ຮູ້ເຖິ່ງບັນຫາແລະແນວທາງການແກ້ໄຂບັນຫາທີ່ແນະນຳໂດຍຜູ້ໃຊ້ດັ່ງນັ້ນການເກັບກຳຄວາມຕ້ອງການຈຶ່ງເປັນກິດຈະກຳທີ່ສຳຄັນເພື່ອຄົ້ນຫາຄວາມຈິງແລະຕ້ອງສ້າງຄວາມຕ້ອງການເຊິ່ງກັນແລະ ກັນເພື່ອສະຫຼຸບອອກມາເປັນຂໍ້ກຳນົດທີ່ມີຄວາມຊັດເຈນໂດຍຂໍ້ກຳນົດເຫຼົ່ານີ້ເມື່ອຜູ້ໃຊ້ທີ່ກ່ຽວຂ້ອງໄດ້ອ່ານແລ້ວຈະຕ້ອງສຶກສາຄວາມໝາຍໄດ້ກົງກັນ.

ຫຼັງຈາກໄດ້ນໍາຄວາມຕ້ອງການຕ່າງໆມາສະຫຼຸບເປັນຂໍ້ກໍານົດທີ່ຊັດເຈນແລ້ວຂັ້ນຕອນຕໍ່ໄປກໍ່ຄືນໍາຂໍ້ກໍານົດເຫຼົ່ານັ້ນໄປພັດທະນາອອກມາເປັນຄວາມຕ້ອງການຂອງລະບົບໃໝ່ໂດຍເຕັກນິກທີ່ໃຊ້ກໍ່ຄືການພັດທະນາແບບຈໍາລອງຂະບວນການ (Process Model) ເຊິ່ງເປັນແຜນພາບທີ່ໃຊ້ອະທິບາຍເຖິງຂະບວນການທີ່ຕ້ອງດຳເນີນໃນລະບົບວ່າມີຫຍັງແດ່ແລະຕໍ່ໄປກໍ່ດໍຳເນີນການພັດທະນາແບບຈຳລອງຂໍ້ມູນ (Data Model) ເພື່ອອະທິບາຍເຖິງຂໍ້ມູນທີ່ຈັດເກັບໄວ້ສໍາລັບສະໜັບສະໜູນການເຮັກວຽກຕ່າງໆ.

ສະຫຼຸບໄລຍະຂອງການວິເຄາະລະບົບປະກອບມີກິດຈະກໍາຕ່າງໆດັ່ງນີ້:

- ວິເຄາະລະບົບງານປັດຈຸບັນ.

- ເກັບກໍາຄວາມຕ້ອງການໃນດ້ານຕ່າງໆແລະນໍາມາວິເຄາະເພື່ອສະຫຼຸບເປັນຂໍ້ກໍານົດຊັດເຈນ

- ນຳຂໍ້ກໍານົດມາພັດທະນາອອກມາເປັນຄວາມຕ້ອງການຂອງລະບົບໃໝ່.

- ສ້າງແບບຈຳລອງຂະບວນການຂອງລະບົບໃໝ່ໂດຍການແຕ້ມແຜນພາບກະແສຂໍ້ມູ (Data Flow Diagram: DFD).

- ສ້າງແບບຈໍາລອງຂໍ້ມູນໂດຍການແຕ້ມ Entity Relationship Diagram: ERD.

### ໄລຍະທີ່ 3 ການອອກແບບ (Design Phase)

ໄລຍະການອອກແບບເປັນການພິຈາລະນາວ່າລະບົບລະດໍາເນີນການໄປໄດ້ແນວໃດເຊິ່ງກ່ຽວ ຂ້ອງກັບຍຸດທະວິທີການຂອງການອອກແບບທີ່ວ່າດ້ວຍການຕັດສິນໃຈວ່າຈະພັດທະນາລະບົບໃໝ່ດ້ວຍແນວທາງໃດເຊັ່ນພັດທະນາຂຶ້ນເອງ, ຊື້ໂປຣແກຣມສຳເລັດຮູບຫຼືວ່າຈ້າງບໍລິສັດພັດທະນາລະບົບໃຫ້ເປັນຕົ້ນ. ນອກຈາກນີ້ໄລ່ຍະການອອກແບບຈະກ່ຽວຂ້ອງກັບການອອກແບບທາງດ້ານສະຖາປັດຕະຍະ ກຳລະບົບທີ່ກ່ຽວຂ້ອງກັບອຸປະກອນຮາດແວ, ຊອບແວ ແລະ ເຄືອຂ່າຍ.

ການອອກແບບລາຍງານ (Out Design) ການອອກແບບໜ້າຈໍເພື່ອປະຕິສໍາພັນກັບຜູ້ໃຊ້ (User Interface), ການອອກແບບຜັງງານລະບົບ (System Flowchart), ເຊິ່ງລວມເຖິ່ງລາຍລະ ອຽດຂອງໂປຣແກມ (Specific Program), ຖານຂໍ້ມູນ (Database) ແລະ ຂໍ້ມູນທີ່ກ່ຽວຂ້ອງແນວໃດກໍ່ຕາມເຖິ່ງວ່າກິດຈະກຳບາງສ່ວນຂອງໄລຍະອອກແບບນີ້ສ່ວນໃຫ່ຍຈະຖືກດຳເນີນການໄປບາງສ່ວນແລ້ວໃນໄລ ຍະຂອງການວິເຄາະແຕ່ໄລຍະການອອກແບບນີ້ຈະເນັ້ນເຖິ່ງການດຳເນີນການແກ້ໄຂບັນ ຫາແນວໃດຫຼາຍກ່ວາໂດຍການນຳຜົນຂອງແບບຈຳລອງທາງ Logical Model. ທີ່ໄດ້ຈາກການວິ ເຄາະມາພັດທະ ນາມາເປັນແບບຈໍາລອງທາງ Physical Model.

- ການວິເຄາະຈະເນັ້ນແກ້ໄຂບັນຫາຫຍັງແດ່.

- ການອອກແບບຈະເນັ້ນການແກ້ໄຂບັນຫາແນວໃດ.

- ສະຫຼຸບໄລຍະການອອກແບບ.

- ພິຈາລະນາແນວທາງໃນການພັດທະນາລະບົບ.

- ອອກແບບສະຖາປັດຕະຍະກໍາລະບົບ (Architecture Design).

- ອອກແບບຖານຂໍ້ມູນ (Database Design).

- ອອກແບບການສະແດງຜົນ (Output Design).

- ອອກແບບການປ້ອນຂໍ້ມູນ (Input Design).

- ອອກແບບສ່ວນຕິດຕໍໍກັບຜູ້ໃຊ້ (User Interface).

- ສ້າງຕົ້ນແບບ (Prototype).

- ອອກແບບໂປຣແກມ (Structure Chart).

### ໄລຍະທີ່ 4 ການນຳໄປໃຊ້ (Implementation Phase)

ໃນໄລຍະການນນໍາໄປໃຊ້ຈະເຮັດໃຫ້ລະບົບເກີດຜົນຂຶ້ນມາໂດຍການສ້າງລະບົບທົດສອບລະ ບົບແລະການຕິດຕັ້ງລະບົບໂດຍຈຸດປະສົງຫຼັກຂອງກິດຈະກຳໃນໄລຍະນີ້ບໍ່ແມ່ນພຽງຄວາມໜ້າເຊື່ອຖືຂອງລະບົບຫຼື່ລະບົບສາມາດເຮັດວຽກໄດ້ດີເທົ່ານັ້ນແຕ່ຕ້ອງໝັ້ນໃຈວ່າຜູ້ໃຊ້ລະບົບຕ້ອງໄດ້ຮັບການເຝິກ ອົບຮົມເພື່ອໃຊ້ງານລະບົບແລະຄວາມຄາດຫວັງໃນອົງກອນທີ່ຕ້ອງການຜົນຕອບແທນໃນດ້ານດີກັບການໃຊ້ລະບົບໃໝ່ລຳດັບກິດຈະກຳຕ່າງໆທຸກກິດຈະກຳຕ້ອງເຂົ້າມາດຳເນີນການຮ່ວມກັນໃນໄລຍະນີ້ເພື່ອໃຫ້ລະບົບການປະຕິບັດງານໄດ້ຮັບຄວາມປະສົບຜົນສຳເລັດໄດ້ໂດຍດີ.

ສະຫຼຸບໄລຍະການນໍາໄປໃຊ້ຈະປະກອບມີກິດຈະກໍາຕ່າງໆດັ່ງຕໍ່ໄປນີ້:

- ສ້າງລະບົບຂຶ້ນມາດ້ວຍການຂຽນໂປຣແກມ.

- ກວດສອບຄວາມຖືກຕ້ອງທາງດ້ານ Verification ແລະ Validation ແລະດໍາເນີນການທົດສອບລະບົບ.

- ແປງຂໍ້ມູນ (Convert Data).

- ຕິດຕັ້ງລະບົບ (System Installation) ແລະສ້າງເອກກະສານຄູ່ມື.

- ຝຶກອົບຮົມຜູ້ໃຊ້ ແລະ ປະເມີນຜົນລະບົບໃໝ່.

ສໍາລັບການສ້າງລະບົບຫຼື່ການຂຽນໂປຣແກມນັ້ນສາມາດໃຊ້ວິທີການຂຽນໂປຣແກມດ້ວຍພາສາຄອມພິວເຕີເຊັ່ນ: ການໃຊ້ພາສາ Visual Basic, C#, PHP, Java... ນອກຈາກນີ້ຫຍັງມີເຕັກນິກອື່ນໆເຊັ່ນ: ເຄື່ອງມືໃນການພັດທະນາ Application ເຊິ່ງເປັນຊອບແວທີ່ເປັນແຫຼ່ງລວມຂອງເຄື່ອງມືຕ່າງໆທີ່ໃຊ້ເພື່ອພັດທະນາ Application ເຮັດໃຫ້ຜູ້ຂຽນໂປຣແກມບໍ່ເຮັດວຽກໜັກຄືເມື່ອກ່ອນມີແຕ່ຮຽນຮູ້ແລະປະຍຸກໃຊ້ເຄື່ອງມືເຫຼົ່ານັ້ນກໍ່ສາມາດພັດທະນາລະບົບງ່າຍຂຶ້ນ.

### ໄລຍະທີ່ 5 ການບໍາລຸງຮັກສາ (Maintenance Phase)

ໂດຍປົກກະຕິແລ້ວໄລຍະການບໍາລຸງຮັກສາຈະບໍ່ນໍາເຂົ້າໄປລວມໃນສ່ວນຂອງ SDLC ຈົນກະ ທັ້ງລະບົບມີການຕິດຕັ້ງເພື່ອໃຊ້ງານແລ້ວເທົ່ານັ້ນໄລຍະນີ້ຈະໃຊ້ເວລາຍາວນານທີ່ສຸດເມື່ອທຽບກັບໄລ ຍະອື່ນໆທີ່ຜ່ານມາເນື່ອງຈາກລະບົບຈະຕ້ອງໄດ້ຮັບການບຳລຸງຮັກສາຕະຫຼອດໄລຍະເວລາທີ່ມີການໃຊ້ລະບົບສິ່ງທີ່ຄາດຫວັງຂອງໜ່ວຍງານກໍ່ຄືຕ້ອງການໃຫ້ລະບົບໃຊ້ງານຍາວນານຫຼາຍປີລະບົບສາມາດຮອງຮັບເຕັກໂນໂລຊີໃໝ່ໆໃນອະນາຄົດໄດ້ດັ່ງນັ້ນໃນຊ່ວງໄລຍະເວລາດັ່ງກາວຈຶ່ງສາມາດເພີ່ມເຕີ່ມຄວາມສາມາດຂອງລະບົບໃຫ້ມີປະສິດຕິພາບສູງຂຶ້ນພ້ອມທັງການແກ້ໄຂປັບປຸງໂປຣແກມໃນກໍລະນີທີ່ເຫັນຂໍ້ຜິດພາດ.

- ການບໍາລຸງຮັກສາລະບົບ (System Maintenannce).

- ການເພີ່ມເຕີມຄວາມສາມາດໃໝ່ໆເຂົ້າໃນລະບົບ (Enhance System).

- ສະໜັບສະໜູນງານຂອງຜູ້ໃຊ້ (Support the User).

ຈາກໄລຍະຕ່າງໆຕາມຂັ້ນຕອນການພັດທະນາລະບົບຕາມແບບແຜນຂອງ SDLC ຈະເຫັນວ່າມີການໃຊ້ຄໍາວ່າໄລຍະແລະກິດຈະກໍາເຊິ່ງສາມາດອະທິບາຍລາຍລະອຽດເພື່ອໃຫ້ເກີດຄວາມເຂົ້າໃຈກົງກັນດັ່ງນີ້:

- ໄລຍະ (Phase) ຄືກຸ່ມກິດຈະກໍາທີ່ກ່ຽວຂ້ອງກັນ.

- ກິດຈະກໍາ (Activity) ຄືກຸ່ມຂອງງານທີ່ກຽວຂ້ອງກັນ.

- ໜ້າວຽກ (Task) ຄືງານທີ່ດໍາເນີນການເຊິ່ງຖືເປັນງານທີ່ນ້ອຍທີ່ສຸດ.

## ເຄື່ອງມືທີ່ໃຊ້ພັດທະນາ

|  |  |
| --- | --- |
| 1. Hardware | 1. Software |
| * ຄອມພິວເຕີ HP Intel(R) Pentium(R) N3710 CPU @1.60 GHz 1.60 GHz, RAM 4GB DDR3 1600MHz | * ລະບົບປະຕິບັດການ Windows 10 Professional 64 Bit * Microsoft Visio 2016 ໃຊ້ແຕ້ມແຜນວາດການໄຫຼຂໍ້ມູນ (DFD, ER, Flowchart) * Studio 3T For MongoDB ແລະ Moon Modeler ໃຊ້ຈັດການຖານຂໍ້ມູນ * Visual Studio Code ໃຊ້ຂຽນໂຄດດ້ວຍພາສາ JavaScript (React, Nodejs) * MS Office 2013 Professional ໃຊ້ເພື່ອສ້າງເອກະສານຕ່າງໆ, ບົດນຳສະເໜີ ແລະ ເຮັດປຶ້ມບົດຈົບຊັ້ນ |

* 1. **ການວິເຄາະຂໍ້ມູນ**
     1. **ແຜນວາດລວມຂອງລະບົບ (System Overview)**





ຜູ້ຈັດການ

ຂໍ້ມູນສິນຄ້າທີ່ສະໜອງ

ຜູ້ສະໜອງ

ລາຍງານ

ຈັດການຂໍ້ມູນພື້ນຖານ

ຂໍ້ມູນສັ່ງສິນຄ້າ

****

ລະບົບ

ຂໍ້ມູນການຂາຍ

ຂໍ້ມູນໃບບິນ

****

ຂໍ້ມູນຂາຍ

ພະນັກງານ

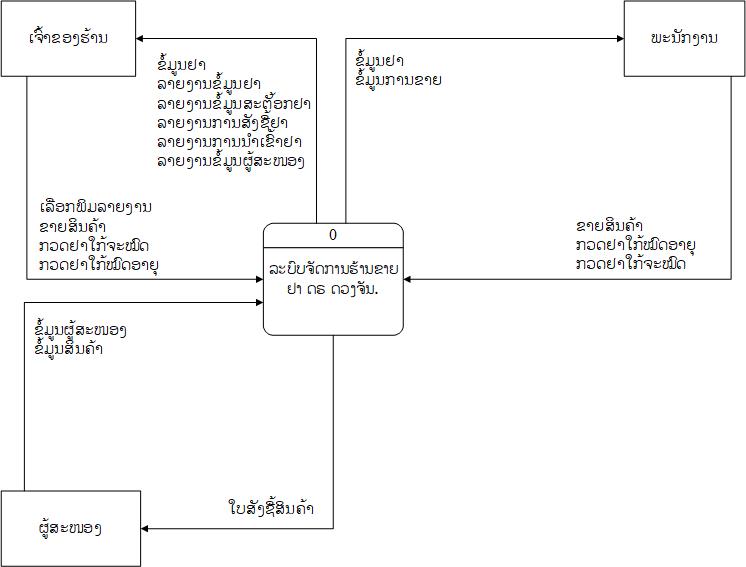
ລູກຄ້າສັ່ງຊື້ສິນຄ້າສິນຄ້າສ

* + 1. **ຕາຕະລາງລາຍລະອຽດຕ່າງໆທີ່ກ່ຽວກັບລະບົບ**

|  |  |  |
| --- | --- | --- |
| External Entity | Process | Data Store |
| ເຈົ້າຂອງຮ້ານ  ພະນັກງານ  ຜູ້ສະໜອງ | 1. **ຈັດການຂໍ້ມູນພື້ນຖານ**    1. ຈັດການຂໍ້ມູນຢາ    2. ຈັດການຂໍ້ມູນປະເພດຢາ    3. ຈັດການຂໍ້ມູນຫົວໜ່ວຍ    4. ຈັດການຂໍ້ມູນຊະນິດຢາ    5. ຈັດການຂໍ້ມູນຜູ້ສະໜອງ    6. ຈັດການຂໍ້ມູນຜູ້ໃຊ້ 2. **ກວດສອບ**    1. ກວດສອບຢາໃກ້ໝົດອາຍຸ    2. ກວດສອບຢາໃກ້ໝົດ 3. **ບໍລິການ**    1. ສັງຊື້ຢາ    2. ນຳເຂົ້າຢາ    3. ຂາຍສິນຄ້າ 4. **ລາຍງານ**    1. ຂໍ້ມູນຢາທັງໝົດ    2. ຂໍ້ມູນຢາໃກ້ໝົດ    3. ຂໍ້ມູນຢາໃກ້ໝົດອາຍຸ    4. ຂໍ້ມູນການນຳເຂົ້າຢາ    5. ຂໍ້ມູນການສັງຊື້    6. ຂໍ້ມູນການຂາຍ | D1 ຂໍ້ມູນຢາ  D2 ຂໍ້ມູນປະເພດຢາ  D3 ຂໍ້ມູນຊະນິດຢາ  D4 ຂໍ້ມູນຫົວໜ່ວຍ  D5 ຂໍ້ມູນຜູ້ສະໜອງ  D6 ຂໍ້ມູນການນຳເຂົ້າ  D7 ຂໍ້ມູນລາຍລະອຽດການນຳເຂົ້າ  D8 ຂໍ້ມູນການຂາຍ  D9 ຂໍ້ມູນລາຍລະອຽດການຂາຍ  D10 ຂໍ້ມູນການສັງຊື້  D11 ຂໍ້ມູນລາຍລະອຽດການສັງຊື້  D12 ຂໍ້ມູນສະຕັ້ອກຢາ  D13 ຂໍ້ມູນຜູ້ໃຊ້ |

ແຜນວາດທີ່ 2: ຕາຕະລາງລາຍລະອຽດຕ່າງໆທີ່ກ່ຽວກັບລະບົບ

* + 1. **ແຜນວາດເນື້ອຫາ (Context Diagram)**

****

ແຜນວາດທີ່ 3: ແຜນວາດເນື້ອຫາ (Context Diagram)

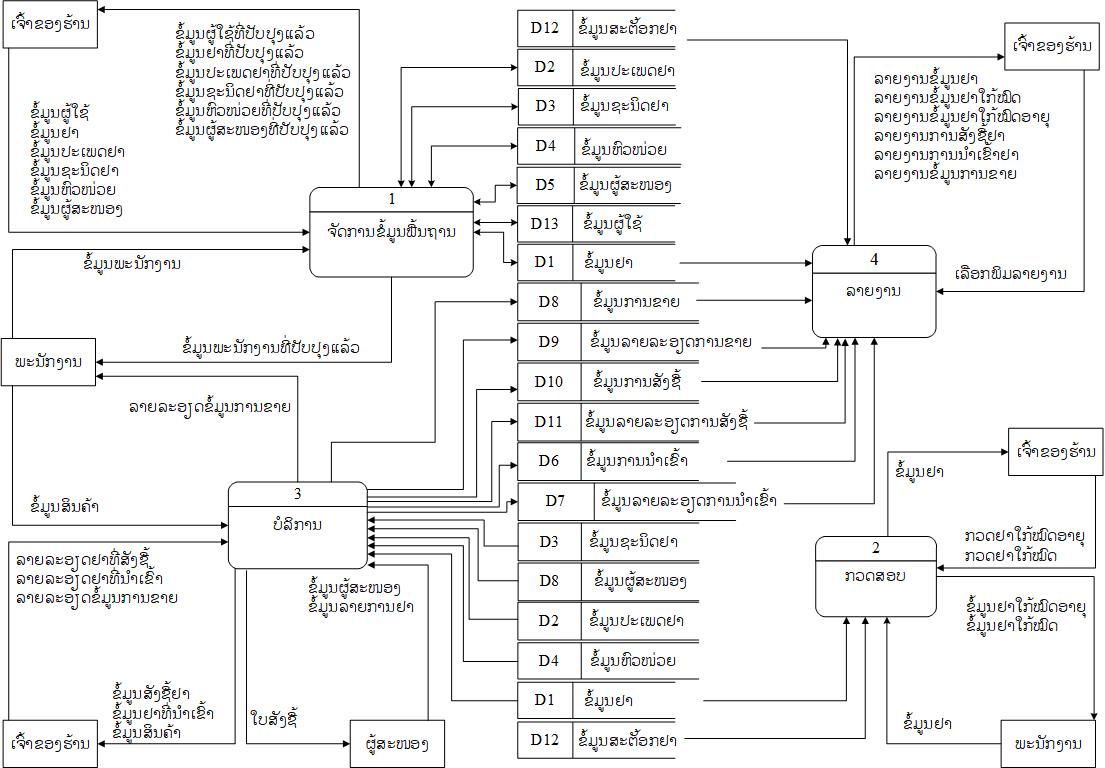
* + 1. **ແຜນວາດລຳດັບຊັ້ນໜ້າທີ (Functional Hierarchy Diagram)**
  1. ສັງຊື້ຢາ
  2. ນຳເຂົ້າຢາ
  3. ຂາຍສິນຄ້າ
  4. ກວດສອບຢາໃກ້ໝົດ
  5. ກວດສອບຢາໃກ້ໝົດອາຍຸ

1. ຈັດການຂໍ້ມູນພື້ນຖານ
   1. ຈັດການຂໍ້ມູນຢາ
   2. ຈັດການຂໍ້ມູນປະເພດຢາ
   3. ຈັດການຂໍ້ມູນຫົວໜ່ວຍ
   4. ຈັດການຂໍ້ມູນຊະນິດຢາ
   5. ຈັດການຂໍ້ມູນຜູ້ສະໜອງ
   6. ຈັດການຂໍ້ມູນຜູ້ໃຊ້
2. ບໍລິການ
3. ລາຍງານ
   1. ຂໍ້ມູນຢາ
   2. ຂໍ້ມູນການສັງຊື້ຢາ
   3. ຂໍ້ມູນສະຕັ້ອກຢາ
   4. ຂໍ້ມູນການນຳເຂົ້າຢາ
   5. ຂໍ້ມູນຜູ້ສະໜອງ

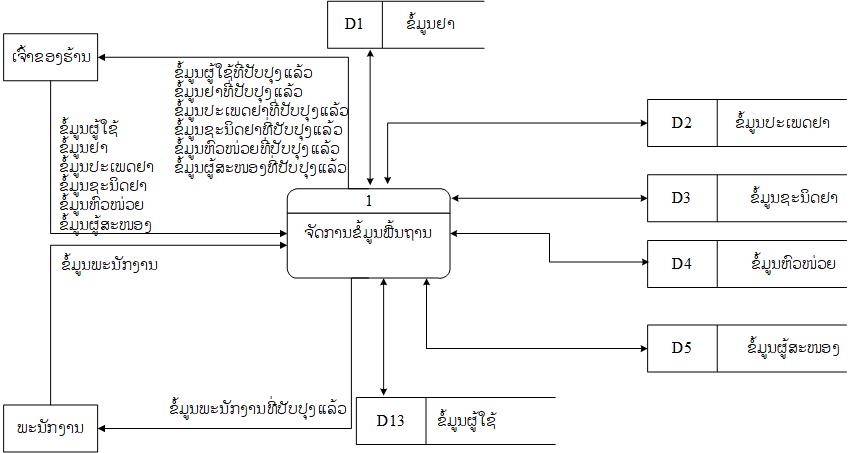
ລະບົບຈັດການຮ້ານຂາຍຢາ ດຣ. ດວງຈັນ

1. ກວດສອບ

ແຜນວາດທີ່ 4: ແຜນວາດລຳດັບຊັ້ນໜ້າທີ່ (Function Hierarchy Diagram)

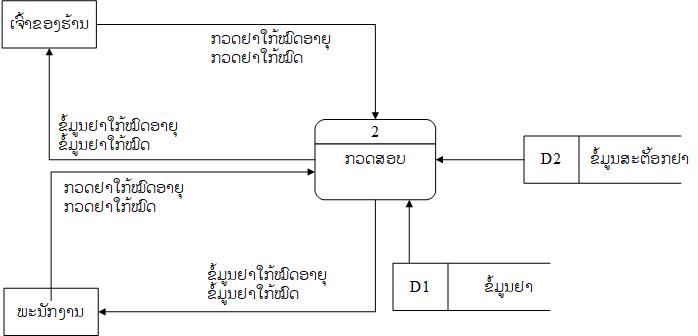
* + 1. **ແຜນວາດການໄຫຼຂໍ້ມູນ (Data Flow Diagram: DFD)**
  1. **ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງແຕ່ລະ Process**

ແຜນວາດທີ່ 5: ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງແຕ່ລະ Process

* 1. **ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 1**

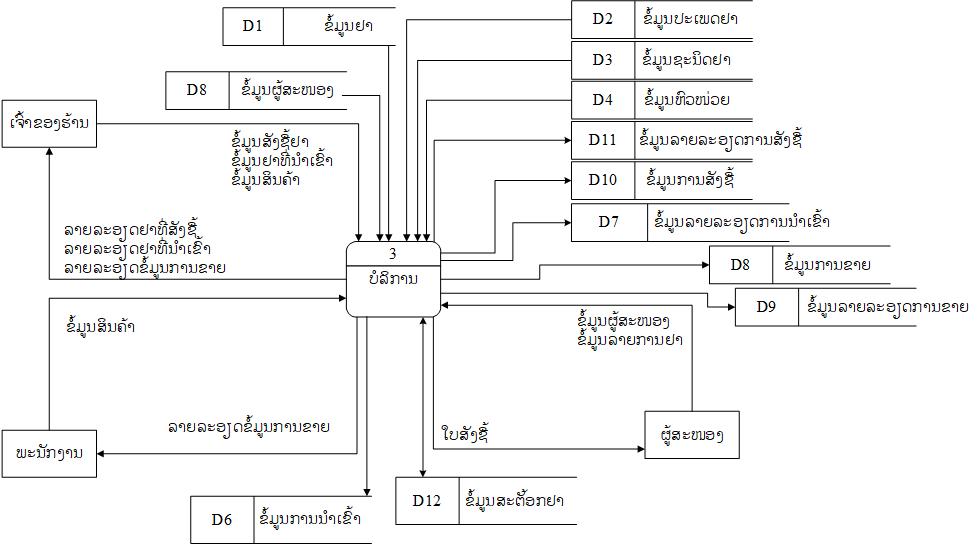
ແຜນວາດທີ່ 6: ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 1

* 1. **ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 2**

****

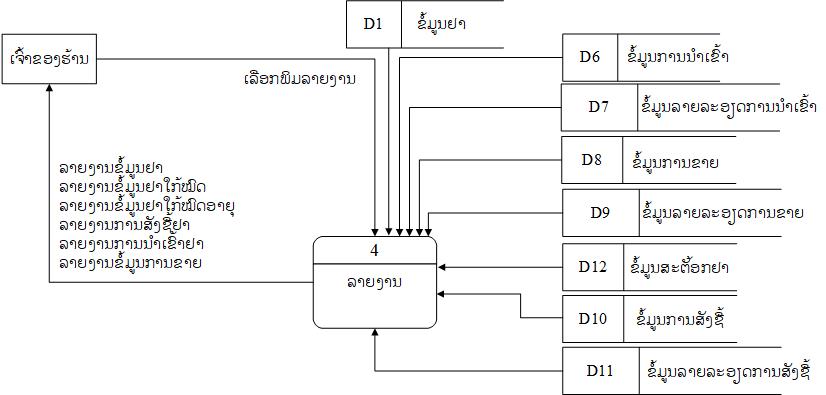
ແຜນວາດທີ່ 7 : ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 2

* 1. **ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 3**

****

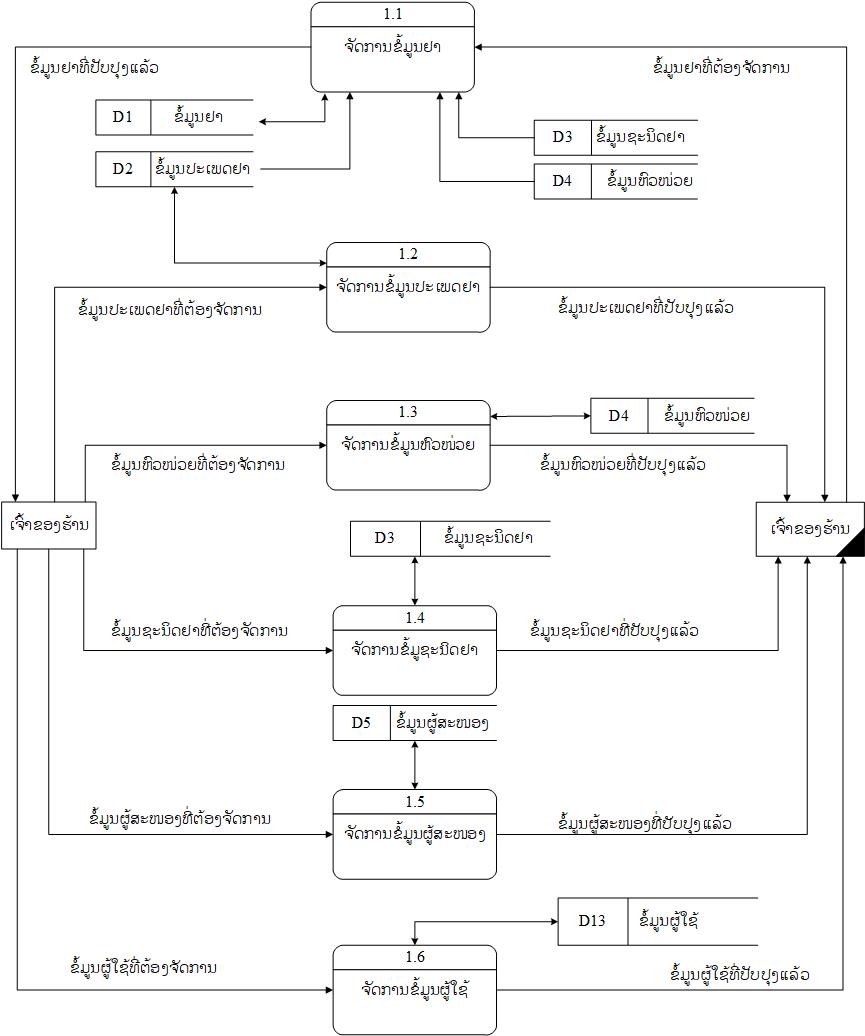
ແຜນວາດທີ່ 8: ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 3

* 1. **ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process 4**

****

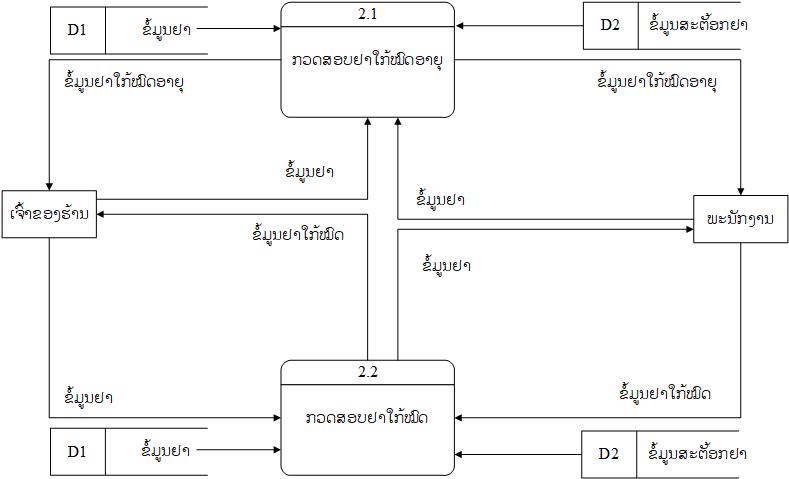
ແຜນວາດທີ່ 9: ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 1 ຂອງ Process4

* 1. **ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 1**

****

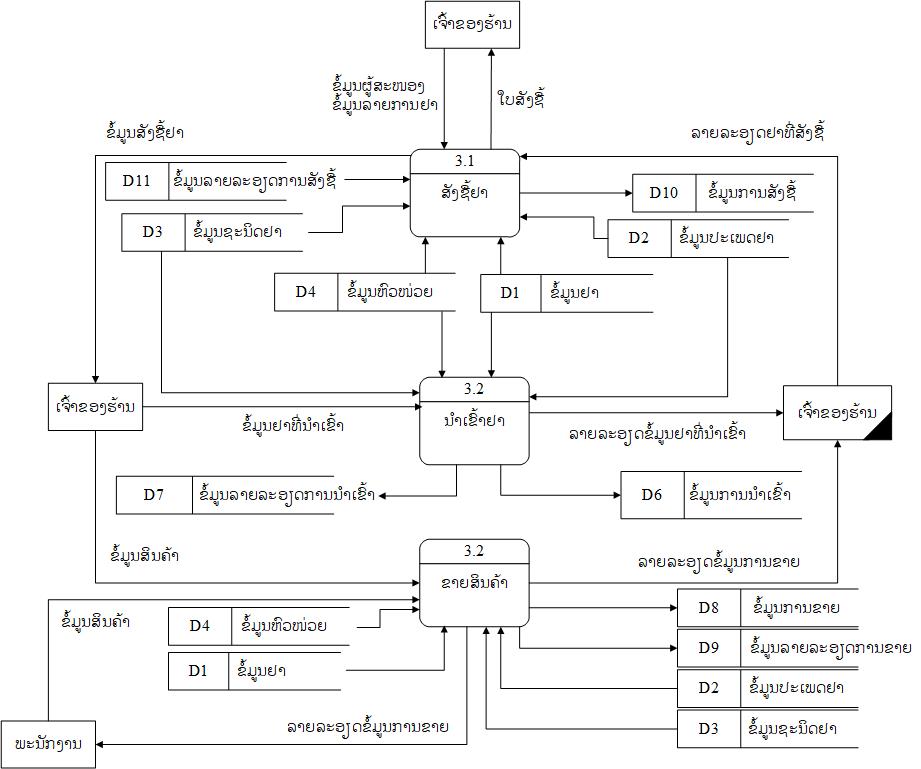
ແຜນວາດທີ່ 10: ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 2 Process 1

* 1. **ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 2**

****

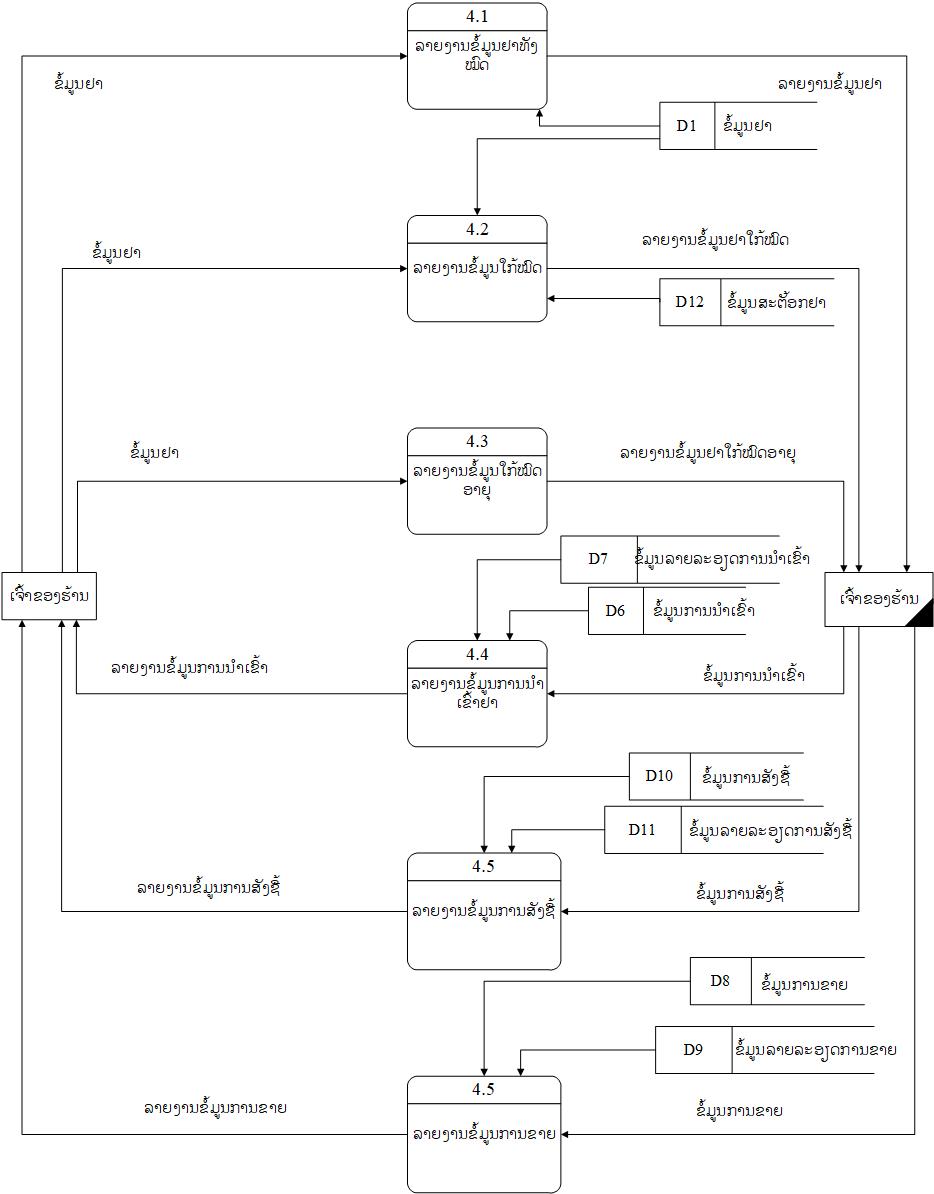
ແຜນວາດທີ່ 11: ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 2

* 1. **ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 3**

****

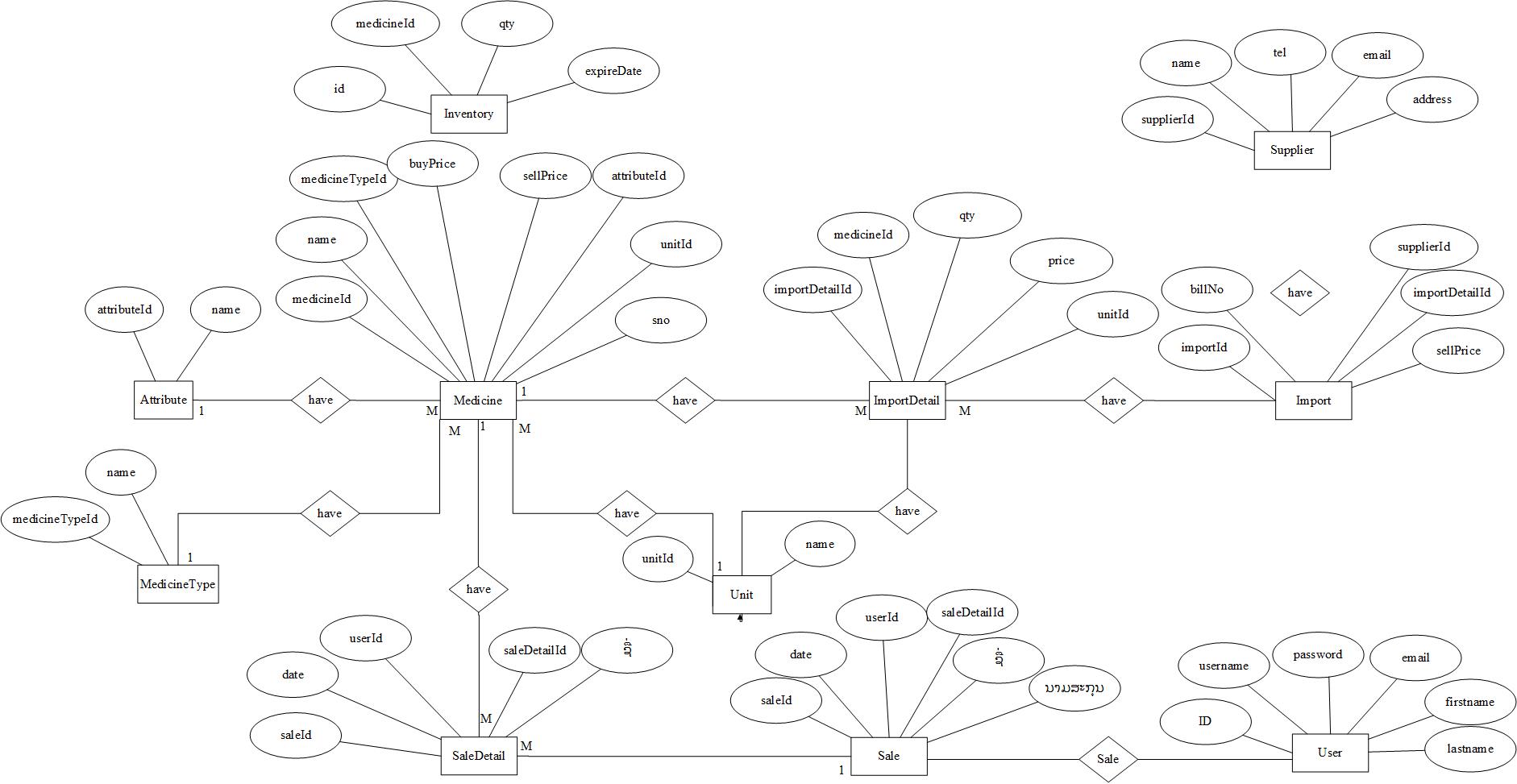
ແຜນວາດທີ່ 12: ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 3

* 1. **ແຜນວາດລວມການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 4**

****

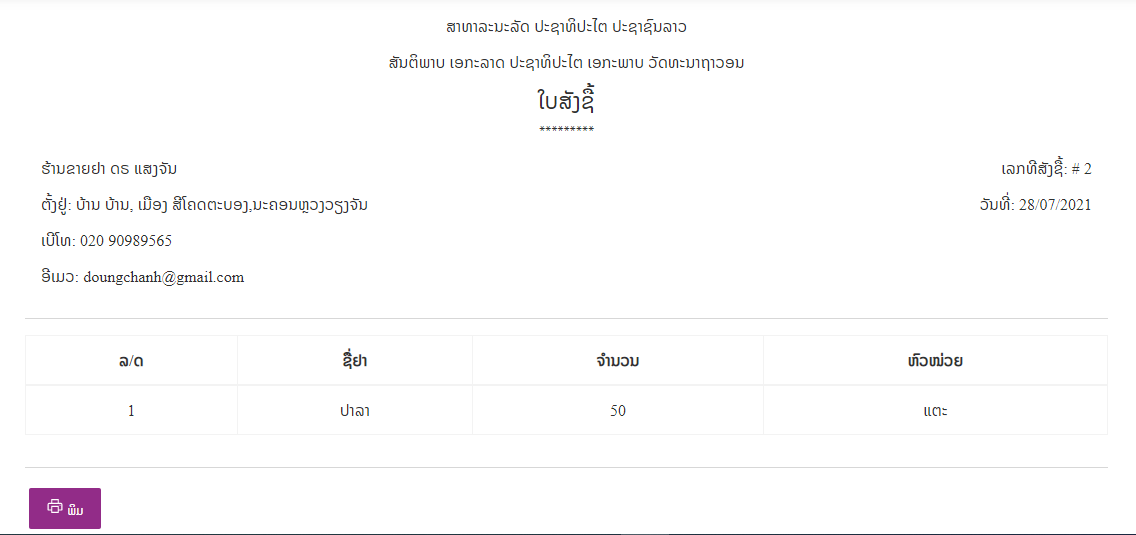
ແຜນວາດທີ່ 13: ແຜນວາດການໄຫຼຂໍ້ມູນລະດັບ 2 ຂອງ Process 4

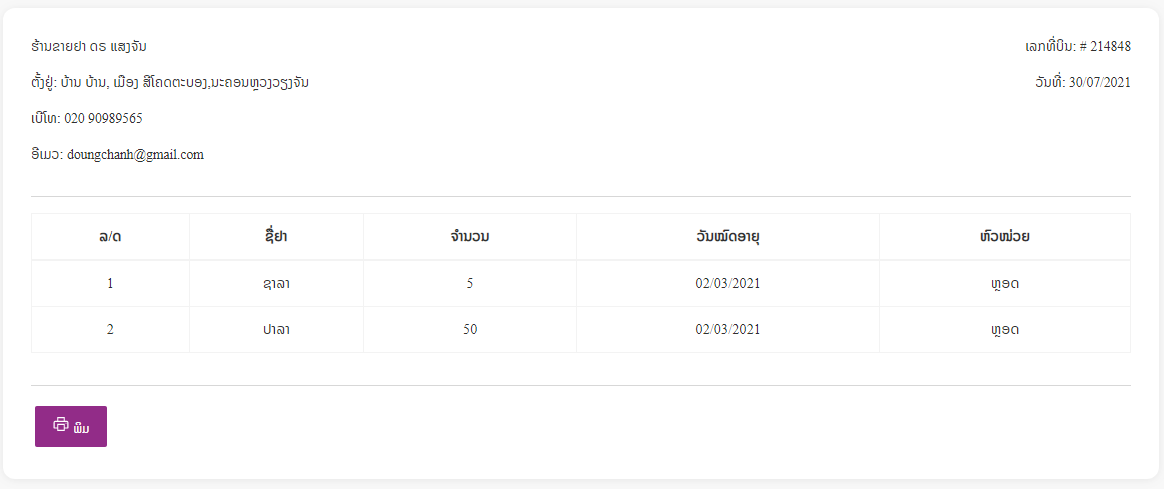
**3.1.2 ແຜນວາດຄວາມສຳພັນຂອງຂໍ້ມູນ (ER Diagram)**

****

ແຜນວາດທີ່ 14: ແຜນວາດຄວາມສຳພັນຂອງຂໍ້ມູນ (ER Diagram)

* 1. **ການອອກແບບລະບົບ**
     1. **ການອອກແບບຮ່າງສະແດງຜົນ (Output Design)**

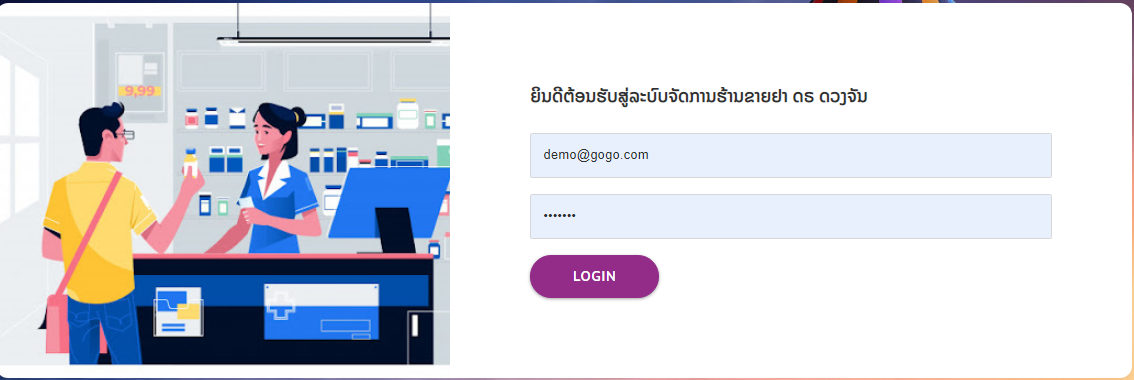
****

****

****

****

* + 1. **ການອອກແບບຮ່າງປ້ອນຂໍ້ມູນ (Input Design)**
* ອອກແບບຟອມເຂົ້າສູລະບົບ

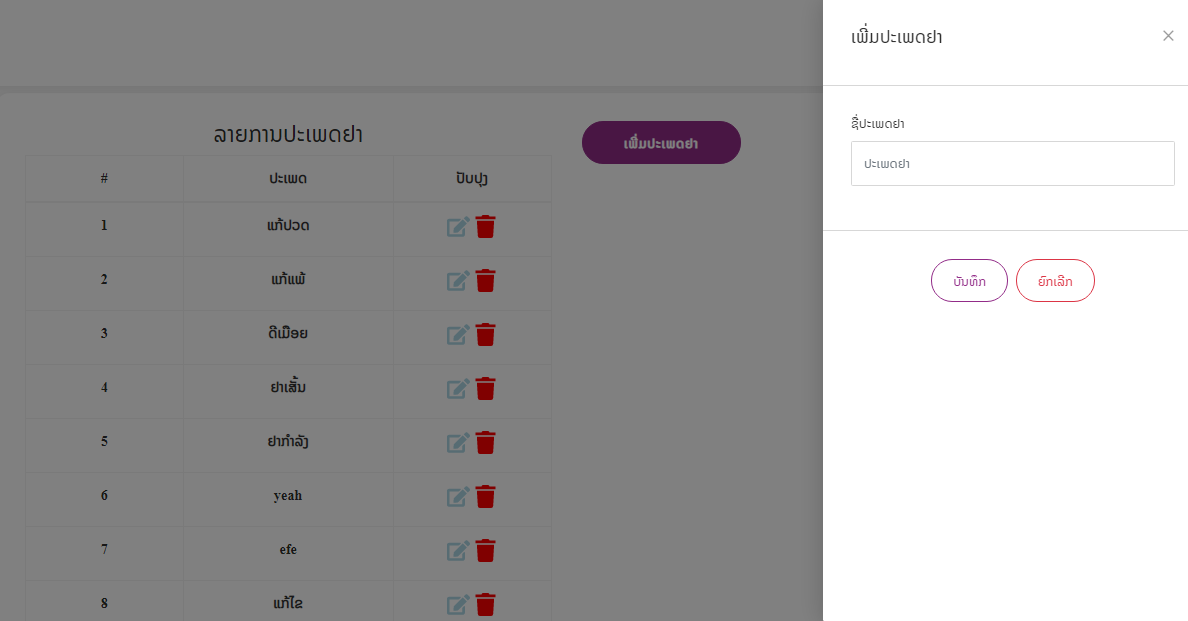


1

2

1 ຊ່ອງປ້ອນຂໍ້ມູນເຂົ້າສູ່ລະບົບ

1. ປຸ່ມເຂົ້າສູ່ລະບົບ

* ອອກແບບຟອມປ້ອນຂໍ້ມູນປະເພດຢາ
* 3****

2

4

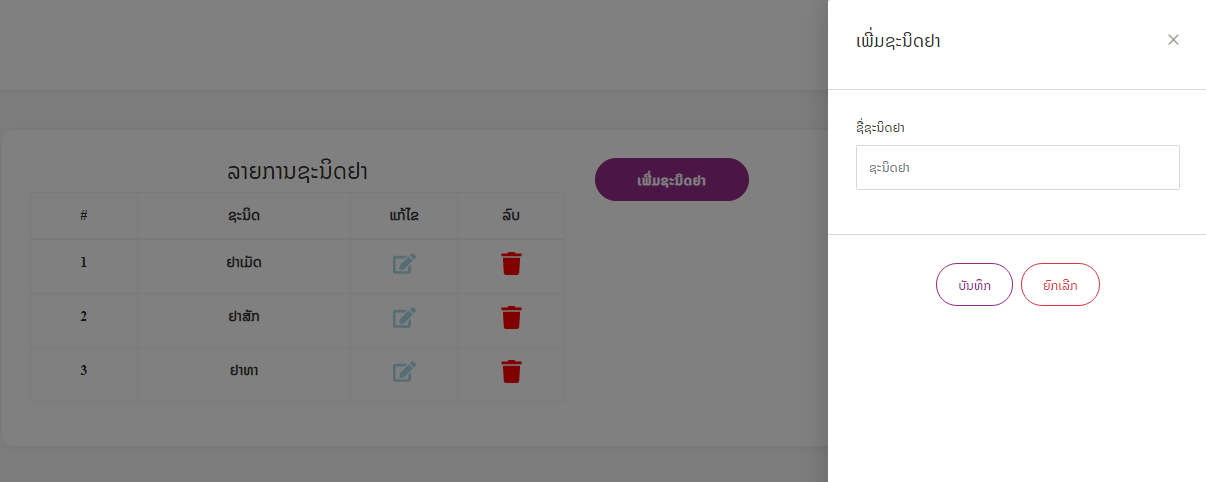
5

1

3

1. ປຸ່ມເຂົ້າໜ້າຟອມເພີ່ມຂໍ້ມູນ
2. ຊ່ອງປ້ອນຂໍ້ມູນປະເພດຢາ
3. ປຸ່ມບັນທຶກ ແລະ ຍົກເລີກ
4. ຕາຕະລາງສະແດງຂໍ້ມູນ
5. ປຸ່ມແກ້ໄຂ ແລະ ລຶບຂໍ້ມູນ

* **ອອກແບບຟອມເພີ່ມຊະນິດຢາ**

****

3

2

1

5

4

5

1 ຊ່ອງປ້ອນຂໍ້ມູນຊະນິດຢາ

2 ປຸ່ມເຂົ້າໜ້າຟອມເພີ່ມຂໍ້ມູນສະນິດຢາ

3 ຕາຕາລາງສະແດງຂໍ້ມູນຊະນິດຢາ

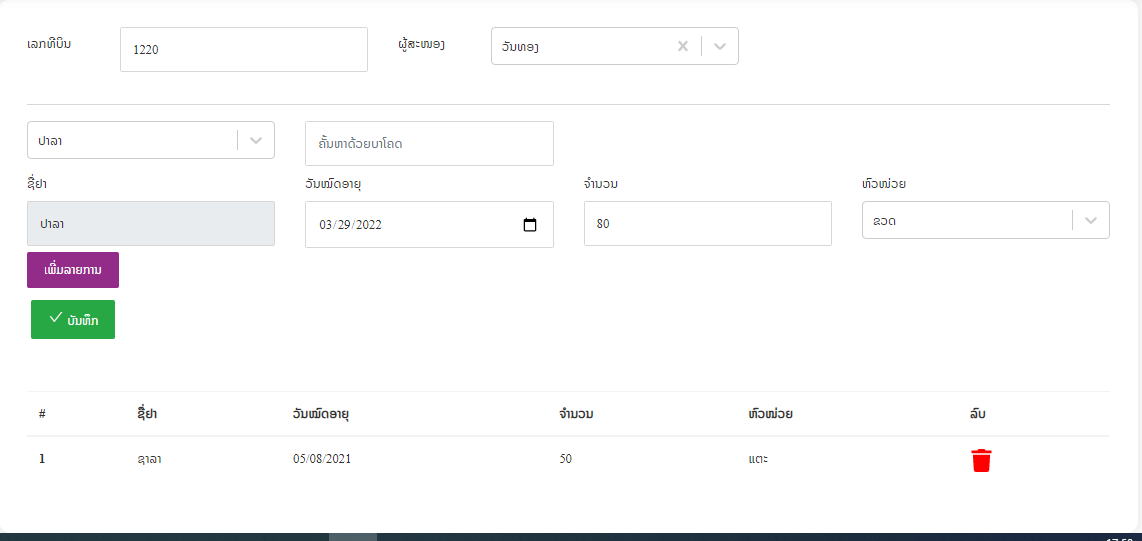
1. ປຸ່ມແກ້ໄຂ

5 ປຸ່ມລຶບ

6 ປຸ່ມບັນທຶກ ແລະ ຍົກເລີກ

* **ອອກແບບຟອມນຳເຂົ້າຢາ**

1



7

4

6

5

3

2

1 ຊ່ອງປ້ອນຂໍ້ມູນຢາ

2 ຊ່ອງເລືອກລາຍການຢາ

3 ຊ່ອງເລືອກລາຍລະອຽດຂໍ້ມູນຢາ

4 ປຸ່ມເພີ່ມຂໍ້ມູນຢາ

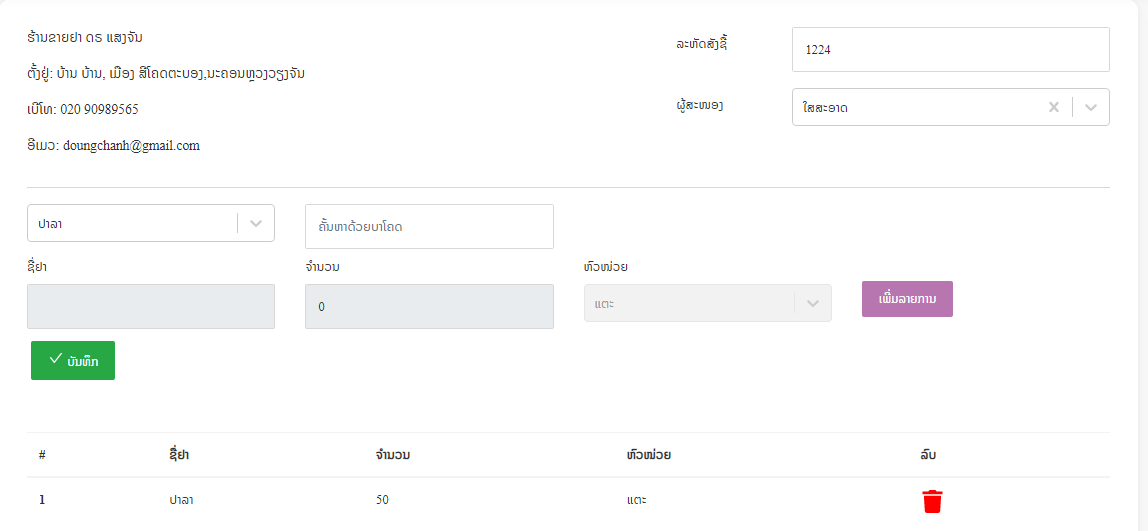
5 ປຸ່ມສະແດງລາຍງານຢານຳເຂົ້າ

6 ບັນທຶກ

* **ອອກແບບຟອມສັງຊື້ຢາ**

2

1

****

3

6

4

7

5

8

1 ຊ່ອງປ້ອນຂໍ້ມູນລະຫັດສັງຊື້ຢາ

2 ຊ່ອງປ້ອນຂໍ້ມູນຜູ້ສະໜອງ

3 ຊ່ອງປ້ອນຂໍ້ມູນເລກທີ່ບິນ

4 ຊ່ອງເລືອກລາຍລະອຽດຂໍ້ມູນຢາ

5 ປຸ່ມສະແດງລາຍງານຢານຳເຂົ້າ

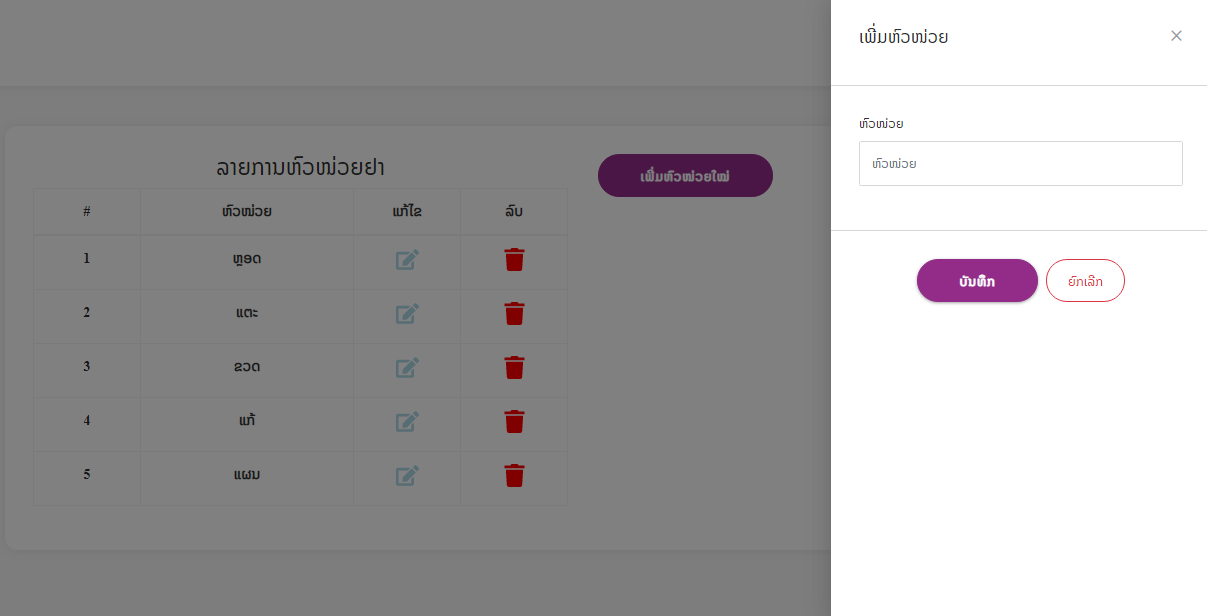
6 ປຸ່ມເພີ່ມລາຍການຢາ

7 ປຸ່ມບັນທຶກ

8 ປຸ່ມລຶບ

* **ອອກແບບຟອມຈັດການຫົວໜ່ວຍ**

2

****

1

3

4

5

1 ຊ່ອງປ້ອນຂໍ້ມູນຫົວໜ່ວຍ

2 ຕາຕາລາງສະແດງລາຍການຫົວໜ່ວຍຢາ

3 ປຸ່ມເຂົ້າໜ້າຟອມເພີ່ມຫົວໜ່ວຍ

4 ປຸ່ມບັນທຶກ

5 ປຸ່ມຍົກເລີກ

* **ອອກແບບຟອມຈັດການຜູ້ສະໜອງ**

****

1

2

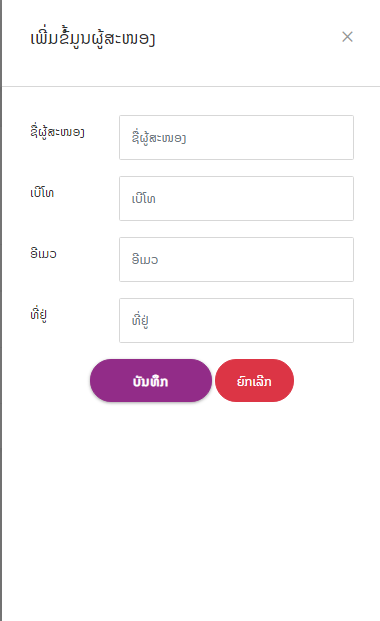
3

**1 ປຸ່ມເພີ່ມຜູ້ສະໜອງ**

**2 ປຸ່ມສະແດງລາຍງານຂໍ້ມູນຜູ້ສະໜອງ**

**3 ປຸ່ມແກ້ໄຂ ແລະ ລຶບ**

* **ຟອມເພີ່ມຜູ້ສະໜອງ**

****

1

3

2

**1 ຊ່ອງປ້ອນຂໍ້ມູນຜູ້ສະໜອງ**

**2 ປຸ່ມບັນທຶກ**

**3 ປຸ່ມຍົກເລີກ**

* + 1. **ການອອກແບບຖານຂໍ້ມູນ**

1. ຕາຕະລາງຂໍ້ມູນຜູ້ໃຊ້ (User)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table: User** | | | | | |
| Field Name | Data Type | Allow Null | Key | Description | Reference |
| id | Int | No | PK | ລະຫັດພະນັກງານ |  |
| firstname | String | No |  | ຊື່ພະນັກງານ |  |
| lastname | String | No |  | ນາມສະກຸນ |  |
| username | String | No |  | ຊື່ຜູ້ໃຊ້້ພະນັກງານ |  |
| password | String | No |  | ລະຫັດຜ່ານ |  |
| email | String | Yes |  | ອີເມວ |  |
| address | String | No |  | ທີ່ຢູ່ |  |

ຕາຕະລາງທີ່ x: ຕາຕະລາງ

1. ຕາຕະລາງຂໍ້ມູນຜູ້ສະໜອງ (Supplier)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table: Supplier** | | | | | |
| Field Name | Data Type | Allow Null | Key | Description | Reference |
| id | Int | No | PK | ລະຫັດຜູ້ສະໜອງ |  |
| name | String | No |  | ຊື່ |  |
| tel | String | No |  | ເບີໂທ |  |
| email | String | No |  | ຊື່ຜູ້ໃຊ້້ອີເມວ |  |
| address | String | No |  | ທີ່ຢູ່ |  |

1. ຕາຕະລາງຂໍ້ມູນສັງຊື້ (Order)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table: Order** | | | | | |
| Field Name | Data Type | Allow Null | Key | Description | Reference |
| id | ObjectId | No | PK | ລະຫັດການສັງ |  |
| orderNo | String | No |  | ຊື່ພະນັກງານ |  |
| supplierId | ObjectId | No | FK | ລະຫັດຜູ້ສະໜອງ | Supplier |
| orderItemId | ObjectId | No | FK | ລະຫັດລາຍລະອຽດ | OrderItem |

1. ຕາຕະລາງຂໍ້ມູນລາຍລະອຽດສັງຊື້ (**OrderItem**)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table: OrderItem** | | | | | |
| Field Name | Data Type | Allow Null | Key | Description | Reference |
| id | objectid | No | PK | ລະຫັດພະນັກງານ |  |
| medicineld | objectid | No | FK | ລະຫັດຢາ | Medicine |
| qty | Number | No |  | ຈຳນວນ |  |
| unitld | objectid | No |  | ຫົວໜ່ວຍ |  |

1. ຕາຕະລາງຂໍ້ມູນນຳເຂົ້າ (Import)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table: Import** | | | | | |
| Field Name | Data Type | Allow Null | Key | Description | Reference |
| id | objectid | No | PK | ລະຫັດນຳເຂົ້າ |  |
| bilNo | String | No |  | ເລກທີບິນ |  |
| supplierld | objectid | No | FK | ລະຫັດຜູ້ສະໜອງ | Supplier |
| importDetail | objectid | Yes | FK | ຊື່ຜູ້ໃຊ້້ພະນັກງານ | Import |
| date | Date | No |  | ວັນທີ່ນຳເຂົ້າ |  |

1. ຕາຕະລາງຂໍ້ມູນລາຍລະອຽດນຳເຂົ້າ (ImportDetail)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table: ImportDetail** | | | | | |
| Field Name | Data Type | Allow Null | Key | Description | Reference |
| id | objectid | No | PK | ລະຫັດ |  |
| medicineld | objectid | No | FK | ລະຫັດຢາ | Medicine |
| qty | Number | No |  | ຈຳນວນ |  |
| unitld | objectid | No | FK | ລະຫັດຫົວໜ່ວຍ | Unit |
| expireDate | Date |  |  | ວັນໝົດອາຍຸ |  |
| price | Number |  |  | ລາຄາ |  |

1. ຕາຕະລາງຂໍ້ມູນຢາ (Medicine)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table: Medicine** | | | | | |
| Field Name | Data Type | Allow Null | Key | Description | Reference |
| id | objectid | No | PK | ລະຫັດຢາ |  |
| name | String | No |  | ຊື່ຢາ |  |
| medicineTypeld | objectid | No | FK | ລະຫັດປະເພດ | MedicineType |
| buyPrice | Number | No |  | ລາຄາຊື້ |  |
| sellPrice | Number | No |  | ລາຄາຂາຍ |  |
| unitld | objectid | No | FK | ລະຫັດຫົວໜ່ວຍ | Unit |
| attributeld | objectid | No | FK | ລະຫັດຊະນິດຢາ | Attribute |
| barcode | String | Yes |  | ບາໂຄດ |  |
| imageUrl | String | Yes |  | ຮູບ |  |
| expireDate | Date | No |  | ວັນໝົດອາຍຸ |  |
| cutStock | Boolean | No |  | ຕັດສະຕ໋ອກ |  |
| inStock | Number | Yes |  | ຈຳນວນສະຕ໋ອກ |  |

1. ຕະລາງຂໍ້ມູນຫົວໜ່ວຍ (Unit)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table: Unit** | | | | | |
| Field Name | Data Type | Allow Null | Key | Description | Reference |
| id | objectid | No | PK | ລະຫັດ |  |
| name | String | No |  | ຊື່ຫົວໜ່ວຍ |  |

1. ຕາຕະລາງຂໍ້ມູນປະເພດຢາ (MedicineType)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table: MedicineType** | | | | | |
| Field Name | Data Type | Allow Null | Key | Description | Reference |
| id | objectid | No | PK | ລະຫັດ |  |
| name | String | No |  | ຊື່ປະເພດ |  |

1. ຕາຕະລາງຂໍ້ມູນຊະນິດຢາ (Attribute)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table: Attribute** | | | | | |
| Field Name | Data Type | Allow Null | Key | Description | Reference |
| id | objectid | No | PK | ລະຫັດ |  |
| name | String | No |  | ຊະນິດຢາ |  |

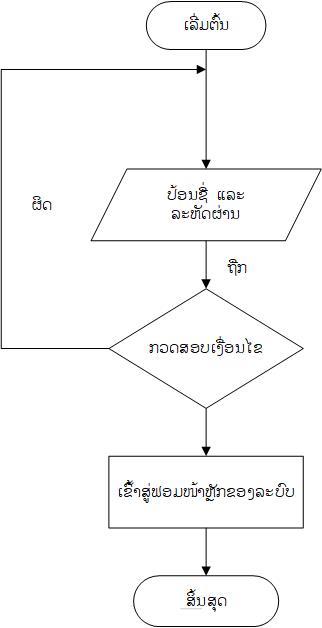
* 1. ຕາຕະລາງຂໍ້ມູນການຂາຍ (Sale)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table: Sale** | | | | | |
| Field Name | Data Type | Allow Null | Key | Description | Reference |
| id | objectid | No | PK | ລະຫັດ |  |
| date | Date | No |  | ວັນທີ |  |
| userld | objectid | No | FK | ລະຫັດຜູ້ໃຊ້ | User |
| saleDetailld[] | objectid | No | FK | ລະຫັດລາຍລະອຽດ | SaleDetail |
| amount | Number | No |  | ເປັນເງິນ |  |

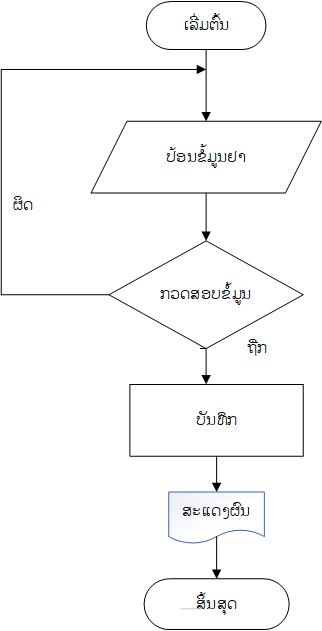
* 1. ຕາຕະລາງຂໍ້ມູນລາຍລະອຽດການຂາຍ (SaleDetail)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table: SaleDetail** | | | | | |
| Field Name | Data Type | Allow Null | Key | Description | Reference |
| id | objectid | No | PK | ລະຫັດພະນັກງານ |  |
| medicineld | objectid | No | FK | ຊື່ພະນັກງານ | Medicine |
| qty | Number | No |  | ຈຳນວນ |  |
| price | Number | No |  | ລາຄາ |  |
| subTotal | Number | No |  | ລວມ |  |

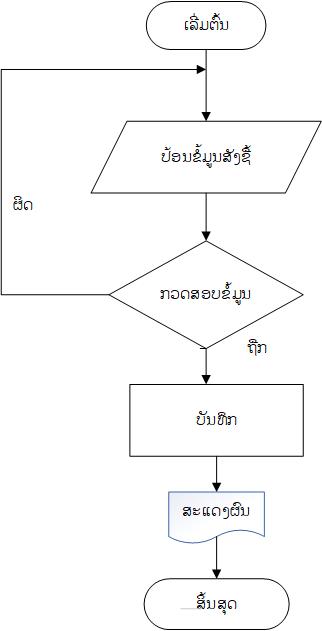
* + 1. **ແຜນວາດຂັ້ນຕອນການເຮັດວຽກ (Flowchart)**
* ຂັ້ນຕອນວິທີການເຂົ້າສູ່ລະບົບ (Algorithm of Form Login)



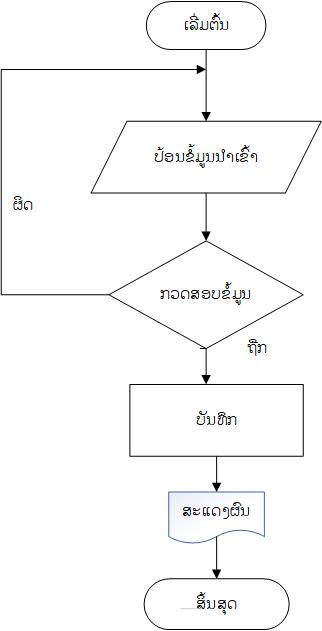
* ຂັ້ນຕອນວິທີຂອງ Function ຈັດການຂໍ້ມູນຫຼັກ (Algorithm of Function Manage main data)



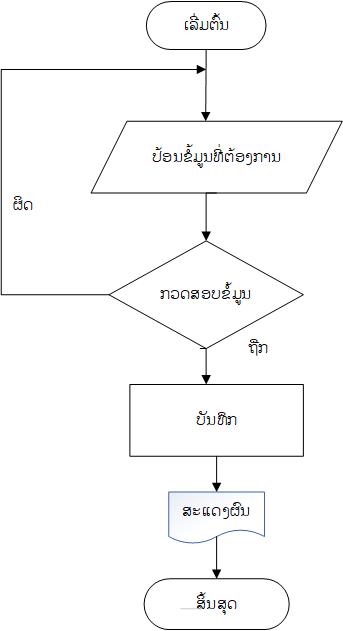
* ຂັ້ນຕອນວິທີຂອງ Function ສັງຊື້ສິນຄ້າ (Algorithm of Function Order)



* ຂັ້ນຕອນວິທີຂອງ Function ນຳເຂົ້າສິນຄ້າ (Algorithm of Function Import)



* ຂັ້ນຕອນວິທີຂອງ Function ຂາຍສິນຄ້າ (Algorithm of Function Sale)

****

# **ບົດທີ 4**

# **ຜົນການສຶກສາ ແລະ ອະທິບາຍຜົນ**

### 4.1 ໜ້າຕ່າງການ (Login Form)

ເມື່ອເປີດໂປຣແປຣມຂຶ້ນມາກໍ່ຈະເຫັນວ່າເຂົ້າສູ່ລະບົບເປັນໜ້າທຳອິດຈາກນັ້ນຈຶ່ງທຳການເຂົ້າສູ່ລະບົບດ້ວຍຊື່ ແລະລະຫັດເຂົ້າໃຊ້ເພື່ອເຂົ້າໄປດ້ານໃນຂອງໂປຣແກຣມດັ່ງຮູບລຸ່ມນີ້:

- ວິທີການເຂົ້າສູ່ລະບົບມີຄື :

1) ໃສ່ຊື່ຜູ້ໃຊ້ລະບົບ.

2) ໃສ່ລະຫັດຜ່ານ.

3) ກົດປຸ່ມເພື່ອເຂົ້າລະບົບ.

4) ຖ້າຊື່ກັບລະຫັດຖືກຈະເປີດນຳໃຊ້ໂປຣແກຣມໄດ້ປົກກະຕິແລ້ວ.

5) ຖ້າຊື່ກັບລະຫັດຜິດຈະມີຂໍ້ຄວາມຂື້ນມາວ່າ ກະລຸນາກວດສອບຂໍ້ມູນຂອງທ່ານໃຫ້ຖືກຕ້ອງ ແລ້ວກົດປຸ່ມ OK

6) ຖ້າບໍ່ເຂົ້າລະບົບແມ່ນກົດປຸ່ມອອກລະບົບ.

ຮູບທີ 4.1 ຟອມຫຼັກ (Login)

### 4.2 ຟ້ອນຫຼັກ (Main Form)

ຟອມຫຼັກແມ່ນຟອມທີ່ເຮົາສາມາດເຂົ້າຫາທຸກໆຟອມໄດ້ເຊັ່ນ ຟອມຈັດການຂໍ້ມູນ, ຟອມປ້ອນຂໍ້ມູນ, ຟອມການບໍລິການ, ຟອມລາຍຮັບ, ຟອມລາຍຈ່າຍ, ຟອມລາຍງານ.

ຮູບທີ 4.2 ຟອມຫຼັກ (Main Form)

# **ບົດທີ 5**

# **ສະຫຼຸບ ແລະ ຂໍ້ສະເໜີ**

## 5.1 ສະຫຼຸບ

ລະບົບຈັດການຮ້ານຂາຍຢາ ດຣ ດວງຈັນ ແມ່ນຖືກພັດທະນາຂຶ້ນເພື່ອຊ່ວຍ ໃຫ້ມີລະບົບທີ່ທັນສະໄໝສະດວກສະບາຍ ແລະ ເຮັດ ໃຫ້ການຈັດການຂໍ້ມູນ, ການບໍລິການພາຍໃນ ຮ້ານມີຄວາມສະດວກ ວ່ອງໄວ, ມີຄວາມເປັນລະບຽບ, ສະດວກໃນການລາຍງານ, ຂໍ້ມູນທີ່ມີຄວາມຖືກຕ້ອງ ແລະ ຊັດເຈນ. ຂອບເຂດຂອງການດໍາເນີນວຽກງານໂປຣແກຣມ ຂອງພວກຂ້າພະເຈົ້າມີຈັດການຂໍ້ມູນ , ສະໝັກສະມາຊິກ , ບໍລິການ ແລະ ລາຍງານ. ເຊິ່ງເປັນການທົດແທນການເຮັດວຽກໃນລະບົບເກົ່າໂດຍລະບົບໃໝ່ເພື່ອຫຼຸດຜ່ອນບັນຫາການເສຍຫາຍຂອງຂໍ້ມູນ. ພວກນ້ອງສາມາດພັດທະນາໂປຣແກຮມ ເພື່ອຊ່ວຍໃຫ້ເຮັດວຽກສະດວກສະບາຍຍິ່ງຂື້ນ ແລະ ໄດ້ອອກແບບໜ້າຟອມການປ້ອນຂໍ້ມູນຕ່າງໆ.

- ຟອມຈັດການຂໍ້ມູນຫຼັກ 3 ຟອມ

- ສ້າງຟອມຈັດການໄດ້ 5 ຟອມ

- ຟອມການຄົ້ນຫາໄດ້ 5 ຟອມ

- ພິມລາຍງານທັງໝົດໄດ້ 9 ລາຍງານ

## 5.2 ຈຸດດີ

- ຮູ້ວິເຄາະຫາບັນຫາ ແລະ ສາເຫດຂອງລະບົບເກົ່າ

- ຈະໄດ້ລະບົບຈອງປີ້ລົດເມສາຍໃຕ້ແບບອອນໄລໃໝ່

- ສາມາດນໍາໃຊ້ເວບໄຊທີ່ສ້າງຂຶ້ນມາເຂົ້າຊ່ວຍໃນການຈອງປີ້ລົດເມແບບອອນລາຍ

- ເວບໄຊທີ່ສ້າງຂຶ້ນມາສາມາດຈອງອອນລາຍໄດ້

- ການເຮັດບົດລາຍງານສະດວກສະບາຍ ແລະ ວ່ອງໄວຂື້ນ

## 5.3 ຈຸດອ່ອນ

ໂປແກມນີ້ຍັງບໍ່ສົມບູນ ແລະ ຄົບຖ້ວນຕາມຄວາມຕ້ອງການຂອງສະຖານີຂົນສົ່ງໂດຍສານສາຍໃຕ້ເທື່ອ ເນື່ອງຈາກວ່າໃນການສຶກສາຍັງບໍ່ທັນມີປະສົບການໃນການຂຽນເວບໄຊມາກ່ອນ,ຍັງບໍ່ຄວບຄຸມເຖິງຄວາມຕ້ອງການຕົວຈິງຂອງຜູ້ໃຊ້ລະບົບ.

## 5.4 ແນວທາງໃນການພັດທະນາ ແລະ ຂະຫຍາຍຕໍ່ຂອງສະຖານີ

ເນື່ອງຈາກວ່າໂປແກມນີ້ເປັນໂປຣແກຣມທີ່ສ້າງຂື້ນມາແລ້ວ ດັ່ງນັ້ນເພື່ອເປັນການກວດສວບຫາຂໍ້ພິດພາດ ແລະ ເພື່ອຫາຈຸດດີຈຸດອ່ອນມາທຳການປັບປຸງ ແກ້ໄຂຈື່ງຄວນນຳເອົາໂປຣແກຣມນີ້ໄປປັບປຸງຈຸດບົກພ່ອງຂອງໂປແກຣມໃນບາງສ່ວນ ເພື່ອເຮັດໃຫ້ໂປຣແກຣມມີປະສິດທິພາບ ແລະ ເຮັດວຽກໄດ້ດີຂື້ນ ເພື່ອຈະນຳເອົາໄປປະກອບສ່ວນເຂົ້າໃນການເຮັດວຽກງານຕົວຈິງ

ດັ່ງນັ້ນໂປຣແກຣມນີ້ເປັນໂປຣແກຣມໜື່ງທີ່ມີຄຸນສົມບັດໃນການໃຊ້ຖານຂໍ້ມູນໄດ້.

**ເອກະສານອ້າງອີງ**



**ປະຫວັດຫຍໍ້ຜູ້ຂຽນ**

ຊື່ ແລະ ນາມສະກຸນ: ນາງ ແສງຈັນ ພົນທະວອນ

ວັນ,ເດືອນ, ປີເກີດ: 08 ເດືອນ ຕຸລາ(10) ປີ 1997

ບ້ານເກິດ: ນາແລ້ງ ເມືອງ: ຫົວເມືອງ ແຂວງ: ຫົວພັນ

ຫຼວງວຽງຈັນ ບ້ານຢູ່ປັດຈຸບັນ: ໂພນສະຫັວນໃຕ້ ເມືອງ:ສີສັດຕະນາກ ແຂວງ: ນະຄອນຫຼວງວຽງຈັນ

ການສຶກສາ: ປີ 2019 ຈົບຊັ້ນສູງ ທີ່ ວິທະຍາໄລສອນພິທັກລາວ

ປີ 2016 ຈົບມັດທະຍົມຕອນປາຍ ມ.ສ ບວມງາມ

ປີ 2013 ຈົບມັດທະຍົມຕອນຕົ້ນ ທີ ມ.ສ ບວມງາມ

ປີ 2007 ຈົບປະຖົມສົມບູນ ທີ ໂຮງຮຽນປະຖົມສົມນາແລ້ງ

ເບີໂທ: 02091830764

ອີເມວ: sengchanh.phonthavone@gmail.com

**ປະຫວັດຫຍໍ້ຜູ້ຂຽນ**

ຊື່ ແລະ ນາມສະກຸນ: ທ້າວ ບຸນໄຊ ໄຊຍະວົງ

ວັນ,ເດືອນ,ປີເກີດ: 11 ເດືອນ 07 ປີ 1994

ບ້ານເກິດ:ບ້ານ ຫ້ວຍຕູ້, ເມືອງ: ບຸນເໜືອ, ແຂວງ: ຜົ້ງສາລີ ບ້ານຢູ່ປັດຈຸບັນ:ບ້ານ ດົງໂດກ, ເມືອງ: ໄຊທານີ, ແຂວງ: ນະຄອນຫຼວງວຽງຈັນ

ການສຶກສາ: ປີ 2017 ຈົບຊັ້ນສູງ ທີ່ ສະຖາບັນພັດທະນາສີມືແຮງງານລາວ - ເກົາຫຼີ

ປີ 2013 ຈົບມັດທະຍົມສົມບູນ ມສ ເມືອງໄຊ ແຂວງ ອຸດົມໄຊ

ປີ 2007 ຈົບປະຖົມສົມບູນ ທີ ໂຮງຮຽນປະຖົມສົມບູນໂຮງຮຽນຫຼັກ69 ເມືງ ບຸນ

ເໜືອ ແຂວງ ຜົ້ງສາລີ

ເບີໂທ: 020 92199562

ອີເມວ:Xaixaiyavong12@gmail.com