

ສາທາລະນະ​ລັດ ປະຊາທິປະ​ໄຕ ປະຊາຊົນ​ລາວ

ສັນຕິພາບ ​ເອກະລາດ ປະຊາທິປະ​ໄຕ ​ເອກະ​ພາບ ວັດທະນາ​ຖາວອນ

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

**​ ຄະນະວິທະຍາສາດທຳມະຊາດ**

**ພາກວິຊາວິທະຍາສາດຄອມພິວເຕີ**

**ບົດສະເໜີໂຄງການຈົບຊັ້ນ ລະດັບຕໍ່ເນື່ອງປະລິນຍາຕີວິທະຍາສາດ**

**ສາຂາ ວິທະຍາສາດຄອມພິວເຕີ**

**ຊື່​ບົດຈົບຊັ້ນ​ (Title)**

ພາສາ​ລາວ: ລະບົບຄຸ້ມຄອງການຮຽນ-ການສອນຂອງໂຮງຮຽນພອນສະຫວັນ ແລະ

ຊົນເຜົ່າກຽມມະຫາວິທະຍາໄລ

ພາສາ​ອັງກິດ: School for Gifted and Ethnic Students Learning and Teaching Management System

**ລາຍຊື່ນັກສຶກສາຂຽນບົດ (Project Team)**

|  |  |  |  |
| --- | --- | --- | --- |
| ລ/ດ | ລະຫັດ | ຊື່ ແລະ ນາມສະກຸນ | ເບີໂທ |
| 1 | 204Q000219 | ນາງ ມີໄຊ ສີສຸລິດ | 020 54787068 |
| 2 | 204N003219 | ນາງ ທິບປະສົງ ແສງທຳມະວົງ | 020 77306019 |

**ອາຈານທີ່ປຶກສາ (Advisor)**

|  |  |  |
| --- | --- | --- |
| ລ/ດ | ຊື່ ແລະ ນາມສະກຸນ | ຕໍາແໜ່ງ |
| 1 |  | ນໍາພາ |
| 2 |  | ຊ່ວຍນໍາພາ |

1. **ຄວາມສຳຄັນຂອງບັນຫາ**

ທຸກວັນນີ້, ເທັກໂນໂລຊີຄອມພີວເຕີ້ຖືວ່າມີບົດບາດສໍາຄັນຫຼາຍຕໍ່ວິຖີການດໍາລົງຊີວິດຂອງຄົນເຮົາ ແລະ ຕໍ່ກັບທຸກຂົງເຂດວຽກງານ ເພື່ອຍົກລະດັບຄຸນນະພາບຊີວິດການເປັນຢູ່ຂອງມວນມະນຸດ ໃຫ້ສະດວກຍີ່ງຂື້ນ ແລະ ທັງຊ່ວຍເພີ່ມປະສິດທິພາບຊີວິດການເປັນຢູ່ຂອງມວນມະນຸດໃຫ້ສະດວກຍີ່ງຂື້ນ ແລະ ທັງຊ່ວຍເພີ່ມປະສິດທີພາບໃນການເຮັດວຽກຕ່າງໆຂອງແຕ່ລະຂະແໜງການ ໂດຍນໍາໃຊ້ເຄື່ອງຄອມພິວເຕີ້ທົດແທນແຮງງານຄົນ. ໃນບັນດາປະເທດທີ່ພັດທະນາແລ້ວເຊັ່ນວ່າ: ປະເທດຍີ່ປຸ່ນ,ສາຫະລັດອາເມລິກາ, ເກົ່າຫຼີ ແລະ ປະເທດອື່ນໆໃນໂລກ ກໍ່ໄດ້ປາຍຸກໃຊ້ໃນວຽກງານການສຶກສາ ເຮັດໃຫ້ສາມາດເພີ່ມຄຸນນະພາບການສຶກສາຂອງນັກຮຽນໃນປະເທດເຫຼົ່ານັ້ນສູງພໍສົມຄວນ ໃນນັ້ນ, ເທັກໂນໂລຊີຄອມພິວເຕີ້ກໍໄດ້ຖືກນຳເຂົ້າມາໃຊ້ໃນວຽກງານການສືກສາຂອງປະເທດລາວເຮົາໄດ້ໃນລະດັບໃດໜື່ງ, ແຕ່ກໍຍັງບໍ່ທັນຕອບສະໜອງຕາມຄວາມຕ້ອງການຂອງວຽກງານການສຶກສາໃນ ສປປ ລາວໄດ້ເທົ່າທີ່ຄວນ.

ໂຮງຮຽນພອນສະຫວັນ ແລະ ຊົ່ນເຜົ່າກຽມມະຫາວິທະຍາໄລ, ມີຊື່ຫຍໍ້ວ່າ ຮ ພ ຊ, ເປັນໂຮງຮຽນທີ່ຂື້ນກັບສຳນັກງານອະທິການບໍດີ ພາຍໃຕ້ການຊີ້ນຳຂອງທ່ານ ຮອງອະທິການບໍດີ ມະຫາວິທະຍາໄລແຫ່ງຊາດ,ເຊີ່ງມີພາລະບົດບາດໃນການສ້າງຊັບພະຍາກອນມະນຸດກໍຄືການສ້າງນັກຮຽນເກັ່ງ, ໂຮງຮຽນແຫ່ງນີ້ໄດ້ຮັບການຊ່ວຍເຫຼືອລ້າຈາກລັດຖະບານ ປະເທດ ສສ ຫວຽດນາມ ລວມມູນຄ່າ 47 ຫວຽກນາມຕື້ດົ້ງ ແລະ ຖືກນຳໃຊ້ຢ່າງເປັນທາງການແຕ່ວັນທີ່ 07 ກຸມພາ 2007 ເປັນຕົ້ນມາເຊີ່ງປະກອບດ້ວຍ 07 ອາຄານຄື: ອາຄານອຳນວຍການ,ອາຄານຮຽນ,ອາຄານຫ້ອງທົດລອງ-ຫໍສະໝຸດ,ສະໂມສອນ,ໂຮງອາຫານ ແລະ ຫໍພັກ ສອງຫຼັງ, ສຳລັບຊັ້ນຮຽນແມ່ນມີແຕ່ຂັ້ນ ມ1 ເຖິງຂັ້ນ ມ7. ລະບົບການຈັດຕັ້ງປະກອບມີ ຄະນະອຳນວຍການ 04 ທ່ານ ປະກອບມີໜ່ວຍງານ, ມີຄູ-ອາຈານທັງໝົດ 56 ທ່ານ ຍິງ 29 ທ່ານ ຍິງ 29 ທ່ານ, ຊັ້ນສູງ 01 ທ່ານ ຍິງ 0, ມີນ້ອງນັກຮຽນທັງໝົດ 479 ນ້ອງ ຍິງ 282 ນ້ອງ ແລະ ໄດ້ມີຄູຈາກ ສສ ຫວຽດນາມ ມີ 01 ທ່ານ ທີ່ມາຊ່ວຍສອນ ວິຊາ ພາສາຫວຽດນາມ.ໂຮງຮຽນພອນສະຫວັນ ແລະ ຊົນເຜົ່າກຽມມະຫາວິທະຍາໄລ ເປັນອົງ​ກອນໜຶ່ງທີ່ມີ​ລັກສະນະ​ເອກະລາດ​ແຫ່ງ​ດຽວ​ໃນ​ປະ​ເທດ ທີ່​ຢູ່ໃຕ້​ການຊີ້​ນຳ ຂອງ​ມະຫາວິທະຍາ​ໄລ​ແຫ່ງ​ຊາດ, ​ເຊິ່ງມີຖານະທຽບ​ເທົ່າ​ກັບພາກວິຊາ, ​ສູນ​ການ​ສຶກສາ ຫຼື ບັນດາຫ້ອງ​ການຕ່າງໆຂອງມະຫາວິທະຍາ​ໄລ​ແຫ່ງ​ຊາດ. ລະບົບຄຸ້ມຄອງການຮຽນ-ການສອນຂອງໂຮງຮຽນພອນ​ສະຫວັນ ມະຫາວິທະຍາໄລແຫ່ງຊາດ ມີຈຸດປະສົງເພື່ອປະຍຸກໃຊ້ຄວາມສາມາດທາງດ້ານເທັກໂນໂລຊີຄອມພິວເຕີ້ມາຊ່ວຍເພີ່ມປະສິດທິພາບໃນການຈັດເກັບຂໍ້ມູນ, ໂດຍຈະຈັດເກັບຂໍ້ມູນຕ່າງໆໃນຮູບແບບຂໍ້ມູນດີຈີຕໍ, ຊ່ວຍການເກັບກໍາຂໍ້ມູນນັກຮຽນ ແລະ ຄູ, ຊ່ວຍບັນທຶກການຂື້ນຫ້ອງ-ລົງຫ້ອງຂອງນັກຮຽນ, ຄູອາຈານ ແລະ ຍັງສາມາດໃຫ້ພໍ່ແມ່ຜູ້ປົກຄອງເດັກສາມາດເບີ່ງຄະແນນຂອງລູກໄດ້, ພ້ອມທັງຊ່ວຍໃຫ້ຂະບວນການລົງທະບຽນງ່າຍ ແລະ ໄວຂື້ນກ່ວາເກົ່າ. ນອກນັ້ນຍັງສາມາດຈັດເກັບຄະແນນຂອງນັກຮຽນ ແລະ ປະເມີນຄູໄດ້ອີກດ້ວຍ. ໃນເວລາດຽວກັນ, ມັນກໍ່ຍັງສາມາດຄົ້ນຫາຂໍ້ມູນຕ່າງໆ ໄດ້ໄວຍີ່ງຂື້ນ ແລະ ລາຍງານຂໍ້ມູນຕ່າງໆໃຫ້ແກ່ຄະນະອຳນວຍການໄດ້ສະດວກວ່ອງໄວຂື້ນ.ເນື່ອງຈາກວ່າ ໂຮງຮຽນພອນ​ສະຫວັນ ຍັງບໍທັນໄດ້ນໍາໃຊ້ໂປແກມທີ່ທັນສະໄໝເຂົ້າມາຊ່ວຍດຳເນີນວຽກງານຂອງເຂົາເຈົ້າ ເຊີ່ງການເກັບກຳຂໍ້ມູນ ແລະ ເຮັດບົດລາຍງານຕໍ່ຄະນະອຳນວຍການແມ່ນຍັງບັນທຶກເປັນແຟ່ມເອກະສານ ແລະມີບາງຄົນກໍບັນທຶກເປັນ word/Excel ມາຊ່ວຍວຽກບັນທຶກຂໍ້ມູນບາງສ່ວນ. ດັ່ງນັ້ນຈື່ງເຮັດໃຫ້ການຈັດເກັບຂໍ້ມູນຕ່າງໆບໍເປັນລະບຽບ, ການບັນທຶກຄະແນນ ແລະ ປະເມີນຜົນການຮຽນຂອງນັກຮຽນຕ້ອງໃຊ້ເວລາດົນ, ການຄົ້ນຫາຂໍ້ມູນຍັງຊ້າບໍທັນໃຈ

ດັ່ງນັ້ນ ພວກຂ້າພະເຈົ້າຈື່ງເກີດມີແນວຄິດຢາກພັດທະນາລະບົບຄຸ້ມຄອງການຮຽນ-ການສອນຂອງໂຮງຮຽນພອນ​ສະຫວັນ ມະຫາວິທະຍາໄລແຫ່ງຊາດ ນີ້ຂື້ນເພື່ອຊ່ວຍແກ້ໄຂບັນຫາຕ່າງໆ ທີ່ໄດ້ກ່າວມາຂ້າງເທີງນັ້ນ. ອີກດ້ານໜື່ງ, ກໍ່ແມ່ນການພັດທະນາລະບົບຂື້ນມາເພື່ອແບ່ງເບົາພາລະ ແລະ ຫຼຸດຜ່ອນເວລາການເຮັດວຽກຂອງພະນັກງານທີ່ກ່ຽວຂ້ອງກັບລະບົບດັ່ງກ່າວ.

1. **ຈຸດປະສົງຂອງການຄົ້ນຄ້ວາ (Objectives)**

* ເພຶ່ອສຶກສາບັນຫາກ່ຽວກັບການຈັດການກັບຂໍ້ມູນຕ່າງໆຂອງໂຮງຮຽນພອນ​ສະຫວັນ ມະຫາວິທະຍາໄລແຫ່ງຊາດ ການເກັບກຳຂໍ້ມູນພື້ນຖານ, ຂະບວນການລົງທະບຽນ, ການຕິດຕາມການຂື້ນຫ້ອງຂອງນັກຮຽນ ແລະ ຄູອາຈານ, ການຈັດເກັບຄະແນນ, ການຈັດອັນດັບທີ່ ແລະ ການປະເມີນຜົນການຮຽນຂອງນັກຮຽນ ແລະ ຄູອາຈານ, ສາມາດຄົ້ນຫາຂໍ້ມູນ ແລະ ລາຍງານຂໍ້ມູນຕ່າງໆໄດ້ໄວກ່ວາເກົ່າ.ແລະພ້ອມກັນນັ້ນຜູ້ປົກຄອງສາມາດຕິດຕາມຄະແນນຂອງລູກໄດ້
* ເພື່ອສ້າງ Web Application ລະບົບຄຸ້ມຄອງການຮຽນ-ການສອນຂອງໂຮງຮຽນ ພອນ​ສະຫວັນ ມະຫາວິທະຍາໄລແຫ່ງຊາດ.
* ເພື່ອແກ້ໄຂບັນຫາ ກ່ຽວກັບການຈັດການຂໍ້ມູນຕ່າງໆທີ່ບໍ່ເປັນລະບົບ, ການເກັບກໍາຂໍ້ມູນພະນັກງານຄູ, ການລົງທະບຽນຊັກຊ້າ ການລາຍງານຂໍ້ມູນທີ່ບໍ່ທັນວ່ອງໄວເທົ່າທີ່ຄວນ ແລະ ບັນຫາການເຮັດໃບຢັ້ງຢືນນັກຮຽນ ແລະ ໃບຄະແນນສະບັບພາສາອັງກິດແມ່ນກຳລັງພັດທະນາ.

1. **ຂອບເຂດການຄົ້ນຄ້ວາ (Scope)**

ລະບົບຄຸ້ມຄອງການຮຽນ-ການສອນຂອງ ໂຮງຮຽນພອນສະຫວັນ ແລະ ຊົນເຜົ່າກຽມມະຫາວິທະຍາໄລແມ່ນລະບົບ Client Server Web Application ເຊີ່ງປະກອບດ້ວຍຫຼາຍໜ້າວຽກຫຼັກດັ່ງລຸ່ມນີ້:

* ຈັດການຂໍ້ມູນພື້ນຖານ ( ຂໍ້ມູນນັກຮຽນ, ຂໍ້ມູນຄູອາຈານ, ຂໍ້ມູນຊັ້ນຮຽນ, ຂໍ້ມູນຫ້ອງຮຽນ, ຂໍ້ມູນວິຊາຮຽນ, ຂໍ້ມູນອາຈານສອນ ແລະ ຂໍ້ມູນຜູ້ໃຊ້).
* ລົງທະບຽນ ( ກວດສອບ ແລະ ລົງທະບຽນ).
* ຈັດການຂໍ້ມູນຄະແນນ( ບັນທຶກຄະແນນ ແລະ ຄົ້ນຫາຄະແນນ).
* ປະເມີນຜົນ ( ຕິດຕາມນັກຮຽນ, ຈັດອັນດັບທີ່, ເລື່ອນຊັ້ນ).
* ລາຍງານ ( ຊີວະປະຫວັດນັກຮຽນ, ສະຖິຕິນັກຮຽນ, ວັນຂາດນັກຮຽນ, ໃບຢັ້ງຢືນນັກຮຽນສະບັບລາວ-ອັງກິດ, ໃບຄະແນນສະບັບລາວອັງກິດ).

1. **ປະໂຫຍດຄາດວ່າຈະໄດ້ (Expected Outcome of the Project)**

ໃນການສຶກສາຄົ້ນຄວ້າຄັ້ງນີ້, ລະບົບຄຸ້ມຄອງການຮຽນ-ການສອນຂອງໂຮງຮຽນພອນ​ສະຫວັນ ມະຫາວິທະຍາໄລແຫ່ງຊາດໄດ້:

* ຊ່ວຍໃຫ້ການຈັດເກັບຂໍ້ມູນເປັນລະບົບລະບຽບເທີ່ງເຄື່ອງຄອມພິວເຕີ້.
* ຊ່ວຍໃຫ້ການເກັບກຳພະນັກງານຄູເປັນລະບົບດີ ແລະ ສະດວກວ່ອງໄວຂື້ນ.
* ຊ່ວຍໃຫ້ການລົງທະບຽນຮຽນໃຫ້ນັກຮຽນໄວຂື້ນ.
* ຊ່ວຍໃຫ້ການຈັດເກັບຂໍ້ມູນຄະແນນ, ການຈັດອັນດັບທີ, ປະເມີນຜົນການຮຽນ ແລະ ປະເມີນຄູໃຊ້ເວລາ
* ຊ່ວຍໃຫ້ພໍ່ແມ່ປົກຄອງສາມາດເບີ່ງຄະແນນລູກໄດ້.

1. **ວິທີດຳເນີນການຄົ້ນຄ້ວາ (Research Methodology)**

ຂັ້ນຕອນວິທີການດໍາເນີນການພັດທະນາລະບົບໃນຄັ້ງນີ້ແມ່ນອີງໃສ່ລັກສະນແບບຈໍາລອງຂອງ Adapted Waterfall Model ເປັນແນວທາງໃນການວິເຄາະ ແລະ ອອແບບລະບົບ ເຊິ່ງປະກອບດ້ວຍໜ້າວຽກດັ່ງນີ້:

ວາງແຜນ

ວິເຄາະ

ອອກແບບ

ພັດທະນາ

ທົດສອບ

ສ້າງເອກະສານ

ແຜນວາດທີ່ 1: ຂັ້ນຕອນວິທີດຳເນີນການຂອງໂຄງການ

* **ວາງແຜນ.**

ດໍາເນີນການຈັດຕັ້ງກຸ່ມຂຽນບົດໂຄງການພຽງພ້ອມກັບການກໍານົດຫົວຂໍ້ຂອງໂຄງການຈາກນັ້ນ, ກໍ່ໄດ້ລົງເກັບກໍາຂໍ້ມູນຢູ່ ໂຮງຮຽນພອນສະຫວັນ ແລະ ຊົນເຜົ່າກຽມມະຫາວິທະຍາໄລ ເພື່ອໃຫ້ໄດ້ຂໍ້ມູນ ແລະ ຂັ້ນຕອນການເຮັດວຽກໂດບລວມກ່ຽວກັບການເຮັດວຽກຕ່າງໆຂອງໂຮງຮຽນດັ່ງກ່າວ.

* **ວິເຄາະ.**

ໄລຍະນີ້ພວກເຮົາຈະສຶກສາລະບົບເກົ່າ ແລະ ຄວາມຕ້ອງການຂອງຜູ້ໃຊ້ຈາກນັ້ນ, ກໍ່ນໍາເອົາຂໍ້ມູນຕ່າງໆທີ່ລວບລວມໄດ້ມາເພື່ອວິເຄາະເປັນຂໍ້ກຳນົດຄວາມຕ້ອງການຂອງລະບົບໃໝ່ພ້ອມທັງແຕ້ມແບບຈຳລອງ DFD ເພື່ອສະແດງເຖິງການໄຫຼຂອງຂໍ້ມູນໄປເຖິງຂະບວນການ ແລະ ແຜນວາດ E-R Diagram ເພື່ອສະແດງຄວາມສຳພັນລະຫວ່າງຂໍ້ມູນ.

* **ອອກແບບ.**

ໄລຍະນີ້ ພວກເຮົາຈະອອກແບບໂປຣແກຣມ, ຕ້ອງແບບໜ້າສະແດງຜົນ, ອອກແບບໜ້າປ້ອນຂໍ້ມູນ, ອອກແບບຖານຂໍ້ມູນໂດຍເຮັດ Normalization ແລະ ສ້າງ Data Dictionary.

* **ພັດທະນາ.**

ດໍາເນີນການສ້າງຖານຂໍ້ມູນຕາມທີ່ໄດ້ເຮັດ Data Dictionary ໃນຂັ້ນຕອນກ່ອນໜ້ານັ້ນຈາກນັ້ນການລົງມືປະຕິບັດຂຽນ Code ເພື່ອຕິດຕໍ່ ແລະ ເຮັດວຽກກັບຖານຂໍ້ມູນທີ່ໄດ້ສ້າງໄວ້ແລ້ວ.

* **ທົດສອບ.**

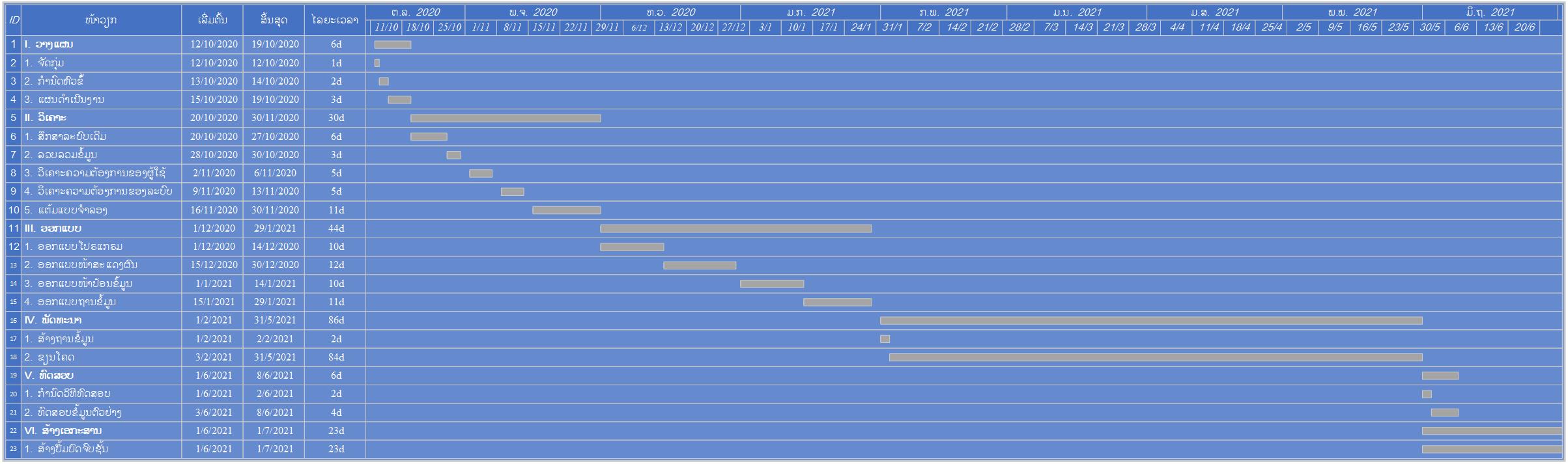
ສ້າງແບບວິທີການທົດສອບລະບົບ ແລະ ນໍາເອົາຊຸດຂໍ້ມູນທີ່ນໍາມາທົດສອບຢ່າງໜ້ອຍ 20 records ມາດໍາເນີນການທົດສອບ ແລ້ວສະຫຼຸບໄດ້ຜົນຮັບ ແລະ ເຮັດການປັບປຸງແກ້ໄຂຈົນໄດ້ຜົນອອກທີ່ຖືກຕ້ອງ.

* **ສ້າງເອກະສານ.**

ສ້າງປຶ້ມບົດຈົບຊັ້ນ ລະບົບຄຸ້ມຄອງການຮຽນ-ການສອນຂອງໂຮງຮຽນພອນສະຫວັນ ແລະ ຊົນເຜົ່າກຽມມະຫາວິທະຍາໄລ ແລ້ວສົ່ງມອບປຶ້ມເຮັດແຜ່ນ CD ທີ່ປະກອບດ້ວຍ Source Code, ເອກະສານບົດຈົບຊັ້ນ ແລະ Presentations.

1. **ສະຖານທີ່ສຶກສາ (Study Site)**

ໂຮງຮຽນພອນສະຫວັນ ແລະ ຊົນເຜົ່າກຽມມະຫາວິທະຍາໄລ

1. **​ໄລຍະປະຕິບັດ (Duration)**
2. **ເຄື່ອງມື​​ທີ່ນຳໃຊ້​ໃນ​ການ​ພັດທະນາ (Development Tools)**

|  |  |
| --- | --- |
| 1. Hardware | 1. Software |
| * ຄອມພິວເຕີ Asus Intel Core(TM) I3-7020U @2.3GHz, RAM 8GB 2133 MHz * USB 16GB 1 ອັນ | * ລະບົບປະຕິບັດການ Windows 10 Professional 64 Bit * Microsoft Visio 2016 ໃຊ້ແຕ້ມແຜນວາດການໄຫຼຂໍ້ມູນ (DFD, ER, Flowchart) * Robo 3T ໃຊ້ຈັດການຖານຂໍ້ມູນ * Visual Studio Code ໃຊ້ຂຽນໂຄດດ້ວຍພາສາ JavaScript * MS Office 2013 Professional ໃຊ້ເພື່ອສ້າງເອກະ ສານຕ່າງໆ, ບົດນຳສະເໜີ ແລະ ເຮັດປຶ້ມບົດຈົບຊັ້ນ |

1. **ເອກະສານອ້າງອີງ (Reference)**

**Style: APA**

1. **ທົບທວນທິດສະດີ ແລະ ບົດທີ່ກ່ຽວຂ້ອງ** 
   1. **ທົບທວນທິດສະດີທີ່ກ່ຽວຂ້ອງ**

ລະບົບຈອງປີ້ລົດເມສາຍໃຕ້ອອນລາຍແມ່ນພັດທະນາຂຶ້ນໂດຍນຳໃຊ້ທິດສະດີ ຫຼື ຄວາມຮູ້ຈາກ 3 ສ່ວນຄື: ທິດສະດີໃນການວິເຄາະ ແລະ ອອກແບບລະບົບ, ທິດສະດີກ່ຽວກັບການສ້າງຖານຂໍ້ມູນ ແລະ ຄວາມຮູ້ກ່ຽວກັບພາສາໃນການພັດທະນາລະບົບ. ລາຍລະອຽດຂອງແຕ່ລະທິດສະດີແມ່ນຈະໄດ້ນຳສະເໜີໂດຍສັງເຂບດັ່ງຕໍ່ໄປນີ້:

* + 1. **ທິດສະດີກ່ຽວກັບການວິເຄາະ ແລະ ການອອກແບບລະບົບ**

ການ​ວິ​ເຄາະ​ລະ​ບົບ​ເປັນ​ຂະ​ບວນ​ການ​ທຳ​ຄວາມ​ເຂົ້າ​ໃຈ ແລະ ກຳ​ນົດ​ລາຍ​ລະ​ອຽດ​ຂອງ​ບັນ​ຫາ​ເພື່ອ​ຈະ​ໄດ້​ພິ​ຈາ​ລະ​ນາ​ນຳ​ເອົາ​ລະ​ບົບ​ຂໍ້​ມູນ​ຂ່າວ​ສານ​ໃດ​ເຂົ້າ​ໄປ​ແກ້​ບັນ​ຫາ​ເລົ່າ​ນັ້ນ, ສ່ວນ​ການ​ອອກ​ແບບ​ລະ​ບົບ​ໝາຍ​ເຖິງ​ຂະ​ບວນ​ການ​ກຳ​ນົດ​ລາຍ​ລະ​ອຽດ​ຕ່າງໆ​ວ່າ​ຈະ​ຕ້ອງ​ເຮັດ​ແນວ​ໃດ​ກັບ​ອົງ​ປະ​ກອບ​ຂອງ​ລະ​ບົບ​ຂໍ້​ມູນ​ຂ່າວ​ສານ​ເພື່ອ​ຈະ​ໄດ້​ນຳ​ໄປ​ໃຊ້​ໃຫ້​ເກີດ​ຜົນ​ໃນ​ທາງ​ພາຍນອກ​ໄດ້. ສິ່ງ​ທີ່​ນຳ​ສະ​ເໜີ​ໃນ​ຫົວ​ຂໍ້​ນີ້​ໄດ້​ຂັດ​ສັນ​ພາກ​ສ່ວນ​ໜຶ່ງ​ຈາກ​ (ພ​ຣະ ເຫຼົາ​ຄຳ ເພັດ​ວິ​ໄລ, ວິ​ລະ​ຍຸດ ວົງ​ທິ​ລາດ ແລະ ເພັດ​ດາວວອນ ທິ​ແກ້ວ, 2016) ແລະ (ສົມ​ມິດ ທຸມ​ມາ​ລີ ແລະ ອາ​ມອນ ຈັນ​ທະ​ພາ​ວົງ​, 2012).

* + - 1. **ວົງຈອນພັດທະນາລະບົບ**
      2. **ແຜນວາດການໄຫຼຂໍ້ມູນ (Data Flow Diagram DFD)**

- ເປັນແຜນພາບທີ່ສະຫຼຸບລວມຂໍ້ມູນທັງໝົດໄດ້ຈາກການວິເຄາະໃນລັກສະນະຂອງຮູບແບບທີ່ເປັນໂຄງສ້າງ.

- ເປັນຂໍ້ຕົກລົງຮ່ວມກັນລະຫວ່າງນັກວິເຄາະລະບົບ ແລະ ຜູ້ຊົມໃຊ້.

- ເປັນແຜນພາບທີ່ໃຊ້ໃນການພັດທະນາຕໍ່ໃນຂັ້ນຕອນຂອງການອອກແບບ.

- ຮູ້ທີ່ໄປທີ່ມາຂອງຂໍ້ມູນທີ່ໄຫຼໃນຂະບວນການຕ່າງໆ.

* + - 1. **ສັນຍະລັກທີ່ໃຊ້ໃນແຜນວາດຂໍ້ມູນ**

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 |

* + - 1. **ຫຼັກການຂຽນແຜນວາດການໄຫຼຂໍ້ມູນ**

1. Process:
   * ເມື່ອມີຂໍ້ມູນເຂົ້າໄປທີ່ Process ກໍ່ຕ້ອງມີຂໍ້ມູນ ຫຼື ຜົນຮັບອອກມາຈາກ Process ເຊັ່ນກັນຈະເປັນໄປບໍ່ໄດ້ທີ່ມີສະເພາະຂໍ້ມູນເຂົາຢ່າງດຽວ.
2. Data store:

* ຂໍ້ມູນຈະໄຫຼຈາກ Data store ໜຶ່ງໄປຫາ Data store ໜຶ່ງໂດຍກົງບໍ່ໄດ້ຈະຕ້ອງຜ່ານ Process ເທົ່ານັ້ນ
* ຂໍ້ມູນທີ່ສົ່ງຜ່ານ External entityບໍ່ສາມາດໄຫຼເຂົ້າໄປ Data storeໂດຍກົງໄດ້ຈະຕ້ອງໃຊ້ Processເປັນຕົວກາງໃນການເຊື່ອມໂຍງເພື່ອຈັດເກັບຂໍ້ມູນໃນ Data store
* ຂໍ້ມູນທີ່ໄຫຼຜ່ານຈາກ Data storeບໍ່ສາມາດເຊື່ອໂຍງເຂົ້າກັບ External entity ໄດ້ໂດຍກົງຈະຕ້ອງຜ່ານ Processເທົ່ານັ້ນ.

1. External entity:

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

1. Data flow:

* ການໄຫຼຂໍ້ມູນທີ່ມີຫົວຊີ້ໄປທີ່ Processໝາຍເຖິງ Processມີການອ່ານ ຫຼື ການດືງຂໍ້ມູນຈາກ Data store ມາໃຊ້ວຽກ
* ການໄຫຼຂໍ້ມູນຈາກ Process ທີ່ມີຫົວລູກສອນຊີ້ໄປຍັງ Data store ໝາຍເຖິງການ Update ຫຼື ການເພີ່ມຂໍ້ມູນລົງໄປທີ່ Data store

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

* + - 1. **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)

ສັນຍະລັກເຫຼົ່ານີ້ເມື່ອຖືກນຳມາເຊື່ອມຕໍ່ກັນ ຈະກາຍເປັນ "ແຜນຜັງ (Flowchart)" ທີ່ສະແດງລຳດັບຂັ້ນຕອນການເຮັດວຽກງານເພື່ອ

* ເປັນ​ເຄື່ອງ​ມື​ໃນ​ການ​ຈັດ​ລຳ​ດັບ​ຄວາມ​ຄິດ
* ເຫັນ​ລຳ​ດັບ​ຂັ້ນ​ຕອນ​ການ​ເຮັດ​ວຽກງານ​ທີ່​ຊັດເຈນ

**ສັນຍະລັກ Flowchart**

|  |  |
| --- | --- |
| **ຮູບ​ພາບ​ສັນຍະລັກ** | **ຄວາມຫມາຍ​ຂອງ​ສັນຍະລັກ** |
|  | ການ​ເລີ່ມ​ຕົ້ນ​ຫລື​ຈົບ Flowchart (Start ຫລື End) |
|  | ການ​ກະທຳ (Process) ຖືກ​ໃຊ້​ເພື່ອ​ສະແດງ​ທີ່​ການ​ກະທຳ​ໃນ Flowchart |
|  | ຮັບຂໍ້ມູນ |
|  | ການ​ຕັດ​ສິນ​ໃຈ (Decision) |
|  | ສະແດງຜົນທາງຈໍພາບ |
|  | ຈຸດ​ເຊື່ອມ​ຕໍ່ (Connector) |
|  | ທິດ​ທາງ​ການ​ເຮັດ​ວຽກງານ (Direction Flow) |

* + 1. **ທິດສະດີກ່ຽວກັບຖານຂໍ້ມູນ**

Database ຫຼື ຖານຂໍ້ມູນຄືກຸ່ມຂອງຂໍ້ມູນທີ່ຖືກເກັບລວບລວມໄວ້ໂດຍມີຄວາມສໍາພັນເຊິ່ງກັນ ແລະ ກັນໂດຍບໍ່ໄດ້ບັງຄັບວ່າຂໍ້ມູນທັງຫມົດນີ້ຈະຕ້ອງເກັບໄວ້ໃນແຟ້ມຂໍ້ມູນດຽວກັນ ຫຼື ແຍກເກັບຫຼາຍໆແຟ້ມຂໍ້ມູນ

ລະບົບຖານຂໍ້ມູນຄືລະບົບທີ່ລວບລວມຂໍ້ມູນຕ່າງໆທີ່ກ່ຽວຂ້ອງກັນເຂົ້າໄວ້ດ້ວຍກັນຢ່າງມີລະບົບ, ມີຄວາມສໍາພັນລະຫວ່າງຂໍ້ມູນຕ່າງໆທີ່ຊັດເຈັນໃນລະບົບຖານຂໍ້ມູນຈະປະກອບດ້ວຍແຟ້ມຂໍ້ມູນຫຼາຍແພ້ມທີ່ມີຂໍ້ມູນກ່ຽວຂ້ອງກັນ,ສໍາພັນກັນເຂົ້າໄວ້ດ້ວຍກັນຢ່າງເປັນລະບົບ ແລະ ເປີດໂອກາດໃຫ້ ຜູ້ໃຊ້ສາມາດໃຊ້ງານ ແລະ ຮັກສາປ້ອງກັນຂໍ້ມູນເຫຼົ່ານີ້ໄດ້ຢ່າງມີປະສິດທິພາບໂດຍມີຊອບແວຣ໌ທີ່ປຽບສະເຫມືອນຊື່ກາງລະຫວ່າງຜູ້ໃຊ້ ແລະໂປຣແກຣມຕ່າງໆທີ່ກ່ຽວຂ້ອງກັບການໃຊ້ຖານຂໍ້ມູນເອີ້ນວ່າລະບົບຈັດການຖານຂໍ້ມູນ ຫຼື DBMS

* + - 1. **ການເຮັດ Normalization**
* Normalization ເປັນຫຼັກການໜື່ງທີ່ຜູ້ອອກແບບຖານຂໍ້ມູນຈະຕ້ອງນຳມາໃຊ້ໃນການແປງຂໍ້ມູນທີ່ຢູ່ໃນຮູບແບບທີ່ຊ້ຳຊ້ອນ ໃຫ້ຢູ່ໃນຮູບແບບທີ່ງ່າຍຕໍ່ການນຳໄປໃຊ້ງານ ແລະ ກໍ່ໃຫ້ເກີດບັນຫານ້ອຍທີ່ສຸດ.
* ໃຮບົດນີ້ຈະສອນຂັ້ນຕອນການເຮັດ Normalization ໃນ 3 ລະດັບເຊັ່ນ:
* Normalization ລະດັບ1 ຫຼືເອີ້ນວ່າ 1NF
* Normalization ລະດັບ 2 ຫຼືເອີ້ນວ່າ2NF
* Normalization ລະດັບ 3 ຫຼືເອີ້ນວ່າ 3NF

ຈຸດປະສົງຂອງການເຮັດ Normalization

* ຫຼຸດຜ່ອນຄວາມຊ້ຳຊ້ອນຂອງຂໍ້ມູນ ເມື່ອຫຼຸດຄວາມຊ້ຳຊ້ອນຈະເຮັດໃຫ້ຫຼຸດເນື້ອທີ່ໃນການຈັດເກັບຂໍ້ມູນ
* ຫຼຸດບັນຫາຄວາມບໍຖືກຕ້ອງຂອງຂໍ້ມູນ ເມື່ອຂໍ້ມູນບໍເກີດຄວາມຊ້ຳຊ້ອນເຮັດໃຫ້ການປັບປຸງຂໍ້ມູນສາມາດເຮັດໄດ້ຈາກແຫຼ່ງຂໍ້ມູນພຽງບ່ອນດຽວ
* ຫຼຸດຄວາມຜິດພາດທີ່ເກີດຈາກການປັບປຸງຂໍ້ມູນ update anomalies)ເຊີ່ງປະກອບດ້ວຍ:
  + - 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 3 ຮູບStrong Entity

* 1. Weak Entity:

ເອັນຕີຕີ້ຊະນິດນີ້ຈະຂຶ້ນກັບເອັນຕີຕີ້ຊະນິດອື່ນໆບໍ່ສາມາດເກີດຂຶ້ນໄດ້ຕາມລໍາພັງ ແລະ ຈະຖືກລົບເມື່ອເອັນຕີຕີ້ຫຼັກຖືກລົບອອກ ສັນຍາລັກທີ່ໃຊ້ຄືຮູບສີ່ຫຼ່ຽມຊ້ອນກັນ.

CUSTOMER

STOCK

ຮູບທີ 4 ຮູບWeak Entity

* 1. ແອັດທຣິບີວ (Attribute)

ແອັດທີຣບີວຄືຄຸນສົມບັດຂອງສົມບັດຂອງເອັນຕີຕີ້ສັນຍາລັກຂອງແອັດທີຣບີວຈະເປັນຮູບວົງມົນແອັດທຣິບີວໃດທີ່ຖືກໃຊ້ເປັນຄີຫຼັກຈະຖືກຂີດເສັ້ນກ້ອງກໍາກັບໄວ້.

emName

Position

emNo

EMPLOYEE

ຮູບທີ 5 ຮູບAttribute

* 1. ຄວາມສໍາພັນ(Relation)

ຄວາມສໍາພັນໃນທີ່ນີ້ໝາຍເຖິງຄວາມສໍາພັນລະຫວ່າງເອັນຕີຕີ້ໂດຍແຕ່ລະຄວາມສໍາພັນຄວນມີຊື່ລະບຸໄວ້ເພື່ອໃຊ້ອະທິບາຍເຊິ່ງປົກກະຕິຈະໃຊ້ສັນຍາລັກຮູບດອກຈັນທີ່ພາຍໃນລະບຸຄໍາກິລິຍາໄວ້ເພື່ອອະທິບາຍຄວາມສໍາພັນ.

SUBJECT

teaches

LUCTURE

M

1

ຮູບທີ 6 ຮູບAttribute

1. ຄວາມສໍາພັນແບບຢູນາຣີ(Unary Relationships):ເປັນຄວາມສໍາພັນລະຫວ່າງເອັນຕີຕີ້

ດຽວໂດຍຄວາມສໍາພັນແບບຢູນາຣີນີ້ກໍ່ຄືຄວາມສໍາພັນແບບຣີເຄີຊີບ (Recursive)ທີ່ເອີ້ນໃຊ້ໃນຕົວ

ນັ້ນເອງ.

manages

EMPLOYEE

1

M

ຮູບທີ 7 ຮູບAttribute

1. ຄວາມສໍາພັນແບບໄບນາຣີ(Binary Relationships):ການສໍາພັນຊະນິດນີ້ຈະມີເອັນຕີຕີ້ທີ່ກ່ຽວຂ້ອງກັນ2ເອັນຕີຕີ້ເຊິ່ງເປັນຄວາມສໍາພັນຊະນິດຫນຶ່ງທີ່ສາມາດພົບເຫັນໄດ້ຫຼາຍໃນຄວາມເປັນຈິງ.

MAJOR

live

FACULTY

M

1

ຮູບທີ 8 ຄວາມສໍາພັນແບບໄບນາຣີ *( Binary Relationships* )

1. ຄວາມສໍາພັນແບບເທີ້ນາຣີ(Ternary Relationships) ເປັນຄວາມສໍາພັນທີມກ່ຽວຂ້ອງ 3 ເອັນຕີຕີ້ດ້ວຍກັນ.

1

M

PATIENT

DOCTOR

write

Read

PRESCRIPTION

Appears in

output

ຮູບທີ 9 ຄວາມສໍາພັນແບບເທີ້ນາຣີ (Ternary Relationships)

* + - 1. **ພາສາ NoSql (Not Only SQL)**

NoSQL ຫຼື “non SQL”, “non relational” ,”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​ຢັງ​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​ກລຸ່ມ ​ແລະ​ກລຸ່ມ​ຍ່ອຍ​ໄວ້ ບາງເທື່ອ​ອາດຈະ​ມີ​ບາງ​ຊະນິດ​ກາບກ່ຽວ​ກັນ​ດ້ວຍເຊັ່ນ:

* **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**

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 ຫລື ລະບົບ network

* + - 1. ລະບົບຈັດການຖານຂໍ້ມູນ (Database Management Systems: DBMS)
    1. ພາສາທີ່ໃຊ້ຂຽນໂປຣແກຣມ
       1. ໂປຣແກຣມ Visual Studio Code
       2. ພາສາ JavaScript
       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 ແຕ່​ໃນ React ຖ້າ​ຈະ​ສ້າງ​ໜ້າ​ເວັບ​ຂຶ້ນ​ມາ​ໄດ້​ນັ້ນ ເຮົາ​ຈະ​ໃຊ້​ເປັນ [Component](https://www.mindphp.com/%E0%B8%9A%E0%B8%97%E0%B9%80%E0%B8%A3%E0%B8%B5%E0%B8%A2%E0%B8%99%E0%B8%AD%E0%B8%AD%E0%B8%99%E0%B9%84%E0%B8%A5%E0%B8%99%E0%B9%8C/tutorial-reactjs/4864-component-and-props.html) ປຽບໄດ້​ວ່າ Component ເປັນ Block ສ່ວນ​ຍ່ອຍ​ຂອງ​ເວັບ​ເຮົາ​ທີ່​ຈະ​ສ້າງ​ອອກ​ມາ ໂດຍ​ເຮົາ​ຈະ​ຂຽນ​ໂຄດພາ ສາ [JavaScript](https://www.mindphp.com/%E0%B8%9A%E0%B8%97%E0%B9%80%E0%B8%A3%E0%B8%B5%E0%B8%A2%E0%B8%99%E0%B8%AD%E0%B8%AD%E0%B8%99%E0%B9%84%E0%B8%A5%E0%B8%99%E0%B9%8C/%E0%B8%AA%E0%B8%AD%E0%B8%99-javascript/2724-%E0%B8%9A%E0%B8%97%E0%B8%97%E0%B8%B5%E0%B9%88-1-javascript-introduction.html) เພື່​ອະທິບາຍ​ໜ້າ​ຕາ ຂອງ block ນັ້ນ​ໆ​ມາ ເວລາ​ສະແດງ​​ໜ້າ Page ເຮົາ​ຈະ​ເອີ້ນ Component ອອກ​ມາ​ສະແດງ​ຜົນໜ້າ​ຕາ​ຂອງ​ໂຄດ​ດັ່ງຕົວຢ່າງ​ນີ້

ReactDOM.render(

<h1>Hello, world!</h1>,

document.getElementById('root')

);

**ສະຫລຸບ​ງ່າຍໆ ມີ​ດັ່ງ​ນີ້**

* React ເປັນ Javascript Libraly ສ້າງ​ແລະ​ພັດທະນາ​ຂຶ້ນ​ໂດຍ Facebook ຂຽນ​ໄດ້​ແຕ່ UI ເທົ່າ​ນັ້ນ ແລະ​ເປີດ​ໃຫ້​ໃຊ້​ຟ​ຣີ
* React ໃຊ້​ໂຄດ HTML , CSS ແລະ Javascript
* React ມີ 3 ຄອນ​ເຊບ​ທີ່​ເຮົາ​ຕ້ອງ​ຮຽນ​ຮູ້ ຄື Component State ແລະ Props
* **ຂໍ້​ດີ​ແລະ​ຂໍ້​ເສຍ​ຂອງ React**

**ກ. ຂໍ້​ດີ**

* Component ເຂົ້າໃຈ​ງ່າຍ ເຮົາ​ສາມາດ​ຮຽນ​ຮູ້​ໄດ້​ດ້ວຍ​ຕົວ​ເອງ
* Tool ຫຼາຍ ພຽງReact ຢ່າງ​ດຽວ​ກໍ່ສາມາດຂຽນເວັບ​ໄດ້​ທັງ​ເວັບ​ແລ້ວ ໂດຍ​ບໍ່​ຕ້ອງ​ຫາ Tool ເພີ່ມເຕິມ ແລະ​ຍັງມີ Tool ພັດທະນາ​ອອກ​ມາ​ຢູ່​ເລື້ອຍໆສາມາດ​ເຮັດ App ໄດ້ React ມີ​ເຄື່ອງ​ມື​ໜຶ່ງ​ທີ່​ຊື່​ວ່າ React Native ເປັນ​ການ​ຂຽນ JavaScript ແລ້ວ​ແປງ​ເປັນ App ແບບ Native ໄດ້​ທັງ​ເທິງ Android ແລະ iOS

**ຂ. ຂໍ້​ເສຍ**

* ຕ້ອງ​ມີ​ພື້ນ​ຖານ​ໃນ Javascript ໃນ​ລະ​ດັບ​ໜຶ່ງ ຄົົນທີ່​ສົນ​ໃຈ​ຮຽນ​ກໍ​ຈະ​ລຳ​ບາກ​ນ້ອຍ​ນຶງ​ຖ້າ​ບໍ່​ໄດ້​ມີ​ພື້ນ​ຖານ Javascript ອາດຈະ​ຕ້ອງ​ໃຊ້​ເວລານ້ອຍໜື່ງ
* Documentation ອ່ານ​ຍາກ React ມີ Documentation ທີ່​ຍັງ​ບໍ່​​ຄ່ອຍ​ດີ ໃນ​ອະນາຄົດ​ອາດຈະ​ມີ​ການ​ພັດທະນາ​ອີກ
  + - 1. **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 ສາມາດ​ run​ໄດ້​ເທິງ platform ທີ່​ຫຼາກ​ຫຼາຍ​ທັງ Windows, Linux, Unix, Mac OS X ແລະ ອື່ນໆ.

ຈຸດ​ເດັ່ນ​ທີ່ສຸດ​ຂອງ Node.js ຄື​ມັນ​ເຮັດວຽກ​ແບບ asynchronous ຜູ້​ອ່ານ​ອາດຈະ​ສົງ​ໃສ​ວ່າ​ແລ້ວ​ມັນ​ດີ​ຢ່າງໃດ? 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​ລັນໂຄ​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​ຕັວ​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​ຕັວ​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 เພື່ອລຸດຄວາມຫຍຸ້ງຍາກທັງໝົດເຖິງເວລາໃນການຕ້ອງຮຽນຮູ້ຫຼາຍໆ 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.

ວັນທີ ……/……./……… ວັນທີ ……/……./……… ວັນທີ ……/……./………

ລາຍເຊັນຄະນະກໍາມະການ ລາຍເຊັນອາຈານທີ່ປຶກສາ ລາຍເຊັນນັກສຶກສາ