THE CRAFTSMEN

HOME REPAIRE

Mobile Application

Helwan university
Faculty of Computer and Artificial Intelligence

Information systems department
Graduation project represented by:

[Farida Abd Elrhman Fathy Elsayed (201900569)]

[Norhan Ashraf Abdelnaby Mohammed(201900912)]

[Menna mounir fathy rizk(201900853)]

[Heba Allah Khaled mohamed ahmed(201900928)]

[Fatma Al-zahraa AboElgheit Abdallah(201900559)]

[Fatma Al-zhraa Mostafa Mohamed Senosy(201900561)]

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence, at the Faculty of Computers & Artificial Intelligence, Helwan University

Supervised by: [Dr. Ahmed Elsayed]

Faculty of Computer science and Artificial intelligence.
Information systems department.

Team members:






- Table of content.

Acknowledgments

Here we are writing the last pages of our 4 years story, the journey was not easy, and we passed by many challenges, but what made all of this much easier is our **families** support and trust, we reached this point and was able to finish our essential 16 years of learning because of them, and we want to thank them from the bottom of our hearts.

We cannot forget our **great professors and teachers assistant** at the Faculty of Computers and Artificial Intelligence, Helwan University, without their help, support, and continuous effort, we would not have reached anything, so we would like to take this chance and thank all of them for being so supportive we also need to mention in particular our supervisor **Dr, Ahmed Elsayed** for the support and guidance making our idea come into light and directing our thinking into the right direction.

And finally, we must thank the **FCAI-HU community**, for being very collaborative and helpful. We are thanking every single person in this community.

Abstract:

- Describing the existing problem:

- Many people face daily problems in their homes. They are looking for the most skilled craftsmen with testimonials from customers , And they need excellent service at a good price.

- The craftsmen App nominates craft services (such as carpentry, and plumbing...) by identifying the problem facing people. The purpose of this application is to facilitate immediate access to professional services wherever you are, which helps to save time and effort.

- Lots of people need trusted craftsmen at their earliest and fastest point

- We aspire to create an application that connects the customer and the craftsman through the services we provide.

- Tools used:

- **Flutter**: is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web from a single codebase. And we are going to use the Flutter Maps library to view flutter maps in flutter apps, by adding googler_maps_flutter package.

- **Android Emulator**: The Android Emulator simulates Android devices on your computer so that you can test your application on a variety of devices and Android API levels without needing to have each physical device. The emulator provides almost all of the capabilities of a real Android device.

- **Firebase :** firebase provides a rich and intelligent code editor for firebase , extended code formatting configuration, on-the-fly error checking, and smart code completion. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Chapter One: Introduction

In this chapter we're going to discuss and go deeper in the overview of the project and know more about its scope, limitations and explain some terminologies we will find throughout the document.

Chapter One: Introduction

1.1 Overview:

It is a mobile application that allows users to find the nearest craftsman when they are facing a problem in their home. We sometimes find it difficult to find reliable craftsmen at good prices.

1.2 Objectives:

Our goal is to make it easy for the user to log into this mobile application and search for the nearest and best craftsmen. They can also book an appointment with him. While booking, they can choose the date, time and facilities they need to solve this problem.

1.3 Purposes:

Help users to communicate to get a job or to request a specific service, whether the user requests a service from a craftsman, such as repairing a car, delaying heavy equipment, purchasing car accessories, or requesting a carpentry or plumbing service and other services available to a user requesting a service.

1.4 Scope:

We planned and scoped our project on Microsoft project management software to dedicate the roles and time scheduling our project,7 months for the project includes planning, designing, coding, testing and documentation (all details on project file)

Planning:

▪ Collecting data about the project and the lack that made us in need of the Mobile Application.
▪ Determining the functional and non-functional requirements.
▪ Setting a Gantt chart for the project.

Designing:

▪ Determining the diagrams to be carried out within the project:
ERD Diagram - Activity Diagram - Class Diagram - Use-case Diagram - Sequence Diagram - Context Diagram - State Diagram.

Coding:

- The main supposed functions to be coded in this App are:
  Sign up – Sign in – Logout – Edit profile – Ranking– Send request – Accept / reject request – Chat – Feedback .

Testing:

- Functional Testing: Unit testing – Regression testing – Integration testing.

- Non-functional Testing: Performance testing – Street testing – Security testing.

Documentation:

▪ The documentation should mainly include these main chapters:

1. Introduction: includes an overview of the project and limitations
2. Project Planning: includes the tools and technologies as well as tasks and timeline plan
3. Project Requirements: includes the functional and nonfunctional requirements
4. Project Design: includes the diagrams
5. Project Implementation: includes user application, owner system and administrator system
6. Project Testing: includes testing types

1.5 General Constraints

▪ Tasks division which cannot be fair enough.
▪ Indiscipline like being late in delivering tasks or attending meetings.
▪ Hesitation, especially when it’s related to taking a serious step or learning a new technique.
▪ Learning new technologies may take much time so we should manage time well.
▪ Underestimating the objectives that may not lead to realistic function.

Chapter Two: Project “Planning and Analysis”

In this chapter we’re going to discuss and go deeper in how we plan the project and show the steps and the instructions that we’ve followed to plan the application

2.1 Project Planning

2.1.1 Feasibility study.

A Feasibility study is used to determine if a business or a specific project is achievable, so for determining the achievability of our project we'll go deeper in the following points:

1- Ranking by rating:

A ranking system is one of the promising techniques that can generate item recommendations from a huge collection of items based on users' preferences called rating-based ranking methods. In traditional recommendation systems, the degree of preference is measured by a rating score. For example, Collaborative Filtering (CF) based ranking algorithms predict the rating scores of unrated items. Given a database of users' past ratings on a set of items, CF can predict the ratings that a user would assign to the unrated items so that items can be recommended to the user by the predicted ratings in descending order.

2- Rating (flutter_rating_bar):

The app we are going to make contains a star rating bar. Users can rate one of our services on a scale from 0 stars to 5 stars.

A simple yet fully customizable rating bar for flutter which also include a rating bar indicator, supporting any fraction of rating.

3- Comment (Opinion Mining):

Create a comment box using flutter by importing a package called “package:comment_box/comment/comment.dart”.
By using it, the user will be able to give a review of the craftsmen and their work. We can also add other Features such as:

- Smooth Scrolling Supported.
- User image Supported.

★ **Fine-graine sentiment analysis:**
The most common use of opinion mining works to categorize comments and statements on a scale of opinion polarity.

4- Chat:

Create a chat using flutter by importing a package called “package:flutter_rating_bar/flutter_rating_bar.dart”.
By using it, the user will be able to give a review of the craftsmen and their work. by using:

- Sign-in .
- send request.
- Chat one-to-one with other users (send text, image, sticker).

2.1.2 Estimated Cost.

A cost estimate approximates the cost of a program, project or operation. The cost estimate is the product of the cost estimating process and our estimated cost for this project comes as following:

1. Meeting in a public workspace: (200 LE per each time).
2. Some resources and programs we used in our advertising and marketing:
    - Making advertisements (video, Brochures):1200LE.
    - Purchasing programs like:
        1. Adobe XD
        2. Adobe Photoshop
    - Obtaining a hosting server: 3000LE (per year).

2.1.3 Gantt chart

	Task Name	Duration	Start	Finish
1	**1 Project creation**	**151 days**	**Sun 10/2/22**	**Tue 5/2/23**
2	**1.1 Plan**	**9 days**	**Sun 10/2/22**	**Thu 10/13/**
3	1.1.1 initial	4 days	Sun 10/2/22 8:	Wed 10/5/22
4	1.1.2 final	4 days	Sun 10/9/22 8:	Wed 10/12/22
5	1.1.3 Finish Plan	1 day	Thu 10/13/22 8	Thu 10/13/22
6	**1.2 Design**	**15 days**	**Sun 10/16/22**	**Thu 11/3/2**
7	1.2.1 UML diagram	8 days	Sun 10/16/22 8	Tue 10/25/22
8	1.2.2 DB	8 days	Mon 10/24/22 8	Wed 11/2/22
9	1.2.3 ERD	9 days	Mon 10/24/22 8	Thu 11/3/22 5
10	**1.3 implementation**	**96 days**	**Sun 1/1/23 8:**	**Thu 3/30/2**
11	1.3.1 front end	3 wks	Sun 1/1/23 8:0	Thu 1/19/23 5
12	1.3.2 Back end	5 wks	Sun 1/22/23 8:	Thu 2/23/23 5
13	1.3.3 ML	1 mon	Sun 2/26/23 8:	Thu 3/23/23 5
14	1.3.4 Finish implementation	1 wk	Sun 3/26/23 8:	Thu 3/30/23 5
15	**1.4 Test**	**30 days**	**Sun 4/2/23 8:**	**Tue 5/2/23**
16	1.4.1 unit testing	9 days	Sun 4/2/23 8:0	Wed 4/12/23
17	1.4.2 Integrated Testing	11 days	Thu 4/13/23 8:	Thu 4/27/23 5
18	1.4.3 Additional Testing	3 days	Sun 4/30/23 8:	Tue 5/2/23 5:
19				
20	2 Deployment	15 days	Wed 5/3/23 8:0	Tue 5/23/23 5
21	3 Maintance	6 days	Wed 5/24/23 8:	Wed 5/31/23

2.2 Analysis and Limitation of existing systems

Logo	Name	Free	Bugs	In Egypt
	Elherafyeen	True	False	True
	CRAFTSMAN myQ	False	True	False
	Craftnote	False	True	False
	CRAFTSMAN	True	False	True
	Primo	True	True	True

All of these systems do the same thing. You can easily login, select your location and then start connecting with your craftsman.
Every system has some problems:

1- Elherafyeen.

- It does not work for all trades.

2- CRAFTSMAN myQ.

- This system has some problems while registering.
- Works only for the US

3- Craftnote .

- This system is so slow, it works only for Russia.
- Requires a login every time you open it.

4- CRAFTSMAN.

- Includes some bugs while booking.
- Loading time is much too long and almost tiresome.

2.3 Need for the new system

Why are we implementing the Craftsmen system? Here are the reasons.

1- Other applications are located outside Egypt.

2- He will find the right craftsman, and avoid
Errors that users of other applications complain about.

3- Customer satisfaction without any problem.

4- From the home screen, users can easily select a
craftsman and book an appointment, making it easier for users to take the actions that are most important to them unlike other existing apps that have
Complexity in the user interface.

5- In this app we provide two way messages for users to
create New contacts with the chosen craftsman, which can be Accessed through the work app.
The conversation starts with one Swipe to see details and news before booking And the agreement

2.4 Analysis of a new system.

2.4.1 user requirements.

Mandatory requirements:

1. Users must be able to register, sign in and out to create his profile.
2. User needs to choose his problem section.
3. Users need to choose a craftsman.
4. Users need to chat with craftsmen.
5. Users need to give him feedback.

Desirable requirements:

1. Users need to be connected to the internet while booking.

2. Users use the application at any time and anywhere.

2.4.2 system requirements.

1. **Internet:**
   The Application must have a good connection with the internet to access it well.

2. **Mobile phone:**
   The application works only with mobiles.

2.4.3 domain requirements.

1- Multiple users must be able to use the application simultaneously without corrupting the database (whatever form it may be).

2- A server must be set up to host the database, and the server must be accessible by all the systems running the inventory tracking software.

3- The database should be backed up every once in a while, in case the original does become corrupt.

4- application must verify all values before making a change in the database.

5- application must have updated capabilities for future versions.

2.4.4 functional requirements.

- Users Requirements [UR].

Requirement ID	Name	Requirement description
-U	sign in	The system must allow users to login with their information.
-U	sign up	The system must allow users and craftsmen to register.
-U	delete account	The system allows the user to delete his account.
-U	password reset	if the user forgets his password he can change to a new one.
-U	edit profile	The system allows the user to edit profile data.
-U	stream notification	The system allows users to send and accept notifications to craftsmen.
-U	send the request	The system allows the user to send requests to the selection craftsman.
-U	view craft work	The system allows the user to see craftsmen work.
-U	feedback	The system must allow users to make their feedback accordingly.
-U	message	The system must allow users to communicate with the craftsman.
-U	logout	The system must allow users to logout the system.

- Craftsmen Requirements [CR].

Requirement ID	Name	Requirement description
-C	sign in	The system must allow craftsmen to login with their information.
-C	sign up	The system must allow craftsmen to register with all their information.
-C	delete account	The system allows craftsmen to delete his account.
-C	password reset	If a craftsman forgets his password he can change to a new one.
-C	edit profile	The system allows the craftsmen to edit profile data.
-C	stream notification	The system allows craftsmen to send and accept notifications to users.
-C	add work	The system allows craftsmen to add his own work.
-C	delete work	The system allows craftsmen to delete his own work.
-C	view the request	The system must allow craftsmen to view the request which is sent by the user and accept or reject him.
-C	message	The system must allow craftsmen to communicate with the user.
-C	view the feedback	The system allows craftsmen to view the feedback which is sent by the user.
-C	logout	The system must allow craftsmen to logout the system.

2.4.4 non-functional requirements.

❖ Usability Requirement:

- Mobile app usability makes it easy for the user to become familiar with the user interface (UI).
- Application must warn the user if he entered his credentials wrong.
- GUI must have an easily navigated map.
- Users can sometimes take actions within an app that they didn't intend to take.
- When a user makes a mistake, Mobile apps need to support undo and redo functions.
- The application will help the client in choosing craftsmen in an easy way.

❖ Reliability Requirement:

- Craftsmen information must be up to date.
- Navigation must be updated and reliable.
- Data stored in and retrieved from the database must be the same.
- The system must be up and running 24/7.

❖ Performance Requirement:

- The system should startup in less than 6 sec.
- Login authentication should take less than 10 sec.
- After confirmation navigation map should take less than 8 sec

❖ Scalability Requirement:

- The app should be capable enough to handle almost 1000 users simultaneously without affecting its performance.

❖ Availability Requirement:

- System functionalities must be always available.
- All the craftsmen near the desired destination must be all shown on the map.

❖ Data integrity:

- Craftsman information must be up to date and validated.
- User Information must be up to date and validated.
- Routes Database must be up to date and validated.

❖ Maintainability Requirement:

- The System must be implemented adhering to OOP principles to ensure easy maintenance.
- SOLID principles also keep our code cleaning and easy to understand.

❖ Flexibility Requirement:

- The System must be implemented adhering to OOP principles to ensure easy change.

2.5 advantages for the new system.

1-Users can easily register, sign in and out.
2-Users can have a complete profile with their details.
3- Helps people to find what they want.
4- Could handle more than 1000 requests per day for every craftsman in the system.
5- Have a high accuracy to find the nearest craftsman for the user.
6- Have a speed search with specific location and area for user needs .
7- Simple navigation for user on Mobile app.

2.6 Risk and risk management.

- The process of risk:

Risk identification:

Risk type	Possible risks
Technology	New technologies for the team lead to a lack of knowledge. Software components that should be reused contain defects that limit their functionality.
People	It is impossible to recruit staff with the skills required. Key staff are unavailable at critical times. Required training for staff is not available.
Organizational	The organization is restructured so that different management is responsible for the project
Tool	Tool can’t integrate the project efficiently
Requirement	Change requirements and add new features after work on something that loses time and cost.
Estimations	The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated.

Risk analysis:

Risk	Probability	Effects
It is impossible to recruit staff with the skills required for the project.	High	Catastrophic
Key staff are ill at critical times in the project.	Medium	Serious
Changes to requirements that require major design rework are proposed.	Medium	Tolerable
Software components that should be reused contain defects which limit their functionality.	High	Serious
Customers fail to understand the impact of requirements change	Medium	Serious

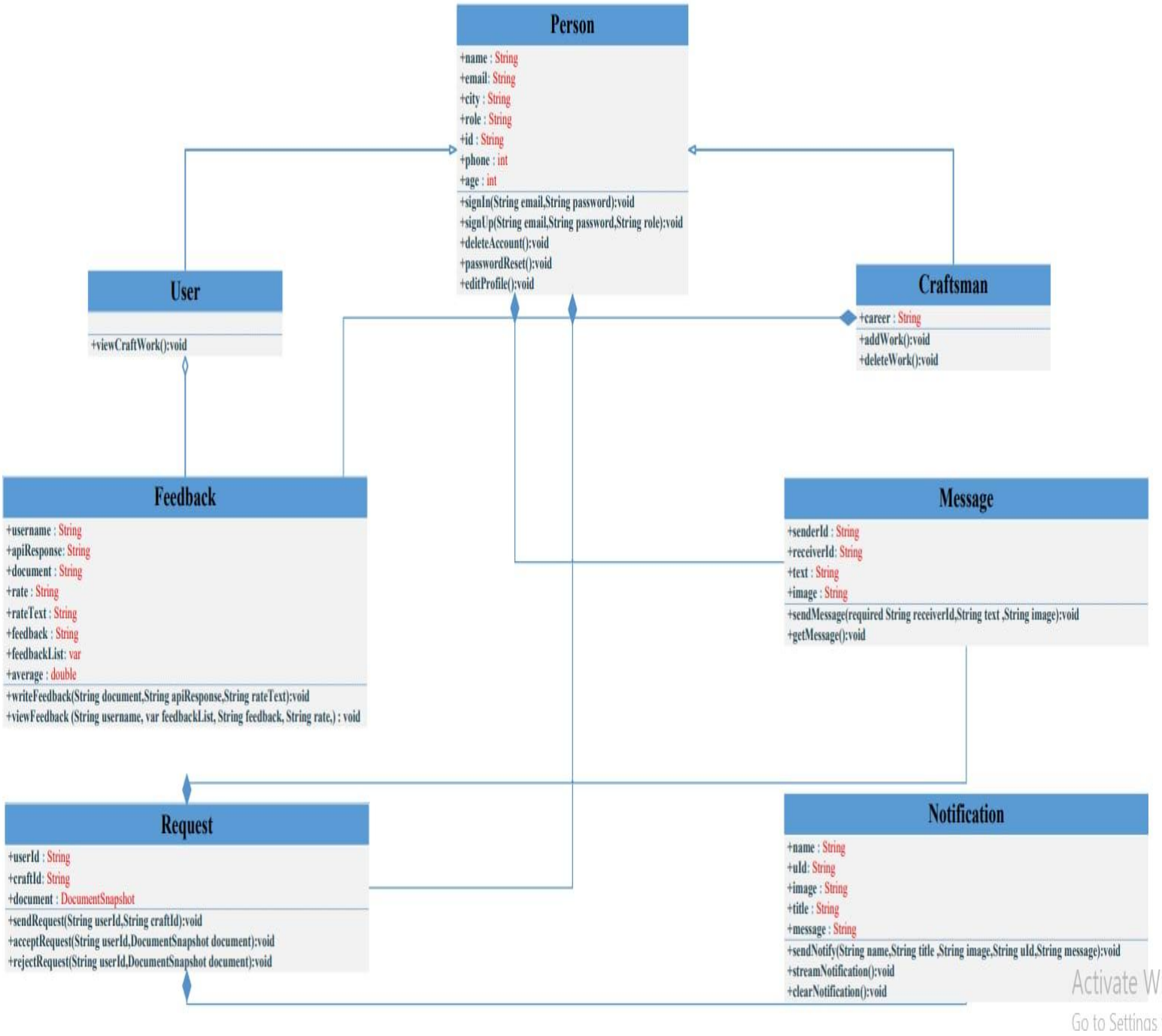
Risk planning:

Risk	Strategy
Recruitment problem	Alert customer of potential difficulties and the possibility of delays, investigate buying- in components
Staff illness Reorganize	the team so that there is more overlap of work and people therefore understand each other’s jobs.
Database performance	Investigate the possibility of buying a higher performance database.
Defective components	Replace potentially defective components with bough tin components of known reliability

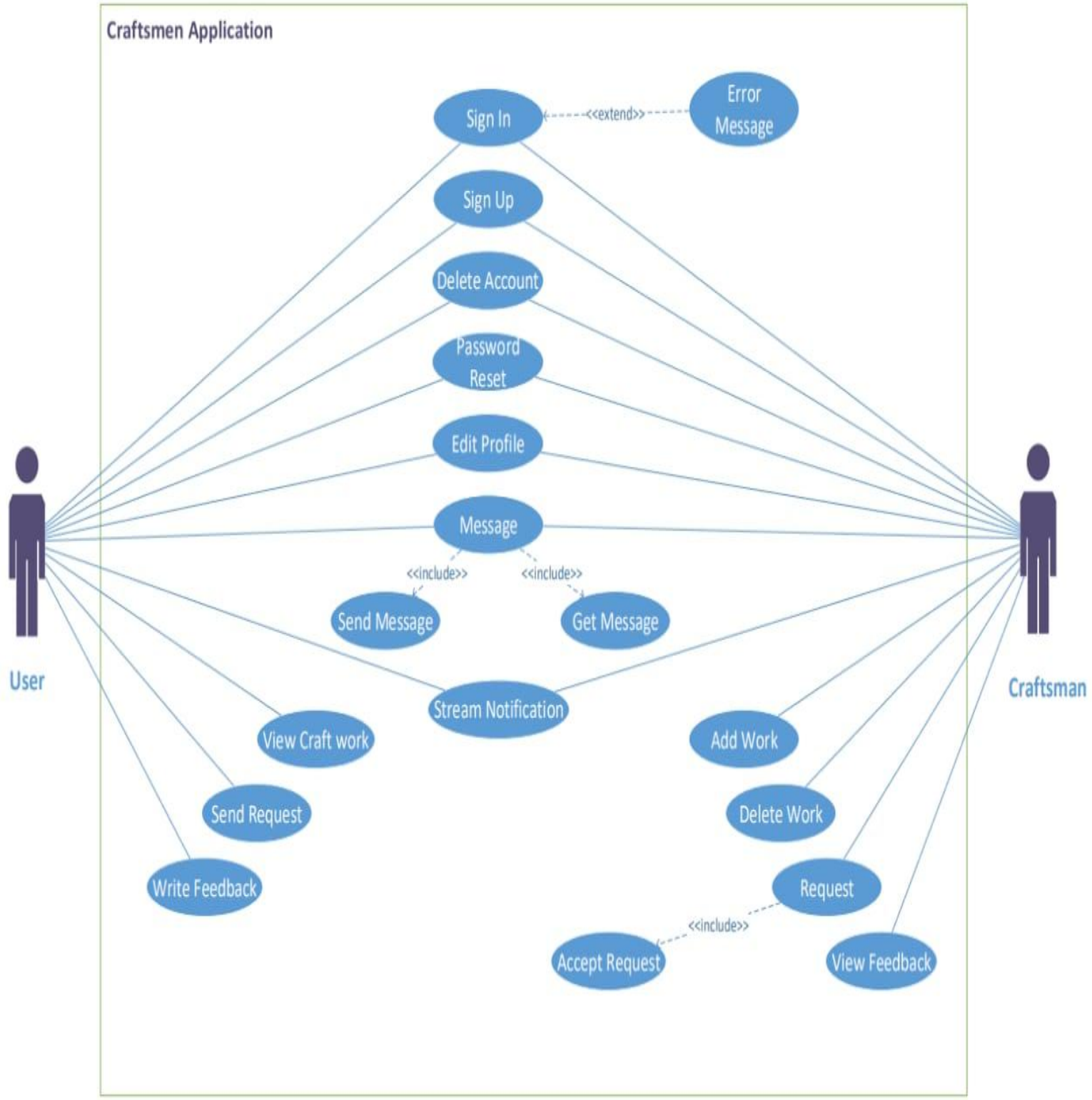
Chapter Three: : Project “Software Design”

In this chapter we’re going to discuss and go deeper in workspace application’s design and present its diagrams and database.

3.1 Design of database. (Class diagram).



3.2 Use case diagrams.



Use-Case scenario

1-Sign in scenario.

Use case Name	sign in.
Use case ID	1
Goal	Actor enters the system Successfully.
Pre-condition	The actor has a valid account.
Post-condition	The system takes me to the main account.
Main Success Scenario	1. Actor chooses to make a Login. 2. System requests the actor to enter his login details. 3. Actor enters his login details. 4. System validates the login details. 5. System logs the actor into the system.
Non Success Scenario	1. Actor enters invalid login details. 2. The System displays an error Message. 3. The system asks the actor to enter his register details again and make all steps in the success scenario again.

table 1 _ Scenario about actors sign in.

2-Sign up scenario.

Use case Name	Sign up
Use case ID	2
Goal	The actor makes an account on the system.
Pre-condition	Noun.
Post-condition	The actor has a valid account on the system.
Main Success Scenario	1. The actor chooses to make a register. 2. The system asks him to enter his register details. 3. Actor enters his register details. 4. System checks if it is valid or not. 5. If it is valid. 6. System creates an account for the actor.
Non Success Scenario	1. Actor enters invalid register details. 2. The System displays an error Message. 3. The system asks the actor to enter his register details again and make all steps in the success scenario again.

table 2_ Scenario about the sign up.

3-Logout scenario.

Use case Name	Logout.
Use case ID	3
Goal	Users and craftsmen can logout successfully.
Pre-condition	User has already made a login.
Post-condition	The actor closes an account on the system.
Main Success Scenario	1. Actor login system. 2. After successful login, click the Logout button This should disable access to the account, unless another successful Login is done.

table 3_ Scenario about logout.

4- Message scenario.

Use case Name	message
Use case ID	4
Goal	Users and craftsmen can communicate easily.
Pre-condition	User chooses craftsmen he needs.
Post-condition	Quickly answer the questions of users and craftsmen.
Main Success Scenario	1. User chooses craftsmen he needs. 2. User chooses to make a request. 3. User clicks the button to send a request. 4. System sends this request to the craftsman. 5. The craftsman accepts a request. 6. Users and craftsmen can communicate easily.

table 4_ Scenario about chat between users and craftsmen .

5-Send request scenario.

Use case Name	Send request.
Use case ID	5
Goal	The user sends a request to the craftsman.
Pre-condition	The user already displays the problem.
Post-condition	The user sends a request to the craftsman.
Main Success Scenario	1. User chooses craftsmen he needs. 2. User chooses to make a request. 3. User clicks the button to send a request. 4. System sends this request to the craftsman. 5. The craftsman accepts a request.

table 5_ Scenario about sending a request to the craftsman.

6-Delete account scenario.

Use case Name	Delete account
Use case ID	6
Goal	The user/craftsman deletes the account in the system.
Pre-condition	The user/craftsman already signed in.
Post-condition	The useruser/craftsman deletes the account successfully.
Main Success Scenario	1. After the user/craftsman enters the homepage . 2.Select button to delete account.

table 6_ Scenario about delete account.

7-Write feedback scenario.

Use case Name	Write feedback.
Use case ID	7
Goal	Actors write feedback about craftsmen.
Pre-condition	The user already has the service from a craftsman.
Post-condition	The feedback is saved into the system.
Main Success Scenario	1. After the user receives the service. 2. System requests him to enter his feedback. 3. User clicks on the button to add a comment. 4. User clicks on the send button. 5. System display thank you for your feedback.

table 7_ Scenario about writing feedback.

8- Add a work scenario.

Use case Name	Add work.
Use case ID	8
Goal	craftsmen add his work details.
Pre-condition	Craftsman has already made a login.
Post-condition	craftsmen add to the homepage his work easily.
Main Success Scenario	1. Craftsman login system. 2. Craftsman clicks the button my work 3. Craftsman adds his work picture. 4. click on the button ok.

table 8_ Scenario about adding business Info.

9-View request scenario.

Use case Name	View request.
Use case ID	9
Goal	The craftsman sees the requests that are sent to him at the same time to accept or reject .
Pre-condition	The craftsman is active and available.
Post-condition	Craftsman accepts the request or egnore.
Main Success Scenario	1. Craftsman login system. 2. Craftsman opens the request box. 3. Craftsman chooses to accept or ignore.

table 9_ Scenario about view request.

10-Delete work scenario.

Use case Name	Delete work.
Use case ID	10
Goal	craftsmen delete his work details.
Pre-condition	Craftsman has already made a login.
Post-condition	craftsmen delete his work eaisly.
Main Success Scenario	1. Craftsman login system. 2. Craftsman clicks the button my work 3. Craftsman deleted his work picture. 4. click on the button ok.

table 10_ Scenario about view rate.

11-View feedback scenario.

Use case Name	View feedback.
Use case ID	11
Goal	Craftsman sees feedback that users added.
Pre-condition	The craftsman is active and available.
Post-condition	Craftsman improves himself.
Main Success Scenario	1. Actor login system. 2. Craftsman see rate when open home page.

table 11_ Scenario about view feedback.

12-Edit Profile scenario.

Use case Name	Edit profile.
Use case ID	12
Goal	Craftsman/user can edit his profile data .
Pre-condition	User has already made a login.
Post-condition	update craftsman/ user profile.
Main Success Scenario	1. After the craftsman/ user login system. 2. Craftsman/user enter his profile. 3. they select button edit. 4. click ok to save new data.

table 12_ Scenario about edit profile.

13-Stream Notification scenario.

Use case Name	stream notification.
Use case ID	13
Goal	Craftsman/user can send or receive request notifications.
Pre-condition	Users send requests.or craftsmen accept or reject requests.
Post-condition	Craftsman/user can send or receive request notifications.
Main Success Scenario	1. After the craftsman/ user login system. 2. user send request to craftsman so send notify. 3. craftsmen accept or reject so send a notification.

table 13_ Scenario about stream notification.

14-View craft work scenario.

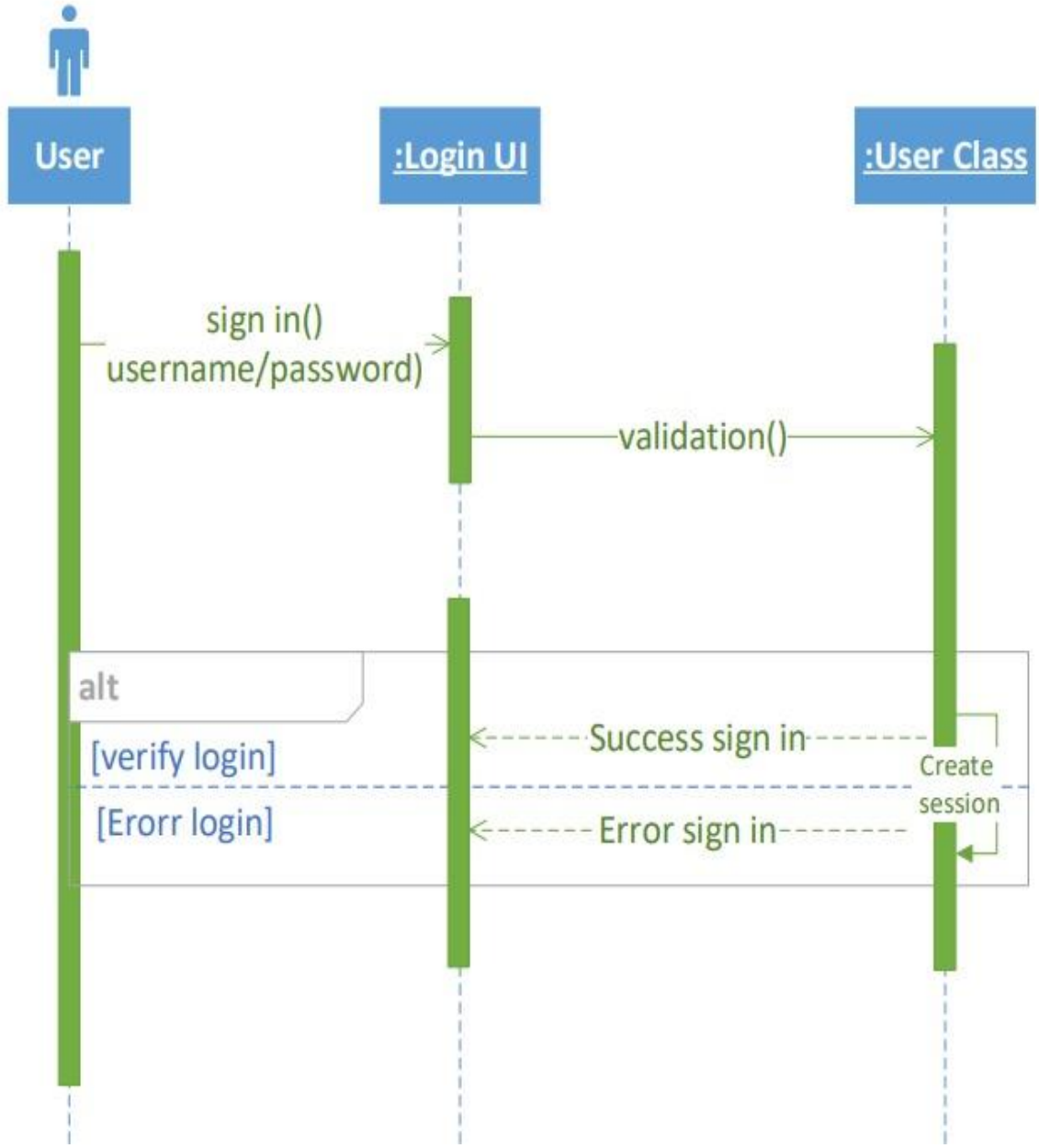
Use case Name	View craft work.
Use case ID	14
Goal	the user can see crats man work .
Pre-condition	User has already sent a request to the craftsman .
Post-condition	the user can see crats man work.
Main Success Scenario	1. After the user login system. 2. user select craftsman to send request. 3. craftsmen accept the request. 4. user open craftsman page to see his work.

table 14_ Scenario about View craft work.

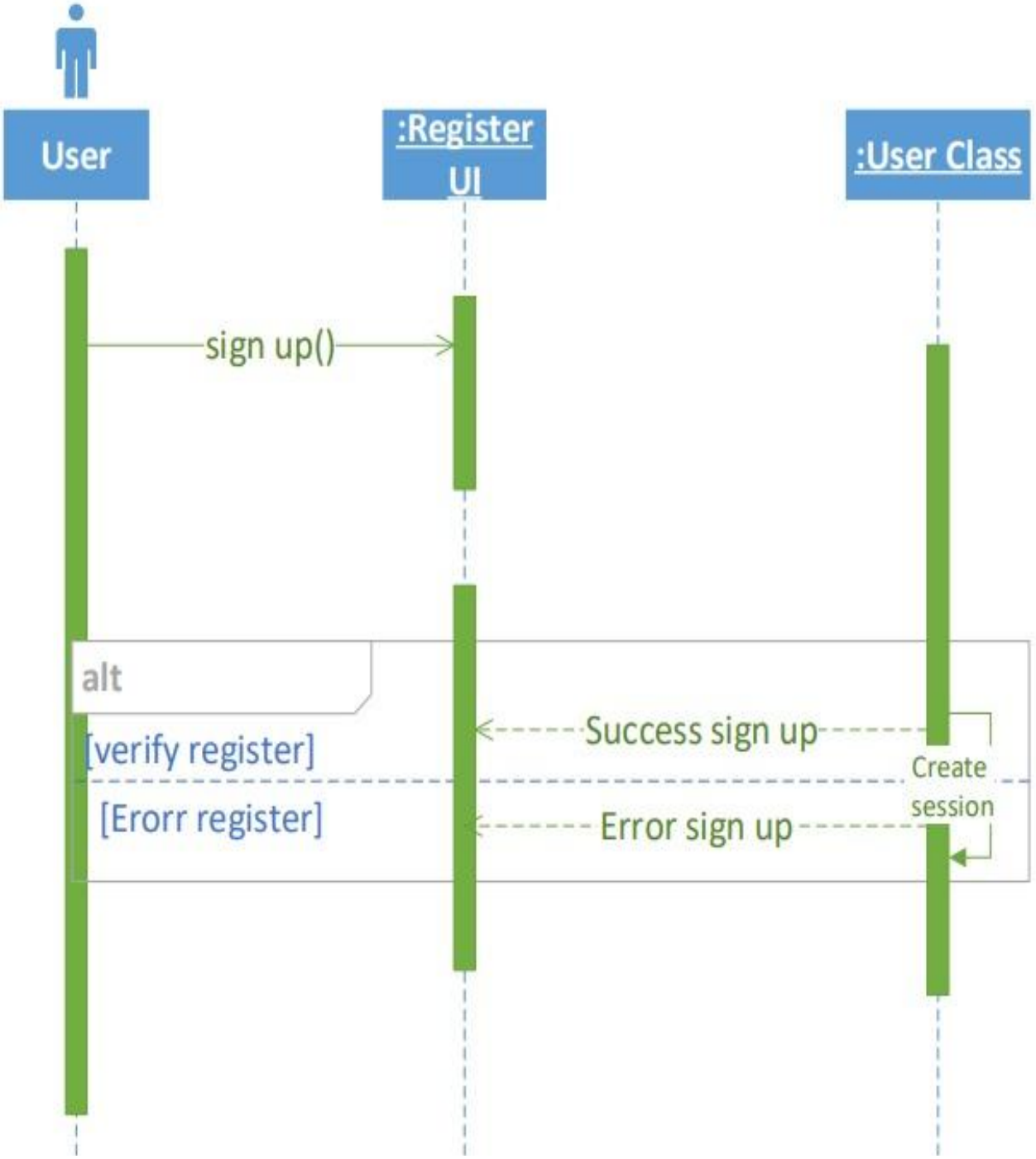
3.3 sequence diagrams.

User

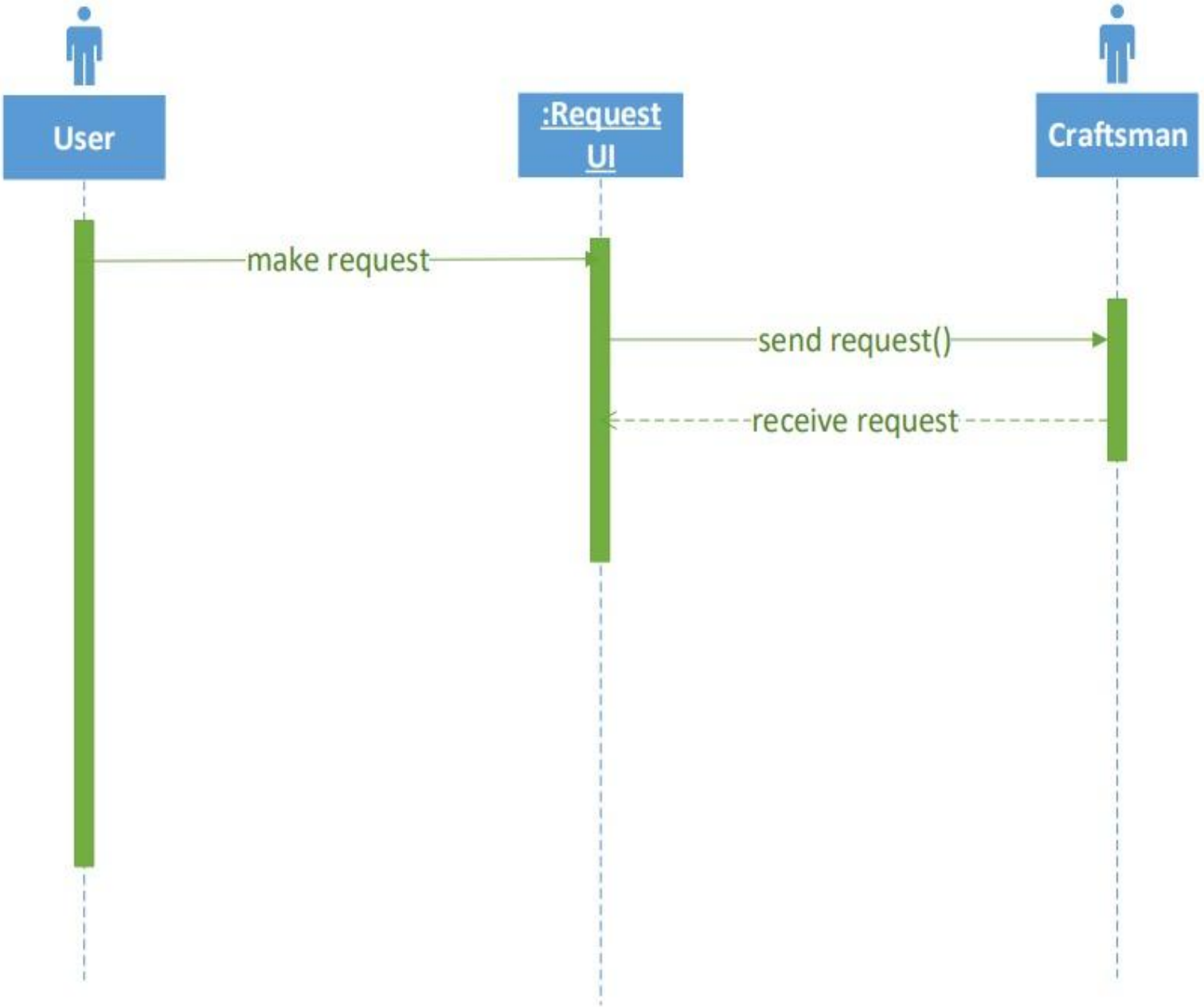
User sign in.



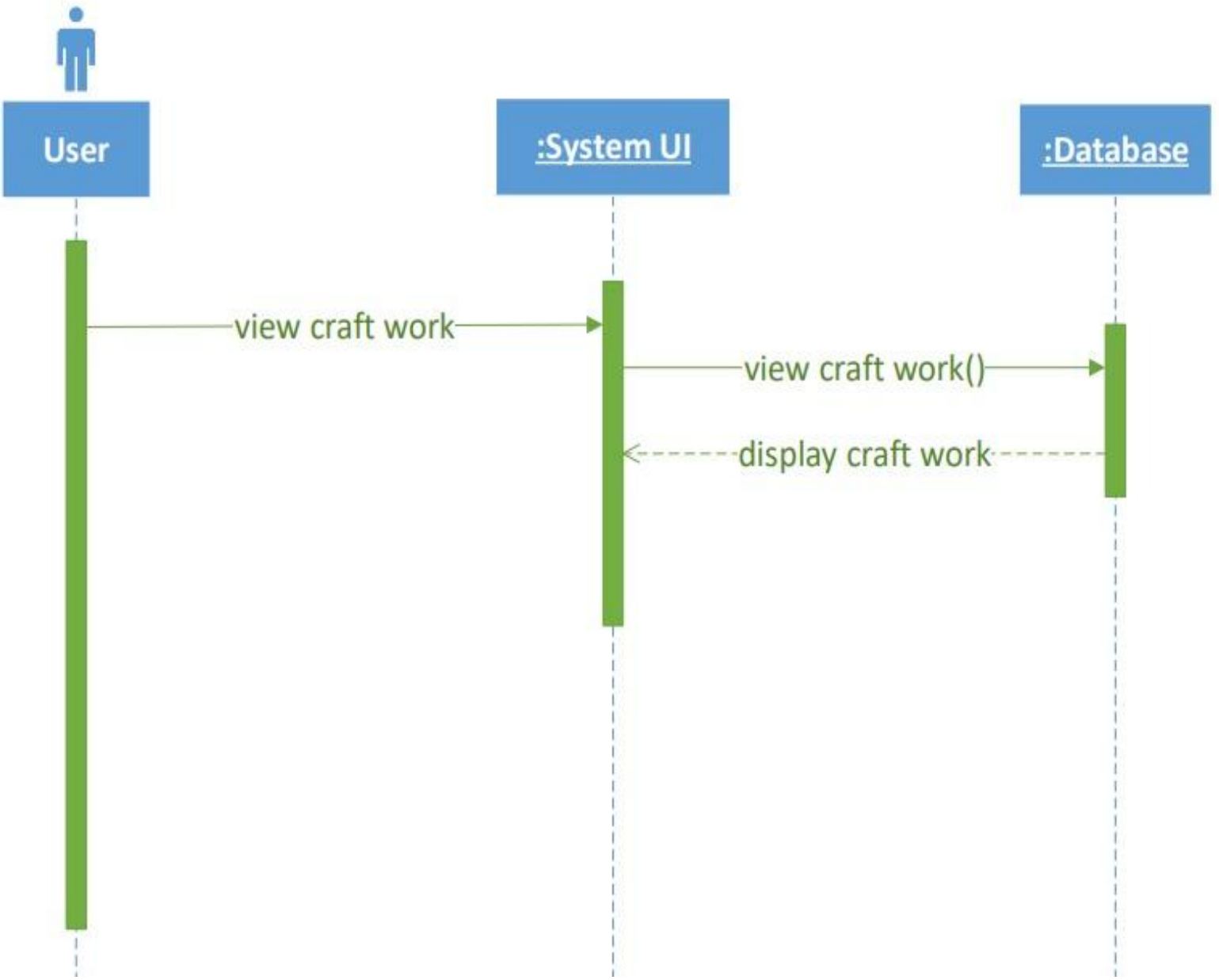
User sign up.


User
:Register UI
:User Class
sign up()
alt
[verify register]
Success sign up
Create session
[Erorr register]
Error sign up

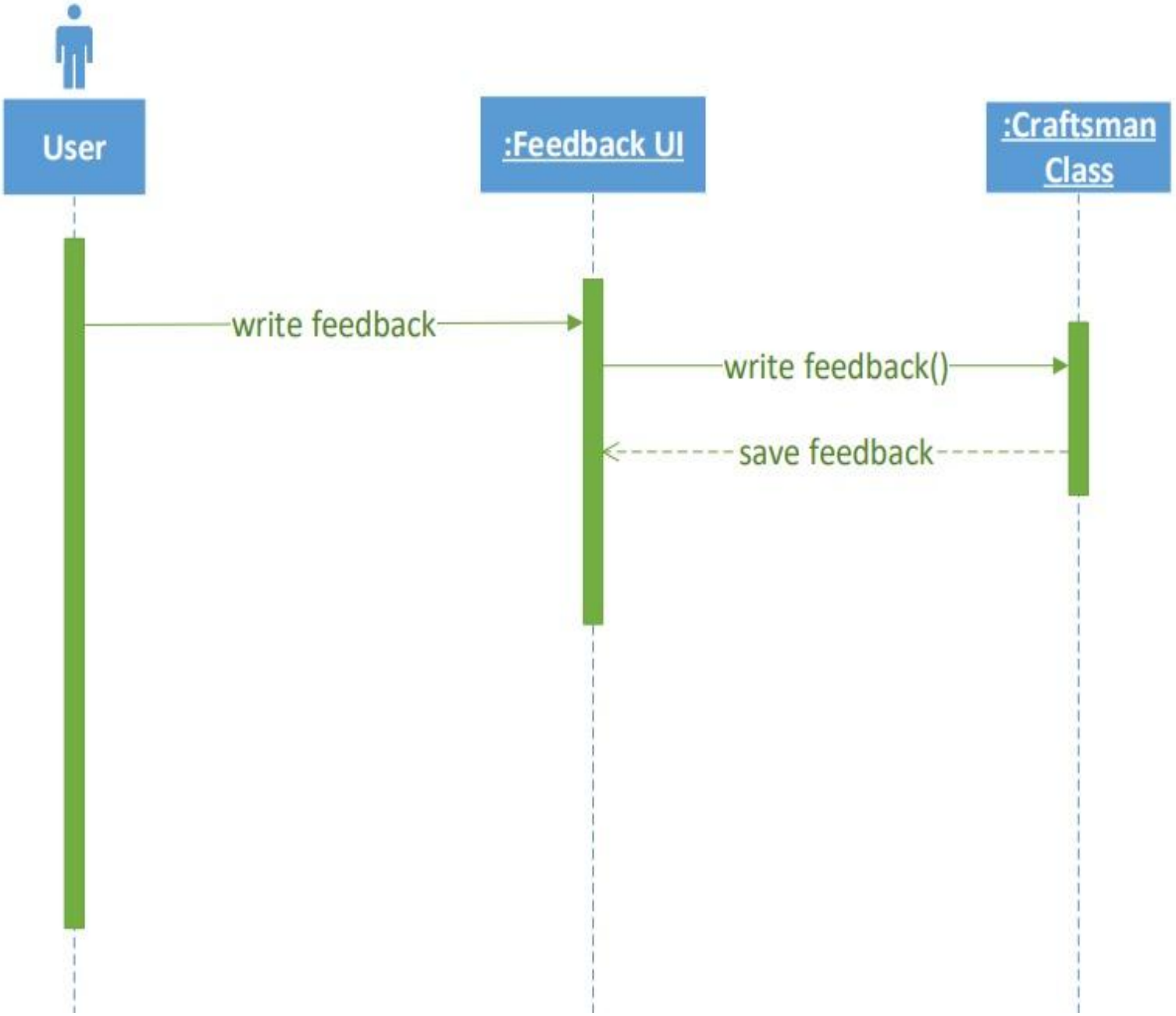
User send request.



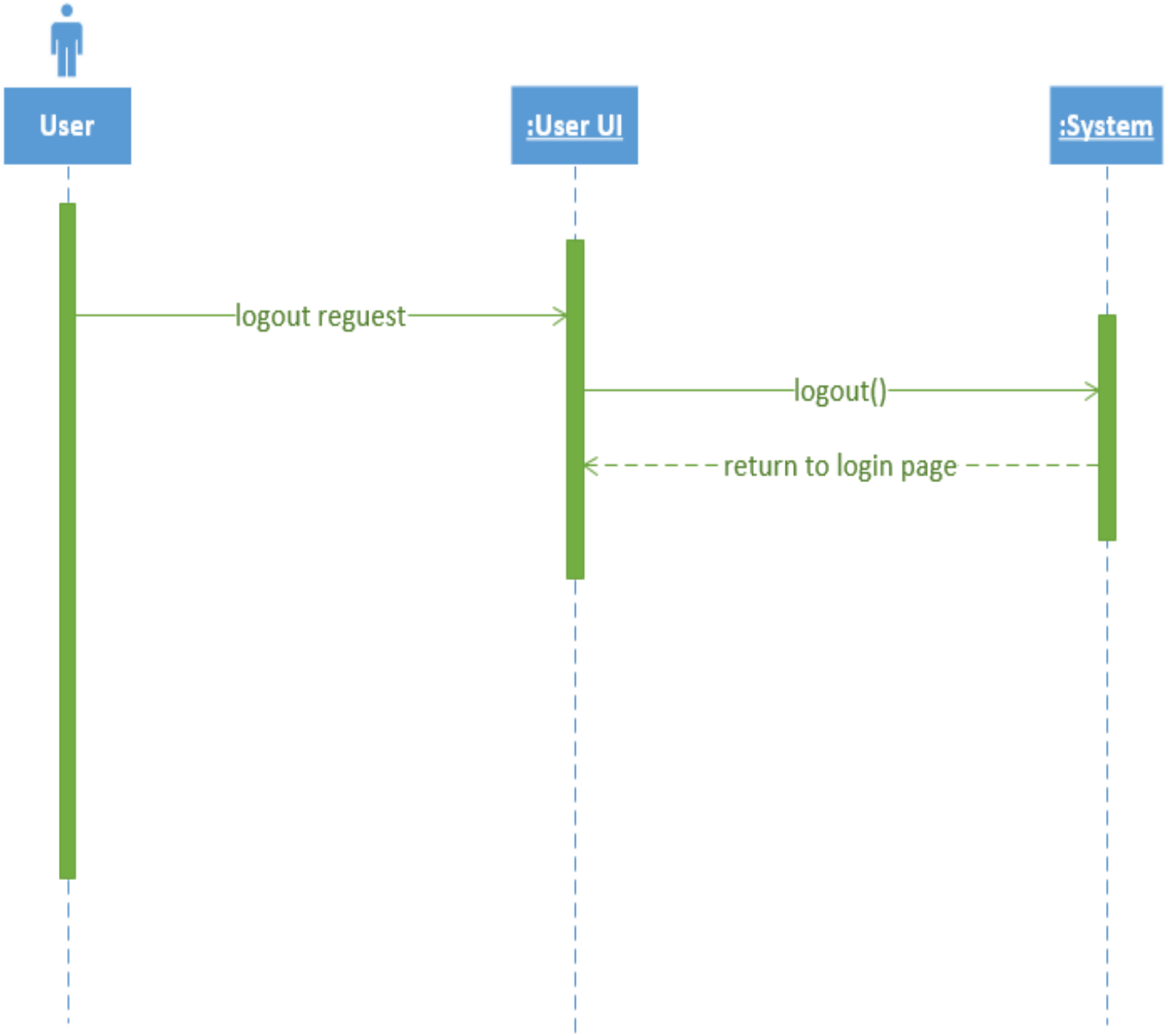
User view craft work .



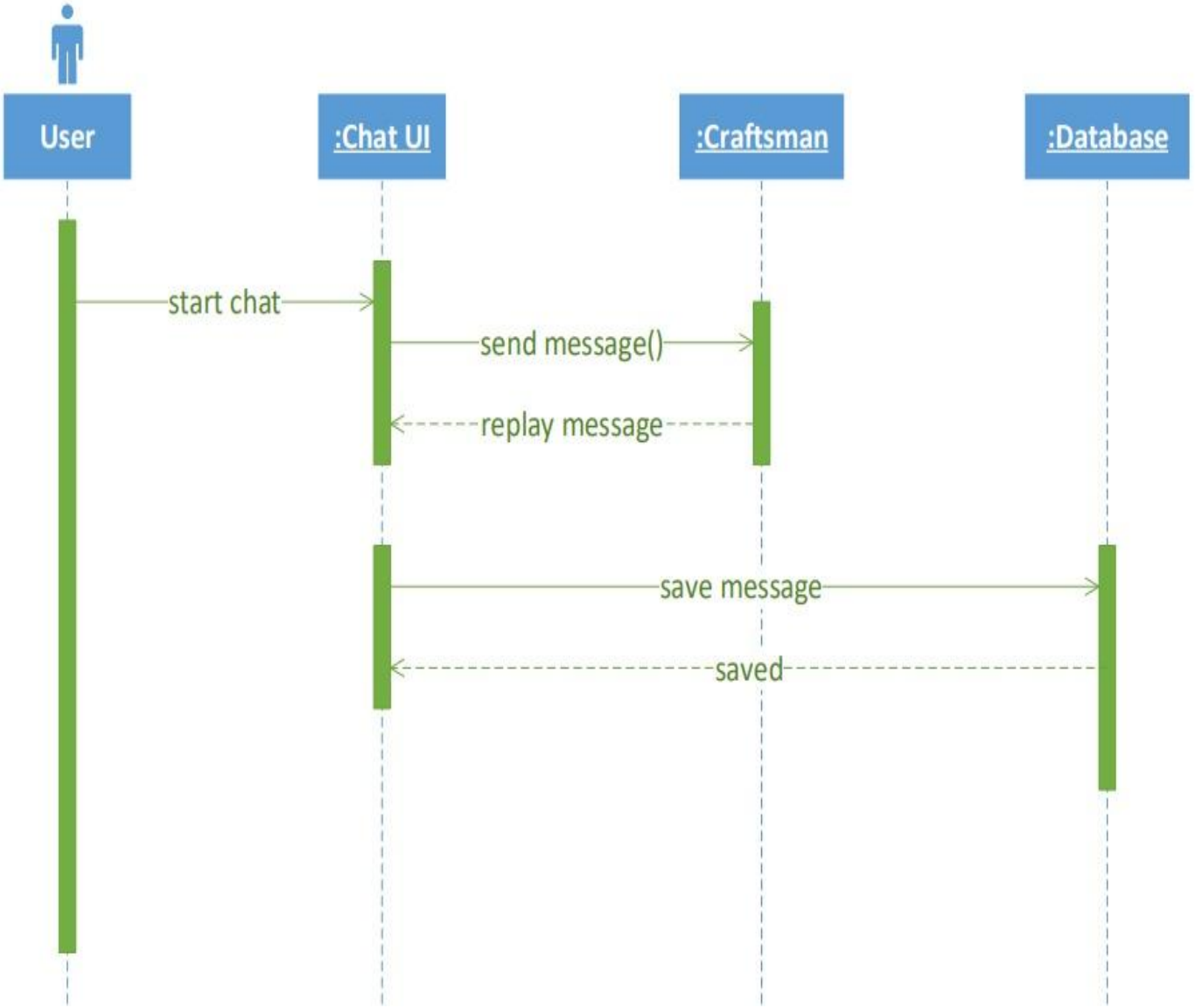
User writes feedback.



User logout.

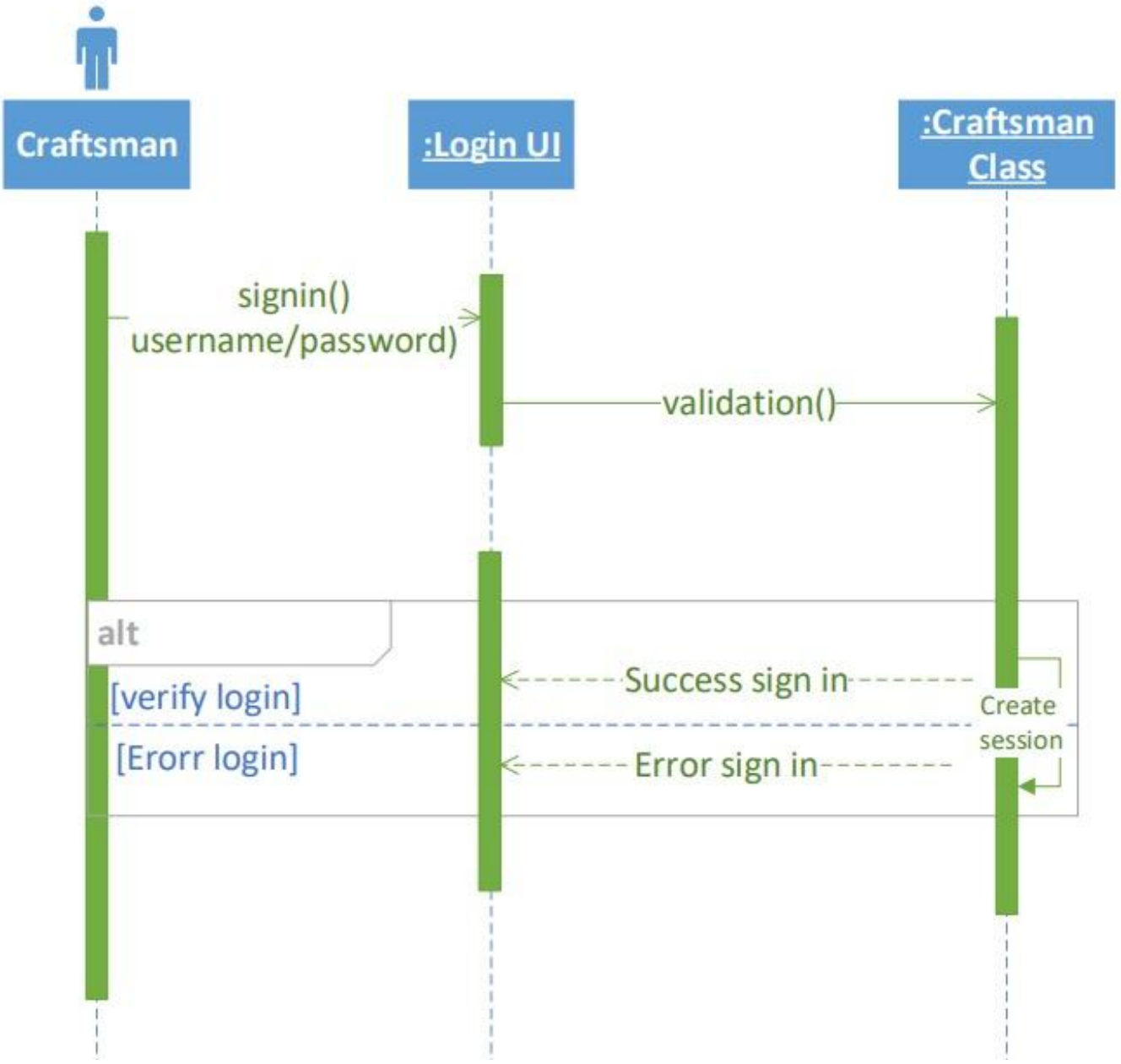

User
:User UI
:System
logout reguest
logout()
return to login page

Chat between user and craftsman.

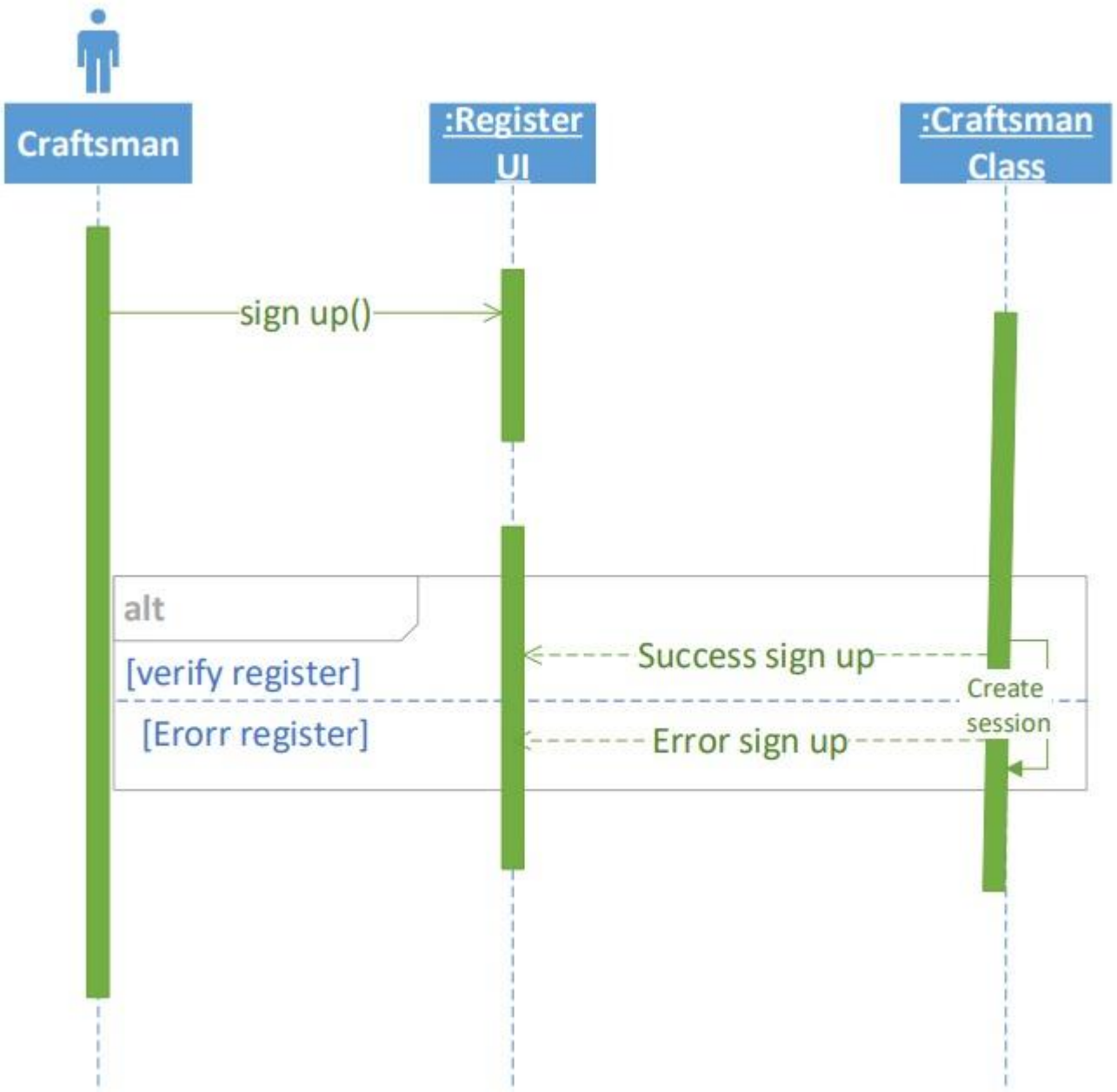


Craftsman

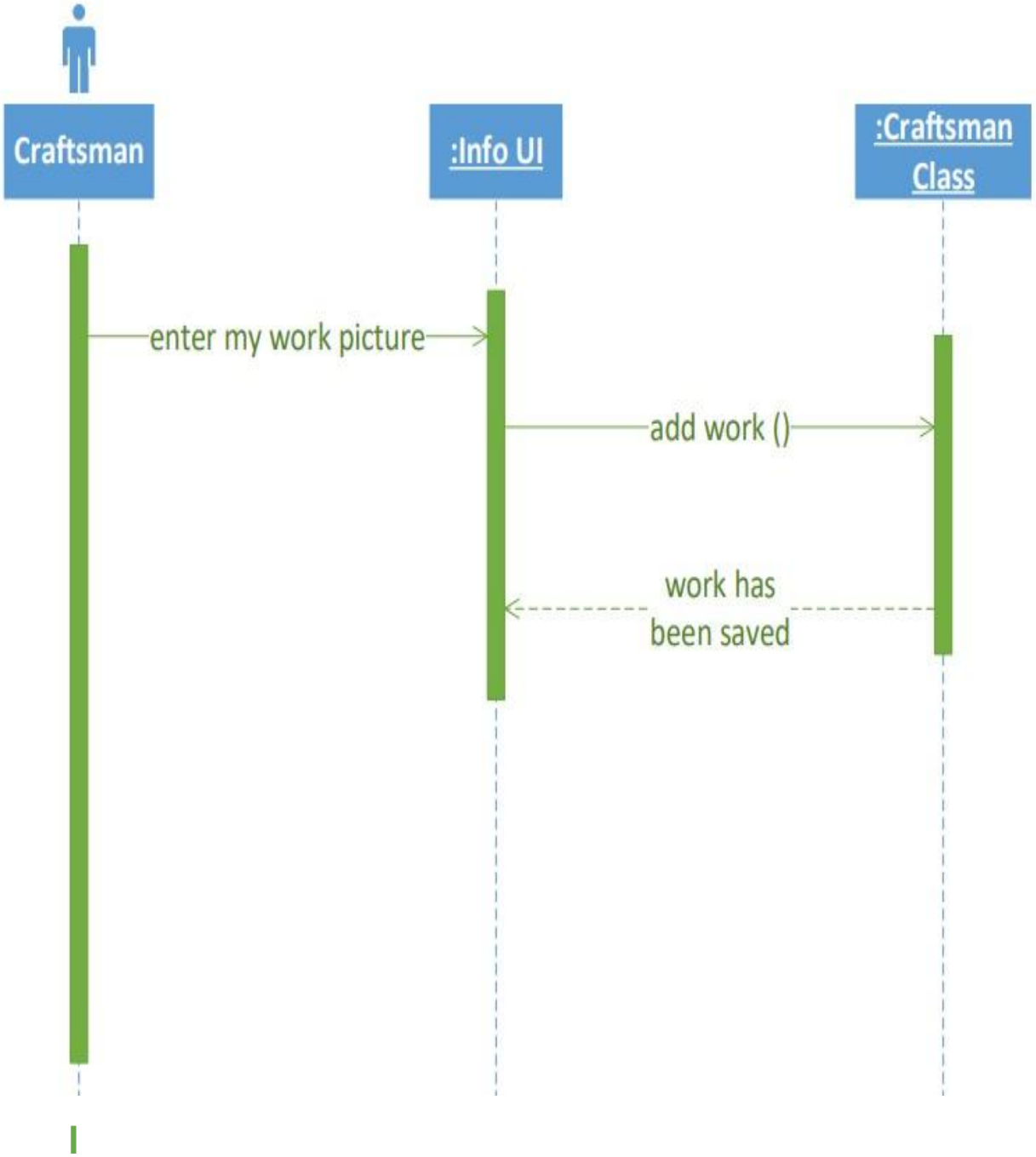
-Craftsman login.



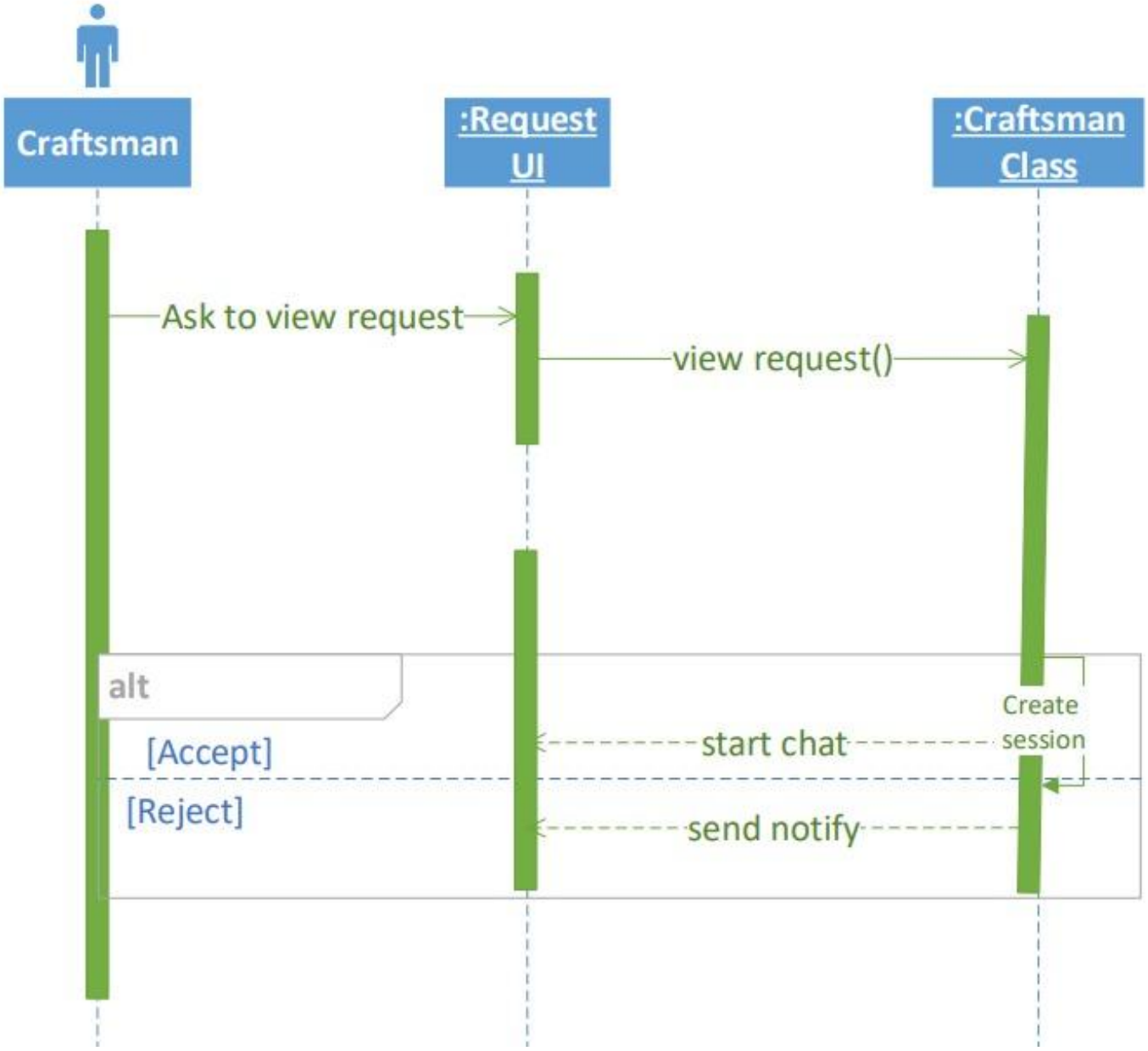
-Craftsman Register.



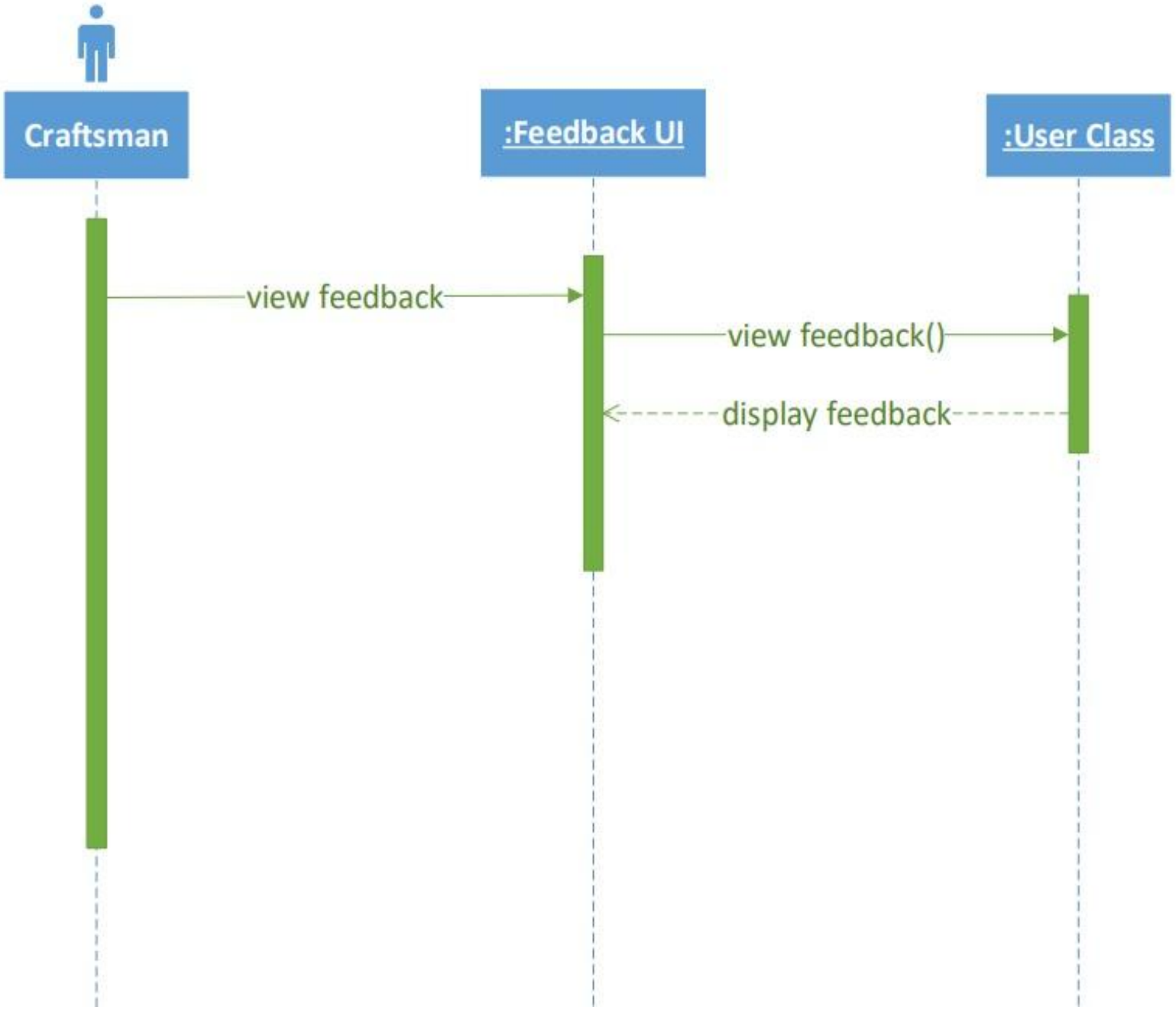
Craftsmen add business info.



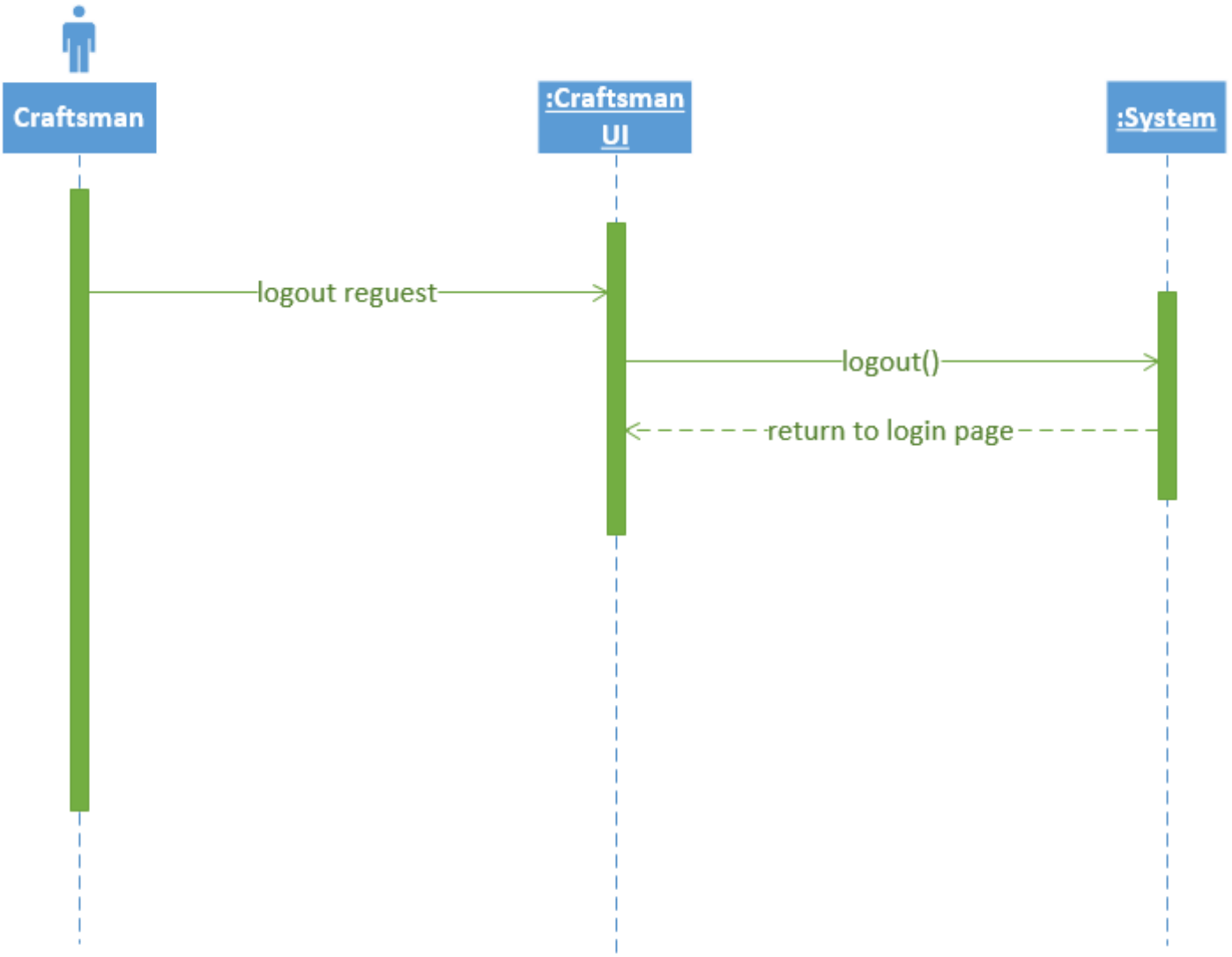
Craftsman view request.



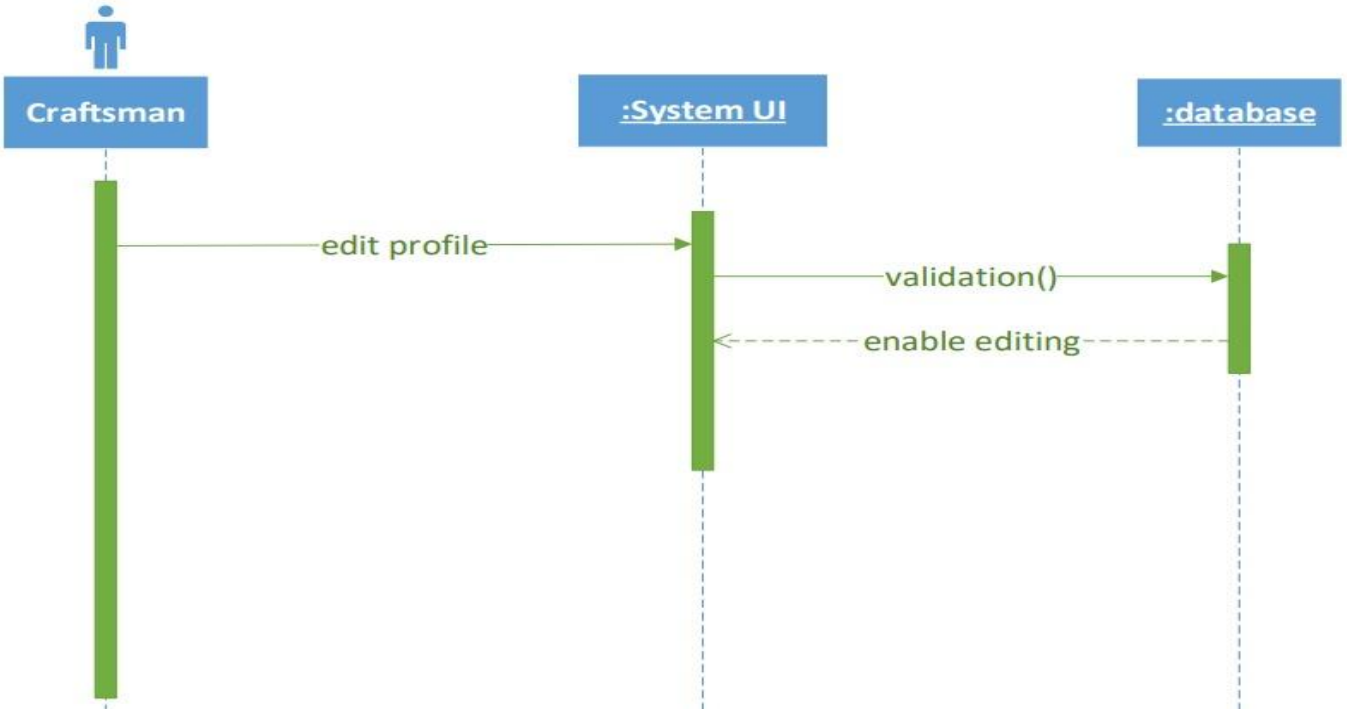
Craftsmen view feedback and rate.



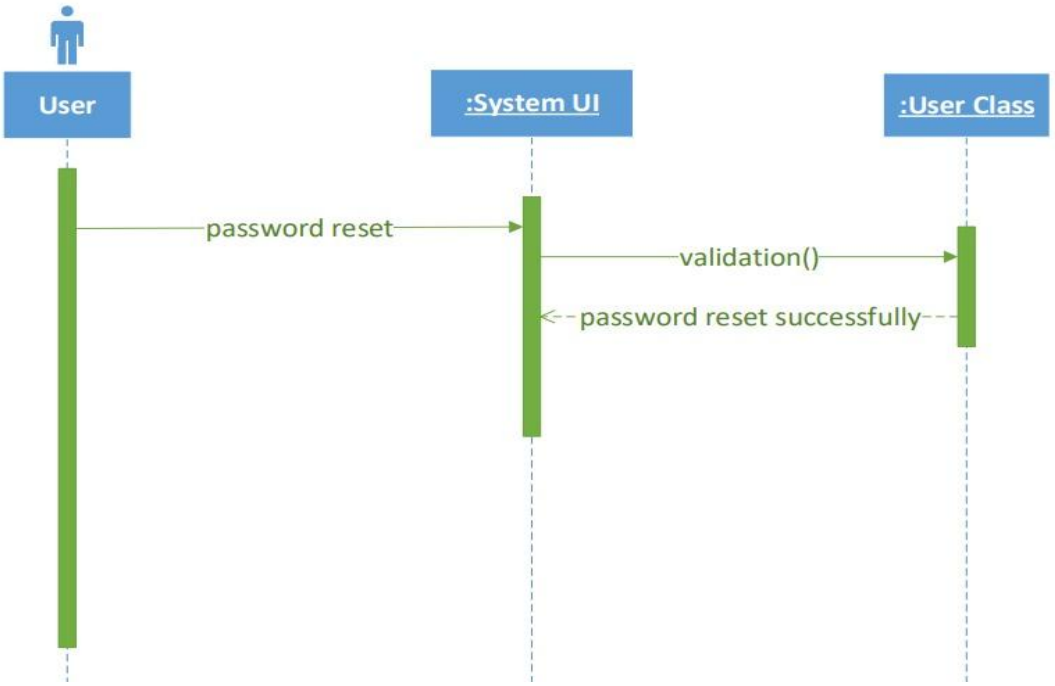
Craftsman logout.



person edit profile.

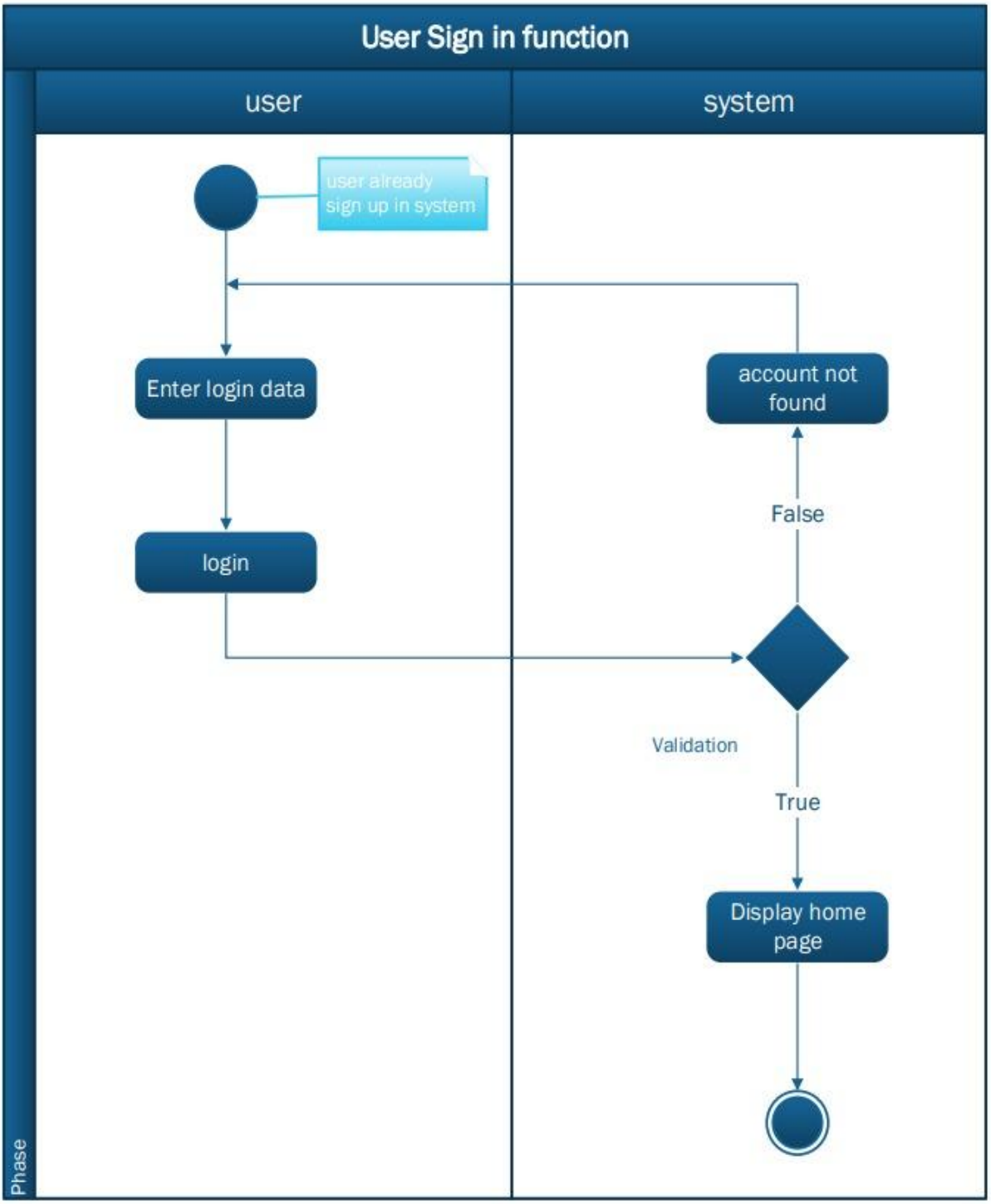


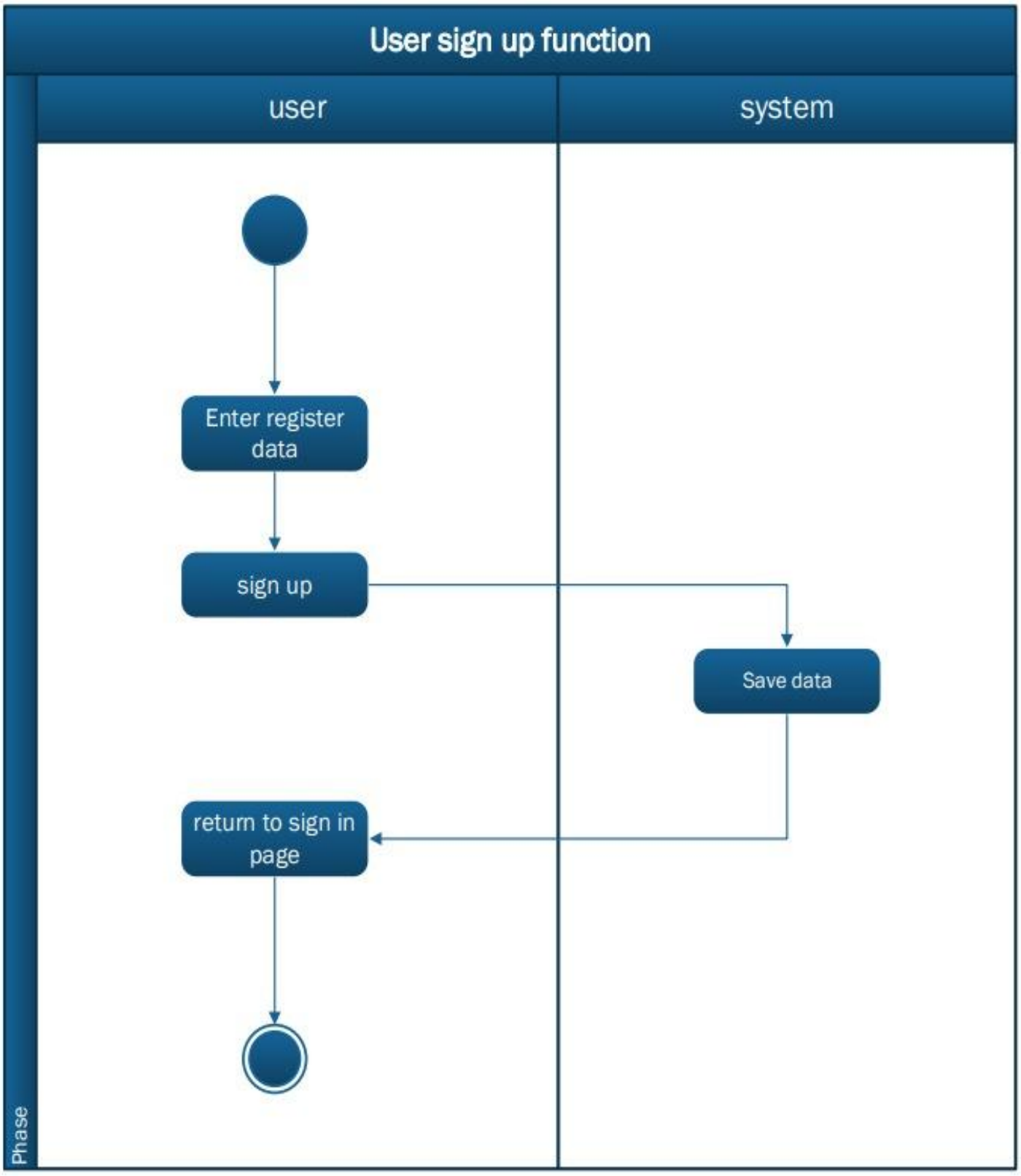
person reset password.

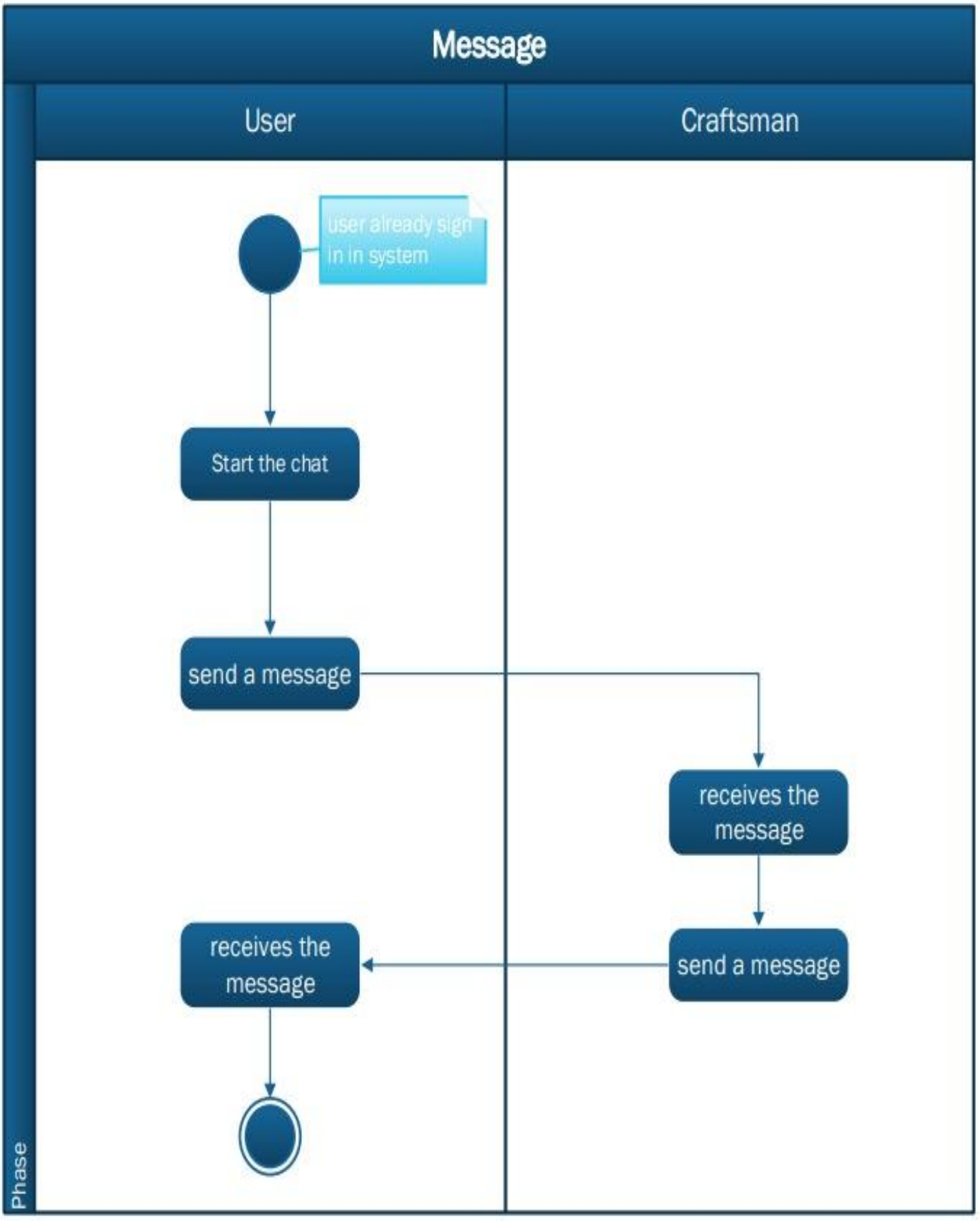


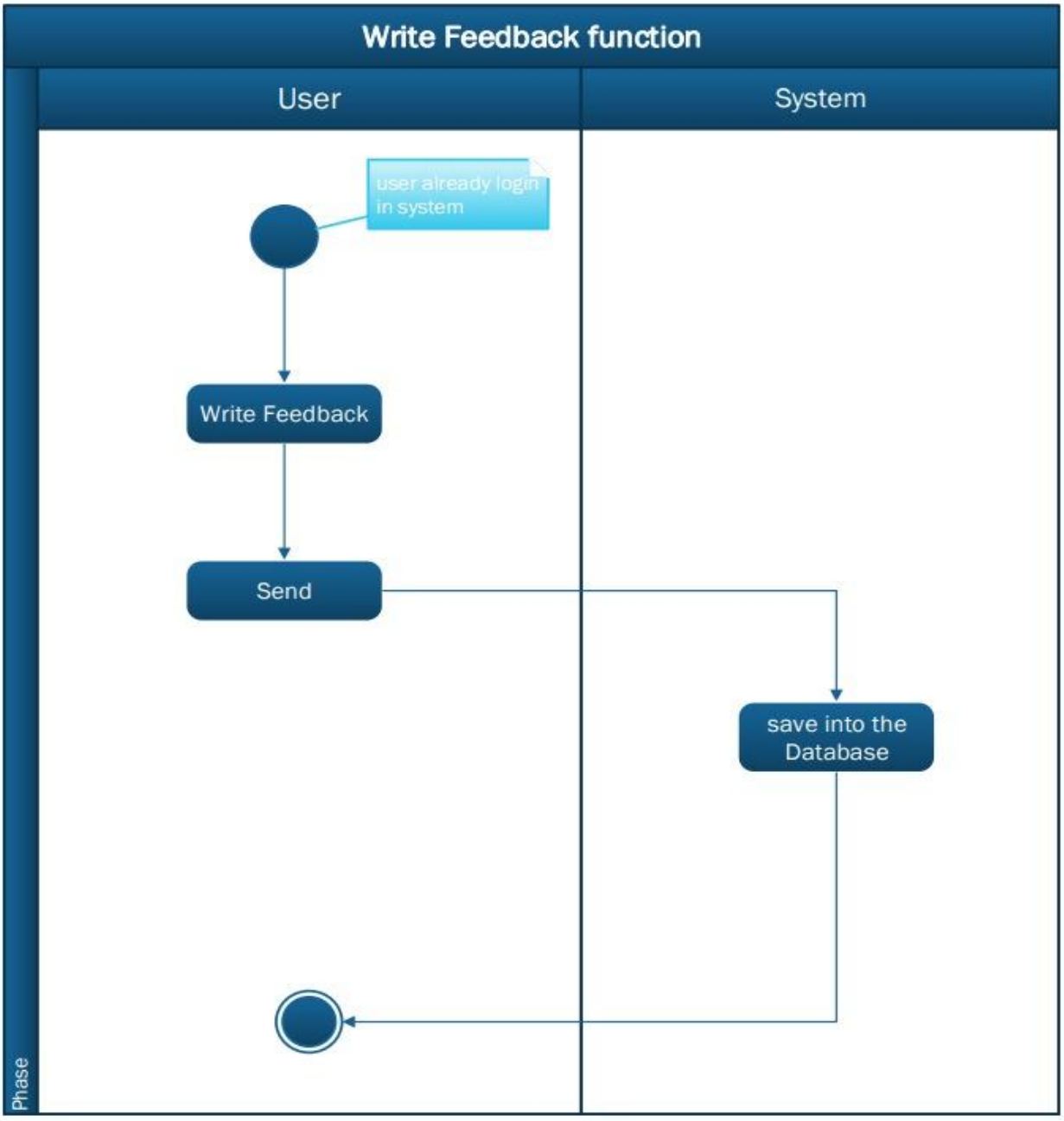
-Activity diagram.

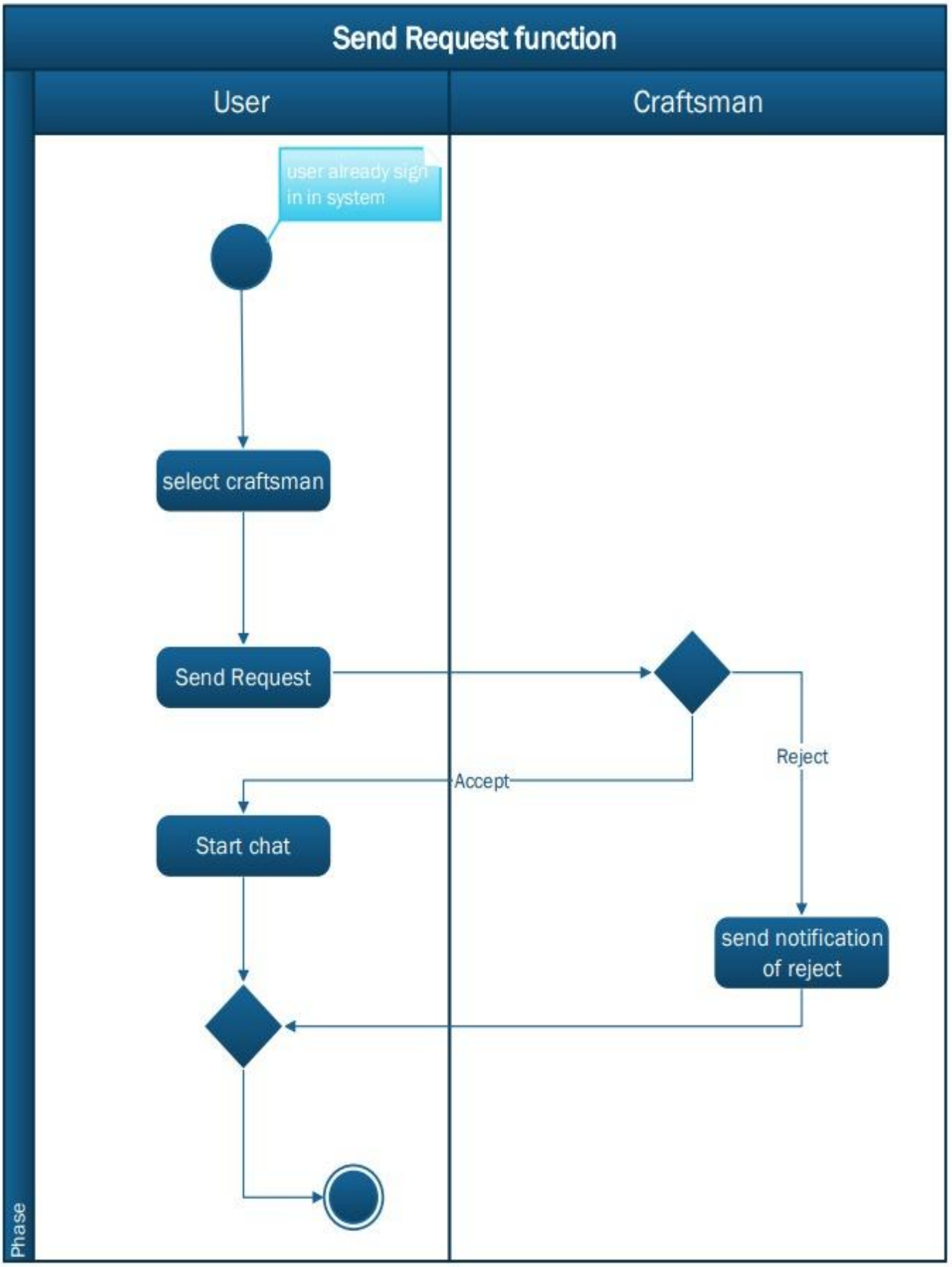
User



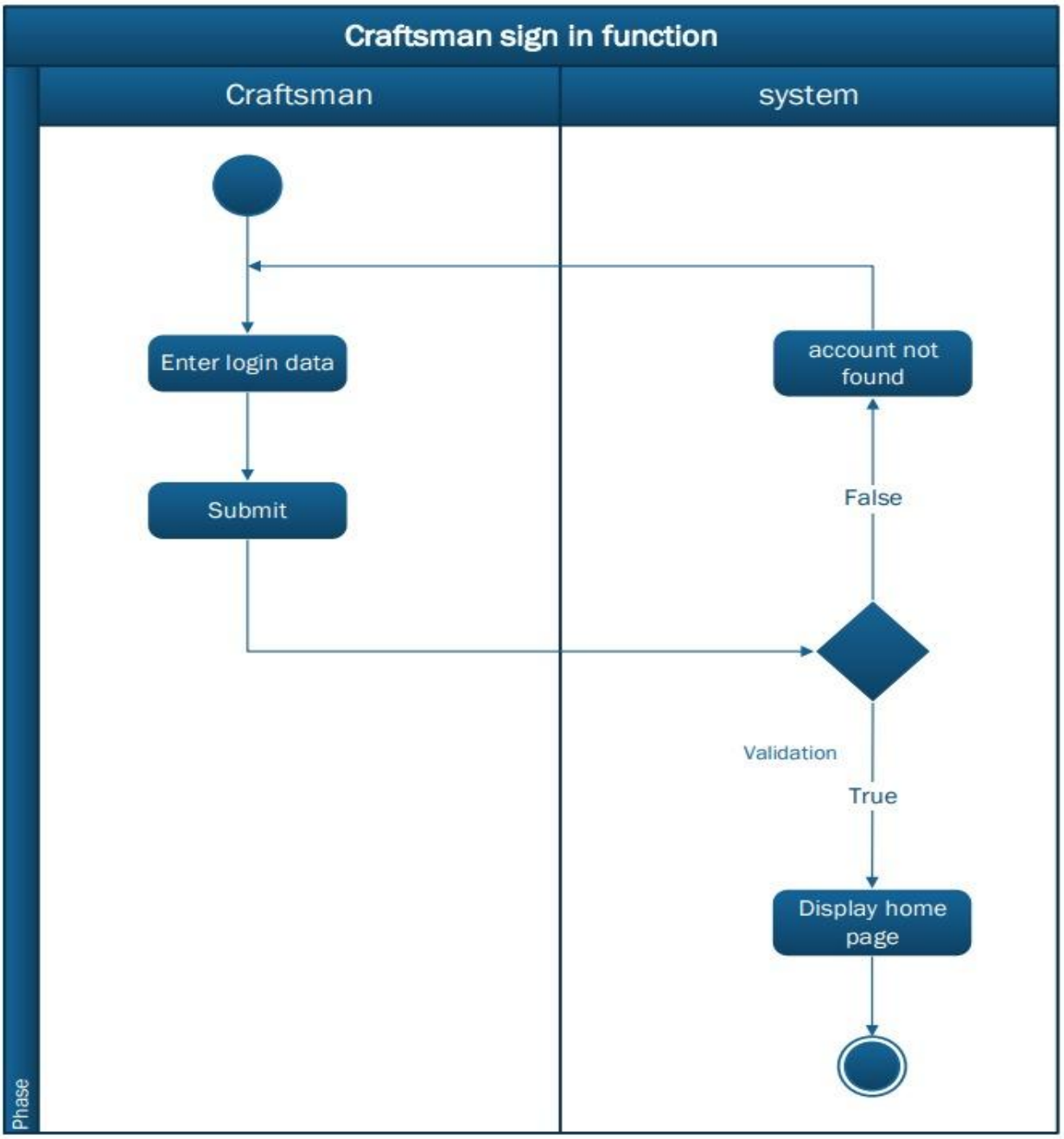

User sign up function
user
system
Enter register data
sign up
Save data
return to sign in page
Phase

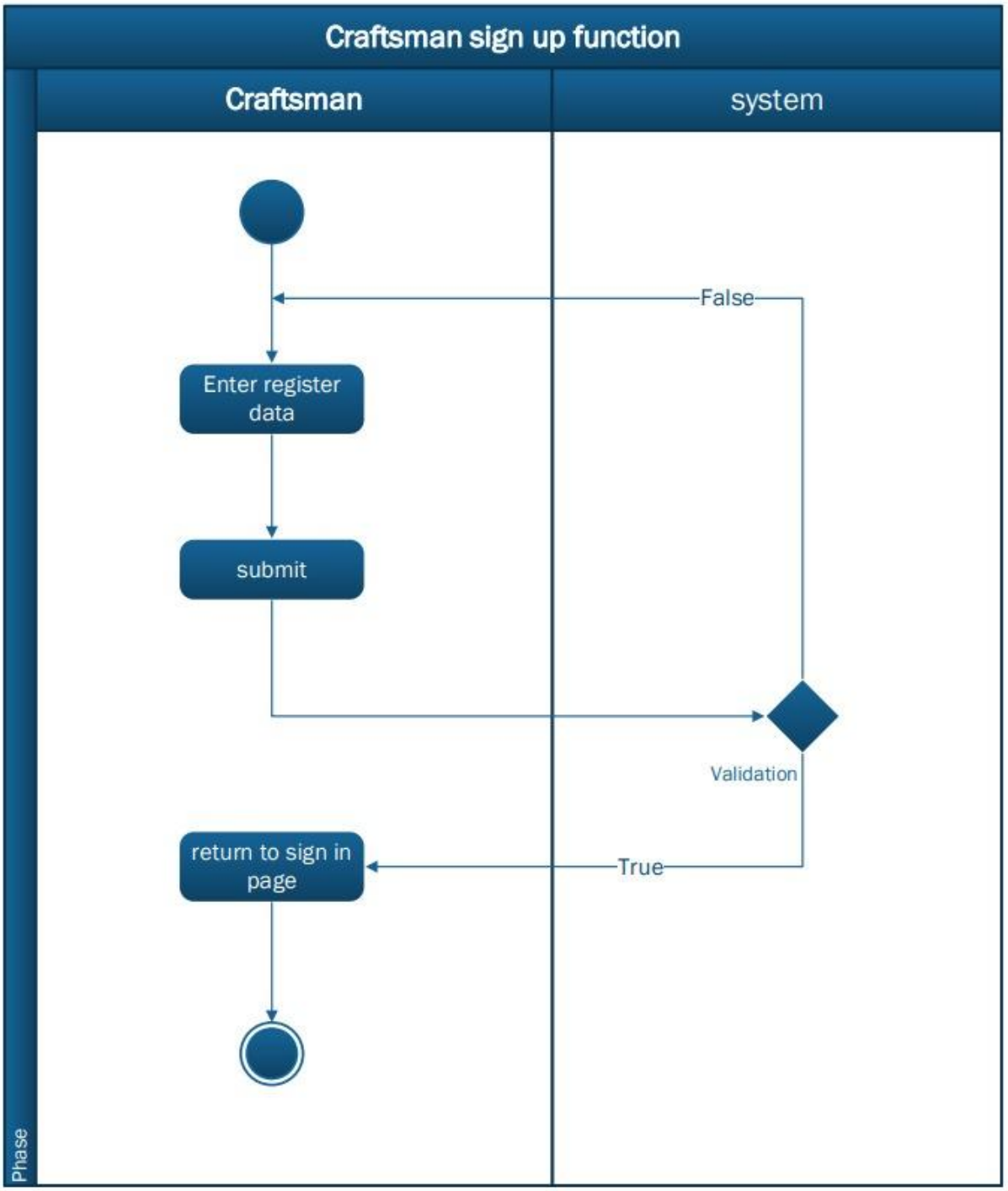

Message
User
Craftsman
user already sign in in system
Start the chat
send a message
receives the message
receives the message
send a message
Phase

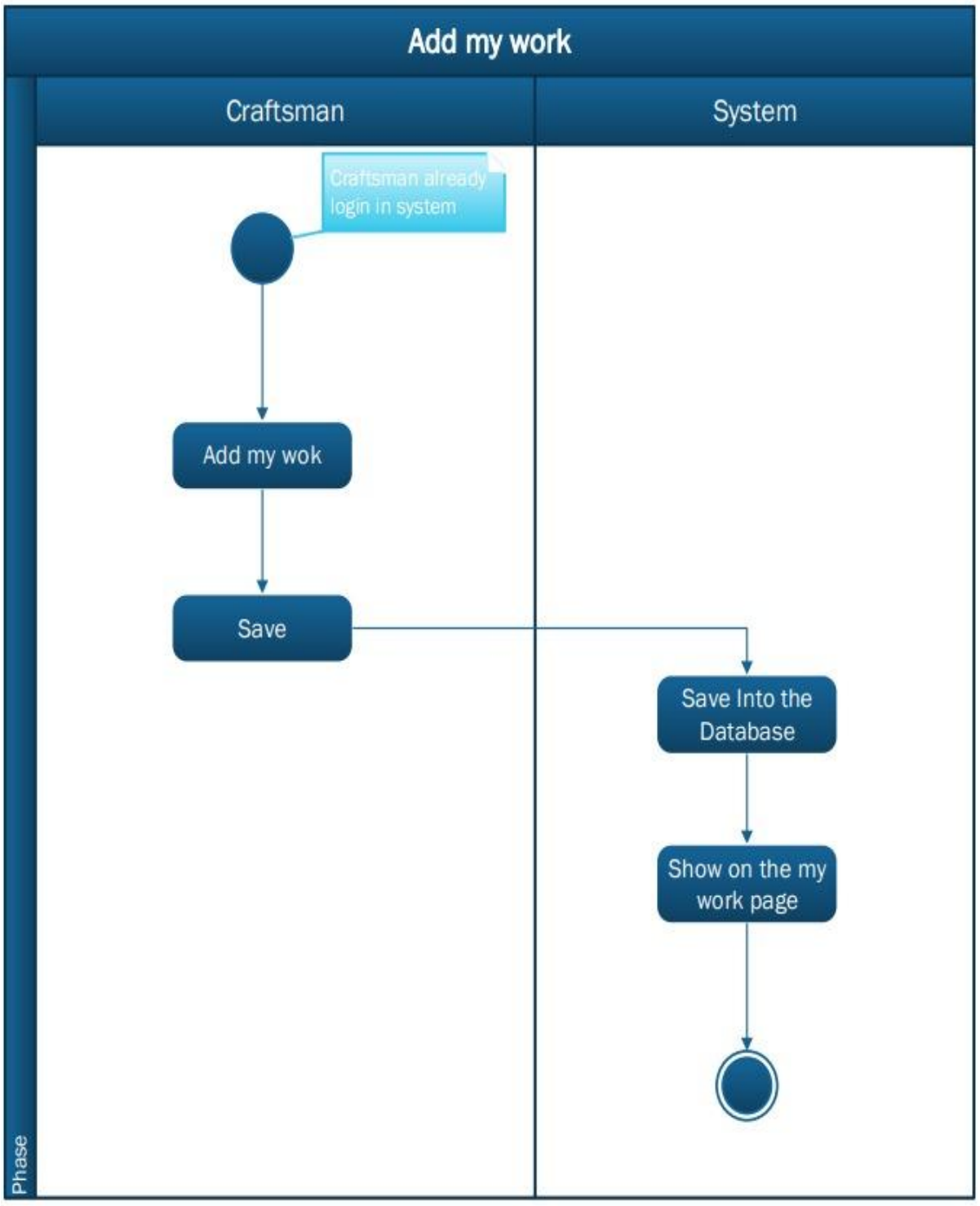

Write Feedback function
User
System
user already login in system
Write Feedback
Send
save into the Database
Phase

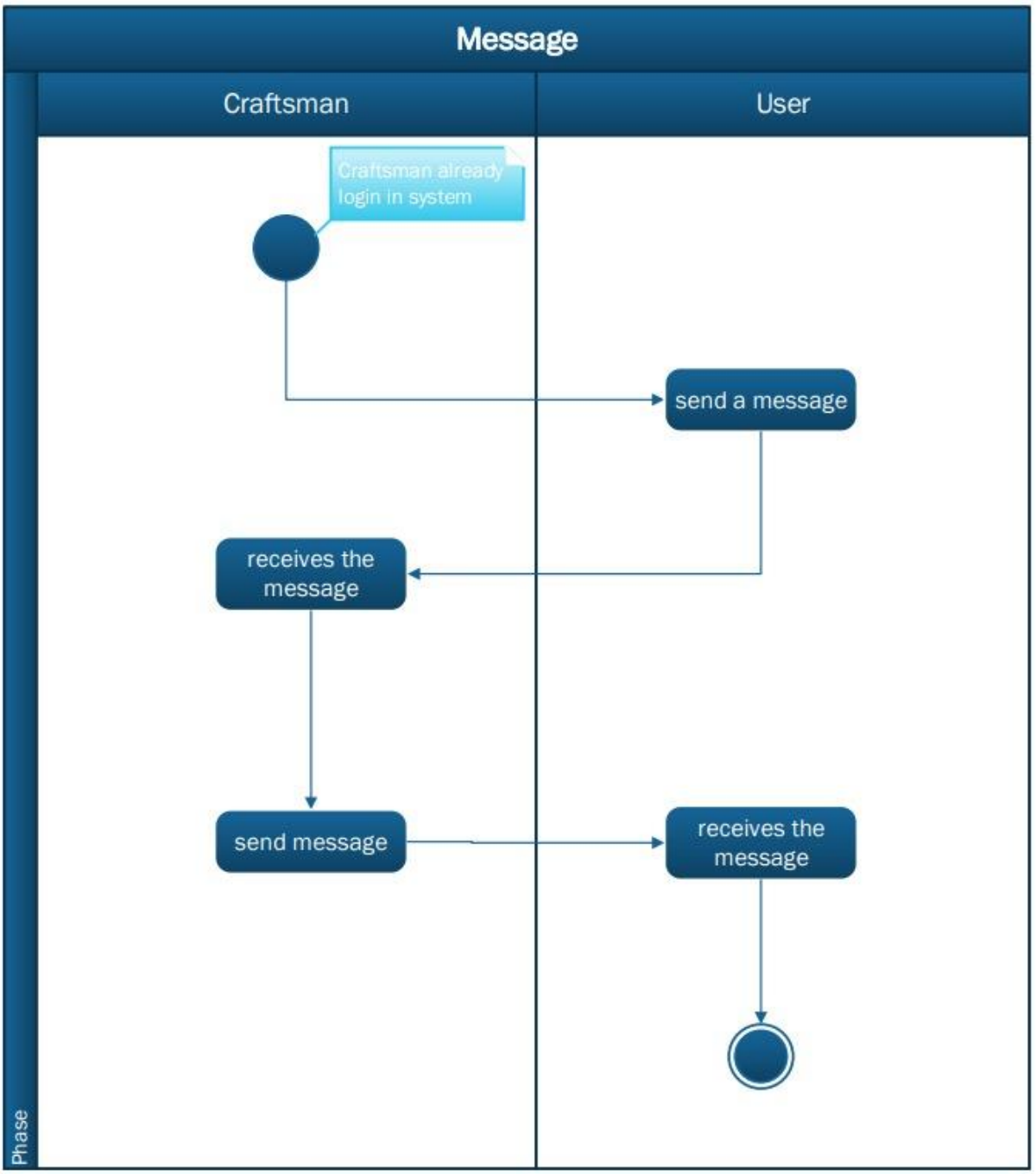

Send Request function
User
Craftsman
user already sign in in system
select craftsman
Send Request
Accept
Reject
Start chat
send notification of reject
Phase

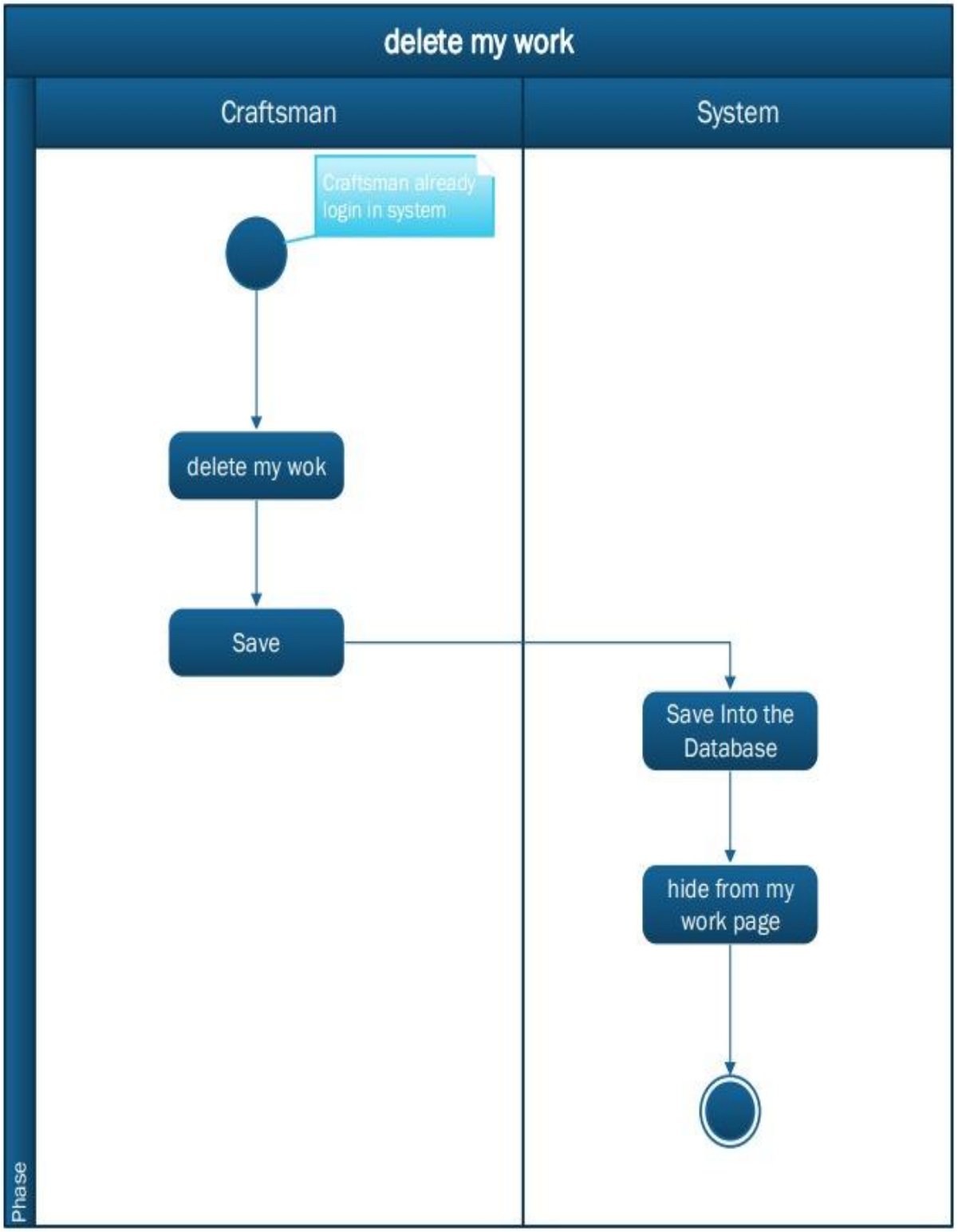
Craftsman

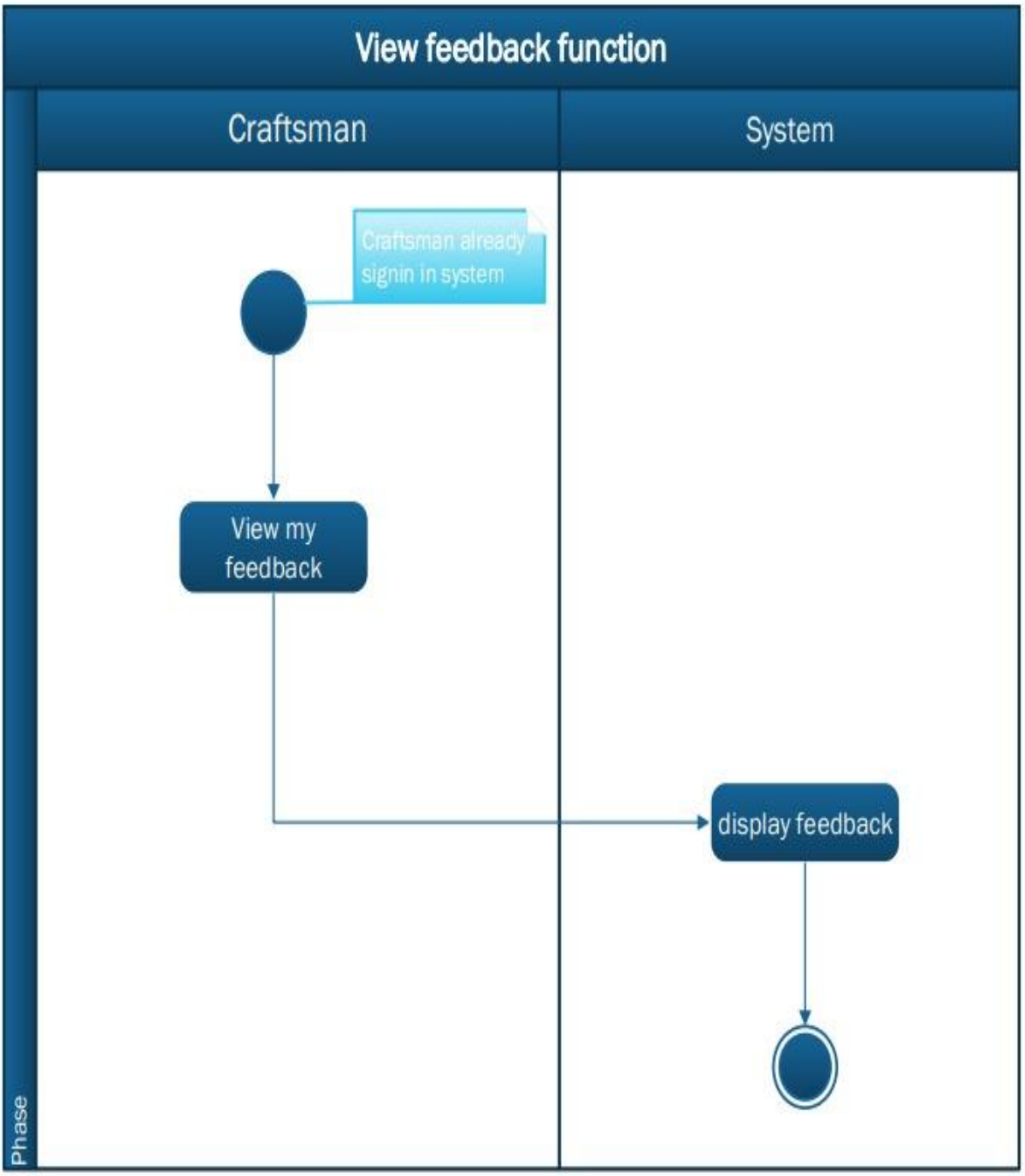

Craftsman sign in function
Craftsman
system
Enter login data
Submit
account not found
False
Validation
True
Display home page
Phase

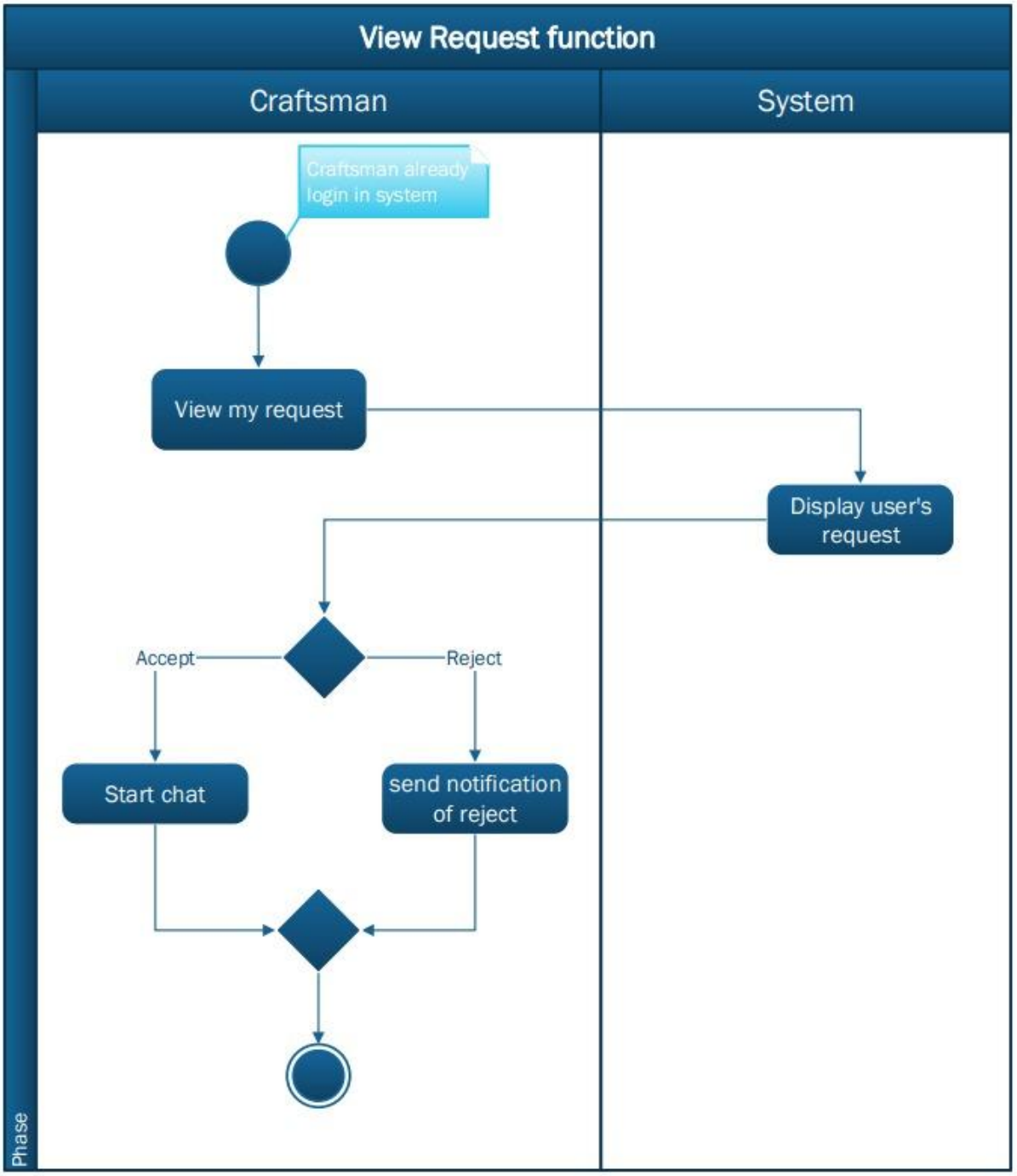

Craftsman sign up function
Craftsman
system
Enter register data
submit
False
Validation
True
return to sign in page
Phase

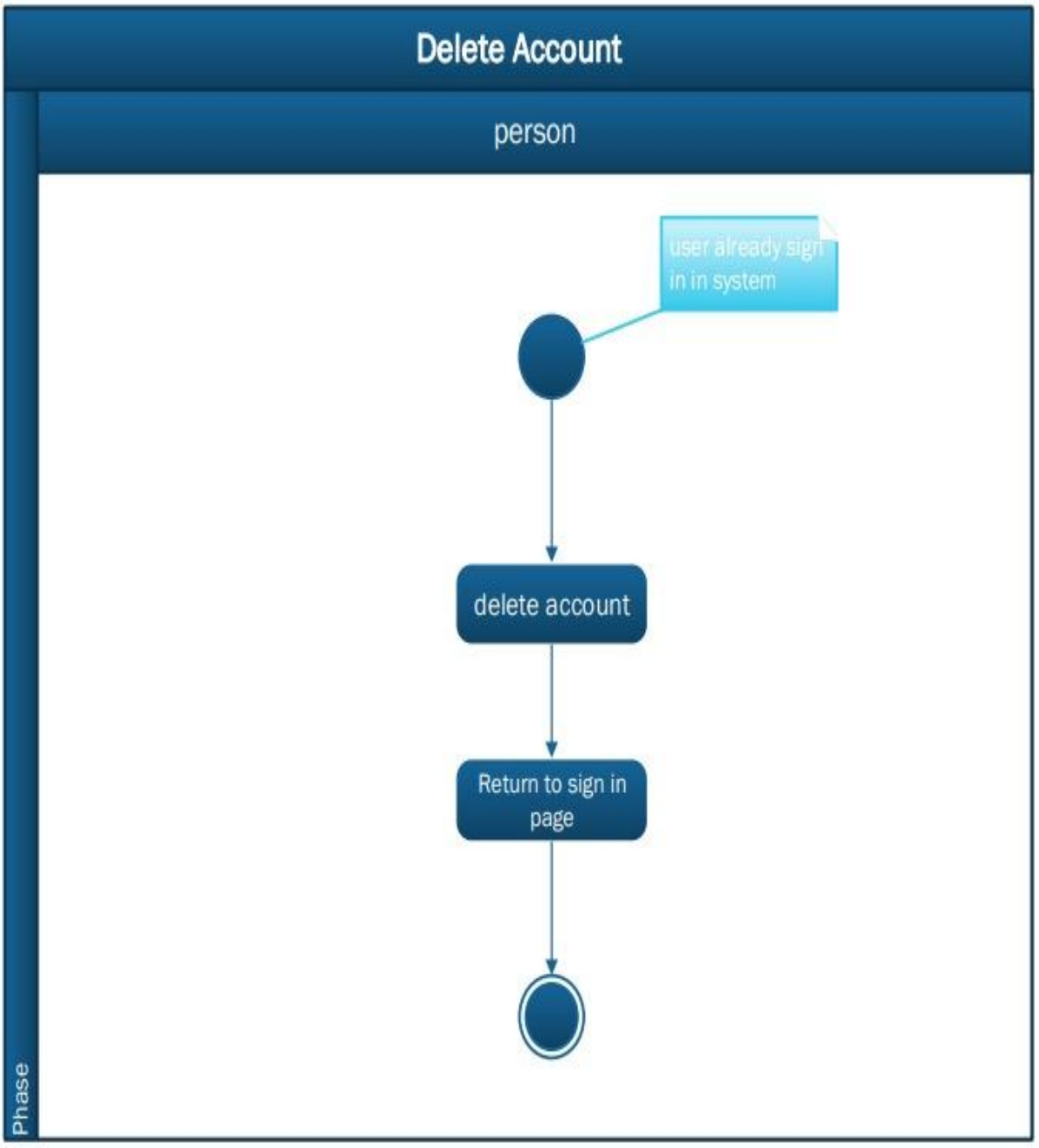

Add my work
Craftsman
System
Craftsman already login in system
Add my wok
Save
Save Into the Database
Show on the my work page
Phase

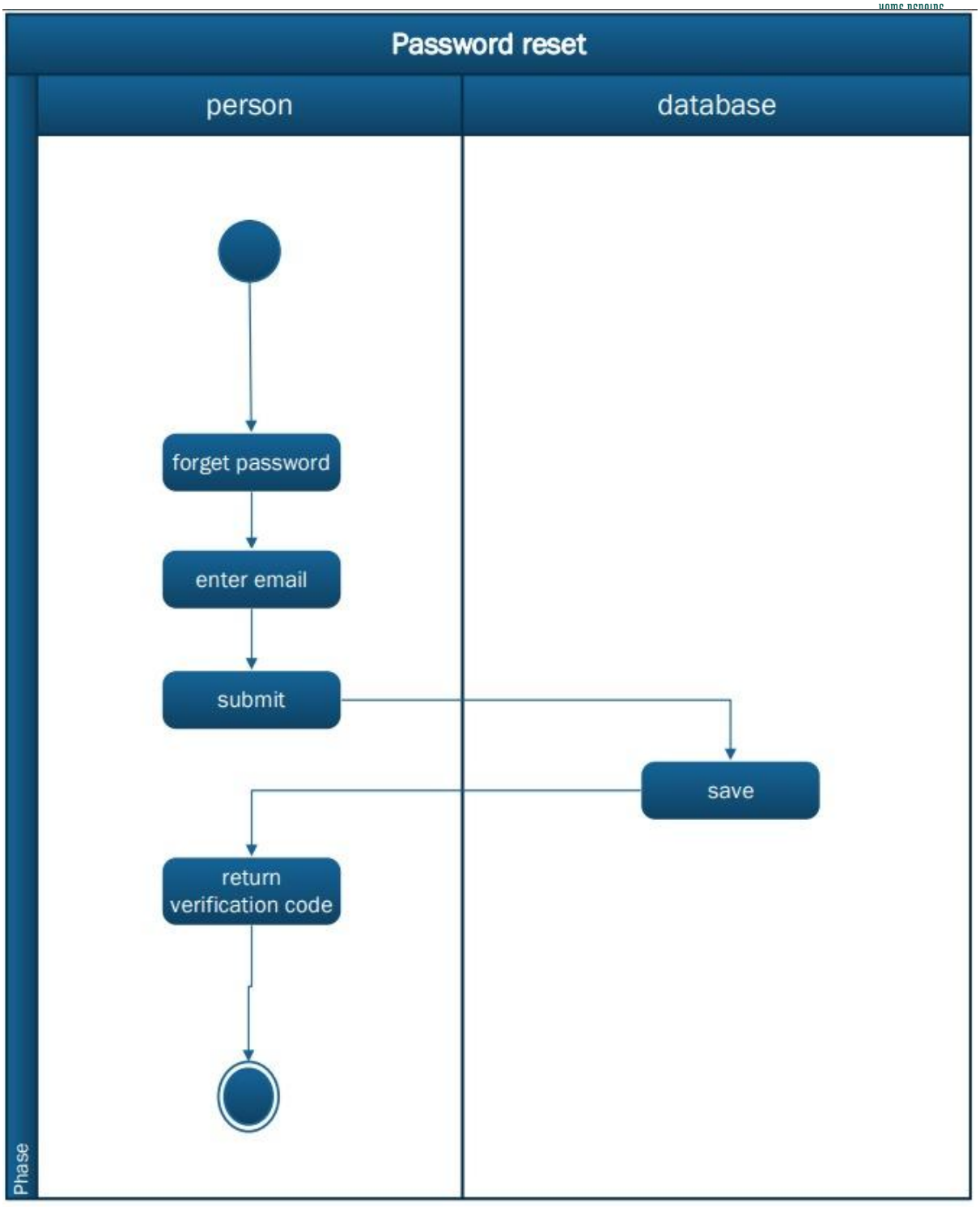

Message
Craftsman
User
Craftsman already login in system
send a message
receives the message
send message
receives the message
Phase

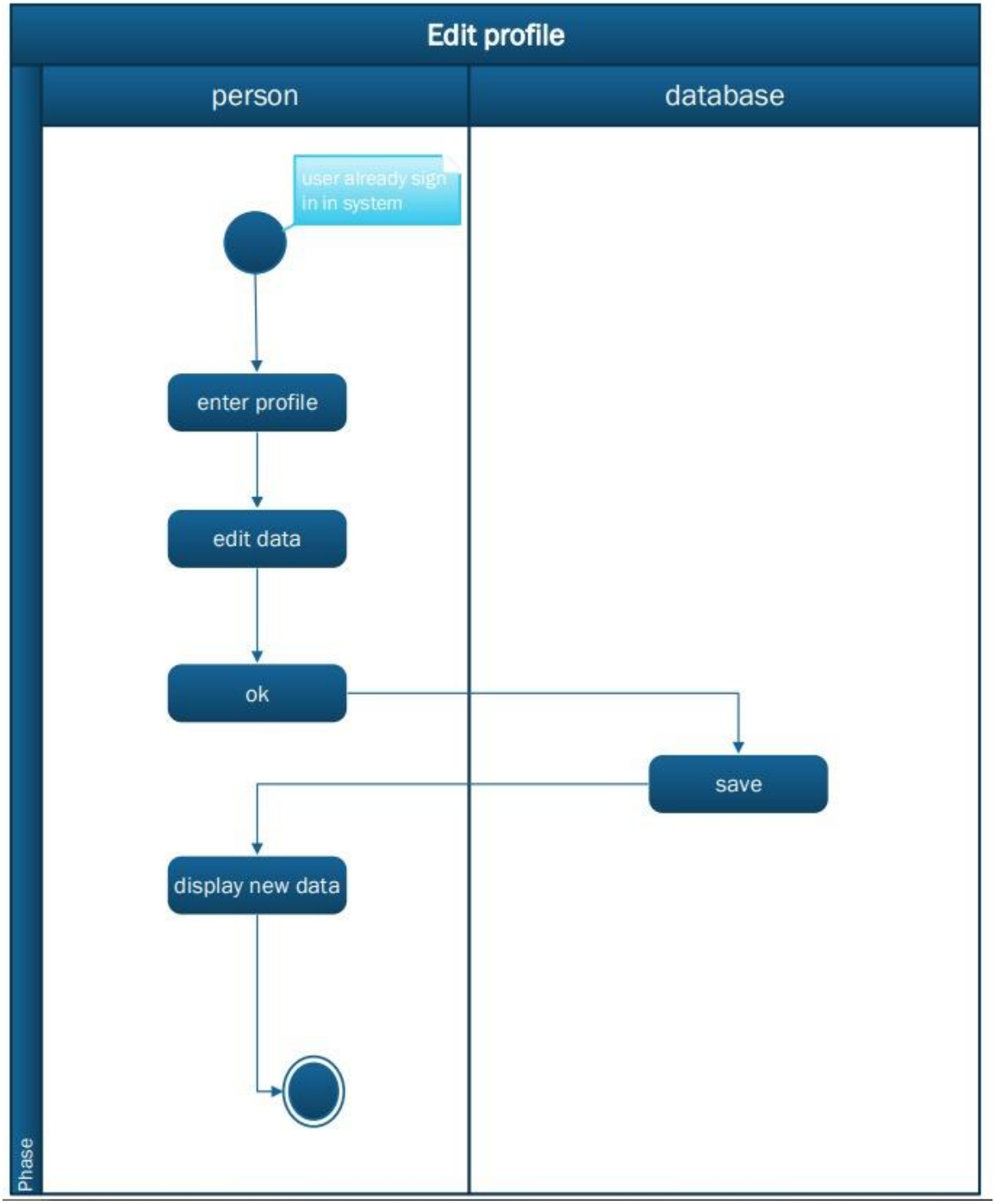

delete my work
Craftsman
System
Craftsman already login in system
delete my wok
Save
Save Into the Database
hide from my work page
Phase

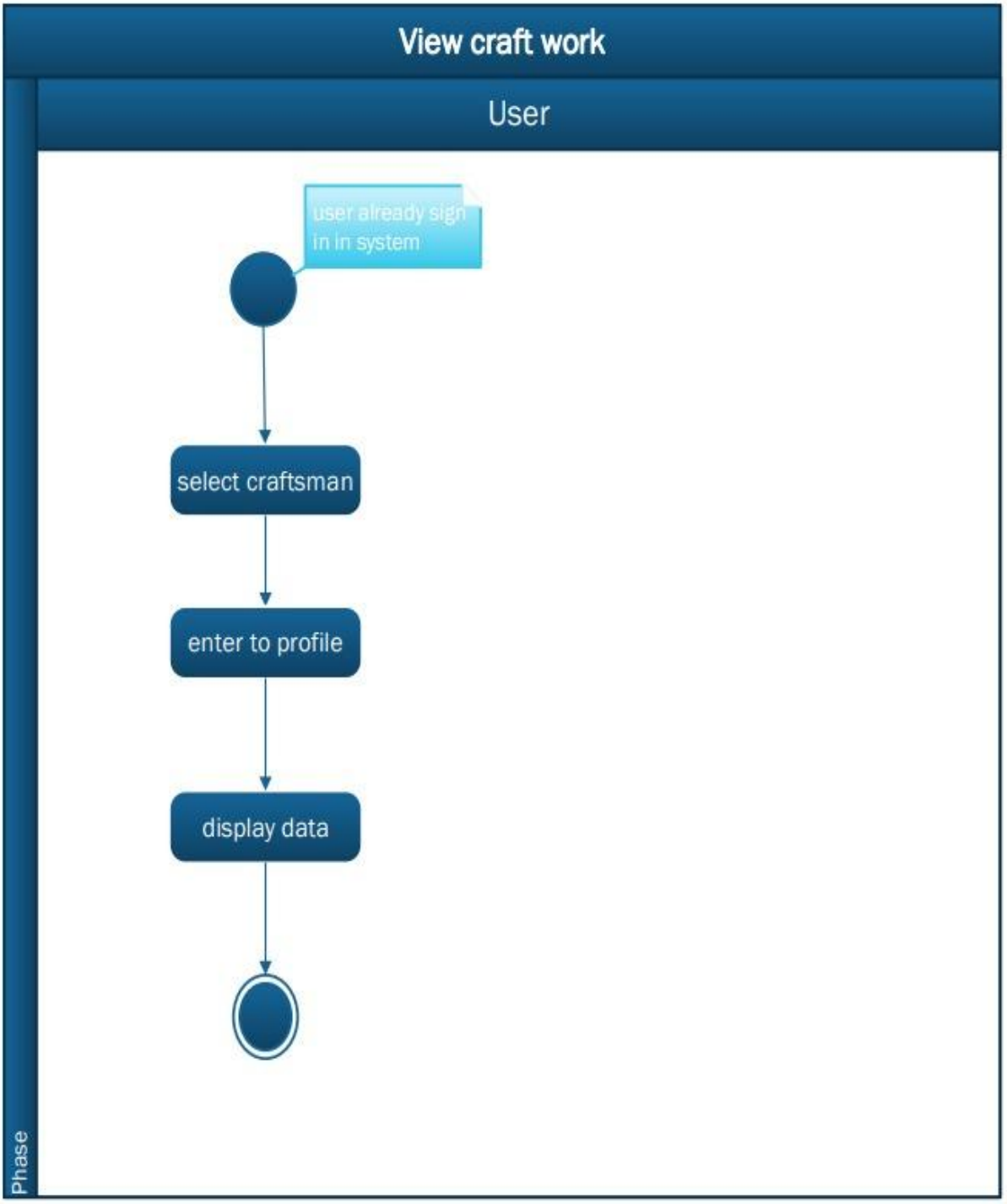

View feedback function
Craftsman
System
Craftsman already signin in system
View my feedback
display feedback
Phase


View Request function
Craftsman
System
Craftsman already login in system
View my request
Display user's request
Accept
Reject
Start chat
send notification of reject
Phase

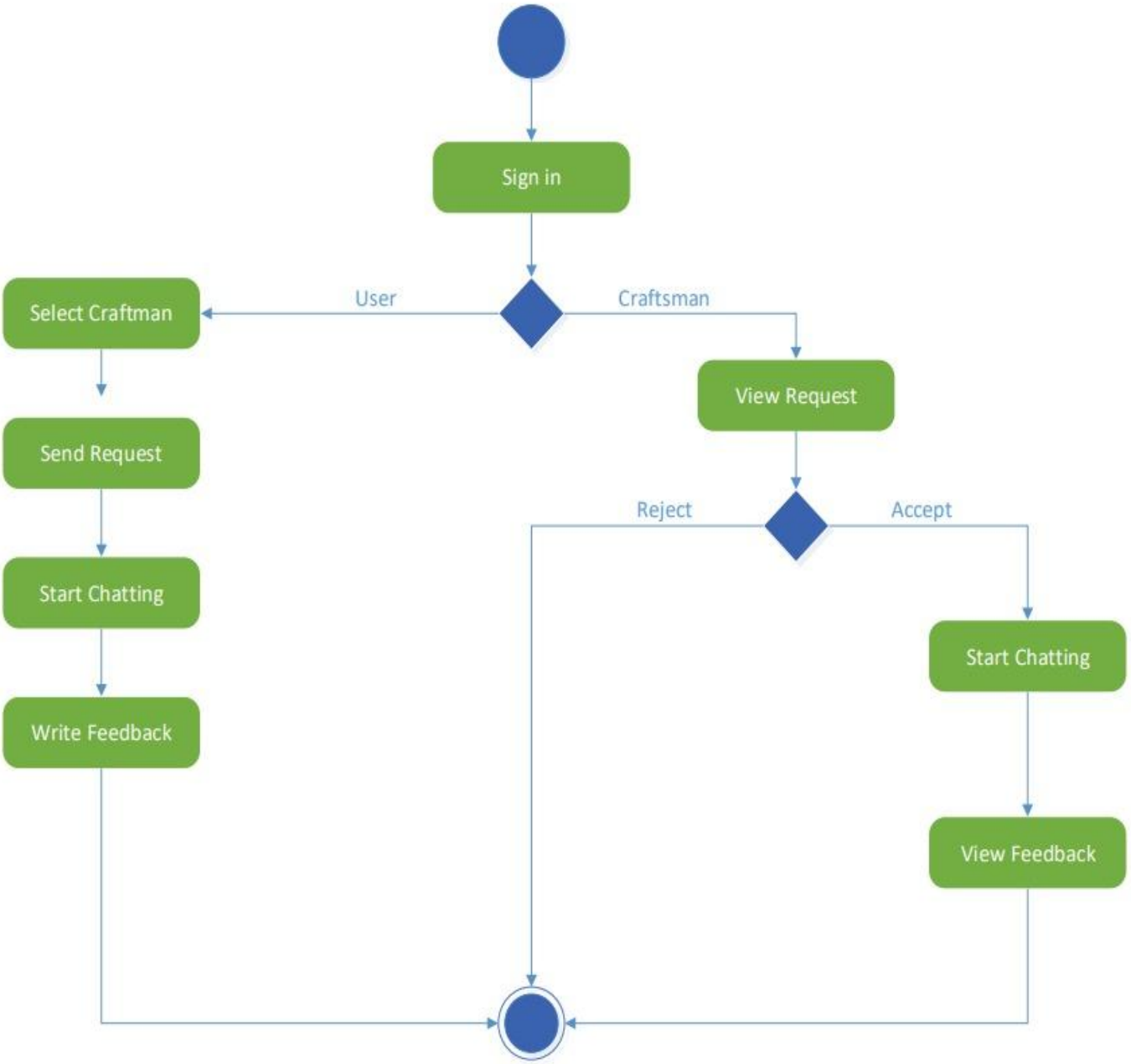

Delete Account
person
user already sign in in system
delete account
Return to sign in page
Phase


Password reset
person
database
forget password
enter email
submit
save
return
verification code
Phase


Edit profile
person
database
user already sign in in system
enter profile
edit data
ok
save
display new data
Phase


View craft work
User
user already sign in in system
select craftsman
enter to profile
display data
Phase

craftsmen state_machine



Chapter Four: Project “Implementation”

In this chapter we’re going to discuss and go deeper into Craftsmen's application implementation and present its code and the algorithms used to build it.

- Screenshots from the code (main functions)

```
Future<void> sendRequest({
  required String userId,
  required String craftId,
  }){async

    final ValueNotifier<bool> isRequested= ValueNotifier<bool> (false);
    isRequested.value = !isRequested.value;
    FirebaseFirestore.instance.collection(
        'craftsman')
       .doc(userId).collection("request").add({
      'Name': myModel?.name,
      'City' : myModel?.city,
      'userId': myModel?.uId,
      'status': 'pending',
    }).then((value) {
      FirebaseFirestore.instance.collection(
          'craftsman')
         .doc(craftId).get()
         .then((value) {
        sendNotify(to: value["deviceToken"],
            title: "Request For You",
            message: "${myModel!
                .name} send you a Request , You can see your requests from Requests",
            userId: myModel!.uId,
            name: myModel!.name,
            image: myModel!.image);
      });
    });}
```

```
Future<void> acceptRequest( {
  required String userId,
  required DocumentSnapshot document,
  }){async
   {
    CollectionReference edit = FirebaseFirestore.instance.collection('craftsman').doc(userId).collection('request');
    DocumentReference docref = edit.doc(document.id);
    await docref.get().then((value) =>
        FirebaseFirestore.instance.collection('craftsman').doc(userId).collection("persons").doc(document['userId'])
            .set({}).then((value) =>
            FirebaseFirestore.instance.collection('user').doc(document['userId']).collection("persons").doc(userId)
                .set({}).then((value) =>
                FirebaseFirestore.instance.collection('user').doc(document['userId']).collection("craftsmen").doc(userId)
                    .set({
                  'Name' : myModel?.name,
                  'Career' : myModel?.role
                })
            ).then((value) => docref.delete())));
    await FirebaseFirestore.instance.collection('user').doc(document["userId"]).get().then((value) {
      sendNotify(
        to: value["deviceToken"],
        title: "Accepted",
        message: "${myModel!.name} accepted your request , you can communicate with each other through chat",
        userId: myModel!.uid,
        name: myModel!.name,
        image: myModel!.image,
      );
    });
    Get.offNamed("/ChatsScreen");}}
```

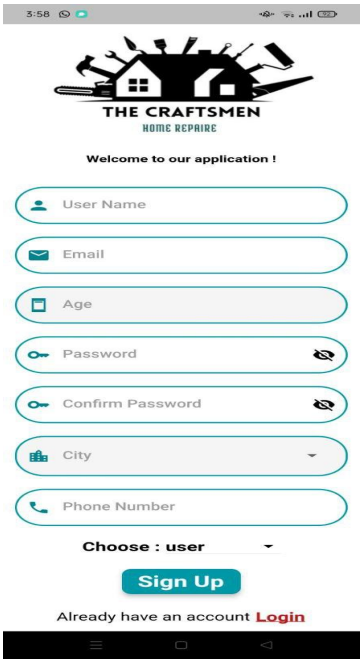
```
Future signIn() async {
  var formData = formState.currentState;
  if (formData!.validate()) {
    formData.save();
    try {
      UserCredential userCredential = await FirebaseAuth.instance.signInWithEmailAndPassword(
        email: email,
        password: password,);
      myId =userCredential.user!.uid;
      shared!.setString('myId', myId!);
      return userCredential;
    } on FirebaseAuthException catch (e) {
      if (e.code == 'user-not-found') {
        Get.defaultDialog (
            title: " No user found for that email ! ",
            titleStyle: const TextStyle(color: Colors.blueGrey),
            middleTextStyle: const TextStyle(color: Colors.black),
            middleText: " Please try again! ",
            actions: [
              const SizedBox(width: 10,),
              ElevatedButton(style: ElevatedButton.styleFrom(backgroundColor: Colors.orange.shade800),
                  onPressed: (){
                    Get.back();
                  }, child: const Text(" Ok ",style: TextStyle(color: Colors.black),)),]);}   // ElevatedButton
      else if (e.code == 'wrong-password') {
        AwesomeDialog(
            context: context, dialogType: DialogType.error, animType: AnimType.rightSlide,
            title: 'Please try again!',
            desc: 'Wrong password provided for that user',
            btnOkOnPress: () {},
            btnOkColor: AppColors.primary).show();  }}}}
```

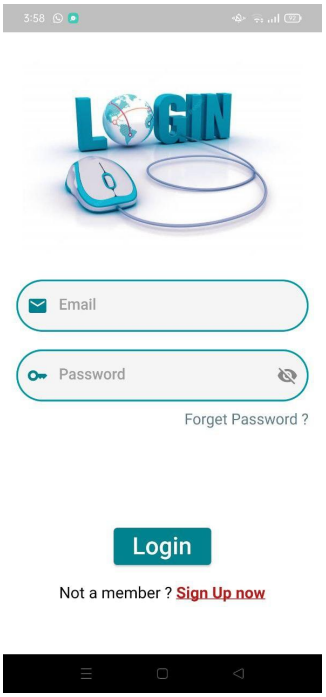
```
void signUp(String email, String password, String role)  {
  const CircularProgressIndicator();
  if (formState.currentState!.validate()) {
        FirebaseAuth.instance.createUserWithEmailAndPassword(email: email, password: password)
        .then((value) {
      myId = value.user!.uid;
      postDetailsToFirestore(email, role );
    })
        .catchError((e) {
          if (e is FirebaseAuthException && e.code == 'email-already-in-use') {
            AwesomeDialog(
              context: context,
              dialogType: DialogType.ERROR,
              animType: AnimType.BOTTOMSLIDE,
              title: 'Warning',
              desc: 'This email is already in use!',
              btnOkText: 'Ok',
              btnOkOnPress: () {},
                btnOkColor: AppColors.primary
            )..show();
          }});
  }
}
```

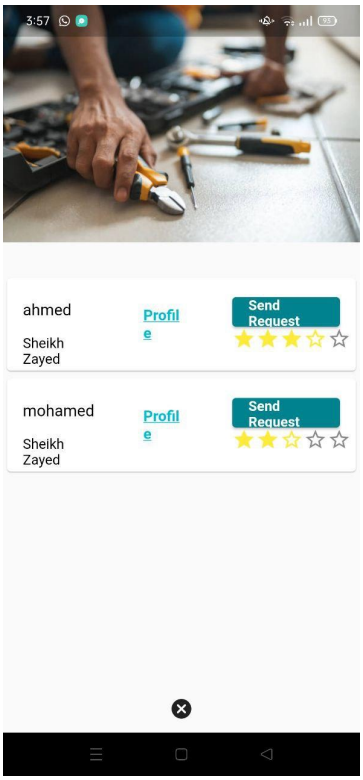
```
Future<void> sendNotify({
  required String to, required String title, required String message, String? userId, String? name, String? image,
}){async
  await http.post(
    Uri.parse('https://fcm.googleapis.com/fcm/send'),
    headers: <String, String>{
      'Content-Type': 'application/json; charset=UTF-8',
      "Authorization": "key=AAAAXKlqos0:APA91bFQ2HPBvWiImGs96W4F-s-0Q-RQ7rZ7FWGA_veFkzURM9S5FuglOwN6XWJ9tNG2mrWvfWY424nFltsTp6XCa2Y
    body: jsonEncode({
      "to":to,
      "notification":{
        "title":title,
        "body":message,
        "sound":"default"},
      "android":{
        "priority":"HIGH",
        "notification":{
          "notification_priority":"priority_MAX",
          "sound":"default",
          "default_sound":"true",
          "default_vibrate_timings":"true",
          "default_light_settings":"true"}},
      "data":{
        "click_action":"FLUTTER_NOTIFICATOIN_CLICK",
        "type":"order",
        "id":userId,
        "name":name,
        "image":image,}},);
  print('FCM request for device sent!');}
```

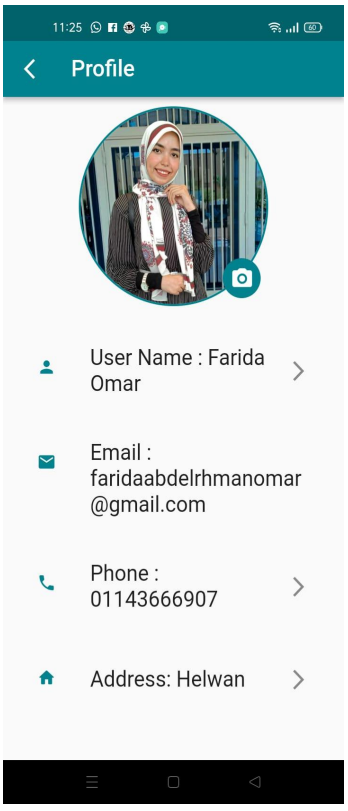
```
void sendMessage({
  required String receiverId,
  required String text ,
  String? image,
}){
  MessageModel message=MessageModel(
    senderId: myId!, text: text, dateTime: Jiffy().yMMMMEEEEdjm, receiverId: receiverId, indexMessage:messages.length,
    image: image,
  );  // MessageModel
  late String from,to;
  if (myModel!.role == 'user') {
    from='user';
    to = 'craftsman';
  }
  else {
    from='craftsman';
    to = 'user';
  }
  FirebaseFirestore.instance.collection(from).doc(myId).collection('persons').doc(receiverId).collection('chat')
      .add(message.toMap()).then((value) {
    (SendMessageSuccessState());
  })
      .catchError((error){
    (SendMessageErrorState());
  });
  FirebaseFirestore.instance.collection(to).doc(receiverId).collection('persons').doc(myId).collection('chat')
      .add(message.toMap()).then((value) {
    (SendMessageSuccessState());
  }).catchError((error){
    (SendMessageErrorState());
  });}
```

- Interface Screens

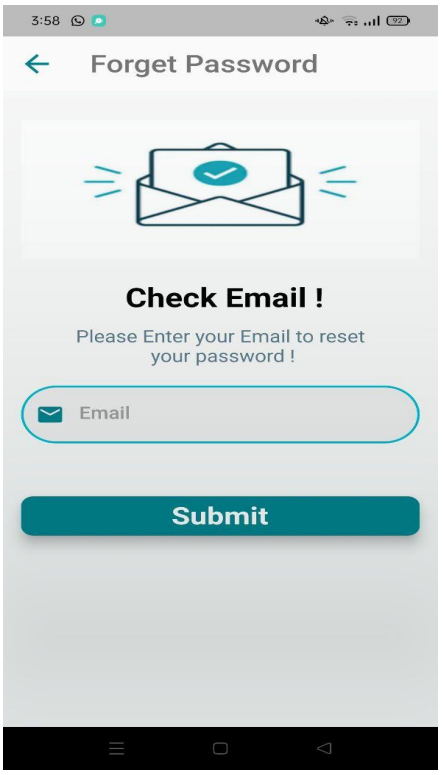


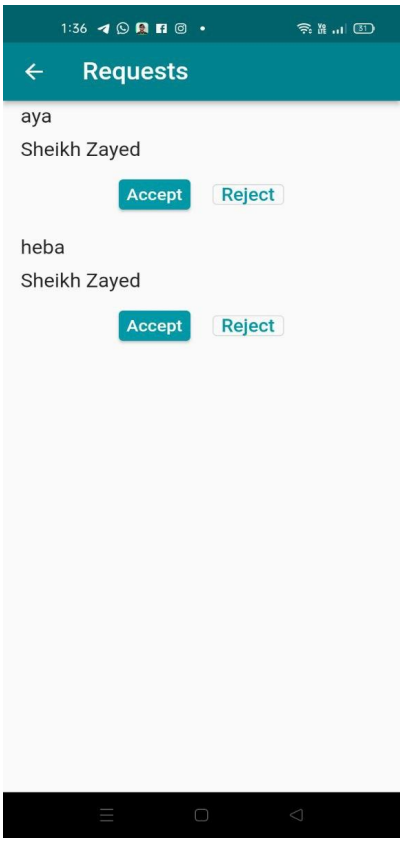


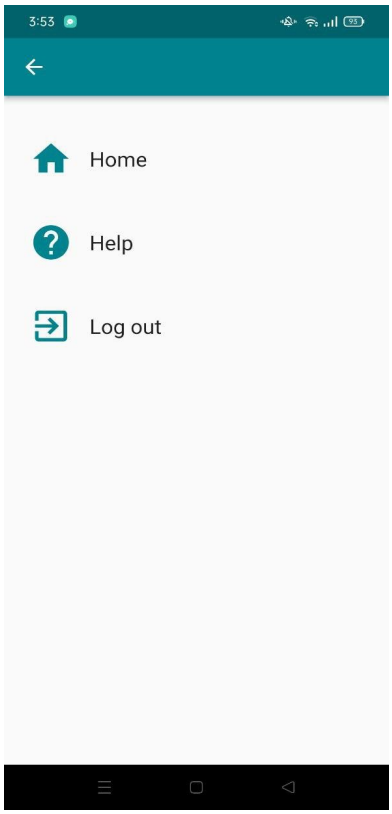


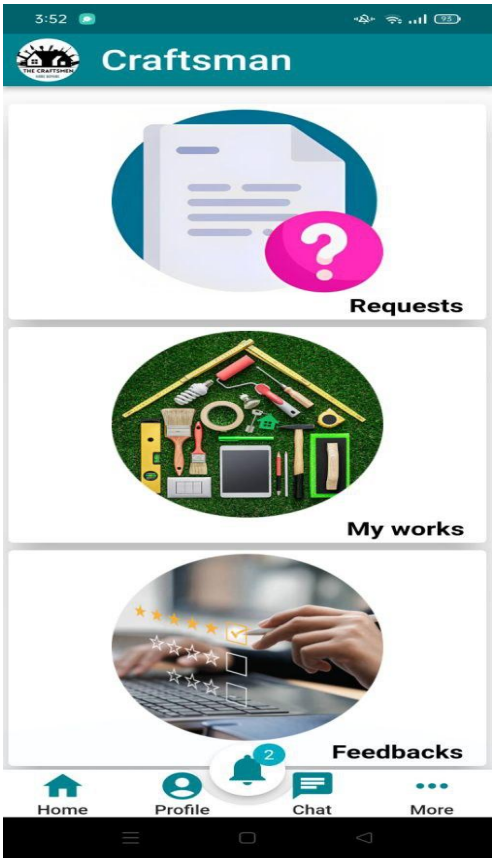


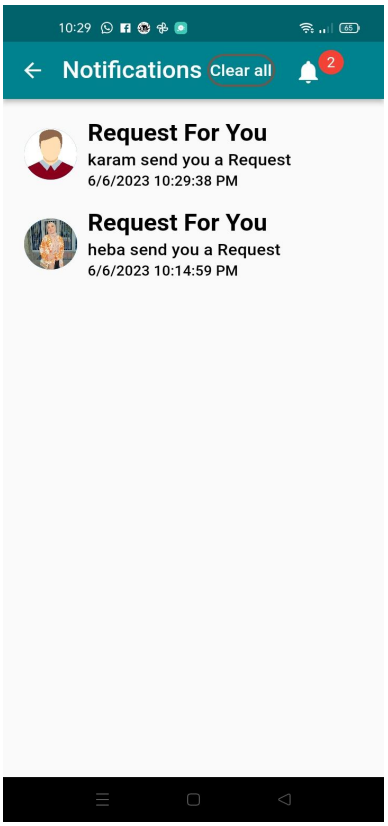

heba
hello Mr ahmed
1:57 AM
hello Mrs heba
How can I help you?
1:57 AM
I have a problem with a plug at home. I will send you a picture to clarify the problem and I want you to fix it
2:04 AM
2:04 AM
ok,
I'll look at my schedule and get back to you
type your message here...


Forget Password
Check Email !
Please Enter your Email to reset your password !
Email
Submit


Requests
aya
Sheikh Zayed
Accept
Reject
heba
Sheikh Zayed
Accept
Reject


Home
Help
Log out


Craftsman
Requests
My works
Feedbacks
Home
Profile
Chat
More


Notifications
Clear all
Request For You
karam send you a Request
6/6/2023 10:29:38 PM
Request For You
heba send you a Request
6/6/2023 10:14:59 PM

CHAPTER FIVE: PROJECT "TESTING"

In this chapter we're going to discuss and go deeper in The craftsmen application's testing and present the types of testing to be used and test cases we examined our application through.

5.1 Functional Testing

5.1.1 Unit Testing

Testing of individual items (e.g., modules, programs, objects,classes, etc.) As part of the coding phase, in isolation from other development items and the system.

5.1.2 Integration Testing

Testing the interfaces between major (e.g. systems level application modules) and minor (e.g. individual programs or components) items within the application, also, testing the application as whole after committing a file into a branch on GitHub.

5.1.3 System Testing

Testing a system behavior after it is fully integrated, when development is finished, hence, the system can be tested as a complete entity.

5.1.4 Regression Testing

To check older functionality after integrating new functionality.

5.1.5 Acceptance testing

Testing to ensure that a development is ready to be deployed into the business, operational or production environment.

5.2 Non-Functional Testing

5.2.1 Performance Testing

Accomplished a designated function regarding processing time and throughput rate.

5.2.2 Load Testing

Measuring the behavior of within creasing load that can be handled by the component or system.

5.2.3 Stress Testing

Evaluate a system or component at or beyond the limits of its specified requirements.

5.2.4 Security Testing

Testing how well the system protects against unauthorized internal or external access.

CHAPTER SIX: PROJECT "RESULTS AND DISCUSSION"

In this chapter we're going to find out the results of the project whether they're achieved or not and also the differences between the desired results and the actual ones.

6.1 Results.

During the analysis and planning phase, we gathered all the information we needed about the project's requirements and the features we'd use to finish it 100%. We planned to create a website that meets the needs of anyone who wants to prepare well for any interview.

6.1.1 Expected results.

- **The user can login and sign up to the application as a user.**
- **The user can update his password if he forgets it.**
- **The user can edit his information.**
- **The user can choose his craftsman.**
- **The user can send a request .**
- **The user can chat with a craftsman and book an appointment.**
- **The user can give feedback or review.**
- **The system can analyze users' problems .**
- **The system can rank craftsmen to users.**
- **The craftsman can login and sign up as a craftsman.**
- **The craftsman can accept or reject the request.**
- **The craftsman can chat with the user.**
- **The craftsman can view feedback.**

6.1.2 Actual results.

- **The user can login and sign up to the application as a user.**
- **The user can update his password if he forgets it.**
- **The user can edit his information.**
- **The user can choose his craftsman.**
- **The user can send a request .**
- **The user can chat with a craftsman and book an appointment.**
- **The user can give feedback or review.**
- **The user can rate the craftsman.**
- **The system can rank craftsmen to users.**
- **The craftsman can login and sign up as a craftsman.**
- **The craftsman can accept or reject the request.**
- **The craftsman can chat with the user.**
- **The craftsman can view feedback.**
- **The craftsman can view the rate.**

6.2 Discussion.

We don’t have differences between the expected result and the actual one as we did our best to make a complete system.

CHAPTER SEVEN: PROJECT "CONCLUSION"

We have done our best to participate and assist in the advancement of the technological community, as well as to work on what serves the people in society, as well as to reduce cost and increase excellent service with limited resources. CRAFTSMAN can help craftsmen, bridegrooms and families to find a professional craftsman .

We have tried to do this with our limited resources, and we are sure that we will expand this application, improve its performance and we will add many new Features if given the right resources and enough time.

CHAPTER EIGHT: PROJECT "FUTURE WORK"

Finally, here we are writing the very last words and putting the last lines of our story and adventure at FCAI-HU, working in this project was really different from any other project we worked on during the last 4 years, this project was full of the feeling of responsibility towards our society and ourselves, that is why we try to do our best in it, regardless of the poor resources we had and the limited time.

We build the application in a spiral way. We have many ideas yet to do in the project. We want to add some features so that the project covers the idea we build it for in all its edges.

8.1 Future work

- We will add a feature of adding his/her problem with voice.
- We will make the owner able to use a mobile application instead of the dashboard.
- We will make the craftsmen able to use a mobile application instead of the dashboard.
- We will add a feature to pay with Vodafone/Etisalat/ Fawry cash.

Bibliography

https://stackoverflow.com/

https://www.youtube.com/@WaelabohamzaFlutter

https://pub.dev/

https://console.firebase.google.com/project/facetest-22f68/firestore/data/~2Fcraft~2F5QDFSigf5YfEgfkXBbJ8FTymuBj1