

HCM University of Technology
Faculty of Computer Science & Engineering



Software Engineering

Assignment 4

Nguyen Phuc Hung - 1752259
Project: Smart Parking

Instructor:
Bui Hoai THANG

HCMC 2019-2020

1 Module Interface

My sub-system has one main module interface is 'IAdminView' (figure 1), which for user view, and our system must follow this interface to render the page.

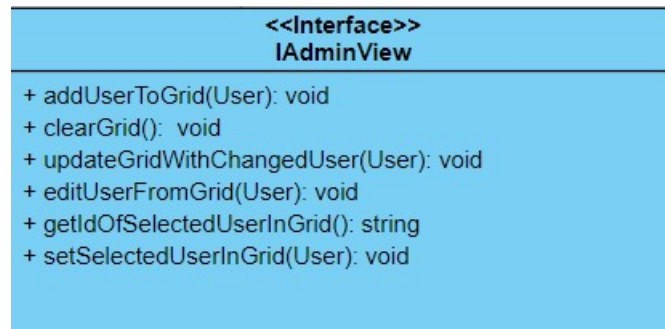


Figure 1: Module Interface

2 Class diagram

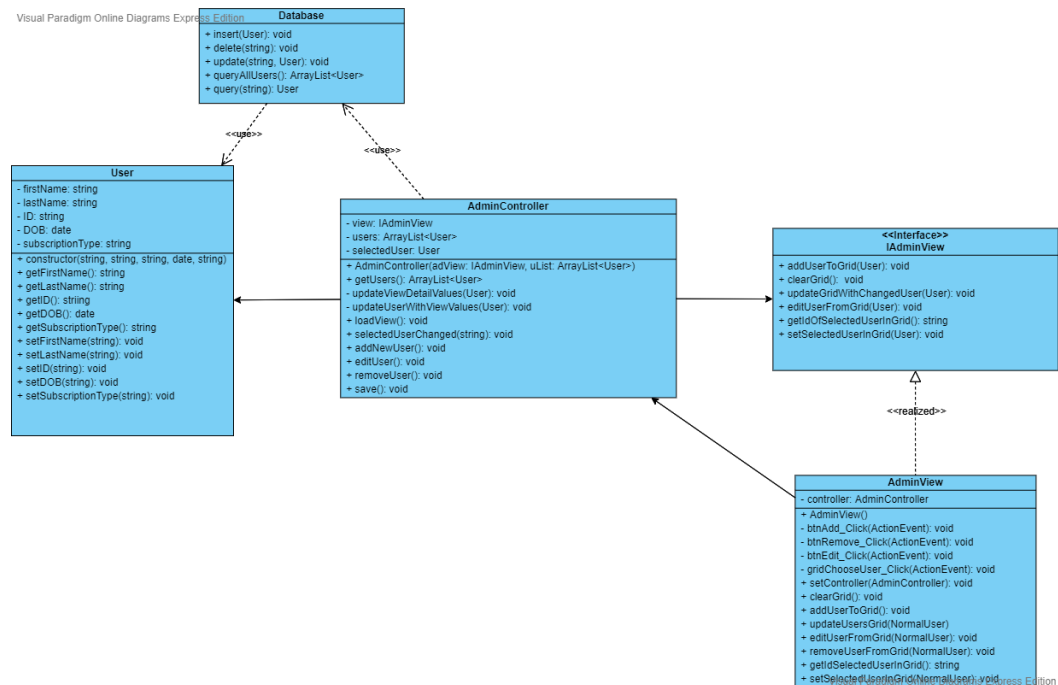


Figure 2: Class diagram

3 Method Description

3.1 AdminView

<i>Method name</i>	<i>Access modifier</i>	<i>Input Parameter Type</i>	<i>Return Type</i>	<i>Functionality</i>
btnAdd_Click	private	ActionEvent	void	Call function AddNewParking() of controller to handle click add button event
btnRemove_Click	private	ActionEvent	void	Call function RemoveParking() of controller to handle click Remove button event
btnEdit_Click	private	ActionEvent	void	Call function EditParking() of controller to Handle click Edit event
gridChooseUser_Click	private	ActionEvent	void	Call function AddNewParking() of controller to Handle click grid event
setController	public	Controller	void	Choose an instance of Controller class
clearGrid	public		void	Clear all information in grid
addUserToGrid	public	User	void	Add the input parking object to grid
updateGrid	public	User	void	Update the input parking object in view
removeUserFromGrid	public	User	void	Remove the input parking object from grid
getIdOfSelectedUser	public	User	string	Return ID of the selected parking in grid in string type

3.2 AdminController

<i>Method name</i>	<i>Access modifier</i>	<i>Input Parameter Type</i>	<i>Return Type</i>	<i>Functionality</i>
getUsers	public		ArrayList<User>	Get value of UserList attribute
addNewUser	public		void	Add new User object to the system
removeUser	public		void	Remove selected User from database and UserList attribute. Then update view
editUser	public		void	Edit User object in the system
loadView	public		void	Update view
save	public		void	Check for duplicate, if not, create User object base on input parameter then add it to UserList attribute and insert to database
updateViewDetailValues	private	User	void	Update the field text details of the user
updateUserWithViewValues	private	User	void	Save all User profile in the view to the UserList
selectedUserChanged	public	string	void	Set the selected User attribute to the input parameter

3.3 Database

<i>Method name</i>	<i>Access modifier</i>	<i>Input Parameter Type</i>	<i>Return Type</i>	<i>Functionality</i>
insert	public	User	void	Insert a user model to database
delete	public	string	void	Delete a user from database
update	public	string, User	void	Update existing user with new data
queryAllUsers	public		ArrayList<User>	Return list of User
query	public	string	User	Return User

3.4 User

Method name	Access modifier	Input Parameter Type	Return Type	Functionality
getFirstName	public		string	Get first name of a user
getLastName	public		string	Get last name of a user
getID	public		string	Get ID of a user
getSubscriptionType	public		string	Get Subscription type of a user
getDOB	public		string	Get Date of birth of a user
setFirstName	public	string	void	Set first name of a user
setLastName	public	string	void	Set last name of a user
setID	public	string	void	Set ID of a user
setDOB	public	string	void	Set Date of birth of a user
setSubscriptionType	public	string	void	Set Subscription type of a user

4 Sequence Diagram

Here is the sequence diagram for “Add any user’s profile” (figure 3)

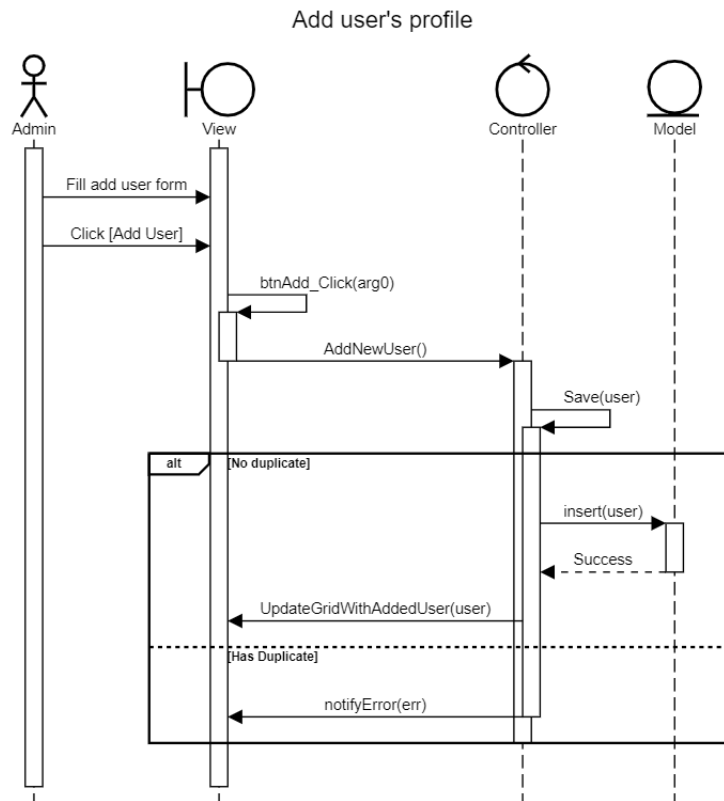


Figure 3: Sequence Diagram for adding user

5 Activity Diagram

Below is the activity diagram for the set of action of adding new user (figure 4)

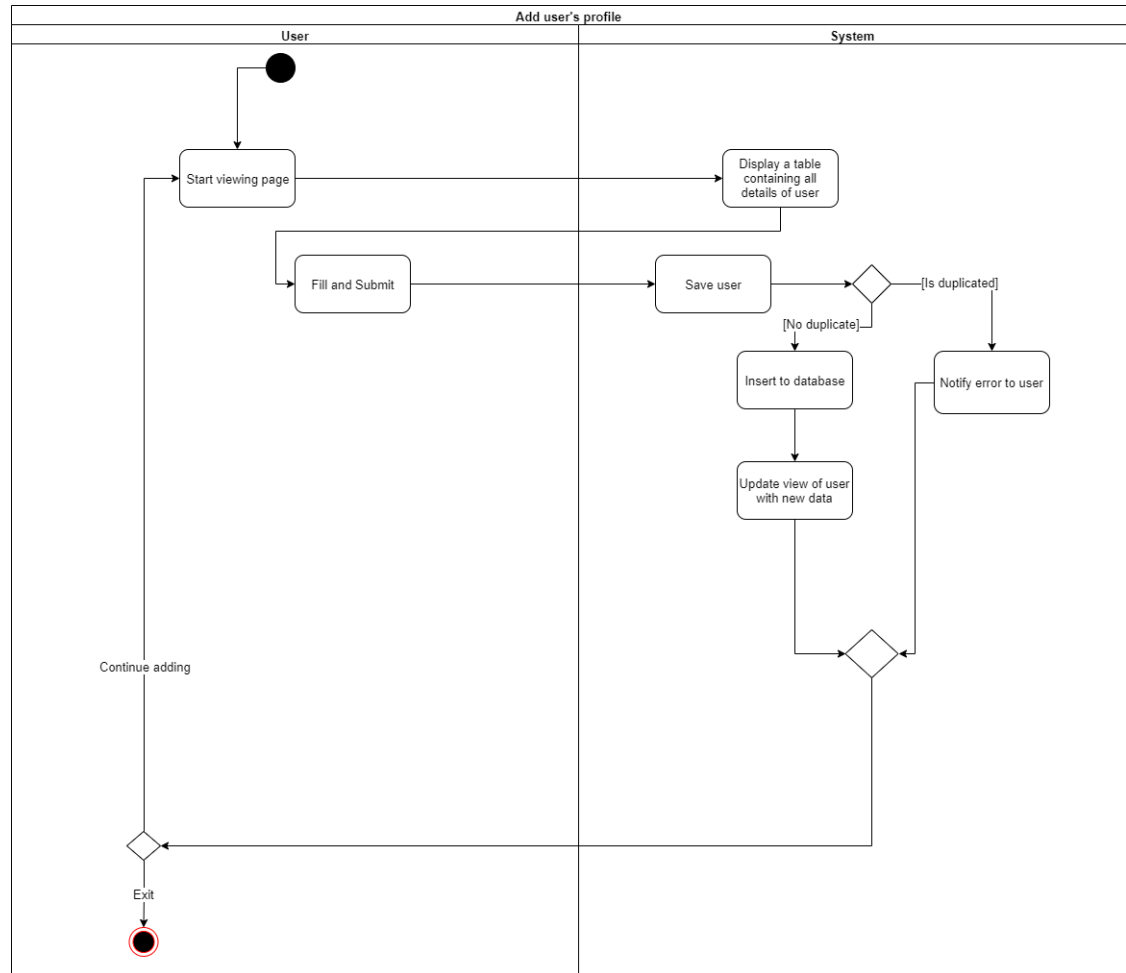


Figure 4: Activity diagram for adding user

6 Design Pattern

In my project, I use Model-View-Controller design Pattern.

I've implemented the Model View Controller because the development model is an established method to solve data separation (model) and interface (view) problems to avoid user interface changes in data handling and to adjust the data

without affecting / changing the user interface.

User
<ul style="list-style-type: none"> - firstName: string - lastName: string - ID: string - DOB: date - subscriptionType: string
<ul style="list-style-type: none"> + constructor(string, string, string, date, string) + getFirstName(): string + getLastName(): string + getID(): string + getDOB(): date + getSubscriptionType(): string + setFirstName(string): void + setLastName(string): void + setID(string): void + setDOB(string): void + setSubscriptionType(string): void

Figure 5: MVC: Model

AdminController
<ul style="list-style-type: none"> - view: IAdminView - users: ArrayList<User> - selectedUser: User
<ul style="list-style-type: none"> + AdminController(adView: IAdminView, uList: ArrayList<User>) + getUsers(): ArrayList<User> - updateViewDetailValues(User): void - updateUserWithViewValues(User): void + loadView(): void + selectedUserChanged(string): void + addNewUser(): void + editUser(): void + removeUser(): void + save(): void

Figure 6: MVC: Controller

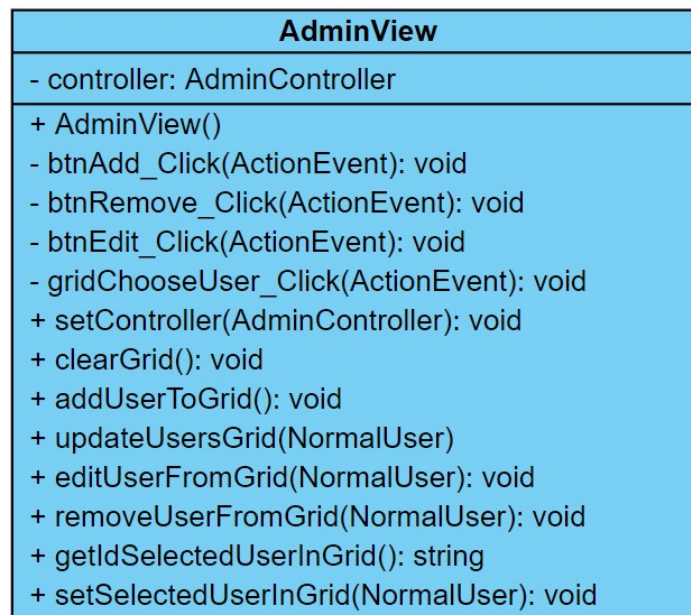


Figure 7: MVC: View

7 Demonstration

SPS

[Home](#) [User Management](#)

Settings

First name

Last name

Date of Birth

Subscription Type

Edit User

Add User

Remove User

Id	First Name	Last Name	Date Of Birth	SubscriptionType
1	Ernest Schuppe ...	Schuppe	12/12/2012	Premium
2	Janis	Vandervort	12/12/2012	Premium
3	Makenzie	Bode	12/12/2012	Premium
4	Clara	Towne	12/12/2012	Premium

SPS

[Home](#) [User Management](#)

Settings

First name

Hung

Last name

Phuc

Date of Birth

04/09/1999

Subscription Type

Premium

Edit User

Add User

Remove User

Figure 9: Add data: when admin when filled in user data

SPS

[Home](#)

User Management

[Settings](#)

First name

Last name

Date of Birth

Subscription Type

Edit User

Add User

Remove User

Id	First Name	Last Name	Date Of Birth	SubscriptionType
1	Ernest Schuppe ...	Schuppe	12/12/2012	Premium
2	Janis	Vandervort	12/12/2012	Premium
3	Makenzie	Bode	12/12/2012	Premium
4	Clara	Towne	12/12/2012	Premium
5	Hung	Phuc	04/09/1999	Premium

Figure 10: Add data: when admin click button Add User

SPS

Home

User Management

Settings

First name

Hung

Last name

Phuc

Date of Birth

04/09/1999

Subscription Type

Premium

Edit User

Add User

Remove User

Id	First Name	Last Name	Date Of Birth	SubscriptionType
1	Ernest Schuppe ...	Schuppe	12/12/2012	Premium
2	Janis	Vandervort	12/12/2012	Premium
3	Makenzie	Bode	12/12/2012	Premium
4	Clara	Towne	12/12/2012	Premium
5	Hung	Phuc	04/09/1999	Premium

SPS

Home

User Management

Settings

First name

Long

Last name

Giang

Date of Birth

02/03/1990

Subscription Type

Premium

Edit User

Add User

Remove User

Id	First Name	Last Name	Date Of Birth	SubscriptionType
1	Ernest Schuppe ...	Schuppe	12/12/2012	Premium
2	Janis	Vandervort	12/12/2012	Premium
3	Makenzie	Bode	12/12/2012	Premium
4	Clara	Towne	12/12/2012	Premium
5	Hung	Phuc	04/09/1999	Premium

Figure 12: Edit data: when admin edit user data

SPS

[Home](#) [User Management](#)

[Settings](#)

First name

Last name

Date of Birth

Subscription Type

Edit User

Add User

Remove User

Id	First Name	Last Name	Date Of Birth	SubscriptionType
1	Ernest Schuppe ...	Schuppe	12/12/2012	Premium
2	Janis	Vandervort	12/12/2012	Premium
3	Makenzie	Bode	12/12/2012	Premium
4	Clara	Towne	12/12/2012	Premium
5	Long	Giang	02/03/1990	Premium

Figure 13: Edit data: when admin click button Edit User

SPS

Home

User Management

Settings

First name

Last name

Date of Birth

Subscription Type

Edit User

Add User

Remove User

Id	First Name	Last Name	Date Of Birth	SubscriptionType
1	Ernest Schuppe ...	Schuppe	12/12/2012	Premium
2	Janis	Vandervort	12/12/2012	Premium
3	Makenzie	Bode	12/12/2012	Premium
4	Clara	Towne	12/12/2012	Premium
5	Hung	Phuc	04/09/1999	Premium

SPS

Home

User Management

Settings

First name

Last name

Date of Birth

Subscription Type

Edit User

Add User

Remove User

Id	First Name	Last Name	Date Of Birth	SubscriptionType
1	Ernest Schuppe ...	Schuppe	12/12/2012	Premium
2	Janis	Vandervort	12/12/2012	Premium
3	Makenzie	Bode	12/12/2012	Premium
4	Clara	Towne	12/12/2012	Premium

Figure 15: Remove data: after admin remove user