

EEET2482/COSC2082 – Software Engineering Design, Advanced Programming Techniques.

Assignment Report

"TIME BANK" APP FOR DAILY ACTIVITIES

Lecturer: Mr. Linh Tran

Team Members:

Tran Manh Cuong (s3974735)

Truong Quang Bao Loc (s3965528)

Phan Nhat Minh (s3978598)

Pham Quang Man (s3804811)

Date: 20/01/2024

Table of Contents

I. INTRODUCTION	3
II. APPLICATION DESIGN AND DEVELOPMENT	3
1. Software Design (Class Diagram)	3
Class Diagram	3
Description of Each Class	5
Class Relationships	17
2. Implementation Result	19
a. Welcome Screen	19
Result Summary	19
Screenshots of Sample Result	20
b. Basic & Time Period Features	21
Result Summary	21
Screenshots of Sample Result	24
III. DISCUSSION & CONCLUSION	49

I. INTRODUCTION

The "Time Bank" app is a program designed to revolutionize the way we assist each other with daily tasks. In today's fast-paced world, finding time for essential activities like plumbing, tutoring, garden care, or house cleaning can be challenging. Our app bridges this gap by creating a community where individuals exchange their time and skills instead of money. Whether you're a skilled plumber or a math whiz, "Time Bank" allows you to offer your expertise in return for help in areas you need. It's a symbiotic ecosystem fostering a spirit of cooperation and mutual support, redefining the currency of time.

- Link to our GitHub: https://github.com/tom474/time-bank-app
- Admin account information:

Username: adminPassword: rmit1234

- How to test our program with predefined data files:
 - 1. In your root folder, open the Terminal and change the directory to:

```
./time_bank_app/src
```

2. Use this command to compile the C++ files:

```
g++ main.cpp ./controllers/*.cpp ./models/service/*.cpp ./models/user/*.cpp ./utils/*.cpp ./views/*.cpp -o Group11_Program.exe
```

3. Use this command to run the program:

```
.\Group11_Program.exe
```

II. APPLICATION DESIGN AND DEVELOPMENT

1. Software Design (Class Diagram)

Class Diagram:

Link to the source file of the Class Diagram (use the RMIT email to view): https://drive.google.com/file/d/1GOqNbBWGfKSYEHaK5GWdb5X6Tu2I5Twy/view?usp=sharing

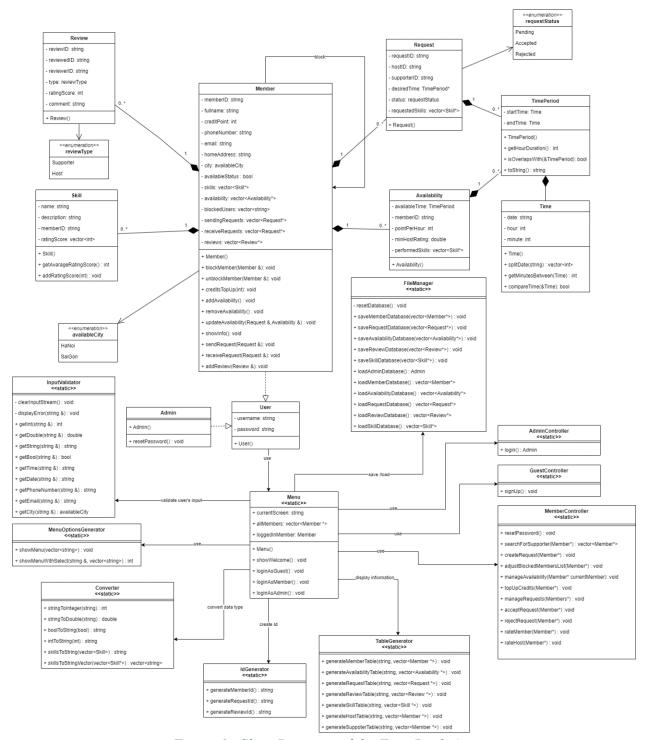


Figure 1: Class Diagram of the Time Bank App

Description of Each Class:

Class: User	Name and Data type	Description	Reason/Explanation
Attribute(s)	username: string	Username of the user's account.	Every user should have a unique username to login to their account.
	password: string	Password of the user's account.	Enhance the security for user's account and use for login purpose.

Class: Admin	Name and Data type	Description	Reason/Explanation
Attribute(s)		Inherit from User	
Method(s)	resetPassword(): void		The application needs this
		Member ID to search and	function to allow the admin
		set a temporary password	for resetting a specific
		for that member.	member password.

Class: Member	Name and Data type	Description	Reason/Explanation
Attribute(s)		Inherit from User	
	memberID: string	ID attribute of member.	Member Id act as an identity of member.
	fullname: string	Full name attribute of member.	Member needs this function to allow other members know their identity.
	creditPoint: int	Credit point attribute of member.	Member can use this point to book for a supporter and this attribute is used for search engine.
	phoneNumber: string	Phone number attribute of member.	Phone number attribute of a member use as contact method between host and supporter.
	email: string	Email attribute of member.	Email attribute of member use as a contact method between the admin and member.
	homeAddress: string	Home address attribute of member	The home address of the member helps the application ensure that the member lives in supported area.

	city: availableCity	The city attribute is constrained by an Enum availableCity [Hanoi, Saigon].	The current location of the member allows the application to provide information of supported in that city.
	availableStatus: bool	The attribute describes the status of member.	The attribute allows the member to control their available status to be booked on the application.
	skills: vector <skill*></skill*>	A list that contains all skills of the member.	Members need this attribute to store their performance skills and showcase to the hosts.
	availability: vector <availability*></availability*>	A list that contains all the available times that the member can receive a request.	Members need this attribute to manage their time and control their performance skills in that specific time periods
	sendingRequests: vector <request*></request*>	A list that contains all the requests has been made by the member.	Members need this attribute to keep track of the status of all requests that they have sent to supporters.
	receivingRequest: vector <request*></request*>	A list that contains all the request has been sent to that member.	Members need this attribute to manage requests that have been sent to them and they can reject or accept the request.
	reviews: vector <review*></review*>	A list that contains all the reviews of a member.	Members need this attribute to view all the reviews they received and help the host to find the most suitable supporter to help them when they can view supporter's reviews.
	blockedUsers: vector <string></string>	A list that contains all the blocked member id.	Members can block a member from booking them or search for them.
Method(s)	blockMember (Member&): void	This function saves the blocked member id to the blocked user vector.	This function allows member to block another member.
	unblockMember (Member&): void	This function allows members to remove a member from their blocked member vector.	This function allows members to unblock a specific member.

1'4-TII (' A)		TT1 114 1
creditsTopUp(int): void	members can use this function to exchange dollar to credit points.	The credit point is the currency in this application and members can use this function when they do not have enough credit point.
addAvailability(): void	This function asks for necessary information and create new Availability for member.	Availability allows members to be book by other members, so they need this function to create new availability.
removeAvailability(): void	This function asks for existing availability and delete that.	Members need this function to delete a specific availability if it is not useful anymore.
updateAvailability (Request &, Availability &): void	This function will automatically cut the time for the request and create new availability if the rest period of previous availability is over one hour.	The application needs this function to update the available time for specific availability to avoid overlap request time of members on an availability.
sendRequest(Request &): void	This function will store request to the supporter's sending request list.	The application needs this function to allow the host creates request to the supporter.
showInfo(): void	Display member detail as many tables.	The application needs this function to allow members view their own information.
receiveRequest (Request&): void	Store requests have been sent to the member.	The application needs this function to automate adding the request to the receiving requests list of supporters when the host creates a new request.
addReview (Review&): void	Store reviews of members.	The application needs this function to allow the host and supporter to rate each other.

Class: Skill	Name and Data type	Description	Reason/Explanation
Attribute(s)	name: string	Name attribute of skill.	This attribute ensures the skill is unique and when it assign to a member other

			member can find an appropriate supporter.
	description: string	Description attribute of skill.	This attribute allows the member to clarify and create a better
			understanding of their skills.
	memberID: string	Member ID of the member.	Used like a foreign key and when the data is load from the data the application can find all member's skill with this attribute.
	ratingScore: vector <int></int>	List of rating scores.	The application needs this attribute to store the rating score from the host's review.
Method(s)	getAverageRating Score(): int	Calculate the average rating score of a skill.	The application needs this function to calculate all the rating scores from the host and displays to user.
	addRatingScore(int): void	Add a score to the rating score list of a skill.	The application needs this function to allow the host rate the supporter's skills.

Class: Availability	Name and Data type	Description	Reason/Explanation
Attribute(s)	availableTime: TimePeriod	Available time of the supporter.	Members needs this attribute to describe their available time.
	memberID: string	Member ID of the supporter user.	Used like a foreign key and when the data is load from the data the application can find all member's availability with this attribute.
	pointPerHour: int	Point per hour (rate) set by supporter user.	Supporter set hourly rate, which hosts must satisfy when requesting.
	minHostRating: double	Minimum host rating, set by supporter.	Supporter can filter hosts.
	performedSkills: vector <skill*></skill*>	Available skills of supporter at corresponding time.	Supporter can select skills corresponds to their availability.

Class: Request	Name and Data type	Description	Reason/Explanation
Attribute(s)	requestID: string	ID attribute for request.	Request Id act as an identity of member.
	hostID: string	Member ID of the host user.	Used like a foreign key and when the data is load from the data the application can find all member's sending requests with this attribute.
	supporterID: string	Member ID of the supporter user.	Used like a foreign key and when the data is load from the data the application can find all member's receiving requests with this attribute.
	desiredTime: TimePeriod	Proposed date & time of the request.	This attribute allows the users to create desire time for their request.
	status: requestStatus	Status of the request is constrained by an Enum ["Pending", "Accepted", "Rejected"].	Upon creation, the request will have status Pending. Supporter can accept or reject the request.
	requestedSkills: vector <skill*></skill*>	Required skills for this request, choose by host.	Host can add relevant skills required for the request.

Class: Review	Name and Data type	Description	Reason/Explanation
Attribute(s)	reviewID: string	ID attribute of review.	The application needs this attribute to ensure all reviews are unique.
	reviewedID: string	MemberID of the member receiving review.	Used like a foreign key to identify the member that was reviewed.
	reviewerID: string	MemberID of the member sending review	Used like a foreign key to identify the member that created the review
	type: reviewType	Type of the review is constrained by an Enum [Supporter, Host].	The application uses the review type to categorize the review as either "Supporter" or "Host" based on member's selection and members can decide to rate a member's host skill or supporter skill.
	ratingScore: int	The score from the reviewer.	The application will use a rating score to improve the

			search engine. It allows users to search for a specific range of scores.
	comment: string	The comment from the reviewed.	The members use this attribute to give a detailed review message for a specific member.
Method(s)	resetPassword(): void	Admin can prompt a Member Id to search and set a temporary password for that member.	The application needs this function to allow the admin reset password for a member.

Class: Time	Name and Data type	Description	Reason/Explanation
Attribute(s)	date: string	Date string of the time.	Storing the date whole date
			string.
	hour: int	Hour part of the date.	The hour of the date string,
			used for storing time.
	minute: int	Minute part of the date.	The minute of the date
			string, used for storing time.
Method(s)	splitDate(string):	Split date data into an int	The application needs this
	vector <int></int>	vector.	function to split dates into
			vectors for easier
			manipulation. Also used for
			comparing dates.
	getMinutesBetween	Get number of minutes	Utility function, used for
	(Time): int	between 2 time periods.	calculating minutes,
			validating requests.
	compare(&Time):	Compare two time	Utility function, used for
	bool	periods.	validating a request
			creation.

Class:	Name and Data type	Description	Reason/Explanation
TimePeriod			
Attribute(s)	startTime	Starting time.	Users need this attribute to
			choose starting time for
			their Request/Availability.
	endTime	Ending time.	User needs this attribute to
			choose ending time for
			Request/Availability.
Method(s)	getHourDuration():	Get hour duration	The application needs this
	int	between start & end time.	function to calculate the
			duration between start time
			and end time to measure the
			point for supporter.

isOverlapsWith	Validate if a time period	Used for validating when
(&TimePeriod): bool	overlaps with another.	user create requests.
toString(): string	Return a string of the	Used for printing to
	time period.	console.

Class: Menu	Name and Data type	Description	Reason/Explanation
Attribute(s)	currentScreen: string	Name of current screen in Menu.	The application needs this attribute to keep track of the current Menu screen to display appropriate menu's title.
	loggedInMember: Member	Logged in member object.	Menu class has logged member as a static variable for other subclasses to use.
Method(s)	showWelcome(): void	Initial method of the app, shows the welcome screen.	The show welcome screen listens for user input to login as: guest member admin.
	loginAsGuest(): void	Allow login as Guest.	User can login as guest, without requiring username or password.
	loginAsMember(): void	Allow login as Member.	User can login to their registered account.
	loginAsAdmin(): void	Allow login as Admin.	Administrators can login to their account.

Class: Guest Controller	Name and Data type	Description	Reason/Explanation
Method(s)	signUp(): void	Allow signup from the Guest screen.	User can register for new account with sign up function.

Class:	Name and Data type	Description	Reason/Explanation
Admin			
Controller			
Method(s)	login(): Admin	Validate username and	The application needs this
		password from the users	function to allow members
		and return Admin or null.	to login with their username
			and password.

Class: Member Controller	Name and Data type	Description	Reason/Explanation
Method(s)	resetPassword(): void	Reset user password with a guided prompt.	The application needs this function to check the logged-in member has a password reset by admin and ask to create a new password.
	searchForSupporter (Member*): vector <member*></member*>	Search for suitable supporters with a guided prompt Members.	Member can search for suitable supporter, the system will base on the requested skills, and availability data to provide the best results.
	createRequest (Member*): void	Create a request for suitable supporters with a guided prompt.	Member can create requests on supporters that they want.
	adjustBlocked MembersList (Member*): void	Manage current logged in user's blocked member list.	Members can block users. The system will filter blocked users from results.
	manageAvailability (Member*): void	Manage current logged in user's availability as a supporter.	Members need this function to manage their current availability. They can set their available status, create, or delete availability.
	topUpCredits (Member*): void	Top up current logged in user credits.	Members can top up their account credits.
	manageRequests (Members*): void	Manage current logged in user requests.	Members need this function to manage their current requests. They can view their sending and receiving requests, accept a

		request and reject a
		request.
acceptRequest	Accept host's request.	Members can
(Member*): void		accept requests and
		the application will
		change association
		information in
		supporter
		availability and
		host's request
		status.
rejectRequest	Reject host's request.	Members can reject
(Member*): void		requests and the
		application will
		change association
		information in
		host's request
		status.
rateMember	Triggers the rate menu, allow user	Members can rate
(Member*): void	to rate hosts/supporters.	other members
		(host/supporters).
rateHost(Member*):	User can rate hosts in previous	Members need this
void	accepted requests.	function to rate
		hosts in previous
		requests and the
		application will
		create a review and
		send to that host.
rateSupporter	User can rate supporters in	Members need this
(Member*): void	previous accepted requests.	function to rate
		supporters in
		previous requests
		and the application
		will create a review
		and send to that
		host.

Class: FileManager	Name and Data type	Description	Reason/Explanation
Method(s)	resetDatabase(): void	Before saving the database, the program will delete all the data and save the new data.	The application needs this function to ensure the database stay consistency.

	saveMemberDatabase (vector <member*>): void</member*>	Save all members to database and call other saving function to save each member's skills, requests, availabilities, reviews,	The application needs this function to save all the members to the database.
	saveRequestDatabase (vector <request*>): void</request*>	Convert request attribute to string and save requests to the csv database.	The application needs this function to save all the requests to the database.
	saveSkillDatabase (vector <skill*>): void</skill*>	Convert Skill attribute to string and save to the csv database.	The application needs this function to save all the skills to the database.
	saveAvailabilityDatabase (vector <availability>): void</availability>	Convert availability attribute to string and save to the csv database.	The application needs this function to save all the availabilities to the database.
	saveReviewDatabase (vector <review*>): void</review*>	Convert review attribute to string and save to the csv database.	The application needs this function to save all the reviews to the database.
	loadAdminDatabase(): Admin	Read file csv and convert to Admin attribute.	The application needs this function to load all the admin data.
	loadMemberDatabase(): vector <member*></member*>	Read file csv then convert to member objects, and call other functions to load skills, reviews, availabilities, requests attributes.	The application needs this function to load all the members data.
	loadAvailabilityDatabase (): vector <availability*></availability*>	Read file csv and convert to Availability objects.	The application needs this function to load all the availabilities data.
	loadRequestDatabase(): vector <request*></request*>	Read file csv and convert to Request objects.	The application needs this function to load all the requests data.

loadReviewDatabase(): vector <review*></review*>	Read file csv and convert to Review objects.	The application needs this function to load all the reviews data.
loadSkillDatabase(): Vector <skill *=""></skill>	Read file csv and convert to Skill objects.	The application needs this function to load all the skills data.

Class: Converter	Name and Data type	Description	Reason/Explanation
Method(s)	stringToInteger(string): int	Convert string to integer.	Utility function to convert string to integer.
	stringToDouble(string): double	Convert string to double.	Utility function to convert string to double.
	boolToString(bool): string	Convert bool to string.	Utility function to convert bool to string.
	skillsToString (vector <skill>): string</skill>	Convert list of skills to a string with skill's name.	Utility function to convert vector of skills to string of skill's name.
	skillsToStringVector (vector <skill*>): vector<string></string></skill*>	Convert list of skills to list of string with skill's name.	Utility function to list of skills to list of string with skill's name.

Class:	Name and Data type	Description	Reason/Explanation
IDGenerator			
Method(s)	generateMemberID():	Search for the latest	The application needs this
	string	member ID and create	function to create a
		the next value of that	member ID for a new
		ID.	user.
	generateRequestID():	Search for the latest	The application needs this
	string	request ID and create	function to create a
		the next value of that	request ID for a new
		ID.	request.
	generateReviewID():	Search for the latest	The application needs this
	string	review ID and create the	function to create a
		next value of that ID.	review ID for a new
			review.

Class:	Name and Data type	Description	Reason/Explanation
TableGenerator			
Method(s)	generateMember Table(string, vector <member *="">): void</member>	Generate a table of member contains their attribute and a title of that table.	The application needs this function to display members data.
	generateAvailability Table(string, vector <availability *="">): void</availability>	Generate a table of availabilities contains their attribute and a title of that table.	The application needs this function to display availabilities data.
	generateRequest Table (string, vector <request *="">): void</request>	Generate a table of requests contains their attribute and a title of that table.	The application needs this function to display requests data.
	generateReview Table (string, vector <review *="">): void</review>	Generate a table of reviews contains their attribute and a title of that table.	The application needs this function to display reviews data.
	generateSkill Table(string, vector <skill *="">): void</skill>	Generate a table of skills contains their attribute and a title of that table.	The application needs this function to display skills data.
	generateHost Table(string, vector <member *="">): void</member>	Generate a table of members contains their attribute and a title of that table.	The application needs this function to display hosts data.
	generateSuppoter Table(string, vector <member *="">): void</member>	Generate a table of members contains their attribute and a title of that table.	The application needs this function to display supporters' data.

Class: InputValidator	Name and Data type	Description	Reason/Explanation
Method(s)	clearInputStream (): void	Clears the input stream in console.	The system automatically clears input stream to prepare for receiving new inputs.
	displayError (string&): void	Display error message to the console.	Utility function, used for displaying error messages upon system action failure.
	getDouble (string&): double	Get a double value user input from console.	Utility function, used for receiving a double value user input from console.
	getString (string&): string	Get a string value user input from console.	Utility function, used for receiving a string value user input from console.

getBool	Get a boolean value	Utility function, used for
(string&): bool	user input from	receiving a boolean value user
	console.	input from console.
getTime	Get a time string	Utility function, used for
(string&): string	value user input	receiving a time string value
	from console.	user input from console.
getDate	Get a date string	Utility function, used for
(string&): string	value user input	receiving a date string value
	from console.	user input from console.
getPhoneNumber	Get a phone number	Utility function, used for
(string&): string	string value user	receiving a phone number string
	input from console.	value user input from console.
getEmail	Get an email string	Utility function, used for
(string&): string	value user input	receiving an email string value
	from console.	user input from console.
getCity(string&):	Get a city string	Utility function, used for
availableCity	value user input	receiving a city string value
	from console.	user input from console.

Class:	Name and Data	Description	Reason/Explanation
MenuOptions	type		
Generator			
Method(s)	showMenu	Prints menu options	Utility function, widely used
	(vector <string>):</string>	denoted with numbers.	within the app to print Menu
	void		options.
	showMenuWith	Prints menu options	Similar to showMenu, this
	Select(string&,	denoted with numbers.	method also prompts user
	vector <string>): int</string>	Users select options	input, performs validation
	- '	with corresponding	and returns user input after it
		number.	exits.

Class Relationships:

• Skill is a composition part of Member.

Explanation: Skills are dependent on members. Members have a unique list of skills when deleting a member, all associated member's skills will be deleted. This will ensure the consistency of the data and avoid skills that do not belong to any members.

• Review is a composition part of Member.

Explanation: Reviews are dependent on members because reviews will lose all their meaning without belonging to any members. Members store reviews in a list and when deleting a member all member's reviews will be deleted. This will ensure the consistency of the review data.

• Member associate with another Member

Explanation: Members can block other members from searching and sending a request to them on the application. Members have a list of blocked members to store blocked member's id.

• TimePeriod is a composition part of Request.

Explanation: Time periods are dependent on requests. Each request has a specific time period to show the desired time of members. When a request is deleted that request's time period will be deleted.

• TimePeriod is a composition part of Availability.

Explanation: Time periods are dependent on availabilities. Each of the availability has a specific time period to show when members are available to support the host. When an availability is deleted, its time periods attribute is also deleted.

• Time is a composition part of TimePeriod.

Explanation: Time is dependent on time periods. TimePeriod has two Time attributes which are startTime and endTime. When a time-period is deleted, both startTime and endTime will be deleted.

• Menu associate with TableGenerator.

Explanation: The menu uses TableGenerator's functions to create a table. The menu has a list of Member attributes and sends it TableGenerator's functions as a parameter. This will enhance the data display for the application.

• Menu associate with MemberController.

Explanation: Menu use MemberController's functions for current Member use. The menu has a currentMember attribute and sends it to MemberController's functions as a parameter to invoke functions and change the current member's data.

• Menu associate with GuestController.

Explanation: Menu use GuestController's function for guest to use.

• Menu associate with FileManager.

Explanation: Menu use FileManager's to save and load members.

• Request is a composition part of Member.

Explanation: Requests are stored in the Member class, it shows the receiving requests and sending requests for members and this request cannot be a stand-alone object they need to be created and managed by members. When a member is deleted all its requests are deleted as well.

• Availability class is a composition part of Member class.

Explanation: Availability class is created and stored inside Member class, User can create additional or remove availability. Availability is deleted when the Member is deleted.

• Member class inherits from User class.

Explanation: Member class is a type of User class. Member can use the username and password attributes from User to login the application.

• Admin class inherits from User class.

Explanation: Admin class is a type of User class. Admin can use the username and password attributes from User to login the application.

• User class associate with Menu class.

Explanation: Users who are either Admin or Member can use Menu to interact with the data and each type of user can access the different functions. To be specific, members will be provided with a list of functions to interact with their data.

• Menu class associate with MenuOptionGenerator class.

Explanation: Menu class use MenuOptionGenerator's function for generating interactive options, which users can select using integer numbers (for instance: 1/2/3).

• Menu class associate with AdminController class.

Explanation: Menu use AdminController's login function.

• Menu class associate with InputValidator class

Explanation: Menu class associate with InputValidator. Menu uses InputValidator's functions to validate user input.

• Menu class associate with Converter class.

Explanation: Menu class associated with Converter class. Converter's functions are utility functions that convert the user's input into proper data types to save or display.

• Menu class associate with IDGenerator class.

Explanation: Menu class associated with IDGenerator class to generate unique ID. Whenever a new object that needs an ID is created. Menu will call IDGenerator's function to provide a unique ID for that object.

2. Implementation Result

a. Welcome Screen

Result Summary:

When any users run the program, the welcome screen will be displayed, and it will show the application's information including the course's details, application's name, and the development team's details. Then, the system will ask users whether they want to use the app as "Guest", "Member", or "Admin" and there will be a specific menu for each role which displays specific features for that role.

Screenshots of Sample Result:

```
EEET2482/COSC2082 ASSIGNMENT
''TIME BANK'' APPLICATION

Instructor: Mr.Tran Duc Linh
Group: Group No.11
s3974735, Tran Manh Cuong
s3965528, Truong Quang Bao Loc
s3978598, Phan Nhat Minh
s3804811, Pham Quang Man

Use the app as:
1. Guest
2. Member
3. Admin

Enter your choice (1-3):
```

Figure 2: Welcome Screen

```
Choose an action:
0. Exit
1. Sign up
2. View supporter details

Enter your choice (0-2):
```

Figure 3: Guest Menu

```
Choose an action:
0. Exit
1. View your information
2. Top up your credit point
3. Manage your availability
4. Manage your requests
5. Search for supporters
6. Create new request
7. Rate and review host/supporter
8. Block/unblock member

Enter your choice (0-8):
```

Figure 4: Member Menu

Choose an action:

0. Exit

1. Reset password for member

Enter your choice (0-1):

Figure 5: Admin Menu

b. Basic & Time Period Features

Result Summary:

Feature Name	Feature Description	Status	Bugs / Limitations
1. Guest sign up	A non-registered user can sign up for an account to access all the member functionalities.	Implemented	None
2. Guest view supporters' details	A non-registered user can view all the details of all the supporters available in the system but not their reviews.	Implemented	None
3. Admin login	An admin can log in using their username and password, then the system will validate and show an error message if the username and password are not correct. If the username and password are correct, the system will show the successful message and display the admin menu.	Implemented	None
4. Admin reset password for member	The system will show a list of members and the admin can choose a specific member to reset the password by entering their member ID. If that ID exists, the admin can set a temporary password for that member.	Implemented	None
5. Member login	A member can login using the account that they registered earlier to access other functionalities.	Implemented	None
6. Member view his/her information	A member can view their information including their account's details, availabilities, skills, requests, and reviews information.	Implemented	None
7. Member top up his/her credit point	A member can top up their credit points by cash so that they can request supporters in the system. Member is also required to verify their password to complete the credit point top up process.	Implemented	None

	1	1	
8. Member set his/her	Members can view their current	Implemented	None
available status	available status and change it to		
	available or unavailable to control		
	whether they will receive requests or		
2.2.4.4.4.4	not.	- 4	
9. Member add new	The system will ask the current member	Implemented	None
availability	for the necessary information to create a		
	new availability. Members can also		
	decide how many skills and which skills		
10.34. 1	they want to add to that availability.	T 1 . 1	3 7
10. Member remove	The system will show a table of	Implemented	None
his/her availability	member's availabilities. Members can		
	remove an availability by entering the		
11) (1 '	correct row number of that availability	Y 1 . 1	3. 7
11. Member view	A member can view a list of requests	Implemented	None
his/her sending	that they sent to other members.		
request	A	T141	Nama
12. Member view	A member can view a list of requests	Implemented	None
his/her receiving	received from other members.		
request	A Garagiannia a diat of a su dia a manager	I1	Nama
13. Member accept	After viewing a list of pending requests	Implemented	None
his/her receiving	received from other members, a member can accept a specific request from that		
request	list by entering the request's id. After the		
	current member accept the request		
	successfully, the request status will be		
	updated from "Pending" to "Accepted",		
	the current member's list of		
	availabilities will be updated, and the		
	system will automatically reject non-		
	accepted requests with overlapped time.		
14. Member reject	After viewing a list of pending requests	Implemented	None
his/her receiving	received from other members, a member		1,0110
request	can reject a specific request from that		
	list by entering the request's id. After the		
	current member reject the request		
	successfully, the request status will be		
	updated from "Pending" to "Reject".		
15. Member search	The system will ask the current member	Implemented	None
for suitable supporter	to enter the desired time, and it will base		
	on his/her living city, list of blocked		
	users, and credit points to find a list of		
	suitable supporters. The system is also		
	check for supporters' available status,		
	availabilities, and list of blocked users.		

16. Member create a new request 17. Member view his/her request history	The system will ask the current member to search for suitable supporters before creating a new request. Then, the current member must enter a valid supporter's id, desired time, and skills that he/she want to request, and the system will create and add a new request to the database. A member can use this function to view their request history. This function's purpose is to help the current member to view his sending and receiving request history before rating and reviewing any	Implemented	None
	hosts or supporters.		
18. Member as supporter rate his/her host	A member uses this function to rate hosts. He/she can only rate hosts in his/her previous accepted requests. The user needs to input the correct Request ID to create a review for the host of that request. The current member should rate the host on a scale of 1 to 5 points and leave a comment for that host.	Implemented	None
19. Member as host rate his/her supporter	A member uses this function to rate supporters. Member can only rate supporters in his/her previous accepted requests. The user needs to input the correct Request ID to create a review for the host of that request. The current member should rate the host on a scale of 1 to 5 and leave a comment for that host. Moreover, the current should create review for the supporter's skills by rate on a scale of 1 to 5 points and leave a comment for each of those skills.	Implemented	None
20. Member block another member	A member can block another member in the system. By adding that user to the blocked list. The blocked user cannot create a request to the current member and vice versa.	Implemented	None
21. Member unblock another member	A member can unblock a member from his/her blocked users list. The user needs to input the correct User ID shown from the blocked list. After the input is validated successfully, the system removes the blocked user from	Implemented	None

the blocked list. Both users can interact	
with each other on the system again.	

Screenshots of Sample Result:

1. Guest sign up

```
----- Guest Menu
Choose an action:
0. Exit
1. Sign up
2. View supporter details
Enter your choice (0-2): 1
  ----- Sign Up -----
Enter your username: nguyenvana
This username is already taken. Please enter another username!
Enter your username: testuser
Enter your password: 123456
Enter your fullname: Test User
Enter your phone number: 12345678
Invalid phone number! Please enter a 10-digit number!
Enter your phone number: 0123456789
Enter your email: testuser123
Invalid email! Please enter a valid email address!
Enter your email: testuser@gmail.com
Enter the city where you are living (Sai Gon/Ha Noi): Da Nang
Invalid input! Please enter 'Sai Gon' or 'Ha Noi'!
Enter the city where you are living (Sai Gon/Ha Noi): Sai Gon
Enter your home address: 123 Nguyen Van Linh
Are you available to support other members now? (yes/no): may be
Invalid input! Please enter 'yes' or 'no'!
Are you available to support other members now? (yes/no): yes
Now you will have to enter at least 1 skill
Enter your skill name: Programming
Please describe that skill: Programming with C++
Do you want to enter another skill? (yes/no): yes
Enter your skill name: UI Design
Please describe that skill: Design with Figma
Do you want to enter another skill? (yes/no): no
Register Successfully!
```

Figure 6: Guest signs up for a new account & Input validation test cases

```
member.csv X

image: bank_app > data > image: member.csv
You, 28 seconds ago | 1 author (You)

MB0001,nguyenvana,123456,Nguyen Van A,100,0123456789,nguyenvana@gmail.com,123 Nguyen Van Linh,Sai Gon,true,false,MB0008,MB0005,

MB0002,nguyenvanb,123456,Nguyen Van B,80,0987654321,nguyenvanb@gmail.com,456 Le Loi,Sai Gon,true,false,MB0010,MB0007,

MB0003,nguyenvanc,123456,Nguyen Van C,120,0112233445,nguyenvanc@gmail.com,789 Nguyen Hue,Sai Gon,true,false,

MB0004,nguyenvand,123456,Nguyen Van D,90,098877665,nguyenvand@gmail.com,567 Tran Hung Dao,Sai Gon,true,false,

MB0005,nguyenvane,123456,Nguyen Van E,110,0123987654,nguyenvane@gmail.com,321 Vo Van Tan,Sai Gon,true,false,

MB0006,nguyenvanf,123456,Nguyen Van F,95,0932109876,nguyenvanf@gmail.com,321 Vo Van Tan,Sai Gon,true,false,

MB0007,nguyenvang,123456,Nguyen Van F,95,09322109876,nguyenvanf@gmail.com,321 Vo Van Tan,Sai Gon,true,false,

MB0008,nguyenvanh,123456,Nguyen Van F,95,0932345678,nguyenvanf@gmail.com,789 Ba Trieu,Ha Noi,true,false,

MB0008,nguyenvani,123456,Nguyen Van H,85,0888777666,nguyenvanh@gmail.com,789 Ba Trieu,Ha Noi,true,false,

MB0009,nguyenvani,123456,Nguyen Van I,105,0777888999,nguyenvani@gmail.com,456 Cau Giay,Ha Noi,true,false,

MB0010,nguyenvanj,123456,Nguyen Van J,115,0666555444,nguyenvanj@gmail.com,456 Cau Giay,Ha Noi,true,false,
```

Figure 7: Member database before guest sign up for a new account

Figure 8: Member database after guest sign up for a new account

Figure 9: Skill database before guest sign up for a new account

```
skill.csv M X
 Cyber Security, Cybersecurity and Ethical Hacking, MB0006, 5, 3,
      Cloud Computing, Cloud Computing with AWS, MB0006,4,5,
       Data Science, Data Science with R and Python, MB0007,4,5,
      Big Data, Big Data Analytics using Hadoop, MB0007,4,4,
       Data Visualization, Data Visualization with Tableau, MB0007, 3, 5,
       Web Development, Web Development using NextJS, MB0008, 5, 4,
      SEO, Search Engine Optimization for Websites, MB0008, 3, 4,
       Mobile App Development, Mobile App Development with Flutter, MB0009,4,5,
       Database Management, Database Management with SQLite, MB0009, 4, 4,
      UI/UX Design,UI/UX Design for Mobile Apps,MB0009,3,5,
       Programming, Programming with Python and JavaScript, MB0010,5,4,
       Web Development, Web Development using React and NodeJS, MB0010, 4, 5,
      Database Management, Database Management with MongoDB, MB0010,3,4,
      Programming,Programming with C++,MB0011,
      UI Design,Design with Figma,MB0011,
```

Figure 10: Skill database after guest sign up for a new account

2. Guest view supporter's details

Choose an 0. Exit 1. Sign up 2. View su							
		All Supp	orters				
ID	Fullname	Skills	Phone Number	Email	Home Address	City	Status
MB0001	Nguyen Van A	Programming, Database Management, Web Development	0123456789	nguyenvana@gmail.com	123 Nguyen Van Linh	Sai Gon	Available
MB0002	Nguyen Van B	Programming, Database Management, Networking	0987654321	nguyenvanb@gmail.com	456 Le Loi	Sai Gon	Available
MB0003	Nguyen Van C	Graphic Design, Web Design, Video Editing	0112233445	nguyenvanc@gmail.com	789 Nguyen Hue	Sai Gon	Available
MB0004	Nguyen Van D	Data Analysis, Database Management	0998877665	nguyenvand@gmail.com		Sai Gon	Available
MB0005	Nguyen Van E	Mobile App Development, UI/UX Design	0123987654	nguyenvane@gmail.com	321 Vo Van Tan	Sai Gon	Available
MB0006	Nguyen Van F	Networking, Cyber Security, Cloud Computing	0932109876	nguyenvanf@gmail.com	234 Hoang Van Thu	Ha Noi	Available
MB0007	Nguyen Van G	Data Science, Big Data, Data Visualization	0912345678	nguyenvang@gmail.com	567 Ton Duc Thang	Ha Noi	Available
MB0008	Nguyen Van H	Web Development, SEO	0888777666	nguyenvanh@gmail.com	789 Ba Trieu	Ha Noi	Available
MB0009	Nguyen Van I	Mobile App Development, Database Management, UI/UX Design	0777888999	nguyenvani@gmail.com	123 Nguyen Trai	Ha Noi	Available
MB0010	Nguyen Van J	Programming, Web Development, Database Management	0666555444	nguyenvanj@gmail.com	456 Cau Giay	Ha Noi	Available

Figure 11: Guest view supporter's details

3. Admin login

```
EEET2482/COSC2082 ASSIGNMENT
"TIME BANK" APPLICATION
Instructor: Mr.Tran Duc Linh
Group: Group No.11
s3974735, Tran Manh Cuong
s3965528, Truong Quang Bao Loc
s3978598, Phan Nhat Minh
s3804811, Pham Quang Man
Use the app as:
1. Guest
2. Member
3. Admin
Enter your choice (1-3): 3
----- Login -----
Please enter your username: admin
Please enter your password: rmit1234
Login Successfully!
----- Admin Menu ------
Choose an action:
0. Exit
1. Reset password for member
Enter your choice (0-1):
```

Figure 12: Admin login successfully

Figure 13: Admin login unsuccessfully

4. Admin reset password for member

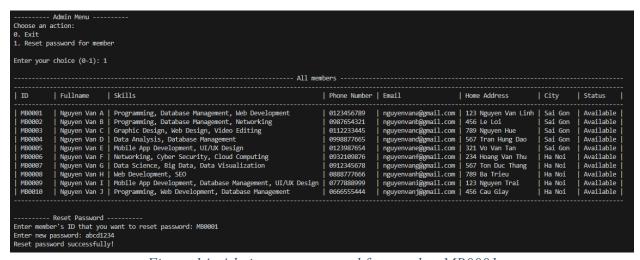


Figure 14: Admin reset password for member MB0001

```
Use the app as:
1. Guest
2. Member
3. Admin
Enter your choice (1-3): 2
----- Login -----
Please enter your username: nguyenvana
Please enter your password: abcd1234
Login Successfully!
Your password has been reset!
Enter new password: 123456
Change password successfully!
----- Member Menu
Choose an action:
0. Exit
1. View your information
2. Top up your credit point
3. Manage your availability

    Manage your requests
    Search for supporters

6. Create new request
7. Rate and review host/supporter
8. Block/unblock member
Enter your choice (0-8):
```

Figure 15: Member MB0001 login after admin reset password

5. Member login

```
Use the app as:
1. Guest
2. Member
3. Admin
Enter your choice (1-3): 2
----- Login ------
Please enter your username: nguyenvana
Please enter your password: 123456
Login Successfully!
----- Member Menu
Choose an action:
0. Exit
1. View your information
2. Top up your credit point
3. Manage your availability
4. Manage your requests
5. Search for supporters
6. Create new request
7. Rate and review host/supporter
8. Block/unblock member
Enter your choice (0-8):
```

Figure 16: Member login successfully

Figure 17: Member login unsuccessfully

6. Member view his/her information

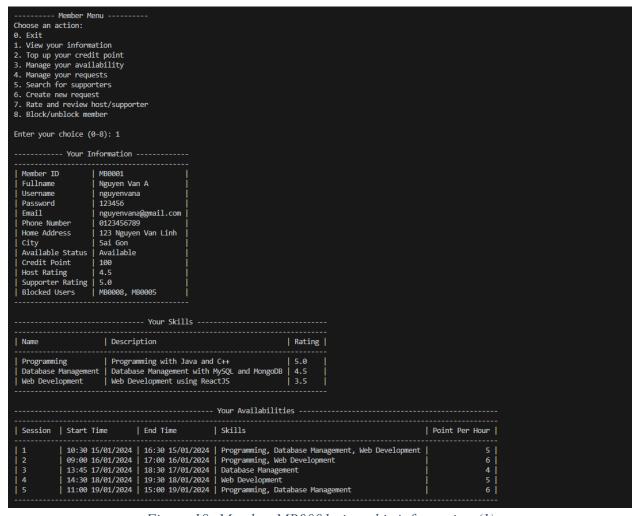


Figure 18: Member MB0001 views his information (1)

			Your Ava	ilabilities				
Session	Start Time	End Time	Skills			Point Per Ho	our	
1 2 3 4 5	10:30 15/01/2024 09:00 16/01/2024 13:45 17/01/2024 14:30 18/01/2024 11:00 19/01/2024	17:00 16/01/2024 18:30 17/01/2024 19:30 18/01/2024	Program Database Web Deve	ming, Web Developmer e Management		 	5 6 4 5 6	
			Your Se	ending Requests				
Request II) Host ID Supp	porter ID Start 1	ime	End Time	Requested Skills		Status	
RQ0009 RQ0010 RQ0011 RQ0012	MB0001 MB00 MB0001 MB00 MB0001 MB00 MB0001 MB00	902 09:30 1 303 17:00 1 304 17:00 1	15/01/2024 16/01/2024 15/01/2024 16/01/2024	11:30 15/01/2024 18:00 16/01/2024 19:00 15/01/2024 09:30 16/01/2024	Graphic Design, Web Des Data Analysis, Database	sign	Accepted Accepted Accepted Accepted	
				Your Receiving Requ	iests			
Request II) Host ID Supp	porter ID Start 1	ime	End Time	Requested Skills			Status
RQ8091 RQ9095 RQ9092 RQ9096 RQ9093 RQ9097 RQ9094 RQ9098	MB0002	901 09:30 1 901 12:15 1 901 08:00 1 901 10:00 1 901 09:00 1 901 15:30 1	15/01/2024 18/01/2024 15/01/2024 16/01/2024 16/01/2024 17/01/2024 17/01/2024 15/01/2024	10:30 18/01/2024 14:15 15/01/2024	Programming, Database Meb Development Programming, Web Developrogramming, Web Developrogramming, Web Developrogramming, Meb Developrogramming, Meb Developrogramming, Database Management Programming, Database Meanagement	opment opment opment	eb Development	Pending Accepted Pending Accepted Pending Accepted Pending Rejected
		Your Revie	ws					
Review ID	Reviewer ID	Reviewee ID Rat	ing Comm	ent				
RV0001 RV0002 RV0003 RV0004	MB0003 MB0004 MB0002 MB0003	MB0001 5 MB0001 5 MB0001 5 MB0001 4	Very Exce	llent Supporter Satisfied About The llent Host and Friendly	Service 			
					·			

Figure 19: Member MB0001 views his information (2)

7. Member top up his/her credit point

Figure 20: Member MB0001 tops up his credit point

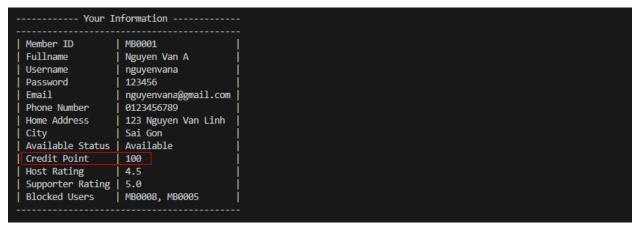


Figure 21: Member MB0001's credit point before topping up

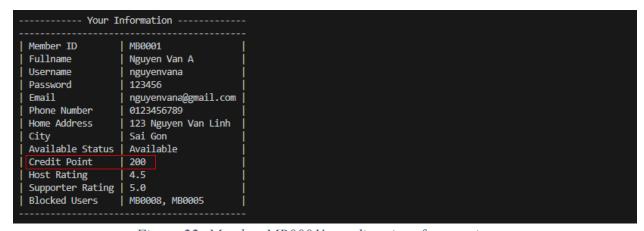


Figure 22: Member MB0001's credit point after topping up

8. Member set his/her available status

Figure 23: Member MB0001 sets his available status

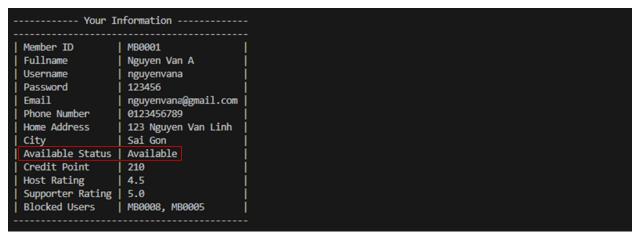


Figure 24: Member MB0001's available status before changing

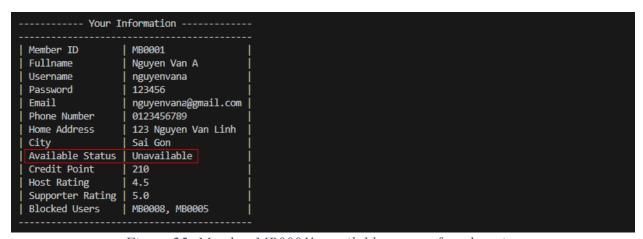


Figure 25: Member MB0001's available status after changing

9. Member add new availability

```
--- Manage Availability ------
Choose an action:
Exit

    Set your available status
    Add availability
    Remove availability

Enter your choice (0-3): 2
                                                           -- Your Current Availability -
| Session | Start Time
                                        | End Time
                                                                 | Skills
                                                                                                                                       | Point Per Hour |
                 10:30 15/01/2024 | 16:30 15/01/2024 |
                                                                    Programming, Database Management, Web Development
                 09:00 16/01/2024 | 17:00 16/01/2024 |
                                                                   Programming, Web Development
                 13:45 17/01/2024 | 18:30 17/01/2024 | Database Management
               | 14:30 18/01/2024 | 19:30 18/01/2024 | Web Development
| 11:00 19/01/2024 | 15:00 19/01/2024 | Programming, Database Management
 ----- Add Availability -----
Enter your start date (dd/mm/yyyy): 20-01-2024
Invalid date format! Please use dd/mm/yyyy format!
Enter your start date (dd/mm/yyyy): 20/01/2024
Enter your start time (hh:mm): 09-00
Invalid time format! Please use hh:mm format!
Enter your start time (hh:mm): 09:00
Enter your end date (dd/mm/yyyy): 20/01/2024
Enter your end time (hh:mm): 15:00
Enter your point per hour: 5
Enter your min host rating: 1
You have to choose at least 1 skill to perform:

    Programming
    Database Management

3. Web Development
Enter a choice: 1
Do you want to add more skill? (yes/no): yes
Enter a choice: 2
Do you want to add more skill? (yes/no): no
Add availability successfully!
```

Figure 26: Member MB0001 adds a new availability

Your Availabilities						
Session Start Time	End Time	Skills	Point Per Hour			
2	2024 17:00 16/01/2024 2024 18:30 17/01/2024 2024 19:30 18/01/2024	. •	5 6 4 5 6			

Figure 27: Member MB0001's availabilities before adding a new availability

Your Availabilities								
Session	Start Time	End Time	Skills	Point Per Hour				
1	10:30 15/01/2024	16:30 15/01/2024	Programming, Database Management, Web Development	5				
2	09:00 16/01/2024	17:00 16/01/2024	Programming, Web Development	6				
3	13:45 17/01/2024	18:30 17/01/2024	Database Management	4				
4	14:30 18/01/2024	19:30 18/01/2024	Web Development	5				
5	11:00 19/01/2024	15:00 19/01/2024	Programming, Database Management	6				
6	09:00 20/01/2024	15:00 20/01/2024	Programming, Database Management	5				
		· 	·					

Figure 28: Member MB0001's availabilities after adding a new availability

10. Member remove his/her availability

```
Manage Availability ------
Choose an action:
Exit

    Set your available status
    Add availability

Remove availability
Enter your choice (0-3): 3
                                              -- Your Current Availability
 Session | Start Time
                              | End Time
                                                  | Skills
                                                                                                       | Point Per Hour |
            10:30 15/01/2024 | 16:30 15/01/2024 | Programming, Database Management, Web Development
             09:00 16/01/2024 | 17:00 16/01/2024 | Programming, Web Development
                                                                                                                      4 5
                                18:30 17/01/2024 | Database Management
             13:45 17/01/2024 |
             14:30 18/01/2024 | 19:30 18/01/2024 | Web Development
             11:00 19/01/2024 | 15:00 19/01/2024 | Programming, Database Management
            09:00 20/01/2024 | 15:00 20/01/2024 | Programming, Database Management
          - Remove Availability --
Enter the session that you want to delete the availability: 6
Remove availability successfully!
```

Figure 29: Member MB0001 removes an existing availability

Your Availabilities								
Session Start Time End	Time Skills		Point Per Hou	ır				
2	30 15/01/2024 Programming, 00 16/01/2024 Programming, 30 17/01/2024 Database Mana	Web Development agement	Development 	5 6 4				
5 11:00 19/01/2024 15:0	30 18/01/2024 Web Developmo 00 19/01/2024 Programming, 00 20/01/2024 Programming,	Database Management		5 6 5				

Figure 30: Member MB0001's availabilities before removing an existing availability

Your Availabilities							
Session	Start Time	End Time	Skills	Point Per Hour			
1 2 3 4 5	09:00 16/01/2024 13:45 17/01/2024 14:30 18/01/2024	17:00 16/01/2024 18:30 17/01/2024 19:30 18/01/2024	Programming, Database Management, Web Development Programming, Web Development Database Management Web Development Programming, Database Management	5 6 4 5 6			

Figure 31: Member MB0001's availabilities after removing an existing availability

11. Member view his/her sending request

```
-- Manage Request ----
Choose an action:
0. Exit
1. View your sending requests
2. View your receiving requests
3. Accept a receiving requests
4. Reject a receiving requests
Enter your choice (0-4): 1
                                          ----- Sending Requests
| Request ID | Host ID | Supporter ID | Start Time
                                                              | End Time
  RQ0009
              MB0001
                          MB0002
                                           09:30 15/01/2024 | 11:30 15/01/2024 | Programming, Database Management
                                                                                                                          Accepted
                                           17:00 16/01/2024 | 18:00 16/01/2024 | Graphic Design, Web Design
17:00 15/01/2024 | 19:00 15/01/2024 | Data Analysis, Database Management
               MB0001
                           MB0003
  R00010
                                                                                                                           Accepted
  R00011
               MB0001
                           MB0004
                                                                                                                           Accepted
  R00012
               MB0001
                          MB0005
                                           08:30 16/01/2024 | 09:30 16/01/2024 | UI/UX Design
                                                                                                                          | Accepted |
```

Figure 32: Member MB0001 views his sending requests

12. Member view his/her receiving request

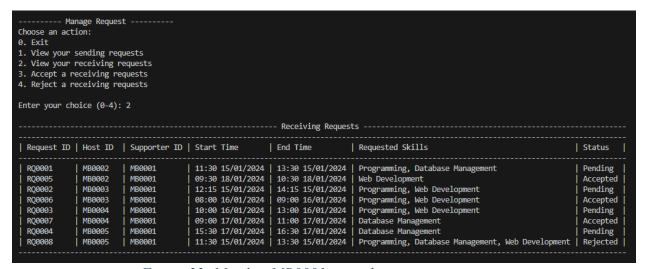


Figure 33: Member MB0001 views his receiving requests

13. Member accept his/her receiving request

```
- Manage Request -----
Choose an action:
0. Exit
1. View your sending requests
2. View your receiving requests
3. Accept a receiving requests
4. Reject a receiving requests
Enter your choice (0-4): 3
              ------ Pending Requests
| Request ID | Host ID | Supporter ID | Start Time
                                                          | End Time
                                                                               Requested Skills
                                                                                                                     | Status |
                                         | 11:30 15/01/2024 | 13:30 15/01/2024 | Programming, Database Management | Pending
              MB0002
                                        | 12:15 15/01/2024 | 14:15 15/01/2024 | Programming, Web Development | 10:00 16/01/2024 | 13:00 16/01/2024 | Programming, Web Development | 15:30 17/01/2024 | 16:30 17/01/2024 | Database Management
  RQ0002
              MB0003
                         MB0001
  RQ0003
              MB0004
                         MB0001
                                                                                                                     Pending
  RQ0004
                         MB0001
                                                                                                                     Pending
              MB0005
 ----- Accept Request ------
Enter the request's ID that you want to accept: RQ0005
Invalid ID! Please enter again!
Enter the request's ID that you want to accept: RQ0001
Are you confirm to accept this request? (yes/no): yes
Request accepted successfully!
Your availabilities has been updated!
```

Figure 34: Member MB0001 accepts request RQ0001

```
------ Your Information ------
Member ID
                MB0001
Fullname
                | Nguyen Van A
Username
               nguyenvana
               123456
Password
Email
               | nguyenvana@gmail.com
Phone Number
               0123456789
               | 123 Nguyen Van Linh
Home Address
                Sai Gon
City
Available Status | Available
Credit Point
                200
Host Rating
                4.5
Supporter Rating | 5.0
                MB0008, MB0005
Blocked Users
```

Figure 35: Member MB0001's credit point before accepting request RQ0001

```
----- Your Information ------
Member ID
               MB0001
Fullname
               | Nguyen Van A
               nguyenvana
Username
               123456
Password
Email
               | nguyenvana@gmail.com
Phone Number | 0123456789
Home Address
               | 123 Nguyen Van Linh
City
                 Sai Gon
Available Status | Available
Credit Point | 210
Host Rating
             4.5
Supporter Rating | 5.0
Blocked Users
               | MB0008, MB0005
```

Figure 36: Member MB0001's credit point after accepting request RQ0001

Session Start Time End Time Skills Point Per Hour 1 10:30 15/01/2024 16:30 15/01/2024 Programming, Database Management, Web Development 5 2 09:00 16/01/2024 17:00 16/01/2024 Programming, Web Development 6 3 13:45 17/01/2024 18:30 17/01/2024 Database Management 4 4 14:30 18/01/2024 19:30 18/01/2024 Web Development 5				Your Availabilities	
2 09:00 16/01/2024 17:00 16/01/2024 Programming, Web Development 6 3 13:45 17/01/2024 18:30 17/01/2024 Database Management 4	Session	Start Time	End Time	Skills	Point Per Hour
3 13:45 17/01/2024 18:30 17/01/2024 Database Management 4	1	10:30 15/01/2024	16:30 15/01/2024	Programming, Database Management, Web Development	5
	2	09:00 16/01/2024	17:00 16/01/2024	Programming, Web Development	6
4 14:30 18/01/2024 10:30 18/01/2024 Web Development		13:45 17/01/2024	18:30 17/01/2024	Database Management	4
4 14.30 10/01/2024 13.30 10/01/2024 Web Development	4	14:30 18/01/2024	19:30 18/01/2024	Web Development	5
5 11:00 19/01/2024 15:00 19/01/2024 Programming, Database Management 6	5	11:00 19/01/2024	15:00 19/01/2024	Programming, Database Management	6

Figure 37: Member MB0001's availabilities before accepting request RQ0001

			Your Availabilities	
Session S	Start Time	End Time	Skills	Point Per Hour
1 0	99:00 16/01/2024	17:00 16/01/2024	Programming, Web Development	6
2 1	3:45 17/01/2024	18:30 17/01/2024	Database Management	4
3 1	4:30 18/01/2024	19:30 18/01/2024	Web Development	5
4 1	1:00 19/01/2024	15:00 19/01/2024	Programming, Database Management	6
5 1	l0:30 15/01/2024	11:30 15/01/2024	Programming, Database Management, Web Development	5
6 1	13:30 15/01/2024	16:30 15/01/2024	Programming, Database Management, Web Development	5

Figure 38: Member MB0001's availabilities after accepting request RQ0001

				- Your Receiving Re	quests	
Request ID	Host ID	Supporter ID	Start Time	End Time	Requested Skills	Status
RQ0001	MB0002	MB0001	11:30 15/01/2024	13:30 15/01/2024	Programming, Database Management	Pending
RQ0005	MB0002	MB0001	09:30 18/01/2024	10:30 18/01/2024	Web Development	Accepted
RQ0002	MB0003	MB0001	12:15 15/01/2024	14:15 15/01/2024	Programming, Web Development	Pending
RQ0006	MB0003	MB0001	08:00 16/01/2024	09:00 16/01/2024	Programming, Web Development	Accepted
RQ0003	MB0004	MB0001	10:00 16/01/2024	13:00 16/01/2024	Programming, Web Development	Pending
RQ0007	MB0004	MB0001	09:00 17/01/2024	11:00 17/01/2024	Database Management	Accepted
RQ0004	MB0005	MB0001	15:30 17/01/2024	16:30 17/01/2024	Database Management	Pending
RQ0008	MB0005	MB0001	11:30 15/01/2024	13:30 15/01/2024	Programming, Database Management, Web Development	Rejected

Figure 39: Member MB0001's receiving requests before accepting request RQ0001

Your Receiving Requests						
Request ID	Host ID	Supporter ID	Start Time	End Time	Requested Skills	Status
RQ0001	MB0002	MB0001	11:30 15/01/2024	13:30 15/01/2024	Programming, Database Management	Accepted
RQ0005	MB0002	MB0001	09:30 18/01/2024	10:30 18/01/2024	Web Development	Accepted
RQ0002	MB0003	MB0001	12:15 15/01/2024	14:15 15/01/2024	Programming, Web Development	Rejected
RQ0006	MB0003	MB0001	08:00 16/01/2024	09:00 16/01/2024	Programming, Web Development	Accepted
RQ0003	MB0004	MB0001	10:00 16/01/2024	13:00 16/01/2024	Programming, Web Development	Pending
RQ0007	MB0004	MB0001	09:00 17/01/2024	11:00 17/01/2024	Database Management	Accepted
RQ0004	MB0005	MB0001	15:30 17/01/2024	16:30 17/01/2024	Database Management	Pending
RQ0008	MB0005	MB0001	11:30 15/01/2024	13:30 15/01/2024	Programming, Database Management, Web Development	Rejected

Figure 40: Member MB0001's receiving requests after accepting request RQ0001

Member ID	MB0002
Fullname	Nguyen Van B
Username	nguyenvanb
Password	123456
Email	nguyenvanb@gmail.com
Phone Number	0987654321
Home Address	456 Le Loi
City	Sai Gon
Available Status	Available
Credit Point	80
Host Rating	0.0
Supporter Rating	0.0
Blocked Users	MB0010, MB0007

Figure 41: Member MB0002's credit point before request RQ0001 was accepted

Your In	formation
Member ID	MB0002
Fullname	Nguyen Van B
Username	nguyenvanb
Password	123456
Email	nguyenvanb@gmail.com
Phone Number	0987654321
Home Address	456 Le Loi
City	Sai Gon
Available Status	Available
Credit Point	70
Host Rating	0.0
Supporter Rating	0.0
Blocked Users	MB0010, MB0007

Figure 42: Member MB0002's credit point after request RQ0001 was accepted

			Your Se	ending Requests		
Request ID	Host ID	Supporter ID	Start Time	End Time	Requested Skills	Status
RQ0001	MB0002	MB0001	11:30 15/01/2024	13:30 15/01/2024	Programming, Database Management	Pending
RQ0005	MB0002	MB0001	09:30 18/01/2024	10:30 18/01/2024	Web Development	Accepted

Figure 43: Member MB0002's sending requests before request RQ0001 was accepted

			Your Sen	nding Requests		
Request ID	Host ID	Supporter ID	Start Time	End Time	Requested Skills	Status
RQ0001	MB0002	MB0001	11:30 15/01/2024	13:30 15/01/2024	Programming, Database Management	Accepted
RQ0005	MB0002	MB0001	09:30 18/01/2024	10:30 18/01/2024	Web Development	Accepted

Figure 44: Member MB0002's sending requests after request RQ0001 was accepted

14. Member reject his/her receiving request

```
-- Manage Request ----
Choose an action:
0. Exit
1. View your sending requests
2. View your receiving requests
3. Accept a receiving requests
4. Reject a receiving requests
Enter your choice (0-4): 4
                                               ----- Pending Requests ----
| Request ID | Host ID | Supporter ID | Start Time
                                                                  | End Time
                                              11:30 15/01/2024 | 13:30 15/01/2024 | Programming, Database Management | Pending
  R09991
               MB0002
                            MB0001
                                             | 12:15 15/01/2024 | 14:15 15/01/2024 | Programming, Web Development | 10:00 16/01/2024 | 13:00 16/01/2024 | Programming, Web Development | 15:30 17/01/2024 | 16:30 17/01/2024 | Database Management
  RQ0002
                 MB0003
                             MB0001
                                                                                                                                  Pending
  RQ0003
                 MB0004
                             MB0001
 RQ0004
                            MB0001
                MB0005
                                                                                                                                 Pending
     ----- Reject Request --
Enter the request's ID that you want to reject: RQ0005
Invalid ID! Please enter again!
Enter the request's ID that you want to reject: RQ0001
Are you confirm to reject this request? (yes/no): yes
Request rejected successfully!
```

Figure 45: Member MB0001 rejects request RQ0001

	Your Receiving Requests					
Request ID	Host ID	Supporter ID	Start Time	End Time	Requested Skills	Status
RQ0001	MB0002	MB0001	11:30 15/01/2024	13:30 15/01/2024	Programming, Database Management	Pending
RQ0005	MB0002	MB0001	09:30 18/01/2024	10:30 18/01/2024	Web Development	Accepted
RQ0002	MB0003	MB0001	12:15 15/01/2024	14:15 15/01/2024	Programming, Web Development	Pending
RQ0006	MB0003	MB0001	08:00 16/01/2024	09:00 16/01/2024	Programming, Web Development	Accepted
RQ0003	MB0004	MB0001	10:00 16/01/2024	13:00 16/01/2024	Programming, Web Development	Pending
RQ0007	MB0004	MB0001	09:00 17/01/2024	11:00 17/01/2024	Database Management	Accepted
RQ0004	MB0005	MB0001	15:30 17/01/2024	16:30 17/01/2024	Database Management	Pending
RQ0008	MB0005	MB0001	11:30 15/01/2024	13:30 15/01/2024	Programming, Database Management, Web Development	Rejected

Figure 46: Member MB0001's receiving requests before rejecting request RQ0001

Your Receiving Requests						
Request ID	Host ID	Supporter ID	Start Time	End Time	Requested Skills	Status
RQ0001	MB0002	MB0001	11:30 15/01/2024	13:30 15/01/2024	Programming, Database Management	Rejected
RQ0005	MB0002	MB0001	09:30 18/01/2024	10:30 18/01/2024	Web Development	Accepted
RQ0002	MB0003	MB0001	12:15 15/01/2024	14:15 15/01/2024	Programming, Web Development	Pending
RQ0006	MB0003	MB0001	08:00 16/01/2024	09:00 16/01/2024	Programming, Web Development	Accepted
RQ0003	MB0004	MB0001	10:00 16/01/2024	13:00 16/01/2024	Programming, Web Development	Pending
RQ0007	MB0004	MB0001	09:00 17/01/2024	11:00 17/01/2024	Database Management	Accepted
RQ0004	MB0005	MB0001	15:30 17/01/2024	16:30 17/01/2024	Database Management	Pending
RQ0008	MB0005	MB0001	11:30 15/01/2024	13:30 15/01/2024	Programming, Database Management, Web Development	Rejected

Figure 47: Member MB0001's receiving requests after rejecting request RQ0001

Request ID Host ID Supporter ID Start Time	
	1
RQ0001	

Figure 48: Member MB0002's sending requests before request RQ0001 was rejected

Request ID Host ID Supporter ID Start Time		Your Sending Requests	
	Request ID Host ID Supporter	ID Start Time End Time	Requested Skills Status
R00005 MB0002 MB0001 09:30 18/01/2024 10:30 18/01/2024 Web Development Accepted	RQ0001		5 5

Figure 49: Member MB0002's sending requests after request RQ0001 was rejected

15. Member search for suitable supporter

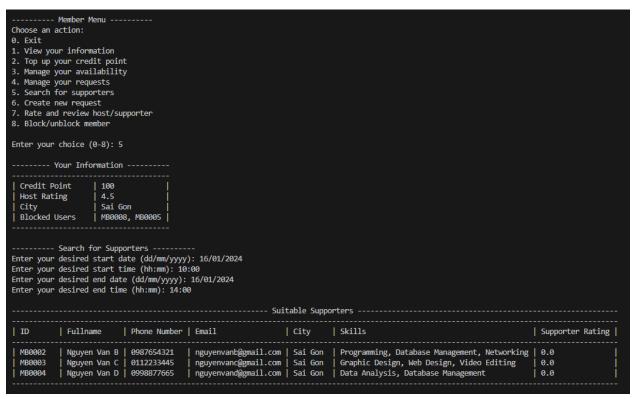


Figure 50: Member MB0001 searches for suitable supporters

16. Member create a new request

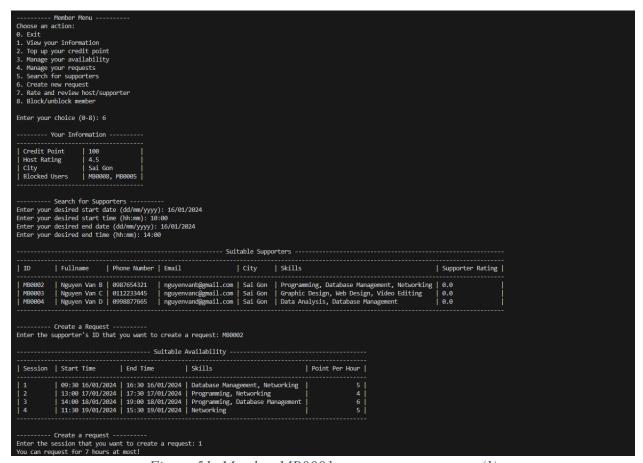


Figure 51: Member MB0001 creates a new request (1)

```
Enter the session that you want to create a request: 1
You can request for 7 hours at most!
Enter your desired start date (dd/mm/yyyy): 16/01/2024
Enter your desired start time (hh:mm): 10:00
Enter your desired end date (dd/mm/yyyy): 16/01/2024
Enter your desired end time (hh:mm): 14:00

Available skills that you can request:

1. Database Management

2. Networking
Enter a choice: 1
Do you want to add another skill? (yes/no): yes
Available skills that you can request:

1. Database Management

2. Networking
Enter a choice: 2
Do you want to add another skill? (yes/no): no
Request created successfully!
```

Figure 52: Member MB0001 creates a new request (2)

			Your	Sending Requests		
Request II) Host ID	Supporter ID	Start Time	End Time	Requested Skills	Status
RQ0009 RQ0010 RO0011	MB0001 MB0001 MB0001	MB0002 MB0003 MB0004	17:00 16/01/2024	1 18:00 16/01/2024	Programming, Database Management Graphic Design, Web Design Data Analysis, Database Management	Accepted Accepted Accepted
RQ0012	MB0001	MB0005		1 09:30 15/01/2024	, , , , , , , , , , , , , , , , , , , ,	Accepted

Figure 53: Member MB0001's sending requests before new request was created

	Your 9	Sending Requests		
Request ID Host ID	Supporter ID Start Time	End Time	Requested Skills	Status
RQ0009 MB0001 RQ0010 MB0001 RQ0011 MB0001 RQ0012 MB0001 RQ0013 MB0001	MB0002 09:30 15/01/2024 MB0003 17:00 16/01/2024 MB0004 17:00 15/01/2024 MB0005 08:30 16/01/2024 MB0002 10:00 16/01/2024	19:00 15/01/2024 09:30 16/01/2024	Graphic Design, Web Design Data Analysis, Database Management UI/UX Design	Accepted Accepted Accepted Accepted Pending

Figure 54: Member MB0001's sending requests after new request was created

		Your Red	eiving Requests		
Request ID Host ID	Supporter ID	Start Time	End Time	Requested Skills	Status
RQ0009 MB0001	MB0002	09:30 15/01/2024	11:30 15/01/2024	Programming, Database Management	Accepted

Figure 55: Member MB0002's sending requests before new request was created



Figure 56: Member MB0002's sending requests after new request was created

17. Member view his/her request history

```
----- Rate & Review Member
Choose an action:
0. Exit
1. View your request history
2. Rate and review your host
3. Rate and review your supporter
Enter your choice (0-3): 1
------- Request History
| Request ID | Host ID | Supporter ID | Start Time
                                                             | End Time
                                                                                 | Requested Skills
                                                                                                                                         Status
  RQ0009
               MB0001
                           MB0002
                                           09:30 15/01/2024 | 11:30 15/01/2024 | Programming, Database Management
                                                                                                                                           Accepted
               MB0001
                           MB0003
                                           17:00 16/01/2024 | 18:00 16/01/2024 | Graphic Design, Web Design
  R00010
                                                                                                                                           Accepted
  RQ0011
               MB0001
                           MB0004
                                           17:00 15/01/2024 | 19:00 15/01/2024 | Data Analysis, Database Management
                                                                                                                                           Accepted
  RQ0012
               MB0001
                           MB0005
                                           08:30 16/01/2024 | 09:30 16/01/2024 | UI/UX Design
                                                                                                                                           Accepted
                                           11:30 15/01/2024 | 13:30 15/01/2024 | Programming, Database Management 09:30 18/01/2024 | 10:30 18/01/2024 | Web Development 12:15 15/01/2024 | 14:15 15/01/2024 | Programming, Web Development
  RQ0001
                           MB0001
                                                                                                                                           Pending
               MB0002
  RQ0005
               MB0002
                           MB0001
                                                                                                                                           Accepted
  RQ0002
               MB0003
                           MB0001
                                                                                                                                           Pending
                                           08:00 16/01/2024 | 09:00 16/01/2024 | Programming, Web Development
  RQ0006
               MB0003
                           MB0001
                                                                                                                                           Accepted
  RQ0003
               MB0004
                           MB0001
                                           10:00 16/01/2024 | 13:00 16/01/2024 | Programming, Web Development
                                                                                                                                           Pending
                                           09:00 17/01/2024 | 11:00 17/01/2024 | Database Management
15:30 17/01/2024 | 16:30 17/01/2024 | Database Management
               MB0004
                           MB0001
  RQ0007
                                                                                                                                           Accepted
  RQ0004
                           MB0001
                                                                                                                                           Pending
               MB0005
                           MB0001
                                           11:30 15/01/2024 | 13:30 15/01/2024 | Programming, Database Management, Web Development | Rejected
```

Figure 57: Member MB0001 views his request history

18. Member as supporter rate his/her host

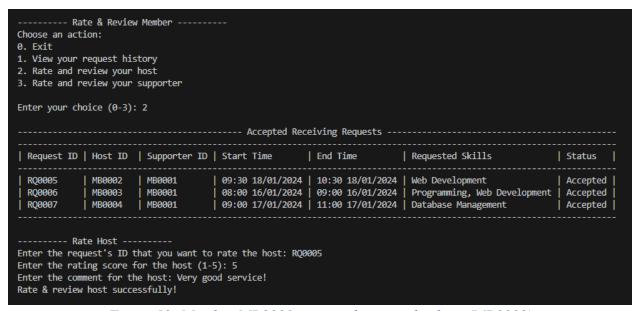


Figure 58: Member MB0001 rates and reviews his host (MB0002)

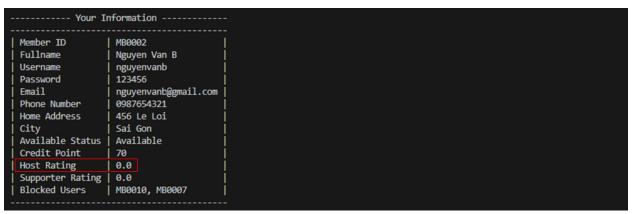


Figure 59: Member MB0002's host rating before receiving MB0001's rating and review

```
------ Your Information ------
Member ID
                 MB0002
Fullname
                 Nguyen Van B
Username
                | nguyenvanb
                123456
Password
Email
                | nguyenvanb@gmail.com
Phone Number
                 0987654321
                456 Le Loi
Home Address
                 Sai Gon
City
Available Status | Available
Credit Point
                  70
Host Rating
                 5.0
Supporter Rating | 0.0
Blocked Users
                | MB0010, MB0007
```

Figure 60: Member MB0002's host rating after receiving MB0001's rating and review

		Your Se	nding Requests		
Request ID Host	ID Supporter ID	Start Time	End Time	Requested Skills	Status
RQ0001 MB00 RQ0005 MB00			13:30 15/01/2024 10:30 18/01/2024	Programming, Database Management Web Development	Pending Accepted
		Your Rec	eiving Requests		
Request ID Host	ID Supporter ID	Start Time	End Time	Requested Skills	Status
RQ0009 MB00	01 MB0002	09:30 15/01/2024	11:30 15/01/2024	Programming, Database Management	Accepted
You have no review	to display!				

Figure 61: Member MB0002's reviews before receiving MB0001's rating and review

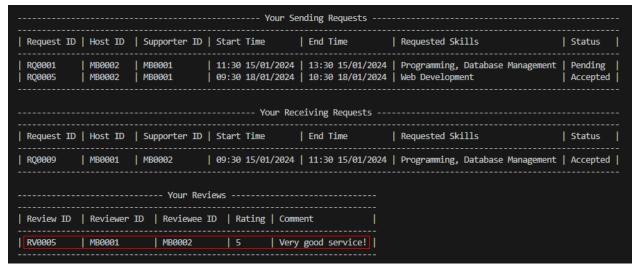


Figure 62: Member MB0002's reviews after receiving MB0001's rating and review

19. Member as host rate his/her supporter

```
-- Rate & Review Member -----
Choose an action:
0. Exit
1. View your request history
2. Rate and review your host
3. Rate and review your supporter
Enter your choice (0-3): 3
             ------ Accepted Sending Requests
| Request ID | Host ID | Supporter ID | Start Time
                                                           | End Time
                                                                              Requested Skills
                                                                                                                    Status
                                         09:30 15/01/2024 | 11:30 15/01/2024 | Programming, Database Management
                                                                                                                    | Accepted |
                                         17:00 16/01/2024 | 18:00 16/01/2024 | Graphic Design, Web Design
17:00 15/01/2024 | 19:00 15/01/2024 | Data Analysis, Database Management
 R00010
               MB0001
                          MB0003
                                                                                                                     Accepted
 R09911
               MB0001
                          MB0004
                                                                                                                     Accepted
 RQ0012
               MB0001
                        MB0005
                                       | 08:30 16/01/2024 | 09:30 16/01/2024 | UI/UX Design
                                                                                                                     Accepted |
 ----- Rate Supporter -----
Enter the request's ID that you want to rate the supporter: RQ0009
Enter the rating score for the supporter (1-5): 4
Enter the comment for the supporter: Good work!
----- Rate Supporter's Skills ------
Skill 1: Programming
Enter the rating score for the supporter's 'Programming' skill (1-5): 5
Skill 2: Database Management
Enter the rating score for the supporter's 'Database Management' skill (1-5): 4
Rate & review supporter successfully!
```

Figure 63: Member MB0001 rates and reviews his supporter (MB0002)

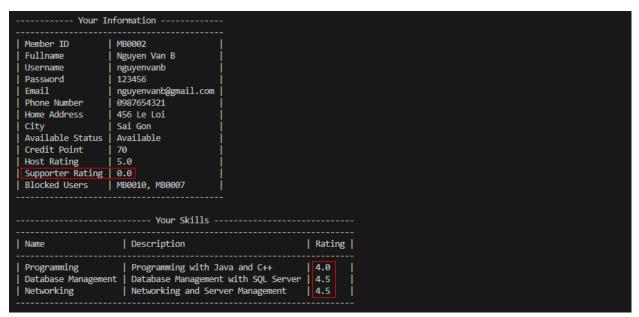


Figure 64: Member MB0002's supporter rating and skill rating before receiving MB0001's rating and review



Figure 65: Member MB0002's supporter rating and skill rating after receiving MB0001's rating and review

			You	ur Receiving Request	:s		
Request ID	Host ID	Supporter ID	Start Time	End Time	Requested Sk	tills	Status
RQ0009	MB0001 I	MB0002	09:30 15/01/	/2024 11:30 15/01/	2024 Programming,	Database Management	Accepted
		Your Revie	ws				
Review ID	Reviewer ID	Reviewee ID	Rating	Comment	1		
RV0005	MB0001	MB0002	5	Very good service!	1		

Figure 66: Member MB0002's reviews before receiving MB0001's rating and review



Figure 67: Member MB0002's reviews after receiving MB0001's rating and review

20. Member block another member

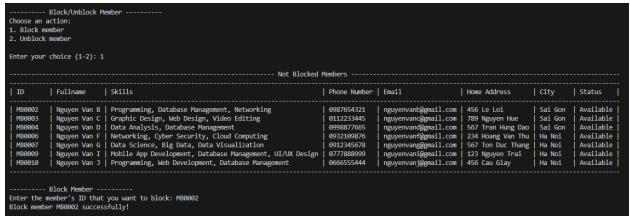


Figure 68: Member MB0001 blocks member MB0002

Figure 69: Member MB0001's blocked users list before block member MB0002

Figure 70: Member MB0001's blocked users list after block member MB0002

21. Member unblock another member

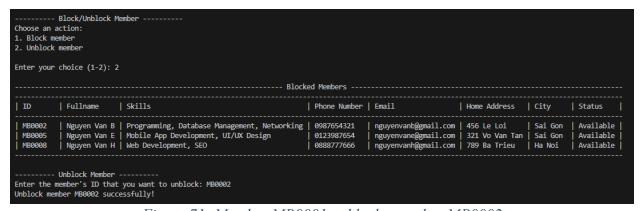


Figure 71: Member MB0001 unblocks member MB0002

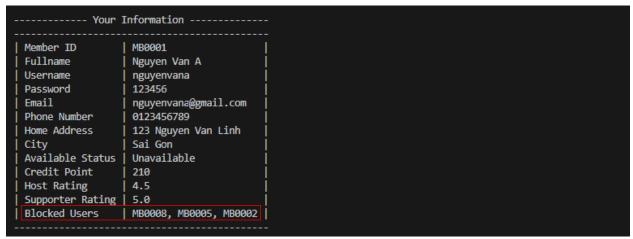


Figure 72: Member MB0001's blocked users list before unblock member MB0002

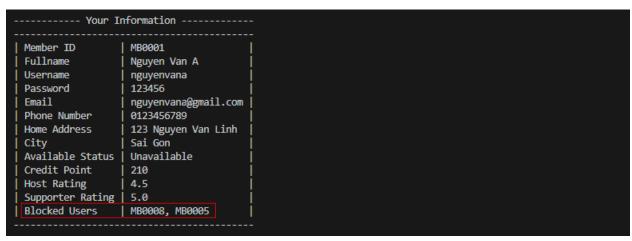


Figure 73: Member MB0001's blocked users list after unblock member MB0002

III. DISCUSSION & CONCLUSION

To sum up, our group met all the assignment requirements, showing that we used the information and skills we learned in class. We did thorough testing using a variety of test cases, trying to cover as many scenarios as possible and including handling a lot of exceptions. However, we are aware that some hidden test cases may not have been fully covered.

As a first step, we made a simple class diagram to see how the program was organized and how system objects interacted with each other. But as the project got further along, we switched to the Model-View-Controller (MVC) framework and added more helper classes, such as InputValidator, Table Generator, TimePeriod, etc. Because of this, our final class diagram was different from the first one, but the basic structure stayed the same. Setting up a class diagram at the start of the project was a smart move that helped our team turn the conceptual framework into concrete code.

We learned important things from this assignment. As a group project, it gave us a chance to improve our teamwork while coding by using tools like GitHub, writing code that looks good and can be easily maintained, and adding meaningful comments. One important thing we learned was how important it is to clearly plan and assign tasks. This made our team's time management

much better and made sure we could work together effectively and efficiently. This project also gave us a great chance to get better at solving problems, thinking critically and systematically, and talking to each other. Lastly, we think that the technical skills we've learned about C++, data structures, and programming will be very useful in our future careers and will give us a solid base for further research and use in these areas.