Nhóm 5

Project Introduction:

* 1. Project Information

● Project name: Xây dựng hệ thống quản lý kí túc xá sinh viên tại đại học FPT

● Software type: Web

1.2 Project Team

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Full Name | Role | Email |
| DE160257 | Nguyễn Văn Tân | Member | tannvde160257@fpt.edu.vn |
| DE160415 | Phạm Ngọc Hùng | Member | hungpnde160415@fpt.edu.vn |
| DE170122 | Ngô Hải Thy | Member | thynhde170122@fpt.edu.vn |
| DE170561 | Đinh Trọng Hải Triều | Member | trieudthde170561@fpt.edu.vn |
| DE170649 | Ngô Hồ Gia Kiệt | Member | kietnhgde170649@fpt.edu.vn |

2. Product Background

Ngày nay, với số lượng sinh viên mỗi năm theo học tại Đại học FPT ngày càng tăng cao, vấn đề quản lý chỗ ở của sinh viên là một trong những vấn đề cấp thiết. Thực tế hiện nay, hầu hết việc quản lý chỗ ở hầu hết cũng chỉ quản lý trên sổ sách, giấy tờ, thủ công, tốn nhiều thời gian và dễ xảy ra sai sót.

Việc áp dụng công nghệ thông tin vào trong việc quản lý kí túc xá thực sự rất cần thiết, giúp cho việc quản lý được dễ dàng hơn, tránh việc thất lạc dữ liệu, xử lý thông tin một cách nhanh chóng hiệu quả với độ chính xác cao, tiết kiệm được thời gian và công sức của người quản lý cũng như nâng cao trải nghiệm tổng thể của sinh viên. Để giải quyết những vấn đề này và cung cấp một cách tiếp cận hợp lý hơn cho việc quản lý kí túc xá của sinh viên với mong muốn áp dụng công nghệ thông tin vào quản lý, nhóm em đề xuất phát triển Hệ thống quản lý kí túc xá sinh viên tại Đại học FPT.

Hệ thống quản lý kí túc xá sinh viên sẽ giúp quản lý việc phân bổ phòng, đăng kí kí túc xá cũng như lưu trữ và xử lý thông tin về sinh viên và tinh hình trong kí túc. Giúp cho việc quản lý trở nên dễ dàng và tiết kiệm thời gian hơn.

3. Existing Systems

3.1 Quản lí kí túc xá tại Đại học Quốc Gia – HCM

Link : <https://ktx.vnuhcm.edu.vn>

Ưu điểm:

● Dễ dàng tìm kiếm thông tin liên quan tới phòng cũng như giá cả cho sinh viên

● Có các thông báo về tinh hình và sự kiện của kí túc xá

Nhược điểm:

● Chưa phát triển hết được các chức năng của việc quản lý kí túc xá

● Sinh viên vẫn phải thực hiện đăng kí phòng theo hình thức thủ công

● Chưa lưu trữ được những thông tin cần thiết

3.2 Kí túc xá Đà Nẵng

Link : <http://ktxdn.vn>

Ưu điểm :

● Đầy đủ thông tin

● Giao diện thân thiện với người dùng

Nhược điểm :

● Chưa lưu trữ được thông tin cần thiết

● Chưa tích hợp được việc đăng kí online

3.3 Quản lí kí túc xá tại Đại học Quốc Gia Hà Nội

Link : <http://dangkynoitru.css.vnu.edu.vn>

Ưu điểm:

● Giao diện thân thiện với người dùng

●

Nhược điểm:

●

Main actors:

Guest :

Đăng kí, đăng nhập tài khoản

Xem các thông báo đến từ kí túc xá

User :

Đăng nhập tài khoản sinh viên

Xem (Các loại phòng có trong kí túc xá,chi tiết các phòng (các thiết bị, số lượng người/phòng, giá cả), quy định)

Search phòng còn trống

Đặt phòng theo giới tính ở thông tin cá nhân tài khoản (Đặt phòng thành công thì gửi mail xác nhận)

+ Cung cấp form điền

Thanh toán theo kì (Thanh toán phòng trước tiền điện nước sau)

+ Nếu thanh toán thành công thì gửi mail xác nhận

+ Xem hợp đồng

+ Xem số điện nước đã dùng

+ Hóa đơn thanh toán điện, nước, phòng

Gửi complain cho quản lý

Quản lý :

Đăng nhập

Quản lý hợp đồng

Quản lý sinh viên

Quản lý phòng ( Xem phòng, phân chia phòng)

Tìm kiếm

Kỷ luật

+ Gửi thông báo đến sinh viên

Thống kê

Biểu đồ Use Case tổng thể

A picture containing diagram, line

Description automatically generated

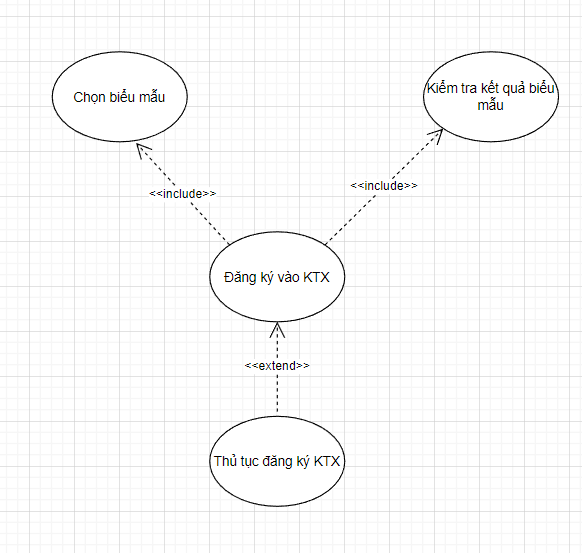
Biểu đồ phân rã các Use Case cho sinh viên

Sử dụng hệ thống

A picture containing text, diagram, circle, line

Description automatically generated

Đăng ký vào KTX



Đăng ký phòng

A picture containing text, diagram, screenshot, circle

Description automatically generated

Biểu đồ phân rã các Use Case cho quản lý

Nhận sinh viên vào KTX

A picture containing text, diagram, circle, screenshot

Description automatically generated

Loại sinh viên ra khỏi KTX

A picture containing text, diagram, circle, screenshot

Description automatically generated

* Database:

create table Account(

[id] int identity(1,1) primary key,

[username] varchar (10),

[password] Nvarchar (50),

[role] int,

CONSTRAINT fk\_acc foreign key (username) references Manager(m\_id),

CONSTRAINT fk\_acc2 foreign key (username) references Student(s\_id),

* [role] của manager là 0, của sinh viên là 1

create table Manager(

[m\_id] varchar (10) primary key,

[img] Nvarchar (50),

[fullname] Nvarchar (50),

[birth] smalldatetime,

[gender] char (1) check (gender in ('F', 'M', 'O')),

[phone] char (10),

[email] varchar (30),

[address] Nvarchar(50),

)

create table Student(

[s\_id] varchar (10) primary key,

[img] Nvarchar (50),

[fullname] Nvarchar (50),

[birth] smalldatetime,

[gender] char (1) check (gender in ('F', 'M', 'O')),

[phone] char (10),

[email] varchar (30),

[address] Nvarchar(50),

[room\_id] varchar (5),

CONSTRAINT fk\_student foreign key (room\_id) references Room (r\_id),

)

create table RoomType(

[type] char (5) primary key,

[size] int,

[price] money

)

create table Room (

[r\_id] varchar (5) primary key,

[r\_type] char (5),

[r\_size] int,

[r\_attendees] int check (r\_attendees <= r\_size),

[gender] char (1) check (gender in ('F', 'M', 'O')),

CONSTRAINT fk\_room foreign key (r\_type) references RoomType(type),

CONSTRAINT fk\_size foreign key (r\_size) references RoomType (size)

)

create table ElectricWaterUsed (

[semester] varchar (5) primary key,

[room\_id] varchar (5),

[first\_mth\_electric] int,

[second\_mth\_electric] int,

[third\_mth\_electric] int,

[final\_mth\_electric] int,

[first\_mth\_water] int,

[second\_mth\_ water] int,

[third\_mth\_ water] int,

[final\_mth\_ water] int,

CONSTRAINT fk\_ewu foreign key (room\_id) references Room (r\_id)

)

create table Bill (

[bill\_id] int identity(1,1) primary key,

[room\_id] varchar (5),

[s\_id] varchar (10),

[m\_id] varchar (10),

[semester] varchar (5),

[Cost] money,

[Date] datetime,

CONSTRAINT fk\_bill foreign key (room\_id) references Room (r\_id),

CONSTRAINT fk\_bill2 foreign key (s\_id) references Student (s\_id),

CONSTRAINT fk\_bill3 foreign key (m\_id) references Manager (m\_id),

CONSTRAINT fk\_bill4 foreign key (semester) references ElectricWaterUsed (semester),

)

create table Request(

[s\_id] varchar (10) primary key,

[room\_id] varchar (5),

[request] Ntext,

CONSTRAINT fk\_request foreign key (s\_id) references Student (s\_id),

)

create table Register(

[s\_id] varchar (10) primary key,

[room\_id] varchar (5),

[date] datetime,

[status] varchar(10) check (status in ('Success', 'Registered', 'Rejected')),

CONSTRAINT fk\_request foreign key (s\_id) references Student (s\_id),

CONSTRAINT fk\_ request2 foreign key (room\_id) references Room (r\_id)

)