Item Maintenance Management System

GROUP CS 40

Computer Science

KEEP UP WITH

UPKEEP

Details of the Project Supervisor, Co-supervisor

Proposed Project Supervisor

Name: Dr. M.D.R.N. Dayarathna

Signature: _____

Date:

Proposed Project Co-supervisor

Name: Mr. C.J. Tennakoon

Signature: _____

Date:

Table of Contents

Table of Figures

1. Introduction

We all are using various kinds of maintainable items in our day-to-day life. By the name of maintainable items following items are preferred.

- Appliances
  - Personal Appliances ex: Laptops, Foot Cycles
  - Household Appliances ex: Refrigerator, Air Conditions
- Vehicles

Some of them would be very pricey and expensive possession. However regardless of their price, if we want to make good use of them and keep them up and running smoothly, we need to maintain them. Otherwise, they will most likely to be breakdown way before their lifespan and because you a lot of trouble concerning the time we need to spend on them as well as the repairs costs that directly affect your expenditure. Even though you know that their maintenance is important, keeping track of their general details and their maintenance tasks has become very difficult. Most of the time many of us don't know about the maintenance until they break down.

The proposed system aims to provide its users with a proper management system to track down the details and maintenance of their maintainable properties and provide them with a platform whenever they need help and guidance with their maintenance throughout the entire life cycle of their maintainable properties. This contains a system to keep records of properties and automatic suggestions about the maintenance tasks that needs to do in the item, the community to post their problems and issues regarding the maintenance, and the ability to find technicians for the repair tasks.

1.1. Problem Statement

After discussing with some people (who are using the properties that need to be maintained and technicians who do the repairs on different appliances following are the problems that we have identified.

The perspective of maintainable item owners

- Most maintainable item owners don’t have a proper way to keep track of the different properties or parts of that items they bought and when will they need to replace them before the usability period expires.

- Most maintainable item owners don't have proper knowledge of the maintenance tasks that they need to do to upkeep their maintainable properties. Most of the time they will tend to find them only when a breakdown occurs. But at that point, they were too late, and they need to either repair or replace the item that cost them their time and money.

- Even if they had the knowledge in maintenance, they might not remember in what periods they want to do certain maintenance. So, they tend to forget the tasks. Missing a maintenance task can lead them not good to upkeep the items

because may lead the items to unexpected problems and small repairs can turn into huge repairs.

- Maintainable item doesn't have a proper involved community to ask about the problems of the Properties. These are the problems such as quick fixes for repairs, equipment making a strange noise, or may be the problems with their automobile that can be solved using a quick fix. The method they are using now is to call someone they know or web browsing. But they aren't much accurate.

- Another problem they face is Difficulty in finding technicians for repair tasks and maintenance. When considering the lifecycle of an above-mentioned maintainable item service of a technician is very important. Because there are some maintenance tasks that we are unable to do by ourselves. But there is no better platform to find technicians rather than social media and some sites. Those sites only give the features to find contacts only.

- Every maintainable item that we use has a certain usability period that they can use. After that period, they need to be properly disposed in a proper way. But most of the people doesn't know how it should be done or even if they know they don't know who to contact.

- Within some of these maintainable items even though they are not functioning well, there can be some parts within that item that is still functional and can be used for some time. And there are people in need to buy those parts. But there is no proper platform for them to find those things when needed

- When we are using lots of items sometimes it's hard to keep track of what where we have put a certain item. Most of the people ten to forget the location they have kept the item.

1.2. Solution

To solve the above-mentioned problems, we propose to implement a maintainable item maintenance management system. This system will provide proper management of all the aspects of the lifecycle of a maintainable item. As we have mentioned in the introduction the properties that we are focusing on inside this system are maintainable properties that have requirement of keeping track of the maintenance. This may include,

- Household appliances
- Personal appliances
- Vehicles

The system will be implemented with the following features in order to cater the maintainable item owners and technicians with own functionalities.

- The system has features for the maintainable item owners to keep records of their maintainable items. They will be able to add their item details and maintenance tasks

into the system and the system will provide them with organized notifications to remind the users of the maintenance schedules.

- Even if the owner doesn't know what they need to do in a certain item they will be able to add them into the system and our system will automatically suggest to them the necessary tasks that they need to do on that certain item. As our system is a community-based system all the suggestions will be generated automatically.

- The system provides a feature of a community platform that allows users to post the problems or concerns about maintenance or any updates on their maintainable item as a community post and other users will be able to answer them by commenting on the posts (basically a Quora for appliances)

- The system should provide the users with the report generation feature to generate a report of overall maintenance they have done in a certain component or all the components.

- The system will provide user with the feature to find technicians for their repairs and maintenance task easily and the technicians will also provide with a platform to find work and to manage their work order.

- The system will provide user with an online marketplace where they can find the buyers for their used items as well as the sub parts of the items (example: compressor of the refrigerator.) the buyers will be another item owner or may a technician who need parts for their repair tasks. The disposal companies will also be within this marketplace and users will be able to sell their unusable items for disposal. (We do not do this development phase in our system)

2. Project Goal

The main goal of the project is to provide maintainable item owners with a fully functional community-based system to manage their properties with ease and provide them with the features for better management of their items throughout their life cycle.

3. Scope of the Project

3.1 Identification and Boundaries

The domain of appliance maintenance is a very large domain that goes from the personal level to the industrial level. With all those, we will be covering household and personal-level item management. With this scope following are the users that we have identified.

1. Maintainable item owner
2. Technician
3. System Admin
4. Guest (unregistered user)

3.2 In Scope

We will be creating a web-based application with mobile responsive interface
Below are the features that item owner can expect

- Can log into the system and add their maintainable properties to the system.
- Add their maintenance tasks to the system
- Get reminder notifications from the system.
- If they don’t know about the tasks or don't have details such as user manuals system will suggest them to users Automatically
- Features to find technicians for repair and maintenance jobs
- Community to post problems and get answers
- Maintenance report generation
-

Below are the features that technician can expect

- Find repair and maintenance jobs according to their work areas.
- Work order management to the jobs they are getting from the system

Below are the features that will be implemented for a guest

- View community posts
- View technicians

3.3 Out Scope

Although the proposed system takes care of many things in maintainable item management, some aspects are out of the scope of this project.

- We will not be developing a mobile application. Instead, we will be developing a web application mobile responsively
- Some of these appliances may have IoT devices integrated with them. The user will not be able to integrate those IoT devices with the system in this phase of development.
- The work order management feature for the technicians for the jobs outside our system will not be implemented within our scope
- Marketplace will not be developed within this phase.

4. Project Constraints & Assumptions

4.1 Constraints

- No frameworks allowed in project constraint.
- Project has to be complete within 10 months.
- User account can only be accessed by user itself.
- User cannot change email address.
- When the unregistered users want to log into the system, they have to create an account.
- Admin is not allowed to view user account data.
- Until the system is ceased for some time automatic suggestions may not work accurately for some items

4.2 Assumptions

- Users will enter the correct details when creating user accounts.
- Assume that some technical parts can be achieved without using any framework such as generating notifications.
- After a predetermined period, the system won't slow down.
- A single person maintains only one account.
- Users who have technical knowledge about maintainable items will add the maintenance activities themselves.

5. Project Feasibility

5.1 Technical Feasibility

Technical feasibility is the process of proving that the system is technically possible. This system is expected to be a web-based software.

We are planning to use the following technologies in order to implement our plans.

- HTML5, CSS3 and JavaScript will be used for frontend development and PHP, MySQL will be used for backend.
    - HTML will be used for structuring the web pages. CSS will be used for styling web pages and creating the website responsive on different devices. JavaScript will be used for adding functionalities for the web pages and make more interactive. AJAX will be used to make the updates and searching live.
    - PHP is one of the most widely used programming languages on the internet. MySQL is used to create and modify the database for our system.

- IDE and Text Editors:
    - Visual Studio Code and PhpStorm by JetBrains will be used.

- Other Tools:
    - Project Management tool: Trello
    - Communication: Discord, zoom
    - Webserver software - XAMPP
    - Version controlling - GitHub
    - Diagramming - Draw.io
    - Documentation - Google docs, MS Office 365

5.2 Ethical and Legal Feasibility

This system does not break or violate any rules or regulations under the law in Sri Lanka. Since the development uses freely available technologies there could not be any legal barriers.
The system will be dealing with some sensitive information of the user such as personal details, Household details, and vehicle details. Because of this nature, the system is ethically bounded not to disclose any information to unauthorized viewers. The following actions have been taken to prevent the ethical conflicts.

- User data – User data such as name, email, etc. would not be visible to anyone without the user's permission
- Handling user data – There would not be any kind of data selling for targeted advertising from external parties. The data added by users such as their inventory details, routine details, vehicle details, etc. are only accessible to the user only. They can only be manipulated by the user within the system.
- Community moderation – the community feature of the system will be rigorously moderated by the admin to prevent any community infringements.

According to the above facts, the system is feasible on legal and ethical sides.

5.3 Economic Feasibility

Cost-benefit analysis of the project is what determines its economic feasibility. This determines whether it is possible to implement the system.

- The development process will make use of open-source software and tools.
- A free hosting server can be used.
- No additional hardware cost.
- Considering the maintenance cost, it doesn't cost more.
- Cost of communication among the team members is manageable.

So, considering the total cost of the system, system has the potential to be inexpensive.

5.4 Operational Feasibility

Operational feasibility evaluates how well the suggested system solve the cited issues.
The main concerns found in end users' side,

- Misplacement of essential items such as user manuals when the user is required them suddenly.
- Unable to keep track of essential needed items records.
- Maintenance records help in making informative decisions.
- When it comes to the maintenance issues users doesn’t have a reliable community to get help from other users. Social media like Facebook have some groups. But with our system they will get a community that uses the same items and have similar experiences.
- Reduced excessive repair expenses

By using this system,

- Save necessary records using the system provided options.
- End user should have somewhat IT and English knowledge to accompany with the system.

So, it is clear from the foregoing that the project is operationally feasible.

5.5 Schedule Feasibility

The project timeline that the team members prepared is shown above. The waterfall model, which is the project's chosen software development model is expected to be used. The project is expected to be completed within the given one-year academic period. The project is therefore assumed to be schedule feasible. As our team consists of four members, So, we believe we have enough time to complete the project.

- The proposed Gantt Chart is attached below in the section 10. From the above, it is evident that the system is schedule feasible.

6. Deliverables

Upon completion of the project, it is expected to have the following items delivered to
The client,

- The complete web-based system with all functionalities tested and proven to be operational.
- User Guidelines
- Guides for maintenance.
- Software Requirements Specification (SRS)

7. Requirement

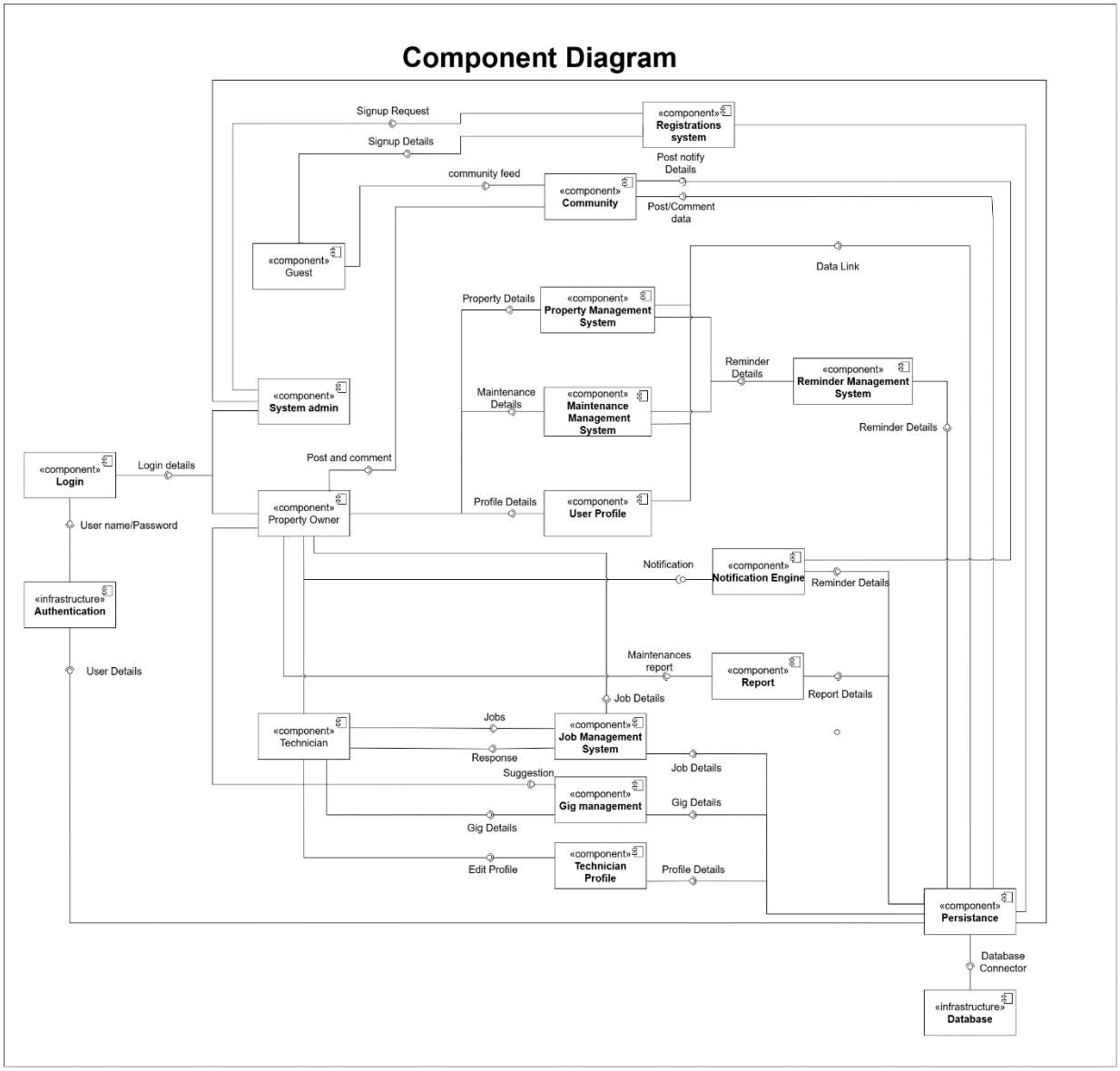
7.1 Functional Requirements

Since our system is based around a single user per home, we have identified 2 main actors of this system. Those actors are as follows,

1. Maintainable item Owner
2. Technician
3. System admin
4. Guest (Unregistered User)

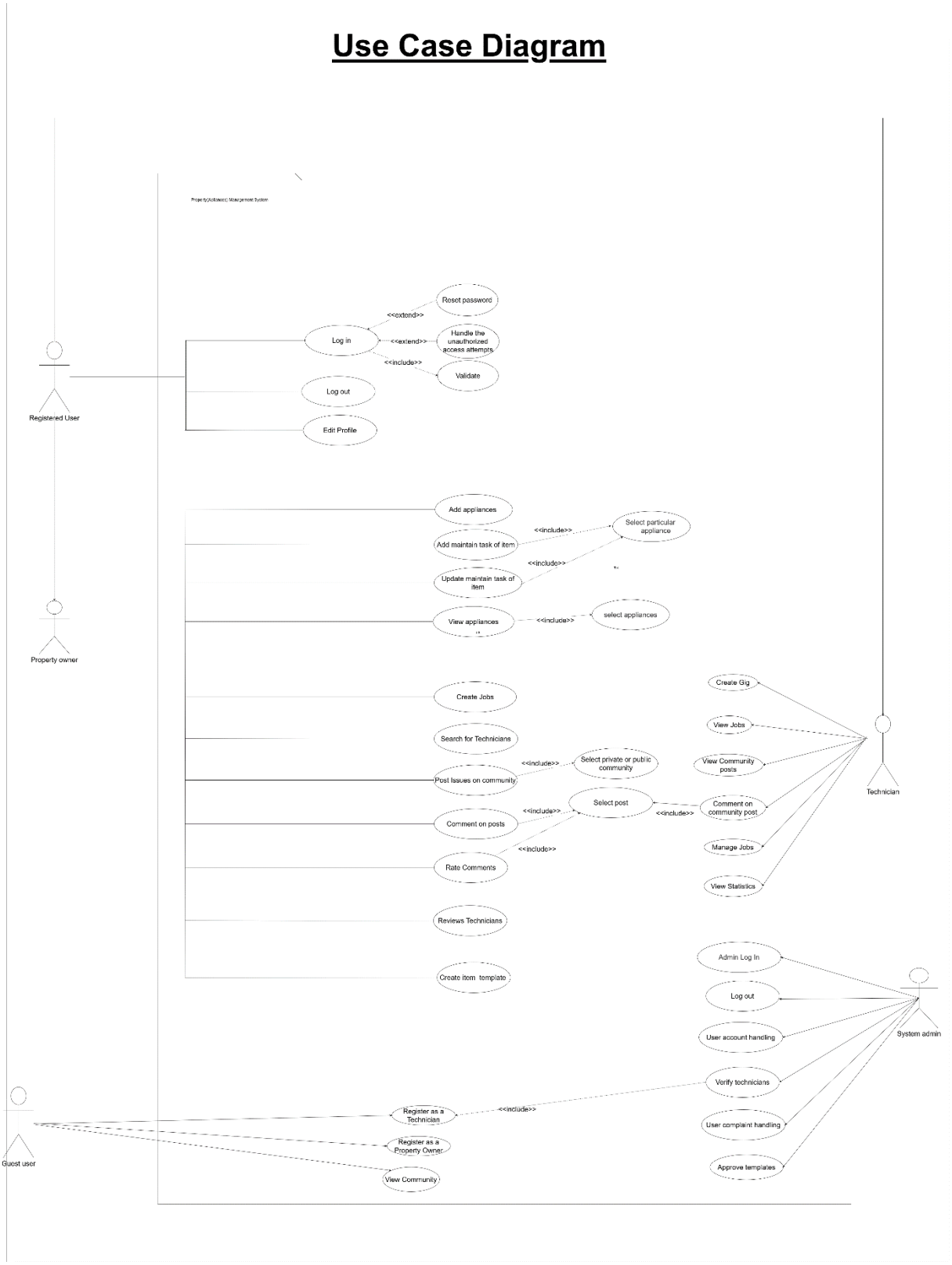
7.1.1 Component diagram

Figure 1: Component Diagram



7.1.2 Use case diagram

[Usecase-drawio link]



7.1.3 Use case Descriptions

Use case	Login
Description	Users can log into the system with his/her unique username and password.
Primary Actor	User
Secondary Actor	None
Pre-condition	
The users must be registered to the system.	
Main Flow	
1. Enter user name 2. Enter password 3. System checks and matches the username and password with its database. 4. If there is matching record, load the page.	
Alternative Event	
1. When the user forgot his/her password, can enter email address and renew the password. 2. When the username and password mismatch the system will give three chances to enter username and password. If the user fails to enter correct username and password then, the system temporary lock the account.	
Post- condition	
User can log into the system and use the system.	

Use case	Log out
Description	To avoid unauthorized access, users can log out from the system after using the system.
Primary Actor	User
Secondary Actor	None
Pre-condition	
Users must be already logged into the system.	
Main Flow	
1. Click the log out button. 2. System displays the confirmation message. 3. If the user still wants to log out from the system, then the system loads the home page.	
Alternative Event	
If the user accidentally clicks the log out button, he/she can go to the previous state before request to log out.	
Post- condition	
Display home page.	

Use case	Edit Profile
Description	Users can edit their profiles according to their needs such as change password, profile picture.
Primary Actor	User
Secondary Actor	None
Pre-condition	
The users must be logged into the system.	
Main Flow	
1. Users log into the system. 2. Go to profile customization. 3. Select the needed option. 4. Make the changes. 5. Save the changes.	
Alternative Event	
Don't save the modification.	
Post- condition	
Users can get customized profiles.	

Use case	Admin Login
Description	Admin can log into the system using username and password.
Primary Actor	Admin
Secondary Actor	None
Pre-condition	
-	
Main Flow	
1. Enter user name 2. Enter password 3. System checks and matches the username and password with its database. 4. If there is matching record, load the page.	
Alternative Event	
When the username and password mismatch the system will give chances to re-enter username and password.	
Post- condition	
Admin can log into the system and use the system.	

Use case	Admin Log out
Description	To avoid unauthorized access, admin can log out from the system after using the system.
Primary Actor	Admin
Secondary Actor	None
Pre-condition	
Admin must be already logged into the system.	
Main Flow	
1. Click the log out button. 2. System displays the confirmation message. 3. If the admin still wants to log out from the system, then the system loads the home page.	
Alternative Event	
If the admin accidentally clicks the log out button, he/she can go to the previous state before request to log out.	
Post- condition	
Display home page.	

Use case	User account management
Description	Admin can solve user account issues such as remove user account, if user requested to changes in profile.
Primary Actor	Admin
Secondary Actor	None
Pre-condition	
The admin must be logged into the system.	
Main Flow	
1. Select the specific user account. 2. If the admin is getting to know that user account is fake, remove the user account and notify user about the deletion. 3. If the user requested to do some changes in user profile such as email address, admin makes changes.	
Alternative Event	
Post- condition	
Admin should be able to make sure users are genuine in the system.	

Use case	Add item
Description	This use case describes the details about the types of items entered by the user into the system. User can add items with its necessary details for the system and in adding process. User can enter the details of the items according to the categorization.
Primary Actor	User
Secondary Actor	None
Pre-condition	
• User must be logged into the system	
Main Flow	
1. Go to the Item tab. 2. Click on the item. 3. Click add items. 4. Users enter item information into the provided form. 5. User clicks on the enter button. 6. User clicks the confirm button.	
Alternative Event	
None	
Post- condition	
• User gets an overview of the added item details.	

Use case	Add maintain task
Description	This use case is used to add maintenance for a corresponding item(s) for the system by the user and in the adding process.
Primary Actor	User
Secondary Actor	None
Pre-condition	
• User must be logged into the system • Users must have already created an item or an item.	
Main Flow	
1. Users navigate to the items maintenance section. 2. Click on the add maintenance task button. 3. Users enter the information about the maintenance task. 4. Click on the add button. 5. User confirm the provided preview.	
Alternative Event	
• Navigate to the item section. • Select a item or item under a sub item.	
Post- condition	
• Reminders will be created according to the task	

Use case	Update maintain task
Description	This use case is used to update the maintenance of an item(s). User can view and delete each item
Primary Actor	User
Secondary Actor	None
Pre-condition	
• User must be logged into the system	
Main Flow	
1. Users navigate to the items maintenance section. 2. Click on the update maintenance task button. 3. System will show the information about the task(s). 4. Click on the update button. 5. Users confirm the provided preview.	
Alternative Event	
None	
Post- condition	
• Show the changing things according to the task.	

Use case	View Item
Description	This use case describes the event of the item details. User can update item details and reminder details about maintenance of the corresponding item. User can view and delete each item. System updates changes which are database records, details of notification process.
Primary Actor	User
Secondary Actor	None
Pre-condition	
• Users must be logged into the system. • Users must have already created a item.	
Main Flow	
1. User navigates item section. 2. User do update/delete/view item details. 3. Confirm the process. 4. System updates the database records and details of notification process. .	
Alternative Event	
None	
Post- condition	
None	

Use case	Create Jobs
Description	User can create jobs according to their maintenance items.
Primary Actor	User
Secondary Actor	Technician
Pre-condition	
The user must be logged into the system and has a maintainable item to maintain.	
Main Flow	
• Users navigate to the create jobs section. • System loads the current job roles and adds job roles features. • User enter the job role and item name according to his maintainable item • Select technicians. • System update the database and update job roles to the technicians.	
Alternative Event	
• Navigate to the update or create maintenance tasks section.	
Post- condition	
• View the current created job roles. • System will generate notification to the corresponding technicians	

Use case	Search for technicians
Description	Users can search technicians for their item maintenance.
Primary Actor	User
Secondary Actor	None
Pre-condition	
The user must be logged into the system and has a maintainable item to maintain.	
Main Flow	
1. Users navigate to the search technicians section. 2. System loads select location feature. 3. User enters his location. 4. System validates the location. 5. If it is correct, the system requests to enter the item name. 6. If not, the user can re-enter the location. 7. Users should enter the item name. 8. System suggests the service providers.	
Alternative Event	
Users will be able to navigate to maintain tasks sections.	
Post- condition	
View the service providers list.	

Use case	Post issues
Description	User can post issues that are connected to their item maintenance.
Primary Actor	User
Secondary Actor	None
Pre-condition	
The user must be logged into the system and has a maintainable item to maintain.	
Main Flow	
1. Users navigate to the community section. 2. System loads the community interface. 3. Users select the choice. 4. Users enter issues details. 5. System shows a preview of the post. 6. System asked, edit or not the post. 7. If yes, navigate to re-enter issues. 8. If not, the user can click the submit button. 9. System sends data to the database. 10. System publishes posts in the community.	
Alternative Event	
None	
Post- condition	
View posts in the community.	

Use case	Comments on post
Description	User can comment on posts that are published by other users in the community.
Primary Actor	User
Secondary Actor	None

Pre-condition
The user must be logged into the system.
Main Flow
1. User navigate to the community tab. 2. System loads the community interface. 3. User clicks on the post. 4. System loads the post. 5. User clicks on the comment button. 6. User enters the comment. 7. System asked, edit or not the comment. 8. If yes, navigate to re-enter comment. 9. If not, the user can click the submit button. 10. System sends data to the database. 11. System notified the owner of the post.
Alternative Event
None
Post- condition
• View posts and comments in the community.

Use case	Rate comments
Description	User (owner of the post) can rate comments on posts that are published by other users in the community.
Primary Actor	User
Secondary Actor	None
Pre-condition	
The user must be logged into the system.	
Main Flow	
1. User navigate to the community tab. 2. System loads the community interface. 3. User clicks on the post. 4. System loads the post. 5. User clicks on the comment button. 6. System loads the comments. • User give ratings. • System send data to the database. • System notified the owner of the comment.	
Alternative Event	
None	
Post- condition	
• View posts and comments, ratings in community.	

Use case	Rate technicians
Description	User can rate technicians considering their work.
Primary Actor	User
Secondary Actor	Technician
Pre-condition	
The user must be logged into the system.	
Main Flow	
1. User navigate to the rate technician tab. 2. System loads interface. 3. User clicks give ratings. 4. System save changes in database. 5. System show the ratings. 6. System notify the corresponding technician.	
Alternative Event	
• View comments about technicians.	
Post- condition	
7. View ratings on technicians.	

Use case	Create item template
Description	User can create a default template for the special item to add their item.
Primary Actor	User
Secondary Actor	None
Pre-condition	
The user must be logged into the system.	
Main Flow	
1. User navigate to item section. 2. Click add new Item. 3. Click new template 4. Add details about the item. 5. System asks for confirmation.	
Alternative Event	
• View comments about technicians.	
Post- condition	
• User can request to add their own item template to the Administrator to add the system to use to other users	

Use case	Create Gig
Description	Technician is able to add a brief description about job and service him/her offer.
Primary Actor	Technician
Secondary Actor	
Pre-condition	
• User must be logged into the system.	
Main event	
1. User navigates the service section 2. Add enter all the necessary details. 3. Confirm the process. 4. Service gigs are published by the system..	
Alternative Event	
• If the technician submits form with important information are empty, then refill massage will be shown. • If the user submits invalid information an Error massage will be shown. • Maximum number of services reached	
Post- condition	
• Gigs will be shown in the corresponding technician profile.	

Use case	View jobs
Description	When the Technician publishes their Service Gigs, User can send a request for a job. In Technician's interface, in the requests section, all the job requests are displayed in the order of the requests that are sent.
Primary Actor	User
Secondary Actor	None
Pre-condition	
• User must be logged into the system.	
Main event	
1. Navigate to the requests section. 2. View the requests received so far. 3. Click and see the details of each of the requests. 4. Able to accept or reject requests. 5. If accept the job, it will show in job section.	
Alternative Event	
None	
Post- condition	
• If technician accepts request, system will send notification to the particular user.	

Use case	Manage Jobs
Description	This use case describes the event of managing technician's jobs. Technician can accept or reject the jobs and if accept a job technician can schedule the jobs according to the user requirements.
Primary Actor	Technician
Secondary Actor	User
Pre-condition	
• Technician must be logged into the system.	
Main event	
1. Navigate to the job section 2. View each jobs 3. Able to schedule the jobs 4. Able to accept or reject jobs	
Alternative Event	
• After two days accepting a job, technician cannot reject the job	
Post- condition	
• System will send notification to the particular user according to the technician changes which are done in schedule jobs.	

Use case	View Community posts
Description	This use case describes the event of the viewing of community. Technician can view community post items which are posted by the Users.
Primary Actor	Technicians
Secondary Actor	None
Pre-condition	
• Technicians must be logged system.	
Main Event	
1. User navigates to the community section 2. Select the items category 3. View community post	
Alternative Event	
Post- condition	
• User can comment the post.	

Use case	Comment on community post
Description	This use case describes the event of the commenting on the community post. After navigating to the community section Technician can select post and can comment on that post.
Primary Actor	Technician
Secondary Actor	User
Pre-condition	
• Technician must be logged into the system.	
Main event	
1. User navigates to the community section 2. Select the items category 3. View community post. 4. Select a post 5. give a comment to the post 6. confirm the process	
Alternative Event	
• Limited number of character can be included in the comment.	
Post- condition	
• After confirmation process, system will send a notification to user who posted the comment.	

Use case	Generate reports
Description	This use case describes the event of generating reports which includes previously done jobs, earning details according to the daily/monthly/ yearly, number of orders, number of cancel orders etc.
Primary Actor	Technician
Secondary Actor	None
Pre-condition	
• User must be logged into the system.	
Main event	
1. User navigates report generate section. 2. User select type of the report. 3. Confirm process 4. System generate the report	
Alternative Event	
Post- condition	
• User will be able to download the report	

Use case	Verify Technicians
Description	This use case describes the event that admin verifies the technician when a signup request came from a technician. When a technician is signed up to the system they need to get verify by an admin to continue to use the system features.
Primary Actor	Admin
Secondary Actor	Technician
Pre-condition	
• Admin must be signed up to the system • There should be signup requests	
Main Flow	
1. Navigates to the signup request section 2. Click on the request 3. Mark the reviewed details 4. If all details are verifiable click on verify 5. Confirm he verification	
Alternative Event	
4.1. If details are unverifiable, click on request resubmission.	
Post- condition	
• The relevant Technician will be notified with the verification status. • Admin will be redirected to the request section	

Use case	View Community
Description	Unregistered users will be able to view the community feed where the registered users post their issues with regard to the properties.
Primary Actor	Guest(Unregistered user)
Secondary Actor	None
Pre-condition	
None	
Main Flow	
1. Visit the URL of the system. 2. Navigate to the community section. 3. System will show the interface with posts. 4. Click on the post. 5. View post.	
Alternative Event	
None	
Post- condition	
• System will show the posts	

1.1.4 Activity Diagram

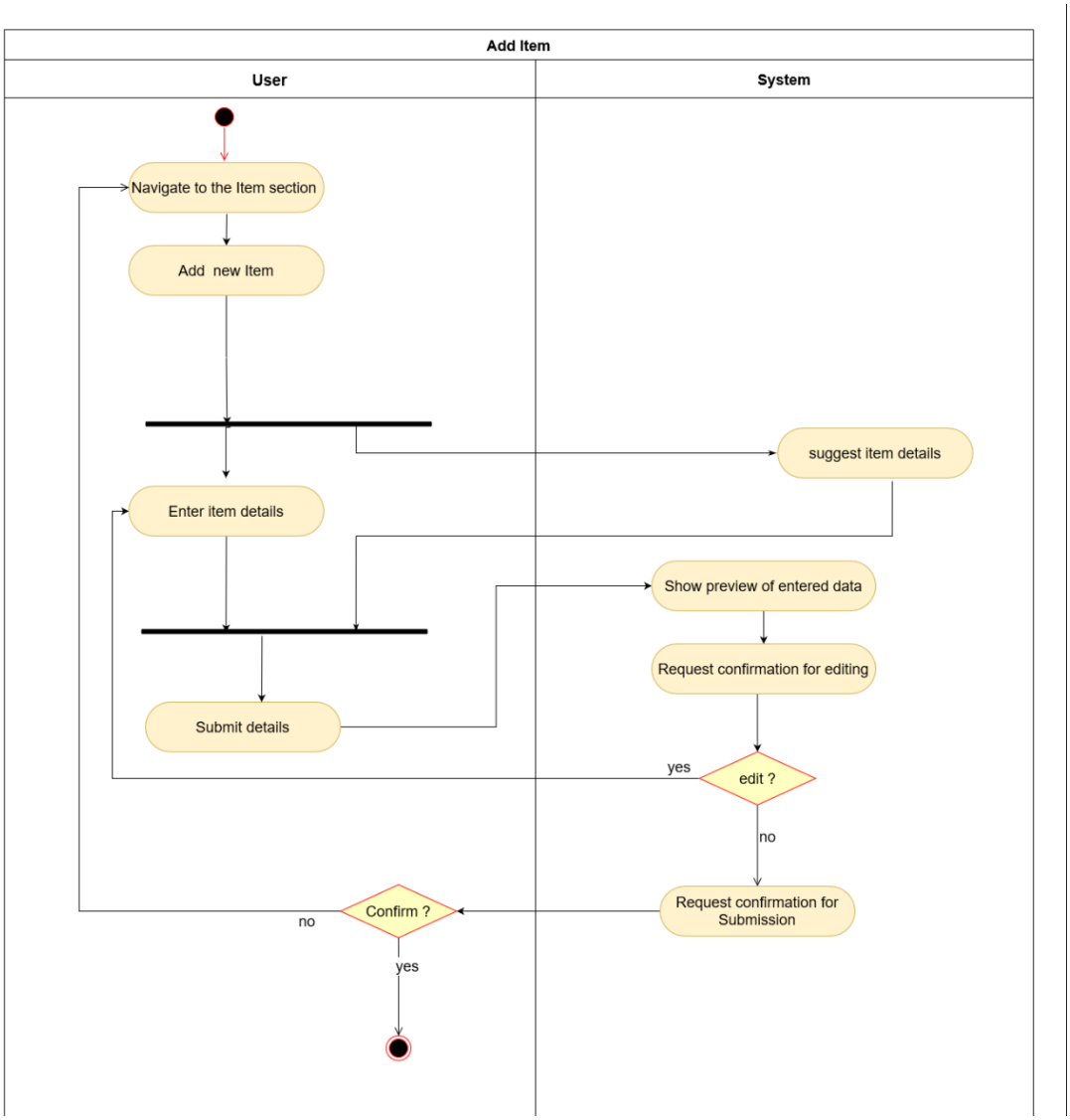


Figure 1 : Add Item

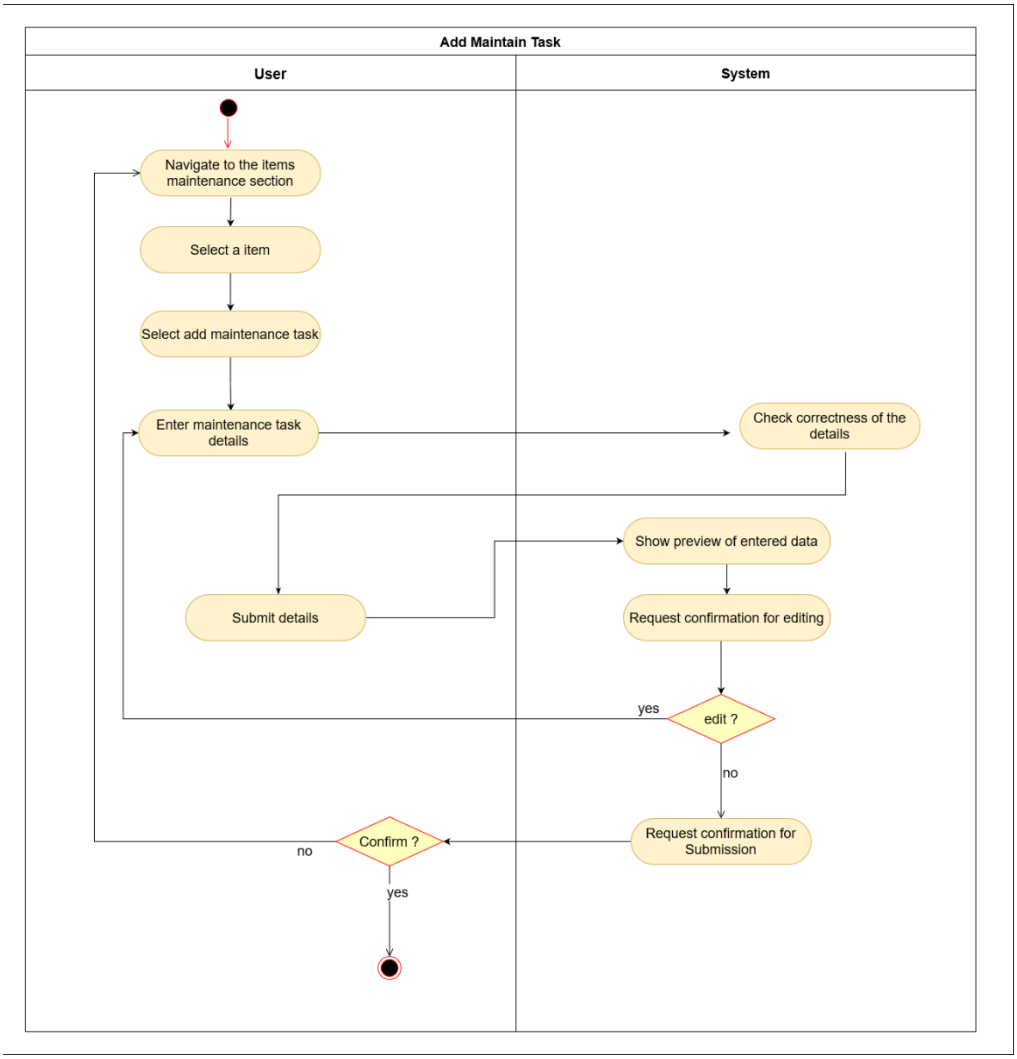


Figure 2 : Add maintain Task

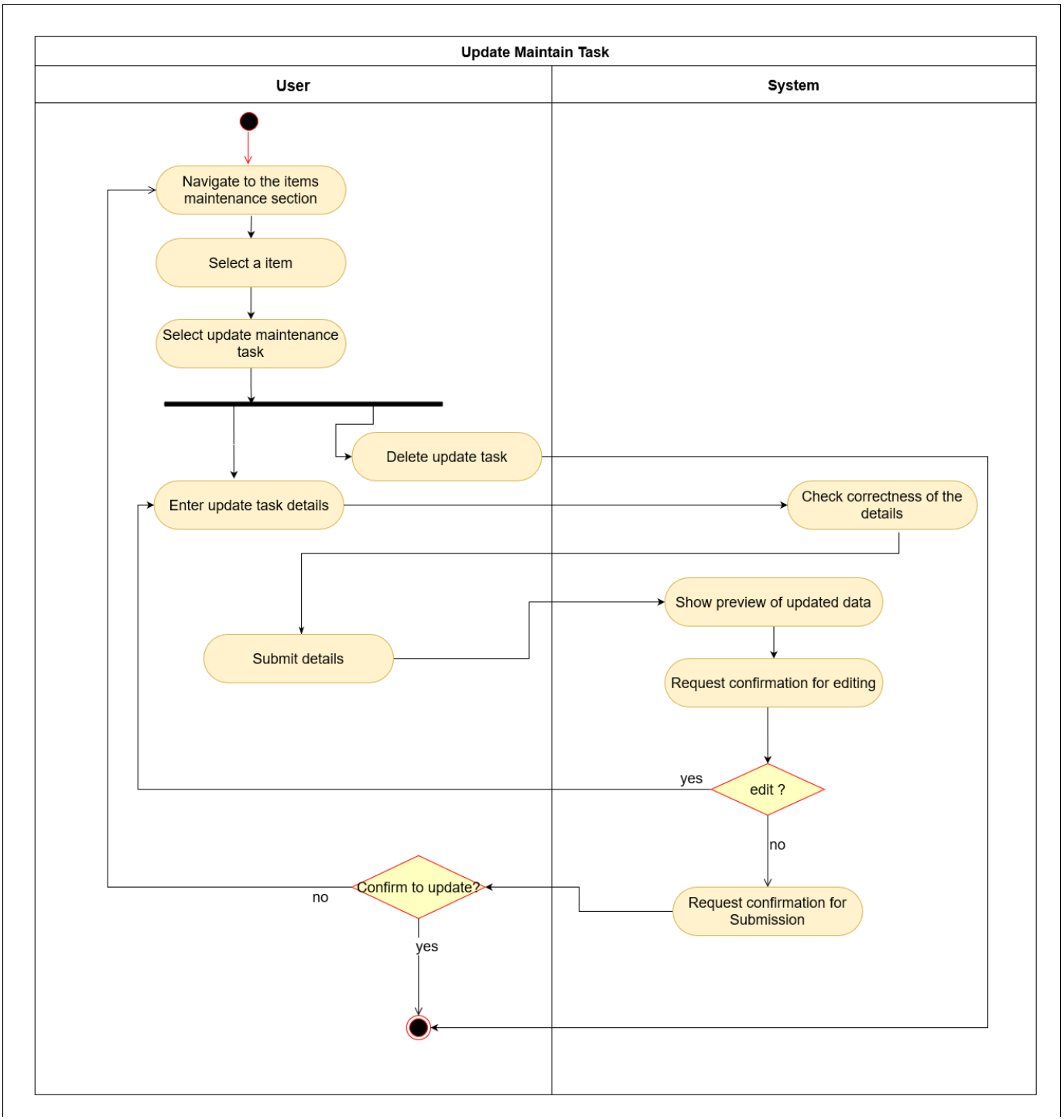


Figure 3 : Update Maintain Task

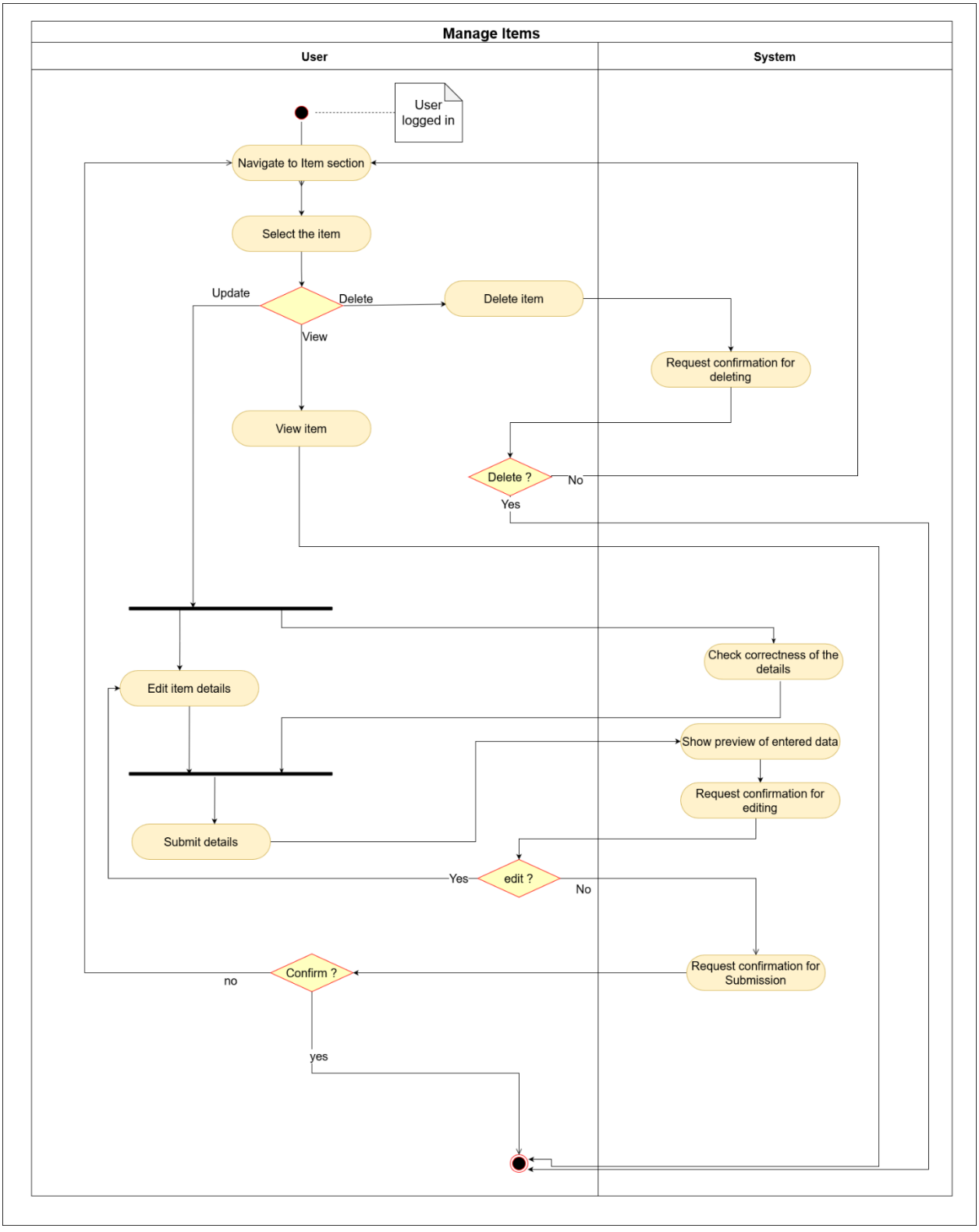


Figure 4: Manage Item

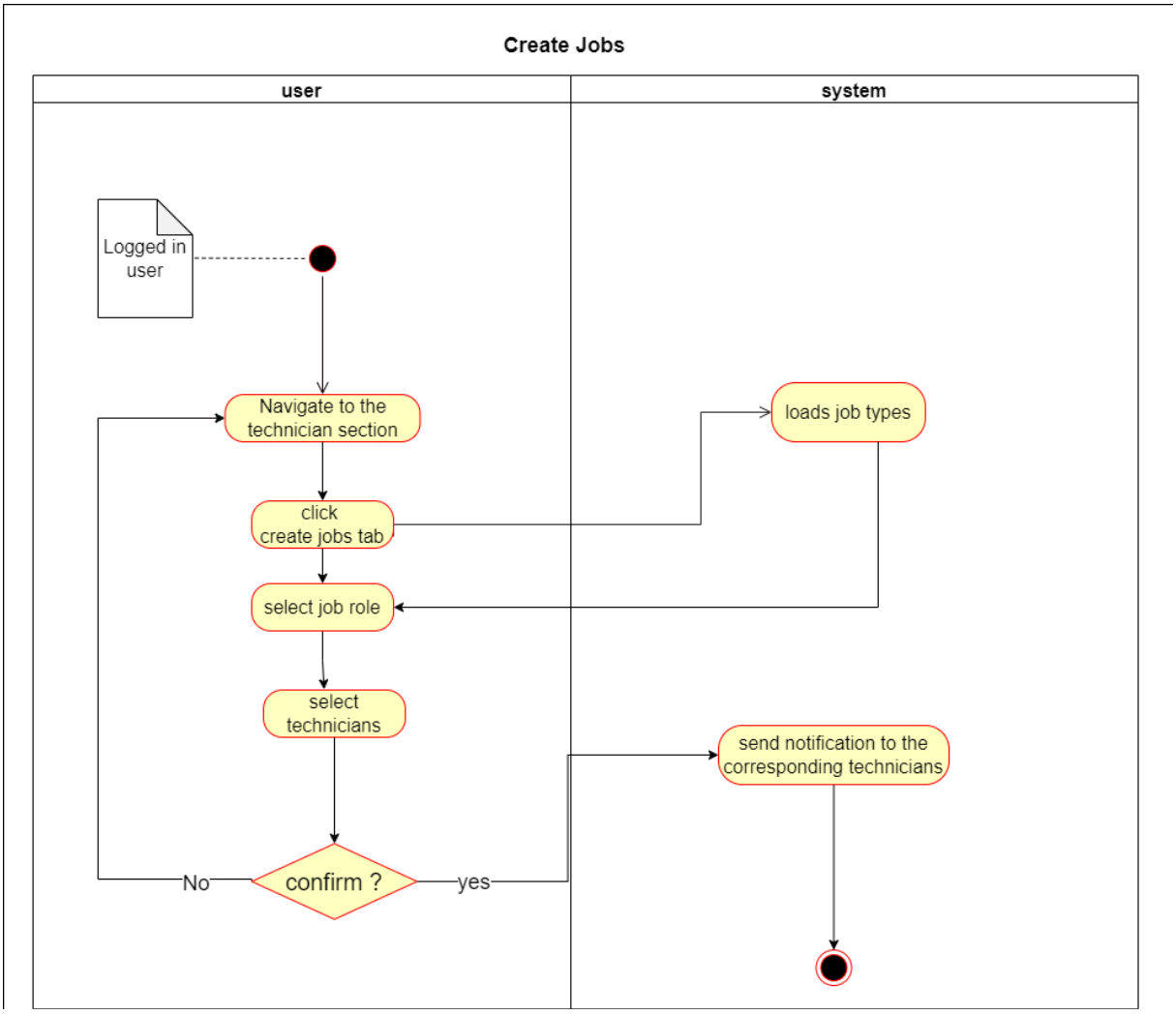


Figure 5 : Create Jobs

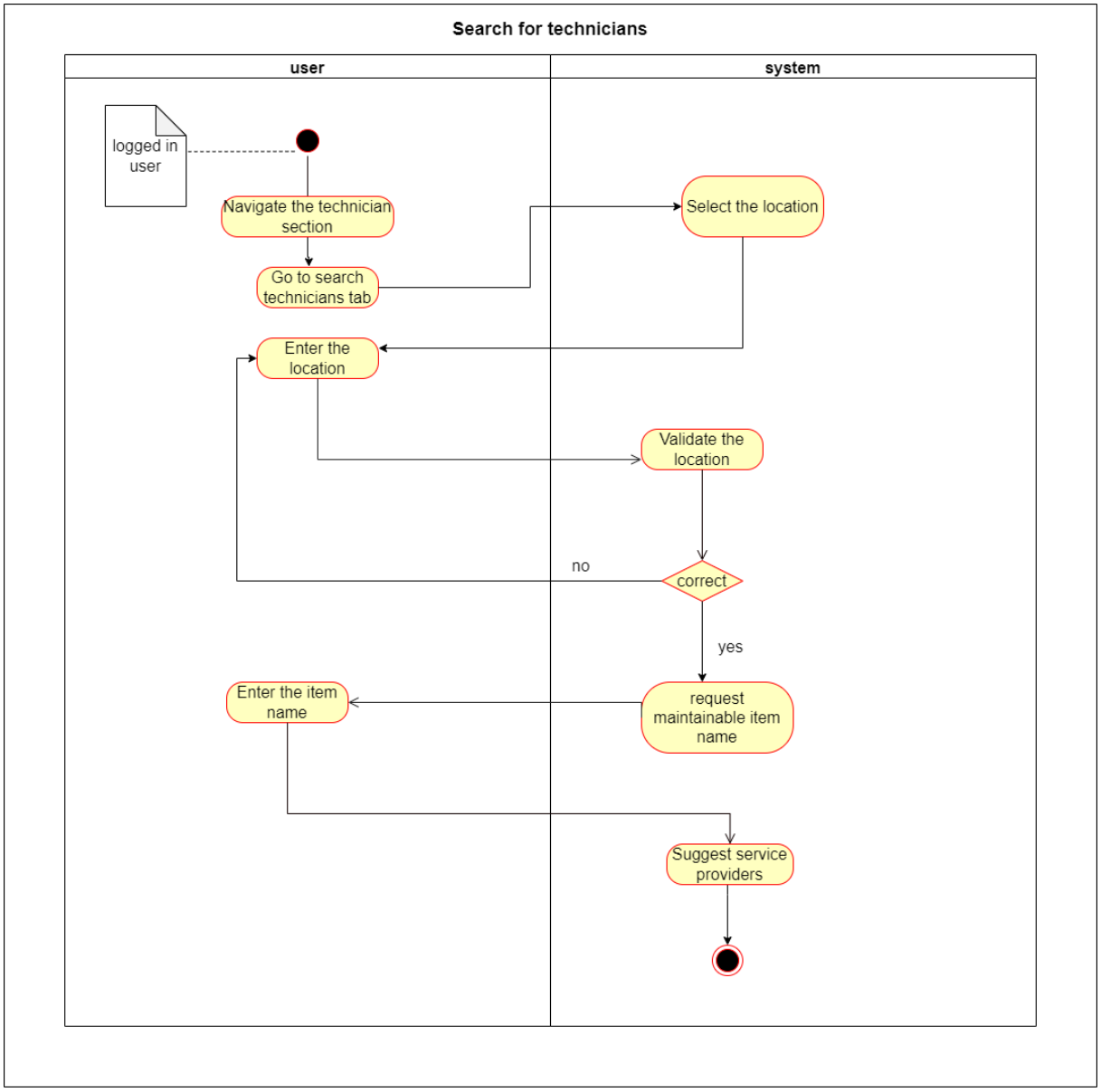


Figure 6 : Search for Technician

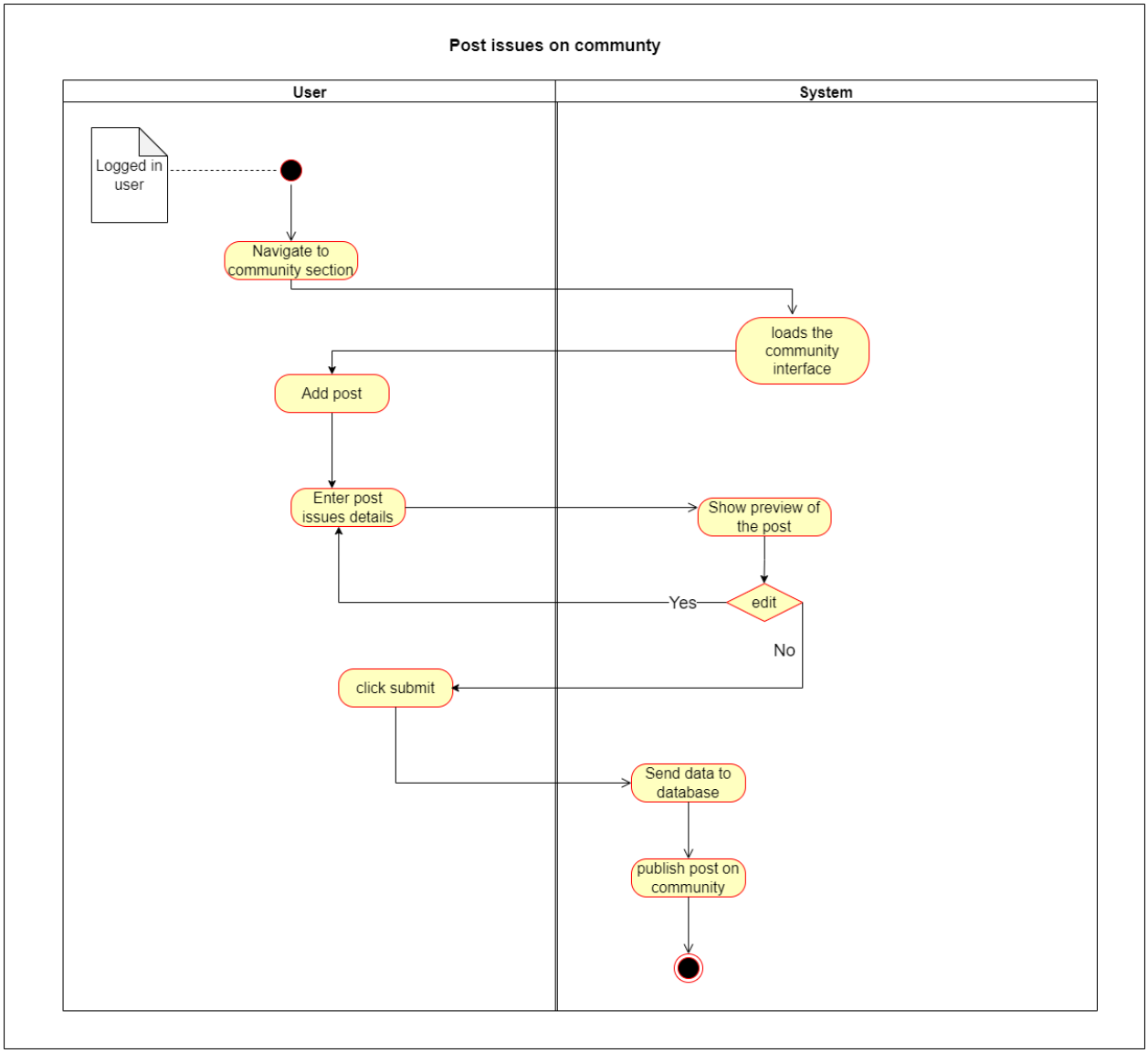


Figure 7 : Post Issues on Community

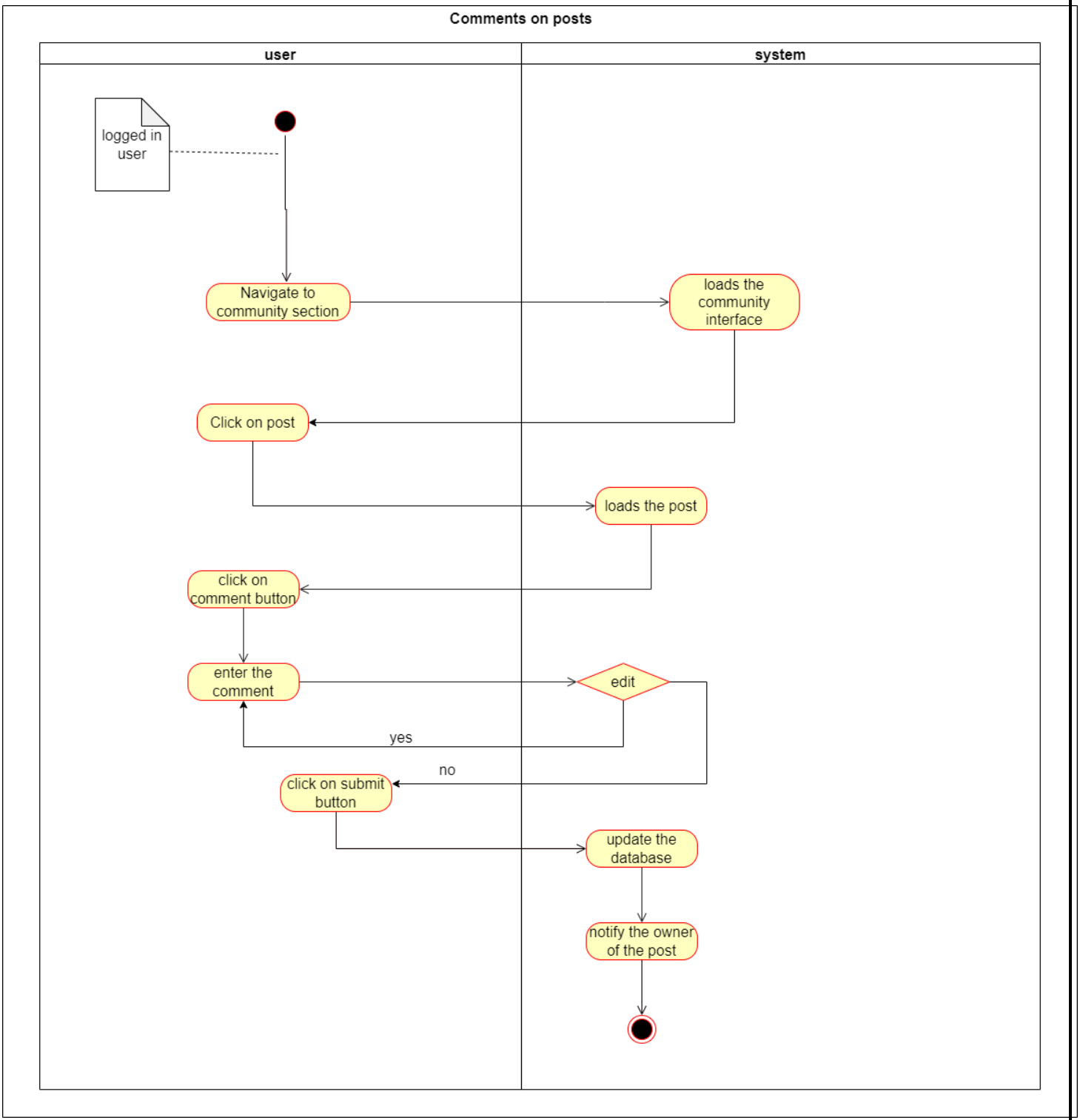


Figure 8 : Comment on posts

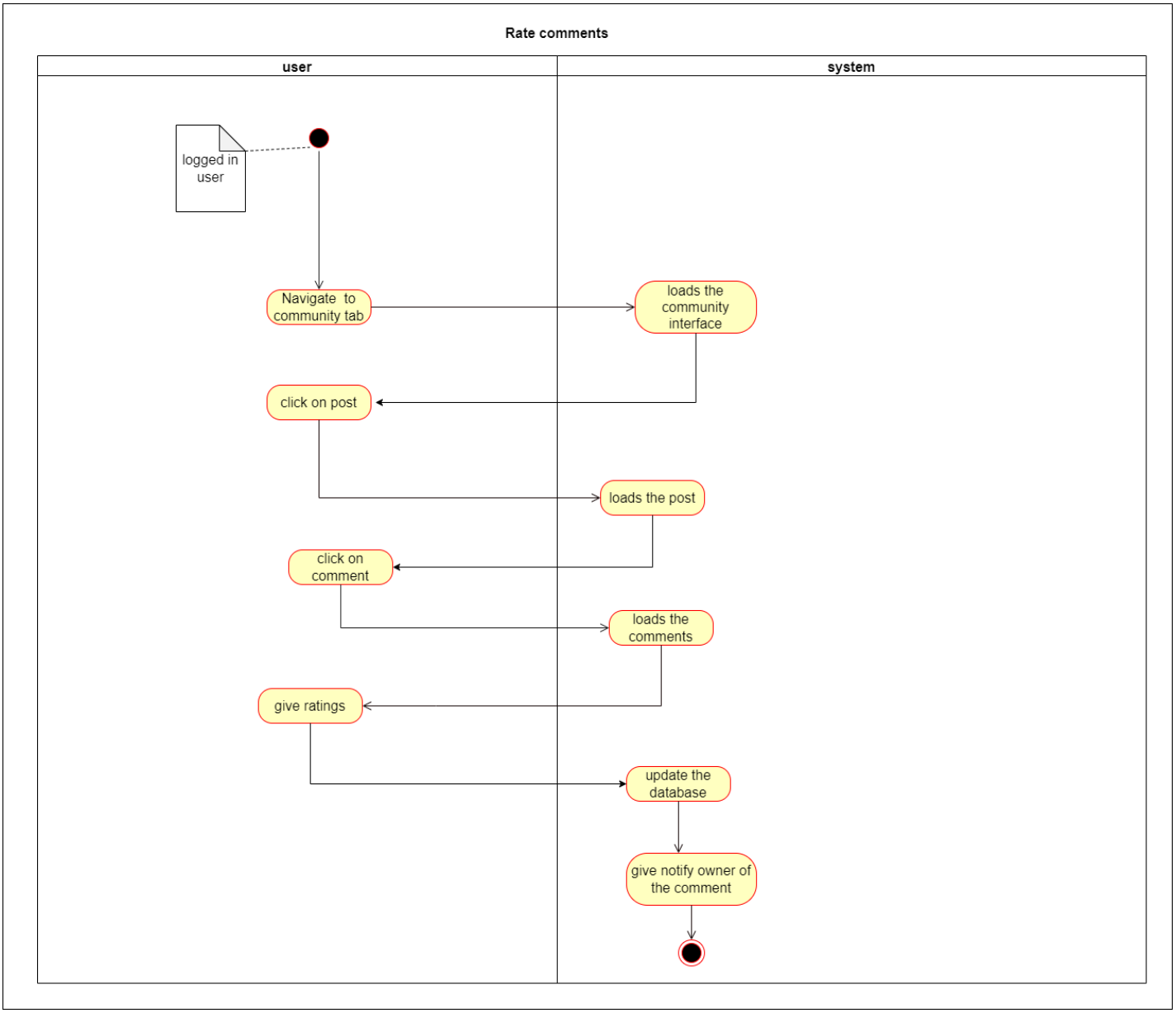


Figure 9: Rate Comments

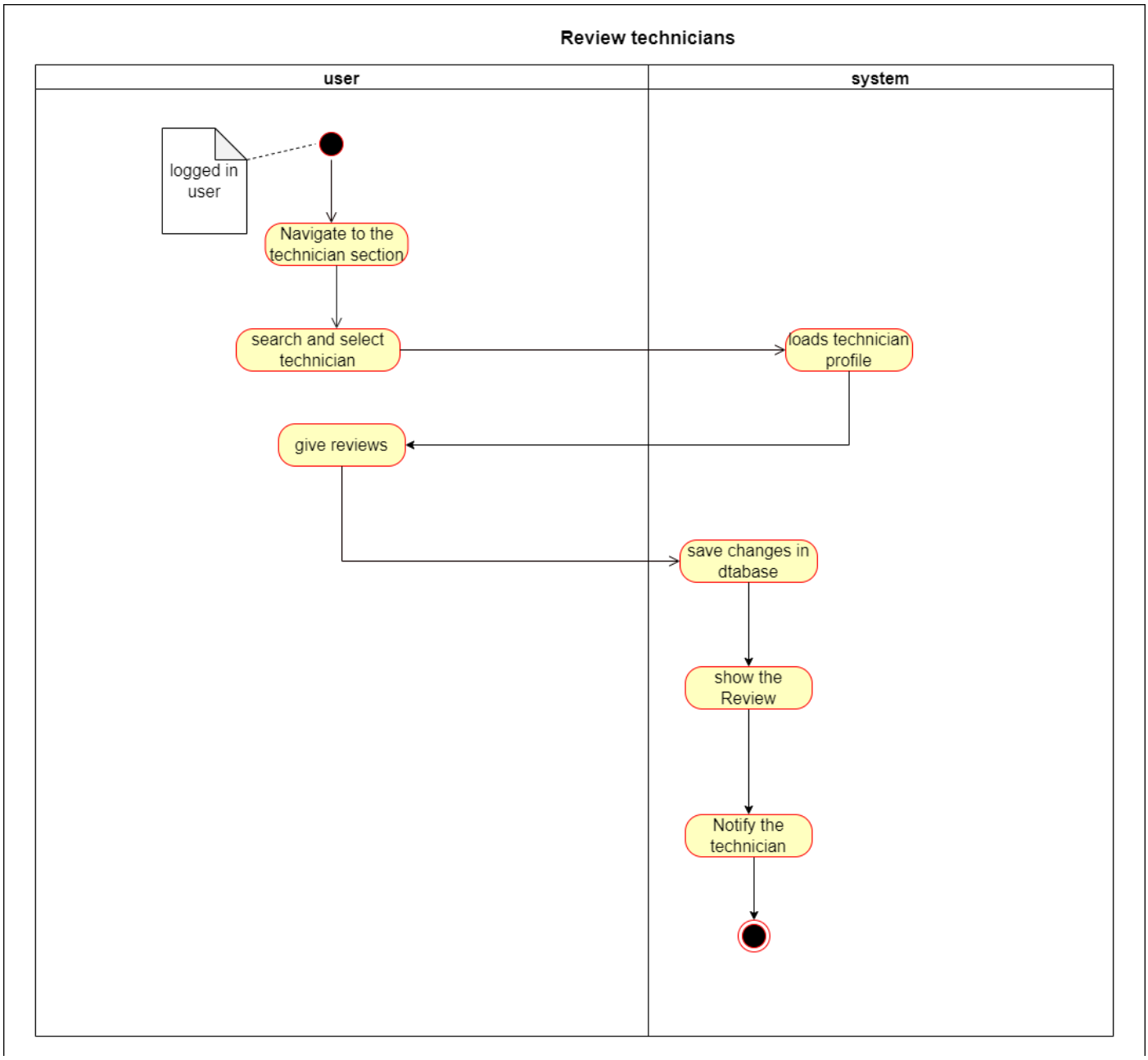


Figure 10 : Review Technicians

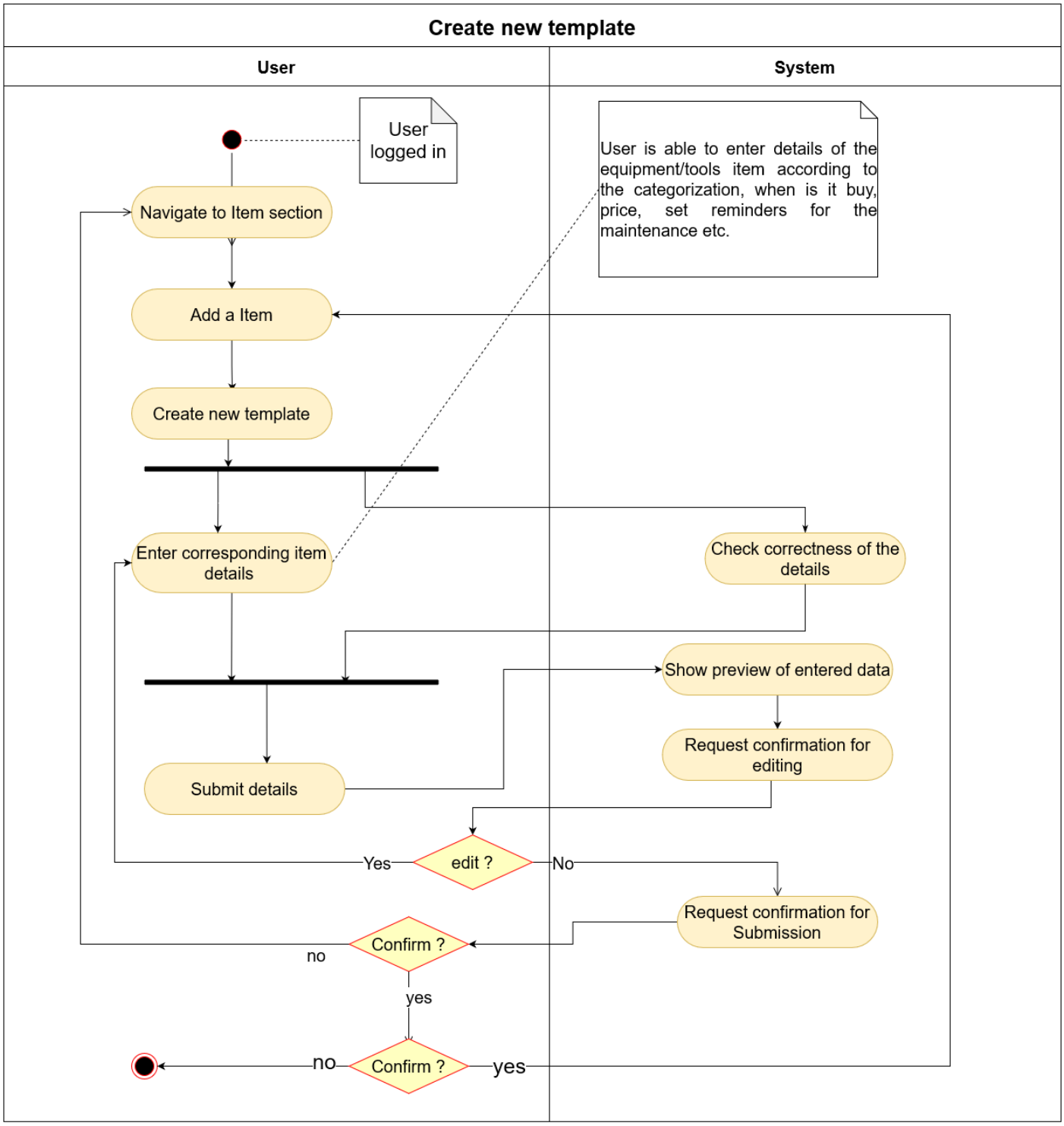


Figure 11 : Create new Templates

Figure 12 : Create Gigs

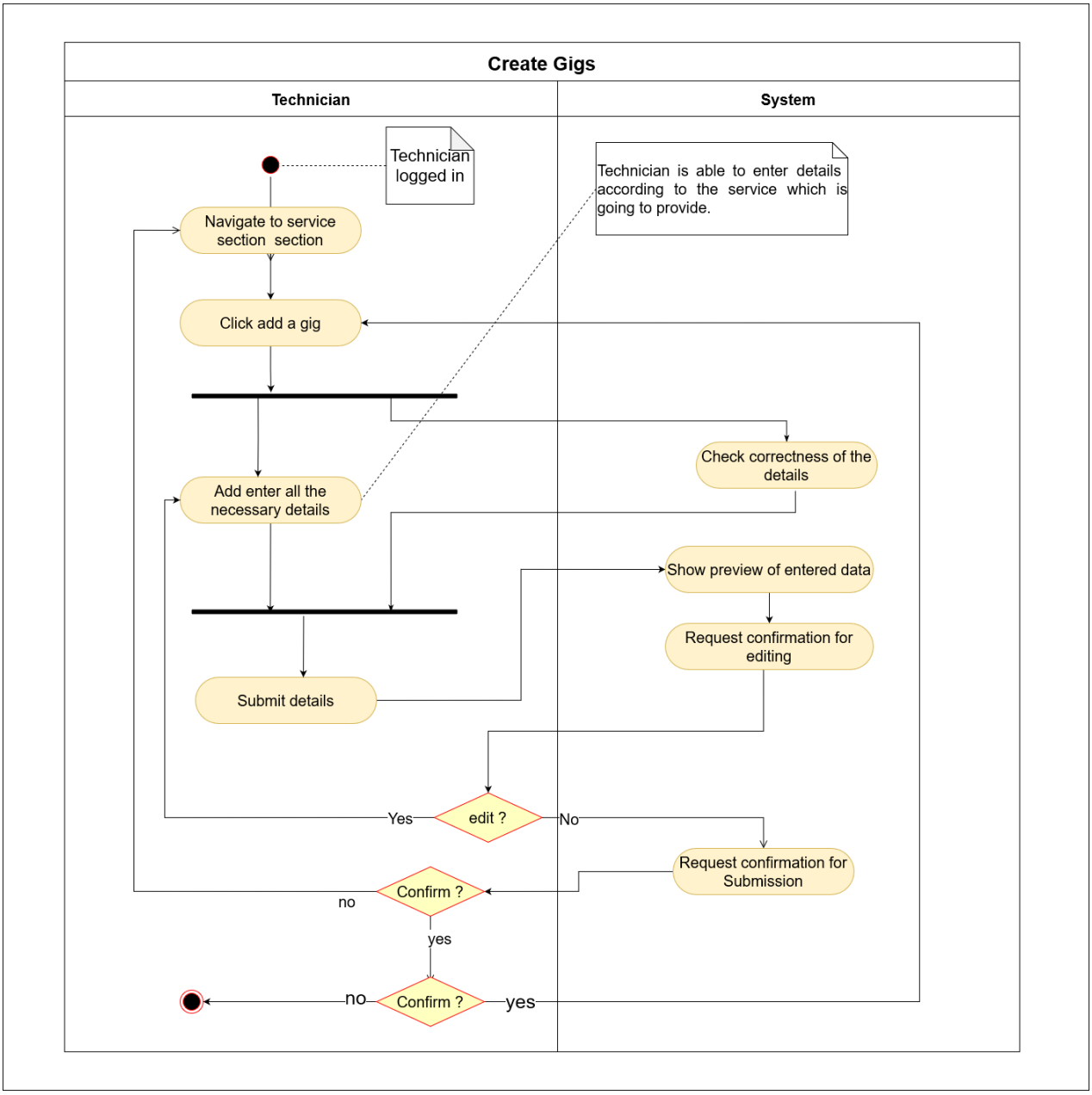


Figure 13 : View Jobs

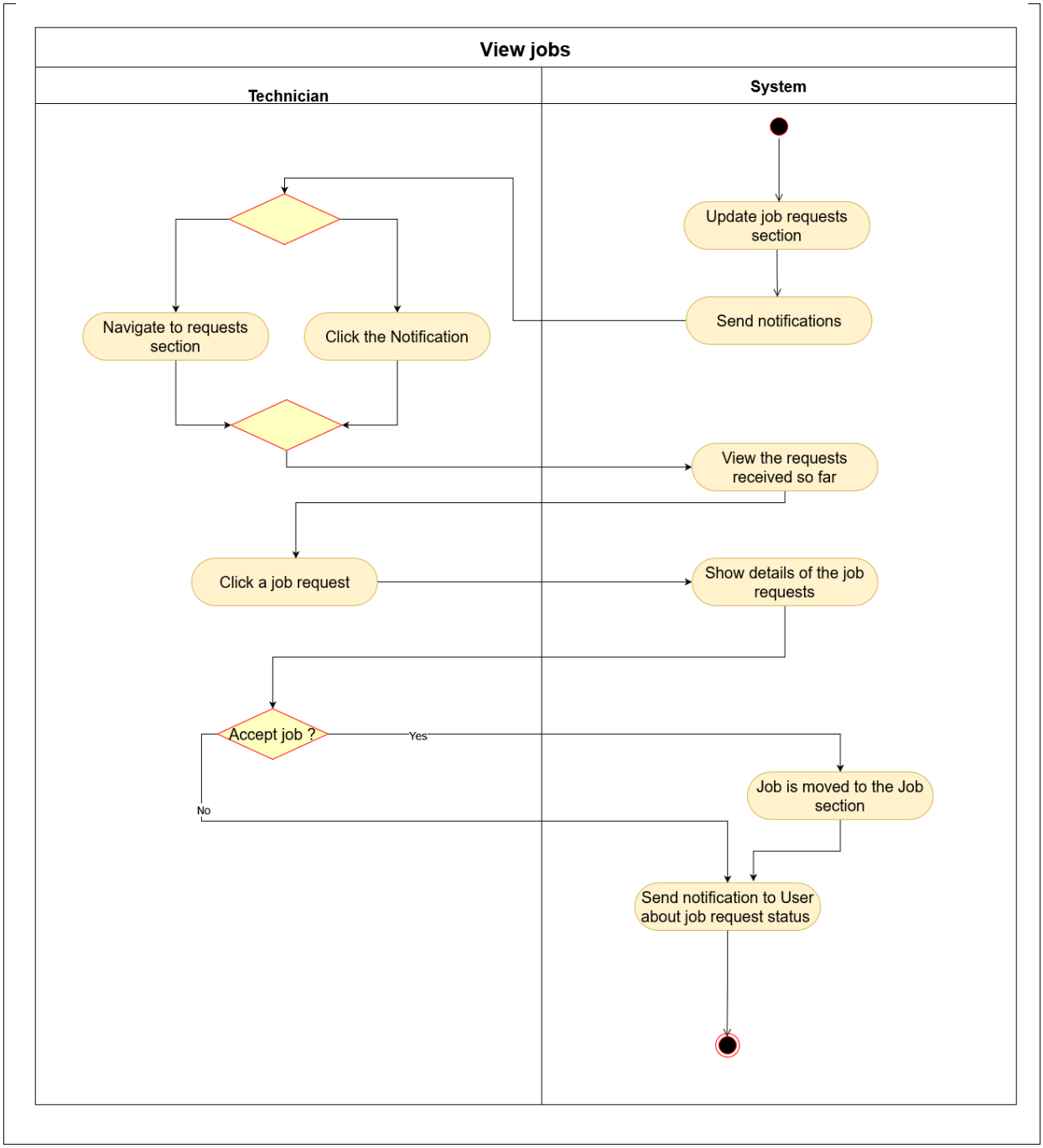


Figure 14 : Manage Jobs

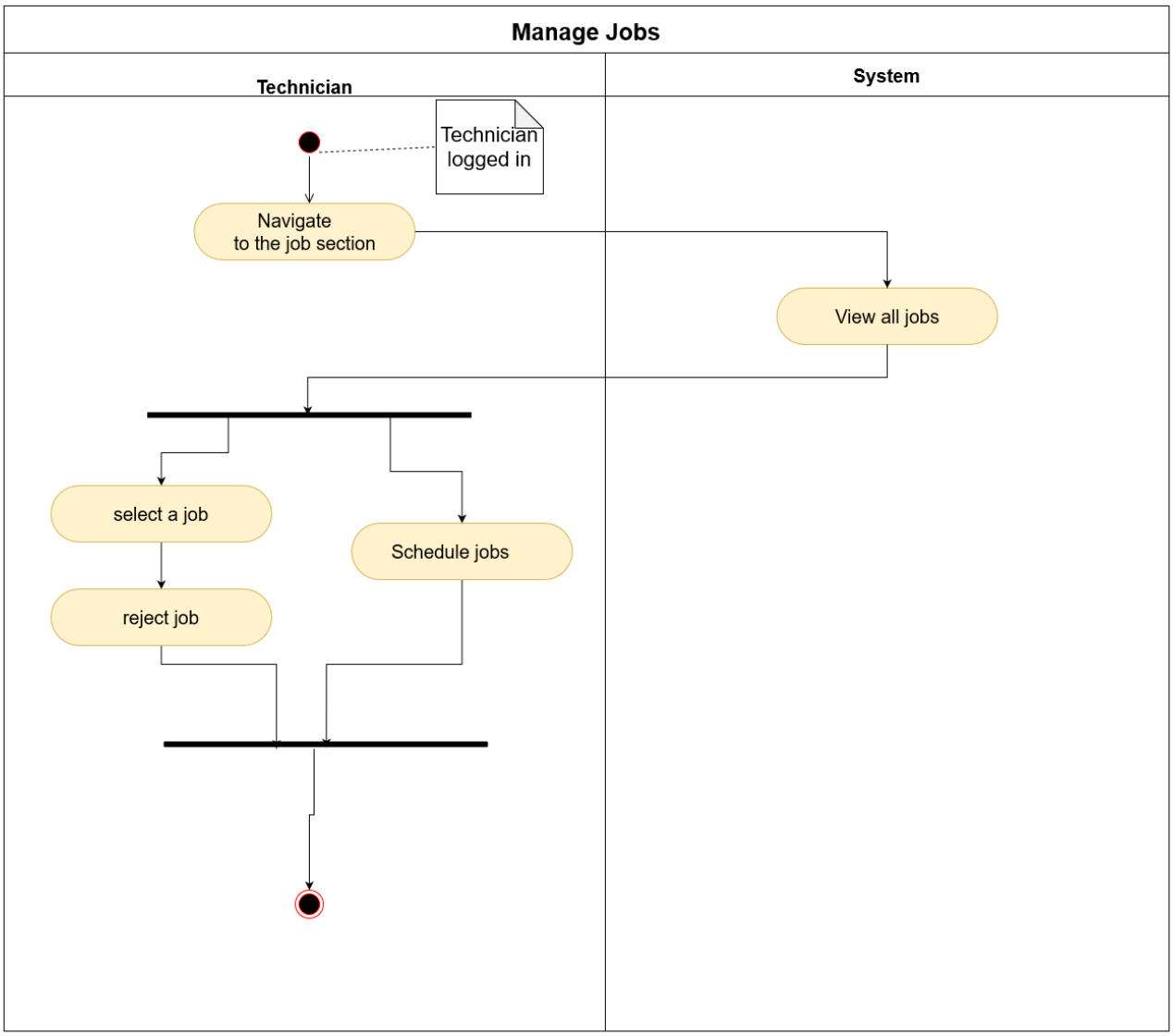


Figure 15 : View Community Posts

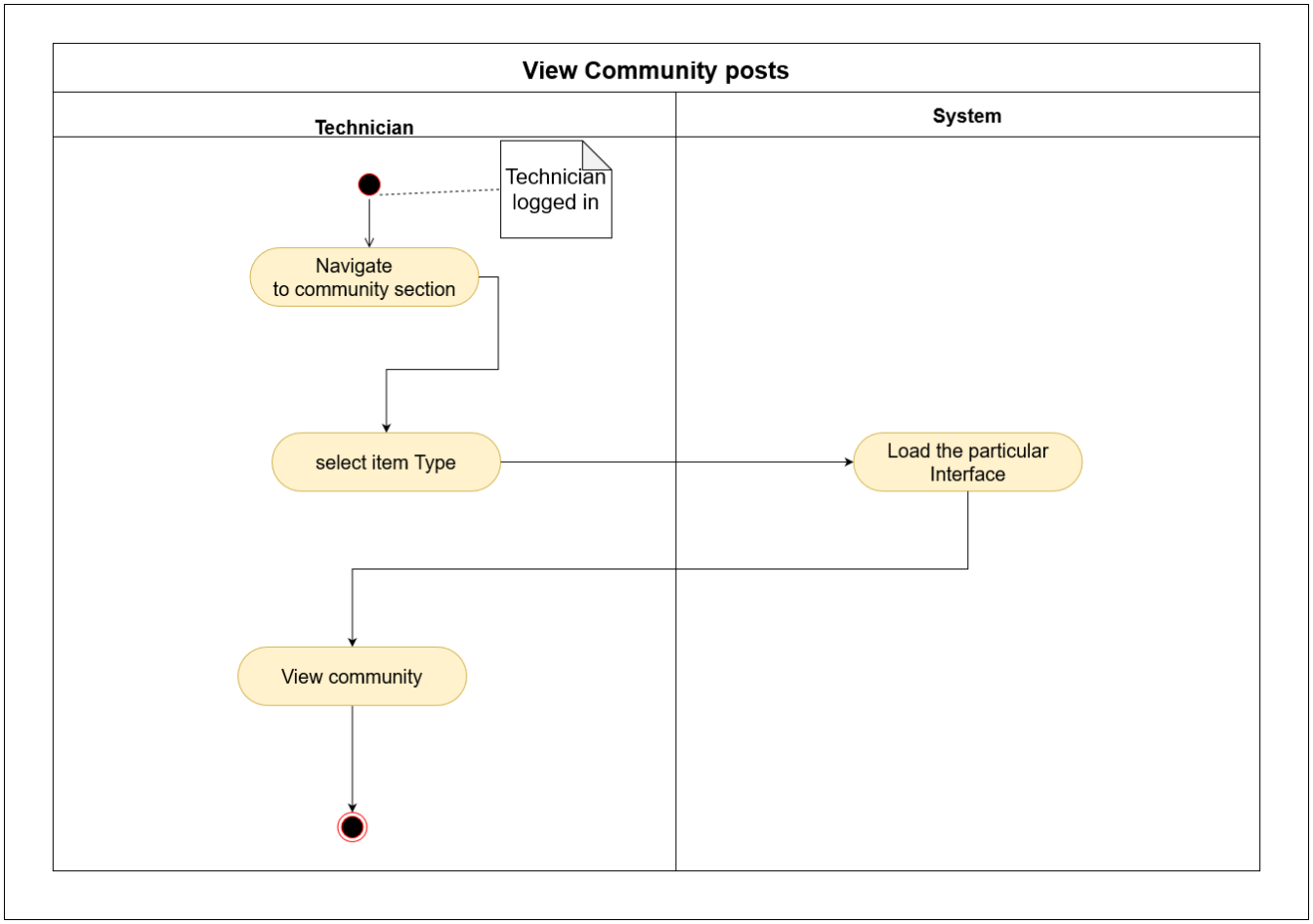


Figure 16 : Comment on Community post

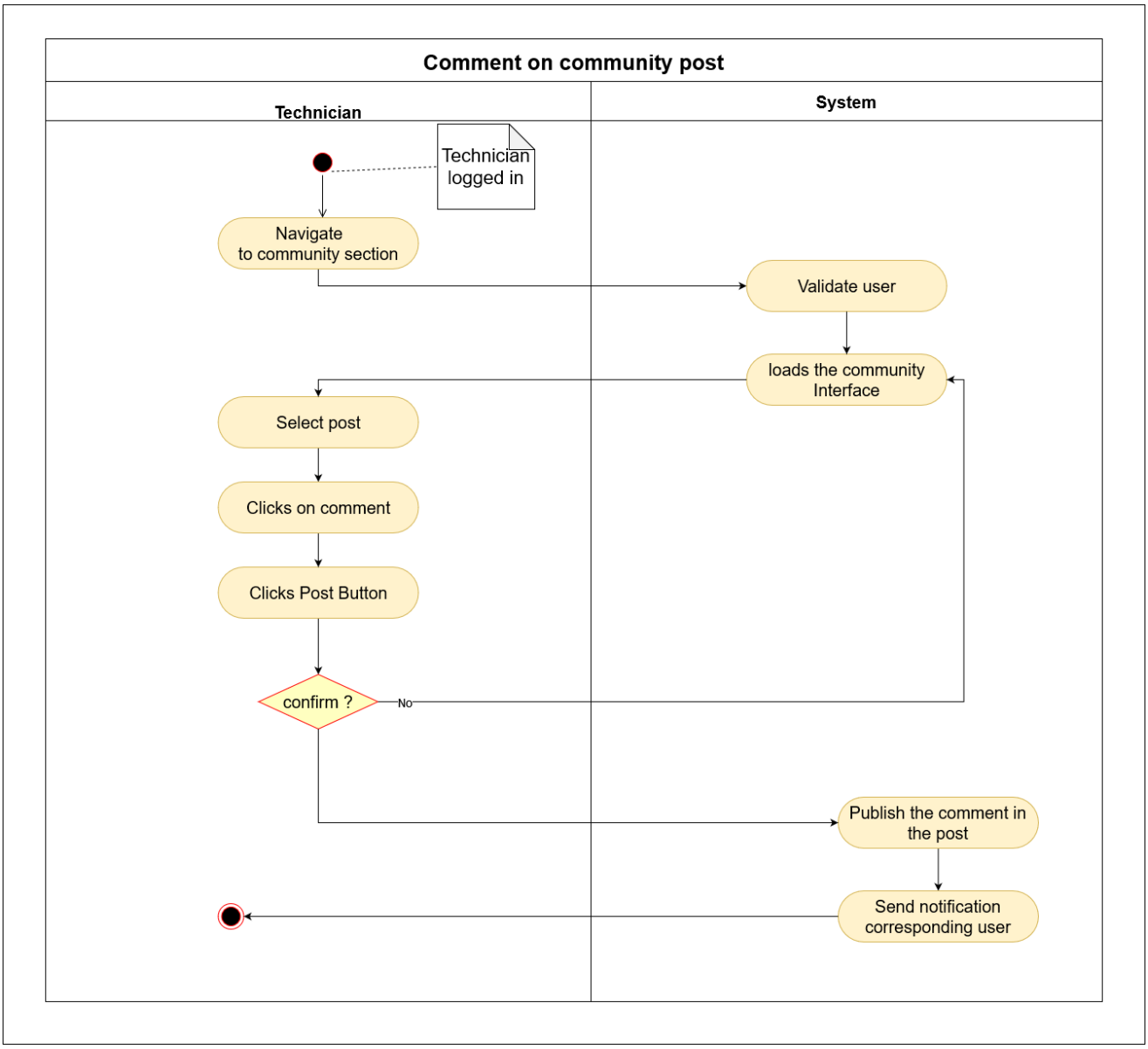


Figure 17 : Generate Reports

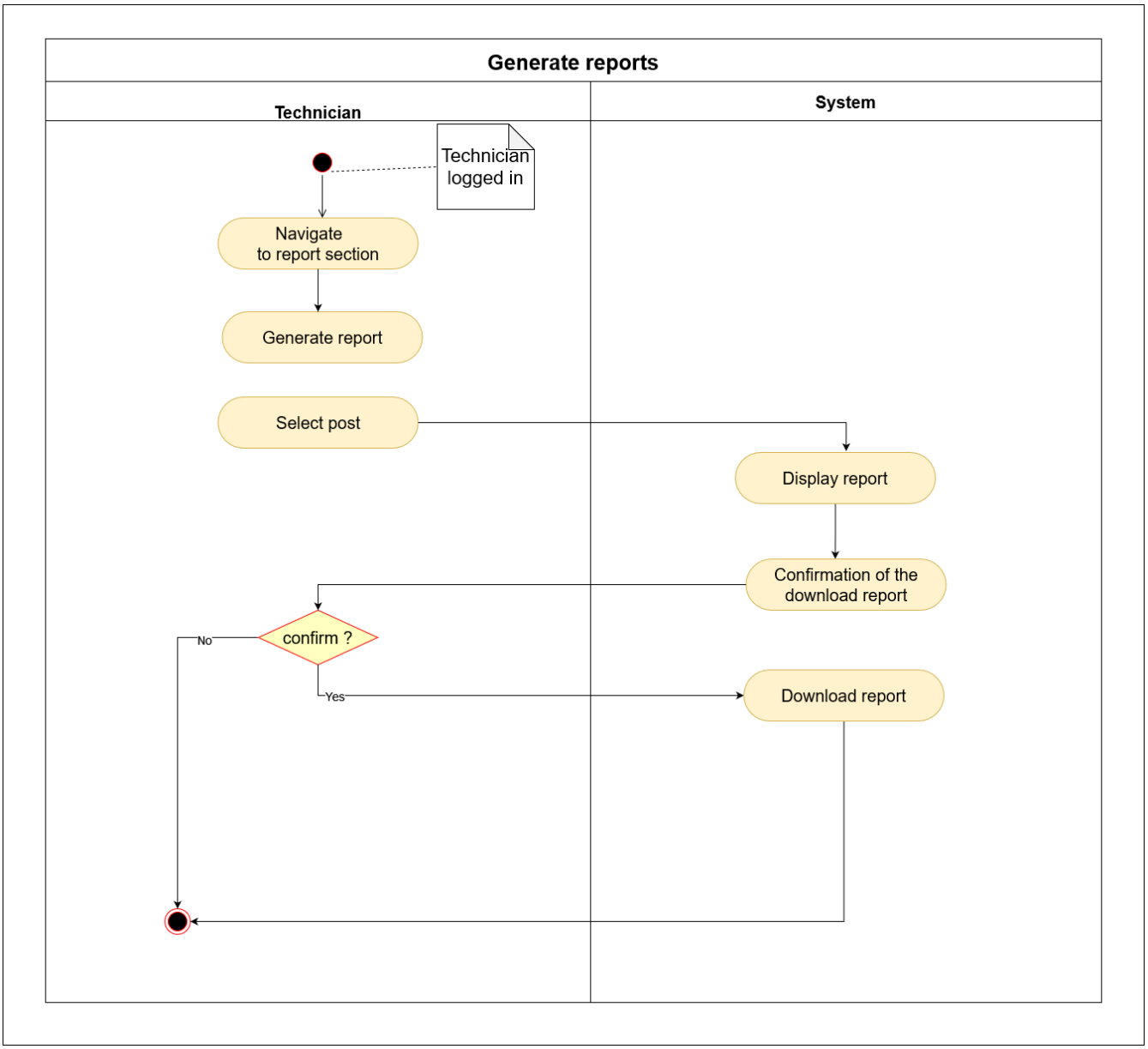
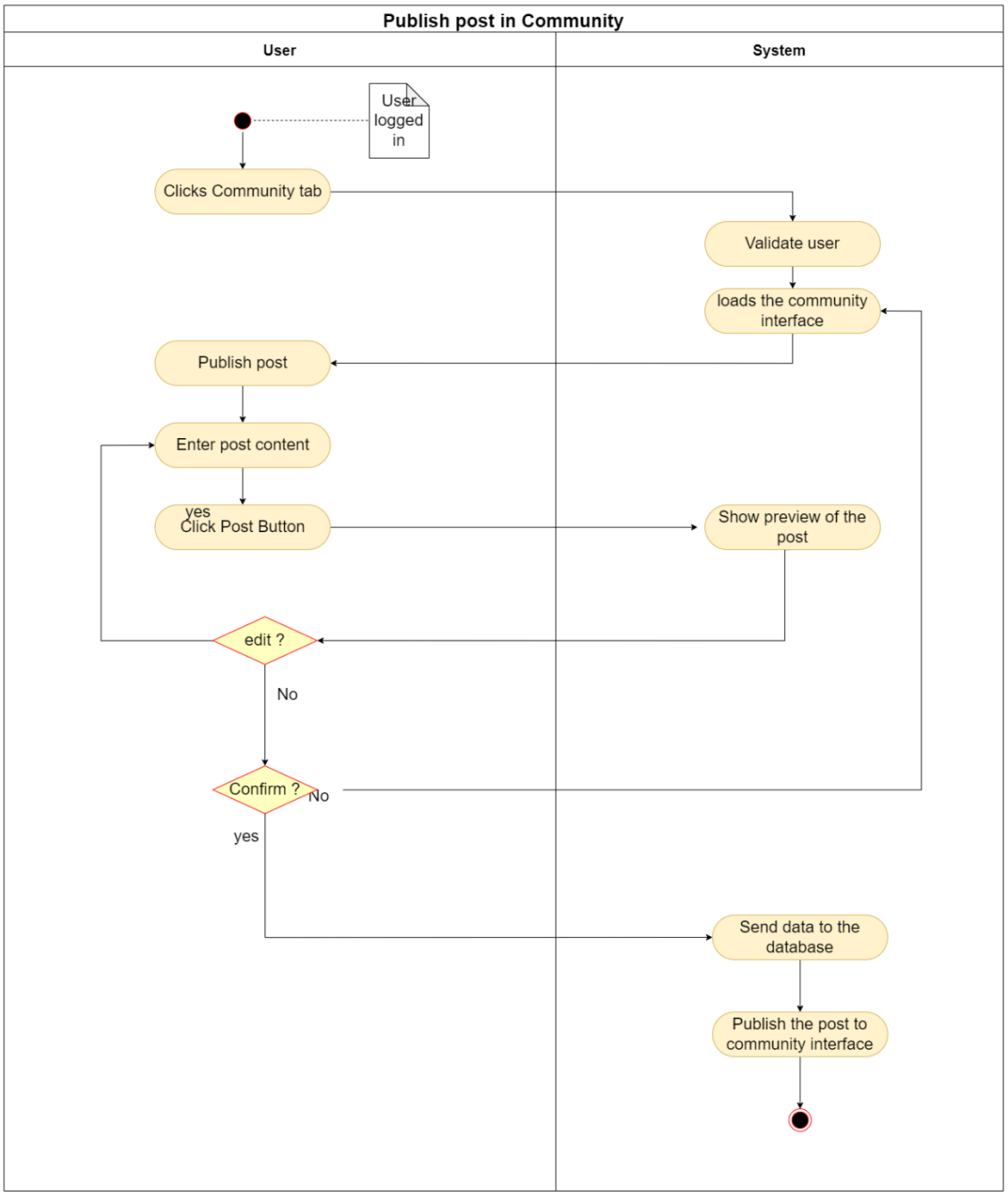


Figure 18: Publish post in Community



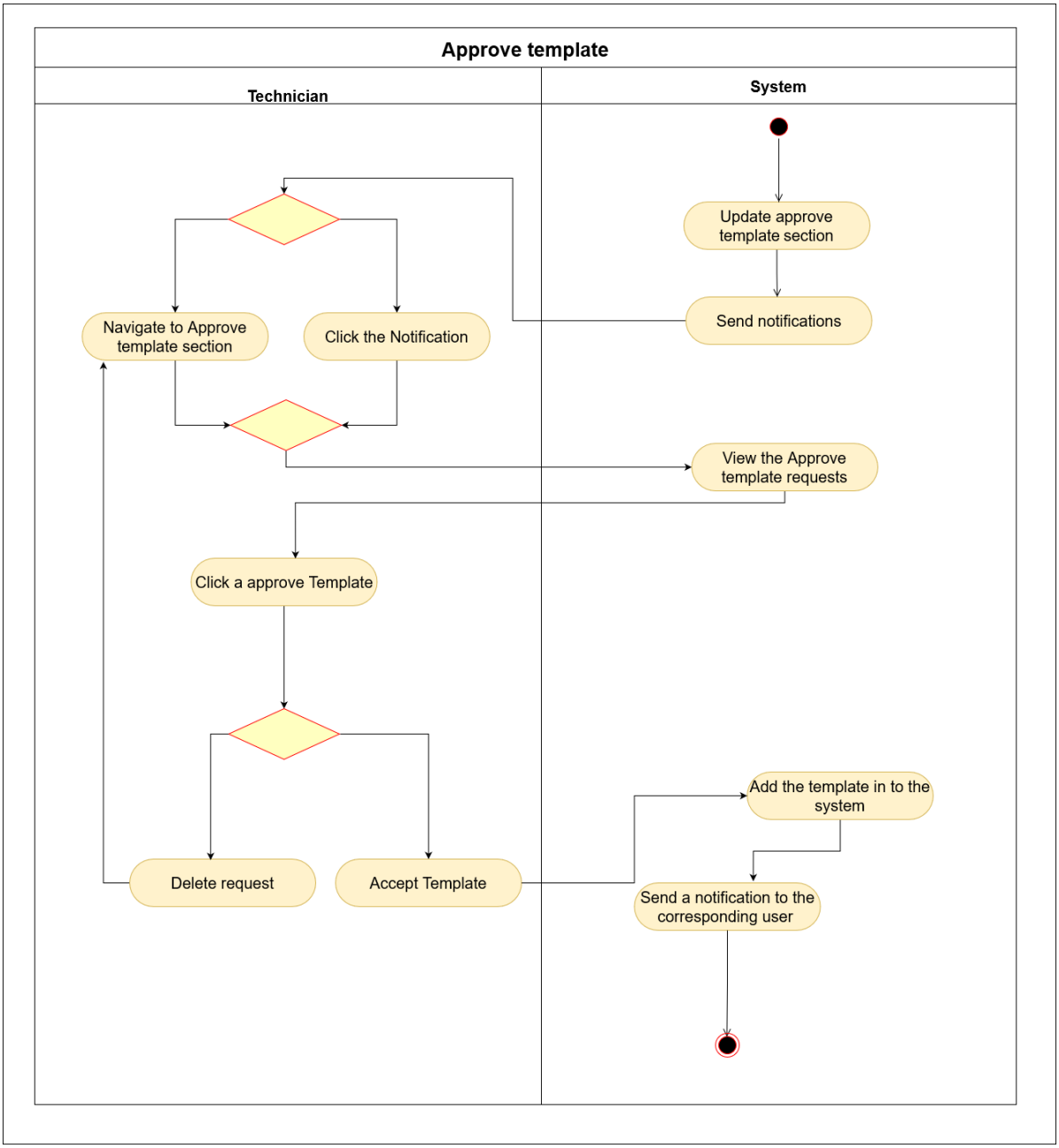


Figure 19 : Approve Template

Figure 20 : User Account Management

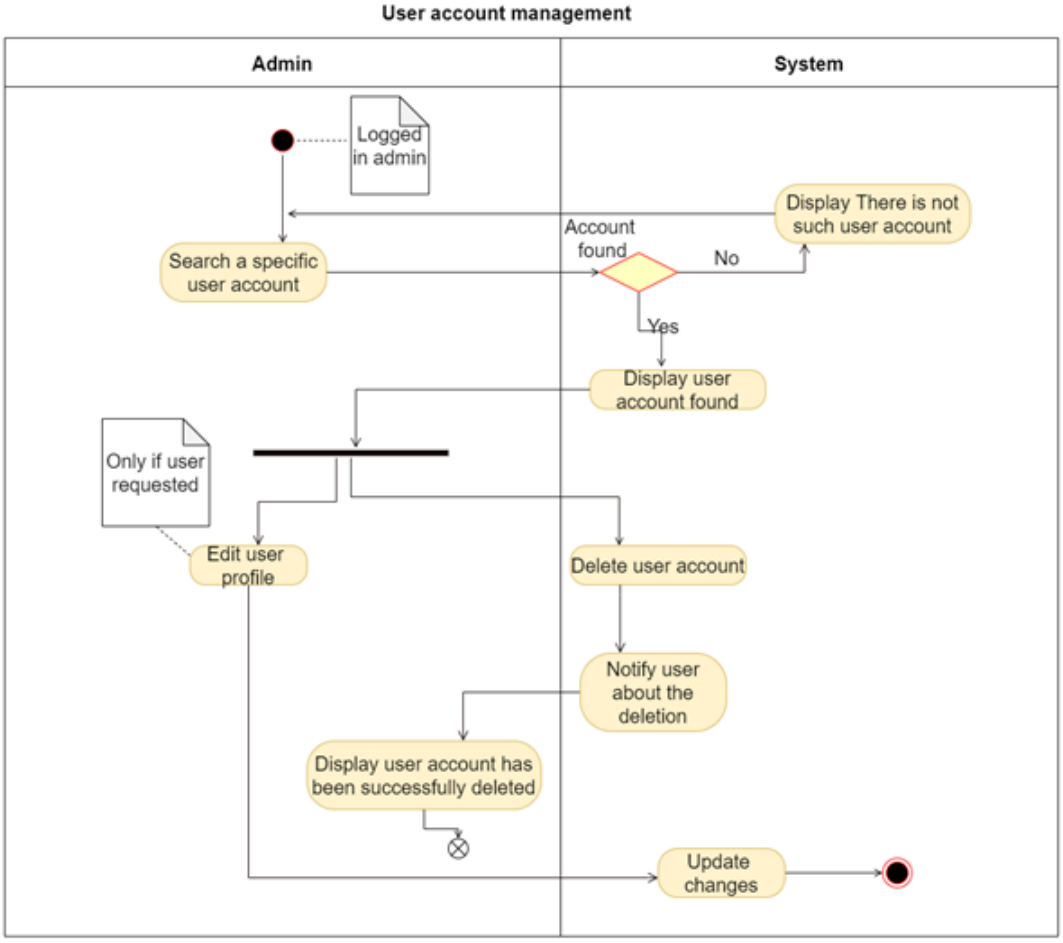


Figure 21 : Verify technician

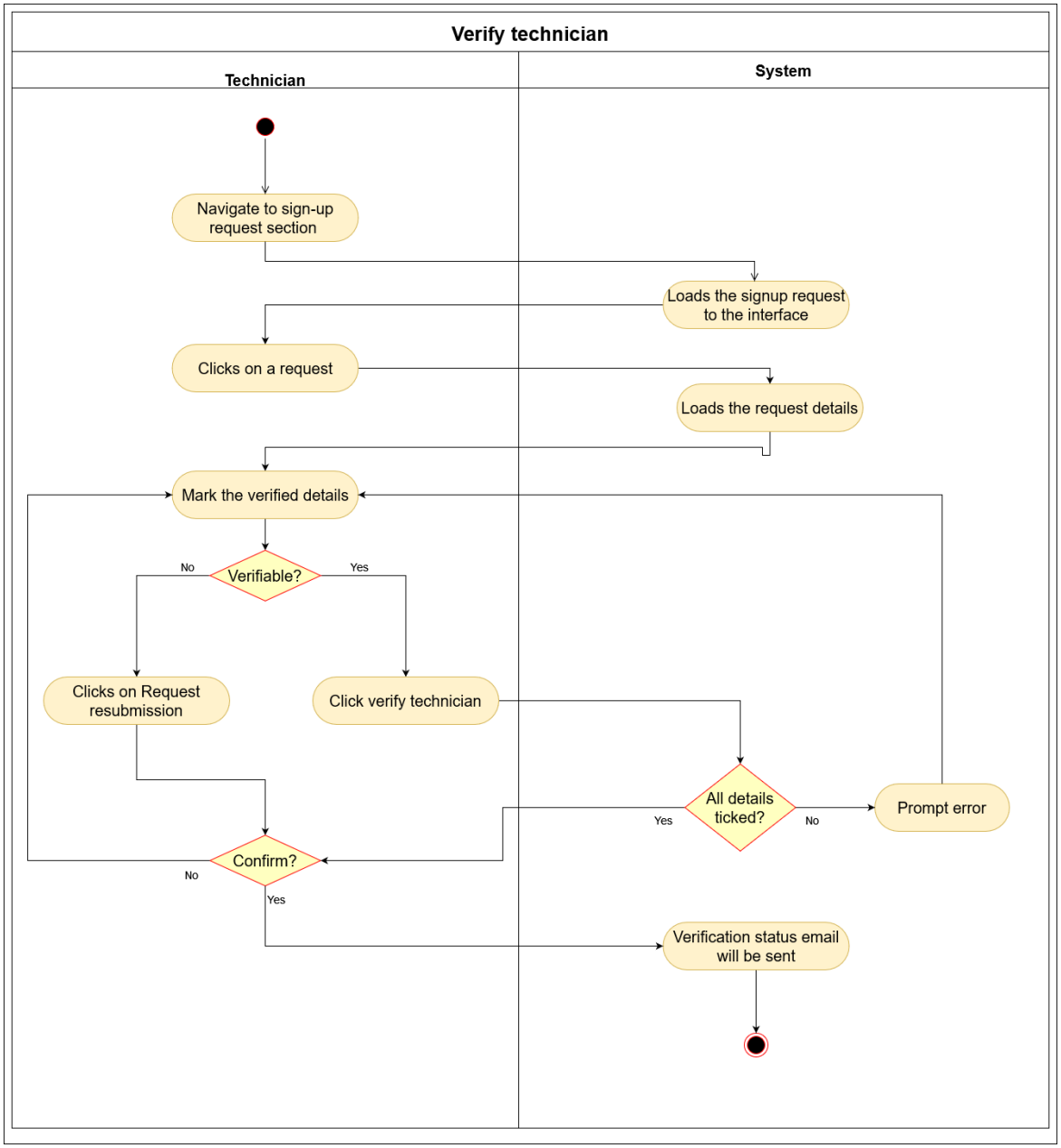


Figure 22 : Register as Technician

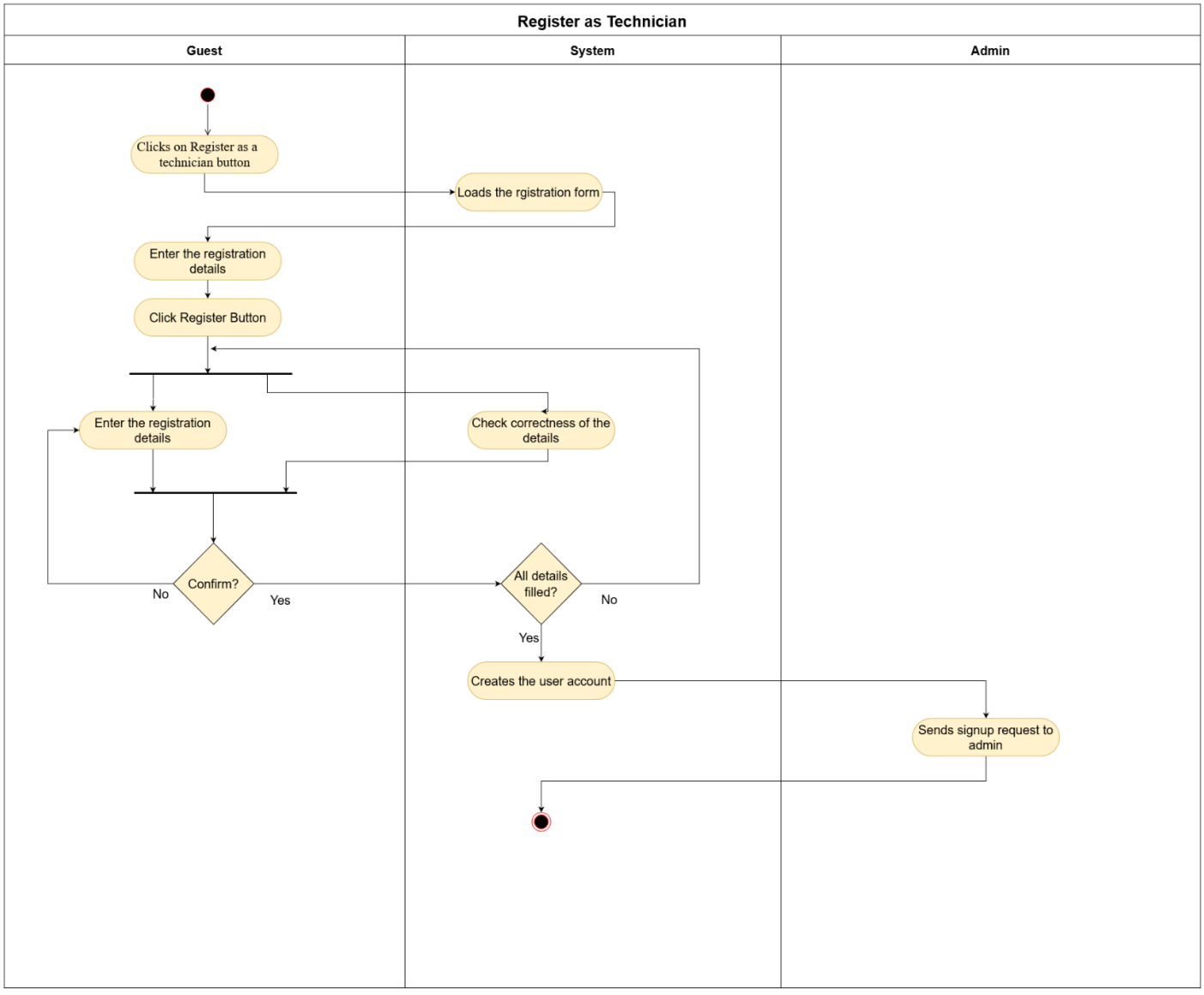


Figure 23 : Register as Property Owner

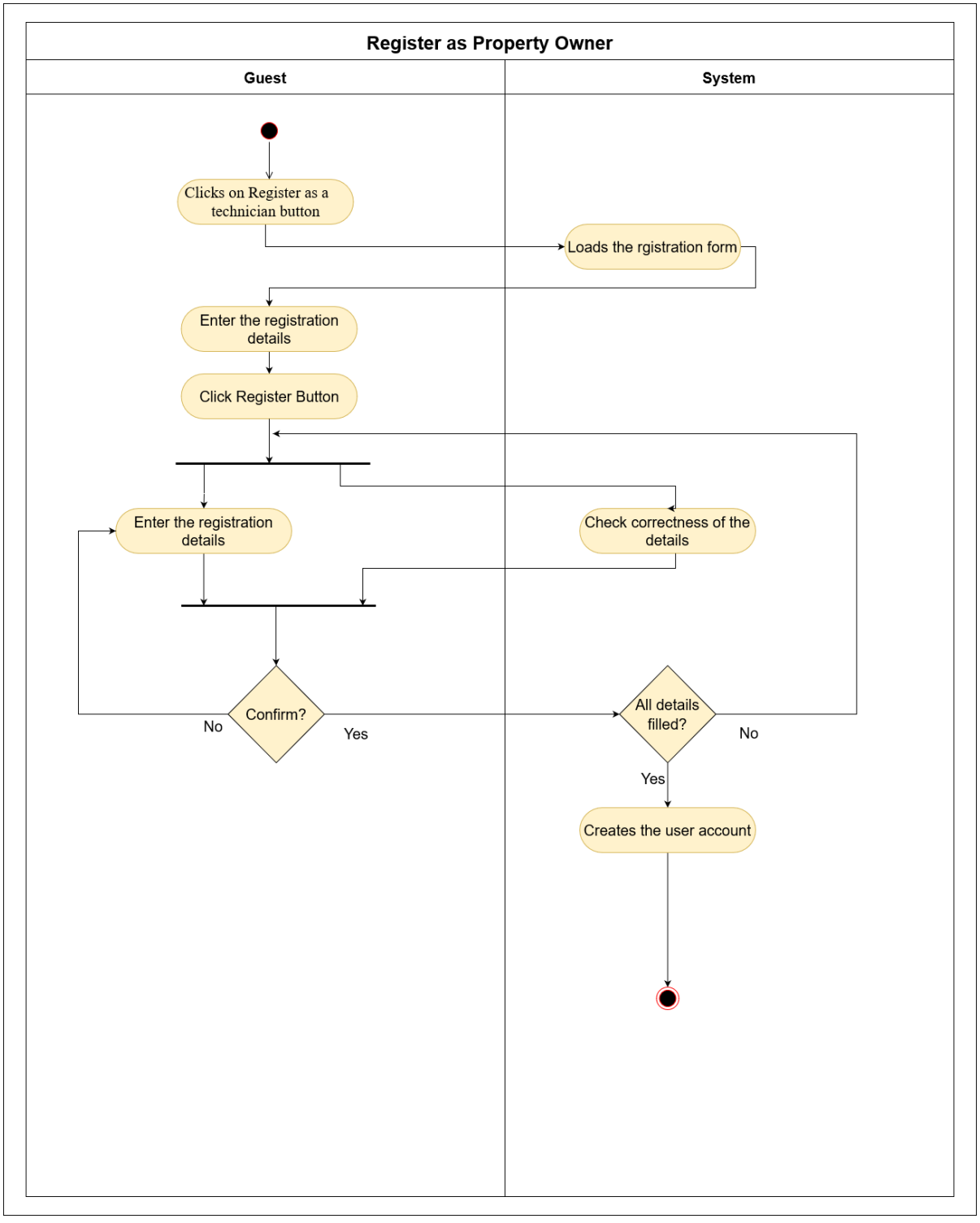


Figure 24 : View Community: Guest

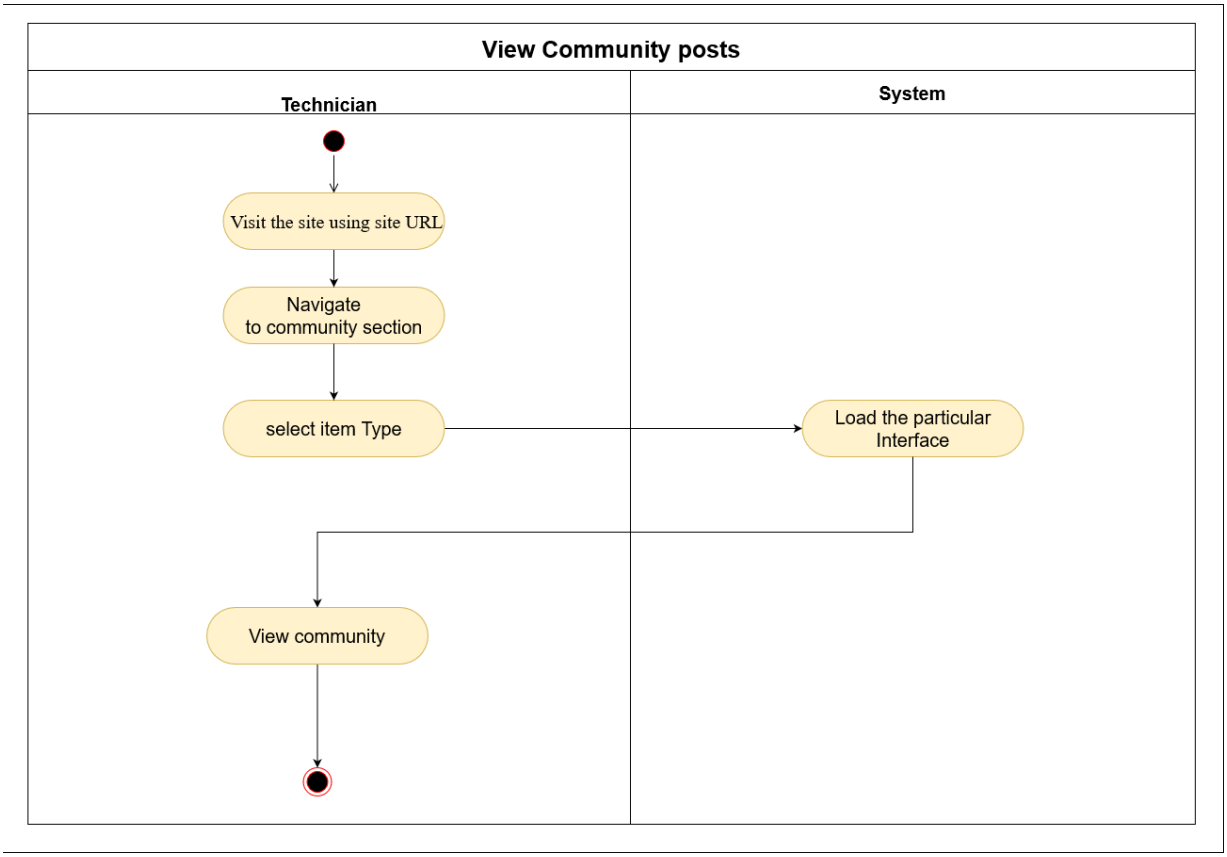
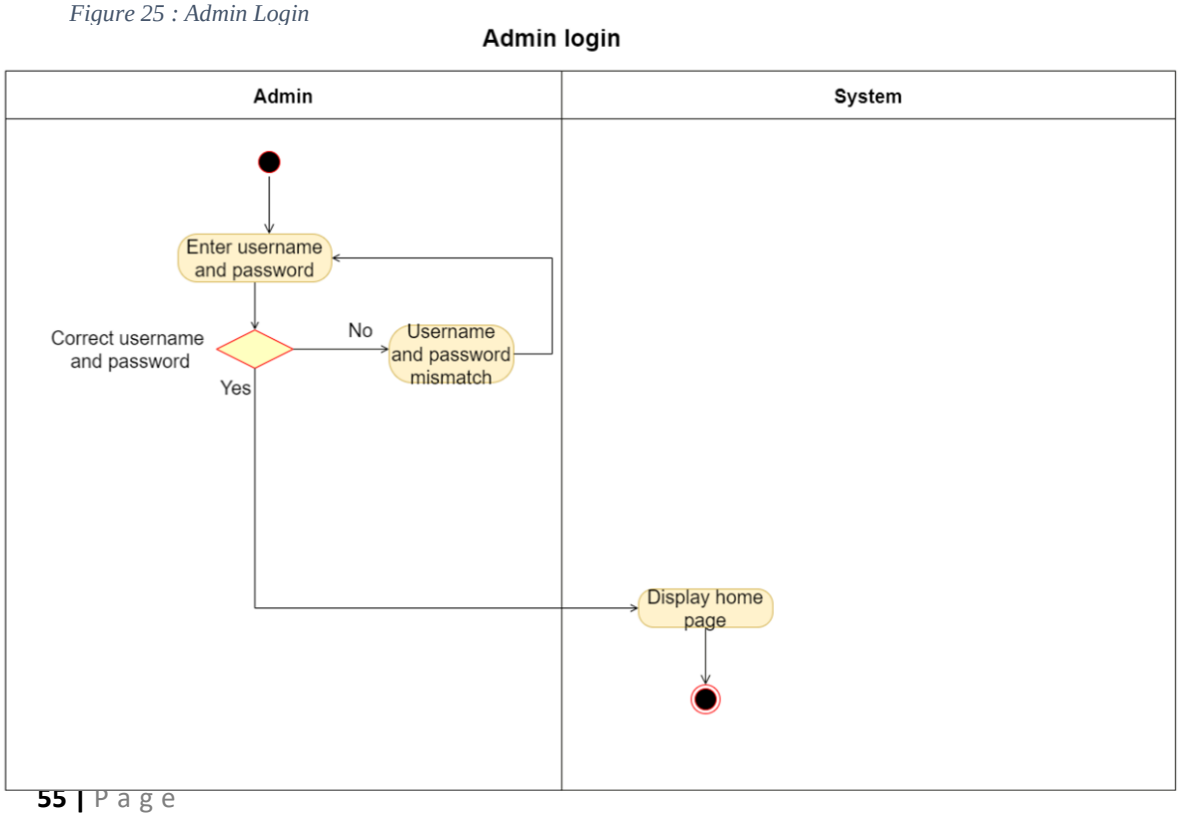


Figure 25 : Admin Login



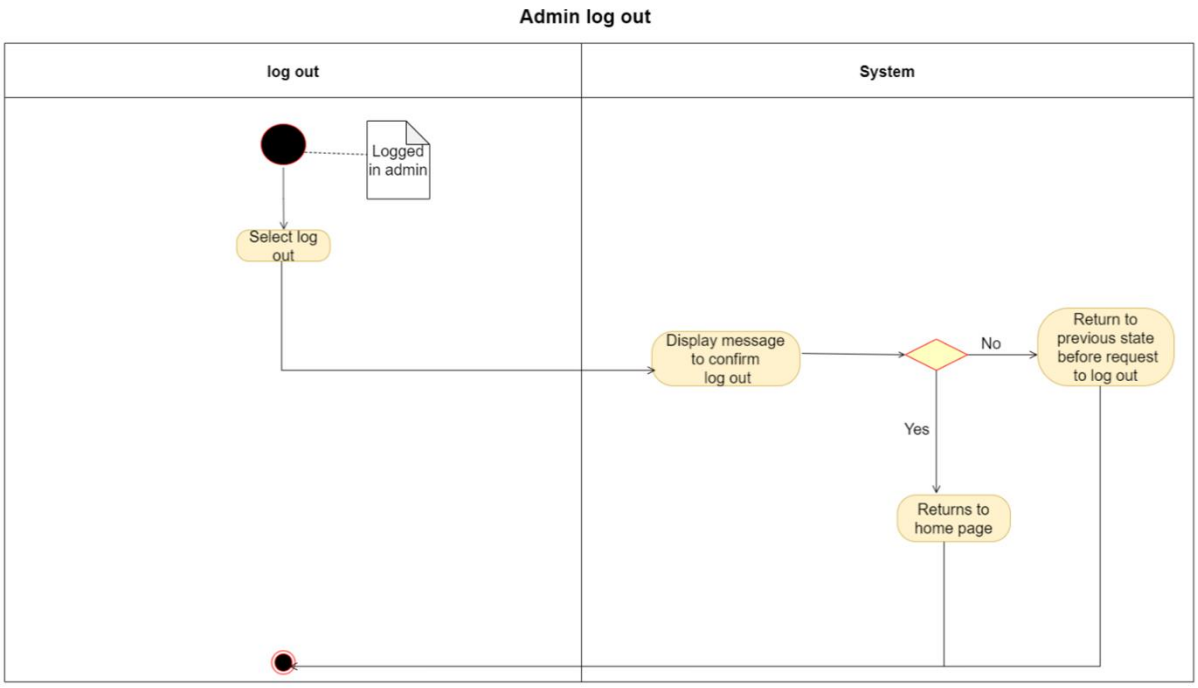


Figure 26: Admin Logout

8. Quality Attributes

8.1 Identification of quality attributes

Below are the quality attributes that are identified to be in our system.

- Security
- Availability
- Usability
- Performance
- Modifiability
- Testability

The quality attribute scenarios for above attributes are as follows.

8.1.1 Security

- The system contains some sensitive information about their users as well as user's household and their vehicles such as names, mobile numbers, emails, vehicle numbers. Therefor achieving security quality attribute is important in this system.
- How to achieve,
    - Each user shall have a user account to log in.
    - Passwords are never viewable after its added. (Password encryption)
    - User must verify using email OTP when resetting the password.
    - The sensitive information will not be shown to other users in community.

8.1.2 Availability

- Availability is the item that describes the ability of the system to carry out its functionalities whenever it is requires.
- The aim of the development is to make this web application available for the user 24x7. But in certain instances, the system will be unavailable to the user due to the system maintenance and system failures.
- These system failures may include Omission of data, crash of the system, incorrect timing and incorrect responses. These failures can occur under normal operation, Startup.
- How to achieve,
    - If the system is unavailable due to a fault, the system should be repaired and available again within 2 hours.
    - The plan is to make the Maintenance tasks at midnight. And since this web application is mostly used by people in Sri Lanka. The business drawbacks will be minimum.

8.1.3 Performance

Performance of the system is concerned with response times and shows the response of the system to specific operation actions for a certain period of time.
We can use the following ways to achieve performance,

- Using efficient database queries to minimize the data retrieving and storing time.
- Using optimization algorithms to reduce time complexity
- Optimize the code as much as possible.
- Optimizing image size used in the web application.
- Using minimal layout for the user interfaces.

8.1.4 Usability

Usability is concerned with how easy to the end user to learn the features of the system and how efficiently the end user can use the system. Usability is improved by end user to allow respond appropriately. For instance, strategy like cancel helps the user either fix mistakes or be more productive. How to achieve,

- Using simple and user-friendly user interface.
- Using mobile phone, end users can easily log into the system.
- Where possible, instructions will be included in the interface.
- Notification system is efficient way for end users to get sufficient information.
- Community chat function helps to improve the interaction among users.

8.1.5 Testability

Ease with which the software can be made to demonstrate errors in the system through testing. Tester can be human or automated tester.
We can use the following ways to achieve performance,

- After each completion of coding increment, corresponding component get tested.
- Using a component-based architecture such as MVC architecture each component can be tested individually

8.1.6 Modifiability

A modification may be made to the platform, capability, or function that the system computes. It takes longer and costs more money to adjust if the user needs to.
We can use the following ways to achieve performance,

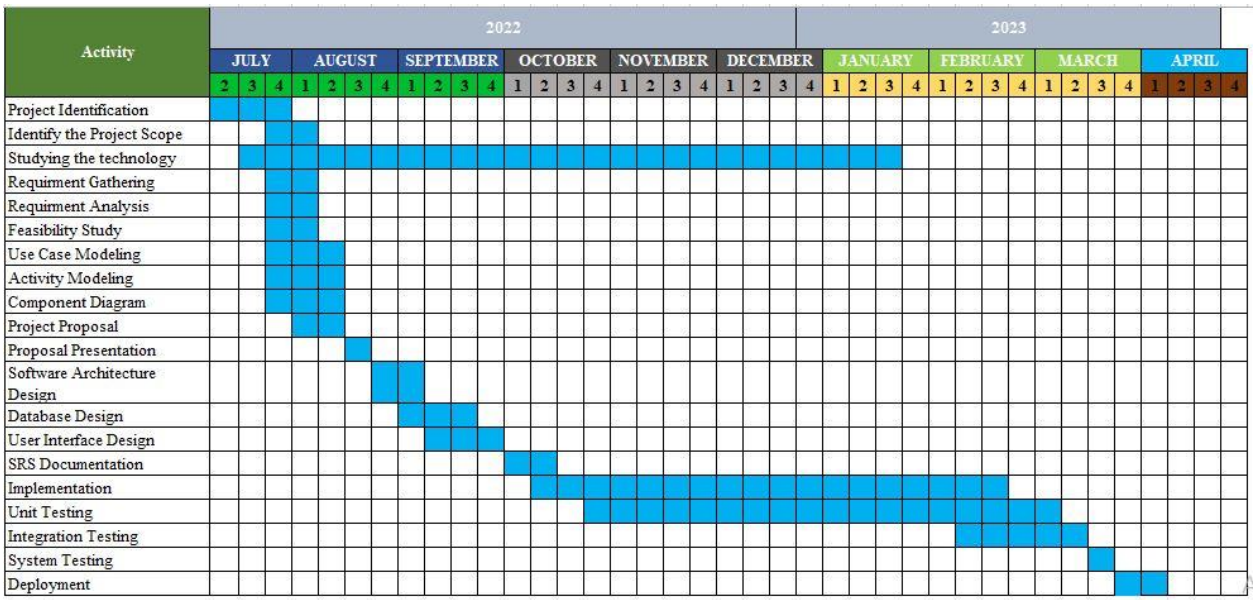
- Reducing coupling between components such that dependencies between modules are minimized as much as possible. Therefore, we are planning to use component-based architecture for the development.

9. Technologies to be used

- Frontend Development: HTML5, CSS3, JavaScript
- Backend Development: PHP, MySQL
- Web development environment: Visual Studio Code, PhpStorm
- Project Management tool: Trello
- Communication: Discord, zoom
- Webserver software - XAMPP
- Version controlling - GitHub
- Diagramming - Draw.io
- Documentation - Google docs, MS Office 365

10. Project Timeline

We are planning to use the "waterfall model" as the software process model for our project. The following Gantt Chart represents the timeline of the proposed system. It is a tentative timeline. The allocated time period for the project is one academic year, but it is desirable to reach the project completion well in advance due to the uncertain circumstances taking place around the country currently.



Declaration

We as members of the project titled **UPKEEP**, Certify that we will carry out this project according to the guidelines provided by the coordinators and supervisors of the course as well as we will not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university. To the best of our knowledge and brief, the project work will not contain any material previously published or written by another person or ourselves except where due reference is made in the text of appropriate places.

Student Name	Index No.	Signature
1) M.K. Pramod	20001371	
2) P.G.N.R. Balangoda	20000197	
3) M.R.D. Siriwardhana	20001802	
4) H.G.S. Hasinika	20000731