

ប្រព័ន្ធគ្រប់គ្រងទ្រពួសម្បត្តិ (Asset Management System)

1. ເຄລບໍ່ແນະ (Objective)

ប្រធានគេប់គ្រងទ្វោសម្បត្តិ ត្រូវបានបង្កើតឡើងដើម្បីគ្រប់គ្រងទ្វោសម្បត្តិនូវក្នុងរវៈ និងឱដិជលកុងស្ថាប័នមួយច្បាស់របស់ខ្លួន និងមានភាពទុកចិត្តបាន។ ប្រធាននេះគឺជាផណ្ឌកបំផើរដឹកនាំទ្វោសម្បត្តិទាំងមូល បាប់ពីការចុះបញ្ជីទ្វោសម្បត្តិ ការផ្តល់ចោរប្រែស់ការក្នុងពិនិត្យស្ថាប័នភាព ការចំណាំ និងបញ្ចប់ដោយការប្រគល់ត្រឡប់ប្រាកាលបាន (Retire)។

២ . ទិន្នន័យនៃប្រព័ន្ធ (System Overview)

ប្រព័ន្ធគ្រប់គ្រងទ្រពូសម្បតីគឺជាកម្មវិធី Desktop ដែលប្រើប្រាស់ក្នុងស្ថាបនកេដមយ។ ប្រព័ន្ធឌ្វោរាណគ្រប់គ្រងដោយអ្នកគ្រប់គ្រង (Administrator) ដែលមានភាពកិច្ចចេះបញ្ចី និងគ្រប់គ្រងទ្រពូសម្បតីដំឡើង។

ប្រព័ន្ធនេះអនុវត្តនាសម្ព័ន្ធដាច់ (Layered Architecture) និងប្រើ Design Patterns ដើម្បីបង្ការកិច្ចរាង UI, Business Logic និង Data Access ទ្វាកសលាសំ។

៣ . ទម្រង់ប្រព័ន្ធ និងមុខងារ (System Forms and Functions)

ប្រព័ន្ធឌ្ឋរាន់បង់ចែកជានម្លៃ (Forms) ផ្សេងៗ ដើម្បីគ្រប់គ្រងភាពកិច្ចនឹមួយា។

៣ . ១ ទម្រង់ទី១៖ ការចុះបញ្ជីព្យសម្រតិ (Asset Registration Form)

ទម្រង់នេះប្រើសម្រាប់គ្រប់គ្រងពីតំបនមូលដ្ឋានរបស់ត្រពុសម្រួលិក។

មុខងារ៖ - បញ្ចូលទ្រព្យសម្បត្តិកី - កែវប្រព័ន្ធមានទ្រព្យសម្បត្តិ - លុបទ្រព្យសម្បត្តិ - បង្កាញបញ្ជីទ្រព្យសម្បត្តិ

ເງິນທີ່ມີລະບົບການຈຳນວຍສູງ - ໂບຊສມາດລົບໂຕງສມຸດ (Asset ID) ມີລະບົບການຈຳນວຍສູງຄູ່ - ຕັດໝາຍສູງຂອງເງິນທີ່ມີລະບົບການຈຳນວຍສູງ

៣ . ២ ទម្រង់ទី២៖ ការផ្តល់ត្រពូសម្រេគិត (Asset Assignment Form)

ទម្រង់នេះប្រើសម្រាប់ផលិតកម្មសម្រាកីឡើបុគ្គលិក។

មុខងារ៖ - ផ្តល់ទ្រព្យសម្បតី - កែវប្រការផ្តល់ - ប្រគល់ទ្រព្យសម្បតីត្រួតព័ត៌មាន

ច្បាប់អាជីវកម្ម - ទ្រព្យសម្រួលដែលមានស្ថានភាព Available បុណ្យភាពអាជ្ញេល់បាន - ទ្រព្យសម្រួល Retired ឬ Under Repair មិនអាចផ្តល់បាន

3 . 3 ទម្រង់ទឹក៖ ស្ថានភាព និងការចែក (Asset Status and Maintenance Form)

ទម្រង់នេះប្រើសម្រាប់គ្រប់គ្រងស្ថានភាពទ្រព្យសម្រួល។

មុខងារ៖ - បញ្ជូនភាពទ្រព្យសម្រួល - កត់ត្រាដំឡើងការចែក

3 . 4 ទម្រង់ទឹក៖ ស្លែងរក និងរាយការណ៍ (Search and Reporting Form)

ទម្រង់នេះប្រើសម្រាប់ត្រួតពិនិត្យ និងស្លែងរកព័ត៌មាន។

មុខងារ៖ - ស្លែងរកតាម Asset ID - ស្លែងរកតាមលេខា - តម្លៃប្រចាំថ្ងៃសម្រួលតាមស្ថានភាព

4 . ច្បាប់អាជីវកម្ម និងក្រុមការណ៍ (Business Rules and Constraints)

- ទ្រព្យសម្រួលត្រូវចុះបញ្ជីជាមុន មុននឹងអាចផ្តល់បាន
 - ទ្រព្យសម្រួលត្រូវត្រួតពិនិត្យសម្រាប់មិនអាចផ្តល់បាន
 - ទ្រព្យសម្រួល Retired មិនអាចផ្តល់ប្រុបតាមស្ថានភាពទ្រព្យសម្រួល។
 - ទ្រព្យសម្រួលត្រូវប្រគល់ត្រូវបានប្រើប្រាស់ មុននឹងផ្តល់ទេរសភាប្រើប្រាស់
 - មិនអនុញ្ញាតឡើងបុរាណទ្រព្យសម្រួលដែលកំពុងដ្ឋានប្រើប្រាស់
-

5 . ករណីប្រើប្រាស់ (Use Case Descriptions)

ករណី៖ ចុះបញ្ជីទ្រព្យសម្រួល - អ្នកប្រើប្រាស់៖ Administrator - លក្ខខណ្ឌដំបូង៖ Asset ID មិនមានស្រាប់ - ដំណើរការ៖ បញ្ចូលព័ត៌មានទៅរក្សាទុក - លទ្ធផល៖ ទ្រព្យសម្រួលរាយការណ៍រក្សាទុកនូវ Database

ករណី៖ ផ្តល់ទ្រព្យសម្រួល - អ្នកប្រើប្រាស់៖ Administrator - លក្ខខណ្ឌដំបូង៖ ស្ថានភាព = Available - ដំណើរការ៖ ប្រើប្រាស់ប្រព័ន្ធប្រើប្រាស់បុគ្គលិក → ផ្តល់ - លទ្ធផល៖ ស្ថានភាពបូរទេ In Use

ករណី៖ ប្រគល់ទ្រព្យសម្រួលត្រូវបានប្រើប្រាស់ - អ្នកប្រើប្រាស់៖ Administrator - លក្ខខណ្ឌដំបូង៖ ទ្រព្យសម្រួលត្រូវត្រួតពិនិត្យសម្រាប់បានផ្តល់បាន - ដំណើរការ៖ ប្រើប្រាស់ការផ្តល់ → ប្រគល់ត្រូវបានប្រើប្រាស់ - លទ្ធផល៖ ស្ថានភាពត្រូវបានប្រើប្រាស់ Available

6 . Design Patterns ដែលបានប្រើប្រាស់ក្នុងប្រព័ន្ធ

ប្រព័ន្ធនេះបានអនុវត្ត Design Patterns ចំនួន 1 4 ដើម្បីធ្វើឲ្យចនាសម្ព័ន្ធប្រព័ន្ធមានភាពច្បាស់លាស់ ងាយចែក និងអាចពន្លឺការណាយ ខាងក្រោមនេះគឺជាការរៀបចំ លេខរៀងត្រីមត្រូវ (6 . 1 → 6 . 1 4) ជាមួយខាងការណ៍ក្នុង C # ។

6 . 1 Singleton Pattern – គ្រប់គ្រង Database Connection

```
public class DatabaseConnection
{
    private static DatabaseConnection instance;
    private SqlConnection connection;

    private DatabaseConnection()
    {
        connection = new SqlConnection("connection_string");
    }

    public static DatabaseConnection Instance
    {
        get
        {
            if (instance == null)
                instance = new DatabaseConnection();
            return instance;
        }
    }

    public SqlConnection GetConnection() => connection;
}
```

6 . 2 Factory Pattern – បង្កើត Asset Objects

```
public class AssetFactory
{
    public static Asset CreateAsset(string type)
    {
        return type switch
        {
            "IT" => new ITAsset(),
            "Office" => new OfficeAsset(),
            _ => new GeneralAsset()
        };
    }
}
```

6 . 3 Abstract Factory Pattern – បង្កើត UI Components

```
public interface IUIFactory
{
```

```
        Button CreateButton();
        TextBox CreateTextBox();
    }
```

6 . 4 **Builder Pattern – បង្កើត Asset ជាដំណោកកាល**

```
public class AssetBuilder
{
    private Asset asset = new Asset();
    public AssetBuilder SetName(string name)
    {
        asset.Name = name;
        return this;
    }
    public Asset Build() => asset;
}
```

6 . 5 **Prototype Pattern – Clone Asset**

```
public class Asset : ICloneable
{
    public object Clone() => this.MemberwiseClone();
}
```

6 . 6 **Strategy Pattern – ច្បាប់ផ្តើកលើស្នានភាព**

```
public interface IAssetStatusStrategy
{
    bool CanAssign();
}
```

6 . 7 **State Pattern – រដ្ឋជីវិត Asset**

```
public abstract class AssetState
{
    public abstract bool AssignAllowed();
}
```

6 . 8 Observer Pattern – Update UI ស័យប្រភី

```
public interface IObserver
{
    void Update();
}
```

6 . 9 MVC Pattern – ប៊ែន Model / View / Controller

```
public class AssetController
{
    public void AddAsset(Asset asset) { }
}
```

6 . 1 0 Repository Pattern – Data Access Layer

```
public class AssetRepository
{
    public void Add(Asset asset) { }
}
```

6 . 1 1 Adapter Pattern – តាប់ Legacy System

```
public class LegacyAssetAdapter
{
    private LegacyAsset legacy;
}
```

6 . 1 2 Facade Pattern – Simplify Operations

```
public class AssetService
{
    public void RegisterAsset(Asset asset) { }
}
```

6 . 1 3 Command Pattern – Encapsulate Actions

```
public interface ICommand
{
```

```
    void Execute();
}
```

6 . 1 4 Template Method Pattern – Workflow នេះ

```
public abstract class AssetProcess
{
    public void Process()
    {
        Validate();
        Save();
    }
    protected abstract void Validate();
    protected abstract void Save();
}
```

6 . 2 Factory Pattern (បង្កើត Asset Object)

```
public class AssetFactory
{
    public static Asset CreateAsset(string type)
    {
        switch(type)
        {
            case "IT": return new ITAsset();
            case "Office": return new OfficeAsset();
            default: return new GeneralAsset();
        }
    }
}
```

6 . 3 Strategy Pattern (ថ្វាប់ផ្តុកលើស្ថានភាព Asset)

```
public interface IAssetStatusStrategy
{
    bool CanAssign();
}

public class AvailableStrategy : IAssetStatusStrategy
{
    public bool CanAssign() => true;
```

```
}

public class RetiredStrategy : IAssetStatusStrategy
{
    public bool CanAssign() => false;
}
```

6 . 4 State Pattern (រដ្ឋជីវិត Asset)

```
public abstract class AssetState
{
    public abstract bool AssignAllowed();
}
```

6 . 5 Observer Pattern (Update UI ស្វែយប្រភើ)

```
public interface IObserver
{
    void Update();
}

public class AssetSubject
{
    private List<IObserver> observers = new();
    public void Attach(IObserver obs) => observers.Add(obs);
    public void Notify()
    {
        foreach(var obs in observers)
            obs.Update();
    }
}
```

6 . 6 MVC Pattern (ប៊ូលុន Model / View / Controller)

```
public class AssetController
{
    private AssetRepository repo;

    public void AddAsset(Asset asset)
```

```
        {
            repo.Add(asset);
        }
    }
```

6 . 7 Repository Pattern (Abstract Data Access)

```
public class AssetRepository
{
    public void Add(Asset asset)
    {
        // Database logic here
    }
}
```

6 . 8 Command Pattern (Encapsulate User Actions)

```
public interface ICommand
{
    void Execute();
}

public class AddAssetCommand : ICommand
{
    public void Execute()
    {
        // Add asset logic
    }
}
```

6 . 9 Template Method Pattern (ដំណើរការតែម្មយមនជំហានចេញ)

```
public abstract class AssetProcess
{
    public void Process()
    {
        Validate();
        Save();
        Log();
    }
}
```

```

    }

    protected abstract void Validate();
    protected abstract void Save();
    protected virtual void Log() { }
}

```

7 . ការផ្តល់នូវ Design Patterns ជាមួយគ្នា

Design Pattern នឹងមើលបានថាបានត្រូវបានប្រើប្រាស់ឡើងទៀត ដើម្បីបង្ហាញពីការអនុវត្តពិតិត្យ។

8 . តារាងសង្គម Design Patterns

តារាងនេះបង្ហាញពី Design Pattern ទាំង 14 មុខងារ និងទីតាំងប្រើប្រាស់របស់វា។

9 . រចនាសម្ព័ន្ធមូលដ្ឋានទិន្នន័យ (Database Design)

ប្រព័ន្ធប្រើប្រាស់ Database ប្រភេទ Relational ដើម្បីរក្សាទុកទិន្នន័យទ្វាយសម្រួល និងការផ្តល់នូវ។

- Assets (AssetID, Name, Category, PurchaseDate, Status)
- Assignments (AssignmentID, AssetID, EmployeeID, AssignDate, ReturnDate)

1 0 . ការគ្រប់គ្រងកំហុស និងការផ្តល់នូវ (Error Handling and Validation)

- ចិនអនុញ្ញាត Asset ID ស្មួន
- ពិនិត្យព័ត៌មានចំណាំ
- ទៅស្ថាត់ស្ថានភាពមិនត្រឹមត្រូវ
- បង្ហាញសារកំហុសដល់អ្នកប្រើប្រាស់

1 1 . ការធ្វើតែស្ថិតិវិធី (Testing and Validation)

ក្នុងការធ្វើ Demo ប្រព័ន្ធ ត្រូវបង្ហាញការធ្វើការនិកតាមទម្រង់ (Forms) នាងក្រាម។ ក្នុងកសារនេះ រួមចាប់នូវការធ្វើ Demo (Screenshot) នៅតាមទីតាំងដែលបានបញ្ជាក់។

Demo 1 : Asset Registration Form

ទីតាំងដែលបានបញ្ជាក់ បន្ទាប់ពីការពិនិត្យនាងក្រាម និងការផ្តល់នូវ។ - Screenshot បង្ហាញការបញ្ចូល Asset ថ្មី - Screenshot បង្ហាញ Asset បន្ទាប់ពីរក្សាទុក

(ដាក់រូបភាព Demo នីនេះ)

Demo 2 : Asset Assignment Form

ទីតាំងដាក់របស់ពីរ បន្ទាប់ពីការពិពណ៌នានៅថ្ងៃទី ២ - Screenshot បង្ហាញរាយផ្លើស Asset - Screenshot បង្ហាញរាយផ្តល់ Asset ទៅបុគ្គលិក

(ដាក់រូបភាព Demo នីនេះ)

Demo 3 : Asset Status and Maintenance Form

ទីតាំងដាក់របស់ខ្លួន និងការបញ្ចូលពីគម្រោង - Screenshot បង្ហាញការប្រើប្រាស់ភាព (Available → Under Repair / Retired)

(ដាក់រូបភាព Demo នីនេះ)

Demo 4 : Search and Reporting Form

ຮູ້ນໍ້າກັບບົກຕະ - ບູກບັບຕີກາມດີຕົກຕາໂຄເຜິຍຮີໄຕ - Screenshot ບັນຫາລາຍງານ Asset - Screenshot ບັນຫາລາຍອຸປະສົງຮກ

(ដាក់ឲ្យបាត Demo ទីនេះ)

សេចក្តីថ្លែងការណ៍ របស់ខ្លួន ត្រូវបានរួមចិត្តក្នុង Screenshot ពីពិភពលោក ដើម្បីបញ្ជាក់ថា ប្រព័ន្ធអាជីវការបានដាក់ស្ថិជាដំឡើង។

1 2 . កម្រិតកំណត់ និងការពើរកអនាគត (Limitations and Future Enhancements)

ក្រុមកំណត់ - ប្រធានអគ្គបច្ចីលម្មយ (Admin) - ជាកម្មវិធ Desktop - មិនគណនាការណ៍តែតិចឡើងស្ថិយប្រភូ

ការពង្រីកអនកតែង - Role-based Access Control - Web-based System - Audit Log និង Deprecation - Bar
Code

១ ៣ . សន្លឹជាន (Conclusion)

ប្រព័ន្ធគ្រប់គ្រងទ្រព្យសម្បត្តិនេះ បង្ហាញពីការអនុវត្ត Design Patterns ចំនួន 14 គូងប្រព័ន្ធទិន្នន័យមានច្បាប់អាណីជកម្ម និងរចនាសម្ព័ន្ធច្បាប់លាស់។ ប្រព័ន្ធនេះអាចថែទាំ ងាយពាណិជ្ជកម្ម និងអាជីវកម្មបែន្ទុមនៅអនាគតភាព។

គម្រោងនេះបានតាមគ្រប់ទ្រីស្តី Design Patterns ដាមួយការអនុវត្តផ្តល់សេង ដោយបំពេញតាមតម្លៃរការផ្តើមសិក្សា និងវិស័យកម្មមួយដើម្បី។