



ສາທາລະນະລັດ ປະຊາທິປະໄຕ ປະຊາຊົນລາວ  
ສັນຕິພາບ ເອກະລາດ ປະຊາທິປະໄຕ ເອກະພາບ ວັດທະນາຖາວອນ

-----((0))-----



ມະຫາວິທະຍາໄລແຫ່ງຊາດ  
ຄະນະວິທະຍາສາດທຳມະຊາດ  
ພາກວິຊາວິທະຍາສາດຄອມພິວເຕີ

ບົດສະເໜີໂຄງການຈົບຊັ້ນ ລະດັບປະລິນຍາຕີວິທະຍາສາດ  
ສາຂາ ວິທະຍາສາດຄອມພິວເຕີ

ຊື່ບົດຈົບຊັ້ນ (Title)

ພາສາລາວ: ລະບົບຈັດການຮ້ານຂາຍຢາ ດຣ ດວງຈັນ.

ພາສາອັງກິດ: Douangchanh Pharmacy Management System

ລາຍຊື່ນັກສຶກສາຂຽນບົດ (Project Team)

ລ/ດ	ລະຫັດ	ຊື່ ແລະ ນາມສະກຸນ	ເບີໂທ
1	204N0017.19	ນາງ ແສງຈັນ ພົນທະວອນ	020 91830764
2	204N0057.19	ທ້າວ ບຸນໄຊ ໄຊຍະວົງ	020 55950147
3	204N0003.19	ທ້າວ ຄອນສະຫວັນ ສິງຊົມພູ	020 55781990

ອາຈານທີ່ປຶກສາ (Advisor)

ລ/ດ	ຊື່ ແລະ ນາມສະກຸນ	ຕຳແໜ່ງ
1	ປທ ວິໄລສັກ ແກ້ວສຸດທາ	ນຳພາ
2	ອຈ ປທ ບົວສິດໄຊຍະຈັກ	ຊ່ວຍນຳພາ

## 1. ຄວາມສໍາຄັນຂອງບັນຫາ

ໃນຍຸກແຫ່ງຄວາມກ້າວໜ້າທາງດ້ານເທັກໂນໂລຊີຂໍ້ມູນຂ່າວສານ ທີ່ນັບມື້ມີການພັດທະນາຢ່າງບໍ່ຢຸດຢັ້ງ ທຸກບາດກ້າວໃນການດໍາລົງຊີວິດລ້ວນແລ້ວແຕ່ນໍາໃຊ້ເທັກໂນໂລຊີ, ບໍ່ວ່າຈະເປັນວິສາຫະກິດ, ທຸລະກິດ ແລະ ອົງກອນຕ່າງໆ. ໄດ້ນໍາໃຊ້ເທັກໂນໂລຊີເຂົ້າມາຊ່ວຍ ໃນການເສີມສ້າງປະສິດທິພາບຂອງການເຮັດວຽກເພື່ອເຮັດໃຫ້ການເຮັດວຽກໄວຂຶ້ນ. ການນໍາໃຊ້ໂປຣແກຣມເຂົ້າມາຊ່ວຍໃນການເຮັດວຽກຕ່າງໆ ແມ່ນນັບຫຼາຍຂຶ້ນ ໃນນັ້ນລະບົບຈັດການການຂາຍ ແມ່ນໂປຣແກຣມໜຶ່ງ ທີ່ຊ່ວຍໃຫ້ທຸລະກິດ ດໍາເນີນໄປຢ່າງມີປະສິດທິພາບ ແລະ ວ່ອງໄວ. ຮ້ານຂາຍຢາ ດຣ. ດວງຈັນ ກໍເປັນອີກຮ້ານໜຶ່ງທີ່ຕ້ອງການນໍາໃຊ້ເທັກໂນໂລຊີທີ່ທັນສະໄໝເຂົ້າມາຊ່ວຍແກ້ໄຂບັນຫາຕ່າງໆໃນການເຮັດວຽກ. ດັ່ງນັ້ນ, ທາງຮ້ານ ດຣ. ດວງຈັນ ຈຶ່ງຢາກພັດທະນາລະບົບດັ່ງກ່າວນີ້ມາແກ້ໄຂບັນຫາທີ່ພົບໃນປັດຈຸບັນ. ການຄົ້ນຫາຂໍ້ມູນມີຄວາມຫຍຸ້ງຍາກ, ຂໍ້ມູນມີການຕົກເຮ່ຍເສຍຫາຍເຮັດໃຫ້ເກີດຄວາມຫຍຸ້ງ ຍາກຊັກຊ້າຕາມມາໃນເວລາທີ່ຕ້ອງການ, ບໍ່ສາມາດກວດສອບຈຳນວນຢາໄດ້, ການສະຫຼຸບບັນຊີແມ່ນມີຄວາມຊັກຊ້າ ບໍ່ມີປະສິດທິພາບ ແລະ ມີຂໍ້ຜິດພາດໃນການຂາຍ.

ຮ້ານ ຂາຍຢາ ດຣ. ດວງຈັນ ຕັ້ງຢູ່ບ້ານ ໂສກປ່າຫຼວງ, ເມືອງ ສີສັດຕະນາກ, ນະຄອນຫຼວງວຽງຈັນ. ເຈົ້າຂອງຮ້ານຊື່ວ່າ: ດຣ. ດວງຈັນ ກອງຜາລີ. ເປີດມາໄດ້ 20 ປີ ແລ້ວຂຶ້ນທະບຽນວິສາຫະກິດສ່ວນບຸກຄົນລົງວັນທີ 3/2/2000, ເລກທະບຽນ 1328/ຈທວ. ເປີດບໍລິການ ວັນຈັນ-ວັນເສົາ ຕັ້ງແຕ່ເວລາ 8:00 ຕອນເຊົ້າ ເຖິງ 22:00 ຕອນແລງ ຮ້ານຂາຍຢາ ດຣ. ດວງຈັນ ແມ່ນຮ້ານຂາຍຢາທີ່ປະກອບດ້ວຍຢາຫຼາຍຊະນິດບໍ່ວ່າຈະເປັນ. ຢາແກ້ປວດ, ຢາດີຫວັດ, ວິຕາມິນ, ຢາຕ້ານເຊື້ອ ແລະ ອື່ນໆ....

ຮ້ານຂາຍຢາ ດຣ. ດວງຈັນ ມີຄົນເຮັດວຽກທັງໝົດ 4 ຄົນຄື: ເຈົ້າຂອງຮ້ານ, ລູກເຈົ້າຂອງຮ້ານ ແລະ ພະນັກງານ 2 ຄົນ ຍິງ 2 ຄົນ ພະນັກງານຍິງຄົນທໍາອິດຈະເຮັດໜ້າທີ່ຈັດການຢາ ແລະ ພະນັກງານຍິງອີກຄົນ ໜຶ່ງແມ່ນປະຈຳຢູ່ຮ້ານເພື່ອຂາຍ ແລະ ຮັບລາຍການສົ່ງຊື້ຈາກລູກຄ້າ. ເຈົ້າຂອງຮ້ານເຮັດໜ້າທີ່ເກັບກຳເງິນ ທີ່ໄດ້ຮັບຈາກການຂາຍແຕ່ລະມື້. ກໍລະນີເຈົ້າຂອງຮ້ານບໍ່ຢູ່ຈະແມ່ນໜ້າທີ່ຂອງລູກເພິ່ນ.

ເນື່ອງຈາກວ່າການບັນທຶກຂໍ້ມູນຕ່າງໆ ບໍ່ວ່າຈະແມ່ນຂໍ້ມູນການຂາຍຢາ ແລະ ລາຍລະອຽດການຈັດການຢາ ຍັງໃຊ້ວິທີການຈົດກ່າຍໃສ່ເຈ້ຍ ແລະ ບາງຄັ້ງເຮັດໃຫ້ເກີດມີຄວາມຜິດພາດເຊັ່ນ: ເນື່ອງຈາກ ການຄິດໄລ່ຕ້ອງຄິດໄລ່ດ້ວຍມືເຮັດໃຫ້ຂາດຄວາມຊັດເຈນ ແລະ ຊັກຊ້າ, ການກວດສອບຢາ ແລະ ຂໍ້ມູນຂອງຢາບໍ່ມີຄວາມລະອຽດ ການກວດກາບັນຊີການຂາຍ, ສະຫຼຸບລາຍຮັບປະຈຳວັນທາງຮ້ານໄດ້ໃຊ້ການຈົດກ່າຍໃສ່ປຶ້ມເຮັດໃຫ້ຂໍ້ມູນຕົກເຮ່ຍເສຍຫາຍ ແລະ ຊັກຊ້າ.

ຈາກບັນຫາຕ່າງໆທີ່ກ່າວມາຂ້າງເທິງພວກຂ້າພະເຈົ້າຈຶ່ງໄດ້ມີຄວາມຄິດຢາກສ້າງລະບົບຈັດການການຂາຍຢາຂຶ້ນມາເພື່ອຊ່ວຍຫຼຸດຜ່ອນໃນຄວາມຫຍຸ້ງຍາກໃນການຈັດການ ແລະ ເພີ່ມຄວາມສະດວກສະບາຍໃຫ້ແກ່ຜູ້ໃຊ້ໃນການຈັດການລະບົບ ຫຼຸດຜ່ອນຄວາມຜິດພາດ ໃນການຄິດໄລ່ລາຍໄດ້ຈາກການຂາຍ, ກໍ່ໃຫ້ເກີດຄວາມສະດວກສະບາຍ ແລະ ວ່ອງໄວໃນການຄົ້ນຫາ ແລະ ແກ້ໄຂຂໍ້ມູນການຂາຍຢາ.

## 2. ຈຸດປະສົງຂອງການຄົ້ນຄວ້າ (Objectives)

ເພື່ອຫຼຸດຜ່ອນຄວາມຜິດພາດໃນການເຮັດວຽກຂອງຮ້ານຂາຍຢາ ດຣ ດວງຈັນ ບໍ່ວ່າຈະເປັນການເກັບກຳຂໍ້ມູນການຂາຍ ແລະ ຄິດໄລ່ລາຍໄດ້ຈາກການຂາຍແຕ່ລະມື້, ບັນທຶກລາຍລະອຽດການສັ່ງຊື້ ແລະ ນຳເຂົ້າຂອງຢາ. ຈຸດປະສົງຫຼັກຂອງການຄົ້ນຄວ້າມີດັ່ງນີ້;

- ເພື່ອສຶກສາບັນຫາທີ່ເກີດຂຶ້ນຂອງຮ້ານຂາຍຢາ ດຣ ດວງຈັນ.
- ເພື່ອແກ້ໄຂບັນຫາໃນການຈັດການຂໍ້ມູນການຂາຍ. ຮ້ານຂາຍຢາ ດຣ ດວງຈັນ ໃຫ້ມີຄວາມປອດໄພ, ຫຼຸດຜ່ອນຄວາມຜິດພາດ ແລະ ປ້ອງກັນການຕົກເຮ່ຍເສຍຫາຍຂອງຂໍ້ມູນ.
- ເພື່ອພັດທະນາໂປຣແກຣມຂາຍຢາຂອງຮ້ານ ດຣ ດວງຈັນ

## 3. ຂອບເຂດການຄົ້ນຄວ້າ (Scope)

ລະບົບຈັດການຮ້ານຂາຍຢາຂອງຮ້ານ ດຣ ດວງຈັນຮູບແບບຂອງລະບົບແມ່ນ ເປັນລະບົບແບບ Web-based Applications ເຊິ່ງປະກອບດ້ວຍໜ້າວຽກຫຼັກດັ່ງນີ້:

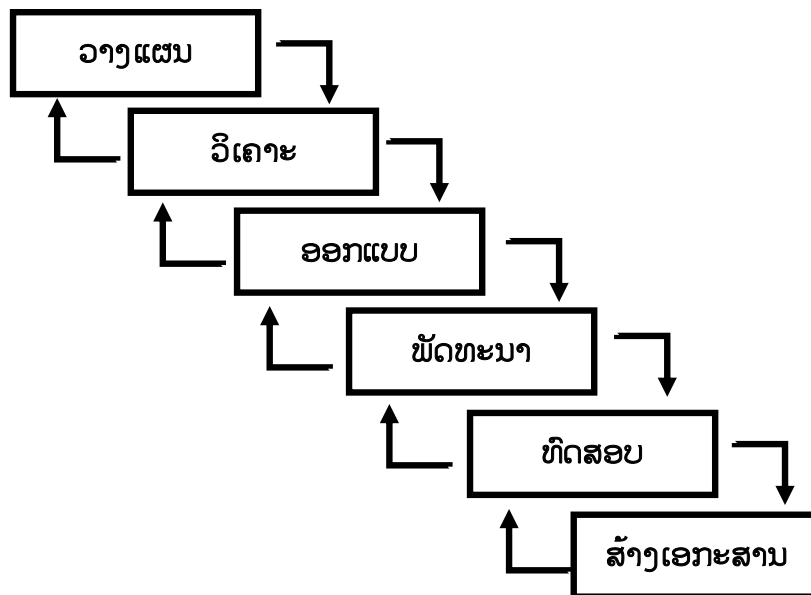
- ຈັດການຂໍ້ມູນພື້ນຖານ : (ຈັດການຂໍ້ມູນຢາ, ຈັດການຂໍ້ມູນປະເພດຢາ, ຈັດການຂໍ້ມູນຫົວໜ່ວຍ, ຈັດການຂໍ້ມູນຜູ້ສະໜອງ, ຈັດການຂໍ້ມູນຜູ້ໃຊ້.)
- ກວດສອບ : (ກວດສອບຢາໃກ້ໝົດອາຍຸ, ກວດສອບຢາໃກ້ຈະໝົດ)
- ບໍລິການ(ສັ່ງຊື້ຢາ, ນຳເຂົ້າຢາ, ຂາຍຢາ)
- ລາຍງານ : (ລາຍງານຂໍ້ມູນຢາທັງໝົດ, ລາຍງານຂໍ້ມູນສະຕ້ອກຢາ, ລາຍງານການນຳເຂົ້າຢາ, ລາຍງານການສັ່ງຊື້, ລາຍງານຂໍ້ມູນຜູ້ສະໜອງ, ລາຍງານລາຍຮັບ-ລາຍຈ່າຍ)

## 4. ປະໂຫຍດຄາດວ່າຈະໄດ້ (Expected Outcome of the Project)

- ຈະໄດ້ໂປຣແກຣມຂາຍຢາຂອງຮ້ານ ດຣ ດວງຈັນ.
- ຈະສະດວກໃນການເກັບກຳຂໍ້ມູນການຂາຍຢາ ແລະ ການສັ່ງຊື້ຢາເຂົ້າໃນຮ້ານ.
- ຈະຊ່ວຍໃຫ້ການຂາຍຢາມີຄວາມສະດວກ ແລະ ວ່ອງໄວຂຶ້ນກວ່າເກົ່າ.
- ຈະໄດ້ລະບົບລາຍງານທີ່ມີຄວາມຖືກຕ້ອງ ແລະ ຊັດເຈນ.

## 5. ວິທີດຳເນີນການຄົ້ນຄວ້າ (Research Methodology)

ຂັ້ນຕອນ ແລະ ການດຳເນີນໂຄງການໂດຍຫຼັກໆແລ້ວ ໄດ້ອີງໃສ່ຕາມວົງຈອນການພັດທະນາລະບົບແບບໂຄງສ້າງ (Structure Analysis and Design). ເຊິ່ງແບບໂຄງສ້າງຈະປະກອບມີການໄຫຼຂອງຂໍ້ມູນ (DFD) ແລະ (ERD) ນຳໃຊ້ວິທີການແບບ Adapted Waterfall Model.



ຮູບທີ່ 1 ວົງຈອນການພັດທະນາແບບນ້ຳຕົກ (Adapted Waterfall Model)

- **ວາງແຜນ**

ດຳເນີນການຈັດຕັ້ງກຸ່ມຂຽນບົດໂຄງການພຽງພ້ອມກັບການກຳນົດຂໍ້ໂຄງການກຳນົດທົວຂໍ້ໂຄງການ. ຈາກນັ້ນ, ກໍໄດ້ລົງເກັບກຳຂໍ້ມູນຢູ່ຮ້ານຂາຍຢາ ດຣ ດວງຈັນ ເພື່ອໃຫ້ໄດ້ຂໍ້ມູນ ແລະ ຂັ້ນຕອນການເຮັດວຽກໂດຍລວມກ່ຽວກັບການເຮັດວຽກຕ່າງໆຂອງຮ້ານຂາຍຢາ ດຣ ດວງຈັນ.

- **ວິເຄາະ**

ໄລຍະນີ້ພວກເຮົາຈະສຶກສາລະບົບເກົ່າ ແລະ ຄວາມຕ້ອງການຂອງຜູ້ໃຊ້. ຈາກນັ້ນ, ກໍນຳເອົາຂໍ້ມູນຕ່າງໆລວບລວມເອົາຂໍ້ມູນຕ່າງໆທີ່ລວບລວມໄດ້ມາເພື່ອວິເຄາະເປັນຂໍ້ກຳນົດຄວາມຕ້ອງການຂອງລະບົບໃໝ່ພ້ອມທັງແຕ້ມແບບຈຳລອງ DFD ເພື່ອສະແດງເຖິງການໄຫຼຂໍ້ມູນໄປເຖິງຂະບວນການ ແລະ ແຜນວາດ E-R Diagram ເພື່ອສະແດງຄວາມສຳພັນລະຫວ່າງຂໍ້ມູນ.

- **ອອກແບບ**

ໄລຍະນີ້ພວກເຮົາຈະອອກແບບໂປຣແກຣມ, ອອກແບບໜ້າສະແດງຜິນ, ອອກແບບໜ້າປ້ອນຂໍ້ມູນ, ອອກແບບຖານຂໍ້ມູນ ໂດຍເຮັດ Normalization ແລະ ສ້າງ Data Dictionary.

- **ພັດທະນາ**

ດຳເນີນການສ້າງຂໍ້ມູນຕາມທີ່ໄດ້ເຮັດ Data Dictionary ໃນຂັ້ນຕອນກ່ອນໜ້ານັ້ນ. ຈາກນັ້ນ, ກໍລົງມືປະຕິບັດຂຽນ Code ເພື່ອຕິດຕໍ່ ແລະ ເຮັດວຽກກັບຖານຂໍ້ມູນທີ່ໄດ້ສ້າງໄວ້ແລ້ວ.

- **ທົດສອບ**

ສ້າງແບບວິທີການທົດສອບລະບົບ ແລະ ນຳເອົາຊຸດຂໍ້ມູນທີ່ນຳມາທົດສອບຢ່າງໜ້ອຍ 20 records ມາດຳເນີນການທົດສອບແລ້ວສະຫຼຸບຜົນໄດ້ຮັບ ແລະ ທຳການປັບປຸງແກ້ໄຂຈົນໄດ້ຜົນອອກທີ່ຖືກຕ້ອງ

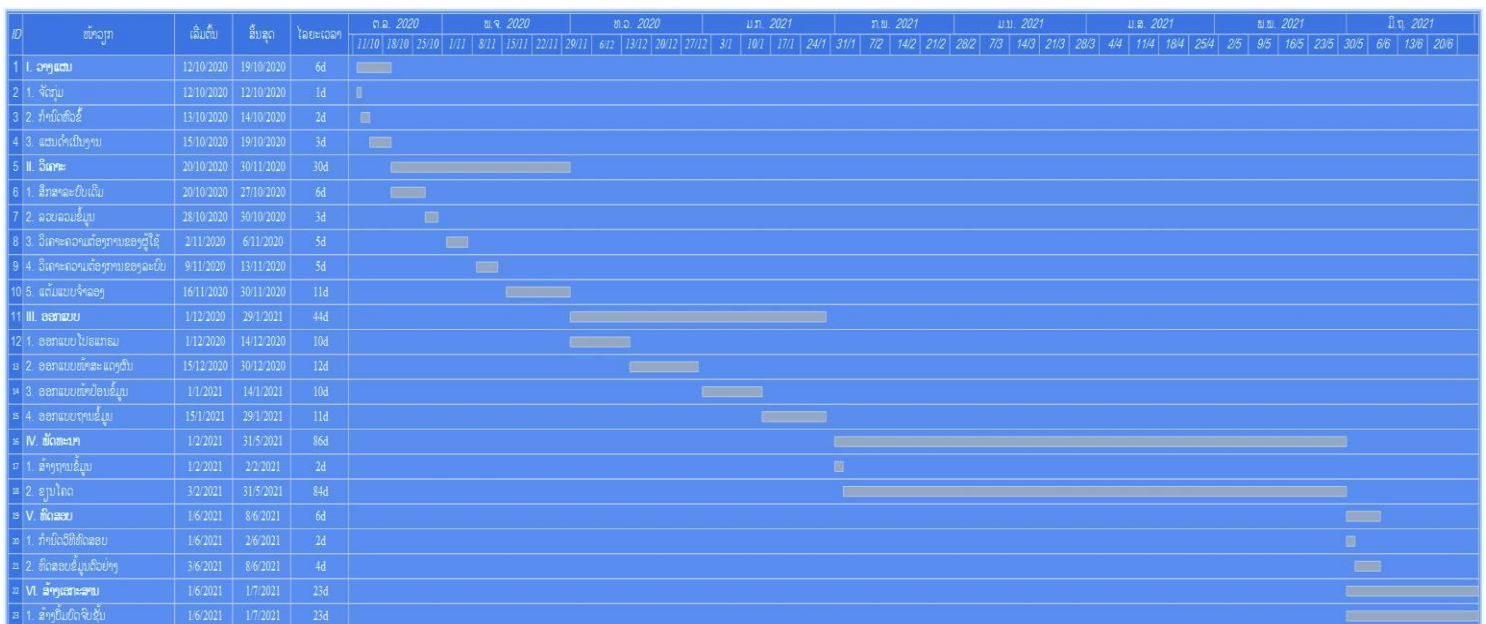
- **ສ້າງເອກະສານ**

ສ້າງປຶ້ມບົດຈົບຊັ້ນ ລະບົບ ຈັດການ ຮ້ານຂາຍຢາ ດຣ ດວງຈັນ ແລ້ວສົ່ງມອບປຶ້ມພ້ອມທັງແຜ່ນ CD ທີ່ປະກອບດ້ວຍ Source Code ,ເອກະສານບົດຈົບຊັ້ນ ແລະ Presentations.

## 6. ສະຖານທີ່ສຶກສາ (Study Site)

ສະຖານທີ່ສຶກສາແມ່ນ ຮ້ານຂາຍຢາ ດຣ ດວງຈັນ. ບ້ານໂສກປ່າຫຼວງ,ເມືອງສີສັດຕະນາກ, ນະຄອນຫຼວງວຽງຈັນ.

## 7. ໄລຍະປະຕິບັດ (Duration)



## 8. ເຄື່ອງມືທີ່ນຳໃຊ້ໃນການພັດທະນາ (Development Tools)

1. Hardware	2. Software
<ul style="list-style-type: none"> <li>➢ ຄອມພິວເຕີ HP Intel(R) Pentium(R) N3710 CPU @1.60 GHz 1.60 GHz, RAM 4GB DDR3 1600MHz</li> </ul>	<ul style="list-style-type: none"> <li>➢ ລະບົບປະຕິບັດການ Windows 10 Professional 64 Bit</li> <li>➢ Microsoft Visio 2016 ໃຊ້ແຕ້ມແຜນວາດການໄຫຼຂໍ້ມູນ (DFD, ER, Flowchart)</li> <li>➢ Studio 3T For MongoDB ແລະ Moon Modeler ໃຊ້ຈັດການຖານຂໍ້ມູນ</li> </ul>

	<ul style="list-style-type: none"> <li>➢ Visual Studio Code ໃຊ້ຊຸມໂຄດດ້ວຍພາສາ JavaScript (React, Nodejs)</li> <li>➢ MS Office 2013 Professional ໃຊ້ເພື່ອສ້າງເອກະສານຕ່າງໆ, ບົດນຳສະເໜີ ແລະ ເຮັດປຶ້ມບົດຈົບຊັ້ນ</li> </ul>
--	--

## 9. ເອກະສານອ້າງອີງ (Reference)

ເອົາມາຈາກປຶ້ມ ໂປຣແກມຂາຍສິນຄ້າຂອງຮ້ານ ນາງແຈ່ມ (ມະນີລາ ວິໄລຄຽນ ດາ ຈັນທະລາ ມີນູ ເພຍລີ ).

## 10. ທົບທວນທິດສະດີ ແລະ ບົດທີ່ກ່ຽວຂ້ອງ

### 10.1. ທົບທວນທິດສະດີທີ່ກ່ຽວຂ້ອງ

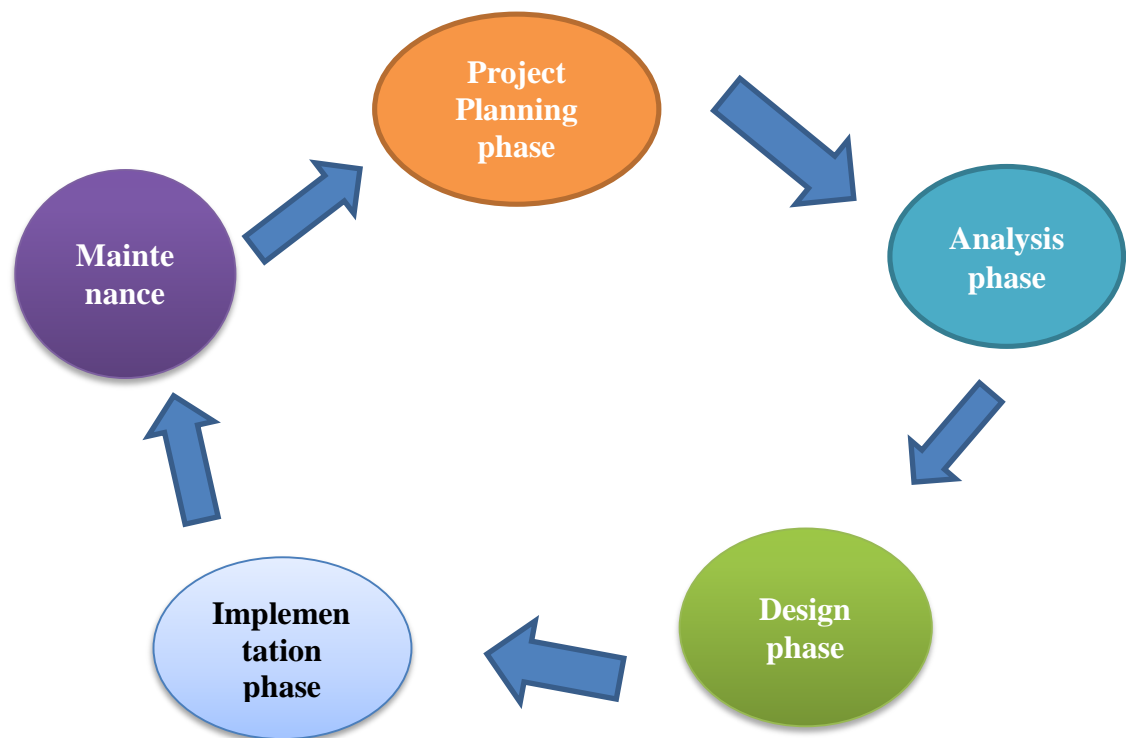
ລະບົບຈັດການຮ້ານຂາຍຢາ ດຣ ດວງຈັນ ແມ່ນພັດທະນາຂຶ້ນໂດຍນຳໃຊ້ທິດສະດີ ຫຼື ຄວາມຮູ້ຈາກ 3 ສ່ວນຄື: ທິດສະດີໃນການວິເຄາະ ແລະ ອອກແບບລະບົບ, ທິດສະດີກ່ຽວກັບການສ້າງຖານຂໍ້ມູນ ແລະ ຄວາມຮູ້ກ່ຽວກັບພາສາໃນການພັດທະນາລະບົບ. ລາຍລະອຽດຂອງແຕ່ລະທິດສະດີແມ່ນຈະໄດ້ນຳສະເໜີໂດຍສັງເຂບດັ່ງຕໍ່ໄປນີ້:

#### 10.1.1. ທິດສະດີກ່ຽວກັບການວິເຄາະ ແລະ ການອອກແບບລະບົບ

ການວິເຄາະລະບົບເປັນຂະບວນການທຳຄວາມເຂົ້າໃຈ ແລະ ກຳນົດລາຍລະອຽດຂອງບັນຫາເພື່ອຈະໄດ້ພິຈາລະນານຳເອົາລະບົບຂໍ້ມູນຂ່າວສານໃດເຂົ້າໄປແກ້ບັນຫາເລົ່ານັ້ນ, ສ່ວນການອອກແບບລະບົບໝາຍເຖິງຂະບວນການກຳນົດລາຍລະອຽດຕ່າງໆວ່າຈະຕ້ອງເຮັດແນວໃດກັບອົງປະກອບຂອງລະບົບຂໍ້ມູນຂ່າວສານເພື່ອຈະໄດ້ນຳໄປໃຊ້ໃຫ້ເກີດຜົນໃນທາງພາຍນອກໄດ້. ສິ່ງທີ່ນຳສະເໜີໃນຫົວຂໍ້ນີ້ໄດ້ຂັດສັນພາກສ່ວນໜຶ່ງຈາກ (ມະນີລາ ວິໄລຄຽນ, ດາ ຈັນທະລາ ແລະ ມີນູ ເພຍລີ, 2015 - 2016) ແລະ (ບຸນມີ ພັນທະວົງ ແລະ ອໍລະດີ ຄຳມະນີວົງ, 2016).

#### 10.1.1.1. ວົງຈອນພັດທະນາລະບົບ

ເປັນຂະບວນການທີ່ສະແດງເຖິງການດຳເນີນຂັ້ນຕອນການເຮັດວຽກຂອງລະບົບຕັ້ງແຕ່ຕົ້ນຈົນຈົບ, ບົດໂຄງການຈົບຊັ້ນນີ້ພວກຂ້າພະເຈົ້າໄດ້ນຳໃຊ້ທິດສະດີການວິເຄາະ ແລະ ອອກແບບລະບົບແບບໂຄງທີ່ປະກອບມີ 5 ໄລຍະຄື:



- 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)

### ກ. ໄລຍະທີ່ 1 ການວາງແຜນໂຄງການ (Project planning phase)

ການວາງແຜນໂຄງການຈັດເປັນຂະບວນການພື້ນຖານໃນຄວາມເຂົ້າໃຈຢ່າງເລິກເຊິ່ງວ່າ ເປັນ ຫຍັງຕ້ອງສ້າງລະບົບໃໝ່ ທີ່ມີງານຕ້ອງພິຈາລະນາວ່າ ຈະຕ້ອງດຳເນີນງານຕໍ່ໄປແນວໃດກ່ຽວກັບ ຂະບວນການສ້າງລະບົບໃໝ່ ກ່ອນອື່ນໝົດຕ້ອງມີຈຸດກຳເນີດຂອງລະບົບງານເຊິ່ງໂດຍປົກກະຕິແລ້ວ ຈຸດກຳເນີດຂອງລະບົບງານມັກເກີດຂຶ້ນຈາກຜູ້ໃຊ້ລະບົບເປັນຜູ້ທີ່ຕິດແທດກັບລະບົບໂດຍກົງເຮັດໃຫ້ມີ ຄວາມໃກ້ສືດກັບລະບົບງານທີ່ດຳເນີນຢູ່ຫຼາຍທີ່ສຸດເມື່ອຜູ້ໃຊ້ລະບົບມີຄວາມຕ້ອງການປັບປຸງລະບົບ ງານດັ່ງນັ້ນຈຶ່ງຖືເປັນຈຸດເລີ່ມຕົ້ນໃນບົດບາດຂອງນັກວິເຄາະລະບົບວ່າຈະຕ້ອງສຶກສາເຖິງຂອບເຂດ ຂອງບັນຫາທີ່ຜູ້ໃຊ້ລະບົບກຳລັງປະສົບບັນຫາຢູ່ ແລະ ຈະດຳເນີນການແກ້ໄຂແນວໃດ ສຶກສາເຖິງ ຄວາມເປັນໄປໄດ້ວ່າລະບົບໃໝ່ທີ່ຈະພັດທະນາຂຶ້ນມານັ້ນມີຄວາມເປັນໄປໄດ້ ແລະ ຄຸ້ມຄ່າກັບການຈະ ລົງທຶນຫຼືບໍ່.

ແນວໃດກໍ່ຕາມໄລຍະຂອງການວາງແຜນໂຄງການປົກກະຕິມັກຈະມີໄລຍະເວລາທີ່ສັ້ນ ແຕ່ກໍ່ຖື ວ່າເປັນຂັ້ນຕອນທີ່ສຳຄັນທີ່ຈະໃຫ້ເກີດຜົນສຳເລັດໄດ້ດັ່ງນັ້ນໃນໄລຍະຂອງການວາງແຜນໂຄງການຈຶ່ງ ໄດ້ອາໄສນັກວິເຄາະລະບົບທີ່ມີຄວາມຮູ້ ແລະ ປະສົບການສູງເນື່ອງຈາກວ່າຫາກນັກວິເຄາະລະບົບບໍ່

ເຂົ້າໃຈເຖິງບັນຫາອັນແທ້ຈິງທີ່ເກີດຂຶ້ນ ກໍ່ຈະບໍ່ສາມາດພັດທະນາລະບົບຂຶ້ນມາເພື່ອແກ້ໄຂບັນຫາໃຫ້ຖືກ ຈຸດໄດ້ ແລະ ມັກຈະມີໂຄງການພັດທະນາລະບົບຫຼາຍໂຄງການທີ່ຢູ່ຈາກໄດ້ດຳເນີນການພັດທະນາ ແລະ ນຳມາໃຊ້ງານແລ້ວປະກົດວ່າບໍ່ສາມາດຕອບສະໜອງຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ງານເຊິ່ງຖືວ່າ ເປັນເລື່ອງທີ່ກໍ່ໃຫ້ເກີດຄວາມສູນເສຍທັງທາງດ້ານການລົງທຶນ ແລະ ໄລຍະເວລາ.

ສະຫຼຸບລວມແລ້ວໄລຍະຂອງການວາງແຜນໂຄງການປະກອບມີກິດຈະກຳຕ່າງໆດັ່ງນີ້:

- ກຳນົດບັນຫາ Problem Definition
- ສຶກສາຄວາມເປັນໄປໄດ້ຂອງໂຄງການ Feasibility study
- ສ້າງຕາຕະລາງກຳນົດເວລາໂຄງການ Project scheduling
- ຈັດຕັ້ງທີມງານໂຄງການ Staff the project

## **ຂ. ໄລຍະທີ່ 2 ການວິເຄາະ (Analysis Phase)**

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

ຈຸດປະສົງຫຼັກຂອງການວິເຄາະຄືຈະຕ້ອງສຶກສາ ແລະ ສ້າງຄວາມເຂົ້າໃຈໃນຄວາມຕ້ອງການ ຕ່າງໆທີ່ໄດ້ລວບລວມມາດັ່ງນັ້ນການລວບລວມຄວາມຕ້ອງການ (Requirements Gathering) ຈຶ່ງຈັດ ເປັນງານສ່ວນພື້ນຖານຂອງການວິເຄາະລະບົບໂດຍຂໍ້ມູນຄວາມຕ້ອງການເຫຼົ່ານີ້ນັກວິເຄາະລະບົບຈະ ນຳມາວິເຄາະເພື່ອທີ່ຈະປະເມີນວ່າຄວນມີຫຍັງແນ່ທີ່ລະບົບໃໝ່ຕ້ອງດຳເນີນການ ແລະ ດ້ວຍເຫດນີ້ ເອງການກຳນົດລາຍລະອຽດກ່ຽວກັບຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ (User Requirements) ຈະທະວີ ຄວາມສຳຄັນຫຼາຍຂຶ້ນເປັນລຳດັບສຳລັບລະບົບທີ່ມີຄວາມສັບຊ້ອນສູງ ແລະ ນັກວິເຄາະຕ້ອງເອົາໃນໃສ່ ກັບການລວບລວມຄວາມຕ້ອງການຈາກຜູ້ໃຊ້ບໍ່ຄວນກຳນົດຄວາມຕ້ອງການຂຶ້ນເອງໂດຍໃຊ້ຄວາມຄິດ ສ່ວນຕົວຂອງຕົນເອງເປັນຫຼັກທີ່ປະເມີນຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ລະບົບບໍ່ເຂົ້າກັບຈຸດປະສົງ ແລະ ຫາກມີການພັດທະນາລະບົບຕໍ່ໄປຈົນແລ້ວລະບົບງານທີ່ໄດ້ກໍ່ຈະບໍ່ກົງກັບຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ ລະບົບຢ່າງແທ້ຈິງເຮັດໃຫ້ຕ້ອງມີຄວາມປັບປຸງຫຼືປ່ຽນແປງພາຍຫຼັງ.

ນັກວິເຄາະລະບົບສາມາດລວບລວມຄວາມຕ້ອງການຕ່າງໆໄດ້ຈາກການສັງເກດ, ການເຮັດ ວຽກຂອງຜູ້ໃຊ້, ການໃຊ້ເຕັກນິກ, ການສຳພາດ ຫຼື ການໃຊ້ແບບສອບຖາມການອ່ານເອກະສານກ່ຽວ ກັບການປະຕິບັດງານຂອງລະບົບງານປັດຈຸບັນລະບຽບກົດເກນຂອງບໍລິສັດ ແລະ ການມອບໝາຍຕຳ ແໜ່ງໜ້າທີ່ຮັບຜິດຊອບເຊິ່ງໃນຊ່ວງຂອງການເກັບກຳຂໍ້ມູນຄວາມຕ້ອງການກໍ່ຈະພົບກັບຜູ້ໃຊ້ໃນລະດັບ ຕ່າງໆທີ່ເຮັດໃຫ້ຮູ້ເຖິງບັນຫາແລະແນວທາງການແກ້ໄຂບັນຫາທີ່ແນະນຳໂດຍຜູ້ໃຊ້ດັ່ງນັ້ນການເກັບກຳ ຄວາມຕ້ອງການຈຶ່ງເປັນກິດຈະກຳທີ່ສຳຄັນເພື່ອຄົ້ນຫາຄວາມຈິງແລະຕ້ອງສ້າງຄວາມຕ້ອງການເຊິ່ງກັນ ແລະ ກັນເພື່ອສະຫຼຸບອອກມາເປັນຂໍ້ກຳນົດທີ່ມີຄວາມຊັດເຈນໂດຍຂໍ້ກຳນົດເຫຼົ່ານີ້ເມື່ອຜູ້ໃຊ້ທີ່ກ່ຽວຂ້ອງ ໄດ້ອ່ານແລ້ວຈະຕ້ອງສຶກສາຄວາມໝາຍໄດ້ກົງກັນ.



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

ສະຫຼຸບໄລຍະຂອງການວິເຄາະລະບົບປະກອບມີກິດຈະກຳຕ່າງໆດັ່ງນີ້:

- ວິເຄາະລະບົບງານປັດຈຸບັນ.
- ເກັບກຳຄວາມຕ້ອງການໃນດ້ານຕ່າງໆແລະນຳມາວິເຄາະເພື່ອສະຫຼຸບເປັນຂໍ້ກຳນົດຊັດເຈນ
- ນຳຂໍ້ກຳນົດມາພັດທະນາອອກມາເປັນຄວາມຕ້ອງການຂອງລະບົບໃໝ່.
- ສ້າງແບບຈຳລອງຂະບວນການຂອງລະບົບໃໝ່ໂດຍການແຕ້ມແຜນພາບກະແສຂໍ້ມູນ (DataFlowDiagram:DFD).
- ສ້າງແບບຈຳລອງຂໍ້ມູນໂດຍການແຕ້ມ Entity Relationship Diagram:ERD.

### ຄ. ໄລຍະທີ່ 3 ການອອກແບບ (Design Phase)

ໄລຍະການອອກແບບເປັນການພິຈາລະນາວ່າລະບົບລະດຳເນີນການໄປໄດ້ແນວໃດເຊິ່ງກ່ຽວຂ້ອງກັບຍຸດທະວິທີການຂອງການອອກແບບທີ່ວ່າດ້ວຍການຕັດສິນໃຈວ່າຈະພັດທະນາລະບົບໃໝ່ດ້ວຍແນວທາງໃດເຊັ່ນພັດທະນາຂຶ້ນເອງ, ຊື້ໂປຣແກຣມສຳເລັດຮູບຫຼືວ່າຈ້າງບໍລິສັດພັດທະນາລະບົບໃຫ້ເປັນຕົ້ນ. ນອກຈາກນີ້ໄລຍະການອອກແບບຈະກ່ຽວຂ້ອງກັບການອອກແບບທາງດ້ານສະຖາປັດຕະຍະກຳລະບົບທີ່ກ່ຽວຂ້ອງກັບອຸປະກອນຮາດແວ, ຊອບແວ ແລະ ເຄືອຂ່າຍ.

ການອອກແບບລາຍງານ (OutDesign) ການອອກແບບໜ້າຈໍເພື່ອປະຕິສຳພັນກັບຜູ້ໃຊ້ (User Interface), ການອອກແບບຜັງງານລະບົບ (SystemFlowchart), ເຊິ່ງລວມເຖິງລາຍລະອຽດຂອງໂປຣແກຣມ (SpecificProgram), ຖານຂໍ້ມູນ (Database) ແລະ ຂໍ້ມູນທີ່ກ່ຽວຂ້ອງແນວໃດກໍ່ຕາມເຖິງວ່າກິດຈະກຳບາງສ່ວນຂອງໄລຍະອອກແບບນີ້ສ່ວນໃຫຍ່ຈະຖືກດຳເນີນການໄປບາງສ່ວນແລ້ວໃນໄລຍະຂອງການວິເຄາະແຕ່ໄລຍະການອອກແບບນີ້ຈະເນັ້ນເຖິງການດຳເນີນການແກ້ໄຂບັນຫາແນວໃດຫຼາຍກ່ວາໂດຍການນຳຜົນຂອງແບບຈຳລອງທາງ Logical Model. ທີ່ໄດ້ຈາກການວິເຄາະມາພັດທະນາມາເປັນແບບຈຳລອງທາງ Physical Model.

- ການວິເຄາະຈະເນັ້ນແກ້ໄຂບັນຫາຫຍັງແດ່.
- ການອອກແບບຈະເນັ້ນການແກ້ໄຂບັນຫາແນວໃດ.
- ສະຫຼຸບໄລຍະການອອກແບບ.
- ພິຈາລະນາແນວທາງໃນການພັດທະນາລະບົບ.
- ອອກແບບສະຖາປັດຕະຍະກຳລະບົບ (Architecture Design).
- ອອກແບບຖານຂໍ້ມູນ (Database Design).
- ອອກແບບການສະແດງຜົນ (Output Design).
- ອອກແບບການປ້ອນຂໍ້ມູນ (Input Design).

- ອອກແບບສ່ວນຕິດຕໍ່ກັບຜູ້ໃຊ້ (User Interface).
- ສ້າງຕົ້ນແບບ (Prototype).
- ອອກແບບໂປຣແກມ (Structure Chart).

## ງ. ໄລຍະທີ 4 ການນຳໄປໃຊ້ (Implementation Phase)

ໃນໄລຍະການນຳໄປໃຊ້ຈະເຮັດໃຫ້ລະບົບເກີດຜົນຂຶ້ນມາໂດຍການສ້າງລະບົບທົດສອບລະບົບແລະການຕິດຕັ້ງລະບົບໂດຍຈຸດປະສົງຫຼັກຂອງກິດຈະກຳໃນໄລຍະນີ້ບໍ່ແມ່ນພຽງຄວາມໜ້າເຊື່ອຖືຂອງລະບົບຫຼືລະບົບສາມາດເຮັດວຽກໄດ້ດີເທົ່ານັ້ນແຕ່ຕ້ອງໝັ້ນໃຈວ່າຜູ້ໃຊ້ລະບົບຕ້ອງໄດ້ຮັບການຝຶກອົບຮົມເພື່ອໃຊ້ງານລະບົບແລະຄວາມຄາດຫວັງໃນອົງກອນທີ່ຕ້ອງການຜົນຕອບແທນໃນດ້ານດີກັບການໃຊ້ລະບົບໃໝ່ລຳດັບກິດຈະກຳຕ່າງໆທຸກກິດຈະກຳຕ້ອງເຂົ້າມາດຳເນີນການຮ່ວມກັນໃນລະໂຍນີ້ເພື່ອໃຫ້ລະບົບການປະຕິບັດງານໄດ້ຮັບຄວາມປະສົບຜົນສຳເລັດໄດ້ໂດຍດີ.

ສະຫຼຸບໄລຍະການນຳໄປໃຊ້ຈະປະກອບມີກິດຈະກຳຕ່າງໆດັ່ງຕໍ່ໄປນີ້:

- ສ້າງລະບົບຂຶ້ນມາດ້ວຍການຂຽນໂປຣແກມ.
- ກວດສອບຄວາມຖືກຕ້ອງທາງດ້ານ Verification ແລະ Validation ແລະດຳເນີນການທົດສອບລະບົບ.
- ແປງຂໍ້ມູນ (Convert Data).
- ຕິດຕັ້ງລະບົບ (System Installation) ແລະສ້າງເອກກະສານຄູ່ມື.
- ຝຶກອົບຮົມຜູ້ໃຊ້ ແລະ ປະເມີນຜົນລະບົບໃໝ່.

ສຳລັບການສ້າງລະບົບຫຼືການຂຽນໂປຣແກມນັ້ນສາມາດໃຊ້ວິທີການຂຽນໂປຣແກມດ້ວຍພາສາຄອມພິວເຕີເຊັ່ນ: ການໃຊ້ພາສາ Visual Basic, C#, PHP, Java... ນອກຈາກນີ້ຍັງມີເຕັກນິກອື່ນໆເຊັ່ນ: ເຄື່ອງມືໃນການພັດທະນາ Application ເຊິ່ງເປັນຊອບແວທີ່ເປັນແຫຼ່ງລວມຂອງເຄື່ອງມືຕ່າງໆທີ່ໃຊ້ເພື່ອພັດທະນາ Application ເຮັດໃຫ້ຜູ້ຂຽນໂປຣແກມບໍ່ເຮັດວຽກໜັກຄືເມື່ອກ່ອນມີແຕ່ຮຽນຮູ້ແລະປະຍຸກໃຊ້ເຄື່ອງມືເຫຼົ່ານັ້ນກໍສາມາດພັດທະນາລະບົບງ່າຍຂຶ້ນ.

## ຈ. ໄລຍະທີ 5 ການບຳລຸງຮັກສາ (Maintenance Phase)

ໂດຍປົກກະຕິແລ້ວໄລຍະການບຳລຸງຮັກສາຈະບໍ່ນຳເຂົ້າໄປລວມໃນສ່ວນຂອງ SDLC ຈົນກະທັ້ງລະບົບມີການຕິດຕັ້ງເພື່ອໃຊ້ງານແລ້ວເທົ່ານັ້ນໄລຍະນີ້ຈະໃຊ້ເວລາຍາວນານທີ່ສຸດເມື່ອທຽບກັບໄລຍະອື່ນໆທີ່ຜ່ານມາເນື່ອງຈາກລະບົບຈະຕ້ອງໄດ້ຮັບການບຳລຸງຮັກສາຕະຫຼອດໄລຍະເວລາທີ່ມີການໃຊ້ລະບົບສິ່ງທີ່ຄາດຫວັງຂອງໜ່ວຍງານກໍຄືຕ້ອງການໃຫ້ລະບົບໃຊ້ງານຍາວນານຫຼາຍປີລະບົບສາມາດຮອງຮັບເຕັກໂນໂລຊີໃໝ່ໆໃນອະນາຄົດໄດ້ດັ່ງນັ້ນໃນຊ່ວງໄລຍະເວລາດັ່ງກ່າວຈຶ່ງສາມາດເພີ່ມເຕີມຄວາມສາມາດຂອງລະບົບໃຫ້ມີປະສິດທິພາບສູງຂຶ້ນພ້ອມທັງການແກ້ໄຂປັບປຸງໂປຣແກມໃນກໍລະນີທີ່ເຫັນຂໍ້ຜິດພາດ.

- ການບຳລຸງຮັກສາລະບົບ (System Maintenance).
- ການເພີ່ມເຕີມຄວາມສາມາດໃໝ່ໆເຂົ້າໃນລະບົບ (Enhance System).

- ສະໜັບສະໜູນງານຂອງຜູ້ໃຊ້ (Support the User).

ຈາກໄລຍະຕ່າງໆຕາມຂັ້ນຕອນການພັດທະນາລະບົບຕາມແບບແຜນຂອງ SDLC ຈະເຫັນວ່າມີການໃຊ້ຄຳວ່າໄລຍະແລະກິດຈະກຳເຊິ່ງສາມາດອະທິບາຍລາຍລະອຽດເພື່ອໃຫ້ເກີດຄວາມເຂົ້າໃຈກົງກັນດັ່ງນີ້:


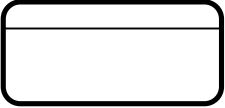
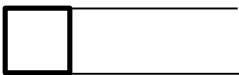


- ໄລຍະ(Phase) ຄືກຸ່ມກິດຈະກຳທີ່ກ່ຽວຂ້ອງກັນ.
- ກິດຈະກຳ (Activity) ຄືກຸ່ມຂອງງານທີ່ກ່ຽວຂ້ອງກັນ.
- ງານ (Task) ຄືງານທີ່ດຳເນີນການເຊິ່ງຖືເປັນງານທີ່ນ້ອຍທີ່ສຸດ.

#### 10.1.1.2. ແຜນວາດການໄຫຼຂໍ້ມູນ (Data Flow Diagram DFD)

- ເປັນແຜນພາບທີ່ສະຫຼຸບລວມຂໍ້ມູນທັງໝົດໄດ້ຈາກການວິເຄາະໃນລັກສະນະຂອງຮູບແບບທີ່ເປັນໂຄງສ້າງ.
- ເປັນຂໍ້ຕົກລົງຮ່ວມກັນລະຫວ່າງນັກວິເຄາະລະບົບ ແລະ ຜູ້ຊົມໃຊ້.
- ເປັນແຜນພາບທີ່ໃຊ້ໃນການພັດທະນາຕໍ່ໃນຂັ້ນຕອນຂອງການອອກແບບ.
- ຮູ້ທີ່ໄປທີ່ມາຂອງຂໍ້ມູນທີ່ໄຫຼໃນຂະບວນການຕ່າງໆ.

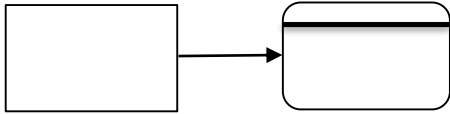
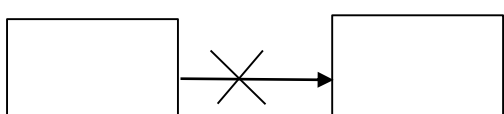
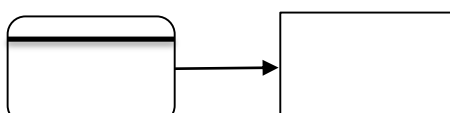
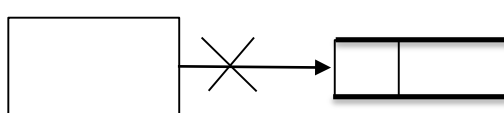
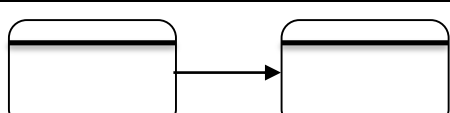
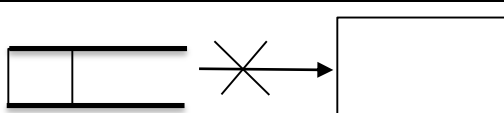
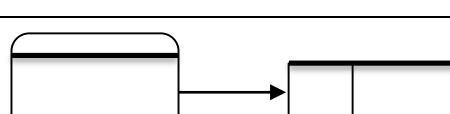
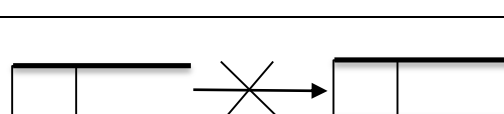
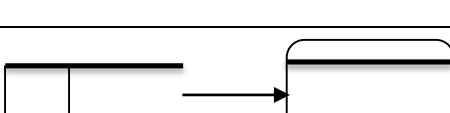
#### 10.1.1.3. ສັນຍາລັກທີ່ໃຊ້ໃນແຜນວາດຂໍ້ມູນ

ຕາຕະລາງ ສະແດງສັນຍາລັກ Data Flow Diagram

ຊື່	ສັນຍາລັກ	ຄວາມໝາຍ
Boundary Or External Entity		ຂອບເຂດໝາຍເຖິງພາກສ່ວນທີ່ກ່ຽວຂ້ອງເກັບລະບົບເຊິ່ງລະບົບບໍ່ສາມາດຄວບຄຸມໄດ້
Process		ປະມວນຜົນຫຼືຫນ້າວຽກທີ່ເຮັດໃນໂຄງການນັ້ນໆ
Data Store		ບ່ອນຈັດເກັບຂໍ້ມູນ
Data Flow		ການໄຫຼຂອງຂໍ້ມູນ
Real-Time Link		ການເຊື່ອມໂຍງແບບໄກທີ່ມີການຕອບກັບແບບທັນທີທັນໃດ

#### 10.1.1.4. ຫຼັກການຂຽນແຜນວາດການໄຫຼຂໍ້ມູນ

ຕາຕະລາງທີ 10 ຮູບການປຸງປະໂຫຍດແຜນວາດການໄຫຼຂໍ້ມູນທີ່ບໍ່ຖືກຕ້ອງ ແລະ ຖືກຕ້ອງ

ອະນຸຍາດ	ບໍ່ອະນຸຍາດ
	
	
	
	
	

##### 1) Process:

- ເມື່ອມີຂໍ້ມູນເຂົ້າໄປທີ່ Process ກໍ່ຕ້ອງມີຂໍ້ມູນ ຫຼື ຜົນຮັບອອກມາຈາກ Process ເຊັ່ນກັນຈະເປັນໄປບໍ່ໄດ້ທີ່ມີສະເພາະຂໍ້ມູນເຂົ້າຢ່າງດຽວ.

##### 2) Data store:

- ຂໍ້ມູນຈະໄຫຼຈາກ Data store ໜຶ່ງໄປຫາ Data store ໜຶ່ງໂດຍກົງບໍ່ໄດ້ຈະຕ້ອງຜ່ານ Process ເທົ່ານັ້ນ
- ຂໍ້ມູນທີ່ສົ່ງຜ່ານ External entity ບໍ່ສາມາດໄຫຼເຂົ້າໄປ Data store ໂດຍກົງໄດ້ຈະຕ້ອງໃຊ້ Process ເປັນຕົວກາງໃນການເຊື່ອມໂຍງເພື່ອຈັດເກັບຂໍ້ມູນໃນ Data store
- ຂໍ້ມູນທີ່ໄຫຼຜ່ານຈາກ Data store ບໍ່ສາມາດເຊື່ອໂຍງເຂົ້າກັບ External entity ໄດ້ໂດຍກົງຈະຕ້ອງຜ່ານ Process ເທົ່ານັ້ນ.

##### 3) External entity:

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

##### 4) Data flow:

- ການໄຫຼຂໍ້ມູນທີ່ມີຫົວຊີ້ໄປທີ່ Process ໝາຍເຖິງ Process ມີການອ່ານ ຫຼື ການດຶງຂໍ້ມູນຈາກ Data store ມາໃຊ້ວຽກ
- ການໄຫຼຂໍ້ມູນຈາກ Process ທີ່ມີຫົວລູກສອນຊີ້ໄປຍັງ Data store ໝາຍເຖິງການ Update ຫຼື ການເພີ່ມຂໍ້ມູນລົງໄປທີ່ Data store








#### 10.1.1.5. Flowchart

ສັນຍະລັກ Flowchart ຄື ຮູບພາບທີ່ໃຊ້ແທນຄວາມໝາຍການເຮັດວຽກງານໃນລັກສະນະຕ່າງໆ ພາຍໃນແຜນຜັງ (Flowchart) ປະກອບໄປດ້ວຍ ການເລີ່ມຕົ້ນ (Start), ການຈົບ (End), ການກະທຳ (Process), ການນຳເຂົ້າຂໍ້ມູນ (Input), ການສະແດງຜົນຂໍ້ມູນ (Output), ການຕັດສິນໃຈ (Decision), ຄຳອະທິບາຍ (Annotation), ຈຸດເຊື່ອມຕໍ່ (Connector), ທິດທາງການເຮັດວຽກງານ (Direction Flow) (ຄວາມໝາຍສັນຍະລັກ Flowchart ຄວາມໝາຍ ແລະ ວິທີໃຊ້ງານ <http://share.olanlab.com/th/it/blog/view/211> Dec 01, 2017).

ສັນຍະລັກເຫຼົ່ານີ້ເມື່ອຖືກນຳມາເຊື່ອມຕໍ່ກັນ ຈະກາຍເປັນ "ແຜນຜັງ (Flowchart)" ທີ່ສະແດງລຳດັບຂັ້ນຕອນການເຮັດວຽກງານເພື່ອ

- ເປັນເຄື່ອງມືໃນການຈັດລຳດັບຄວາມຄິດ
- ເັນລຳດັບຂັ້ນຕອນການເຮັດວຽກງານທີ່ຊັດເຈນ

#### ສັນຍະລັກ Flowchart

ຮູບພາບສັນຍະລັກ	ຄວາມໝາຍຂອງສັນຍະລັກ
	ການເລີ່ມຕົ້ນຫລືຈົບ Flowchart (Start ຫລື End)
	ການກະທຳ (Process) ຖືກໃຊ້ເພື່ອສະແດງທີ່ການກະທຳໃນ Flowchart
	ຮັບຂໍ້ມູນ
	ການຕັດສິນໃຈ (Decision)
	ສະແດງຜົນທາງຈຳພາບ
	ຈຸດເຊື່ອມຕໍ່ (Connector)
	ທິດທາງການເຮັດວຽກງານ (Direction Flow)

### 10.1.2. ທິດສະດີກ່ຽວກັບຖານຂໍ້ມູນ

Database ຫຼື ຖານຂໍ້ມູນຄືກຸ່ມຂອງຂໍ້ມູນທີ່ຖືກເກັບລວບລວມໄວ້ໂດຍມີຄວາມສໍາພັນເຊິ່ງກັນ ແລະ ກັນໂດຍບໍ່ໄດ້ບັງຄັບວ່າຂໍ້ມູນທັງໝົດນີ້ຈະຕ້ອງເກັບໄວ້ໃນແຟມຂໍ້ມູນດຽວກັນ ຫຼື ແຍກເກັບ ຫຼາຍໆແຟມຂໍ້ມູນ

ລະບົບຖານຂໍ້ມູນຄືລະບົບທີ່ລວບລວມຂໍ້ມູນຕ່າງໆທີ່ກ່ຽວຂ້ອງກັນເຂົ້າໄວ້ດ້ວຍກັນຢ່າງມີລະບົບ, ມີຄວາມສໍາພັນລະຫວ່າງຂໍ້ມູນຕ່າງໆທີ່ຊັດເຈນໃນລະບົບຖານຂໍ້ມູນຈະປະກອບດ້ວຍແຟມຂໍ້ມູນຫຼາຍ ແຟມທີ່ມີຂໍ້ມູນກ່ຽວຂ້ອງກັນ, ສໍາພັນກັນເຂົ້າໄວ້ດ້ວຍກັນຢ່າງເປັນລະບົບ ແລະ ເປີດໂອກາດໃຫ້ ຜູ້ໃຊ້ ສາມາດໃຊ້ງານ ແລະ ຮັກສາປ້ອງກັນຂໍ້ມູນເຫຼົ່ານີ້ໄດ້ຢ່າງມີປະສິດທິພາບໂດຍມີຊອບແວທີ່ປຸງປະ ເຫມືອນຊື່ກາງລະຫວ່າງຜູ້ໃຊ້ ແລະ ໂປຣແກຣມຕ່າງໆທີ່ກ່ຽວຂ້ອງກັບການໃຊ້ຖານຂໍ້ມູນເອີ້ນວ່າລະບົບ ຈັດການຖານຂໍ້ມູນ ຫຼື DBMS

#### 10.1.2.1. ການເຮັດ Normalization

Normalization ເປັນຫຼັກການໜຶ່ງທີ່ຜູ້ອອກແບບຖານຂໍ້ມູນຈະຕ້ອງນໍາມາໃຊ້ໃນການແບ່ງຂໍ້ມູນ ທີ່ຢູ່ໃນຮູບແບບທີ່ຊໍ້າຊ້ອນ ໃຫ້ຢູ່ໃນຮູບແບບທີ່ງ່າຍຕໍ່ການນໍາໄປໃຊ້ງານ ແລະ ກໍ່ໃຫ້ເກີດບັນຫນ້ອຍທີ່ ສຸດ.

#### ກ. ຈຸດປະສົງຂອງການເຮັດ Normalization

- ຫຼຸດຜ່ອນຄວາມຊໍ້າຊ້ອນຂອງຂໍ້ມູນ ເມື່ອຫຼຸດຄວາມຊໍ້າຊ້ອນຈະເຮັດໃຫ້ຫຼຸດເນື້ອທີ່ໃນການຈັດເກັບຂໍ້ມູນ
- ຫຼຸດບັນຫາຄວາມບໍ່ຖືກຕ້ອງຂອງຂໍ້ມູນ ເມື່ອຂໍ້ມູນບໍ່ເກີດຄວາມຊໍ້າຊ້ອນເຮັດໃຫ້ການປັບປຸງຂໍ້ມູນ ສາມາດເຮັດໄດ້ຈາກແຫຼ່ງຂໍ້ມູນພຽງບ່ອນດຽວ
- ຫຼຸດຄວາມຜິດພາດທີ່ເກີດຈາກການປັບປຸງຂໍ້ມູນ (update anomalies)

#### ຂ. ຂັ້ນຕອນການເຮັດ Normalization

- Normalization ລະດັບ 1 ຫຼືເອີ້ນວ່າ 1NF
- Normalization ລະດັບ 2 ຫຼືເອີ້ນວ່າ 2NF
- Normalization ລະດັບ 3 ຫຼືເອີ້ນວ່າ 3NF

#### 10.1.2.2. ແຜນວາດຄວາມສໍາພັນ Entity (ER Diagram)

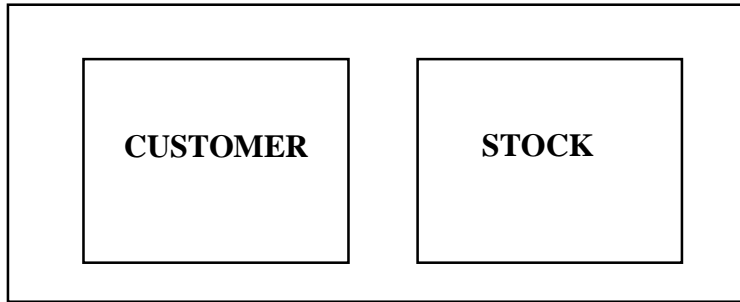
ການອອກແບບຈໍາລອງຖານຂໍ້ມູນ ER ນັ້ນກ່ອນອື່ນຕ້ອງຮູ້ຈັກຄວາມໝາຍສັນຍາລັກຕ່າງໆທີ່ໃຊ້ ໃນການອອກແບບຈໍາລອງ ER ໄດ້ຢ່າງຖືກຕ້ອງດັ່ງນັ້ນໃນຫົວຂໍ້ນີ້ຈະເວົ້າເຖິງຄວາມໝາຍ ແລະ ການ ໃຊ້ງານສັນຍາລັກຕ່າງໆຂອງ ER.

##### 1) ເອັນຕິຕີ (Entity)

ເອັນຕິຕີຄືວັດຖຸທີ່ເຮົາສົນໃຈເຊິ່ງອາດເປັນໄດ້ທັງບຸກຄົນ, ສະຖານທີ່, ວັດຖຸ, ເຫດການ ຫຼື ແນວ ຄິດທີ່ກໍ່ໃຫ້ເກີດກຸ່ມຂອງຂໍ້ມູນທີ່ຕ້ອງການເອັນຕິຕີແບ່ງອອກເປັນ 2 ປະເພດຄື:

##### 2) Strong Entity:

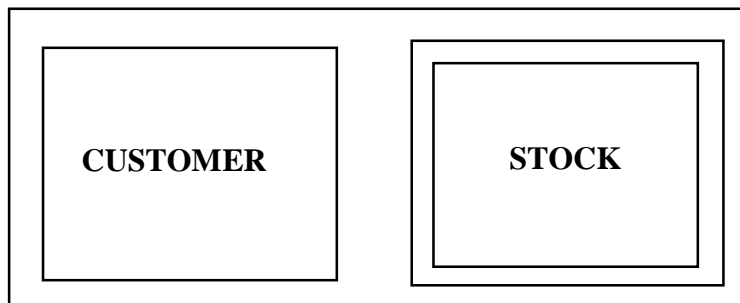
ເປັນເອັນຕິຕີທີ່ເກີດຂຶ້ນດ້ວຍຕົນເອງເປັນອິດສະຫຼະບໍ່ຂຶ້ນກັບເອັນຕິຕີໃດສັນຍາລັກທີ່ໃຊ້ຄືຮູບສີ່ຫຼ່ຽມ ແລະ ສາມາດເອີ້ນ Strong Entity ໄດ້ອີກຊື່ໜຶ່ງວ່າ Regular Entity.



ຮູບທີ 2 ຮູບ Strong Entity

### 3) Weak Entity:

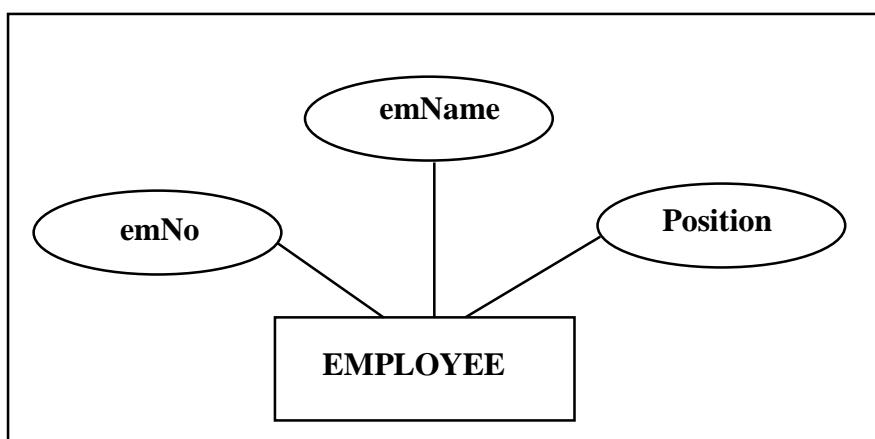
ເອັນຕິຕີຊະນິດນີ້ຈະຂຶ້ນກັບເອັນຕິຕີຊະນິດອື່ນໆບໍ່ສາມາດເກີດຂຶ້ນໄດ້ຕາມລຳພັງ ແລະ ຈະຖືກລົບເມື່ອເອັນຕິຕີໜຶ່ງຖືກລົບອອກ ສັນຍາລັກທີ່ໃຊ້ຄືຮູບສີ່ຫຼ່ຽມຊ້ອນກັນ.



ຮູບທີ 3 ຮູບ Weak Entity

### 4) ແອັດທຣີບິວ (Attribute)

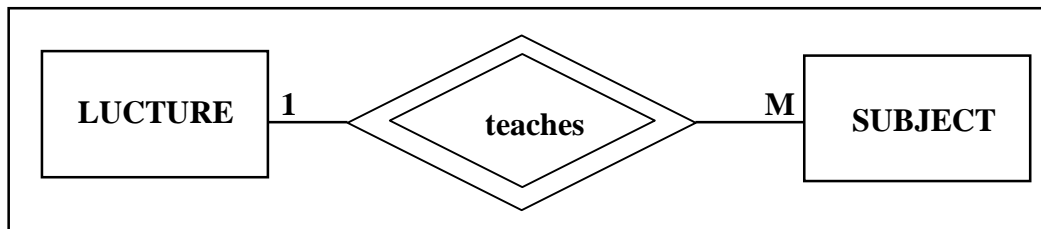
ແອັດທຣີບິວຄືຄຸນສົມບັດຂອງສົມບັດຂອງເອັນຕິຕີສັນຍາລັກຂອງແອັດທຣີບິວຈະເປັນຮູບວົງມົນ ແອັດທຣີບິວໃດທີ່ຖືກໃຊ້ເປັນຄືໜ້າຈະຖືກຂີດເສັ້ນກ້ອງກຳກັບໄວ້.



ຮູບທີ 4 ຮູບ Attribute

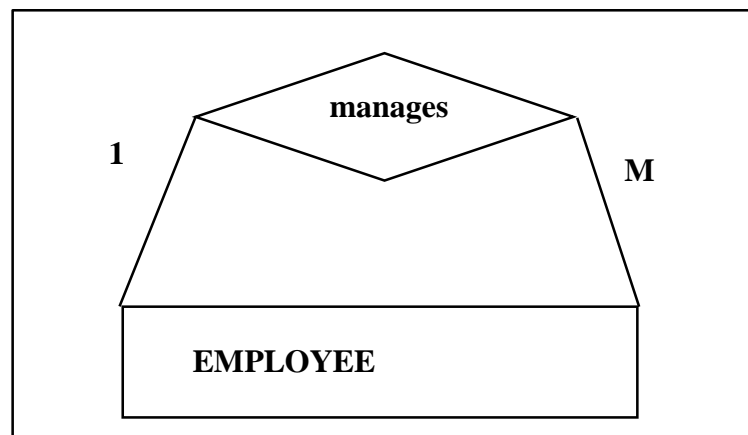
## 5) ຄວາມສຳພັນ (Relation)

ຄວາມສຳພັນໃນທີ່ນີ້ໝາຍເຖິງຄວາມສຳພັນລະຫວ່າງເອັນຕິຕີໂດຍແຕ່ລະຄວາມສຳພັນຄວນມີຊື່ລະບຸໄວ້ເພື່ອໃຊ້ອະທິບາຍເຊິ່ງປົກກະຕິຈະໃຊ້ສັນຍາລັກຮູບດອກຈັນທີ່ໝາຍໃນລະບຸຄຳກິລິຍາໄວ້ເພື່ອອະທິບາຍຄວາມສຳພັນ.



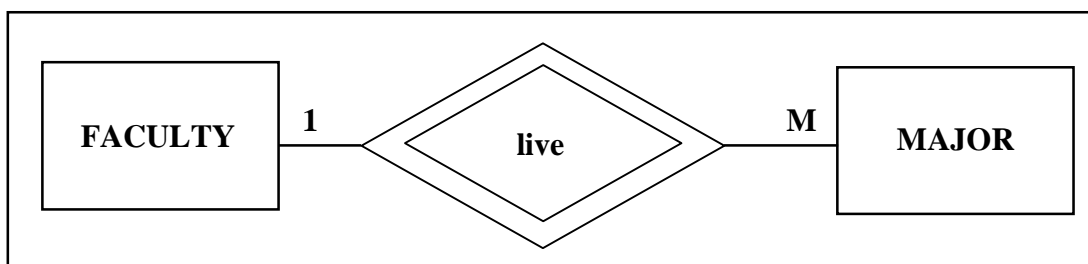
ຮູບທີ 5 ຮູບ Attribute

1) ຄວາມສຳພັນແບບຢູນາຣີ (Unary Relationships) : ເປັນຄວາມສຳພັນລະຫວ່າງເອັນຕິຕີດຽວໂດຍຄວາມສຳພັນແບບຢູນາຣີນີ້ກໍ່ຄືຄວາມສຳພັນແບບຣີເຄີຊິບ (Recursive) ທີ່ເອີ້ນໃຊ້ໃນຕົວນັ້ນເອງ.



ຮູບທີ 6 ຮູບ Attribute

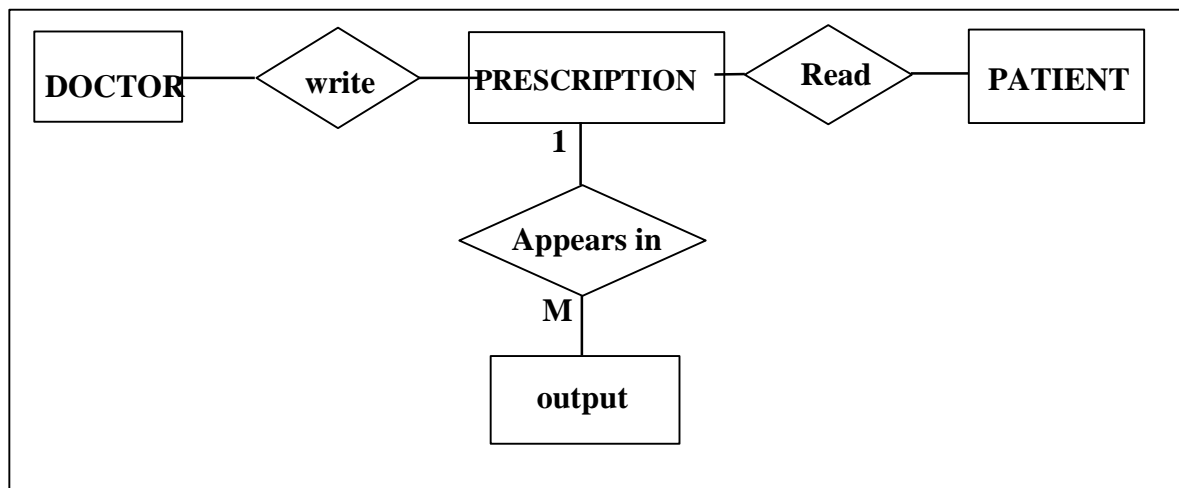
2) ຄວາມສຳພັນແບບໄບນາຣີ (Binary Relationships): ການສຳພັນຊະນິດນີ້ຈະມີເອັນຕິຕີທີ່ກ່ຽວຂ້ອງກັນ 2 ເອັນຕິຕີເຊິ່ງເປັນຄວາມສຳພັນຊະນິດໜຶ່ງທີ່ສາມາດພົບເຫັນໄດ້ຫຼາຍໃນຄວາມເປັນຈິງ.



ຮູບທີ 7 ຄວາມສຳພັນແບບໄບນາຣີ ( Binary Relationships )



3) ຄວາມສຳພັນແບບເທີນາຣີ (Ternary Relationships) ເປັນຄວາມສຳພັນທີມີກ່ຽວຂ້ອງ 3 ເອັນຕິຕີດ້ວຍກັນ.



ຮູບທີ 8 ຄວາມສຳພັນແບບເທີນາຣີ (Ternary Relationships)

#### 10.1.2.3. ພາສາ NoSql (Not Only SQL)

NoSQL ຫຼື “non SQL”, “non relational”, “not only SQL” ໝາຍເຖິງການທີ່ database ໃຊ້ວິທີການສ້າງຖານຂໍ້ມູນແລະການດຶງຂໍ້ມູນ ທີ່ແຕ່ງຕາມຈາກຮູບແບບປົກກະຕິ (SQL) ທີ່ອອກມາໃນຮູບແບບຕາຕະລາງເຊັ່ນດຽວກັບ relational databases ຊຶ່ງຕັ້ງແຕ່ຊ່ວງ 1960 ທີ່ database ໄດ້ກຳເນີດຂຶ້ນແຕ່ NoSQL ກໍຍັງບໍ່ເປັນທີ່ນິຍົມ ຈົນກະທັ້ງຊ່ວງປີ 2000 NoSQL ເລີ່ມຖືກເພີ່ມເຂົ້າມາໃນ Web2.0 ເຮັດໃຫ້ຫຼາຍບໍລິສັດຊັ້ນນຳເຊັ່ນ Facebook, Google ແລະ Amazon.com ໄດ້ເລີ່ມນຳມາໃຊ້ ສ່ວນຫລາຍຖືກນຳມາໃຊ້ງານກ່ຽວກັບ big data ແລະ real-time web applications ນັ້ນຄືທີ່ມາຂອງ NoSQL ໂດຍປົກກະຕິແລ້ວ NoSQL ງ່າຍຕໍ່ການຂະຫຍາຍລະບົບ ທີ່ເປັນຮູບແບບ cluster (ແນ່ນອນ) ຊຶ່ງນີ້ເປັນຈຸດອ່ອນຂອງ relational database ຫຼັກການຂອງ NoSQL ຄືໃຊ້ໂຄງສ້າງຂອງຂໍ້ມູນໄດ້ຫຼາຍຮູບແບບ key-value, wide column, graph ຫລື document ຊຶ່ງຈະແຕກຕ່າງຈາກຮູບແບບຂອງ relational database ຊຶ່ງບາງເທື່ອເຮັດໃຫ້ NoSQL ມີຄວາມວ່ອງໄວຫລາຍກວ່າ (ຮູ້ຈັກ NoSQL ຕ່າງຫຍັງກັບ SQL <https://saixiii.com/database-nosql/> April 26, 2017).

#### ➤ ປະເພດຂອງ NoSQL

ມີການຈຳແນກໄດ້ຫຼາຍຮູບແບບສຳລັບ NoSQL database ໂດຍແຕ່ລະຮູບແບບໄດ້ມີການຈັດກຸ່ມ ແລະກຸ່ມຍ່ອຍໄວ້ ບາງເທື່ອອາດຈະມີບາງຊະນິດກາບກ່ຽວກັນດ້ວຍເຊັ່ນ:

- Column: Accumulo, Cassandra, Druid, HBase, Vertica, SAP HANA
- Document: Apache CouchDB, ArangoDB, Clusterpoint, Couchbase, DocumentDB, HyperDex, IBM Domino, MarkLogic, MongoDB, OrientDB, Qizx, RethinkDB

- Key-value: Aerospike, ArangoDB, Couchbase, Dynamo, FairCom c-treeACE, FoundationDB, HyperDex, InfinityDB, MemcacheDB, MUMPS, Oracle NoSQL Database, OrientDB, Redis, Riak, Berkeley DB
- Graph: AllegroGraph, ArangoDB, InfiniteGraph, Apache Giraph, MarkLogic, Neo4J, OrientDB, Virtuoso, Stardog
- Multi-model: Alchemy Database, ArangoDB, CortexDB, Couchbase, FoundationDB, InfinityDB, MarkLogic, OrientDB

## ກ. Key-value store

ເຮັດວຽກໃນຮູບແບບ array (Map ຫລື Dict) ຊຶ່ງເປັນໂຄງສ້າງພື້ນຖານທີ່ສ້າງຄວາມສໍາພັນເປັນຄູ່ ຊຶ່ງ Key ຈະຕ້ອງບໍ່ຊ້ຳກັນ ຮູບແບບນີ້ຄືແບບທີ່ງ່າຍທີ່ສຸດ ແລະ ມີຄວາມສາມາດໃນການປະມວນຜົນທີ່ສູງຫລາຍ

## ຂ. Document store

ຮູບແບບຂອງ document store ຄືການ ເຂົ້າລະຫັດຂໍ້ມູນດ້ວຍຮູບແບບ format ມາດຕາຖານ ເຊັ່ນ XML, YAML, JSON ຫຼື binary ແລະເກັບໄວ້ໃນ database ດ້ວຍ unique key ທີ່ເຊື່ອມກັບຂໍ້ມູນ ຄ້າຍຮູບແບບຂອງ key-value ໂດຍ database ຈະມີ API ຫລື ພາສາທີ່ໃຊ້ query ຂໍ້ມູນສາມາດຕິດຕັ້ງໄດ້ຫຼາຍຮູບແບບ ຕາມກຸ່ມ documents

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

ຖ້າທຽບກັບ relational database ຈະເຫັນວ່າ ແຕ່ລະ record ສາມາດມີຈຳນວນ field ບໍ່ເທົ່າກັນໄດ້ ຊຶ່ງບໍ່ຄືກັບຮູບແບບຕາຕະລາງ

## ຄ. Graph

ຖືກອອກແບບສໍາລັບຂໍ້ມູນທີ່ມີຕ້ອງສະແດງເປັນ graph ທີ່ປະກອບໄປດ້ວຍຂໍ້ມູນຕົວເລກນັບບໍ່ຖ້ວນ ເຊັ່ນ Social Relations, link ຂອງການຂົນສົ່ງ, ຖະໜົນແຜນທີ່ ຫຼື ລະບົບ network

### 10.1.2.4. ລະບົບຈັດການຖານຂໍ້ມູນ (Database Management Systems: DBMS)

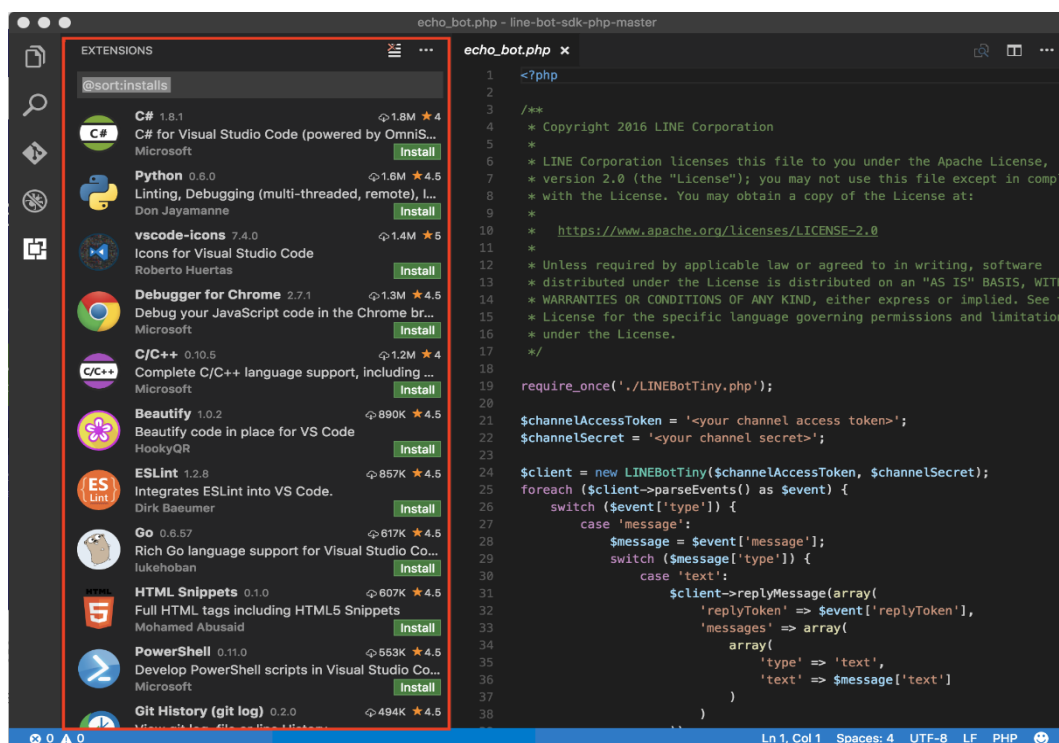
ລະບົບຈັດການຖານຂໍ້ມູນ ແມ່ນໂປຣແກຣມທີ່ໃຊ້ໄວ້ເພື່ອຕອບກັບຖານຂໍ້ມູນເຊິ່ງປະກອບດ້ວຍພັງຊັນຕ່າງໆທີ່ໃຊ້ໃນການຈັດການຖານຂໍ້ມູນ ແລະ ໃຊ້ພາສາ NoSQL (Not Only SQL) ໃນການໂຕ້ຕອບລະຫວ່າງຜູ້ໃຊ້ກັບລະບົບດ້ວຍການສ້າງ, ເອີ້ນໃຊ້ ແລະ ບໍາລຸງຮັກສາຖານຂໍ້ມູນ. ນອກຈາກນີ້, ຜູ້ໃຊ້ຍັງສາມາດໂຕ້ຕອບກັບຖານຂໍ້ມູນຜ່ານໂປຣແກຣມປະຍຸກໄດ້ເຊັ່ນ: ການສ້າງໂປຣແກຣມເພື່ອໂຕ້ຕອບກັບ DBMS ດ້ວຍພາສາ JavaScript ໂດຍນຳໃຊ້ໂປຣແກຣມ Robo 3T ເຊິ່ງເປັນເຄື່ອງມືຈັດການຖານຂໍ້ມູນ MongoDB.

### 10.1.3. ພາສາທີ່ໃຊ້ຂຽນໂປຣແກຣມ

#### 10.1.3.1. ໂປຣແກຣມ Visual Studio Code

Visual Studio Code ຫຼື ທີ່ຫຼາຍຄົນນິຍົມຮຽກຫຍັງເປັນ “vs code” ຂໍບອກກ່ອນວ່າ Editor ໂຕນີ້ມັນອອກມາຕັ້ງແຕ່ 29 ເມສາ ປີ 2015 ແລ້ວ ພັດທະນາຂຶ້ນໂດຍບໍລິສັດຍັກໃຫຍ່ນາມ ໄມໂຄຣຊອບ(Microsoft) ເປັນທັງໂຕແກ້ໄຂ ແລະ ປັບແຕ່ງໂຄດ (code optimized editor) ທີ່ຕັດຄວາມສາມາດມາຈາກ Visual Studio ລຸ້ນປົກກະຕິ (ພວກ GUI designer) ອອກໄປ ເຫຼືອແຕ່ໂຕ editor ຢ່າງດຽວ ສາມາດເຮັດວຽກໄດ້ຂ້າມແພລດຟອມທັງໝົດວິນ Windows, Mac ແລະ Linux ຊັບພອດພາສາຫຼາຍຮ້ອຍພາສາອີກດ້ວຍ ຊຶ່ງທາງໄມໂຄຊອບເອງນັ້ນໄດ້ເປີດໃຫ້ໃຊ້ຟຣີອີກດ້ວຍ.

ຄວາມສາມາດຂອງ“vs code” ນັ້ນຈະມີຄວາມສາມາດໃນການເປີດໄດ້ຄືກັບ editor ໂຕອື່ນໆ ເຊັ່ນ sublime, Atom, Notepad++ ນັ້ນເອງ ທັງໝົດເຖິງຄວາມສາມາດໃນການຕິດຕັ້ງເຄື່ອງມືເສີມ (Extension) ໂດຍຮັບ ຮອງໄດ້ວ່າມີຊັບພອດຢ່າງແນ່ນອນເພາະວ່າມັນຖືກພັດທະນາມາໃຫ້ຕອບໂຈດນັກພັດທະນາຫຼາຍທີ່ສຸດ ແມ່ນການດຶງຊາຍໜ້າຕາ ໃຫ້ເປັນຮູບແບບທີ່ເຂົ້າໃຈ ແລະ ໃຊ້ງານໄດ້ງ່າຍ ບໍ່ຕ້ອງສຶກສາຫຍັງເພີ່ມເຕີມກໍໃຊ້ງານໄດ້ເລີຍ ສິ່ງທີ່ເຮັດໃຫ້ມັນໂດນເດັ່ນກວ່າໂຕອື່ນໆ ຄືການທີ່ອອກແບບໃຫ້ການຄົ້ນຫາສິ່ງຕ່າງໆ ເຮັດອອກມາໃຫ້ໃຊ້ງານໄດ້ງ່າຍແລະເບິ່ງງ່າຍກວ່າໂຕອື່ນໆ ທັງໝົດເຖິງການທີ່ສ້າງໃຫ້ສາມາດເຊື່ອມຕໍ່ກັບ Git ໄດ້ຢ່າງວ່ອງໄວ ແລະ ງ່າຍດາຍ ມີພັງຊັນໃນການ commit, push & pull ຢູ່ໃນໂຕ ຫຼື ຈະເບິ່ງ change ຂອງຟາຍທີ່ເກີດຂຶ້ນກໍໄດ້ແບບງ່າຍດາຍບໍ່ຕ້ອງສຶກສາຫຍັງເພີ່ມແຕ່ກໍໃຊ້ງານໄດ້ເລີຍ (ມາຮູ້ຈັກ “VS Code” ອາວຸດຄູ່ມືສາຍພັດທະນາຂອງໃຜຫຼາຍໆຄົນ April 3, 2017).



ຮູບທີ 9 ໂປຣແກຣມ VSCode

### 10.1.3.2. ພາສາ JavaScript

ພາສາ JavaScript ຫລືຫຍໍ້ JS ເປັນພາສາຂຽນໂປຣແກຣມທີ່ຖືກພັດທະນາແລະປະຕິບັດຕາມຂໍ້ກຳນົດມາດຕະຖານຂອງ ECMAScript; ພາສາ JavaScript ນັ້ນເປັນພາສາລະດັບສູງ ຄອມພາຍໃນຂະນະທີ່ໂປຣແກຣມຮັ່ງ(JIT) ແລະ ເປັນພາສາຂຽນໂປຣແກຣມແບບຫຼາຍຂະບວນເຊັ່ນ: ການຂຽນໂປຣແກຣມແບບຂັ້ນຕອນ, ການຂຽນໂປຣແກຣມແບບວັດຖຸ, ພາສາ JavaScript ມີໄວຍະກອນທີ່ຄືກັບພາສາ C ໃຊ້ວົງເລັບເພື່ອກຳນົດບ່ອກຂອງຄຳສັ່ງ ນອກຈາກນີ້ JavaScript ຍັງເປັນພາສາທີ່ມີປະເພດຂໍ້ມູນແບບໄດນາມິກ (Dynamic) ເປັນພາສາແບບ Prototype-based ແລະ First-class function.

ພາສາ JavaScript ນັ້ນຖືວ່າເປັນເທັກໂນໂລຊີຫຼັກຂອງການພັດທະນາເວັບໄຊ (World Wide Web) ມັນເຮັດໃຫ້ໜ້າເວັບສາມາດຕອບໂຕ້ກັບຜູ້ໃຊ້ໄດ້ໂດຍທີ່ບໍ່ຈຳເປັນຕ້ອງລິເຟດໜ້າໃໝ່ (Dynamic website) ເວັບໄຊຈຳນວນຫຼາຍໃຊ້ພາສາ JavaScript ສຳລັບຄວບຄຸມການເຮັດວຽກງານທີ່ດ້ານ Client-side ນັ້ນເຮັດໃຫ້ເວັບເບຣາວເຊີຕ່າງໆ ມີ JavaScript engine ທີ່ໃຊ້ສຳລັບປະມວນຜົນສະຄົບຂອງພາສາ JavaScript ທີ່ຮັ້ນເທິງເວັບບາວເຊີເນື່ອງຈາກພາສາ JavaScript ເປັນພາສາຂຽນໂປຣແກຣມແບບຫຼາຍຮູບແບບ ເຮັດໃຫ້ມັນຮອງຮັບການຂຽນໂປຣແກຣມທັງແບບ Event-driven, Functional ແລະ ແບບລຳດັບຂັ້ນຕອນ ມັນມີ Library (APIs) ສຳລັບເຮັດວຽກກັບຂໍ້ຄວາມ, ວັນທີ, Regular expression ແລະ ໂຄງສ້າງຂໍ້ມູນພື້ນຖານຢ່າງ Array ແລະ Map ຫຼື ແມ່ນທັງ Document Object Model (DOM) ຊຶ່ງເປັນ API ທີ່ໂດຍທົ່ວໄປແລ້ວສາມາດໄດ້ເທິງເວັບບາວເຊີ.

ຢ່າງໃດກໍຕາມ ຕົວຂອງພາສາ JavaScript ເອງບໍ່ໄດ້ມີພັງຊັນສຳລັບອິນພຸດ/ເອົາພຸດ(I/O) ທີ່ມາກັບພາສາເຊັ່ນ ພັງຊັນກ່ຽວກັບ Network ວຽກກ່ຽວກັບໄຟລ ຫຼື Library ກ່ຽວກັບກຣາຟິກ ໂດຍທົ່ວໄປແລ້ວສິ່ງເຫຼົ່ານີ້ຈະຖືກໃຫ້ມາໂດຍ Host environment (ສະພາບແວດລ້ອມທີ່ໃຊ້ຮັ້ນພາສາ JavaScript) ເຊັ່ນ ເວັບເວັບບາວເຊີ ຫຼື Node.js ຊຶ່ງຈະແຕກຕ່າງກັນອອກໄປ ຕົວຢ່າງເຊັ່ນ ການຮັບຄຳໃນເວັບເວັບບາວເຊີຈະຜ່ານພັງຊັນ prompt ຊຶ່ງເປັນສ່ວນໜຶ່ງຂອງ Browser Object Model (BOM) ຫລືຮັບຄຳຈາກ HTML ຟອມຊຶ່ງເປັນສ່ວນໜຶ່ງຂອງ Document Object Model (DOM) ໃນຂະນະທີ່ເທິງ Node.js ເຮົາສາມາດຮັບຄຳໄດ້ຈາກ Input/Output Stream ຂອງ Command line ເຖິງແມ່ນວ່າມັນຈະມີຄວາມຄ້າຍຄືກັນລະຫວ່າງພາສາ Java ແລະ JavaScript ເຊັ່ນ: ຊື່ຂອງພາສາໄວຍະກອນຫລື Library ມາດຕະຖານຕ່າງໆ ຢ່າງໃດກໍຕາມ ທັງສອງພາສາແຕກຕ່າງກັນຢ່າງສິ້ນເຊີງໃນແງ່ຂອງການອອກແບບ ພາສາ Java ເປັນພາສາທີ່ມີປະເພດຂໍ້ມູນແບບຄົງທີ່ (Static-typing) ໃນຂະນະທີ່ພາສາ JavaScript ມີປະເພດຂໍ້ມູນແບບໄດນາມິກ (Dynamic-typing) ພາສາ Java ຖືກຄອມພາຍເປັນ Byte-code ກ່ອນທີ່ຈະລັນ ໃນຂະນະທີ່ພາສາ JavaScript ຈະຄອມພາຍໃນຕອນທີ່ໂປຣແກຣມລັນພາສາ Java ເປັນພາສາແບບ Class-based ໃນຂະນະທີ່ພາສາ JavaScript ເປັນພາສາແບບ Prototypebased



## ຮູບທີ 10 ພາສາ JavaScript

### ກ. ປະຫວັດຄວາມເປັນມາຂອງພາສາ JavaScript

ພາສາ JavaScript ຖືກອອກແບບແລະສ້າງໂດຍ Brendan Eich ສໍາລັບເປັນພາສາສະຄິບທີ່ເຮັດວຽກເທິງເວັບເບຣາວເຊີ Navigator ທີ່ເປັນຜະລິດຕະພັນຂອງບໍລິສັດ Netscape ເພື່ອເຮັດໃຫ້ໜ້າເວັບທີ່ໃນຕອນທໍາອິດນັ້ນເປັນແບບ Static ສາມາດຕອບໂຕ້ກັບຜູ້ໃຊ້ໄດ້ໂດຍທີ່ບໍ່ຈໍາເປັນຕ້ອງຮີເຟດໜ້າໃໝ່ (Dynamic) ເຊັ່ນ ການສົ່ງຂໍ້ມູນເບື້ອງໜຶ່ງໄປຍັງ Server ແລະລໍຖ້າຮັບຜົນຕອບກັບມາດ້ວຍ AJAX; ພາສາ JavaScript ໄດ້ຖືກເປີດຕົວ ແລະ ເປັນສ່ວນໜຶ່ງຂອງເວັບບາວເຊີ Navigator ໃນເດືອນກັນຍາ 1995 ໂດຍໃຊ້ຊື່ວ່າ LiveScript ແລະ ໄດ້ປ່ຽນເປັນ JavaScript ໃນອີກສາມເດືອນຕໍ່ມາ.

ໃນເດືອນພະຈິກ 1996 Netscape ໄດ້ລົງພາສາ JavaScript ໄປຍັງ ECMA International ເພື່ອເປັນຈຸດເລີ່ມຕົ້ນສໍາລັບກໍານົດມາດຕະຖານໃຫ້ທຸກເວັບບາວເຊີປະຕິບັດຕາມມາດຕະຖານດັ່ງກ່າວ ເພື່ອໃຫ້ການພັດທະນາ JavaScript engine ເປັນໄປໃນທິດທາງດຽວກັນ ນັ້ນໃຫ້ເກີດການເປີດຕົວຢ່າງເປັນທາງການສໍາລັບຂໍ້ກໍານົດມາດຕະຖານ ECMAScript ໃນເດືອນມິຖຸນາ 1997 ໃນຊ່ວງເວລາຫຼັງຈາກນີ້ ບໍລິສັດຕ່າງໆ ທີ່ພັດທະນາເວັບບາວເຊີຕ່າງກໍຍັງພັດທະນາ JavaScript engine ບໍ່ເປັນໄປໃນທິດທາງດຽວກັນເທົ່າໃດ ນັ້ນເຮັດໃຫ້ນັກພັດທະນາເວັບຕ້ອງຂຽນໂຄດຫຼາຍເວີຊັນເພື່ອໃຫ້ເຮັດວຽກໄດ້ໃນທຸກເວັບບາວເຊີຈົນກະທັ້ງໃນເດືອນກໍລະກົດ 2008 ໄດ້ມີການຈັດການປະຊຸມຂຶ້ນທີ່ Oslo ຈາກອົງກອນແລະຝ່າຍຕ່າງໆ ທີ່ພັດທະນາ JavaScript engine ເຮັດໃຫ້ເກີດຂໍ້ຕົກລົງຂຶ້ນໃນຕົ້ນປີ 2009 ເພື່ອລວບລວມງານທີ່ກ່ຽວຂ້ອງທັງໝົດຂອງພາສາ JavaScript ແລະ ຊຸກຍູ້ພາສາໃຫ້ຢ່າງໄປຂ້າງໜ້າ ນັ້ນເຮັດໃຫ້ເກີດຂໍ້ກໍານົດມາດຕະຖານ ECMAScript ເວີຊັນທີ 5 (ES5) ອອກມາໃນເດືອນທັນວາ 2009 ແລະ ກ່ອນໜ້ານີ້ໃນປີ 2008 Google ໄດ້ເປີດຕົວເວັບບາວເຊີ Chrome ທີ່ມາພ້ອມກັບ V8 JavaScript engine ທີ່ມີແນວຄິດໃນການພັດທະນາແບບຄອມພາຍໃນຕອນທີ່ໂປຣແກຣມລັນ (Just-in-time compilation: JIT) ຊຶ່ງມັນເຮັດວຽກໄດ້ໄວກວ່າຫຼາຍ ເຮັດໃຫ້ຜູ້ພັດທະນາເວັບບາວເຊີອື່ນໆ ຕ້ອງປັບປຸງ JavaScript engine ຂອງພວກເຂົາໃຫ້ເຮັດວຽກໃນຮູບແບບ JIT ຫຼັງຈາກທີ່ພັດທະນາຕໍ່ເນື່ອງມາອີກຫຼາຍປີ ໃນປີ 2015 ໄດ້ມີການເພີ່ມຄຸນສົມບັດໃໝ່ໆ ທີ່ຫຼາກຫຼາຍເຂົ້າມາ ຊຶ່ງຖືວ່າເປັນ

ການປ່ຽນແປງເທື່ອສໍາຄັນ ແລະເຮັດໃຫ້ເກີດຂໍ້ກໍານົດມາດຕະຖານ ECMAScript 2015 ຫຼື ເວີຊັນທີ 6 (ES6) ຈົນກະທັ້ງໃນປີ 2015 ຕອນນີ້ເບິ່ງຄືວ່າພາສາ JavaScript ຈະພັດທະນາມາຈົນເຖິງທີ່ສຸດແລ້ວ ເຮັດໃຫ້ລະຫວ່າງປີ 2016 - 2019 ເວີຊັນໃໝ່ຂອງ ECMAScript ທີ່ຖືກເຜີຍແຜ່ອອກມາໃນແຕ່ລະປີມີການປ່ຽນແປງແລະເພີ່ມຄຸນສົມບັດພຽງນ້ອຍໆເທົ່ານັ້ນ.

## ຂ. ຄຸນສົມບັດຂອງພາສາ JavaScript

ECMAScript 2015 (ES6) ເປັນພາສາ JavaScript ທີ່ຖືວ່າພັດທະນາມາຈົນເຖິງຈຸດສູງສຸດແລ້ວກໍວ່າໄດ້ ມັນຖືກເຜີຍແຜ່ໃນເດືອນມິຖຸນາ 2015 ຊຶ່ງໃນເວີຊັນນີ້ ໄດ້ເພີ່ມໄວຍະກອນໃໝ່ຂອງພາສາຫຼວງຫລາຍເຊັ່ນ ການສ້າງຄາດດ້ວຍຄໍາສັ່ງ class ການສ້າງໂມດູນແລະໃຊ້ງານມັນດ້ວຍຄໍາສັ່ງ import ແລະ export ແລະ ຄໍາສັ່ງສໍາລັບປະກາດຕົວປ່ຽນ let ແລະ ປະກາດຄ່າຄົງທີ່ const ຊຶ່ງເຮັດໃຫ້ຕົວປ່ຽນສາມາດມີຂອບເຂດໃນບລັອກທີ່ມັນຖືກສ້າງຂຶ້ນໄດ້ ແລະສິ່ງອື່ນໆ ທີ່ຖືກເພີ່ມເຂົ້າມາເປັນຈຳນວນຫລາຍ ເຊັ່ນ Map, Set, WeakMap, Promise, Reflection, Proxies, Template string ແລະອື່ນໆ.

ໃນເດືອນມິຖຸນາ 2016 ໄດ້ມີການເປີດຕົວເວີຊັນ 7 ຫລື ECMAScript 2016 (ES7) ໄດ້ມີການເພີ່ມຕົວດໍາເນີນການຍົກກຳລັງ \*\* (ທີ່ກ່ອນໜ້ານີ້ເຮົາຈະໃຊ້ຜ່ານພັງຊັນ Math.pow) ຄໍາສັ່ງ await async ສໍາລັບການຂຽນໂປຣແກຣມທີ່ເຮັດວຽກບໍ່ພ້ອມກັນ ແລະ ພັງຊັນ includes ຂອງອາເລ ແລະ ໃນປະຈຸບັນ ພາສາ JavaScript ຖືກພັດທະນາມາຈົນເຖິງ ECMAScript 2020 (ES11) ຊຶ່ງມີການປ່ຽນແປງທີ່ເພີ່ມຂຶ້ນບໍ່ເທົ່າໃດຫຼັງຈາກ ES7

## ຄ. JavaScript engine ແມ່ນຫຍັງ?

JavaScript engine ຄືໂປຣແກຣມຄອມພິວເຕີທີ່ໃຊ້ສໍາລັບປະມວນຜົນໂຄດຂອງພາສາ JavaScript ຊຶ່ງ JavaScript engine ໃນຊ່ວງເລີ່ມຕົ້ນເປັນພຽງແຕ່ຕົວປ່ຽນພາສາ (Interpreter) ເທົ່ານັ້ນ ແຕ່ໃນປະຈຸບັນໄດ້ມີການພັດທະນາມາໃຫ້ຢູ່ໃນຮູບແບບຂອງຄອມພາຍເລີທີ່ມີການຄອມພາຍໃນຕອນທີ່ໂປຣແກຣມລັນ (Just-in-time compilation: JIT) ເພື່ອເພີ່ມປະສິດທິພາບການເຮັດວຽກງານຂອງໂປຣແກຣມ ໂດຍທົ່ວໄປແລ້ວ JavaScript engine ຈະຖືກພັດທະນາໂດຍຜູ້ພັດທະນາເວັບບາວເຊີທີ່ປະຕິບັດຕາມຂໍ້ກໍານົດມາດຕະຖານຂອງ ECMAScript ([ແນະນຳພາສາ JavaScript](http://marcuscode.com/lang/javascript/introducing-to-javascript) <http://marcuscode.com/lang/javascript/introducing-to-javascript> 28 July 2020).

### 10.1.3.3. Reactjs

React ເປັນເທັກໂນໂລຢີໜຶ່ງທີ່ມາແຮງຫຼາຍໆ ທັງໃນໄທ ແລະ ຕ່າງປະເທດ ໂດຍສ້າງມາຈາກພື້ນຖານແນວຄວາມຄິດແບບ MVC (Model View Controller) ຊຶ່ງໝາຍຄວາມວ່າ React ມີໜ້າທີ່ຈັດການກັບ Model ຫລື View ແຕ່ສ່ວນໃຫຍ່ຈະເປັນ View ກ່ອນໜ້ານັ້ນເວລາຈະຂຽນໜ້າເວັບ ເຮົາກໍຈະເຮັດຜ່ານ HTML ມີການໃຊ້ CSS ໃນການປັບປ່ຽນໜ້າຕາຂອງ UI ແຕ່ໃນ React ຖ້າຈະສ້າງໜ້າເວັບຂຶ້ນມາໄດ້ນັ້ນ ເຮົາຈະໃຊ້ເປັນ Component ປຸງໄດ້ວ່າ Component ເປັນ Block ສ່ວນຍ່ອຍຂອງເວັບເຮົາທີ່ຈະສ້າງອອກມາ ໂດຍເຮົາຈະຂຽນໂຄດພາສາ JavaScript ເພື່ອອະທິບາຍໜ້າຕາ ຂອງ

block ນັ້ນໆມາ ເວລາສະແດງໜ້າ Page ເຮົາຈະເອີ້ນ Component ອອກມາສະແດງຜົນໜ້າຕາຂອງ ໂຄດດັ່ງຕົວຢ່າງນີ້

```
ReactDOM.render(  
  <h1>Hello, world!</h1>,  
  document.getElementById('root')  
)
```



## ຮູບທີ 11 React.js

### ➤ ສະຫລຸບງ່າຍໆ ມີດັ່ງນີ້

- React ເປັນ Javascript Library ສ້າງແລະພັດທະນາຂຶ້ນໂດຍ Facebook ຂຽນໄດ້ແຕ່ UI ເທົ່ານັ້ນ ແລະ ເປີດໃຫ້ໃຊ້ຟຣີ
- React ໃຊ້ໂຄດ HTML, CSS ແລະ Javascript
- React ມີ 3 ຄອນເຊບທີ່ເຮົາຕ້ອງຮຽນຮູ້ ຄື Component, State ແລະ Props

### ➤ ຂໍ້ດີແລະຂໍ້ເສຍຂອງ React

#### ກ. ຂໍ້ດີ

- Component ເຂົ້າໃຈງ່າຍ ເຮົາສາມາດຮຽນຮູ້ໄດ້ດ້ວຍຕົວເອງ
- Tool ຫຼາຍ ພຽງ React ຢ່າງດຽວກໍ່ສາມາດຂຽນເວັບໄດ້ທັງເວັບແລ້ວ ໂດຍບໍ່ຕ້ອງຫາ Tool ເພີ່ມເຕີມ ແລະ ຍັງມີ Tool ພັດທະນາອອກມາຢູ່ເລື້ອຍໆສາມາດເຮັດ App ໄດ້ React ມີເຄື່ອງມືໜຶ່ງທີ່ຊື່ວ່າ React Native ເປັນການຂຽນ JavaScript ແລ້ວແປງເປັນ App ແບບ Native ໄດ້ທັງເທິງ Android ແລະ iOS

#### ຂ. ຂໍ້ເສຍ

- ຕ້ອງມີພື້ນຖານໃນ Javascript ໃນລະດັບໜຶ່ງ ຄົນທີ່ສົນໃຈຮຽນກໍຈະລຳບາກນ້ອຍໜຶ່ງຖ້າບໍ່ໄດ້ມີພື້ນຖານ Javascript ອາດຈະຕ້ອງໃຊ້ເວລານ້ອຍໜຶ່ງ



- Documentation ອ່ານຍາກ React ມີ Documentation ທີ່ຍັງບໍ່ຄ່ອຍດີ ໃນອະນາຄົດອາດຈະມີ ການພັດທະນາອີກ (ມາທຳຄວາມຮູ້ຈັກ React <https://www.mindphp.com/ບາດເລີຍນອດນໂລ/ tutorial-reactjs/4883-react.html> 13 May 2017).

#### 10.1.3.4. Nodejs

Node.js ເປັນ open-source ແລະ cross-platform JavaScript runtime environment ທີ່ກຳລັງໄດ້ຮັບຄວາມນິຍົມສູງ ໂດຍທົ່ວໄປເຮົາຈະໃຊ້ JavaScript ໃນຝັ່ງ client ແຕ່ Node.js ເຮັດໃຫ້ເຮົາໃຊ້ JavaScript ໃນຝັ່ງ Server ໄດ້ດ້ວຍ ຊຶ່ງ Node.js ສາມາດ run ໄດ້ເທິງ platform ທີ່ຫຼາກຫຼາຍ ທັງ Windows, Linux, Unix, Mac OS X ແລະ ອື່ນໆ.



#### ຮູບທີ 12 Nodejs

ຈຸດເດັ່ນທີ່ສຸດຂອງ Node.js ຄືມັນເຮັດວຽກແບບ asynchronous ຜູ້ອ່ານອາດຈະສົງໃສວ່າ ແລ້ວມັນດີແນວໃດ? ລອງມາເບິ່ງຕົວຢ່າງການຈັດການໄຟລເມື່ອມີການຮ້ອງຂໍຈາກ client ມາທີ່ server ຂອງ PHP ກັບ Node.js ທຽບກັນເບິ່ງເພື່ອໃຫ້ເຂົ້າໃຈຫລາຍຂຶ້ນ

➤ ກໍລະນີ PHP ເມື່ອມີການຮ້ອງຂໍເຂົ້າມາມັນຈະເຮັດດັ່ງນີ້:

1. ສົ່ງ task ໄປທີ່ລະບົບໄຟຂອງຄອມພິວເຕີ
2. ລໍຖ້າຈົນກະທັ້ງລະບົບໄຟລເປີດແລະອ່ານໄຟລສຳເລັດ
3. ສົ່ງເນື້ອຫາຂອງໄຟລກັບມາໃຫ້ client
4. ພ້ອມສຳລັບຮັບຄຳຮ້ອງຂໍຖັດໄປ

➤ ກໍລະນີ Node.js ເມື່ອມີການຮ້ອງຂໍເຂົ້າມາມັນຈະເຮັດດັ່ງນີ້:

1. ສົ່ງ task ໄປທີ່ລະບົບໄຟຂອງຄອມພິວເຕີ
2. ພ້ອມສຳລັບຮັບຄຳຮ້ອງຂໍຖັດໄປ
3. ພໍລະບົບໄຟລເປີດແລະອ່ານໄຟລສຳເລັດ server ຈະສົ່ງເນື້ອຫາຂອງໄຟລກັບມາໃຫ້ client

ຈາກຕົວຢ່າງດ້ານເທິງຈະເຫັນວ່າ Node.js ຈະຕັດຂັ້ນຕອນການລໍຖ້າຖ້າແລ້ວໄປເຮັດຄຳຮ້ອງຖັດໄປເລຍ ທີ່ເປັນແບບນີ້ເພາະ Node.js ຈະ run ແບບ single-threaded ແລະໃນ Library



ມາດຕະຖານກໍຈະມີເຊັດຂອງ asynchronous I/O primitives ທີ່ຊ່ວຍປ້ອງກັນໂຄດ JavaScript ຈາກການ blocking ເຮັດໃຫ້ລະບົບຄ່ອງແຄ້ວ ແລະ ມີປະສິດທິພາບຫຼາຍຂຶ້ນ

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

## ➤ ປະຫວັດ Node.js

ແຕ່ເດີມພາສາ JavaScript ຖືກພັດທະນາມາເພື່ອໃຊ້ສຳລັບ Browser ທີ່ຊື່ Netscape Navigator ໃນປີ 1995 ໃນຕອນນັ້ນ Netscape ຕັ້ງໃຈຈະຂາຍ Web Server ທີ່ມີ environment ຊື່ Netscape LiveWire ຊຶ່ງສາມາດສ້າງ dynamic page ໂດຍໃຊ້ JavaScript ທາງຝັ່ງ server ດ້ວຍແຕ່ໜ້າເສຍດາຍທີ່ Netscape LiveWire ບໍ່ປະສົບຄວາມສຳເລັດ ແລະ ການໃຊ້ JavaScript ທາງຝັ່ງ server ກໍບໍ່ໄດ້ຮັບຄວາມນິຍົມເລີຍຈົນກະທັ້ງ Node.js ຖືກກຳເນີດຂຶ້ນມາ.

ສິ່ງທີ່ເຮັດໃຫ້ Node.js ເປັນທີ່ນິຍົມຂຶ້ນມາຄືການທີ່ມັນມາໃນຊ່ວງເວລາທີ່ເໝາະສົມ ເມື່ອທຽບກັບ JavaScript ທີ່ເກີດມາຕັ້ງແຕ່ປີ 1995 ແລ້ວ Node.js ຫາກໍເກີດມາເມື່ອປີ 2009 ເທົ່ານັ້ນ ຕ້ອງ

ຂອບໃຈ "Web 2.0" applications (ເຊັ່ນ Flickr, Gmail ແລະ ອື່ນ ໆທີ່ສະແດງໃຫ້ໂລ (ກຮູ້ວ່າເວັບສະໄໝໃໝ່ຄວນໜ້າຕາເປັນຢ່າງໃດ

ບໍ່ດົນກ່ອນທີ່ Node.js ຈະເກີດ ນັກພັດທະນາເບື້ອງໜຶ່ງ browser ຊຶ່ງທັງຫຼາຍແຂ່ງຂັນກັນເຮັດວຽກຢ່າງໜັກເພື່ອຈະໃຊ້ JavaScript ໃຫ້ໄດ້ດີທີ່ສຸດແລະຫາທາງເຮັດໃຫ້ JavaScript ສາມາດລັນໄດ້ໄວຫຼາຍຂຶ້ນ ເພື່ອໃຫ້ຜູ້ໃຊ້ງານໄດ້ຮັບປະສິດທິພາບທີ່ດີທີ່ສຸດ ຊຶ່ງຜົນຈາກການແຂ່ງຂັນນີ້ເຮັດໃຫ້ເກີດການພັດທະນາ Chrome V8 (open-source JavaScript engine ຂອງ The Chromium Project) ຂຶ້ນມາ ແລະ Node.js ກໍໃຊ້ engine ນີ້ເອງ

- ແຕ່ການທີ່ Node.js ເປັນທີ່ນິຍົມຂຶ້ນມາບໍ່ແມ່ນແຕ່ວ່າມັນມາຖືກທີ່ຖືກເວລາເທົ່ານັ້ນ ແຕ່ເພາະມັນໄດ້ສະແດງໃຫ້ເຫັນແລ້ວວ່າ ການອອກແບບແລະແນວຄິດຂອງມັນຊ່ວຍນັກພັດທະນາທັງຫຼາຍໃຫ້ສາມາດໃຊ້ JavaScript ທາງຝັ່ງ server ໄດ້ງ່າຍຂຶ້ນຫຼາຍອີກດ້ວຍ. (Nodejs ແມ່ນຫຍັງ [https://expert-programming-tutor.com/tutorial/article/A06\\_NodeJS01\\_HelloWorld.php\\_13\\_May\\_2017](https://expert-programming-tutor.com/tutorial/article/A06_NodeJS01_HelloWorld.php_13_May_2017)).

#### 10.1.3.5. ExpressJS

Express.js ເປັນ Web Application Framework ຊຶ່ງທີ່ໄດ້ຮັບຄວາມນິຍົມຫຼາຍ ສໍາລັບເຮັດວຽກເທິງ platform ຂອງ Node.js ຊຶ່ງເປັນ Server ຕົວໜຶ່ງໂດຍທັງ Express.js ແລະ Node.js ຕ່າງກໍໃຊ້ພາສາ JavaScript ໃນການພັດທະນາ ຖ້າເປັນ Web Application Framework ໃນສະໄໝກ່ອນຄົນທີ່ພັດທະນາຈະຕ້ອງມີຄວາມຮູ້ຫຼາຍກວ່າ 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 ກໍຍົກເອົາມາເປັນຂໍ້ເດັ່ນໃນເລື່ອງຄວາມໄວ ໃນເລື່ອງການການຮຽນຮູ້ການຂຽນ Express.js ຈະໃຊ້ຮູບແບບທີ່ງ່າຍໃນການຮຽນຮູ້ຫຼາຍທີ່ສຸດສໍາລັບການ ພັດທະນາ Express.js ໃນເວັບໄຊຈະເວົ້າເຖິງການໃຊ້ Routing (ການກໍານົດເສັ້ນທາງຂອງລະບົບ) ແລະ Middleware (ການຮັບສົ່ງຂໍ້ມູນຂອງລະບົບ) ສາມາດຂຽນໄດ້ໃນຮູບແບບ MVC ສ່ວນການເຊື່ອມຕໍ່ກັບຖານຂໍ້ມູນສາມາດໃຊ້ MongoDB ຫລືຈະໃຊ້ MySql ກໍໄດ້ສໍາລັບນາມສະກຸນຂອງໄຟລຕີ .js ຂະນະນີ້ໄດ້ພັດທະນາມາເຖິງເວີຊັນທີ່ 4.x ແລ້ວ. (express.js ແມ່ນຫຍັງ <https://www.mindphp.com/73-คืออะไร/3874-what-is-express-js.html> 8 May 2016).

express

### ຮູບທີ 13 express.js

ວັນທີ ...../...../.....

...../...../.....

ລາຍເຊັນຄະນະກຳມະການ

ລາຍເຊັນນັກສຶກສາ

ວັນທີ ...../...../.....

ລາຍເຊັນອາຈານທີ່ປຶກສາ

ວັນທີ